Partial Matching of 3D Shapes using Deformable Diversity

Anonymous CVPR submission

Paper ID ****

Abstract

We propose a novel approach for the matching of partial deformable shapes in 3D. Inspired by recent advances in 2D template matching techniques, our method relies on the concept of deformable diversity similarity(DDIS), extends and adapts it from an image to the 3D shape domain, and leverages the distinct behavior of this framework in different scales to achieve shape correspondences. We evaluate this framework on the SHREC16 partial matching of deformable shapes and show state of the art performance in achieving sparse correspondences. **Currently done Section 3 & 4**

1. Introduction

Shape correspondence is a fundamental and challenging problem in computer vision and graphics. It has usage in various applications such as transferring texture and animation. Shapes rarely, if ever manifest in only one pose. While rigid transformations between surfaces is a well researched topic with many adequate solutions, a more challenging problem arises when a shape is deformed non-rigidly, a case all too common for people, animals and objects. Moreover, the shape acquisition process almost always lead to partiality of the scanned object. Occlusions arise from different angles of acquisition, which cause an object to occlude itself, or stem from other occluding objects. An additional type of difficulty which might be occur is topological noise, occurring when shapes touch pn another, thus making sensors unable to seperate them. All of these combined give rise to the challenging problem of partial correspondences, where a deformed and incomplete shape, possibly with topological changes, has to be matched with its full version. The goal of this paper is to deal with this challenging problem.

While in a rigid setting the problem can be solved by RANSAC and ICP like approaches[28, 11], extending these to non-rigid case produces mediocre results due to an underlying assumption of small deformations. Early methods specialized for the non-rigid problem focused on minimization of intrinsic metric distortion[7, 39] and regularity of parts[**?**, 5]. These methods all contain with them a global assumption of isometry which holds only approximately, these tended to break down with it, and are also unable to handle extreme partiality. Another family of method is based on functional correspondence. These methods model correspondences as a linear operator of a known nature between a space of functions on manifolds[22]. These methods, originally designed for the full shape correspondence scenario have achieved state of the art results on various partial matching tasks in the recent years[18, 40, 26], and produce dense correspondence maps, but are not parallelizable, and their reliance on intrinsic metrics makes them invariant to symmetry.

We take a different approach. We take advantage of the fact that while the isometric property tends to break over large distances, it usually holds approximately in limited environments. These also tend to suffer a lot less from boundary effects, especially when concentrated around the extremities of a shape.

We can thus treat the problem of partial correspondences as matching of multiple templates, each smaller then the partial surface centered around shape landmarks.

In addition, since point descriptors are known to be modified by partiality and deformations, instead of using them directly, we follow the approach of[37](**DDIS**) which tackles template matching in 2D and use simple statistical assumptions on the nature of nearest neighbors between small patch descriptors, along with the assumption of an approximate conservation of distances in medium environments to obtain similarity scores between these partial shape templates.

We analyze the behavior of DDIS similarity in different scales and devise a multi scale scheme which leverages the advantages of each scale while masking their shortcomings.

We show that using this approach, we are able to generate a set of sparse correspondences, which are less prone to symmetrical assignment than functional correspondence reliant methods, and are of superior quality on the SHREC16 Partial matching challenge[9]. We then demonstrate how these sparse correspondences can be used as an input to existing functional correspondence algorithms to obtain dense correspondences or a higher quality. In summary, our contributions are:

- A non trivial extension of Deformable Diversity from 2 to 3 Dimensions.
- A modified DDIS similarity measure which is more well suited to handle matching of templates with a different number of points.
- An empirical analysis of DDIS behavior in different scales, leading to an improved multi-scale framework.
- A multi-template approach to partial matching of deformable shapes which can both produce state of the art sparse correspondences, and be used as an input to functional correspondence algorithms, significantly improving the results obtained by these.

The rest of the work is organized as follows: in section 2 we go over related works in the field of shape analysis. Section 3 introduces our Deformable Diversity framework for 3D shape matching. Experiments and results are given in section 4, and the conclusions are in section 5.

2. Related work

2.1. Matching Of Deformable Surfaces

As a fundamental problem in computer graphics and vision, an extensive body of work have been done on the matching of surfaces. A variety of shape descriptors have been devised for this task which can be roughly divided in to 2 families. Extrinsic ones, such as PFH[30], SHOT[38] and FPFH[28] which are usually calculated in euclidean space and are thus sensitive to non rigid deformations, but can discern between reflections and are also more robust to noise, topological artifacts and boundary effects. On the other hand intrinsic features such as Heat[8] and Wave Kernel signatures[2] are invariant under isometric transformations, but are very sensitive to partiality and are unable to discern between symmetric parts. These have been commonly used to generate rough correspondences between surfaces and point clouds based on their similarity, but are noisy and offer little in terms of bijectivity and continuity of the solution. a measure of global consistency using these can be achieved by solving an energy minimization of the disimilarity matrices steming from an assignment, and the auction algorithm has been commonly employed for this purpose. Other methods use pairwise relations between points such as geodesic distances[32, 33, 34], and search for a configuration which minimizes the distortions of these. These methods usually carry a high complexity, both due to calculating the pairwise relations, and the combinatorial configuration search, and are thus either obtain sparse matches[32, 33, 34] to alleviate this complexity, or used strategies such as coarse to fine solutions. Another common approach has been to embed the shapes into a different lower dimension "canonical" space, this has been done by generalized MDS[7], an embedding into the mobius group[17], or by representation in the LBO basis[35] A notable family of works are derived from functional correspondences. Introduced at[22, 24, 15, 40] these assume that functions can be mapped from one manifold to another via a linear operator, finding this transfer operator allows to embed point in a space where the ICP method can obtain correspondences. Lately there has been a large body of works which employ learning methods such as Random Forests[25] and deep learning architectures[19, 4, 20]. These show the promise of achieving state of the art performance, but require a lot of annotated data.

2.2. Partial Matching of Deformable shapes

The introduction of partiality adds complications which are not present in the full correspondence scenario. Spectral quantities change drastically, while geodesic paths disappear. For the rigid setup, the Iterative Closest Point(ICP)[1] algorithm, preceded by initial alignment[31] tackle partial matching successfully. Adapting this to the rigid setup however has proved to have limited success due to the alignment which is necessary, and thus is only fit for very small nonrigid deformation.

Early works which were designed with partial matching in mind[5, 6] formulated an energy minimization problem over metric distortion and regularity of corresponding parts. Following works relaxed the regularity requirement by allowing for sparse correspondences[39, 27]. Other works[33, 32] minimized the distortion metric over the shape extremities by doing combinatorial search of least distortion matches and then densify them while employing a refining scheme in the process.

In[23] a bag of words point-wise descriptors on a part in conjunction with a constraint on area similarity and the regularity of the boundary length to produce correspondence less matching parts without point to point correspondences by energy minimization.

Another line of works employ machine learning techniques to learn correspondences between manifolds. Recently [26] had proven that partiality induces a slanted diagonal structure in the correspondence matrix and found the Laplacian eigenfunctions from each basis which induces this structure. Current state of the art[18] uses this notion in conjunction with joint diagnolization. The main drawback of this method, shared with other intrinsic methods, is its invariance to symmetries.

3D Shape Descriptors

2.3. Template matching in 2D

Template matching in 2D is a well researched topic. Similarly to 3D objects are going complex deformations of pose, and are only seen partially depending on the camera point of view. Recently a series of works which use a very simplistic framework based on the statistical properties of nearest neighbors in low level feature space had made good strides in tackling this complex task.

Best Buddies Similarity Great strides had been achieved in the field of 2D template matching. Best Buddies Similarity[10] is a simple framework which employs a statistical assumption - if two regions $\mathcal{N}$,$\mathcal{M}$ contain the same template patches should maintain Bi Directional Similarity. That is - given a point $n_i \in \mathcal{N}$ and a corresponding point $m_i \in \mathcal{M}$ they should point too each other as nearest neighbors - that is if $NN_{\mathcal{M}}(n_i) = m_j$ then on a matching template we should expect $NN_{\mathcal{N}}(m_j) = n_i$. Solving for a matching template then amounts to finding the region which has the highest count of best buddies. This amazingly simple scheme has been show to be able to handle occlusions, missing parts and complex deformations of templates.

Deformable Diversity Similarity Building upon the above work, [37] relaxed the requirement for a best buddy relation, and added a requirement for spatial coherency.

The rather cumbersome best buddy relation has been relaxed to requiring only that the diversity of the set of nearest neighbors sets between corresponding templates should be high. This is actually prerequisite to a high best buddies similarity score and serves as a rough approximation of it. For this end diversity is formally defined as:

$$\underset{\mathcal{N}\rightarrow\mathcal{M}}{DIS} = c \cdot |\{n_i \in \mathcal{N} : \exists m_j \in \mathcal{M}, NN(m_j, \mathcal{N}) = n_i\}| \quad (1)$$

where $|.|$ denotes group size and $c = 1/min(|\mathcal{M}|, |\mathcal{N}|)$ is a normalization factor. Between non corresponding windows, indeed one should expect most points to have no real corresponding point, and thus be mapped to a very and remote nearest neighbors. On the other hand, regions containing matching objects are drawn from the same distribution, thus the diversity of nearest neighbors should be high. To accommodate this assumption not only did they rewarded high diversity of nearest neighbors, but also penalized mapping to the same patch. To this end, another, a negative diversity measure had been defined:

$$\kappa_{\mathcal{M}}(n_i) = |\{m \in \mathcal{M} : NN^a(m, \mathcal{N}) = n_i\}| \quad (2)$$

With x_i^a denoting the appearance descriptor of point x_i. Thus the contribution of a patch $m_j : NN^a(m_j, \mathcal{N}) = n_i$ is $exp(1 - \kappa_{\mathcal{M}}(n_i))$. An additional observation made has been that while non isometric deformations do occur, they should be restricted, small, in real objects. With distance on the window pixel grid between 2 nearest neighbor points defined as $r_j = d(m_j^l, n_i^l)$ with x_i^l denoting the location of x_i on a grid, the final Deformable Diversity Similarity formulation becomes:

$$\underset{\mathcal{N}\rightarrow\mathcal{M}}{DDIS} = c \sum_{m_j \in \mathcal{M}} \frac{1}{1 + r_j} \cdot exp(1 - \kappa(NN^a(m_j, \mathcal{N}))) \quad (3)$$

3. General Approach

Given two surfaces $\mathcal{M}$ and $\mathcal{N}$, the goal is to find the best match of $\mathcal{N}$ within $\mathcal{M}$. In particular, we aim at extracting a sparse set of point correspondences between the models. Our approach is based on three key ideas, which we describe hereafter.

First, inspired by [37], similarity is captured by two properties of the Nearest Neighbor field. (1) When $\mathcal{N}$ and a patch of $\mathcal{M}$ match, most points in $\mathcal{M}$ have a unique NN-match 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. (2) Arbitrary matches typically imply a large deformation, whereas correct matches should preserve the distance between pair of points. Therefore, Similarity should be based both on the diversity of the Nearest-Neighbor field and on the consistency of the distances between the points.

Second, rather the realizing the similarity test, described above, on $\mathcal{N}$ as a whole, it is preferable to perform it on a set of small sub-surfaces of $\mathcal{N}$. This is so not only since a small sub-surfaces is more likely to exhibit consistent distances, but also since it is less likely to be matched to a repeating pattern, which would lead to smaller diversity.

Third, a multi-scale approach with respect to the size of matched sub-surfaces is beneficial. This is so since larger surfaces contain more global context, resulting in matches which lie in a correct region, but provide poor localization. On the other hand, matching smaller surfaces lead to results which are better locally, but may be globally inconsistent. **AT: what do you mean by globally inconsistent?**

Therefore, our algorithm, which is illustrated in Figure 1, consists of the following steps.

1. **Pre-processing**. Shape descriptors are calculated for every vertex of both meshes and an approximate nearest neighbor field is computed for the vertices, as described hereafter.

   Many descriptors have been proposed in the literature [31, 38, 36]. We use the FPFH [28], which is robust to small deformations and partiality of the data, yet sensitive to symmetrical flips. **AT: what makes it sensitive to symmetrical flips?** Therefore, it addresses a major drawback of matching a right arm, for example, to the left one. We then compute a nearest neighbor field mapping, by assigning each vertex of $\mathcal{M}$ its nearest neighbor in $\mathcal{N}$, FPFH-wise.

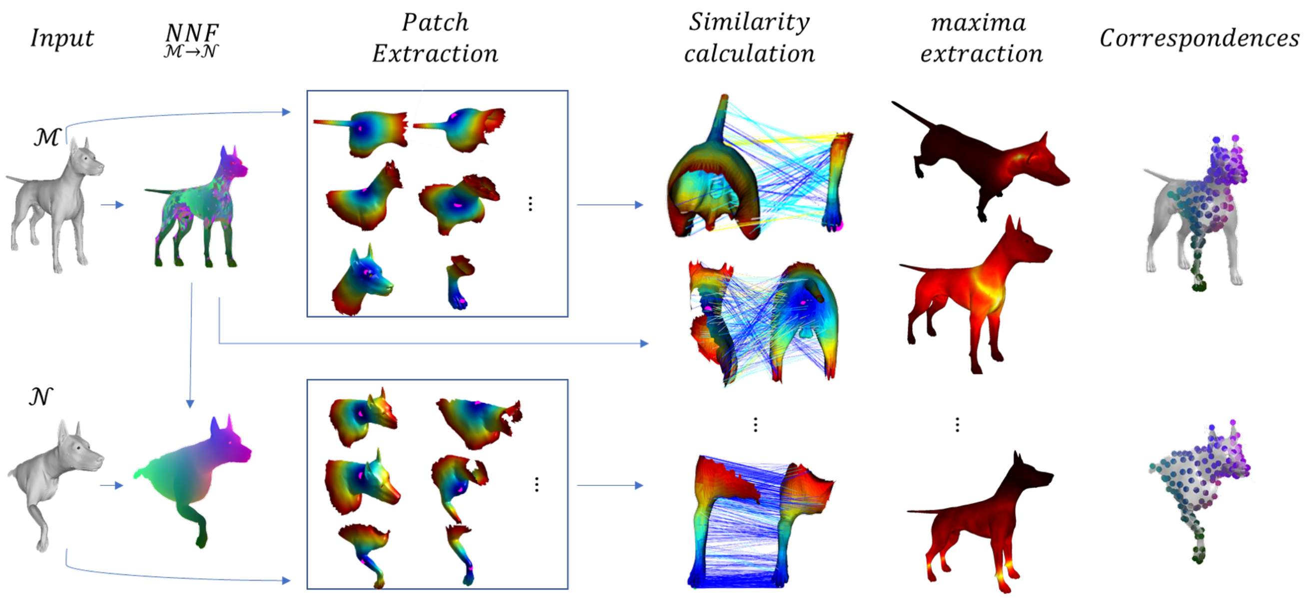

Figure 1. **Algorithm outline.** In the first step, we compute the nearest-neighbor field for $\mathcal{N}$ and $\mathcal{N}$. Then, patches of the surfaces are extracted for every sample point in $\mathcal{N}$ and for every vertex on $\mathcal{M}$. These patches cover the surface (i.e., may overlap) and represent semantic regions. Note that exact segmentation is not needed. Step 3 is the core of the algorithm, in which the similarity between the patches is computed. Finally, in Step 4, for every sample of $\mathcal{N}$ we set the vertex of $\mathcal{M}$ that achieves the maximal score as its corresponding point. **AT: This figure should be re-done**

2. **Patch extraction.** Inline with the second key idea, we aim at extracting a meaningful set of sub-surfaces, which cover (rather than partition) the surface. This is done in two steps: First, we extract a meaningful set of points, whose neighborhoods provide a good cover of the surface. We then extract the patches using this sample. We elaborate hereafter.

   To extract the sample point set, we start from the extremities of the surface, which are considered salient points. A vertex is considered to be an extremity if it resides on a tip of the surface (e.g., tips of limbs) [12], In practice, we define them to be vertices that are local maxima of the sum of the geodesic distance functional. Formally, $\forall v \in S$, 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 of mesh S. Vertex v is an extremity if it satisfies

$$\sum_{v_i \in S} GeoDist(v, v_i) > \sum_{v_n \in S} GeoDist(v_n, v_i). \quad (4)$$

   Then, we iteratively add more samples, choosing the next sample point 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 next point to be added to the set is a vertex whose geodesic distance to any sample point in the set is minimal and does not fall in any of the forbidden regions. This process stops when the entire surface is marked forbidden.

   Once the set of representing sample set is defined, a disc (sub-surface) of geodesic distance R_T is extracted around each sample point, which is the sought-after set of patches. Specifically, $R_T = \beta \cdot \sqrt{Area(\mathcal{M})}$. As our approach is multiscale, 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 key idea.

   For each pair of patches of the same scale, $Q_i \subset \mathcal{M}$ and $P_i \subset \mathcal{N}$, we compute a similarity value. Recall that our goal is to reward a nearest-neighbor field with high diversity and low deformation. We will define the similarity function $DDIS$ that achieves it in Section 4. This is done in a multi-scale manner.

4. **Extracting a sparse set of corresponding points.** Given the similarity values between the patches, our goal now is to extract a set of corresponding points between $\mathcal{N}$ to $\mathcal{M}$. 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.

   In our multi-scale approach, we proceed from coarse to fine. Suppose that $P_i \subset \mathcal{N}$ and $Q_j \subset \mathcal{M}$ were found to have the highest similarity in a coarsest scale (i.e. Q_j is the largest). The coarsest match 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. the sample points that define the patches in Step 2). When moving to the finer scale, we replace Q_j with a smaller (finer) patch in which w_j is the center and set the new w_j to be the vertex on this patch that maximizes the similarity function $DDIS$ on this patch. The finest w_j is the corresponding point of v_i; see Figure 2.

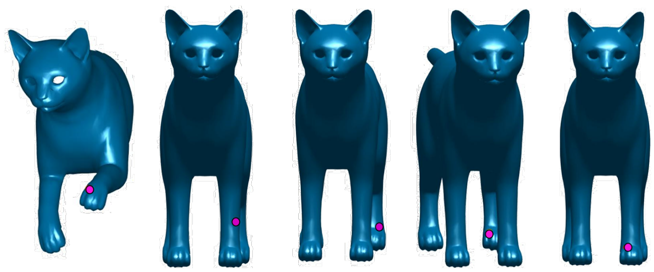

Figure 2. **Multi-scale similarity.** Given the sample point on the left (in magenta), the corresponding points on $\mathcal{M}$ are shown on the right. In the coarsest level, the general region of the matching point is found, but the point is imprecise. In subsequent levels, the general region is not found. Our multi-scale approach manages to find the precise point. **AT: Replace the image.** **NA: done**

5. **Coherency-based correspondence refinement** The result of Step 4 is a set of corresponding pairs of points. In most cases ($> 92\%$ on 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 on 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.

   To detect these outliers, we check the sum of geodesic distances. For a pair of points (v, w) to be considered correct, they should satisfy:

$$\sum_{v_i \in \mathcal{M}} |GeoDist(v, v_i) - GeoDist(w, w_i)| < C \quad (5)$$

   where w_i is the corresponding point of v_i and C is 0.15 of the mean of Equation (5) on all points. We replace the corresponding point of an outlier by a point that minimizes Equation (5) and is a local maximum of the similarity function of Step 3.

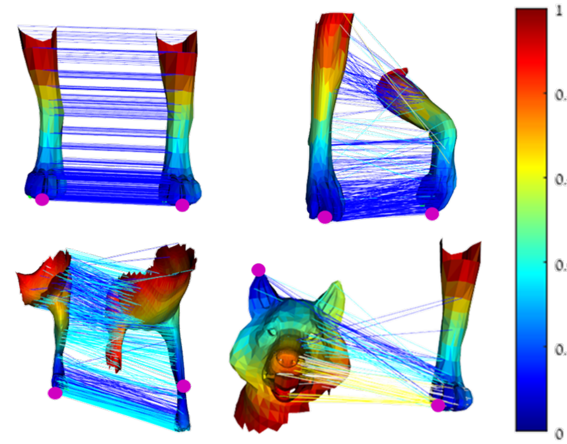

Figure 3. **Nearest Neighbor Field.** The surfaces are colored by the geodesic distances from the magenta point; the lines are colored by the **NA: difference of geodesic distances from the purple point**. **AT: deviation of what?** Clearly, similar surfaces on the top (even when deformed) exhibit diversity in matching (i.e. different points on one surface are matched to different points on the other). Furthermore, in this case, most lines are blue, which indicates similar distances from the source point. This is not the case at he bottom, where the surfaces are highly different from one another.

4. Similarity

This section elaborates on Step 3, which is the core of the algorithm. Given pair of patches of the same scale, $Q \subset \mathcal{M}$ and $P \subset \mathcal{N}$, this section defines a similarity function, termed *3D Deformable Diversity Similarity (3D-DDIS)*, between them. This function should be oblivious to non-rigid transformation, different resolutions of the meshes, noise, and partiality of the data.

Recall that the key idea is to reward a point (center of the patch) whose nearest-neighbor field satisfied two properties: it has both high diversity and low deformation. As for diversity, when Q and P correspond, most of the points on Q points 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, and therefore 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 the NN-field is highly *diverse*, pointing to many different points in P. In addition, if two patches correspond, pairs matching (nearest neighbors) points tend to have similar geodesic distances to the centers of the patches they reside on. Conversely, arbitrary matches typically do not maintain such distances. **AT: why is it called deformation?**

To realize these requirements, we start by formally defining the diversity, $Div(Q, P)$, and the deformation, $Def(Q, P)$, and then show how to put them together into a similarity function. An intuitive and efficient way to measure diversity is to count the number of unique NNs between the points of Q and P:

$$Div(Q,P) = |\{p_i \in P : \exists q_j \in Q, NN(q_j, P) = p_i\}|, \quad (6)$$

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.

Next, we define $Def(Q,P)$, the deformation from Q to P. Let $p \in P$ and $q \in Q$ 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:

$$def(q_j, p_i, Q, P) = |GeodDist(q_j, q) - GeodDist(p_i, p)\epsilon|, \quad (7)$$

where $0 < \epsilon << Area(\mathcal{M})$ and is a used for numerical stability.

For each $p_i \in P$, we find the minimal deformation $r_i^* = min_{q_j \in Q} def(q_j, p_i, Q, P)$ such that p_i is the nearest neighbor of q_j. Note that some points in P might not be associated with any point in Q (since they are not the nearest neighbors of any point $q_j \in Q$); in this case $r_i^* = \infty$.

Having defined diversity between patches and deformation between points on these patches, we define the similarity between patches p and Q as:

$$DDIS(P,Q) = \sum_{p_i \in \mathcal{P}} \frac{1}{1 + r_i^*}. \quad (8)$$

It is easy to see how the 3D-DDIS rewards low deformation. However, we will explain hereafter why it also rewards high diversity. Consider a case where $r_i^* \in \{0, \infty\} \forall p_i$. When this occurs $\frac{1}{(1+r_i^*)} \in \{1, 0\}$ and it indicates that a point p_i has a point in Q that considers p_i to be its nearest neighbor. In this scenario, 3D-DDIS simply counts the number of points in Q that are nearest neighbors of some point in P. But, this is precisely the diversity function we seek-after. **AT: copy a convincing explanation for the general case** **NA: In the general case, the contribution of every point is inversely weighted by its deformation r_i^*, which gives preference to plausible deformations of real objects.**

5. Results

We have evaluated our method both qualitatively and quantitatively on several datasets: (1) the benchmark of *SHREC'16A—partial matching of deformable shapes* [9]; (2) the even more challenging benchmark of *SHREC16B—matching of deformable shapes with topological noise* [16]. **AT: (3) Faust, (4) Archaeology** **NA: (3) *FAUST*[3] includes real scans of human subjects. For this set we provide only qualitative evaluation. NA: (4) *Archeological artifacts*[3] which include different archeological artifacts.** In all cases, our method either outperformed the state-of-the-art methods or was competitive.

SHREC'16A contains 400 partial shapes, each is a near-isometrically deformed version of one of eight base model, given in a neutral pose. The dataset is further divided into two 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. *SHREC16B* is contains 10 shapes, which are derived from the same base human shape that undergoes deformations and topological changes stemming from self-intersections.

NA: *FAUST* contains 200 real world scans of 10 different human subjects. The acquisition process of which introduces topological artifacts and missing parts due to occlusions.

AT: paragraph: explain that there are two distinct, yer related challenges: sparse and dense **NA: Surface correspondence algorithms can divided into 2 by the density of their correspondences. (1) Sparse correspondences which aim to cover the surface area uniformly, but sparsely. (2) Dense correspondences which match every vertex on one shape to the other shape.** **AT: paragraph: explain that you compute only sparse and then use... to convert it into dense.** **NA: Our method produces sparse correspondences, yet these can be converted to dense ones with a minimal loss of quality. This is achieved by using the sparse correspondences as input to the method of [18].**

Qualitative results: Figure 4 illustrates our results on two models from *SHREC'16A*. In this figure, the input model $\mathcal{M}$ is color-coded according to its coordinates. The matches on the partial model $\mathcal{N}$ (partial models and holes) are colored according to the match. Therefore, it is easy to visually verify if the match is correct or not.

It can be seen that our method obtains sparse correspondences of a high quality, since the dots on the model suit in color to the matching parts of $\mathcal{M}$. Furthermore, when comparing our dense-correspondence results to those of [26] (which computes dense correspondence directly), our method produces better results. It should be noted that symmetries are less of a problem in our method, due to distance preservation between points in Equation (8) (note the legs). This is especially important when the model contains holes.

Figure 5 further demonstrates it, by color-coding the errors. The larger the error, the more reddish the color is. **AT: add this figure** Figure 6 shows a couple of examples from the *SHREC'16B* dataset. On the topological noise benchmark we can see bad matches are typically limited to the

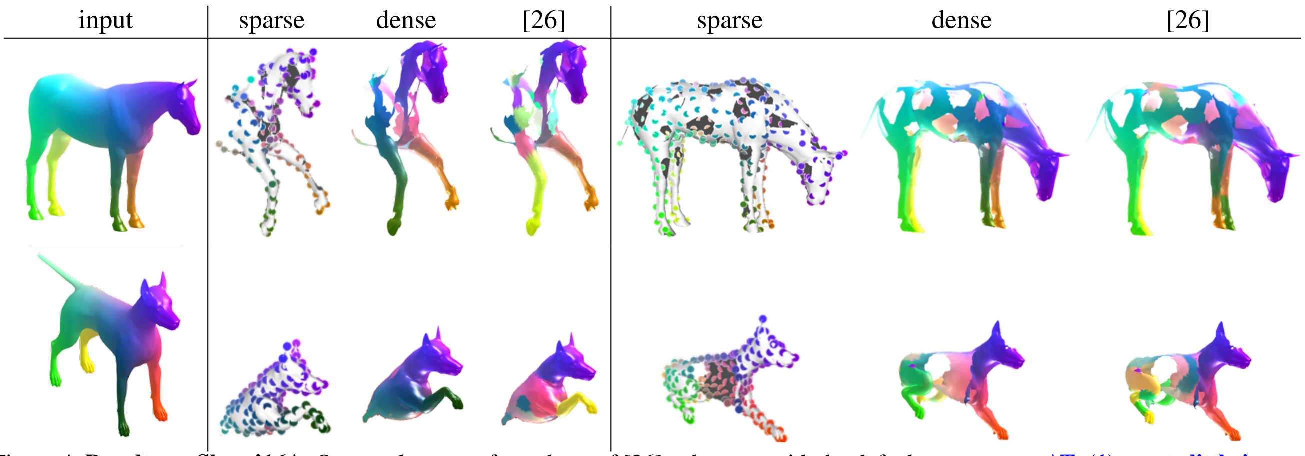

Figure 4. **Results on Shrec'16A.** Our results outperform those of [26], when run with the default parameters. **AT: (1) create little images (2) remove boundaries (3) change the model color in sparse correspondences (4) the models must be in the same size**

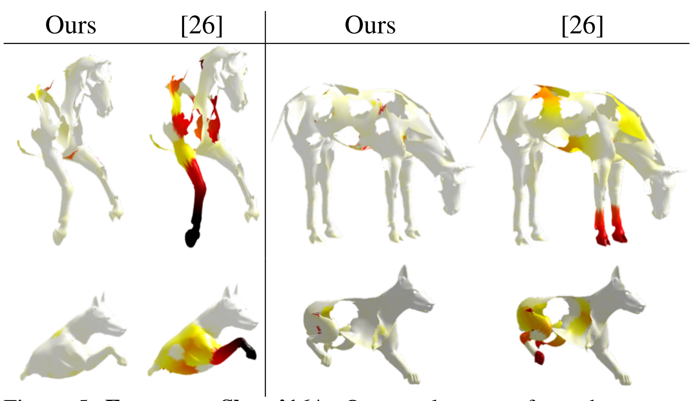

Figure 5. **Errors on Shrec'16A.** Our results outperform those of [26]

area of the topological noise, while PFM breaks completely on occasions.

NA: TBA: Faust

Quantitative results: Next, we provide quantitative evaluation of our method on the above datasets w.r.t previously reported results. The common error metric used is the normalized geodesic distance [13]. Specifically, let the corresponding point of ($q \in \mathcal{N}$, as found by the algorithm, be $p \in \mathcal{M}$) and let the ground truth corresponding point of q be $p^* \in \mathcal{M}$). The error of for q is the normalized geodesic distance between p and p^* on $\mathcal{M}$:

$$\varepsilon(q) = \frac{GeoDist_{\mathcal{M}}(p, p^*)}{\sqrt{area(\mathcal{M})}} \tag{9}$$

Figure 7 shows the cumulative curve, which indicates the percentage of errors falling below a varying geodesic threshold on *SHREC'16A*. The figure shows both sparse correspondences (dashed lines) and dense correspondences (solid lines), compared to other state-of-the-art algorithms [18, 26], **AT: add all references** as provided in the benchmark site. **AT: cite the site** In both cases, our method considerably outperforms state-of-the-art algorithms, 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 by 10% for the cuts subset and by 20% for the holes subset.

Figure 6. **Results on Shrec'16B.** Our results outperform those of [26].

AT: Add a paragraph describing Figure 8 **NA: Fig-**

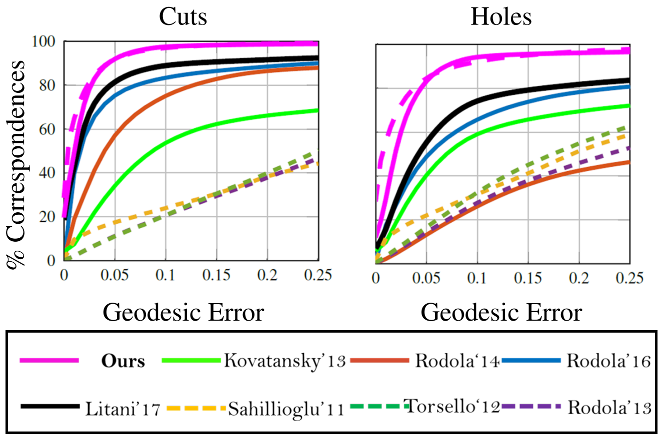

Figure 7. **Cumulative geodesic error curves on Shrec16A.** Our method (in magenta) outperforms other algorithms, both for the dense correspondence and for the sparse correspondence, on the two subsets of the dataset. **AT: 1. three images, (2) [Author'19] or Ours**

ure 8 shows the mean geodesic error of the mapping from a partial model $\mathcal{N}$ to the full model $\mathcal{M}$ as a function of $area(\mathcal{N})/area(\mathcal{M})$, hereafter referred to as partiality. It can be seen that our method is weakly dependent on the partiality of the model.

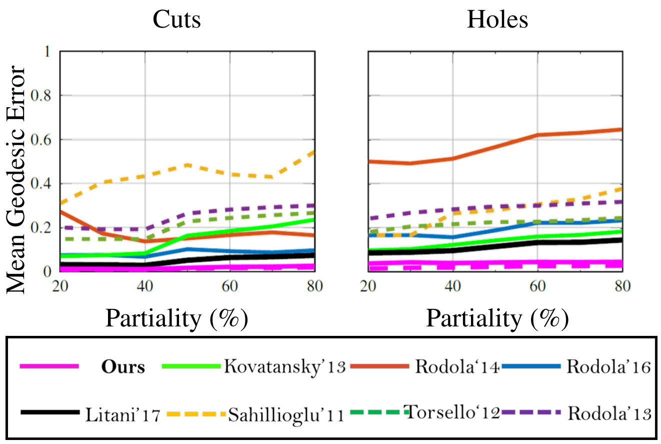

Figure 8. Error as a function of model partiality.

Figure 9 compares our results to results of state-of-the-art algorithms for *SHREC'16B*. Our method is competitive with that of [40], despite of the fact that our algorithm heavily depends on geodesic distances, and topological noise e.g., connecting the hand to the face) shortens these distances.

Drawbacks: xxx

AT: fifth paragraph: drawbacks with examples NA: Our method exhibits a few drawbacks. First and foremost the algorithm has a high asymptotic runtime -

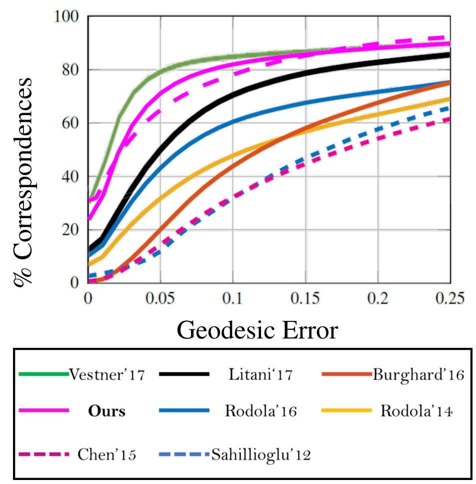

Figure 9. **Cumulative error curve on SHREC'16B. AT: same changes to the figure**

$O(n^2 logn) + O(|S|n^2)$(App. 5). This limits the utilization of our method to relatively small shapes with less than 15K vertices, which takes 200s. The algorithm also has six parameters which need to be tuned. In addition,the algorithm has some typical failure cases that can be seen in Figure 10: (1)Surface patches such as the dog's tail that are low in distinct shapes are hard to match. (2)Topological noise due to the intersection of different model parts changes the geodesic paths of the deformed model w.r.t to the base model. (3) Extreme deformations can cause shape descriptors of the deformed model to be vastly different than those of the base model.

Implementation details: xxx **AT: sixth paragraph: implementation details, how do we move from sparse to dense NA: Our code which produces the sparse correspondences is implemented entirely in C++ . We have used the Fast Marching Method[14] to obtain geodesic distances. FPFH shape descriptors are calculated using the Point Cloud library[29]. The Nearest Neighbor field is computed with FLANN[21] with χ^2 distance. The entire code is parallelized using OpenMP. Since computing similarity for a given patch is totally independent from its calculation for other patches, the obtained speedup is nearly linear in the number of threads.**

NA: To move from sparse to dense correspondence we employ the method of [18]. We had replaced the input dense descriptor field with localized smooth delta functions around our corresponding pairs. We have found that satisfying results are already achieved after 1 iteration on SHREC16'A, and 2 on SHREC'16B. We have tuned the parameters of [18] on 15 models of cats from the SHREC16'A training set.

a b c

Figure 10. NA: Typical failure cases of our method. (a) Strong topological noise affects geodesic distances, causing the right hand to be mapped to the leg. (b) Narrow surfaces which contain a low number of points are harder to match. (c)Extreme deformations make shape descriptors unreliable.

Alternatives: AT: sixth paragraph: alternatives—comparison to Lihi's function NA: We have used the training set provided with SHREC'16A to experiment with different variations and parameters of our method, which will be described hereafter.

NA: *Similarity formulation* We have tested our 3DDIS formulation against the formulation of [37]. The main difference between the two is as follows: The similarity metric of[37], expects the NNF to be nearly bijective. Thus, it penalizes the similarity score of a point in P being a nearest neighbor of multiple points in Q. The comparison, illustrated in Figure 11 shows a significant improvement over the original formulation. stemming from a higher robustness to partiality. Figure 11b implies that 3DDIS handles high model partiality better.

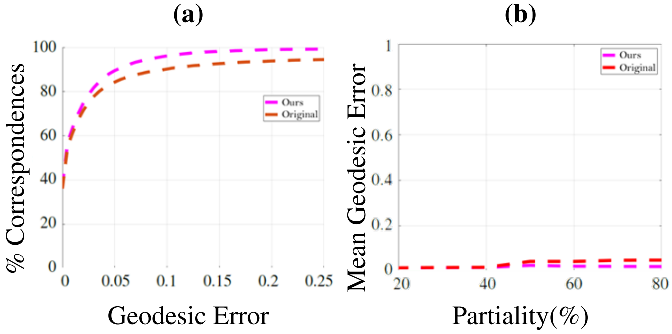

Figure 11. NA: A comparison between Our formulation and the original DDIS. AT: same changes to the figure

Robustness: AT: parameter tuning NA: *Surface patch radius* We have explored the behaviour of 3DDIS given different values of the surface patch radius R_T. The effects of different choices are illustrated in Figure 12. It can be seen that in general, a smaller surface radius leads to results which are better locally, as evident from the higher percentage of matches below 0.1% error. This, however, comes at the cost of some globally inconsistent matches, which cause this precentage to drop the farther along the curve we go. The application of our multiscale framework results in an overall better result than any single scale alone.

NA: *Coherency based refinement* The effect of our coherency based refinement has been tested on SHREC16'A training set. A comparison is shown in Figure ??. A 1.5% improvement in performace can be seen

NA: *Feature radius* We have also

Applications: xxx AT: seventh paragraph: uses in archaeology NA: We have also used 3DDIS to detect repeating patterns in archaeological artifact. Here we first found the geodesic center of the desired pattern in the scene, and extracted the geodesic disc around it. Then we employed our DDIS pipeline between our part and the geodesic disc. TBA -figures

References

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4pointss congruent sets for robust pairwise surface registration. *ACM Trans. Graph.*, 27(3):85:1–85:10, Aug. 2008. 2

[2] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011. 2

[3] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 6

[4] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016. 2

[5] A. M. Bronstein and M. M. Bronstein. Not only size matters: regularized partial matching of nonrigid shapes. In *Computer Vision and Pattern Recognition Workshops, 2008.*

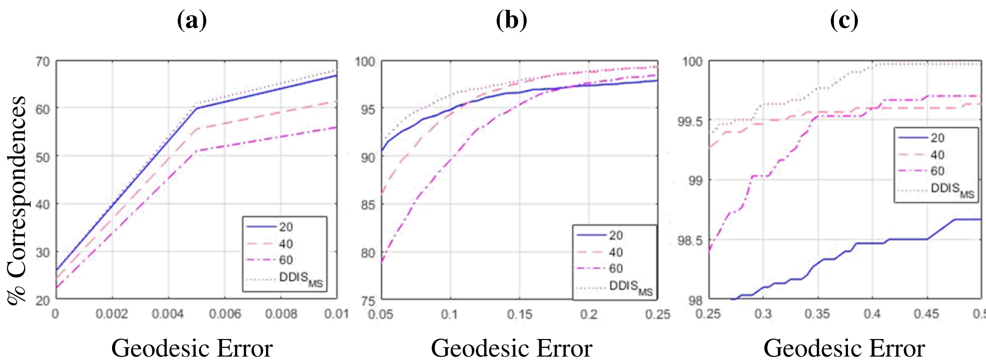

Figure 12. **NA: A comparison of different piece radii. (a) We can see that a radius of 20% $\sqrt{area(\mathcal{M})}$ has the most correspondences with an error below the threshold 0.01%.(b) shows a 40% radius overtakes it near the 0.1% mark.**

CVPRW'08. IEEE Computer Society Conference on, pages 1–6. IEEE, 2008. 1, 2

[6] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Partial similarity of objects, or how to compare a centaur to a horse. *International Journal of Computer Vision*, 84(2):163, 2009. 2

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006. 1, 2

[8] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1704–1711. IEEE, 2010. 2

[9] L. Cosmo, E. Rodolà, M. Bronstein, A. Torsello, D. Cremers, and Y. Sahillioglu. Shrec16: Partial matching of deformable shapes. *Proc. 3DOR*, 2, 2016. 1, 6

[10] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman. Best-buddies similarity for robust template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2029, 2015. 3

[11] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robotics & Automation Magazine*, 22(4):110–124, 2015. 1

[12] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005. 4

[13] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. 7

[14] R. Kimmel. Fast marching methods for computing distance maps and shortest paths. 1996. 8

[15] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 905–914, 2015. 2

[16] Z. Lähner, E. Rodolà, M. Bronstein, D. Cremers, O. Burghard, L. Cosmo, A. Dieckmann, R. Klein, and Y. Sahillioglu. Shrec16: Matching of deformable shapes with topological noise. *Proc. 3DOR*, 2:11, 2016. 6

[17] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)*, 28(3):72, 2009. 2

[18] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*, volume 36, pages 247–258. Wiley Online Library, 2017. 1, 2, 6, 7, 8

[19] J. Masci, E. Rodolà, D. Boscaini, M. M. Bronstein, and H. Li. Geometric deep learning. In *SIGGRAPH ASIA 2016 Courses*, page 1. ACM, 2016. 2

[20] F. Monti, D. Boscaini, and J. Masci. Geometric deep learning on graphs and manifolds using mixture model cnns. 2

[21] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009. 8

[22] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4):30:1–30:11, July 2012. 1, 2

[23] J. Pokrass, A. M. Bronstein, and M. M. Bronstein. Partial shape matching without point-wise correspondence. *Numerical Mathematics: Theory, Methods and Applications*, 6(1):223–244, 2013. 2

[24] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum*, volume 32, pages 459–468. Wiley Online Library, 2013. 2

[25] E. Rodolà, S. R. Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 4177–4184, Washington, DC, USA, 2014. IEEE Computer Society. 2

[26] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017. 1, 2, 6, 7

[27] E. Rodola, A. Torsello, T. Harada, Y. Kuniyoshi, and D. Cremers. Elastic net constraints for shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1169–1176, 2013. 2

[28] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. Citeseer, 2009. 1, 2, 3

[29] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. 8

[30] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Learning informative point classes for the acquisition of object model maps. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 643–650. IEEE, 2008. 2

[31] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008. 2, 3

[32] Y. Sahillioğlu and Y. Yemez. Minimum-distortion isometric shape correspondence using em algorithm. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2203–2215, 2012. 2

[33] Y. Sahillioğlu and Y. Yemez. Scale normalization for isometric shape matching. In *Computer Graphics Forum*, volume 31, pages 2233–2240. Wiley Online Library, 2012. 2

[34] Y. Sahilliolu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1461–1470. Wiley Online Library, 2011. 2

[35] A. Shtern and R. Kimmel. Matching the lbo eigenspace of non-rigid shapes via high order statistics. *Axioms*, 3(3):300–319, 2014. 2

[36] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1383–1392, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association. 3

[37] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template matching with deformable diversity similarity. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1311–1319, 2017. 1, 3, 9

[38] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. 2, 3

[39] A. Torsello. A game-theoretic approach to deformable shape matching. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 182–189, Washington, DC, USA, 2012. IEEE Computer Society. 1, 2

[40] M. Vestner, Z. Lähner, A. Boyarski, O. Litany, R. Slossberg, T. Remez, E. Rodola, A. Bronstein, M. Bronstein, R. Kimmel, et al. Efficient deformable shape correspondence via kernel matching. In *3D Vision (3DV), 2017 International Conference on*, pages 517–526. IEEE, 2017. 1, 2, 8

Appendices

Some Appendix The contents...

Appendix A: Runtime and complexity **NA: We calculate asymptotic runtime complexity using the following valid assumptions: (a)$|S| << |\mathcal{M}|$ - this is satisfied since we only compute sparse correspondences typically ¡300 for a full model. (b) $|\mathcal{N}| \approx |\mathcal{M}| = n$. The complexity of different algorithm stages is thus:**

1. **NA: *FPFH Calculation* Calculation of FPFH takes $O(n{\cdot}k)$ where k is the number of neighbors for each point in a defined spherical neighborhood of a radius R_F. In our case $R_F = 0.03\sqrt{area(\mathcal{M})}$ which implies $k << n$.**

2. **NA: *NN search* Approximate nearest neighbor are calculated by building a kd-tree for $\mathcal{N}$ which takes $O(dnlogn)$, where d is the feature dimension, In the case of FPFH, $d = 33$ and is thus treated as a constant. We perform a NN search for each vertex of $\mathcal{M}$, each taking $O(logn)$ operations. Thus the overall complexity of this stage is $O(nlogn)$.**

3. **NA: *Geodesics* Calculating fast marching geodesics distances from a single point to all other points on a mesh has a runtime complexity of $O(nlogn)$. Since we repeat this process for each point on both meshes the total complexity of this stage becomes $O(n^2 logn)$**

4. **NA: *Similarity map calculation* Calculation of similarity between a pair of patches P, Q, requires a single pass over all the points in both. Thus, requires $O(|P| + |Q|) \approx O(n)$ operations. We compute similarity scores for all possible patches in $\mathcal{M}$ to all sample patches in S. Overall we make $O(|S|n)$ similarity calculations. Thus the runtime complexity of this stage has an upper bound of $O(|S|n^2)$ operations.**

5. **NA: *Correspondence Refinement* Detecting outliers is done in $O(|S|^2)$ - we sum the geodesic distance differences of each sample point to all other points on its respective model. We set the numbers of alternative 3DDIS maximas to 100 for each point. This can be done in a linear time in $|S|$ for each sample, thus the maxima detection also requires $O(|S|^2)$ operations. Overall the stage has a runtime complexity of $O(|S|^2)$.**

NA: In total, the algorithm has an effective runtime complexity of $O(|S|n^2)+O(n^2 logn)$. As previously mentioned, this limits us to running on models of 15,000k

vertices. Mesh simplification algorithms can be applied for downsampling if running on larger models is required.