Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Partial Correspondence of 3D Shapes using Properties of the Nearest-Neighbor Field

Nadav Yehonatan Arbel[a,*], Ayellet Tal[1], Lihi Zelnik-Manor[1]

[a]*Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion, Haifa, 3200003, Israel*

ARTICLE INFO

Article history:
Received May 21, 2019

Keywords: Shape correspondence, Partial matching

ABSTRACT

This paper proposes an algorithm for finding correspondences between shapes in 3D. The method is designed to address three challenging cases: large deformations, partiality of the shapes, and topological noise. At the core of the method lies a novel, yet simple, similarity measure that analyzes statistical properties of the nearest-neighbor field from the source surface to the target. This information is shown to be powerful, compared to minimizing some function of distances. In particular, the proposed similarity function analyzes the diversity of the nearest-neighbor field and its preservation of distances. Empirical evaluation on partial matching benchmarks shows that our method outperforms state-of-the-art techniques, both quantitatively and qualitatively.

1. Introduction

Shape correspondence is a fundamental problem in computer vision and computer graphics, both in 2D and in 3D. Numerous applications require robust correspondences, for instance in animation, reconstruction, and shape analysis [1]. The focus of this paper is on shape correspondence between meshes in 3D.

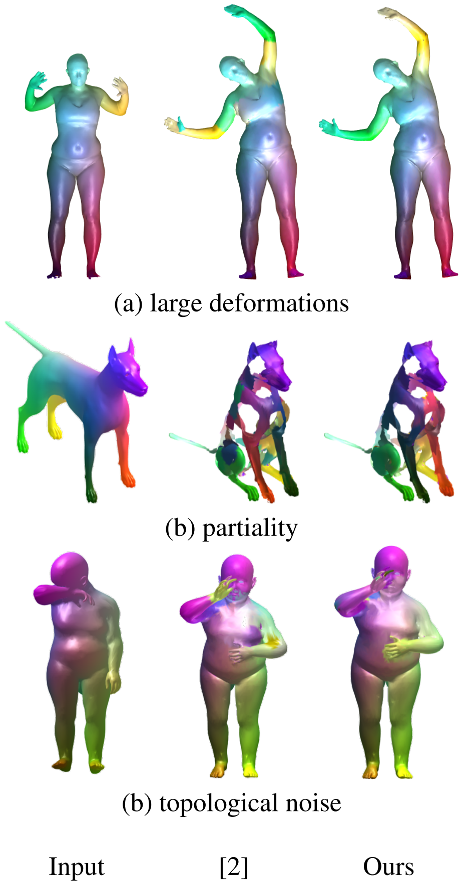
Finding correspondences between shapes is highly challenging, even when the objects are rigid and full [3, 4, 5, 6]. This paper, however, addresses the problem of shape correspondence, when the following additional challenges are added (see Figure 1): (1) The objects may have gone through non-rigid deformations [7, 8, 9, 10, 11]; (2) only part of the shape is given and should be matched to the correct region within the full shape (*partiality*) [12, 13, 2, 14], and (3) non-adjacent parts of the surfaces intersect [15, 16, 17, 18] (*topological noise*). All of the above frequently occur in real world scenarios.

Previous approaches have focused on minimization of some distortion criteria, of either point-wise shape descriptors [16, 2], pair-wise shape descriptors [14], or the combination of both [18]. Impressive result have been exhibited , yet some downfalls still exist. This is in particular evident in the three cases mentioned above, in particular when the deformation is extreme, when partiality is severe, and in many cases of topological noise.

Some recent approaches have utilized deep neural networks [19, 20, 21, 22]. These show a lot of promise on a couple of full-shape benchmarks. In [19] the results are analyzed also for a partial correspondence benchmark; we will show that our method outperforms their results. All these methods require a significant amount of labeled training data, which is currently difficult to acquire.

The algorithm proposed in this paper belongs to the first class of algorithms, which does not utilize deep learning. It proposes a novel similarity function, which analyzes the nearest neighbor field in vertex shape descriptor space. That is to say, for each vertex of a source mesh we find the nearest neighbor in the target mesh, in terms of a specific shape descriptor and a distance function. Rather than minimizing some function of distances, we analyze statistics of this field. The statistical nature of our method lets it ignore outliers, which are the source of unreliability in some other methods. Specifically, the statistics we are concerned about regard two aspects of the nearest neighbor field: (1) the diversity of the field, i.e. how many different matches the nearest neighbor field contains, and (2) preservation of pairwise distances of the matches in the nearest neighbor

[*] Corresponding author: Tel.: +972-54-247-2883;
e-mail: `nyarbel85@gmail.com` (Nadav Yehonatan Arbel), `nyarbel@campus.technion.ac.il` (Nadav Yehonatan Arbel)


(a) large deformations

(b) partiality

(b) topological noise

Fig. 1. Challenging correspondences. Corresponding vertices are colored similarly. (a) While the corresponding arms are switched in [2], our algorithm manages to match the arms correctly. (b) When given a highly partial model of a dog as input, our algorithm manages to match its four dog's legs correctly. (c) The right hand is well matched by our algorithm.

field.

We have tested our method on the two challenging benchmarks of *SHREC'16*, the one that contains partial deformed shapes [23] and and the one that contains deformed shapes with topological noise [24]. We exhibit the benefit of our method both qualitatively (Figure 1) and quantitatively. In particular, our method obtains a 10-20% improvement over the state-of-the-art on the first dataset and is competitive on the second. In addition, we also show qualitative results on the FAUST dataset [25].

Hence, our contribution is twofold.

1. We introduce a new approach for finding correspondences between given meshes, which is robust to deformations, partiality and topological noise. The novelty of our approach is relying on properties of the nearest-neighbor field.
2. We demonstrate the benefits of our algorithm on the commonly-used benchmarks, both quantitatively and qualitatively.

2. Related work

Correspondence of deformable shapes in 3D. The problem of finding shape correspondences between deformable objects in 3D has been studied extensively. Most methods attempt to minimize some distortion criteria, which falls into one of three categories: (1) local shape similarity, commonly computed as the distance between corresponding point descriptors [26, 27, 28, 29, 30, 31, 32, 33], (2) pairwise relations [15, 34, 11], or (3) a combination of both [18].

The underlying assumption of most of these methods is that the shapes are either approximately isometric or that they are topologically homeomorphic. This assumption usually do not hold in the case of partial correspondence and topological noise.

A variety of approaches have been proposed to handle topological differences. In [35], resilience to topological shortcuts in the context of intrinsic symmetry detection of deformable shapes is studied. Wang et al. [36] considered the metrics induced by commute-time kernels as a more robust alternative to geodesic distance. In [37, 38] sparse relaxations to this framework were introduced. A different kernel was proposed by [39] and bilateral maps were suggested by [17]. Chen and Koltun [15] reformulated the isometric embedding problem with a robust norm accounting for topological artifacts. Boscaini et al. [40] proposed a CNN-based shape descriptor to address the problem. Litani et al. [16] modified the functional mapping of [41] to better handle topological noise. Vestner et al. [18] formulated the problem as a quadratic assignment problem that incorporates matching of both point-wise and pair-wise descriptors.

Partial correspondence was first tackled, assuming that the shapes are rigid [42, 43, 13]. In the non-rigid case, the notion of minimum distortion correspondence was utilized [44, 37, 38]. A voting-based method was proposed by [45], to match shape extremities. Other works include the alignment of tangent spaces [46] and the design of robust descriptors for partial matching [17]. In the context of collections of shapes, partial correspondence has been considered in [47, 48]. Masci et al. [21] introduced a deep learning framework for computing a dense correspondence between deformable shapes. [19] improved upon this by introducing anisotropic convolution kernels. In [2, 16] the notion of functional maps [41] was adapted to the partial matching scenario.

The method introduced in this paper addresses both partial matching and topological noise. We present a novel similarity measure that inherently differs from the above methods. Instead of relying explicitly on distances between descriptors, our similarity measure is based on statistics of simple properties of the nearest neighbor field between the points of the two given surfaces.

Correspondence of deformable shapes in images. In images, partial matching is often termed *template matching*. Numerous papers have attempted to solve the problem; a good review is given in [49]. The commonly-used methods are pixel-wise [50, 51]. Geometric transformations have also been addressed [52, 53]. Another group of methods considers a global probabilistic property of the template [54, 55]. Recently, machine learning based techniques have also been used [56].

Our work is inspired by the methods of [57, 58], which also look at various statistics of the nearest-neighbor field of the correspondence. In particular, in [57] it is proposed to simply count points which are mutually nearest neighbors of each other. In [58] it is suggested to rely mostly on a subset of matches—on points that have distinct nearest neighbors. We adopt this general approach, but use other criteria, which are more suitable to surfaces in 3D that are orderless and lack constant density.

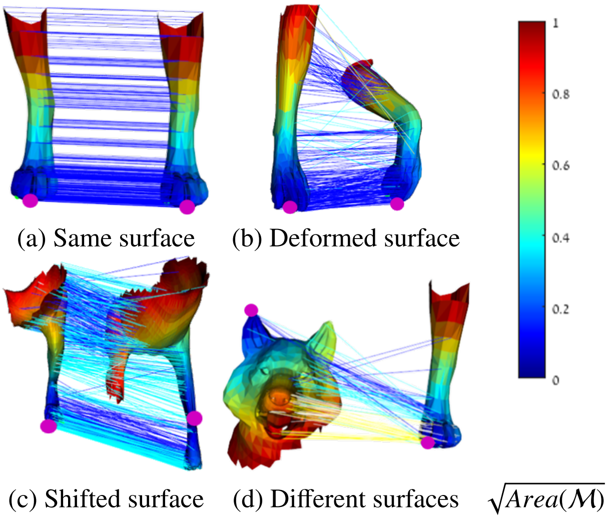
3. General Approach

Given two surfaces $\mathcal{M}$ and $\mathcal{N}$, represented as triangulated meshes, the goal is to find the best match of $\mathcal{N}$ (a partial shape) within $\mathcal{M}$ (a full mesh). In particular, we aim at extracting a sparse set of point correspondences between the shapes. Our approach is based on the following four key ideas, of which the first two capture properties of the nearest-neighbor field in shape descriptor space between the surfaces.

First, inspired by [58], when $\mathcal{N}$ and a part of $\mathcal{M}$ correspond, most points in that part of $\mathcal{M}$ have unique NN-matches in $\mathcal{N}$. This implies that the NN field should be highly diverse, in the sense that many different points in $\mathcal{N}$ are being matched. This is illustrated in Figure 2(a) where the two surfaces are equal and in Figure 2(b) where the two surfaces correspond, since one is a deformation of the other. Therefore, in both of these cases, most points have unique matches and hence, each line connects a different pair of points. Conversely, Figure 2(d) shows a case of inherently-different surfaces, resulting in a small number of target points that happen to be somewhat similar to the input points.

Second, arbitrary matches typically imply a large inconsistency in the location of the matches, whereas correct matches are consistent. We propose to measure the degree of consistency by looking at the geodesic distances of points within a surface. Specifically, when two surfaces correspond, the pairwise geodesic distances between a reference point and other vertices in that surface should match the pairwise distances between the reference point and other vertices on the other surface. In Figure 2 this is encoded by the color of the lines that connect the two shapes. In Figure 2(a,b), which show corresponding surfaces, the pairwise distances from their reference points are similar, and hence they are colored in blue. Conversely, in Figure 2(c,d), the colors of the lines indicate inconsistent distances between the corresponding points, and many of the lines are yellow or cyan, as expected since the surfaces are non-corresponding. Combining the above two ideas leads to a new similarity measure, which is based both on the diversity of the Nearest-Neighbor field and on the consistency of the distances between the points.

Third, rather than realizing the above two ideas on $\mathcal{N}$ as a whole, it is preferable to perform it on a set of smaller sub-surfaces of $\mathcal{N}$. This is due to two reasons: (1) A small sub-surface is more likely to exhibit consistent distances, especially in the presence of partiality, holes or topological noise. For instance, take two corresponding vertices, one on a full surface and the other on a surface with holes; the geodesic distances between each of these vertices and other vertices on their associated surfaces will greatly vary, as a result of ”bypassing” holes. (2) In the case of repeating patterns and a large surface, the diversity will be small, contrary to what we seek-after.



(a) Same surface (b) Deformed surface

(c) Shifted surface (d) Different surfaces

Fig. 2. Properties of the nearest-neighbor field. The surfaces are colored by their geodesic distances from the magenta reference point (center of the surface); the lines are colored by the difference between the geodesic distances from the reference point of the corresponding pairs of points. Clearly, similar surfaces (a,b), even when deformed, exhibit diversity in matching. This can be seen by the fact that the points of the corresponding pairs, connected by lines, are all distinct, rather than having some point(s) into which many lines converge. Furthermore, in this case, most lines are blue, which indicates similar distances from the reference point. Conversely, in (c), though the surfaces are similar, the reference point is different. In this case there are many cyan lines, indicating a worse correspondence due to bad localization. Finally, when the surfaces are highly different (d), there are many yellow and cyan lines, indicating bad correspondence.

Fourth, a multi-scale approach with respect to the size of matched sub-surfaces is beneficial [59]. This is so since larger surfaces contain more global context, resulting in matches which lie within the correct region, but provide poor localization. On the other hand, matching smaller surfaces leads to results that are better locally, but may be globally inconsistent (e.g., mapping a hind leg to a front leg, which is identical locally).

Our algorithm, which is illustrated in Figure 3, realizes these ideas. It consists of five steps, as follows.

1. Descriptors & Nearest-Neighbors. Shape descriptors are calculated for every vertex of both meshes. Many descriptors have been proposed in the literature [60, 32, 31]. We use the *Fast Point Feature Histogram (FPFH)* [30], which captures the relative angular directions of the normals with respect to one another. FPFH is robust to small deformations and partiality of the data, whilst sensitive to symmetrical flips, since it relies on the angles between many local reference frames around each point [61]. Therefore, it addresses a major drawback of matching a right arm, for example, to the left one.

We then compute an approximate nearest neighbor field (*NNF*) mapping. This is done by assigning each vertex of $\mathcal{M}$ its nearest neighbor in $\mathcal{N}$, in terms of their FPFH descriptors.

2. Patch extraction. Inline with the third key idea, we aim at extracting a meaningful set of sub-surfaces, which cover (rather

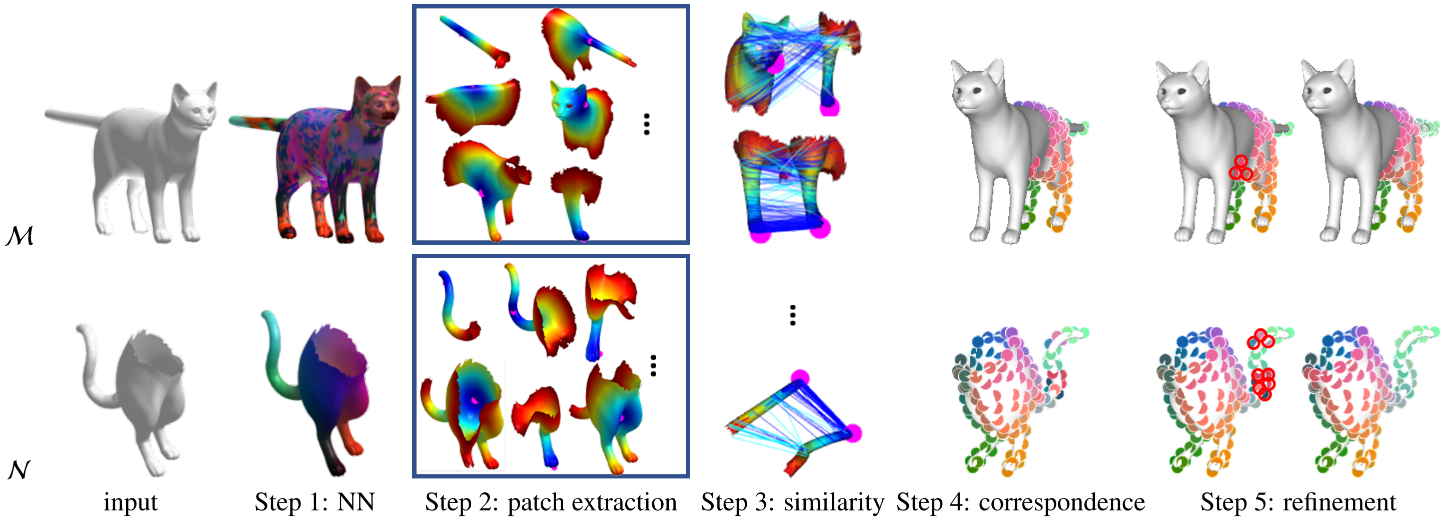


Fig. 3. Algorithm outline. In Step 1, we compute the nearest-neighbor field that maps the shape descriptors of $\mathcal{M}$ to their corresponding nearest neighbors on $\mathcal{N}$. In Step 2, sub-surfaces (patches) are extracted for every sample point in $\mathcal{N}$ and for every vertex in $\mathcal{M}$. Examples of six such patches are shown; note that they may overlap. A reference point on each subsurface is shown in magenta and the colors on the surface encode the geodesic distance from it. Step 3, which is the core of the algorithm, computes the similarity between the patches in $\mathcal{M}$ and $\mathcal{N}$. In the figure, the nearest neighbors are connected by lines. The color of the lines encodes the difference of the geodesic distances between each of these vertices and the reference point in its own subsurface. In Step 4, the color of the points represent their correspondence— points that achieve the maximal similarity score are corresponding (and have the same color on both models). Step 5 refines the correspondences, by modifying the outliers according to the correspondences in their environment. See, for instance, the changes of the corresponding points of those marked by red circles.

than partition) the surface. This is done in two steps: First, we extract a meaningful set of sample points. We then extract the patches using these samples. We elaborate hereafter.

To extract the sample point set S, we start from the extremities of the surface, which are considered salient points [62]. A vertex is defined as an extremity if it resides on a tip of the surface (e.g., tips of limbs), In practice, we define an extremity to be a vertex that is a local maximum of the sum of the geodesic distance functional.

Formally, $\forall v \in \mathcal{N}$, let N_v be the set of neighboring vertices of vertex v. Let $GeoDist(v_i, v_j)$ be the geodesic distance between vertices v_i and v_j on mesh $\mathcal{N}$. Vertex v is an extremity if it satisfies

$$\sum_{v_i \in \mathcal{N}} GeoDist(v, v_i) > \sum_{v_i \in \mathcal{N}} GeoDist(v_n, v_i) \; \forall v_n \in N_v. \qquad (1)$$

Then, we iteratively add more samples,so as to gradually cover more and more of the mesh. At each iteration, the next sample point is chosen as follows. We construct a ”forbidden” region around every point in the set. This region is a geodesic disc of radius $0.05\sqrt{Area(\mathcal{M})}$; the constant is chosen so as the number of points would be on par with other sparse methods [23]. The next point to be added to the set is a vertex whose geodesic distance to any sample point is minimal and it must not fall in any of the forbidden regions. This process stops when the entire surface is marked forbidden.

Once the set of representing samples is defined, a disc (subsurface) of geodesic radius R_T is extracted around each sample point. This is the sought-after set of patches that covers the surface. Specifically, $R_T = \beta \cdot \sqrt{Area(\mathcal{M})}$. As our approach is multi-scale, β, which was found empirically by minimizing the error of correspondences on a training set, varies. In practice we use $\beta = \{0.6, 0.4, 0.2\}$.

3. Computing similarities between pairs of patches. This step is the core of our algorithm, which realizes the first two key ideas.

For each pair of patches of the same scale, $P_i \subset \mathcal{N}$ and $Q_j \subset \mathcal{M}$, constructed around vertices v_i and w_j respectively, as described in Step 2, we compute a similarity value. Recall that our goal is to reward a nearest-neighbor field with high diversity and low inconsistency. We will define the similarity function that achieves it in Section 4. We note that this step, as well, is performed in a multi-scale manner.

4. Extracting a sparse set of corresponding points. Given the similarity values between the patches, as computed in Step 3, our goal now is to extract a set of corresponding points between $\mathcal{N}$ and $\mathcal{M}$. If we had a single scale, then for each sample point (the center of a patch) of $\mathcal{N}$, we would choose the vertex of $\mathcal{M}$ that maximizes the similarity function. However, if the scale is too coarse, the exact matching point is likely not to be found; and if the scale is too fine we may find a corresponding vertex, but on other parts of the model (e.g., on almost-symmetric parts).

Our multi-scale approach addresses these difficulties. We proceed from coarse to fine, first finding the most likely region the corresponding point should lie on and then refining the exact location. Suppose that $P_i \subset \mathcal{N}$ and $Q_j \subset \mathcal{M}$ were found to have the highest similarity in a coarsest scale. The coarsest correspondence is then set between $v_i \in P_i$ and $w_j \in Q_j$, where v_i, w_j are the *geodesic centers* of P_i, Q_j, respectively (i.e. these are the sample points that define the patches in Step 2). When moving to a finer scale, we replace Q_j with a smaller patch in which w_j is the center. Once this patch is constructed, we replace the w_j-centered patch by a new patch of the same (small) scale centered around w_{j_n}. The latter is set to be the patch that maximizes the similarity function of Stage 3 compared to all the patches centred at vertices within w_j-centered patch. Since the patch around w_{j_n} is more similar to the patch centered at v_i than any other patch, w_{j_n} is considered to be the new corresponding point of v_i. In this manner we move from one scale to the next and refine the corresponding vertex of v_i.

Figure 4 illustrates how the correspondence found at the first scale improves throughout the scales. This is compared to the case in which a single scale is used and the corresponding point is either inexact ($\beta = 0.6$) or is erroneous ($\beta = 0.4, 0.2$) .

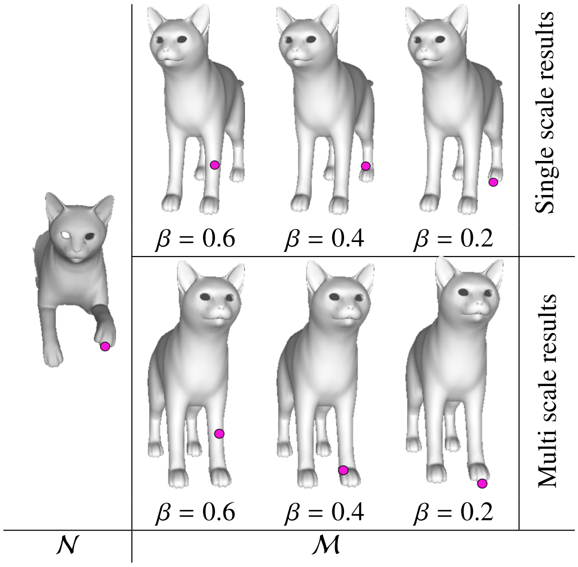


Fig. 4. Multi-scale similarity. The input is the the sample point on the left (in magenta). In the coarsest level ($\beta = 0.6$)), the region of the corresponding point is found, but the point itself is imprecise. In a single-scale approach (top row), finer scales miss the correct region altogether. Conversely, our multi-scale approach (bottom row) utilizes the coarse corresponding region to keep refining the correspondence, and the precise correspondence is found in the finest scale.

5. Coherency-based correspondence refinement. The result of Step 4 is a set of corresponding pairs of points. In most cases (> 92% in all our examples), the correspondences are correct. The goal of this step is to identify the incorrect ones and replace them by the correct correspondences.

The key idea is to utilize coherency, i.e., if all points in the neighborhood of point $v \in \mathcal{N}$ are mapped to points that reside in the same region on $\mathcal{M}$, it is expected that the corresponding point of v, $w \in \mathcal{M}$, will also reside in this region. In other words, we are looking for outliers of the mapping in order to fix them.

To detect these outliers, we check the sum of difference of geodesic distances induced by the mapping between a pair of points in $\mathcal{N}$ and their corresponding points in $\mathcal{M}$. On average, outliers will result in a large difference of geodesic distances and can therefore be detected by an empirically-set threshold. Let us define the sum of differences of geodesic distances between a point $v_j \in \mathcal{N}$ and its corresponding point $w_j \in \mathcal{M}$ as

$$\Delta_{v_j,w_j} = \sum_{v_i \in S} |GeoDist(v_j, v_i) - GeoDist(w_j, w_i)|, \qquad (2)$$

where w_i is the corresponding point of v_i. We consider a correspondence (v_j, w_j) to be correct if it is smaller than the average distance of all other correspondences,

$$\Delta_{v_j,w_j} < C \frac{\sum_{v_i \in S} \Delta_{v_i,w_i}}{|S|}, \qquad (3)$$

where C is empirically set to 1.15.

If Equation 3 is violated, it indicates geodesic distance inconsistency with many other correspondences, thus, v_j is considered an outlier. In this case, we replace w_j, the corresponding point of v_j, by a "better" point w_j*. We require this point to satisfy two conditions: (1) it is a local maximum of the similarity function of Step 3 (to be discussed hereafter in Section 4) and (2) among the vertices that satisfy condition (1) we choose as w_j* the vertex whose Δ_{v_j,w_j*} is minimal. The first condition means that the two vertices are indeed similar. The second condition means that the correspondence is consistent.

4. Similarity between sub-surfaces

This section elaborates on Step 3, which is the core of the algorithm. Given a pair of patches of the same scale, $Q \subset \mathcal{M}$ and $P \subset \mathcal{N}$, this section defines a similarity function between them. We require that this function be oblivious to non-rigid deformations, to different resolutions of the meshes, to noise, to topological noise, and to partiality of the data.

We would like to reward a correspondence for which the *nearest-neighbor (NN)* field satisfies two properties: it has high diversity of the corresponding points, as well as low inconsistency of the distances. In what follows we explain these two properties.

Diversity. When Q and P correspond, each point on Q should have a unique NN-match on P. Conversely, if Q and P do not correspond, most of the points on Q do not have a good match on P. In the latter case, the nearest neighbors are likely to belong to a small set of points that happen to be somewhat similar to the points of Q. This implies that if the patches correspond, their NN-field is highly *diverse*, i.e., pointing to many different points in P.

An intuitive and efficient way to measure diversity is to count the number of unique nearest neighbors between the points of Q and P:

$$Div(Q, P) = |\{p_i \in P : \exists q_j \in Q, NN(q_j, P) = p_i\}|, \qquad (4)$$

where $\{p_i\}_{i=1}^{|P|}$ and $\{q_j\}_{j=1}^{|Q|}$ are the set of points of Q and P, respectively and the nearest neighbors is computed between the descriptors (FPFH) of the points. However, we will see below that the diversity can be calculated implicitly.

Distance inconsistency. A relatively stable property of deformed surfaces is their geodesic distances. Therefore, if two patches correspond, points that belong to a nearest neighbor pair tend to have similar geodesic distances to the centers of the patches they reside on. Conversely, arbitrary matches typically do not hold this relation.

To realize this observation, we define $DistInconst(Q, P)$, the inconsistency of Q and P, as follows. Let $p \in P$ and $q \in Q$

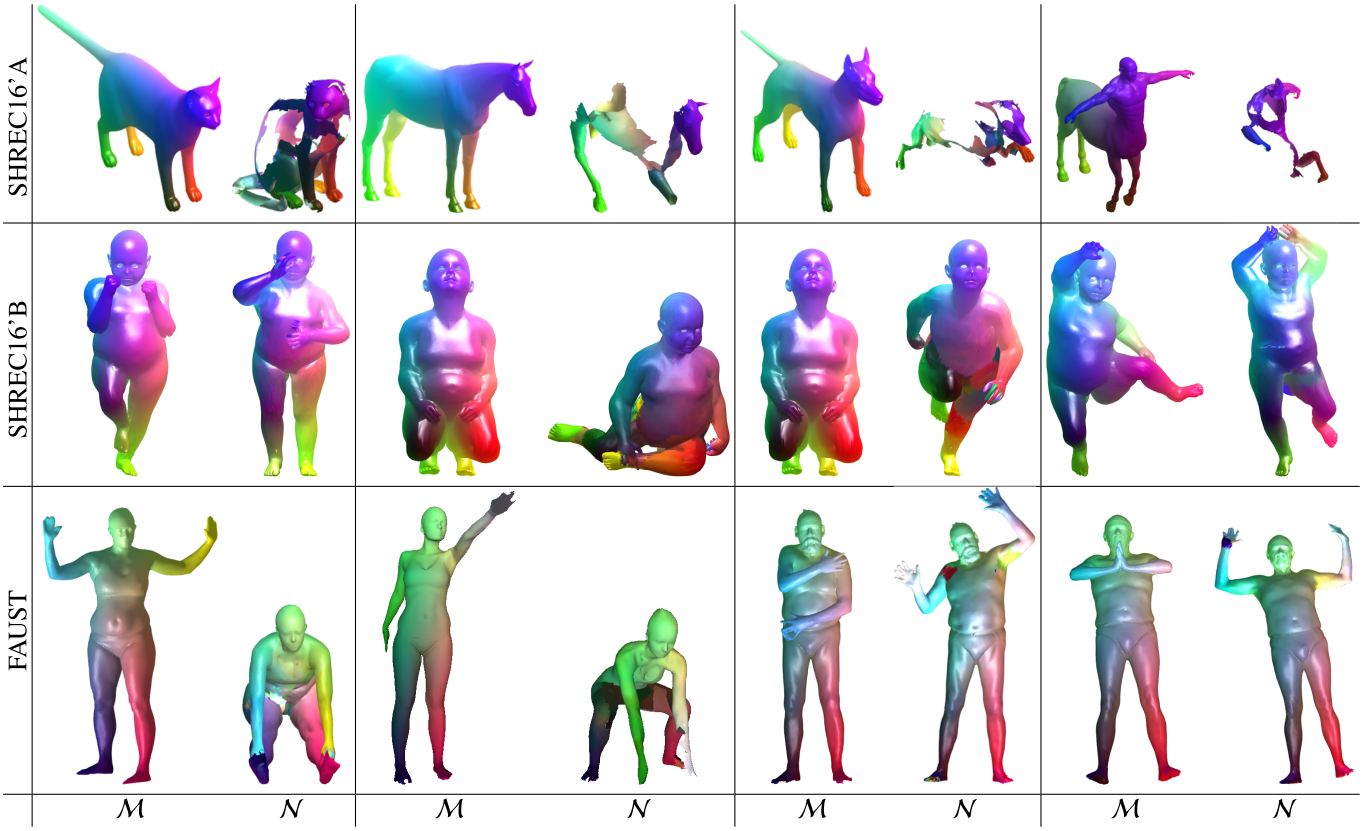


Fig. 5. Results of our algorithm on examples from various datasets. The corresponding points are colored in the same color. Note the accuracy of our algorithm in cases of symmetry (all), partiality (top), topological noise (middle) and large deformations (bottom).

be the centers of P and Q, respectively; furthermore, let $p_i = NN(q_j, P)$ be the nearest-neighbor of $q_j \in Q$. The deformation implied by the NN-Field for p_i, q_j is defined by:

$$DistInconst(q_j, p_i, Q, P) = \\ |GeodDist(q_j, q) - GeodDist(p_i, p)|/\epsilon, \tag{5}$$

where $0 < \epsilon << Area(\mathcal{M})$ is a used for numerical stability. We note that diversity was defined between patches, whereas inconsistency was defined between points; this will be clarified later, when we show how to use these ideas within our general similarity function.

Similarity function. Next, we should integrate the above considerations within a similarity definition, such that similar surfaces will have high diversity and small distance inconsistency. For each $p_i \in P$, we find the minimal inconsistency

$$r_i^* = min_{q_j \in Q} DistInconst(q_j, p_i, Q, P),$$

such that p_i is the nearest neighbor of q_j in the descriptor (FPFH) space. Note that some points in P might not be associated with any point in Q, since they are not nearest neighbors of any point $q_j \in Q$; in this case we set $r_i^* = \infty$, in order to make the contribution of p_i be zero.

Finally, we define the similarity between patches p and Q as:

$$Similarity(P, Q) = \sum_{p_i \in P} \frac{1}{1 + r_i^*}. \tag{6}$$

It is easy to see that this function rewards low inconsistency. However, why does it also reward high diversity of the NN-Field? To understand this, consider the special case where $r_i^* \in \{0, \infty\}$. When this occurs, the value of Equation 6 is either 1 (if $r_i* = 0$) or 0 (if $r_i* = \infty$). In the former case, this indicates that a point p_i has a point in Q that considers p_i to be its nearest neighbor. In this scenario, the similarity function simply counts the number of points in P that are nearest neighbors of some point in Q. But, this is precisely the diversity function we seek-after.

In the general case, the contribution of every point is inversely weighted by its inconsistency r_i^*. This gives preference to NN Fields that preserve pair-wise distances.

5. Results

We have evaluated our method both qualitatively and quantitatively on the two datasets of SHREC'16: (1) the benchmark of *SHREC'16A—partial matching of deformable shapes* [23]; (2) the even more challenging benchmark of *SHREC'16B—matching of deformable shapes with topological noise* [24]. In both cases, our method either outperforms the results of state-of-the-art methods or is competitive. In addition, we provide qualitative evaluation on challenging objects from *FAUST* [25]; see Figure 5.

SHREC'16A contains 400 partial shapes, each is a near-isometrically deformed version of one of eight base models, given in a neutral pose. The dataset is further divided into two

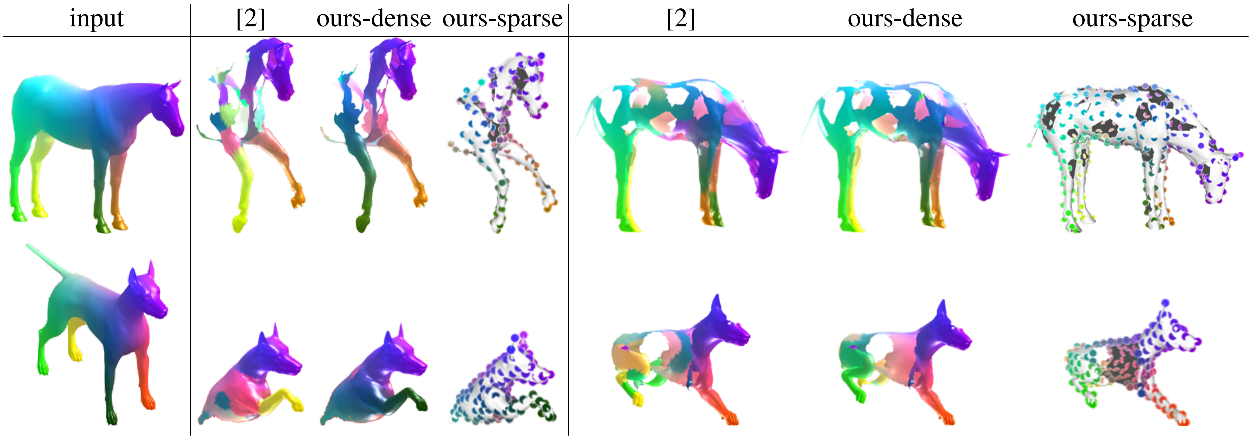


Fig. 6. Results on examples from SHREC'16A. The left column shows the input model in a neutral pose. The other columns compare our results (both dense and sparse) to the SOTA results of [2], for which the code is gratefully provided. The models are partial, deformed and have many holes. Our results outperform those of [2], when run with the default parameters. For instance, the front leg of the dog has the correct color in our result, whereas the result of [2] matched it to the rear leg (in yellow).

subsets, according to the type of partiality: (1) *cuts*, which is composed of shapes produced by dividing shapes by a plane, and (2) *holes*, obtained by eroding many areas around random vertices. *SHREC'16B* contains 10 shapes, which are derived from the same base human shape and underwent deformations and topological changes stemming from self-intersections. *FAUST* contains 60 pairs of high-resolution real-world scans of 10 different human subjects. The acquisition process introduces topological artifacts and missing parts due to occlusions.

Correspondence algorithms can be categorized into two classes, according to the density of the resulting correspondences: sparse and dense. Dense-correspondences algorithms match every vertex on one shape to a vertex on the other shape. Sparse-correspondence algorithms cover the surface by a sparse set of points and find correspondences only for them.

Our algorithm belongs to the latter class: it produces a sparse set of correspondences. However, as a post-processing step, we can convert the set of sparse correspondences to a dense set using the method of [16].

Qualitative results. Figure 5 illustrates our results on various shapes, which contain symmetries, large deformations, partiality and topological noise. In this figure, the input model $\mathcal{M}$ is color-coded according to the coordinates of the vertices. The matches on the target model $\mathcal{N}$ are colored according to the color of their corresponding vertices. Therefore, it is easy to visually verify the accuracy of our results, by comparing the color "by eye".

Figure. 6 compares our dense-correspondence results from *SHREC'16A* to those of [2] (which computes dense correspondence directly). Our method produces better results especially in cases of symmetries (e.g., the legs). This is due to distance preservation between points in Equation 6. This is particularly important when the model contains holes. This figure also demonstrates our results of sparse correspondence, where the dots on the model suit in color the matching parts of $\mathcal{M}$.

Figure 7 further demonstrates the quality of our results, by color-coding the errors. The larger the error, the more reddish the color is. It can be seen that our results hardly have any yellow/red, whereas the results of [2] have yellow/red regions.

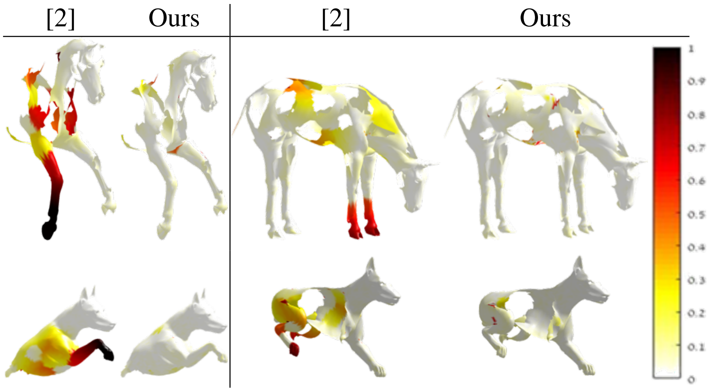


Fig. 7. Errors on examples from SHREC'16A. The error is color-coded from white (no error) to red (large error). Our results evidently are less erroneous than those of [2].

Similarly, Figure 8 shows a couple of examples from *SHREC'16B*, where the models have topological noise, i.e. regions that should not intersect semantically, do intersect geometrically (e.g., the triangles of the face and the head intersect). Generally, our method outperforms that of [2], having fewer and scarce failures. Moreover, our failures are constrained to regions near the topological noise.

Figure 9 compares our result to the reported failure case of [20], which introduces a deep learning model. It can be seen that our method is able to handle extreme partiality.

Quantitative results. Next, we provide a quantitative evaluation of our method on the above datasets w.r.t previously reported results. The common error metric used in previous work is the *normalized geodesic distance (NGD)* [63]. NGD is defined as follows: Let the corresponding point of $p \in \mathcal{N}$, as found by the algorithm, be $q \in \mathcal{M}$, and let the ground truth corresponding point of p be $q^* \in \mathcal{M}$. The error for p is the

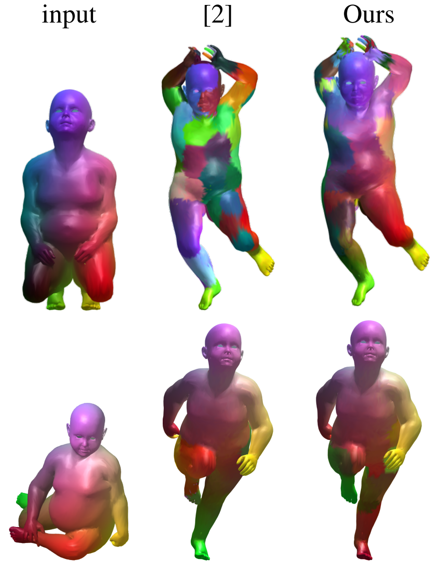


Fig. 8. Results on SHREC'16B. Top: the result of [2] contains more erroneous segments than ours (e.g., segments on the legs). Furthermore, our errors are more constrained to regions that indeed contain topological noise. Bottom: the legs are switched in [2], but not in our method. This is not only thanks to our similarity function that maintains distances, but also thanks to our multi-scale approach.

normalized geodesic distance between q and q^* on $\mathcal{M}$:

$$NGD(p) = \frac{GeoDist_{\mathcal{M}}(q, q^*)}{\sqrt{area(\mathcal{M})}}. \quad (7)$$

Figure 10 shows the cumulative curves, which indicate the percentage of correspondences falling below a varying threshold of NGD errors. The figure shows both sparse correspondences (dashed lines) and dense correspondences (solid lines), compared to other state-of-the-art algorithms [64, 16, 37, 65, 2, 66, 38], as provided in the benchmark site [23]. In both cases, our method considerably outperforms state-of-the-art algorithms on *SHREC'16A*, both on the subset of the dataset that contains models with holes and on the subset that contains partial models. The obtained increase in performance in 10% for the cuts subset and 20% for the holes subset.

Figure 11 shows the mean NGD error of the correspondence between a partial model $\mathcal{N}$ and a full model $\mathcal{M}$, as a function of their *partiality*. The latter is defined as the ratio between their surface areas:

$$1 - area(\mathcal{N})/area(\mathcal{M}).$$

It can be seen that our method is less dependent on the partiality of the model than other methods. This benefit of our method is mainly due to the inherent use of properties of the nearest-neighbor field (rather than explicit distances between descriptors), which is more robust, as well as due to our multi-scale approach.

Figure 12 compares our results to those of state-of-the-art algorithms [15, 23, 16, 65, 2, 67, 18] for *SHREC'16B*. Our method outperforms most of the other algorithms and is competitive with that of [18]. This is interesting since our algorithm heavily depends on geodesic distances, while topological noise shortens these distances.

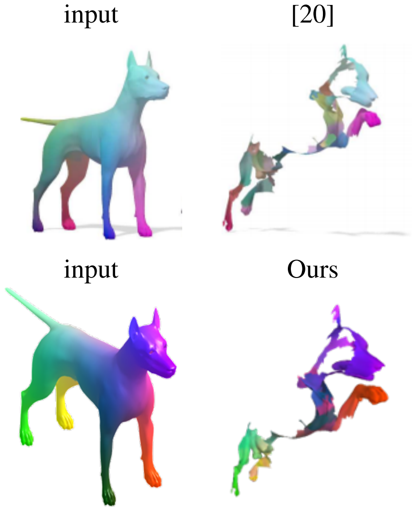


Fig. 9. Comparison with [20]. Our method manages to handle extreme partiality, compared to a result of a recent deep learning model.

Robustness to parameters. The parameters of the algorithm were tuned once on the *cuts* subset of the training set provided by SHREC'16A and used for the test sets of all datasets. Figures 13–15 show that our algorithm is robust to the choice of these parameters

In particular, Figure 13 shows the performance of the algorithm when changing the radius of the neighborhood used in the calculation of the FPFH in Stage 1 of the algorithm. It shows that the cumulative error curves achieved by choosing different radii hardly change.

Figure 14 shows the performance of the algorithm when changing the sizes of the patches (β) used in Stage 2 of the algorithm. It shows that the algorithm is fairly robust to the choice of this parameter, where the distance between any two curves is at most 3% of the matches (this happens for $NGD = 0.05$).

Finally, Figure 15 shows the influence of the threshold C on detecting outliers and replacing them in Stage 5 (refinement). As before, our algorithm is robust to the choice of C. The best result is obtained for $C = 1.15$, whereas the worst results is obtained for $C = 4$. This yields only a 1.5% increase in the percentage of matches (specifically, falling below an error threshold of $NGD = 0.05$). This can be explained by the fact that for most objects, our algorithm works well without refinement; yet for some objects (such as the half cat in Figure 3), refinement is beneficial.

Runtime and complexity. The average runtime of our algorithm for a pair of meshes from SHREC16'A is $\sim 50s$ on an Intel i7-4970. Out of the $50s$, $15s$ are devoted to the computation of the geodesic distances (stage 2) and $33s$ for the similarity calculation loop (stage 3). The other stages amount to $2s$ altogether. If densification is required, this adds $40s$. For comparison, the running time of [2] is $\sim 450s$ and of [16] $\sim 120s$. We note that the runtime of GPU-based deep learning methods, excluding training, is 1-4 seconds.

The asymptotic complexity of the algorithm is $O(|S|n^2) + O(n^2 logn)$, where n denotes the number of vertices of $\mathcal{N}$ and

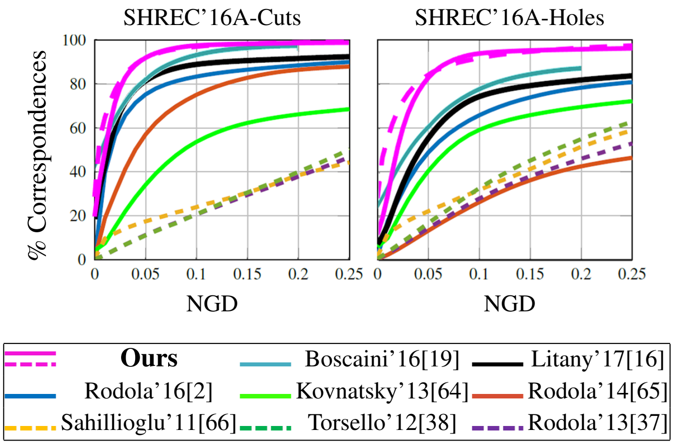


Fig. 10. Cumulative normalized geodesic error (NGD) curves on SHREC'16A. Our method (in magenta) outperforms other algorithms, both for dense correspondence (solid line) and for sparse correspondence (dashed line), on the two subsets of the dataset: cuts & holes.

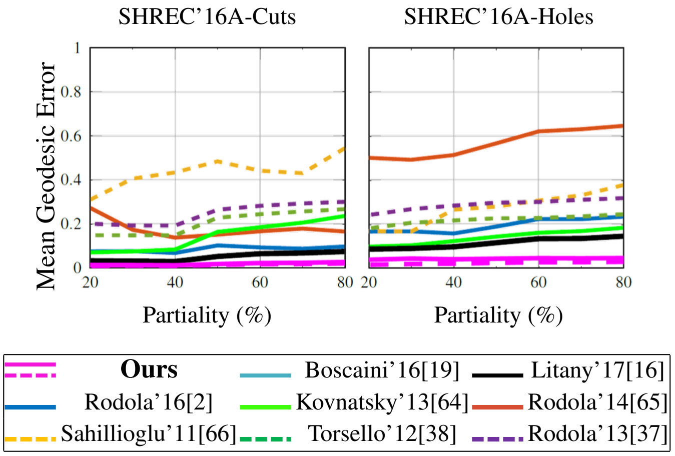


Fig. 11. Mean NGD as a function of model partiality. The error of our method increases less than the errors of other methods.

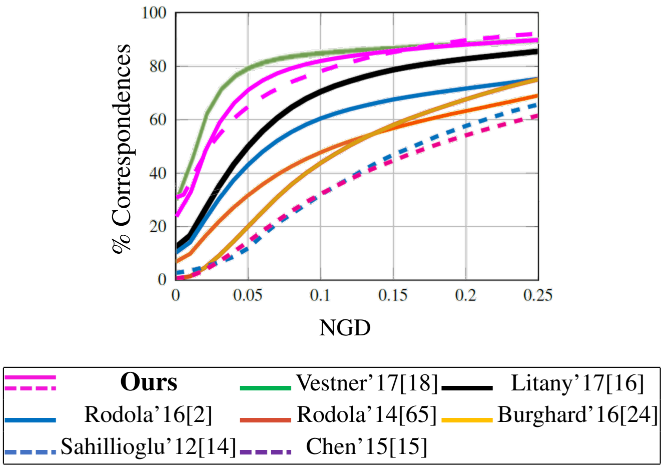


Fig. 12. Cumulative NGD on SHREC'16B. Our method is competitive with [18] and is better than other reported methods.

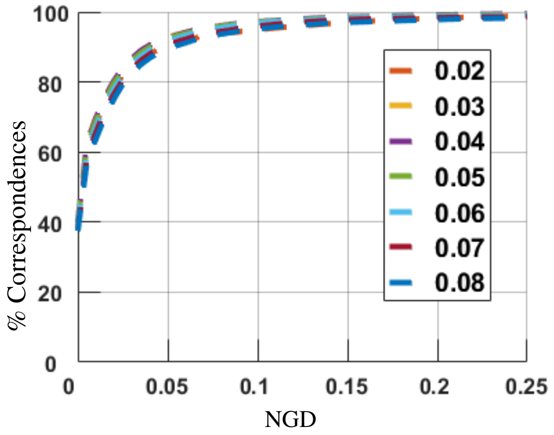


Fig. 13. Robustness of the algorithm for different radii of FPFH. The radius is calculated as a percentage of area of the mesh (e.g., $0.02\sqrt{area(\mathcal{M})}$).

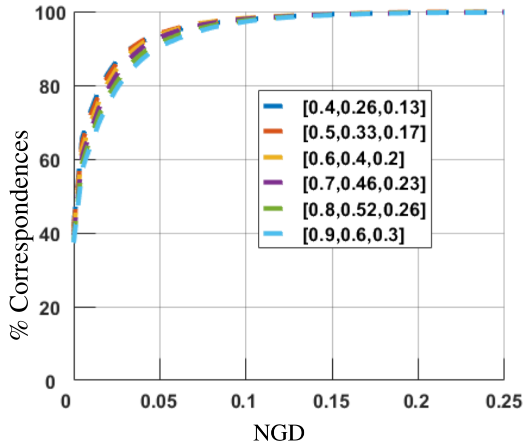


Fig. 14. Robustness of the algorithm for different patch radii (β) in Stage 2.

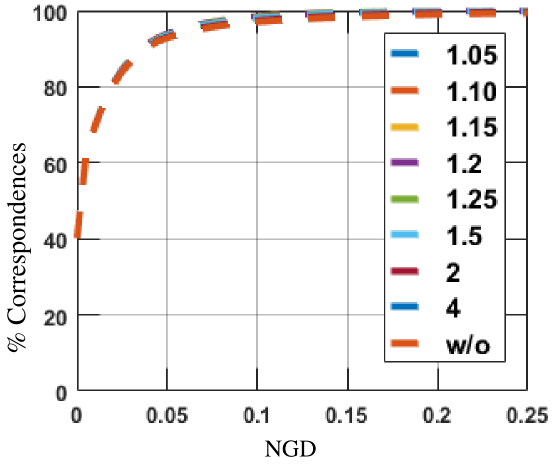


Fig. 15. Robustness of the algorithm to different choices of outlier thresholds (C) in Stage 5. Note that the lines are invisible, as they overlap.

$\mathcal{M}$ and $|S|$ is the number of samples. We elaborate on the complexity of each stage of the algorithm below.

In Stage 1, FPFH calculation is $O(n \cdot k)$ where k is the number of neighbors for each point [30]; the approximated nearest neighbor calculation takes $O(n \log n)$ on average [68]. Stage 2 is dominated by the computation of the geodesic distances between all pairs of points, which takes $O(n^2 \log n)$ [69]. Stage 3 computes the similarities between all pairs of patches (i.e., all patches of $\mathcal{M}$ and a sample of $|S|$ patches of $\mathcal{N}$). A pass over the vertices of a patch costs the size of the patch, which is bounded by n. Since this is performed for $O(|S|n)$ pairs of patches, the total complexity is $O(|S|n^2)$. Stage 4 simply computes the maxima of the similarities, which is $O(|S|n)$. Finally, Stage 5, which detects outliers, takes $O(|S|^2)$ (as the geodesic distances are already computed).

Implementation details. We implemented our code in C++ and used the Point Cloud library (PCL) [70] implementation for the FPFH shape descriptors and for the approximate nearest neighbor field computation. The entire code is parallelized using OpenMP. Since most of the work is devoted to computing the similarities between points, and the similarities are independent on each other, the obtained speedup is almost linear. Our implementation is available at `https://github.com/pitbullil/Partial-Correspondence-3D-NNF`.

Limitations. Figure 16 shows two types of failures, the first is due to strong topological noise and the other is due to highly-complex deformation. These are highly challenging models, on which other methods are unseccessful as well. The errors can be explained by the fact that the FPFH descriptors do not capture the shape sufficiently well and the geodesic distances are erroneous due to the elasticity of the models.

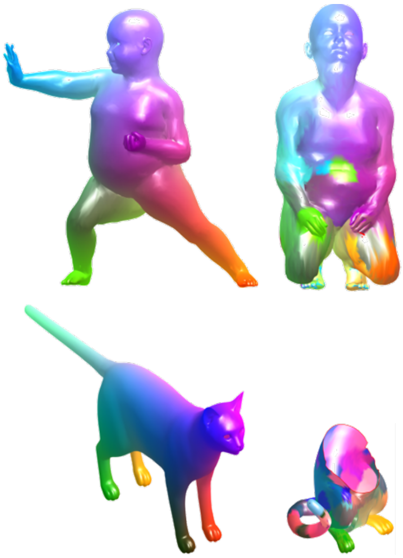


Fig. 16. Limitations. Our method might fail in cases of topological noise (top) or highly-deformed partial shapes (bottom), in which the cat's legs are folded and its tail is curled. Note that the topological noise may seem similar to the poses of FAUST in Figure 5, where our method works well. However, the noise here is much larger, as the entire arms are fused to the torso and the upper legs to the lower.

6. Conclusion

This paper has introduced a novel approach for finding correspondences between shapes in 3D. This approach is based on a simple observation: Statistical properties the nearest-neighbor field from the source surface to the target provide robust information about the correspondence. In particular, we use the diversity of the nearest-neighbor field and the consistency of the internal distances within the surface of corresponding points. Two additional ideas of our approach are the use of small subsurfaces when computing the similarity (rather than using the whole surface) and utilizing a multi-scale approach.

Our approach improves the state-of-the-art results both quantitatively and qualitatively on the challenging benchmarks of SHREC'16. In particular, these benchmarks contain examples having large deformations, symmetries, partiality of the shapes, and topological noise. We have demonstrated that our method is robust to the scale of partiality, as well as to its own parameters.

Acknowledgements. This research was supported by the Israel Science Foundation under Grants 1089/16 and 1083/18 and by the Ollendorf foundation. We are grateful to Dr. Uri Wollner for his insightful comments that helped us to improve the paper.

References

[1] Van Kaick, O, Zhang, H, Hamarneh, G, Cohen-Or, D. A survey on shape correspondence. Computer Graphics Forum 2011;30(6):1681–1707.

[2] Rodolà, E, Cosmo, L, Bronstein, MM, Torsello, A, Cremers, D. Partial functional correspondence. Computer Graphics Forum 2017;36(1):222–236.

[3] Biasotti, S, Marini, S, Mortara, M, Patane, G, Spagnuolo, M, Falcidieno, B. 3D shape matching through topological structures. In: International Conference on Discrete Geometry for Computer Imagery. 2003, p. 194–203.

[4] Barequet, G, Sharir, M. Partial surface and volume matching in three dimensions. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997;19(9):929–948.

[5] Mellado, N, Aiger, D, Mitra, NJ. Super 4pcs fast global pointcloud registration via smart indexing. Computer Graphics Forum 2014;33(5):205–215.

[6] Zeng, A, Song, S, Nießner, M, Fisher, M, Xiao, J, Funkhouser, T. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, p. 1802–1811.

[7] Dubrovina, A, Kimmel, R. Matching shapes by eigendecomposition of the laplace-beltrami operator. In: Proc. 3DPVT; vol. 2. 2010, p. 2.

[8] Lipman, Y, Funkhouser, T. Möbius voting for surface correspondence. ACM Transactions on Graphics (TOG) 2009;28(3):72.

[9] Maron, H, Dym, N, Kezurer, I, Kovalsky, S, Lipman, Y. Point registration via efficient convex relaxation. ACM Transactions on Graphics (TOG) 2016;35(4):73.

[10] Solomon, J, Peyré, G, Kim, VG, Sra, S. Entropic metric alignment for correspondence problems. ACM Transactions on Graphics (TOG) 2016;35(4):72.

[11] Vestner, M, Litman, R, Rodolà, E, Bronstein, AM, Cremers, D. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In: CVPR. 2017, p. 6681–6690.

[12] Biasotti, S, Marini, S, Spagnuolo, M, Falcidieno, B. Sub-part correspondence by structural descriptors of 3d shapes. Computer-Aided Design 2006;38(9):1002–1019.

[13] Itskovich, A, Tal, A. Surface partial matching and application to archaeology. Computers & Graphics 2011;35(2):334–341.

[14] Sahillioğlu, Y, Yemez, Y. Minimum-distortion isometric shape correspondence using em algorithm. IEEE transactions on pattern analysis and machine intelligence 2012;34(11):2203–2215.

[15] Chen, Q, Koltun, V. Robust nonrigid registration by convex optimization. In: IEEE International Conference on Computer Vision (ICCV). 2015, p. 2039–2047.

[16] Litany, O, Rodolà, E, Bronstein, AM, Bronstein, MM. Fully spectral partial shape matching. Computer Graphics Forum 2017;36(2):247–258.

[17] van Kaick, O, Zhang, H, Hamarneh, G. Bilateral maps for partial matching. Computer Graphics Forum 2013;32(6):189–200.

[18] Vestner, M, Lähner, Z, Boyarski, A, Litany, O, Slossberg, R, Remez, T, et al. Efficient deformable shape correspondence via kernel matching. In: 3D Vision (3DV). 2017, p. 517–526.

[19] Boscaini, D, Masci, J, Rodolà, E, Bronstein, M. Learning shape correspondence with anisotropic convolutional neural networks. In: Advances in Neural Information Processing Systems. 2016, p. 3189–3197.

[20] Litany, O, Remez, T, Rodolà, E, Bronstein, A, Bronstein, M. Deep functional maps: Structured prediction for dense shape correspondence. In: IEEE International Conference on Computer Vision. 2017, p. 5659–5667.

[21] Masci, J, Boscaini, D, Bronstein, M, Vandergheynst, P. Geodesic convolutional neural networks on riemannian manifolds. In: IEEE international conference on computer vision workshops. 2015, p. 37–45.

[22] Monti, F, Boscaini, D, Masci, J, Rodola, E, Svoboda, J, Bronstein, MM. Geometric deep learning on graphs and manifolds using mixture model cnns. In: IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 5115–5124.

[23] Cosmo, L, Rodolà, E, Bronstein, M, Torsello, A, Cremers, D, Sahillioglu, Y. Shrec16: Partial matching of deformable shapes. In: 3DOR; vol. 2. 2016,.

[24] Lähner, Z, Rodolà, E, Bronstein, M, Cremers, D, Burghard, O, Cosmo, L, et al. Shrec16: Matching of deformable shapes with topological noise. 3DOR 2016;2:11.

[25] Bogo, F, Romero, J, Loper, M, Black, MJ. Faust: Dataset and evaluation for 3d mesh registration. In: IEEE Conference on Computer Vision and Pattern Recognition. 2014, p. 3794–3801.

[26] Aubry, M, Schlickewei, U, Cremers, D. The wave kernel signature: A quantum mechanical approach to shape analysis. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE; 2011, p. 1626–1633.

[27] Bronstein, MM, Kokkinos, I. Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE; 2010, p. 1704–1711.

[28] Jain, V, Zhang, H, van Kaick, O. Non-rigid spectral correspondence of triangle meshes. International Journal of Shape Modeling 2007;13(01):101–124.

[29] Rusu, RB, Marton, ZC, Blodow, N, Beetz, M. Learning informative point classes for the acquisition of object model maps. In: Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on. IEEE; 2008, p. 643–650.

[30] Rusu, RB, Blodow, N, Beetz, M. Fast point feature histograms (fpfh) for 3d registration. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. 2009, p. 3212–3217.

[31] Sun, J, Ovsjanikov, M, Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion. In: Computer graphics forum; vol. 28. 2009, p. 1383–1392.

[32] Tombari, F, Salti, S, Di Stefano, L. Unique signatures of histograms for local surface description. In: European conference on computer vision. 2010, p. 356–369.

[33] Dey, TK, Li, K, Luo, C, Ranjan, P, Safa, I, Wang, Y. Persistent heat signature for pose-oblivious matching of incomplete models. In: Computer Graphics Forum; vol. 29. Wiley Online Library; 2010, p. 1545–1554.

[34] Coifman, RR, Lafon, S, Lee, AB, Maggioni, M, Nadler, B, Warner, F, et al. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. The national academy of sciences 2005;102(21):7426–7431.

[35] Ovsjanikov, M, Sun, J, Guibas, L. Global intrinsic symmetries of shapes. In: Computer graphics forum; vol. 27. 2008, p. 1341–1348.

[36] Wang, C, Bronstein, MM, Bronstein, AM, Paragios, N. Discrete minimum distortion correspondence problems for non-rigid shape matching. In: International Conference on Scale Space and Variational Methods in Computer Vision. 2011, p. 580–591.

[37] Rodola, E, Torsello, A, Harada, T, Kuniyoshi, Y, Cremers, D. Elastic net constraints for shape matching. In: IEEE International Conference on Computer Vision. 2013, p. 1169–1176.

[38] Rodola, E, Bronstein, AM, Albarelli, A, Bergamasco, F, Torsello, A. A game-theoretic approach to deformable shape matching. In: IEEE Conference on Computer Vision and Pattern Recognition. 2012, p. 182–189.

[39] Boscaini, D, Girdziušas, R, Bronstein, MM. Coulomb shapes: using electrostatic forces for deformation-invariant shape representation. In: 7th Eurographics Workshop on 3D Object Retrieval. 2014, p. 9–15.

[40] Boscaini, D, Masci, J, Rodolà, E, Bronstein, MM, Cremers, D. Anisotropic diffusion descriptors. Computer Graphics Forum 2016;35(2):431–441.

[41] Ovsjanikov, M, Ben-Chen, M, Solomon, J, Butscher, A, Guibas, L. Functional maps: a flexible representation of maps between shapes. ACM Transactions on Graphics (TOG) 2012;31(4):30.

[42] Aiger, D, Mitra, NJ, Cohen-Or, D. 4pointss congruent sets for robust pairwise surface registration. ACM Trans Graph 2008;27(3):85:1–85:10.

[43] Albarelli, A, Rodolà, E, Torsello, A. Fast and accurate surface alignment through an isometry-enforcing game. Pattern Recognition 2015;48(7):2209–2226.

[44] Bronstein, AM, Bronstein, MM, Bruckstein, AM, Kimmel, R. Partial similarity of objects, or how to compare a centaur to a horse. International Journal of Computer Vision 2009;84(2):163.

[45] Sahillioğlu, Y, Yemez, Y. Multiple shape correspondence by dynamic programming. In: Computer Graphics Forum; vol. 33. 2014, p. 121–130.

[46] Brunton, A, Wand, M, Wuhrer, S, Seidel, HP, Weinkauf, T. A low-dimensional representation for robust partial isometric correspondences computation. Graphical Models 2014;76(2):70–85.

[47] Cosmo, L, Rodolà, E, Albarelli, A, Mémoli, F, Cremers, D. Consistent partial matching of shape collections via sparse modeling. Computer Graphics Forum 2017;36(1):209–221.

[48] Huang, Q, Wang, F, Guibas, L. Functional map networks for analyzing and exploring large shape collections. ACM Transactions on Graphics (TOG) 2014;33(4):36.

[49] Ouyang, W, Tombari, F, Mattoccia, S, Di Stefano, L, Cham, WK. Performance evaluation of full search equivalent pattern matching algorithms. IEEE transactions on pattern analysis and machine intelligence 2012;34(1):127–143.

[50] Chen, JH, Chen, CS, Chen, YS. Fast algorithm for robust template matching with m-estimators. IEEE Transactions on signal processing 2003;51(1):230–243.

[51] Hel-Or, Y, Hel-Or, H, David, E. Matching by tone mapping: Photometric invariant template matching. IEEE transactions on pattern analysis and machine intelligence 2014;36(2):317–330.

[52] Tian, Y, Narasimhan, SG. Globally optimal estimation of nonrigid image distortion. International journal of computer vision 2012;98(3):279–302.

[53] Tsai, DM, Chiang, CH. Rotation-invariant pattern matching using wavelet decomposition. Pattern Recognition Letters 2002;23(1-3):191–201.

[54] Comaniciu, D, Ramesh, V, Meer, P. Real-time tracking of non-rigid objects using mean shift. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662); vol. 2. 2000, p. 142–149.

[55] Oron, S, Bar-Hillel, A, Levi, D, Avidan, S. Locally orderless tracking. International Journal of Computer Vision 2015;111(2):213–228.

[56] Aberman, K, Liao, J, Shi, M, Lischinski, D, Chen, B, Cohen-Or, D. Neural best-buddies: sparse cross-domain correspondence. ACM Transactions on Graphics (TOG) 2018;37(4):69.

[57] Dekel, T, Oron, S, Rubinstein, M, Avidan, S, Freeman, WT. Best-buddies similarity for robust template matching. In: IEEE Conference on Computer Vision and Pattern Recognition. 2015, p. 2021–2029.

[58] Talmi, I, Mechrez, R, Zelnik-Manor, L. Template matching with deformable diversity similarity. In: IEEE Conf. on Computer Vision and Pattern Recognition. 2017, p. 1311–1319.

[59] Raviv, D, Dubrovina, A, Kimmel, R. Hierarchical matching of non-rigid shapes. In: International Conference on Scale Space and Variational Methods in Computer Vision. Springer; 2011, p. 604–615.

[60] Rusu, RB, Marton, ZC, Blodow, N, Dolha, M, Beetz, M. Towards 3d point cloud based object maps for household environments. Robotics and Autonomous Systems 2008;56(11):927–941.

[61] Shtrom, E, Leifman, G, Tal, A. Saliency detection in large point sets. In: Proceedings of the IEEE International Conference on Computer Vision. 2013, p. 3591–3598.

[62] Katz, S, Leifman, G, Tal, A. Mesh segmentation using feature point and core extraction. The Visual Computer 2005;21(8-10):649–658.

[63] Kim, VG, Lipman, Y, Funkhouser, T. Blended intrinsic maps. ACM Transactions on Graphics (TOG) 2011;30(4):79.

[64] Kovnatsky, A, Bronstein, MM, Bronstein, AM, Glashoff, K, Kimmel, R. Coupled quasi-harmonic bases. Computer Graphics Forum 2013;32(2pt4):439–448.

[65] Rodolà, E, Rota Bulo, S, Windheuser, T, Vestner, M, Cremers, D. Dense non-rigid shape correspondence using random forests. In: IEEE Conference on Computer Vision and Pattern Recognition. 2014, p. 4177–4184.

[66] Sahilliolu, Y, Yemez, Y. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. Computer Graphics Forum 2011;30(5):1461–1470.

[67] Sahillioğlu, Y, Yemez, Y. Scale normalization for isometric shape matching. In: Computer Graphics Forum; vol. 31. 2012, p. 2233–2240.

[68] Bentley, JL. Multidimensional binary search trees used for associative searching. Communications of the ACM 1975;18(9):509–517.

[69] Kimmel, R. Fast marching methods for computing distance maps and shortest paths 1996;.

[70] Rusu, RB, Cousins, S. 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). 2011,.