

000
001
002
003
004
005
006
007
008
009
010
011

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 minimiza-

tion 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 ex-

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
isting 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 stemming 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 non-rigid 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 diagnoilization. The main drawback of this method, shared with other intrinsic methods, is its invariance to symmetries.

3D Shape Descriptors

216 **2.3. Template matching in 2D**
217

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

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

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

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

$$247 \quad DIS = c \cdot |\{n_i \in \mathcal{N} : \exists m_j \in \mathcal{M}, NN(m_j, \mathcal{N}) = n_i\}| \quad (1)$$

248 where $|\cdot|$ denotes group size and $c = 1/\min(|\mathcal{M}|, |\mathcal{N}|)$ is
 249 a normalization factor. Between non corresponding win-
 250 dows, indeed one should expect most points to have no real
 251 corresponding point, and thus be mapped to a very and re-
 252 mote nearest neighbors. On the other hand, regions contain-
 253 ing matching objects are drawn from the same distribution,
 254 thus the diversity of nearest neighbors should be high. To
 255 accommodate this assumption not only did they reward
 256 high diversity of nearest neighbors, but also penalized map-
 257 ping to the same patch. To this end, another, a negative
 258 diversity measure had been defined:

$$259 \quad \kappa_{\mathcal{M}}(n_i) = |\{m \in \mathcal{M} : NN^a(m, \mathcal{N}) = n_i\}| \quad (2)$$

260 With x_i^a denoting the appearance descriptor of point x_i .
 261 Thus the contribution of a patch $m_j : NN^a(m_j, \mathcal{N}) = n_i$
 262 is $\exp(1 - \kappa_{\mathcal{M}}(n_i))$. An additional observation made has
 263 been that while non isometric deformations do occur, they
 264 should be restricted, small, in real objects. With distance
 265 on the window pixel grid between 2 nearest neighbor points

266 defined as $r_j = d(m_j^l, n_i^l)$ with x_i^l denoting the location
 267 of x_i on a grid, the final Deformable Diversity Similarity
 268 formulation becomes:

$$269 \quad DDIS = c \sum_{\mathcal{N} \rightarrow \mathcal{M}} \frac{1}{1 + r_j} \cdot \exp(1 - \kappa(NN^a(m_j, \mathcal{N}))) \quad (3)$$

270 **3. General Approach**
271

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

277 First, inspired by [37], similarity is captured by two
 278 properties of the Nearest Neighbor field. (1) When \mathcal{N} and a
 279 patch of \mathcal{M} match, most points in \mathcal{M} have a unique NN-
 280 match in \mathcal{N} . This implies that the NN field should be
 281 highly diverse, in the sense that many different points in
 282 \mathcal{N} are being matched. (2) Arbitrary matches typically imply
 283 a large deformation, whereas correct matches should pre-
 284 serve the distance between pair of points. Therefore, Simi-
 285 larity should be based both on the diversity of the Nearest-
 286 Neighbor field and on the consistency of the distances be-
 287 tween the points.

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

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

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

- 303 **1. Pre-processing.** Shape descriptors are calculated for
 304 every vertex of both meshes and an approximate near-
 305 est neighbor field is computed for the vertices, as de-
 306 scribed hereafter.

307 Many descriptors have been proposed in the litera-
 308 ture [31, 38, 36]. We use the FPFH [28], which is ro-
 309 bust to small deformations and partiality of the data,
 310 yet sensitive to symmetrical flips. **AT: what makes**
 311 **it sensitive to symmetrical flips?** Therefore, it ad-
 312 dresses a major drawback of matching a right arm, for
 313 example, to the left one. We then compute a nearest
 314 neighbor field mapping, by assigning each vertex of
 315 \mathcal{M} its nearest neighbor in \mathcal{N} , FPFH-wise.

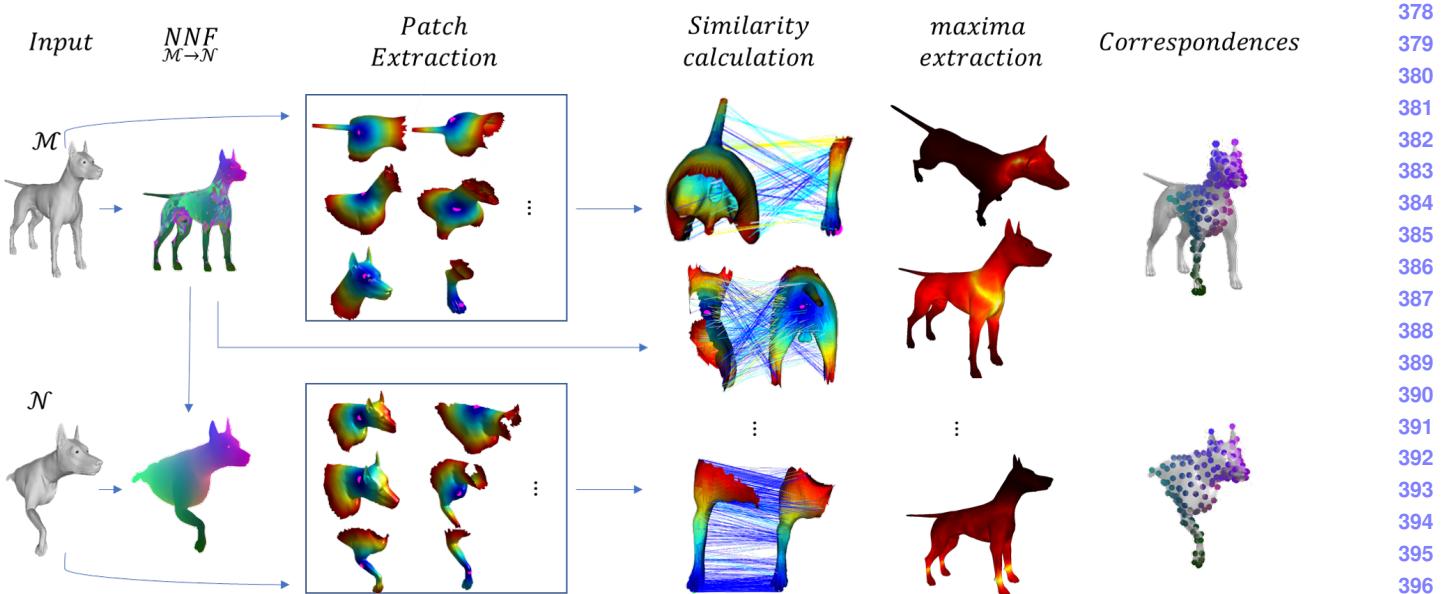


Figure 1. **Algorithm outline.** In the first step, we compute the nearest-neighbor field for \mathcal{N} and \mathcal{N} . Then, patches of the surfaces are extracted for every sample point in \mathcal{N} and for every vertex on \mathcal{M} . These patches cover the surface (i.e., may overlap) and represent 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

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

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.

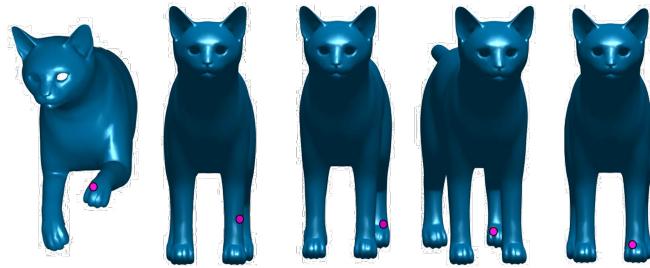
For each pair of patches of the same scale, $Q_i \subset \mathcal{M}$ and $P_i \subset \mathcal{N}$, we compute a similarity value. Recall that our goal is to reward a nearest-neighbor field with high diversity and low deformation. We will define the similarity function $DDIS$ that achieves it in Section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

432 found to have the highest similarity in a coarsest scale
 433 (i.e. Q_j is the largest). The coarsest match is then
 434 set between $v_i \in P_i$ and $w_j \in Q_j$, where v_i, w_j are
 435 the *geodesic centers* of P_i, Q_j , respectively (i.e. the
 436 sample points that define the patches in Step 2). When
 437 moving to the finer scale, we replace Q_j with a smaller
 438 (finer) patch in which w_j is the center and set the new
 439 w_j to be the vertex on this patch that maximizes the
 440 similarity function $DDIS$ on this patch. The finest w_j
 441 is the corresponding point of v_i ; see Figure 2.
 442



443
 444 **Figure 2. Multi-scale similarity.** Given the sample point on the
 445 left (in magenta), the corresponding points on \mathcal{M} are shown on
 446 the right. In the coarsest level, the general region of the matching
 447 point is found, but the point is imprecise. In subsequent levels, the
 448 general region is not found. Our multi-scale approach manages to
 449 find the precise point. **AT: Replace the image. NA: done**

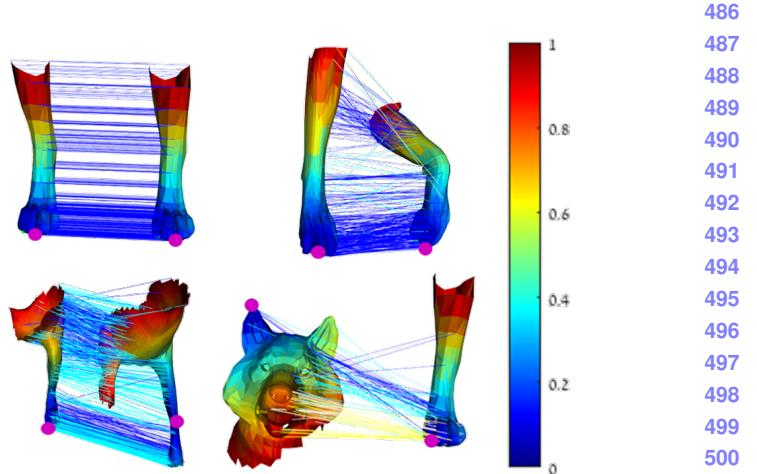
450
 451
 452
 453
 454
 455
 456
 457
 458
 459
5. Coherency-based correspondence refinement The result of Step 4 is a set of corresponding pairs of points. In most cases ($> 92\%$ on all our examples), the correspondences are correct. The goal of this step is to identify the incorrect ones and replace them by the correct correspondences.

460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 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.



486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
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 the bottom, where the surfaces are highly different from one another.

4. Similarity

514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 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.

536
 537
 538
 539
 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?**

536
 537
 538
 539
 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 mea-

540 sure diversity is to count the number of unique NNs between
 541 the points of Q and P :

$$543 \quad Div(Q, P) = |\{p_i \in P : \exists q_j \in Q, NN(q_j, P) = p_i\}|, \quad (6)$$

544 where $\{p_i\}_{i=1}^{|P|}$ and $\{q_j\}_{j=1}^{|Q|}$ are the set of points of Q and P ,
 545 respectively and the nearest neighbors is computed between
 546 the descriptors (FPFH) of the points. However, we will see
 547 below that the diversity can be calculated implicitly.
 548

549 Next, we define $Def(Q, P)$, the deformation from Q to
 550 P . Let $p \in P$ and $q \in Q$ be the centers of P and Q , respec-
 551 tively; furthermore, let $p_i = NN(q_j, P)$ be the nearest-
 552 neighbor of $q_j \in Q$. The deformation implied by the NN-
 553 Field for p_i, q_j is defined by:
 554

$$555 \quad def(q_j, p_i, Q, P) = |GeodDist(q_j, q) - GeodDist(p_i, p)|\epsilon, \quad (7)$$

556 where $0 < \epsilon \ll Area(\mathcal{M})$ and is used for numerical
 557 stability.
 558

559 For each $p_i \in P$, we find the minimal deformation
 560 $r_i^* = \min_{q_j \in Q} def(q_j, p_i, Q, P)$ such that p_i is the nearest
 561 neighbor of q_j . Note that some points in P might not be as-
 562 sociated with any point in Q (since they are not the nearest
 563 neighbors of any point $q_j \in Q$); in this case $r_i^* = \infty$.
 564

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

$$568 \quad DDIS(P, Q) = \sum_{p_i \in P} \frac{1}{1 + r_i^*}. \quad (8)$$

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

5. Results

587 We have evaluated our method both qualitatively and
 588 quantitatively on several datasets: (1) the benchmark of
 589 *SHREC’16A—partial matching of deformable shapes* [9];
 590 (2) the even more challenging benchmark of *SHREC’16B—*
 591 *matching of deformable shapes with topological noise* [16].
 592 **AT: (3) Faust, (4) Archaeology NA: (3) FAUST[3] in-**
 593 **cludes real scans of human subjects. For this set we**

594 **provide only qualitative evaluation. NA: (4) Archeologi-**
 595 **cal artifacts[3] which include different archeological ar-**
 596 **tifacts.** In all cases, our method either outperformed the
 597 state-of-the-art methods or was competitive.

598 *SHREC’16A* contains 400 partial shapes, each is a near-
 599 isometrically deformed version of one of eight base model,
 600 given in a neutral pose. The dataset is further divided into
 601 two subsets, according to the type of partiality: (1) *cuts*,
 602 which is composed of shapes produced by dividing shapes
 603 by a plane, and (2) *holes*, obtained by eroding many areas
 604 around random vertices. *SHREC’16B* is contains 10 shapes,
 605 which are derived from the same base human shape that
 606 undergoes deformations and topological changes stemming
 607 from self-intersections.

608 **NA: FAUST contains 200 real world scans of 10 dif-**
 609 **ferent human subjects. The acquisition process of which**
 610 **introduces topological artifacts and missing parts due to**
 611 **occlusions.**

612 **AT: paragraph: explain that there are two distinct,**
 613 **yer related challenges: sparse and dense NA: Surface**
 614 **correspondence algorithms can divided into 2 by the**
 615 **density of their correspondences. (1) Sparse correspon-**
 616 **dences which aim to cover the surface area uniformly,**
 617 **but sparsely. (2) Dense correspondences which match**
 618 **every vertex on one shape to the other shape. AT: para-**
 619 **graph: explain that you compute only sparse and then**
 620 **use... to convert it into dense. NA: Our method pro-**
 621 **duces sparse correspondences, yet these can be con-**
 622 **verted to dense ones with a minimal loss of quality. This**
 623 **is achieved by using the sparse correspondences as input**
 624 **to the method of [18].**

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

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

641 Figure 5 further demonstrates it, by color-coding the er-
 642 rors. The larger the error, the more reddish the color is. **AT:**
643 add this figure Figure 6 shows a couple of examples from
 644 the *SHREC’16B* dataset. On the topological noise bench-
 645 mark we can see bad matches are typically limited to the
 646

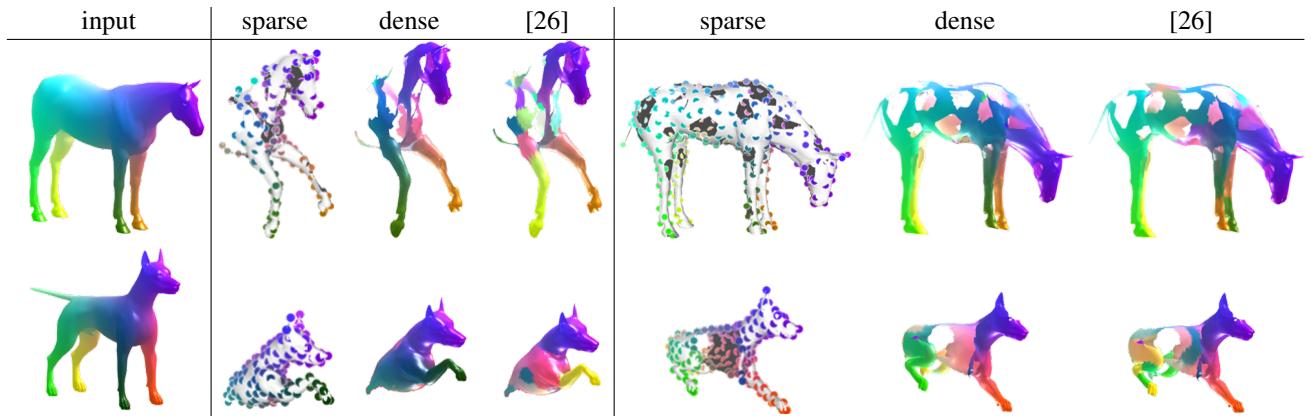


Figure 4. Results on Shrec'16A. Our results outperform those of [26], when run with the default parameters. **AT:** (1) create little images (2) remove boundaries (3) change the model color in sparse correspondences (4) the models must be in the same size

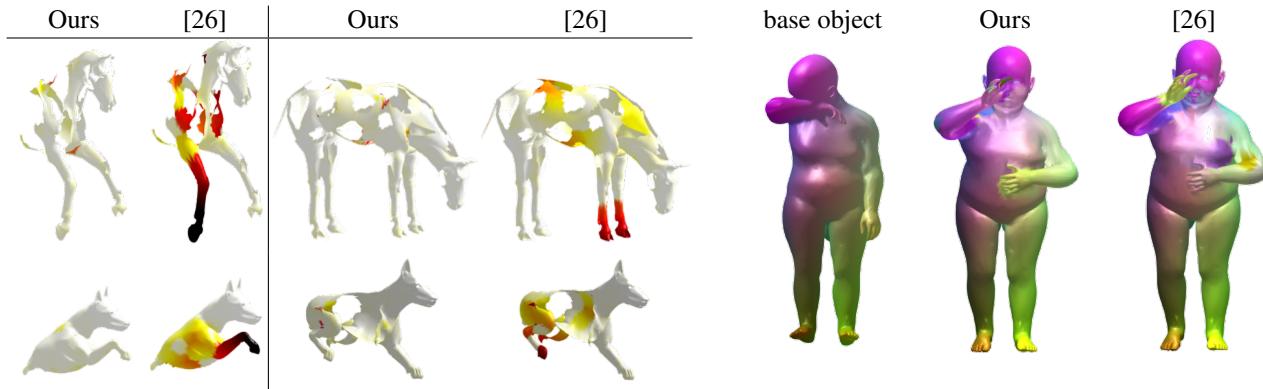


Figure 5. Errors on Shrec'16A. Our results outperform those of [26]

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

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} :

$$\varepsilon(q) = \frac{\text{GeoDist}_{\mathcal{M}}(p, p^*)}{\sqrt{\text{area}(\mathcal{M})}} \quad (9)$$

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



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

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.

AT: Add a paragraph describing Figure 8 NA: Figure 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 $\text{area}(\mathcal{N})/\text{area}(\mathcal{M})$, hereafter referred to as partial-

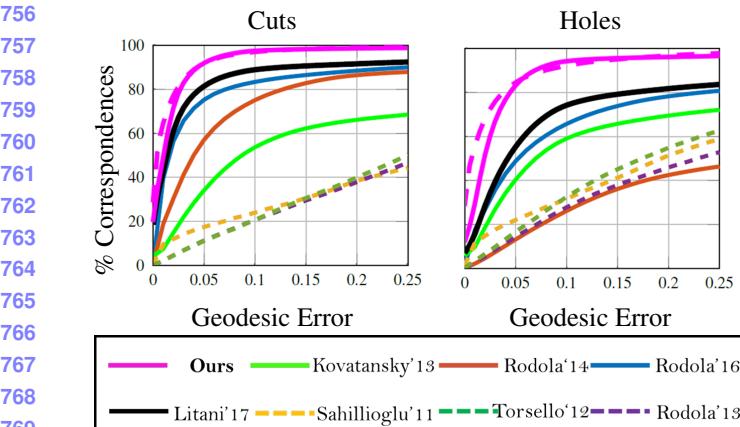


Figure 7. **Cumulative geodesic error curves on Shrec16A.** Our method (in magenta) outperforms other algorithms, both for the dense correspondence and for the sparse correspondence, on the two subsets of the dataset. **AT: 1. three images, (2) [Author'19] or Ours**

ity. It can be seen that our method is weakly dependent on the partiality of the model.

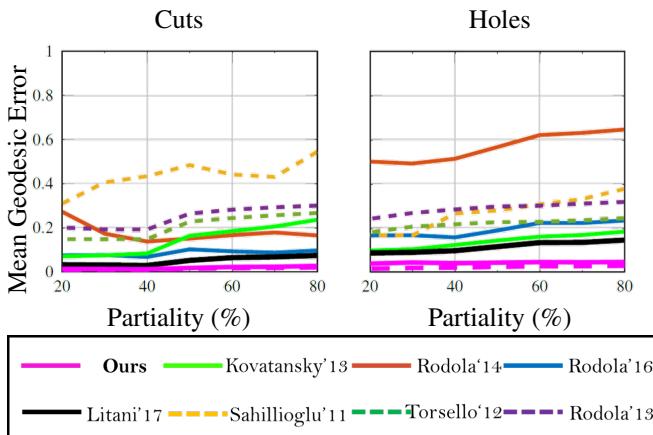


Figure 8. Error as a function of model partiality.

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 - $O(n^2 \log n) + O(|S|n^2)$ (App. 5). This limits the utilization of our method to relatively small shapes with less than 15K vertices. The algorithm also has six parameters which need to be tuned. In addition, the algorithm

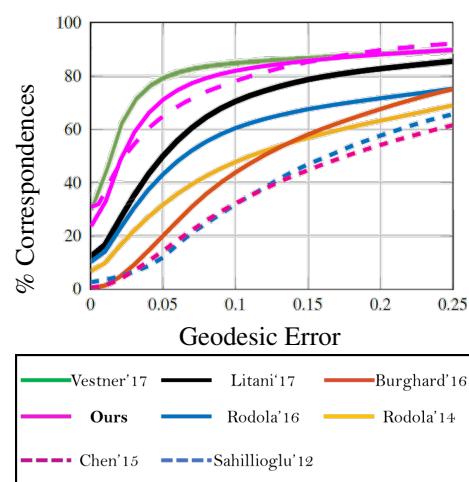


Figure 9. **Cumulative error curve on SHREC'16B. AT: same changes to the figure**

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.

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,



Figure 10. **NA:** Some notable failure cases. Correspondence ‘leakage’ and misassignments happen near topological noise. Narrow parts with few matches are hard to match(dog’s tail). Severe isometry and shape altering deformations on occasion still produces a match which is completely incorrect.

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], penalizes the sharing of a NN in P by multiple points in Q . Points that share a NN with κ other points will have their contribution to the similarity score attenuated by $e^{(1-\kappa)}$. 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.

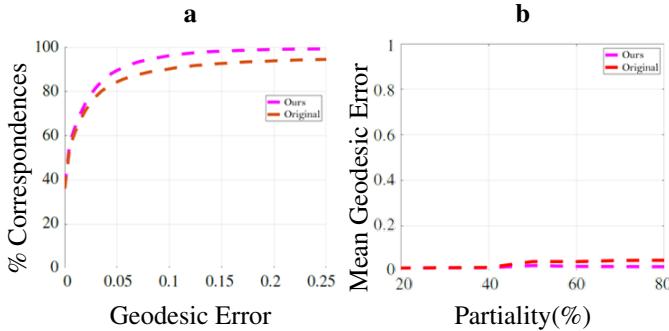


Figure 11. A comparison between Our formulation and the original DDIS. **AT:** same changes to the figure

Robustness: **AT:** parameter tuning **NA:** *Surface patch radius* We have experimented with various patch radii. 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 percentage 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.

Feature radius

Applications: xxx **AT:** seventh paragraph: uses in archaeology

Appendix A: Runtime and complexity **NA:** We calculate asymptotic runtime complexity using the following valid assumptions: (a) $|S| \ll |\mathcal{M}|$ - 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:

- 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 $R_F = 0.03\sqrt{\text{area}(\mathcal{M})}$ which implies $k \ll n$.
- NA: NN search** Approximate nearest neighbor are calculated by building a kd-tree for \mathcal{N} which takes $O(dn\log n)$, 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 $O(\log n)$ operations. Thus the overall complexity of this stage is $O(n\log n)$.
- NA: Geodesics** Calculating fast marching geodesics distances from a single point to all other points on a mesh has a runtime complexity of $O(n\log n)$. Since we repeat this process for each point on both meshes the total complexity of this stage becomes $O(n^2\log n)$
- 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.

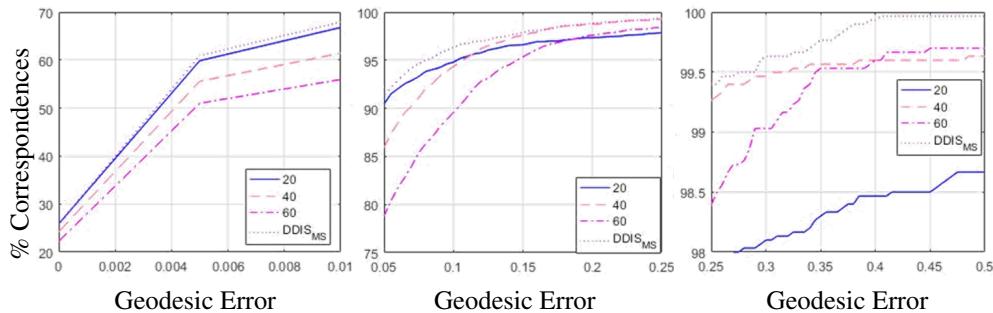
972
973
974
975
976
977
978
979
980
981
982
983
984

Figure 12. A comparison of different piece radii. AT: same changes to the figure

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 $O(|S|n^2) + O(n^2 \log n)$. 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.

5.1. Experimental Results

In this section we will briefly go over the available Datasets **SHREC 2016 Partial Correspondence** The SHREC partial matching dataset[9] consists of 8 base, neutral pose models: cat, centaur, dog, horse, wolf, and 3 humans – 2 males, and 1 female, each containing 10,000 vertices except for the wolf which contains only 4,500 vertices. Each basic model has corresponding deformed partial shapes obtained either by cutting the shape with a plane or by adding holes on a deformed shape. The set has been divided into train and test sets. The train set is composed of 15 cuts for each base models totaling 120 models, and 10 holed shapes for each model for which ground truth point to polygon correspondences has been provided in barycentric coordinates. The test set is composed of additional 200 cuts and 200 holed shapes. We have tuned our parameters only on the 120 pairs of cuts.

5.2. Error Metrics

The output of partial matching algorithms (as defined in[9]) are sub-vertex point-to-point correspondences between partial shapes. For all experiments we use the standard practice of not penalizing symmetric solutions. Quality is measured according to the Princeton benchmark pro-

tocol [13]. For a pair of points $(x, y) \in \mathcal{N} \times \mathcal{M}$ between the full object \mathcal{M} and the partial shape \mathcal{N} produced by an algorithm, where (x, y^*) is the ground truth correspondence the inaccuracy is measured by. We plot curves showing the fraction of correspondences whose errors are below a threshold of the normalized geodesic error

$$\varepsilon(x) = \frac{d_{\mathcal{M}}(y, y^*)}{\sqrt{\text{area}(\mathcal{M})}} \quad (10)$$

where $d_{\mathcal{M}}(y, y^*)$ is the geodesic distance on \mathcal{M} , and has units of normalized length on \mathcal{M} . For dense correspondences over a dataset, $\varepsilon(x)$ is averaged over all matching instances.

5.3. Sparse Correspondences on the SHREC16 Test set

We have tested the performance of DDIS in producing sparse correspondences on the SHREC16 Partial Matching of Deformable Shapes competition. We had tuned our parameters on the SHREC16 training dataset using only the cuts part of it. The best results had been produced using FPFH with $r_F = 0.03\sqrt{\text{Area}(\mathcal{M})}$, and a piece size radii of $R_T = [0.6, 0.4, 0.2]\sqrt{\text{Area}(\mathcal{M})}$. For Geodesic distances we have found the fast marching algorithm to work the fastest, while giving the lowest error w.r.t. to exact geodesics. For a 10,000 vertices mesh it takes 60s to produce a full distance matrix, Though it should be noted this algorithm has a more efficient GPU implementation, and parallelization on a core brought the run time to 12s with 6 threads. FPFH and Nearest Neighbor field takes 2s, and similarity between 2 pieces of 10000 vertices each takes 120s on average, running on 6 threads of i7-4790. Unlike optimization based algorithms this is highly parallelizable. We achieve results comparable to the state of the art [18] quality wise, even though sparser in nature on both the Cuts and the Holes datasets, Where a particularly impressive result is reported on the Holes dataset, which can then be expanded with a minimal loss of quality by feeding these to the FSPM[18] as input instead of low level shape descriptors.

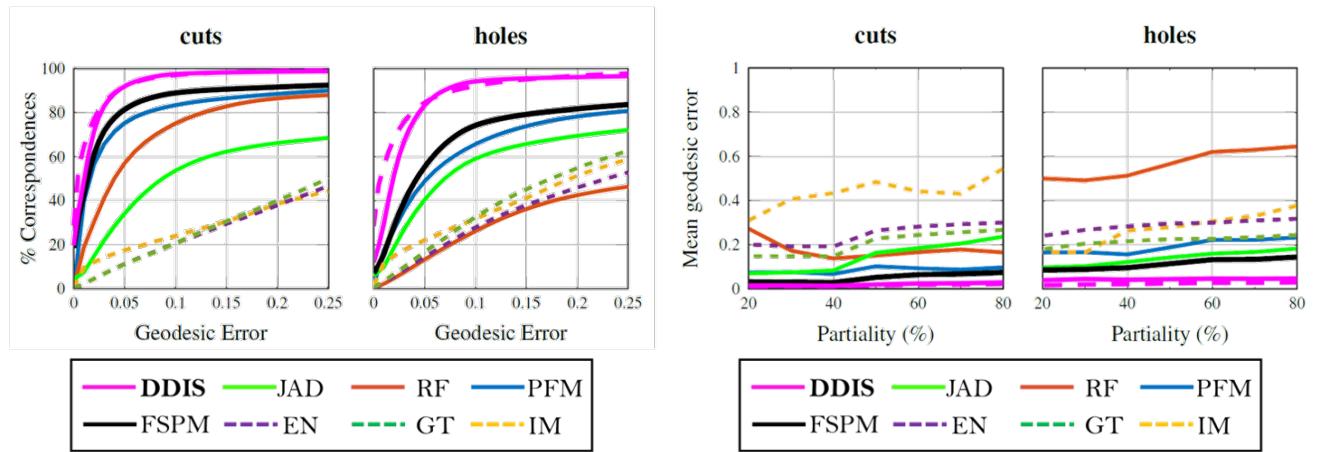


Figure 13. comparison with other state of the art algorithms - it can be seen that although sparse in nature, the correspondence obtained by DDIS are much more accurate than the other methods. A separate analysis has been done for correspondences which include boundary points, which tend to be more noisy, and internal points which are more sparse

[htb]

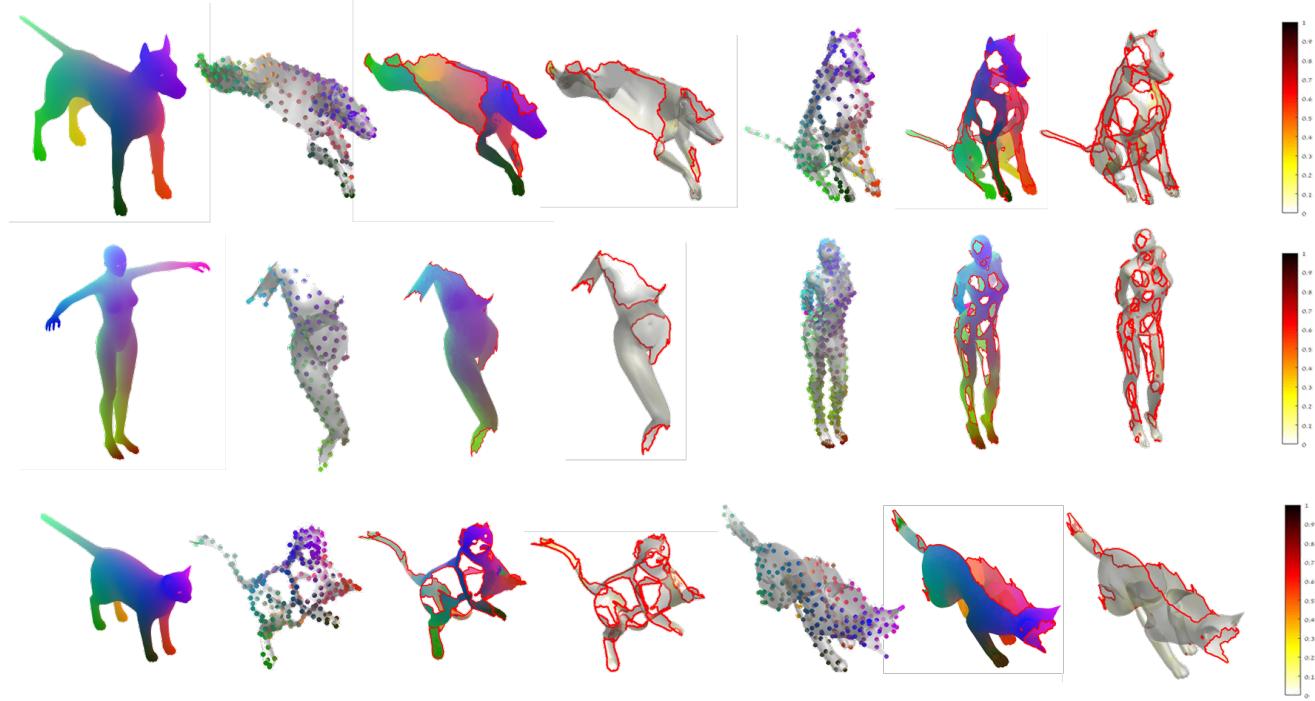


Figure 14. Good correspondences obtained by our method

	PFM	RF	IM	EN	GT	DDIS
cuts	dense	dense	61.3	87.8	51.0	132.2
holes	dense	dense	78.2	112.6	76.4	77.3

Table 1. mean number of correspondence obtained by the algorithms in the SHREC 16 competition and our algorithm. Note that our algorithm can combine wi

implementation Our code which produces the sparse correspondences is implemented entirely in C++ using the Point Cloud Library. For geodesics we have used Fast

Marching Geodesics adapted from the code published by Ron Kimmel to run in parallel on multiple cores. We calculate FPFH using $R_{FPFH} = 0.03 \cdot \sqrt{Area(\mathcal{M})}$, and NNF using FLANN. The average runtime on i7-4790 for 2 surfaces of 12000 vertices is 150s.

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

- 1188 [2] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel
1189 signature: A quantum mechanical approach to shape analy-
1190 sis. In *Computer Vision Workshops (ICCV Workshops), 2011
1191 IEEE International Conference on*, pages 1626–1633. IEEE,
1192 2011. 2
- 1193 [3] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust:
1194 Dataset and evaluation for 3d mesh registration. In *Proceed-
1195 ings of the IEEE Conference on Computer Vision and Pattern
1196 Recognition*, pages 3794–3801, 2014. 6
- 1197 [4] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learn-
1198 ing shape correspondence with anisotropic convolutional
1199 neural networks. In *Advances in Neural Information Pro-
1200 cessing Systems*, pages 3189–3197, 2016. 2
- 1201 [5] A. M. Bronstein and M. M. Bronstein. Not only size mat-
1202 ters: regularized partial matching of nonrigid shapes. In
1203 *Computer Vision and Pattern Recognition Workshops, 2008.
1204 CVPRW’08. IEEE Computer Society Conference on*, pages
1205 1–6. IEEE, 2008. 1, 2
- 1206 [6] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and
1207 R. Kimmel. Partial similarity of objects, or how to compare a
1208 centaur to a horse. *International Journal of Computer Vision*,
1209 84(2):163, 2009. 2
- 1210 [7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Gener-
1211 alized multidimensional scaling: a framework for isometry-
1212 invariant partial surface matching. *Proceedings of the National
1213 Academy of Sciences*, 103(5):1168–1172, 2006. 1, 2
- 1214 [8] M. M. Bronstein and I. Kokkinos. Scale-invariant heat ker-
1215 nel signatures for non-rigid shape recognition. In *Computer
1216 Vision and Pattern Recognition (CVPR), 2010 IEEE Confer-
1217 ence on*, pages 1704–1711. IEEE, 2010. 2
- 1218 [9] L. Cosmo, E. Rodolà, M. Bronstein, A. Torsello, D. Cremers,
1219 and Y. Sahillioglu. Shrec’16: Partial matching of deformable
1220 shapes. *Proc. 3DOR*, 2, 2016. 1, 6, 10
- 1221 [10] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Free-
1222 man. Best-buddies similarity for robust template matching.
In *Proceedings of the IEEE Conference on Computer Vision
1223 and Pattern Recognition*, pages 2021–2029, 2015. 3
- 1224 [11] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke.
1225 Registration with the point cloud library: A modular frame-
1226 work for aligning in 3-d. *IEEE Robotics & Automation Mag-
azine*, 22(4):110–124, 2015. 1
- 1227 [12] S. Katz, G. Leifman, and A. Tal. Mesh segmentation us-
1228 ing feature point and core extraction. *The Visual Computer*,
1229 21(8-10):649–658, 2005. 4
- 1230 [13] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic
1231 maps. In *ACM Transactions on Graphics (TOG)*, volume 30,
1232 page 79. ACM, 2011. 7, 10
- 1233 [14] R. Kimmel. Fast marching methods for computing distance
1234 maps and shortest paths. 1996. 8
- 1235 [15] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Van-
1236 derghenst. Functional correspondence by matrix comple-
1237 tion. In *Proceedings of the IEEE conference on computer
1238 vision and pattern recognition*, pages 905–914, 2015. 2
- 1239 [16] Z. Lähner, E. Rodolà, M. Bronstein, D. Cremers,
1240 O. Burghard, L. Cosmo, A. Dieckmann, R. Klein, and
Y. Sahillioglu. Shrec’16: Matching of deformable shapes
1241 with topological noise. *Proc. 3DOR*, 2:11, 2016. 6
- 1242 [17] Y. Lipman and T. Funkhouser. Möbius voting for surface
1243 correspondence. *ACM Transactions on Graphics (TOG)*,
1244 28(3):72, 2009. 2
- 1245 [18] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein.
1246 Fully spectral partial shape matching. In *Computer Graphics
1247 Forum*, volume 36, pages 247–258. Wiley Online Library,
2017. 1, 2, 6, 7, 8, 10
- 1248 [19] J. Masci, E. Rodolà, D. Boscaini, M. M. Bronstein, and
H. Li. Geometric deep learning. In *SIGGRAPH ASIA 2016
1249 Courses*, page 1. ACM, 2016. 2
- 1250 [20] F. Monti, D. Boscaini, and J. Masci. Geometric deep learning
1251 on graphs and manifolds using mixture model cnns. 2
- 1252 [21] M. Muja and D. G. Lowe. Fast approximate nearest neigh-
1253 bors with automatic algorithm configuration. *VISAPP (1)*,
1254 2(331-340):2, 2009. 8
- 1255 [22] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and
L. Guibas. Functional maps: A flexible representation of
1256 maps between shapes. *ACM Trans. Graph.*, 31(4):30:1–
1257 30:11, July 2012. 1, 2
- 1258 [23] J. Pokrass, A. M. Bronstein, and M. M. Bronstein. Partial
1259 shape matching without point-wise correspondence. *Nu-
1260 matical Mathematics: Theory, Methods and Applications*,
1261 6(1):223–244, 2013. 2
- 1262 [24] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprech-
1263 mann, and G. Sapiro. Sparse modeling of intrinsic cor-
1264 respondences. In *Computer Graphics Forum*, volume 32,
1265 pages 459–468. Wiley Online Library, 2013. 2
- 1266 [25] E. Rodolà, S. R. Bulò, T. Windheuser, M. Vestner, and
D. Cremers. Dense non-rigid shape correspondence using
1267 random forests. In *Proceedings of the 2014 IEEE Confer-
1268 ence on Computer Vision and Pattern Recognition, CVPR
1269 ’14*, pages 4177–4184, Washington, DC, USA, 2014. IEEE
1270 Computer Society. 2
- 1271 [26] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and
D. Cremers. Partial functional correspondence. In *Computer
1272 Graphics Forum*, volume 36, pages 222–236. Wiley Online
1273 Library, 2017. 1, 2, 6, 7
- 1274 [27] E. Rodola, A. Torsello, T. Harada, Y. Kuniyoshi, and D. Cre-
1275 mers. Elastic net constraints for shape matching. In *Pro-
1276 ceedings of the IEEE International Conference on Computer
1277 Vision*, pages 1169–1176, 2013. 2
- 1278 [28] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature
1279 histograms (fpfh) for 3d registration. In *Robotics and Au-
1280 tomation, 2009. ICRA’09. IEEE International Conference
1281 on*, pages 3212–3217. Citeseer, 2009. 1, 2, 3
- 1282 [29] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library
1283 (PCL). In *IEEE International Conference on Robotics and
1284 Automation (ICRA)*, Shanghai, China, May 9-13 2011. 8
- 1285 [30] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Learn-
1286 ing informative point classes for the acquisition of object
1287 model maps. In *Control, Automation, Robotics and Vi-
1288 sion, 2008. ICARCV 2008. 10th International Conference
1289 on*, pages 643–650. IEEE, 2008. 2
- 1290 [31] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and
M. Beetz. Towards 3d point cloud based object maps for
1291 household environments. *Robotics and Autonomous Sys-
1292 tems*, 56(11):927–941, 2008. 2, 3

- 1296 [32] Y. Sahillioğlu and Y. Yemez. Minimum-distortion iso- 1350
1297 metric shape correspondence using em algorithm. *IEEE 1351
1298 transactions on pattern analysis and machine intelligence*, 1352
1299 34(11):2203–2215, 2012. 2 1353
1300 [33] Y. Sahillioğlu and Y. Yemez. Scale normalization for iso- 1354
1301 metric shape matching. In *Computer Graphics Forum*, volume 1355
1302 31, pages 2233–2240. Wiley Online Library, 2012. 2 1356
1303 [34] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine combinatorial 1357
1304 matching for dense isometric shape correspondence. In *Com- 1358
1305 puter Graphics Forum*, volume 30, pages 1461–1470. Wiley 1359
1306 Online Library, 2011. 2 1360
1307 [35] A. Shtern and R. Kimmel. Matching the lbo eigenspace of 1361
1308 non-rigid shapes via high order statistics. *Axioms*, 3(3):300– 1362
1309 319, 2014. 2 1363
1310 [36] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and prov- 1364
1311 ably informative multi-scale signature based on heat diffusion. 1365
1312 In *Proceedings of the Symposium on Geometry Pro- 1366
1313 cessing*, SGP ’09, pages 1383–1392, Aire-la-Ville, Switzer- 1367
1314 land, Switzerland, 2009. Eurographics Association. 3 1368
1315 [37] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template match- 1369
1316 ing with deformable diversity similarity. In *Proc. of IEEE 1370
1317 Conf. on Computer Vision and Pattern Recognition*, pages 1371
1318 1311–1319, 2017. 1, 3, 9 1372
1319 [38] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of 1373
1320 histograms for local surface description. In *European con- 1374
1321 ference on computer vision*, pages 356–369. Springer, 2010. 2, 3 1375
1322 [39] A. Torsello. A game-theoretic approach to deformable shape 1376
1323 matching. In *Proceedings of the 2012 IEEE Conference on 1377
1324 Computer Vision and Pattern Recognition (CVPR)*, CVPR 1378
1325 ’12, pages 182–189, Washington, DC, USA, 2012. IEEE 1379
1326 Computer Society. 1, 2 1380
1327 [40] M. Vestner, Z. Lähner, A. Boyarski, O. Litany, R. Slossberg, 1381
1328 T. Remez, E. Rodola, A. Bronstein, M. Bronstein, R. Kim- 1382
1329 mel, et al. Efficient deformable shape correspondence via 1383
1330 kernel matching. In *3D Vision (3DV), 2017 International 1384
1331 Conference on*, pages 517–526. IEEE, 2017. 1, 2, 8 1385
1332 1386
1333 1387
1334 1388
1335 1389
1336 1390
1337 1391
1338 1392
1339 1393
1340 1394
1341 1395
1342 1396
1343 1397
1344 1398
1345 1399
1346 1400
1347 1401
1348 1402
1349 1403