Augmented Paths and Reodesics for Topologically-Stable Matching

YUSUF SAHILLIOĞLU, Middle East Technical University, Turkey
DEVIN HORSMAN, Arcturus Studios, Canada

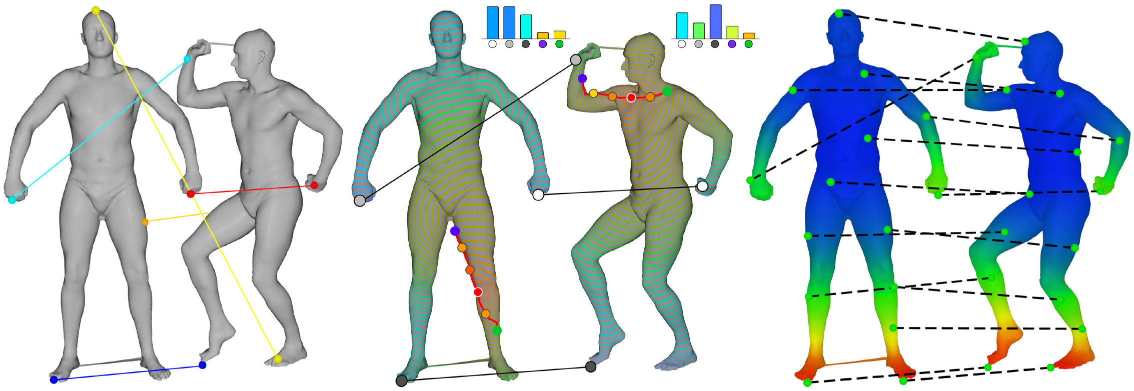


Fig. 1. Overview of our approach. Brute-force map between shape extremities (left) helps us augment the paths in comparison (middle). Sparse map obtained by casting fuzzy votes over augmented paths guides the dense matching (right). Mismatches in the brute-force map (head-foot) can be tolerated. Heat sent from a given path vertex (red) to path endpoints (blue-green) and robust matches (black-gray-white) are visualized via charts where bars are colored according to the heat color. Note that that although path lengths are similar, augmented path information is not, which would prevent voting for this incompatible pair.

We propose a fully-automatic method that computes from scratch point-to-point dense correspondences between isometric shapes under topological noise. While relying on pairwise distance preservation constraints is common and generally sufficient to handle isometric deformations, presence of topological noise needs further actions that we present as our main contributions. First, instead of comparing distances over two paths on two input surfaces, we cast fuzzy votes at the path endpoints based on topologically-robust heat diffusion from path vertices. Second, we make the matching even more stable to topological noise by introducing the so-called reodesics, which are locally shortest geodesics that go through robust matches. In addition to the five standard datasets for isometric shape correspondence with and without topological noise, we employ and release a sixth one geared specifically towards topological noise evaluation with ground-truth information. We demonstrate our qualitative and quantitative advantages over seven recent state-of-the-art methods on these six datasets.

CCS Concepts: • **Computing methodologies → Shape analysis;**

Additional Key Words and Phrases: Shape correspondence, isometry, topological noise, dense matching

ACM Reference format:
Yusuf Sahillioğlu and Devin Horsman. 2022. Augmented Paths and Reodesics for Topologically-Stable Matching. *ACM Trans. Graph.* 42, 2, Article 17 (October 2022), 15 pages.
https://doi.org/10.1145/3554978

1 INTRODUCTION

Correspondence information between shapes is very important in computer graphics and geometry processing in order to enable useful applications. Solution to the shape correspondence problem gives the desired answer in the form of point-to-point maps. Being well studied for noise-free shapes that differ by rigid and non-rigid transformations, the problem is yet at its infancy for handling the topological noise which is a frequent occurrence while capturing 3D data. We address this challenging scenario to provide point-to-point dense maps between non-rigid isometric shapes under topological noise.

The task of non-rigid shape correspondence is difficult as it requires a search over the entire target shape to match a given point in the source shape. The target shape exhibits rigid motions, isometric deformations such as bending, mild stretching, and in our case extra topological noise. In order to deal with this large non-linear search space we propose a coarse-to-fine matching strategy (Figure 1) where we first obtain a brute-force map between shape extremities. We then use a subset of this map to match sparsely and uniformly distributed samples via voting. We finally obtain our smooth dense map between every vertex in the guidance of

This work was partially supported by TUBITAK under the project EEEAG-119E572.
Authors' addresses: Y. Sahillioğlu, Middle East Technical University, Computer Engineering Department, Ankara, 06800, Turkey; email: ys@ceng.metu.edu.tr; D. Horsman, Arcturus Studios, 16 Wentworth Street, Dartmouth, NS, B2Y 2S6, Canada; email: devin.horsman@arcturus.studio.

0730-0301/2022/10-ART17 $15.00
https://doi.org/10.1145/3554978

the sparse map. This roadmap involves several novel contributions for the matching process:

- ***Augmented paths:*** sparse matching compares two paths augmented by heat vectors defined on uniformly distributed path vertices.
- ***Reodesics:*** geodesic distances are made robust to topological noise by making them go through special samples.
- ***Consistency-based filtering***: dense matching ensures that symmetric maps are consistent.
- ***Symmetric flip alleviation:*** sparse matching alleviates the symmetric flip problem by adjusting [Sahillioğlu and Yemez 2013a] conveniently.
- ***Speed:*** full dense map is computed for a 12.5K-vertex pair in 12 seconds from scratch without any initial information and highly parallelizable.
- ***Scalability:*** Execution time grows linearly, e.g., 23 seconds to match a 28K-vertex pair from scratch.
- ***Versatility:*** partial matching scenario, in addition to the main topic of matching under topological noise, is handled under certain conditions without any modification to the algorithm.

2 RELATED WORK

The shape correspondence problem is related to many other geometry processing problems such as registration [Sahillioğlu and Kavan 2021], morphing [Aydınlılar and Sahillioğlu 2021], exploration [Kim et al. 2012], and information transfer. It consequently comes in different settings depending on the application. All these settings are detailed in surveys [Sahillioğlu 2020; van Kaick et al. 2011], yet we here address the recent work concerning our method, namely isometric dense correspondence with and without topological noise.

One can investigate isometric dense correspondence methods under deformation-based and similarity-based categories. The former deforms one shape directly towards the other [Eisenberger et al. 2020; Ezuz et al. 2019; Maron et al. 2016; Schmidt et al. 2019] or deforms both towards a common intermediate domain such as sphere [Aigerman et al. 2017; Baden et al. 2018], plane [Aigerman and Lipman 2015; Kim et al. 2011; Weber and Zorin 2014], and hyperbolic surface with cone singularities [Aigerman and Lipman 2016]. The naïve closest point matches in the deformed configurations then reveal the desired correspondences. One can remove the non-rigid bending transformation from consideration and match rigidly different shapes, a case that would still be isometric but easier to deal with Choy et al. [2020] and Wang and Solomon [2019]. Even if both intermediate maps locally minimize the intrinsic distortion, there is potentially high distortion in the composed map, which would make the direct deformation approach more preferable. Regardless of whether direct or indirect deformation is chosen, this category tends to be less efficient than the similarity-based approaches like ours due to the cost of a high-quality nonlinear deformation energy evaluation.

The recent literature in the latter category is dominated by the introduction of the revolutionary functional maps framework in 2012 [Ovsjanikov et al. 2012], which replaces the traditional point-to-point correspondences by matching real-valued functions over surfaces. Their particular choice of low-frequency Laplace-Beltrami eigenfunctions for real-valued functions puts emphasis on smoothness whereas [Azencot and Lai 2021] aims for a spectrum that allows for better feature matching. Although functional maps are conveniently available for fast linear-algebraic analysis and manipulation, they are prone to conversion errors when recovering the point-to-point correspondence from the optimal functional mapping. Significant work has already been carried out in order to improve this conversion step using isometric alignment [Rodola et al. 2017], smoothness-based deblurring [Ezuz and Ben-Chen 2017], linear assignment [Vestner et al. 2017b], quadratic assignment [Vestner et al. 2017a], vector field flow [Azencot et al. 2016; Corman et al. 2015], and fast Sinkhorn filters [Pai et al. 2021]. A different branch of follow-up work improves functional map estimation by formulating the descriptor preservation constraints via commutativity with an underlying map [Nogneng et al. 2018; Nogneng and Ovsjanikov 2017; Wang et al. 2018], and by using deep neural networks [Donati et al. 2020; Litany et al. 2017; Sharma and Ovsjanikov 2020] or iterative spectral up-sampling techniques [Huang et al. 2020; Melzi et al. 2019a]. Ren et al. [2021] proposes a discrete solver to obtain a stronger link between optimized properties of functional and point-to-point maps.

Other map representations can also be used to seek similarity-based correspondences, e.g., probabilistic soft maps [Solomon et al. 2012], transport plans [Mandad et al. 2017], genetic chromosomes [Edelstein et al. 2019; Sahillioğlu 2018], local reference frames [Melzi et al. 2019b], and point-to-point [Ovsjanikov et al. 2010]. Our choice in this paper is the intuitive point-to-point map representation.

For the challenging and less-studied topological noise case, we see most papers in the similarity-based category. Due to the ambiguities involved in this setting, e.g., a small tunnel may be a shortcut noise or a model feature, these methods and our own only provide robustness up to a certain extent. Solomon et al. [2016] employs entropy-regularized version of the Gromov–Wasserstein objective function, whereas [Rodolà et al. 2014] and [Wei et al. 2016] prefer learning-based approaches that use decision trees and classical extrinsic convolutional neural networks, respectively. Functional maps are made more resilient to this type of noise by the introduction of topological constraints [Poulenard et al. 2018] and geometric matrix completion formulation [Kovnatsky et al. 2015]. Replacing geodesics with spectral diffusion distances provides a minor improvement for this noisy setting [Bronstein et al. 2010; Lahner et al. 2016; Sahillioğlu and Yemez 2012; Sharma et al. 2011]. Our principled solution based on path augmentation with heat diffusion and robust geodesics through robust matches provides a more stable solution than these alternatives do.

3 PROBLEM STATEMENT AND OVERVIEW

We aim to establish the optimal dense map ϕ^* between two isometric (or nearly isometric) shapes under topological noise. To this end, we first obtain the map ϕ^*_{brute} between M farthest point samples on each mesh [Eldar et al. 1997] via brute-force search based on our reodesic metric. A subset of ϕ^*_{brute} leads to the sparse map ϕ^*_{sparse} between $N > M$ farthest point samples according to the proposed heat vectors. Reodesics are also active in this stage while voting, filtering and during map evaluation. A refined subset of

ALGORITHM 1: Our matching algorithm for isometric shape correspondence under topological noise.

```
Input: $S$ and $T$ s.t. $|S| = |T| = N$, samples on two meshes
Output: $\phi^*_{\text{brute}}$, $\phi^*_{\text{sparse}}$: $S \rightarrow T$, and $\phi^*$ between all vertices

  Let $S_M$ and $T_M$ be the set of first $M$ samples in $S$ and $T$, respectively
  for $i \leftarrow 1$ to $M!$ do
      $\phi_{\text{test}} : S_M \rightarrow T'_M$ where $T'_M$ is the $i$th permutation of $T_M$
      if $\mathcal{D}_{\text{iso}}(\phi_{\text{test}}) <$ minIso then
          minIso $= \mathcal{D}_{\text{iso}}(\phi_{\text{test}})$ and $\phi^*_{\text{brute}} = \phi_{\text{test}}$      ▷ $\phi^*_{\text{brute}}$ ready
      end if
  end for

  for each triplet of matches $\{(a_k, b_k)\} \in \phi^*_{\text{brute}}$ do
      for each path $p_{il} = s_i \leftrightarrow s_l \in S$ do
          overallDissim = 0                ▷ Select the most similar path for $p_{il}$
          for each path $q_{jm} = t_j \leftrightarrow t_m \in T$ do
              if not filtered out then                              ▷ Section 4.8
                  for each path vertex $p_u \in p_{il}$ and $q_u \in q_{jm}$ do
                      overallDissim $+=$ $||\mathbf{h}_{p_u} - \mathbf{h}_{q_u}||_{L_\infty}$     ▷ Section 4.6
                  end for each
                  if overallDissim $<$ minOvDissim then
                      minOvDissim = overallDissim
                      $t_g = t_j$ and $t_h = t_m$
                  end if
              end if
          end for each
          $\delta =$ minOvDissim$/U$                              ▷ $U$ is # path vertices
          $s_i$.votes$[t_g] += 1 - \delta$ and $s_l$.votes$[t_h] += 1 - \delta$       ▷ Voting
      end for each
      Get $\phi_{\text{sparse}}$ using the highest votes                    ▷ Section 4.6.1
      if $\mathcal{D}_{\text{iso}}(\phi_{\text{sparse}}) <$ minIso2 then
          minIso2 $= \mathcal{D}_{\text{iso}}(\phi_{\text{sparse}})$
          $\phi^*_{\text{brute3}} = \{(a_k, b_k)\}$ and $\phi^*_{\text{sparse}} = \phi_{\text{sparse}}$     ▷ $\phi^*_{\text{sparse}}$ ready
      end if
  end for each

  for each source vertex $v$ do
      for each target vertex $w$ do
          if not filtered out then                                  ▷ Section 4.8
              if $||\mathbf{h}_v - \mathbf{h}_w||_{L_\infty} <$ minOvDiss then        ▷ $\phi^*_{\text{sparse}}$ in use
                  minOvDiss $= ||\mathbf{h}_v - \mathbf{h}_w||_{L_\infty}$
                  $w^* = w$                          ▷ most similar vertex for $v$
              end if
          end if
          Match $v$ to $w^*$
      end for each
      $\phi^* = \phi^* \cup (v, w^*)$                                      ▷ $\phi^*$ ready
  end for each
```

ϕ^*_{sparse} is then used to produce the desired ϕ^* between all V vertices, guided by reodesics and heat vectors. The entire algorithm is outlined as a pseudocode in Algorithm 1.

4 THE METHOD

4.1 Reodesics

We introduce **robust geodesics**, **reodesics** in short, that are more stable than the regular minimal geodesics under topological noise. Reodesics are locally shortest but not necessarily globally shortest, which is a nice property when dealing with topological noise because noisy shortcut tunnels always shorten the paths so we need elongated paths.

4.2 Reodesics for a Single Shape

In order to define the reodesic distance between two vertices s_i and s_l, we need a through-vertex a on the surface. Once we compute the minimal geodesics $g(.,.)$ from a via Dijkstra's shortest paths algorithm, we compute the reodesic distance $d(.,.)$ as:

$$d(s_i, s_l) = g(s_i, a) + g(a, s_l) \tag{1}$$

While the minimal geodesics are computed from source s_i to all other vertices in $O(VlogV)$ time, reodesics require merely $O(V)$ time. This is because the through-vertex is selected from the sample set for which geodesic distances are already computed [Eldar et al. 1997]. Using the precomputed geodesics in Equation (1) in $O(1)$-time leads to the $O(V)$ complexity while setting reodesic distances from s_i to all other V vertices, i.e., reodesic computation brings no extra cost.

4.3 Reodesics for a Shape Pair

A trivial extension of reodesics to a shape pair to be matched requires a robust map $\phi_{\text{robust}} = \{(a_k, b_k)\}$ between the two shapes:

$$\begin{aligned} d(s_i, s_l) &= \max_k \{g(s_i, a_k) + g(a_k, s_l)\} \\ d(t_j, t_m) &= \max_k \{g(t_j, b_k) + g(b_k, t_m)\} \end{aligned} \tag{2}$$

Taking the maximum of k distance candidates elongates the paths consistently on two shapes and the consistent pairwise distances can now be compared for shape matching purposes (Equation (7)). For ϕ_{robust}, we use the first three matches of the current permutation during the brute-force map computation (Section 4.5) and the current triplet $\phi_{\text{brute3}} \subset \phi^*_{\text{brute}}$ during the sparse map computation (Section 4.6). ϕ_{robust} is the winner triplet ϕ^*_{brute3} during the dense map computation (Section 4.7).

In Figure 2, we visualize the computation of reodesics on two shapes that include shortcut tunnels (also on two noise-free shapes at the bottom right corner). We can see its advantage over minimal geodesics as well as spectral distances that are advertised to be topology-aware.

We, however, may still experience problems with this trivial reodesic extension for the topologically noisy configurations since one of the two reodesic paths may involve a shortcut tunnel, as depicted in Figure 3. Since it is impossible to solve this problem with pre-computed reodesics, we switch to the so-called on-the-fly reodesics which are computed right before they are needed, e.g., $d(s_i, s_l)$ in Equation (8) is computed using the optimal through-vertex a_{k^*} whose correspondence b_{k^*} makes $d(t_j, t_m)$ as close as possible to the former. In other words, we set:

$$k^* = \underset{k}{\text{argmin}} \{|g(s_i, a_k) + g(a_k, s_l) - g(t_j, b_k) - g(b_k, t_m)|\} \tag{3}$$

$$d(s_i, s_l) = g(s_i, a_{k^*}) + g(a_{k^*}, s_l) \tag{4}$$

Note that this new on-the-fly definition of reodesics is insensitive to the robust map choice. For example, in Figure 3, close distances $d(s_i, s_l) = g(s_i, a_3) + g(a_3, s_l)$ and $d(t_j, t_m) = g(t_j, b_3) + g(b_3, t_m)$ would be selected for the left pair and right pair even though their robust maps are different. Note that if a_3 and b_3 were on the left knee for the right pair, then the closest distances would

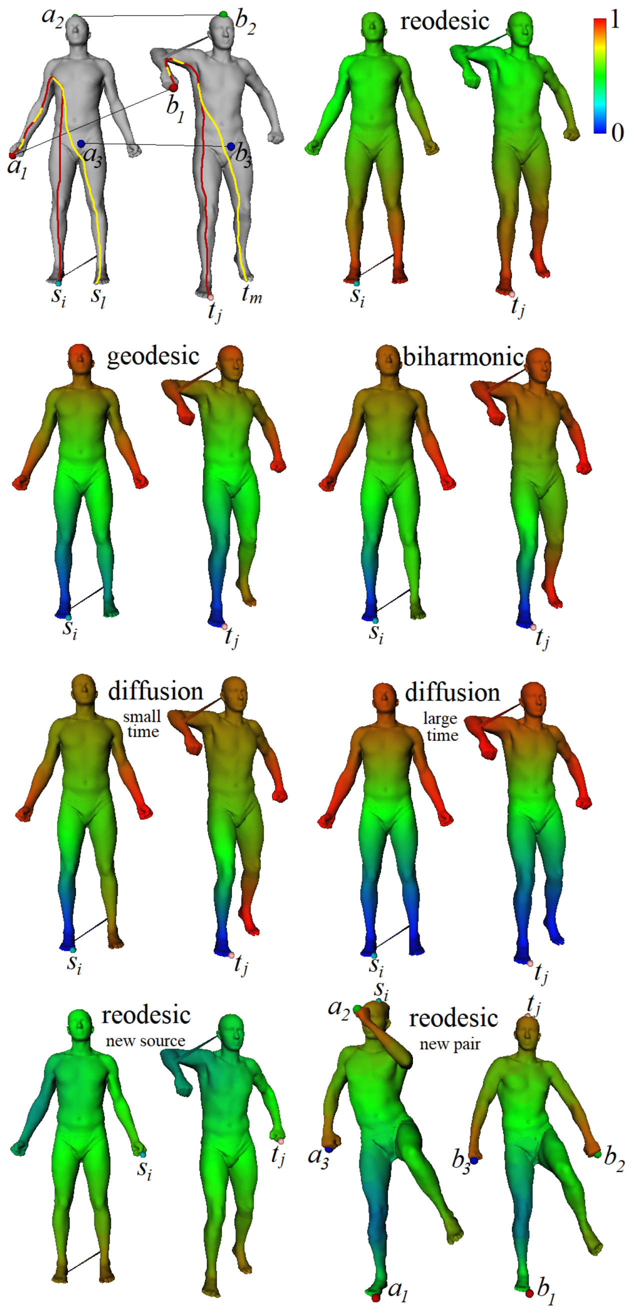


Fig. 2. Right toe (s_i) to left toe (s_l) reodesic uses the longest path through a_1 (union of red and yellow paths). Similarly it goes through b_1 on the other shape from t_j to t_m. Reodesic and competitor spectral distances from right toe to all other vertices are colored from blue (min) to red (max). For the bottom row, reodesic source is left hand (left) and head (right). Robust map for the new pair at right is highlighted via spheres. Note that reodesic is also symmetry-aware (left and right legs are colored differently when head is the source).

use a_2 and b_2 on the head, which would still give a fairly compatible $d(s_i, s_l)$ and $d(t_j, t_m)$ pair. Reodesics here, however, would have been even closer without the extra head-elbow tunnel. We, therefore, conclude that although on-the-fly reodesics are quite powerful to deal with the topological noise, they may still fail especially while using a small robust map on challenging cases with multiple tunnels. This observation leads us to investigate the topologically-robust heat diffusion process while augmenting our paths (Section 4.6), which empirically gave slightly better performance (Figure 5). Besides, during augmentation, for each path vertex, on-the-fly reodesic computation has to go through the entire robust map again to find the most compatible through-vertex, a complexity we avoid by using heat vector that needs to traverse the robust map only once for the whole path. We note that reodesics are still essential to our system (Section 3).

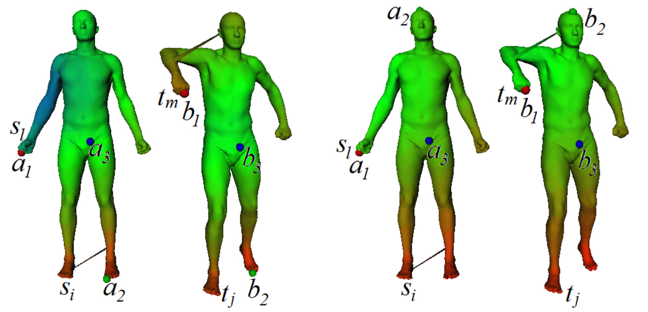


Fig. 3. RGB spheres are 3 robust matches $\{(a_k, b_k)\}$ in use. In the left pair, reodesics from s_i to s_l and from t_j to t_m have the max distances with green through-vertices. The shortcut feet tunnel in the left mesh, however, makes the corresponding reodesic distances inconsistent (blueish vs. reddish) as $g(s_i, a_2) + g(a_2, s_l) < g(t_j, b_2) + g(b_2, t_m)$. This problem disappears with a different robust map (right pair) where $g(s_i, a_2) + g(a_2, s_l) \sim g(t_j, b_2) + g(b_2, t_m)$. A more principled solution is to use on-the-fly reodesics.

4.4 Background for Heat Diffusion

For the sake of completeness and reproducibility, we provide the well-known Laplacian spectrum that made the heat kernel in Sections 4.6 and 4.7 available: $k_\tau(p, r) = \sum_{l=1}^{E} e^{-\tau\lambda_l}\varphi_l(p)\varphi_l(r)$, where we use $E = 15$ smallest non-zero eigenvalues $\{\lambda_l\}$ and eigenfunctions $\{\varphi_l\}$ of the cotangent discretization $\mathbf{L}$ of the Laplace-Beltrami operator:

$$\mathbf{L}_{ij} = \begin{cases} \sum_j w_{ij} & \text{if } i = j \\ -w_{ij} & \text{if } (i, j) \text{ is an edge} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $w_{ij} = (\cot\alpha_{ij} + \cot\beta_{ij})/2$ with α_{ij} and β_{ij} being the angles facing the edge (i, j) [Meyer et al. 2003]. We switch to the generalized eigenvalue problem in order to be able to compute eigenfunctions that are orthonormal with respect to the diagonal area matrix $\mathbf{A}$:

$$\mathbf{L}\varphi_l = \lambda_l \mathbf{A}\varphi_l \quad \rightarrow \quad \mathbf{A}^{-1}\mathbf{L}\varphi_l = \lambda_l\varphi_l \tag{6}$$

where $\mathbf{A}_{ii}$ stores the Voronoi area of the ith vertex, i.e., one third of the sum of the areas of the adjacent triangles. The problem is then solved entirely within the C++ framework using the fast shift-and-invert mode of the Spectra library. Note that biharmonic distance [Lipman et al. 2010] can also be incorporated into our system seamlessly using the same spectrum: $d(p, r) = \sum_{l=1}^{E} (\varphi_l(p) - \varphi_l(r))^2/\lambda_l^2$. We indeed replaced our reodesics with biharmonics (Figure 2) in the entire system but obtained poorer results.

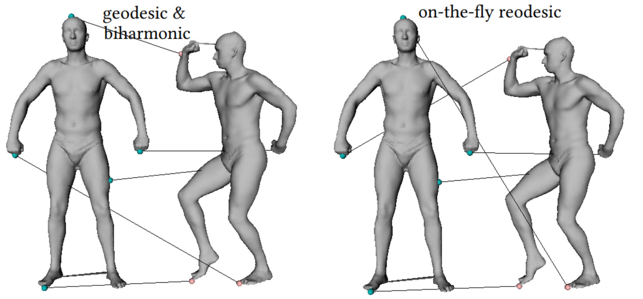


Fig. 4. Low-resolution brute-force map ϕ^*_{brute} fails with minimal geodesic and biharmonic distance (left) but not with our on-the-fly reodesic (right).

4.5 Brute-Force Map

The first M samples of the N farthest point samples are matched by checking $M!$ permutations of target samples $\{t_j\}$ with the source samples $\{s_i\}$ under the following isometric distortion measure $\mathcal{D}_{\text{iso}}$. Note that sampling starts from a vertex that is farthest from an arbitrary initial vertex. Typically, $M = 5$ and $N = 50$.

$$\mathcal{D}_{\text{iso}}(\phi) = \frac{1}{|\phi|} \sum_{(s_i, t_j) \in \phi} d_{\text{iso}}(s_i, t_j) \tag{7}$$

where ϕ denotes the set of correspondence pairs between source and target samples, and $d_{\text{iso}}(s_i, t_j)$ is the contribution of the constituent match (s_i, t_j) to the overall isometric distortion:

$$d_{\text{iso}}(s_i, t_j) = \frac{1}{|\phi'|} \sum_{(s_l, t_m) \in \phi'} (d(s_i, s_l) - d(t_j, t_m))^2 \tag{8}$$

where $\phi' = \phi - \{(s_i, t_j)\}$ and $d(.,.)$ is our novel on-the-fly robust geodesic distance between two vertices on a given surface that we refer to as reodesic for short (Section 4.1). $\mathcal{D}_{\text{iso}}$ is a variant of the measure used in Sahillioğlu and Yemez [2011]. ϕ^*_{brute} is the one-to-one correspondence (bijection) minimizing Equation (7). Note that Equation (7) utilizing on-the-fly reodesics will be used during the upcoming sparse map computation stage as well. A typical improvement due to our on-the-fly reodesics at this stage is shown in Figure 4.

4.6 Sparse Map

In order to be insensitive to the potential sampling inconsistencies due to small M (Figure 7(left)), we compute ϕ^*_{sparse} based on each triplet of matches $\phi_{\text{brute3}} \subset \phi^*_{\text{brute}}$. Each pair (s_i, s_l) looks for the most similar pair (t_j, t_m) by sampling along the the shortest paths $\mathbf{p}_{il}$ (from s_i to s_l) and $\mathbf{q}_{jm}$ (from t_j to t_m), respectively. We then check the similarities of the heat-based, hence topologically-stable, vectors over the augmented paths $\mathbf{p}_{il}$ and $\mathbf{q}_{jm}$.

Let $\mathbf{P}_{il}$ be the set of uniformly sampled U vertices on $\mathbf{p}_{il}$ including s_i and s_l. $\mathbf{Q}_{jm}$ is defined similarly. We use $U = 10$ and define the heat vector $\mathbf{h}_{p_u}$ for each vertex $p_u \in \mathbf{P}_{il}$ as follows:

$$\begin{aligned}\mathbf{h}_{p_u} = \{&k_{\tau_1}(p_u, s_i), k_{\tau_2}(p_u, s_i), \ldots, k_{\tau_5}(p_u, s_i),\\ &k_{\tau_1}(p_u, s_l), k_{\tau_2}(p_u, s_l), \ldots, k_{\tau_5}(p_u, s_l),\\ &k_{\tau_1}(p_u, a_1), k_{\tau_2}(p_u, a_1), \ldots, k_{\tau_5}(p_u, a_1),\\ &k_{\tau_1}(p_u, a_2), k_{\tau_2}(p_u, a_2), \ldots, k_{\tau_5}(p_u, a_2),\\ &k_{\tau_1}(p_u, a_3), k_{\tau_2}(p_u, a_3), \ldots, k_{\tau_5}(p_u, a_3)\}\end{aligned} \tag{9}$$

where $\phi_{\text{brute3}} = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$. Each component of this vector is the heat kernel $k_\tau(p_u, r)$ that measures the amount of heat diffused from p_u to $r \in \mathbf{R}$ at five different times $\tau_{1..5}$, where $\mathbf{R}$ consists of the path endpoints as well as the current triplet retrieved from ϕ_{brute3}. We select small time values as they capture local properties of the shape around p_u. Larger time values, on the other hand, reflect the global structure of the shape from the point of view of p_u, which may confuse the system in the presence of topological noise. Similarly, connecting p_u to r through single shortest paths would not be the proper choice under topological noise. Heat kernel, instead, gives a weighted average over all paths possible in time τ, i.e., it is inversely related to the connectivity of points p_u and r by paths of length τ.

Once $\mathbf{h}_{q_u}$ is defined similarly for each $q_u \in \mathbf{Q}_{jm}$, and $\mathbf{h}_{p_u}$ and $\mathbf{h}_{q_u}$ are normalized to have unit lengths, we evaluate the similarity of $\mathbf{p}_{il}$ and $\mathbf{q}_{jm}$ by adding $||\mathbf{h}_{p_u} - \mathbf{h}_{q_u}||_{L_\infty}$ for $1 \le u \le U$ to the overall dissimilarity amount. Let $\mathbf{q}_{gh}$ be the least dissimilar augmented path for $\mathbf{p}_{il}$ with the dissimilarity amount of δ. We then add two fuzzy votes to the corresponding endpoints (s_i, t_g) and (s_l, t_h) of these two matching augmented paths via $1 - \delta$ (Algorithm 1). While there are many alternatives for the selection of fuzzy votes, e.g., 1 minus pointwise descriptor difference or uniform votes, we choose the overall dissimilarity amount that has been carefully maintained thus far using heat vector differences.

We observe the powers of the heat vector and the reodesic vector $\mathbf{r}_{p_u} = \{d(s_i, p_u), d(s_l, p_u), d(a_1, p_u), d(a_2, p_u), d(a_3, p_u)\}$ in Figure 5. Here, the normalized values in $[0, 1]$ that correspond to the most different vector entries (L_∞) provide the colors of each of the $U = 10$ path vertices. Such vertices are also colored as blocks of 10 consecutive cells in the grid-based visualization of Figure 6. With this visualization scheme, we are able to demonstrate the situation of many matching and non-matching path pairs compactly.

4.6.1 *Vote to Map Conversion.*

We convert the votes into a sparse map by matching each source sample to the target sample with the highest vote. To ensure bijectivity, we maintain only one match for each target sample by deleting the ones with lower votes. The sparse map obtained after this conversion is evaluated via $\mathcal{D}_{\text{iso}}$ (Equation (7)). The whole procedure is repeated for each of the $\binom{M}{3}$ triplets $\{\phi_{\text{brute3}}\}$ and the minimum-distortion (Equation (7)) map ϕ^*_{sparse} based on ϕ^*_{brute3} (Figure 7(middle)) proceeds to the next stage.

4.6.2 *Trim Recover Refine.*

Trimming the sparse map ϕ^*_{sparse} removes the matches whose votes are below the average vote. As noted by Lipman and Funkhouser [2009], some low-vote matches may still be good. We, therefore, then launch a recovery procedure that takes a low-vote match (s_i, t_j) and compares vectors of geodesic distances $g(.,.)$ from s_i and target vertices $\{t\}$ to ϕ_{robust}, similar to Equation (10). These comparisons give us the optimal target vertex t^* that is best for s_i. We then recover (s_i, t_j) if $g(t_j, t^*) < \zeta$ where ζ is 0.1 of the maximum distance over the entire surface.

We finally refine the current ϕ^*_{sparse} by moving the matched target samples within their neighborhoods, as shown in the inset where the bottom row is the refined map. This refinement algorithm can also be seen as a novel matching-aware sampling strategy and it can quickly improve maps computed by any isometric correspondence method.

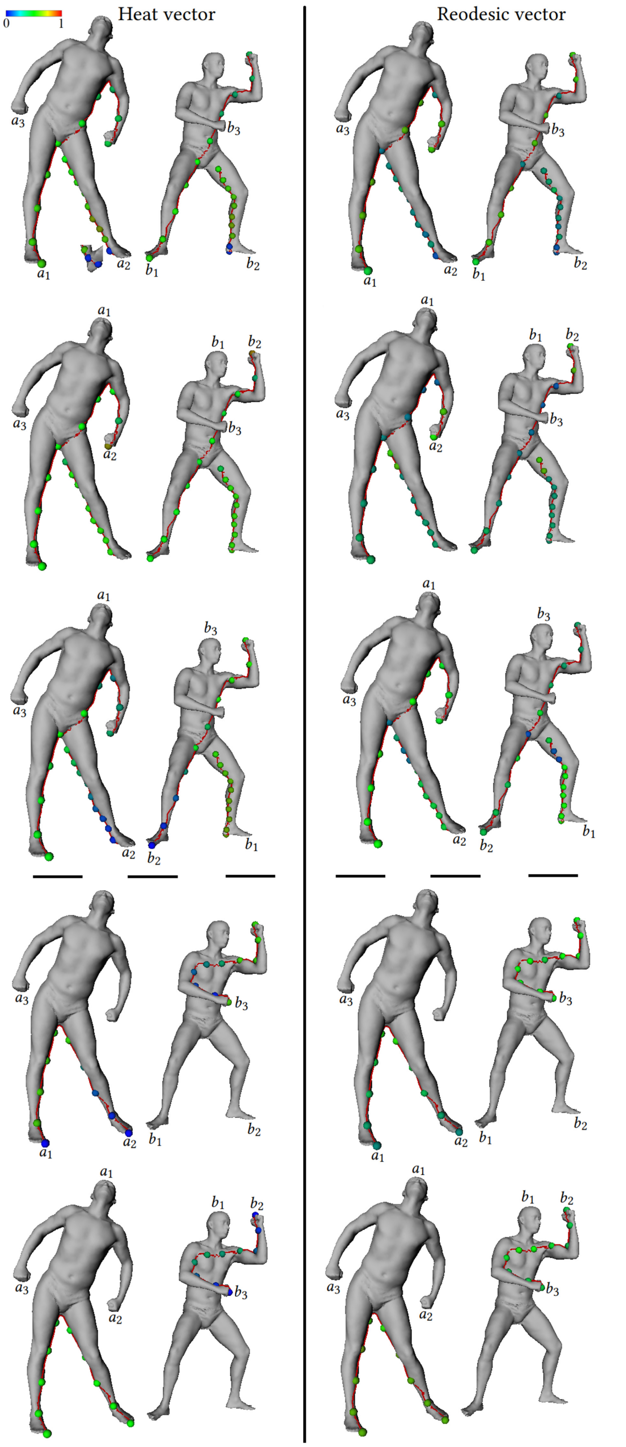


Fig. 5. Visualization of the heat vector (left) and reodesic vector (right) w.r.t. different triplets of matches $\{(a_k, b_k)\}$. Two different paths are used above the dashed line (foot-hand and heel-groin). Paths of similar lengths can be distinguished thanks to the augmentation (below the dashed line).

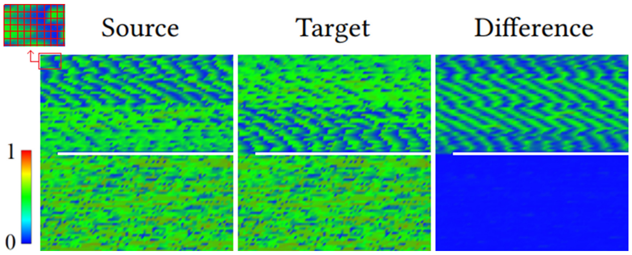
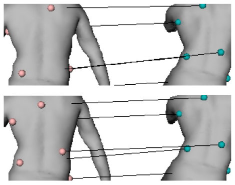


Fig. 6. Visualization of the heat vectors computed using a good triplet of matches. Columns are for source shape, target shape, and difference. Format of each grid: non-matching path pairs, blank padding, matching path pairs.



For each match $(s_i, t_j) \in \phi^*_{\text{sparse}}$, we find the optimal vertex w^* in the patch $\{w\}$ centered at t_j by checking the compatibility of distances from s_i and w to each vertex in a given robust map $\phi_{\text{robust}} = \{(a_k, b_k)\}$, i.e.,

$$w^* = \underset{w}{\operatorname{argmin}} ||g(s_i, \{a_k\}) - g(w, \{b_k\})||_{L_2} \qquad (10)$$

We initially use ϕ^*_{brute3} as ϕ_{robust} and then append (s_i, w^*) to ϕ_{robust} if displacement from t_j to w^* is minimal, e.g., less than 3 average edge lengths. This minimal movement implies that this match is already confident and therefore deserves to be in ϕ_{robust}. Note that w^* may be t_j itself, in which case confident (s_i, t_j) will definitely be in ϕ_{robust}. We repeat this refinement process multiple times with robust maps improved based on the results of previous iterations. During repetitions, in addition to ϕ^*_{brute3}, we use 10 farthest matches of the current sparse map while initializing ϕ_{robust}. The improved robust map is important to increase the accuracy of the refinement as well as the dense map to be computed.

A typical improvement after refinement is shown in Figure 7(right) for two different executions. Note that the bad head-back match in ϕ^*_{brute} at the top row is discarded naturally as we search through $\binom{M}{3}$ triplets. Using more extremities, e.g, $M = 6$, would bring the missing farthest samples, e.g., head and back, which would lead to a better ϕ^*_{brute}, still very quickly (since evaluation time of Equation (7) is negligible). This strategy, however, would result in many more triplets to process for sparse matching whose computation time is not negligible (Section 4.10).

Note that our sparse maps (Figure 7(middle) and (right)) are guaranteed to be one-to-one functions, i.e., every matched target sample is the image of exactly one source sample. Some target samples, however, may remain unmatched after converting votes into maps, e.g., no source sample selects target t_j as its highest-vote partner (Section 4.6.1).

4.7 Dense Map

The same heat diffusion idea that has guided the computation of ϕ^*_{sparse} lays the foundation of our dense map computation. In the absence of paths to be augmented, we must, however, rely on more anchor points that attract heat. To this effect, we use the final ϕ_{robust} made available in the end of Section 4.6.2. We also supplement it further by appending (i) 10 farthest matches

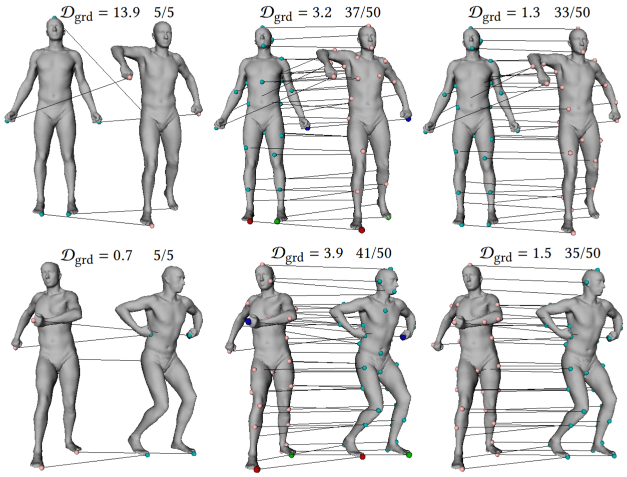

$\mathcal{D}_{\text{grd}} = 13.9$ 5/5
$\mathcal{D}_{\text{grd}} = 3.2$ 37/50
$\mathcal{D}_{\text{grd}} = 1.3$ 33/50
$\mathcal{D}_{\text{grd}} = 0.7$ 5/5
$\mathcal{D}_{\text{grd}} = 3.9$ 41/50
$\mathcal{D}_{\text{grd}} = 1.5$ 35/50

Fig. 7. Brute-force map (left) leads to the sparse map (middle) that is trimmed, recovered, and refined to yield the final ϕ^*_{sparse} (right). Optimal triplets shown in middle with bigger spheres. Map qualities (Equation (14)) and resolutions are also given, e.g., 33/50 means 33 out of 50 samples are matched.

of ϕ^*_{sparse}, (ii) matches of ϕ^*_{sparse} whose individual d_{iso} is less than the overall $\mathcal{D}_{\text{iso}}$ (Equations (7) and (8)), and (iii) the winner triplet ϕ^*_{brute3}. Some matches in this supplemented map ϕ'_{robust} are naturally repeated which would effectively give more weights to those matches in the decision making process. ϕ'_{robust} typically has 25 and 70 distinct matches when $N = 50$ and 100, respectively (Figure 8). Note that we highlight ϕ'_{robust} with bigger spheres and matching lines in the upcoming dense map figures.

For a given source vertex v, we first define its heat vector via Equation (9) by using ϕ'_{robust} instead of ϕ_{brute3}. We then decide to match it with the vertex on the target shape whose heat vector is closest under L_∞ norm. Repeating this process for each source vertex reveals the desired many-to-one dense map ϕ^*.

4.8 Filtering

In order to make our shape correspondence method work well and fast, we filter out paths $\mathbf{p}_{il}$ and $\mathbf{q}_{jm}$ immediately if reodesic distances from their corresponding endpoints to the current triplet $\phi_{\text{brute3}} = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$ are inconsistent. To this end, we set

$$\rho = \min_{k \in \{1,2,3\}} \min \left(\frac{d(s_i, a_k)}{d(t_j, b_k)}, \frac{d(t_j, b_k)}{d(s_i, a_k)} \right) \tag{11}$$

where reodesic distances $d(.,.)$ are computed using ϕ_{brute3}. Path pair $(\mathbf{p}_{il}, \mathbf{q}_{jm})$ is filtered out, i.e., skipped, if $\rho < 0.95$.

Similarly, we skip further processing if the pointwise descriptors f at the endpoints are inconsistent (more than 0.1 difference), where f is the heat kernel signature $k_\tau(s_i, s_i)$ (Section 4.4) averaged over small time parameter values [Sun et al. 2009] and divided by the maximum value to be normalized in [0, 1]. The same skips take place during the dense matching using f and using the winner triplet ϕ^*_{brute3} instead of the current triplet ϕ_{brute3} above.

We find an additional filtering strategy during dense matching very effective, especially in the presence of topological noise (or partial matching) where noisy (or extra) regions need to remain unmatched. It also improves general accuracy. In this strategy, we filter out the match (v, z) if $g(v, v') > \zeta$ even though v and z have the closest heat vectors (Section 4.7). Here, v' is the best match for z with the closest heat vector. Intuitively, we go from source shape (v) to target (z) and then come back to the source again, hoping to arrive at a consistent location (identity map in the best case). To perform this filtering test efficiently, we compute the geodesic between two arbitrary vertices v and v' using the pre-computed geodesics $g(.,.)$ for the sample vertices. To this end, we, in the beginning, associate each vertex v with its closest three

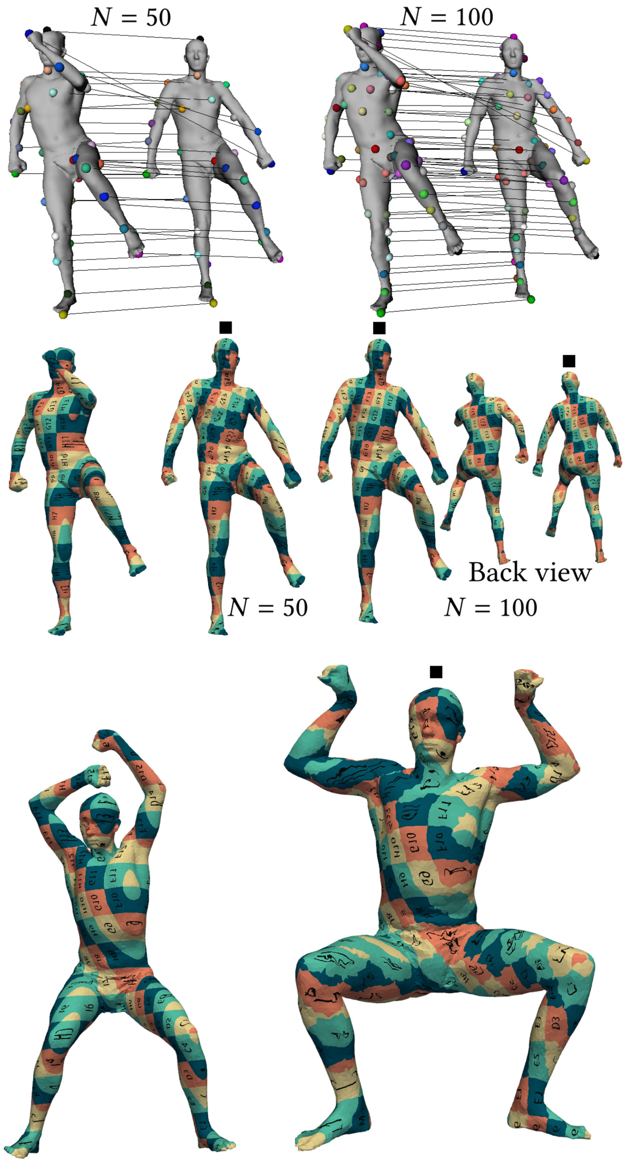


Fig. 8. (Row 1) Robust sparse maps at two different resolutions are shown via spheres and matching lines. (Row 2) Texture in the source surface is transferred to the target surface (indicated by a square as we do in all the upcoming figures) via our smooth dense map ϕ^*. Two different robust maps are used, the ones based on $N = 50$ and 100. (Row 3) Another texture transfer where the target surface is scaled up for visual convenience.

representative samples $r_{\{1,2,3\}}$ using the surface-based barycentric coordinates:

$$v.\alpha = e^{-g(r_1,v)/\sigma} \quad v.\beta = e^{-g(r_2,v)/\sigma} \quad v.\gamma = e^{-g(r_3,v)/\sigma} \qquad (12)$$

where σ is the sampling radius computed by averaging the distances between the closest samples and barycentric coordinates α, β, γ are divided by their sum to map into the $[0, 1]$ interval. We are now able to find the desired distance in constant time:

$$g(v,v') = v.\alpha \times g(r1,v') + v.\beta \times g(r2,v') + v.\gamma \times g(r3,v') \quad (13)$$

We demonstrate this filtering in Figure 9, where v is black and v' is green. Note that, in addition to leaving the noisy (or extra) regions unmatched, this skipping also has a potential to increase the overall accuracy as it allows v to match with a more consistent match even if it fails to possess the closest heat vector. Setting $g(v,v') = ||v - v'||_{L_2}$ instead of barycentric-based geodesic distance led to inferior results.

4.9 Symmetric Flip Handling

We also alleviate the symmetric flip problem by checking the isometric distortion $\mathcal{D}_{\text{iso}}$ (Equation (7)) of the sparse maps created by the first triplet $\phi_{\text{brute3}} \subset \phi^*_{\text{brute}}$ and its flipped versions, namely bilateral flip (e.g., both hands and feet flipped) and two different one-way flips (e.g., only hands flipped). If the flipped triplet from the lowest-resolution ϕ^*_{brute} leads to a lower distortion at the current higher-resolution sparse map, then we proceed with the flipped version of the superset ϕ^*_{brute}. Since sparse map of size N is significantly denser than the brute-force map of size M, $\mathcal{D}_{\text{iso}}$ is more likely to catch symmetry clues, if they exist [Sahillioğlu and Yemez 2013a]. The flipped version is obtained by clustering the symmetric points via the symmetry-invariant **average geodesic distance (AGD)** descriptor [Hilaga et al. 2001] and swapping the matches within the clusters. Namely, for each of the M extremities, we find the extremity with the closest AGD value and create the cluster if it is close enough, e.g., finding the left hand for the right hand.

A similar multi-initialization takes place to improve the accuracy of the sparse map computation. Namely, we repeat Section 4.6 c times with c different brute-force maps to be used as ϕ^*_{brute}. We use the first c brute-force maps after sorting them w.r.t. their $\mathcal{D}_{\text{iso}}$ values. Out of c candidates, the sparse map with minimum $\mathcal{D}_{\text{iso}}$ value is selected as the final ϕ^*_{sparse}. This heuristic is effective especially for the topologically noisy inputs where the min-distortion brute-force map $\phi_{c0\text{brute}}$ may fail to possess at least three good matches we require. Even if $\phi_{c0\text{brute}}$ has three good matches, another valid brute-force map, e.g., $\phi_{c4\text{brute}}$, may provide an even better triplet that eventually leads to a better sparse map. We use $c = 12$.

4.10 Timing

We break down the execution time of each step of our algorithm as measured for an average SCAPE pair with 12.5K vertices on a 16GB 3.20GHz PC: 0.05, 0.17, 0.4, 0, 4.5, 6.8 seconds for Laplacian Construction, Eigendecomposition ($E = 15$), Sampling ($N = 50$), ϕ^*_{brute}, ϕ^*_{sparse} ($M = 5$), ϕ^*, respectively. In total, it takes 11.92 seconds to fully match a SCAPE pair. This time measure is from a sequential execution of the algorithm, but the structure of the solution is highly parallel. Note that for shapes with many extremities, we prefer $M = 7$ (Figure 10(first three rows)), which in turn leads to $\binom{7}{3}$ triplets instead of $\binom{5}{3}$, tripling the computation time of ϕ^*_{sparse}. Please refer to Table 2 for our timing on all of the datasets in our test suite.

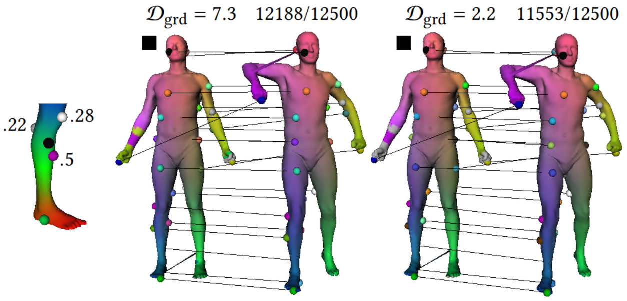

$\mathcal{D}_{\text{grd}} = 7.3$ 12188/12500
$\mathcal{D}_{\text{grd}} = 2.2$ 11553/12500
.22
.28
.5

Fig. 9. Barycentric coordinates for the black vertex are written near the corresponding representative samples (left). Consistency-based filtering is turned off (middle) and on (right), the latter leading to a lower distortion. Matching vertices have the same color and unmatched ones are in gray. Guiding robust sparse map is highlighted via spheres and matching lines.

5 EXPERIMENTS

5.1 Datasets

We tested our method on six different datasets. The first one is a reconstructed pose sequence of a human actor from the SCAPE benchmark [Anguelov et al. 2005], which contains 71 non-uniformly triangulated models. We add shortcut tunnels to the SCAPE models, obtaining our own SCAPET dataset. SHREC'19 [Dyke et al. 2019] comes with non-uniformly sampled human and hand models. This third dataset comes with two classes, SHRiso for the clean models and SHRtopo for the topologically noisy models. The fourth TOPKIDS dataset [Lahner et al. 2016] enables matching the uniform reference kid model to its articulated and topologically noisy versions. All these four datasets have meshes with about 10K vertices. The fifth one TOSCA [Bronstein et al. 2008] represents, with about 20K uniformly sampled vertices, the motion of an articulated object, which we refer to as Cat, Centaur, Dog, Gorilla, Horse, and Wolf. Other objects are 52K-vertex meshes of humans called David, Michael, and Victoria. All of the datasets have ground-truth matching information for an automatic quantitative evaluation. The sixth and last dataset is the 170K-vertex raw scan meshes of FAUST [Bogo et al. 2014] which involves topological noise and holes. We used, in order, 142, 142, 41 (SHRiso) + 17 (SHRtopo), 25, 81, and 40 pairs in total from these six datasets.

5.2 Evaluation Metrics

In addition to the qualitative evaluation through the visuals in Figures 7–17, we provide a quantitative evaluation that quantifies the quality of a given map ϕ via

$$\mathcal{D}_{\text{grd}}(\phi) = \frac{1}{|\phi|} \sum_{(s_i,t_j)\in\phi} g(\varrho(s_i), t_j) \ / \ \rho \qquad (14)$$

where $\varrho(s_i)$ is the ground-truth correspondence of s_i on the target mesh and ρ is the average edge length on the target mesh (for the interpretation of the map distortion in terms of edge counts). In order to simplify comparison to the most of the methods in literature, we use the error-fraction metric in Figure 16 with the units of the plots identical to the ones used in Kim et al. [2011].

5.3 Results

Two sets of isometric pair matchings are shown in Figures 8 and 10 where the former also displays the robust maps that guide the dense matching. Both figures demonstrate cases free of topological noise.

For the more challenging case of matching under topological noise, we still establish stable robust maps (Figure 11) that enable us to produce promising smooth dense maps with possibly unmatched regions around the noisy areas (Figures 12–15, and 17).

We perform further qualitative analysis with the topological noise case using real scan data, specifically using meshes with topological noise and holes caused by bad acquisition (Figure 13(c) and (d)). For quantitative evaluation of this scan set, we use the online server that reports in cm the average ground-truth error per pair using the Euclidean distance between the estimated projections and the ground-truth projections. By obtaining a score of 12.849cm over 40 test pairs, we outperform only 3 of the 35 public methods. We improve to 4.385cm after excluding our 13 symmetrically flipped results, e.g., first two pairs in Figure 13(d), making us outperform six more methods. Note that we had to match our unmatched vertices (Section 4.8) in order to comply with the server. Note also that most of the methods in this benchmark are learning-based and trained on the same dataset, hence lack in generality. The other ones rely on a pose prior where the shapes are brought to a potentially manual initial alignment for good convergence. Our method is general, training-free, fully-automatic, and did not use even one landmark correspondence initialization in order to produce its results.

Quantitative analysis over all other datasets with ground-truth information are given in the sequel in comparison with the state-of-the-art methods (Table 1 and Figure 16), where we also conduct an ablation study concerning different components of our algorithm (Section 5.3.2).

5.3.1 Comparisons. We compare our performance with seven different methods which have been the state of the art when they were first released, namely the learning-based **Deep Functional Maps** [Litany et al. 2017] **(DeepFM)**, and other rule-based [Ezuz et al. 2019] (Elastic), [Pai et al. 2021] (FastSink), [Nogneng and Ovsjanikov 2017] (Commute), [Ren et al. 2018] (BCICP), [Melzi et al. 2019a] (ZoomOut), and [Eisenberger et al. 2020] (Shells). As shown in Figures 14–16 and Table 1, we mostly outperform these methods for the topological noise case and are only slightly worse than FastSink, ZoomOut, and Shells for the noise-free isometric mappings. FastSink, ZoomOut, and Shells perform well on isometric input such as SCAPE and TOSCA, as they are refinement algorithms over the well-studied isometric functional maps framework. Specifically, we initialize FastSink, Elastic, and ZoomOut with Ovsjanikov et al. [2012], landmark-based [Aigerman and Lipman 2016], and BCICP, respectively. Similarly, Shells fuses registration with the same functional map framework. They are also almost on a par with our method on the topological noise input such as SCAPET and TOPKIDS since they represent the shapes as point clouds in high dimensional space and then compute a pointwise registration, which in these dataset naturally removed topologically wrong connections in most of the cases. On the other topology datasets SHRtopo (and TOPKIDS for FastSink and ZoomOut),

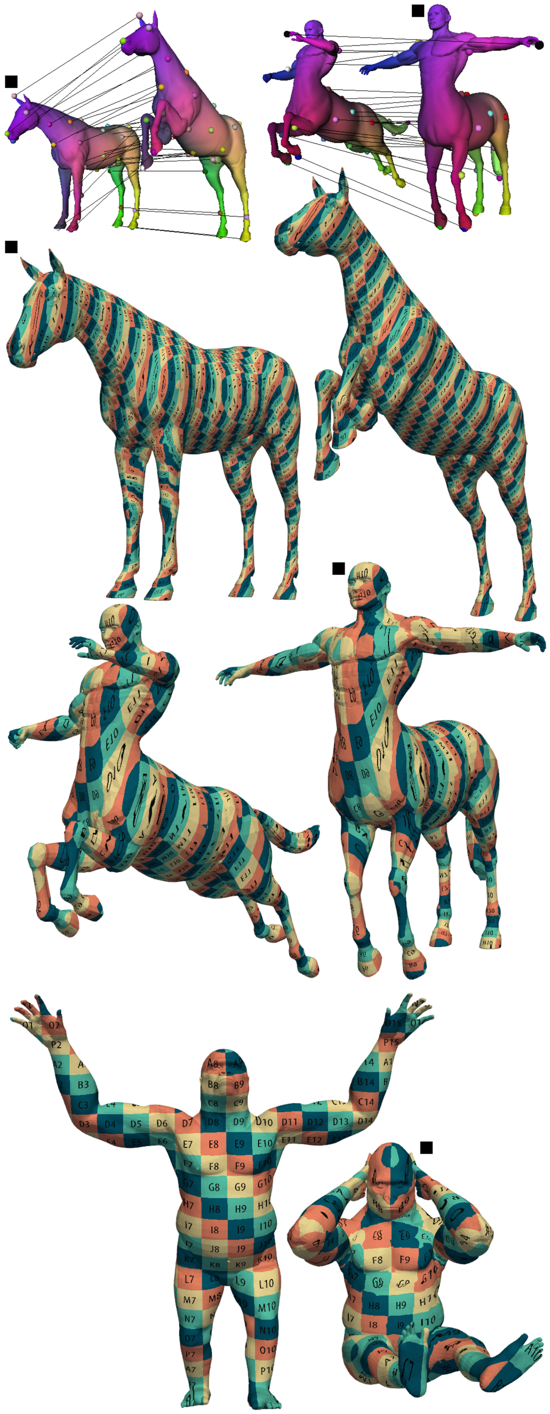


Fig. 10. (Row 1) Our robust sparse maps (spheres and lines) shown along with color transfer through our dense maps. (Rows 2–3) Same dense maps depicted via texture transfer. (Row 4) Another non-human texture transfer.

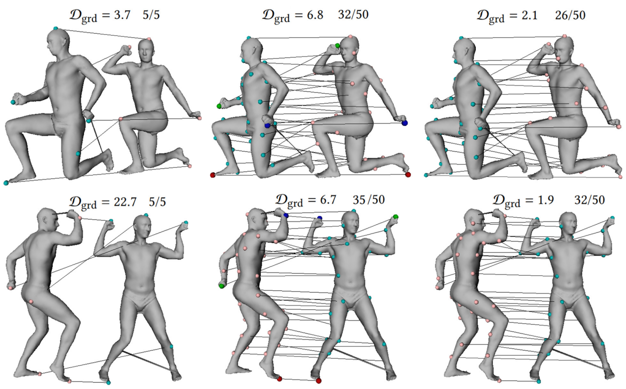

$\mathcal{D}_{grd}$ = 3.7 5/5
$\mathcal{D}_{grd}$ = 6.8 32/50
$\mathcal{D}_{grd}$ = 2.1 26/50
$\mathcal{D}_{grd}$ = 22.7 5/5
$\mathcal{D}_{grd}$ = 6.7 35/50
$\mathcal{D}_{grd}$ = 1.9 32/50

Fig. 11. SCAPET brute-force and sparse maps in the same format as Figure 7.

however, these methods are outperformed by our method. We also note that unlike these methods, our method does not require any initialization, i.e., computes the dense maps from scratch, and any costly extrinsic deformation. We also avoid the cheese pull effect problem of Shells that may occur depending on the source shape selection. Our method, on the other hand, is insensitive to the selection of source and target shapes. We finally show the comparative matching results of the same shape pair with and without topological noise in Figure 15 in order to verify the robustness of our method.

We also compare our execution times with the competitors based on the 12.5K vertex SCAPE pair. While we require merely 11.92 seconds (Section 4.10), Elastic, Commute, BCICP, and Shells run in the order of minutes. FastSink and ZoomOut achieve our fast execution time, whereas DeepFM handles queries almost instantly. Note that DeepFM requires a significant training work and also is not as flexible as the other methods, including ours, since it needs to be trained separately for each shape class (we used their public pre-trained model for humans). To show our scalability, we report 5.1 secs (sparse) + 17.8 secs (dense) = 22.9 secs as the execution time for 28K-vertex Cat pair. We observe no significant overhead in this higher resolution pair as most of the computations are based on pre-computed geodesics on N samples. Although we have $O(V^2)$ dense matching time, most of the pairs are not processed due to filtering (Section 4.8), leading to an almost linear increase for the dense map computation.

5.3.2 Ablation Study. We finally compare our method to itself by disabling and enabling the refinement (Section 4.6.2) and symmetric flip alleviation (Section 4.9) modules to emphasize their improvements (Table 2). Note that, these are independent improvement steps that can be seamlessly incorporated into any isometric correspondence method. The same table also provides other comparisons as well as our results w.r.t. different choices of M and N values. We can summarize the findings of this ablation study as follows:

- Turning off the filtering degrades the results for the topologically noisy datasets (SCAPET, SHRtopo, TOPKIDS) as the votes collected from bad path pairs around wrong geodesic paths are large enough to confuse the system.

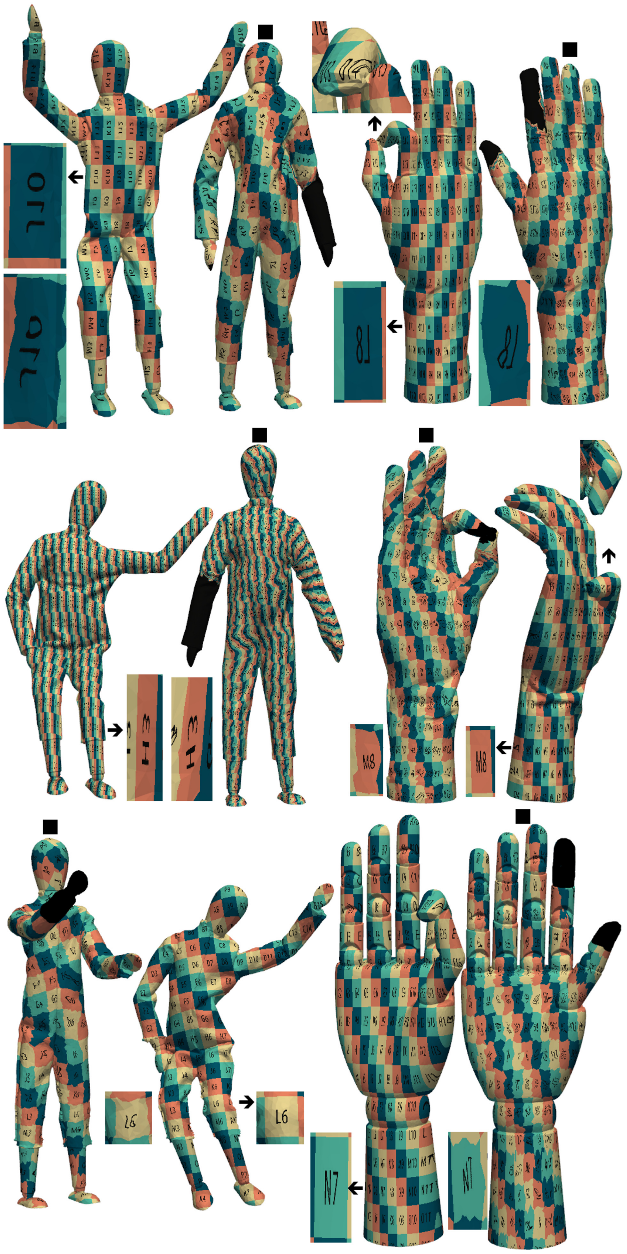


Fig. 12. SHRtopo topological noise cases. Unmatched vertices are in black.

- Regardless of the dataset, turning off the filtering significantly slows down the system as paths are augmented (sparse part) and heat vectors are compared (sparse and dense parts) unnecessarily, e.g., SCAPE execution time quadruples.
- Changing $M = 5$ to $M = 7$ triples the sparse matching timing due to the processing of $\binom{7}{3}$ triplets instead of $\binom{5}{3}$. Overall timing does not triple though, as the dense matching part is not affected by the choice of M, e.g., 4.5 + 6.8 vs. 13.7 + 6.2 for SCAPE (plus additional minor operations such as sampling). Accuracy is likely to increase slightly as new triplet options are available with $M = 7$. These options turn out to

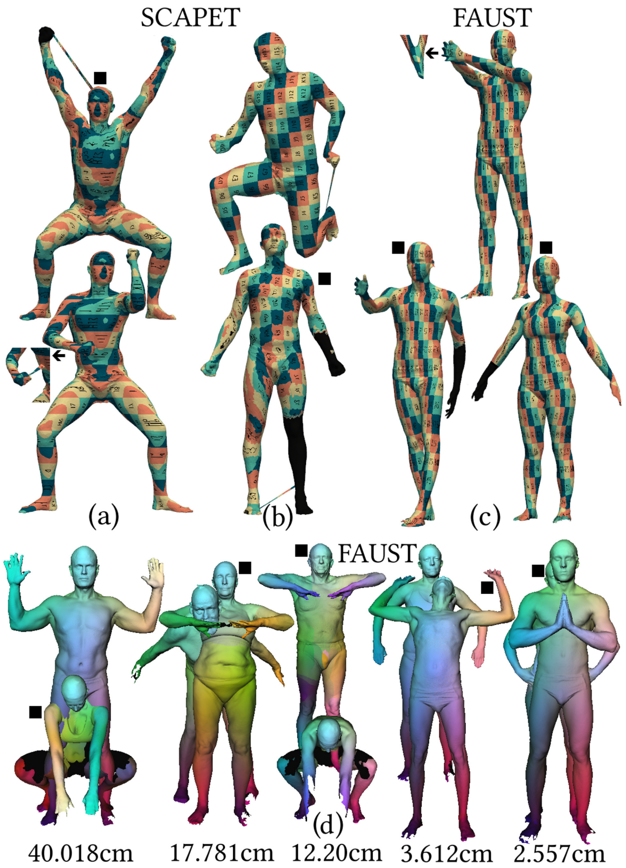


Fig. 13. Our dense matching of the topological noise cases from SCAPET (a) and (b) and FAUST scans (c). Note that in addition to the noise, our algorithm handles a certain amount of non-isometry as the source shape (glued model at top) is matched to not only itself but also to a different person. Further FAUST scan evaluation with their quantitative error metric is provided (d).

Table 1. Quantitative Evaluation of the Maps using $\mathcal{D}_{\text{grd}}$ in Equation (14)

	DFM	Elast	FSink	BCICP	ZOut	Shells	Ours
SCAPE	14.22	12.80	0.10	1.41	0.10	**0.09**	1.13
SCAPET	21.49	24.61	3.79	6.00	3.82	3.19	**2.03**
SHRiso	11.75	10.28	6.17	7.00	6.65	4.33	**4.02**
SHRtopo	29.04	27.03	15.93	16.97	13.32	7.92	**4.99**
TOPKIDS	39.25	39.28	16.09	16.36	6.94	**3.10**	3.50
TOSCA5	7.61	11.79	1.56	1.86	1.57	**1.47**	1.77
TOSCA7	n/a	12.07	**1.91**	2.74	1.96	1.96	1.92

The best-performing method for each dataset is written in bold. DFM and ZOut are shorts for DeepFM and ZoomOut, respectively. TOSCA5 = {David, Michael, Victoria, Gorilla} and TOSCA7 is for the rest ($M = 7$). Values for Commute from top to bottom: 1.46, 6.63, 7.15, 19.03, 16.62, 5.11, 6.86.

be more critical when handling shapes with many extremities (TOSCA7).

- Changing $N = 50$ to $N = 100$ makes sparse matching six and nine times slower for $M = 5$ and $M = 7$ cases, respectively, due to the increase in the number of paths to be processed. This change does not significantly affect the dense matching timing.
- Results slightly improve with larger N as there are more options for the sparse matching part. Improvement is, however, only marginal because our refinement already moves samples around to obtain a similar effect with smaller N. Please also see the visual comparison of different N values in the first row of Figure 8.
- Results are not affected on the clean datasets when redoesics and heat vectors are replaced by geodesic counterparts. They, however, get worse on the noisy datasets, verifying the importance of our contributions under topological noise.

One last ablation study that is not present in Table 2 is the replacement of our sparse map module with well-known alternatives from literature [Sahillioğlu 2020]. While such a replacement works for clean isometric datasets (mostly at the expense of execution time), it is highly likely to fail on topologically noisy datasets (SCAPET, SHRtopo, TOPKIDS). Lipman and Funkhouser [2009], for instance, requires genus zero surfaces (sphere topology), a property that is always violated with the introduction of shortcut handles. Geodesic-based solutions [Solomon et al. 2016] will fail as topological noise greatly affects the geodesic paths. Deformation-based solutions are also likely to fail as the regularization term would prevent proper unfolding around noisy regions [Zhang et al. 2008]. Our procedure is free of deformation and genus restriction. It is also robust to the changes in geodesic paths thanks to the reodesics and heat vectors.

5.3.3 *Unmatched Regions.*

We finish our evaluation by showing how our competitors behave on the regions that are left unmatched by our method (Table 3). Note that such regions arise only in the topological cases due to filtering with the following statistics: The number of shape pairs that have unmatched regions are 119, 15, and 22 for SCAPET, SHRtopo, and TOPKIDS, respectively. The average ratios of areas of the unmatched regions w.r.t. the shape areas are 6.3%, 5.1%, and 5.5%.

We naturally do not consider those vertices while evaluating Equation (14) or plotting. Similarly, for our competitors we do not consider the vertices labeled as unmatched by our method. While this strategy achieves fairness on comparisons, it hides the performance of our competitors on regions we are unable to match. It turns out that these restricted maps are consistently worse than the global maps used in the previous evaluations (Table 3). FastSink and ZoomOut are again relatively stable in these datasets, while the latter being especially successful for the TOPKIDS, providing acceptable maps on the regions that we could not match.

6 LIMITATIONS

Failure cases arise under significant deviation from isometry, e.g., giraffe vs. cat where neck stretches significantly. Note that we can easily replace the geodesic distance $g(.,.)$ in reodesic computation (Equation (2)) with spectral diffusion and biharmonic distances. Even so, we have not seen any improvement due to the nature of these distances.

Despite our symmetric flip alleviation scheme that improves the flipping situation by about %70 (Table 2), the analysis with $N = 50$ samples remains insufficient to resolve all intrinsic symmetries.

Our augmented path is a special case of the region of interest based analysis in Sahillioğlu and Yemez [2013b] and van Kaick et al.

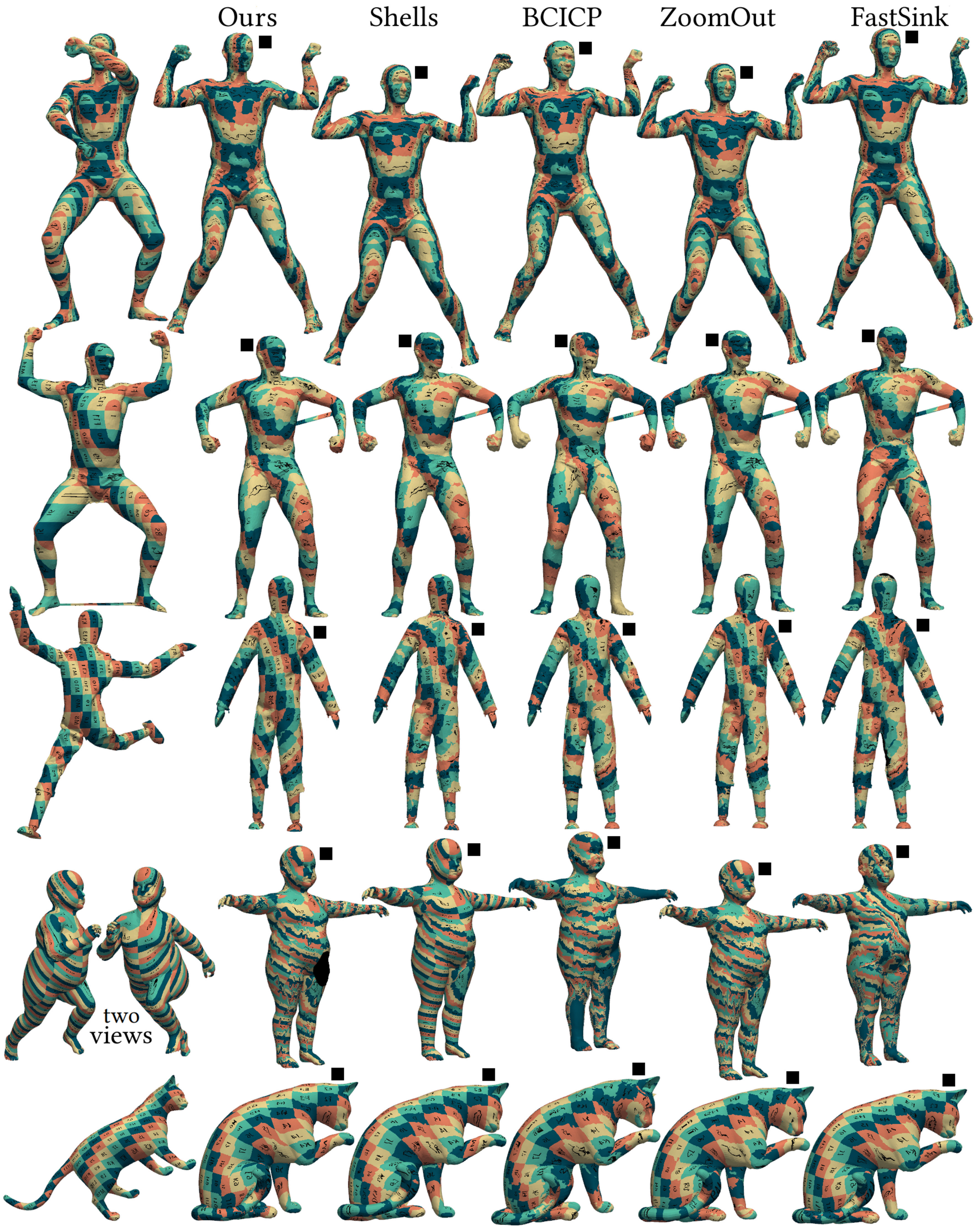


Fig. 14. Comparisons of the maps produced by our method (source and target pair at left) and the competitors (only the target model is shown).

[2013] which report successful partial matching results. We therefore run our method without any modification on the partially isometric dataset of Rodolà et al. [2017] and observe success up to a certain level of partiality. In particular, our brute-force map involves at least three good matches (required by a healthy sparse map computation) only if the partial model runs through the whole surface, with missing parts whose symmetric counterparts are present, e.g., cut by *yz*-plane but not *xy*-plane (Figure 17(a) and (b)).

While our algorithm is quite stable to the topological noise in the form of shortcut tunnels (Table 2(SCAPET)), it may be unsatisfactory when one part is glued to the other over a continuous

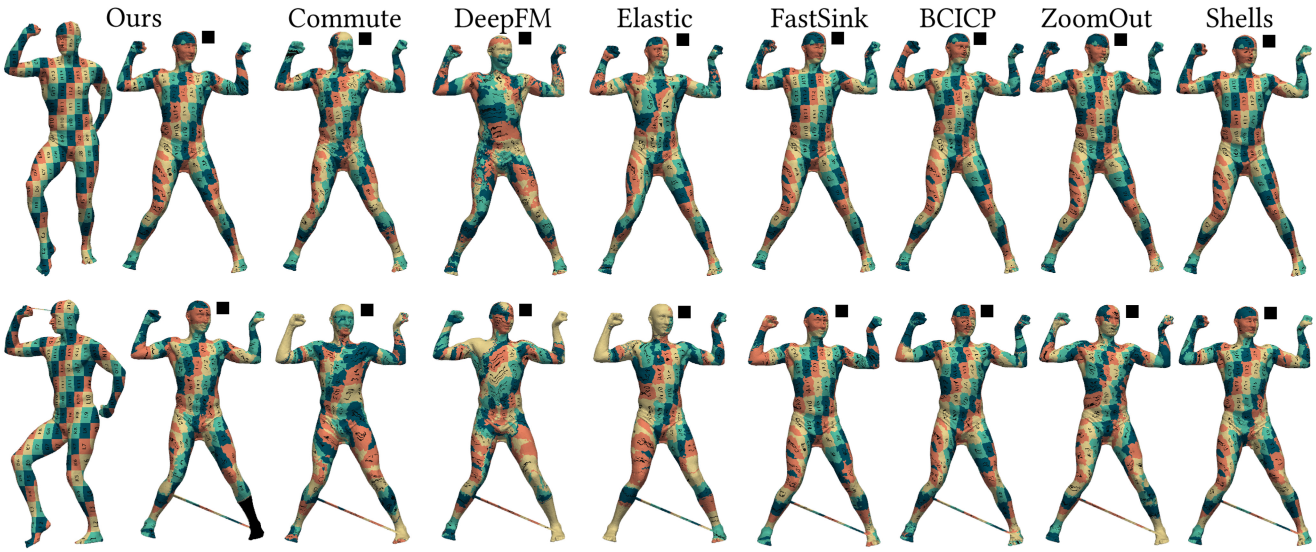


Fig. 15. Our method and all of the competitors are compared for the same pair with (bottom) and without (top) topological noise (hand-head and foot-knee).

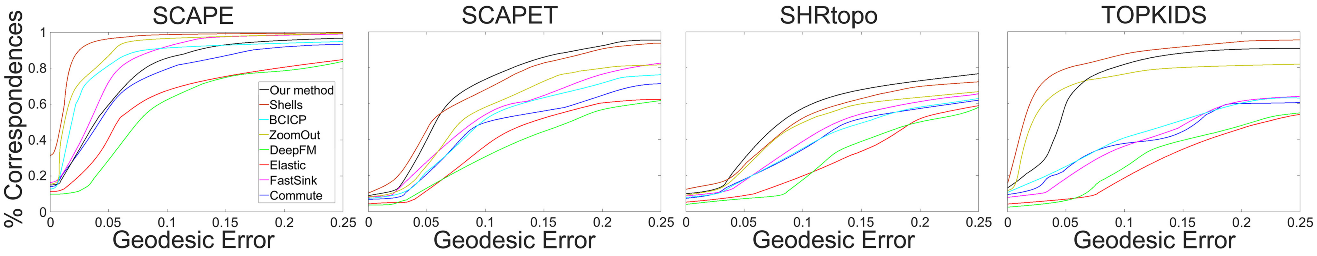


Fig. 16. Error-fraction plots of our method and competitors on topologically clean (SCAPE) and noisy (others) isometric datasets.

Table 2. Ablation Study Concerning Refinement (Section 4.6.2), Symmetric Flip Handling, Reodesic-based Filtering, Consistency-based Filtering (Section 4.8), M and N Parameters (Section 3), Reodesics and Heat Vectors (Section 4.1)

	$(\mathcal{D}_{\text{iso}}, \mathcal{D}_{\text{grd}})$		% flips		$\mathcal{D}_{\text{grd}}$	$(\mathcal{D}_{\text{grd}}$, secs) $\boldsymbol{N} = \mathbf{50}$		$(\mathcal{D}_{\text{grd}}$, secs) $N = 100$		$\mathcal{D}_{\text{grd}}$
	Ref. off	Ref. on	Allev. off	Allev. on	Filts. off	$\boldsymbol{M} = \mathbf{5}$	$M = 7$	$M = 5$	$M = 7$	Pure geo.
SCAPE	(3.4e-05, 3.101)	(1e-05, 1.131)	61.7	15.9	1.160	**(1.13, 11.9)**	(1.07, 21.2)	(1.10, 34.8)	(1.01, 47.9)	1.15
SCAPET	(7.5e-03, 4.413)	(6e-03, 2.028)	67.1	30.9	11.145	**(2.03, 12.2)**	(1.99, 22.7)	(2.00, 35.4)	(1.97, 49.2)	13.6
SHRiso	(7.7e-04, 7.001)	(1.3e-04, 4.019)	60.6	21.8	4.12	**(4.02, 10.9)**	(4.01, 19.1)	(3.95, 33.2)	(3.94, 44.6)	4.03
SHRtopo	(8.1e-04, 8.113)	(1.4e-04, 4.993)	56.7	24.1	16.113	**(4.99, 11.1)**	(4.94, 19.8)	(4.59, 33.6)	(4.50, 46.1)	18.9
TOPKIDS	(9.5e-03, 7.173)	(8e-03, 3.498)	76.7	39.3	19.501	**(3.50, 11.5)**	(3.34, 22.3)	(3.21, 35.1)	(3.16, 48.7)	29.4
TOSCA5	(4.1e-05, 3.908)	(1.9e-05, 1.772)	68.5	21.1	1.829	**(1.77, 49.3)**	(1.71, 80.3)	(1.65, 168.7)	(1.64, 192.7)	1.71
TOSCA7	(1.2e-04, 3.979)	(2.1e-05, 1.921)	64.3	20.1	7.451	**(2.58, 21.7)**	(1.92, 33.1)	(2.30, 57.9)	(1.87, 63.8)	1.89

Reodesics and heat vectors are replaced with regular geodesics and geodesic vectors for the last one (please see Figure 17(d) as well). Bold configuration ($M = 5$ and $N = 50$) is the promoted one that provides a good compromise between accuracy and time.

long path (Table 2(TOPKIDS) and (SHRtopo) and Figure 17(c)). The main reason of this limitation is the lack of consistent through-vertices for the reodesic computation. We, however, still observe smooth dense maps when the glue is sufficiently small (Figure 17(e), Figure 14(row 4), and Figure 13(c)). We perform better on SCAPET (tunnels) compared to TOPKIDS and SHRtopo (gluing). For the latter, we still find reodesics and heat vectors valuable as it leads to much worse results when they are replaced with pure regular geodesics (Figure 17(d) and Table 2).

7 CONCLUSION

We presented augmented paths and reodesics as two main contributions in order to address the isometric shape correspondence problem under topological noise, a relatively challenging scenario that has attracted less attention in the literature. Our algorithm is fully automatic, fast, and accurate. It can be applied to shapes with arbitrary genus and resolution, requires no initial input matches, and can even support partial matching under certain conditions. It is also insensitive to the source selection. Evaluation on different

Table 3. $\mathcal{D}_{\mathrm{grd}}$ Values of the Maps Corresponding to the Regions Left Unmatched by Our Algorithm

	DeepFM	Elast.	FSink	BCICP	ZoomOut	Shells
SCAPET	50.657	48.129	7.904	12.547	8.133	6.931
SHRtopo	63.776	55.821	21.497	25.470	19.301	16.701
TOPKIDS	91.331	76.057	25.789	25.418	13.049	5.551

Commute from top to bottom: 19.02, 33.41, 27.11.

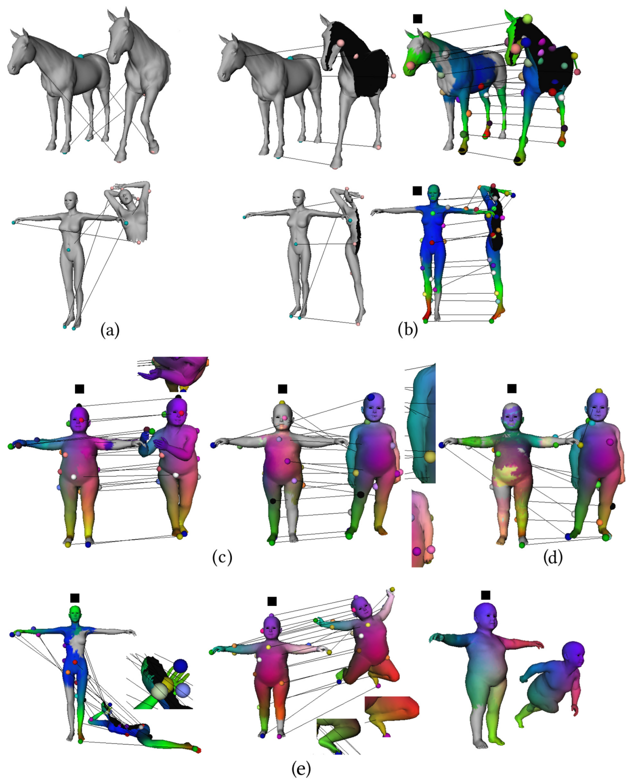


Fig. 17. Different cutting schemes that make our algorithm fail (a) and work (b) for the partial matching scenario ((e) works too). Different gluing schemes that make our algorithm fail (c) and work (e). Our map in (c) still outperforms (d) that is purely geodesic based.

types of topological noises reveals that our algorithm is more suitable for the shortcut tunnel noise which can potentially hurt the geodesic paths.

Topological noise scenario can be addressed in a dynamic setting in order to maintain or track the correspondence within the animation sequence, which would be very helpful for the transfer of captured motion in volumetric videos. We plan to address this issue as well as the limitations mentioned in Section 6 as future works.

ACKNOWLEDGMENTS

We thank Vladislav Samsonov from Arcturus Studios for his help with the experiments, and Joseph Silverman from Arcturus Studios for preparing the teaser image and the SCAPET dataset. We also thank the anonymous reviewers for their comments.

REFERENCES

Noam Aigerman, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Spherical orbifold tutte embeddings. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

Noam Aigerman and Yaron Lipman. 2015. Orbifold tutte embeddings. *ACM Trans. Graph.* 34, 6 (2015), 190–1.

N. Aigerman and Y. Lipman. 2016. Hyperbolic orbifold tutte embeddings. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 35, 6 (2016), 217.

D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. 2005. Scape: Shape completion and animation of people. *ACM Trans. Graph.* 24, 3 (2005), 408–416.

Melike Aydınlılar and Yusuf Sahillioğlu. 2021. Part-based data-driven 3D shape interpolation. *Computer-Aided Design* 136 (2021), 103027.

Omri Azencot and Rongjie Lai. 2021. A data-driven approach to functional map construction and bases pursuit. *Computer Graphics Forum (Proc. SGP)* 40 (2021).

Omri Azencot, Orestis Vantzos, and Mirela Ben-Chen. 2016. Advection-based function matching on surfaces. In *Computer Graphics Forum*, Vol. 35. 55–64.

Alex Baden, Keenan Crane, and Misha Kazhdan. 2018. Möbius registration. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 211–220.

F. Bogo, J. Romero, M. Loper, and M. Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. (2014), 3794–3801.

A. M. Bronstein, M. M. Bronstein, and R. Kimmel. 2008. *Numerical Geometry of Non-Rigid Shapes.* Springer.

A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. 2010. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *Intl. Journal of Computer Vision* 89, 2–3 (2010), 266–286.

Christopher Choy, Wei Dong, and Vladlen Koltun. 2020. Deep global registration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* 2514–2523.

Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. 2015. Continuous matching via vector field flow. In *Computer Graphics Forum*, Vol. 34. 129–139.

Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. 2020. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8592–8601.

R. M. Dyke, C. Stride, Y.-K. Lai, P. L. Rosin, M. Aubry, A. Boyarski, A. M. Bronstein, M. M. Bronstein, D. Cremers, M. Fisher, T. Groueix, D. Guo, V. G. Kim, R. Kimmel, Z. Lähner, K. Li, O. Litany, T. Remez, E. Rodolà, B. C. Russell, Y. Sahillioğlu, R. Slossberg, G. K. L. Tam, M. Vestner, Z. Wu, and J. Yang. 2019. Shape correspondence with isometric and non-isometric deformations. In *Eurographics Workshop on 3D Object Retrieval.*

Michal Edelstein, Danielle Ezuz, and Mirela Ben-Chen. 2019. Enigma: Evolutionary non-isometric geometry matching. *arXiv preprint arXiv:1905.10763.*

Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. 2020. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 12265–12274.

Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. 1997. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Processing* 6 (1997), 1305–1315.

D. Ezuz and M. Ben-Chen. 2017. Deblurring and denoising of maps between shapes. *Computer Graphics Forum* 36, 5 (2017), 165–174.

Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. 2019. Elastic correspondence between triangle meshes. 38, 2 (2019), 121–134.

M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. *Proc. SIGGRAPH* (2001).

Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. 2020. Consistent ZoomOut: Efficient spectral map synchronization. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 265–278.

V. Kim, Y. Lipman, and T. Funkhouser. 2011. Blended intrinsic maps. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4 (2011).

Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Stephen DiVerdi, and Thomas Funkhouser. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11.

A. Kovnatsky, M. Bronstein, X. Bresson, and P. Vandergheynst. 2015. Functional correspondence by matrix completion. *Proc. Computer Vision and Pattern Recognition* (2015), 905–914.

Zorah Lahner, Emanuele Rodolà, Michael Bronstein, Daniel Cremers, Oliver Burghard, Luca Cosmo, Andreas Dieckmann, Reinhard Klein, and Yusuf Sahillioğlu. 2016. SHREC'16: Matching of deformable shapes with topological noise. In *Proc. Eurographics Workshop on 3D Object Retrieval.*

Y. Lipman and T. Funkhouser. 2009. Möbius voting for surface correspondence. *ACM Trans. Graph.* 28, 3 (2009).

Yaron Lipman, Raif M. Rustamov, and Thomas A. Funkhouser. 2010. Biharmonic distance. *ACM Transactions on Graphics (TOG)* 29, 3 (2010), 1–11.

Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. 2017. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE International Conference on Computer Vision.* 5659–5667.

Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. 2017. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman. 2016. Point registration via efficient convex relaxation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35, 4 (2016), 73.

Simone Melzi, Jing Ren, Emanuele Rodola, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. 2019a. ZoomOut: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–14.

Simone Melzi, Riccardo Spezialetti, Federico Tombari, Michael M. Bronstein, Luigi Di Stefano, and Emanuele Rodola. 2019b. Gframes: Gradient-based local reference frame for 3D shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4629–4638.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III.* Springer, 35–57.

Dorian Nogneng, Simone Melzi, Emanuele Rodola, Umberto Castellani, Michael Bronstein, and Maks Ovsjanikov. 2018. Improved functional mappings via product preservation. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 179–190.

Dorian Nogneng and Maks Ovsjanikov. 2017. Informative descriptor preservation via commutativity for shape matching. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 259–267.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11.

Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. 2010. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1555–1564.

Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. 2021. Fast Sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 384–393.

Adrien Poulenard, Primoz Skraba, and Maks Ovsjanikov. 2018. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 13–25.

Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. 2021. Discrete optimization for shape matching. In *Computer Graphics Forum*, Vol. 40. 81–96.

Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2018. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)* 37, 6 (2018), 1–16.

E. Rodolà, S. Bulo, T. Windheuser, M. Vestner, and D. Cremers. 2014. Dense non-rigid shape correspondence using random forests. (2014), 4177–4184.

Emanuele Rodolà, Luca Cosmo, Michael M. Bronstein, Andrea Torsello, and Daniel Cremers. 2017. Partial functional correspondence. In *Computer Graphics Forum*, Vol. 36. 222–236.

Emanuele Rodola, Michael Möller, and Daniel Cremers. 2017. Regularized pointwise map recovery from functional correspondence. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 700–711.

Yusuf Sahillioğlu. 2018. A genetic isometric shape correspondence algorithm with adaptive sampling. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 1–14.

Yusuf Sahillioğlu. 2020. Recent advances in shape correspondence. *The Visual Computer* 36, 8 (2020), 1705–1721.

Yusuf Sahillioğlu and Ladislav Kavan. 2021. Scale-Adaptive ICP. *Graphical Models* (2021), 101113.

Yusuf Sahillioğlu and Yücel Yemez. 2011. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum (Proc. SGP)* 30, 5 (2011), 1461–1470.

Yusuf Sahillioğlu and Yücel Yemez. 2012. Minimum-distortion isometric shape correspondence using EM algorithm. *IEEE Trans. PAMI* 34, 11 (2012), 2203–2215.

Yusuf Sahillioğlu and Yücel Yemez. 2013a. Coarse-to-fine isometric shape correspondence by tracking symmetric flips. *Computer Graphics Forum* 32, 1 (2013), 177–189.

Yusuf Sahillioğlu and Yücel Yemez. 2013b. Partial 3D correspondence from shape extremities. *Computer Graphics Forum* 33, 6 (2013), 63–76.

Patrick Schmidt, Janis Born, Marcel Campen, and Leif Kobbelt. 2019. Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–15.

A. Sharma, R. Horaud, J. Cech, and E. Boyer. 2011. Topologically-robust 3D shape matching based on diffusion geometry and seed growing. *Proc. Computer Vision and Pattern Recognition (CVPR'11).* 2481–2488.

Abhishek Sharma and Maks Ovsjanikov. 2020. Weakly supervised deep functional maps for shape matching. *Advances in Neural Information Processing Systems* 33 (2020).

Justin Solomon, Andy Nguyen, Adrian Butscher, Mirela Ben-Chen, and Leonidas Guibas. 2012. Soft maps between surfaces. In *Computer Graphics Forum*, Vol. 31. 1617–1626.

J. Solomon, G. Peyre, V. Kim, and S. Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35, 4 (2016), 72.

J. Sun, M. Ovsjanikov, and L. Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum* 28, 5 (2009).

O. van Kaick, H. Zhang, and G. Hamarneh. 2013. Bilateral maps for partial matching. *Computer Graphics Forum* 32, 6 (2013), 189–200.

O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. 2011. A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707.

Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel, et al. 2017a. Efficient deformable shape correspondence via kernel matching. In *2017 International Conference on 3D Vision (3DV).* IEEE, 517–526.

M. Vestner, R. Litman, E. Rodolà, A. Bronstein, and D. Cremers. 2017b. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. *Proc. Computer Vision and Pattern Recognition* (2017).

Larry Wang, Anne Gehre, Michael M. Bronstein, and Justin Solomon. 2018. Kernel functional maps. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 27–36.

Yue Wang and Justin M. Solomon. 2019. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 3523–3532.

Ofir Weber and Denis Zorin. 2014. Locally injective parametrization with arbitrary fixed boundaries. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–12.

L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li. 2016. Dense human body correspondences using convolutional networks. (2016).

Hao Zhang, Alla Sheffer, Daniel Cohen-Or, Quan Zhou, Oliver Van Kaick, and Andrea Tagliasacchi. 2008. Deformation-driven shape correspondence. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1431–1439.

Received 3 September 2021; revised 20 June 2022; accepted 1 August 2022