

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[27, 10], 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[6, 38] and regularity of parts[?, 4]. 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[21]. 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[17, 39, 25], 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[36](**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[8]. 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[29], SHOT[37] and FPFH[27] 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[7] 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[31, 32, 33], 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[31, 32, 33] 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[6], an embedding into the mobius group[16], or by representation in the LBO basis[34]. A notable family of works are derived from functional correspondences. Introduced at[21, 23, 14, 39] 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[24] and deep learning architectures[18, 3, 19]. 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[30] 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[4, 5] formulated an energy minimization problem over metric distortion and regularity of corresponding parts. Following works relaxed the regularity requirement by allowing for sparse correspondences[38, 26]. Other works[32, 31] 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[22] 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 [25] 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[17] 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
217
218
219
220
221
222
223
224

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[9] 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 to 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 shown to be able to handle occlusions, missing parts and complex deformations of templates.

Deformable Diversity Similarity Building upon the above work, [36] 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:

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

where $|\cdot|$ 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:

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

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 [36], 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 than 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 [30, 37, 35]. We use the FPFH [27], 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.

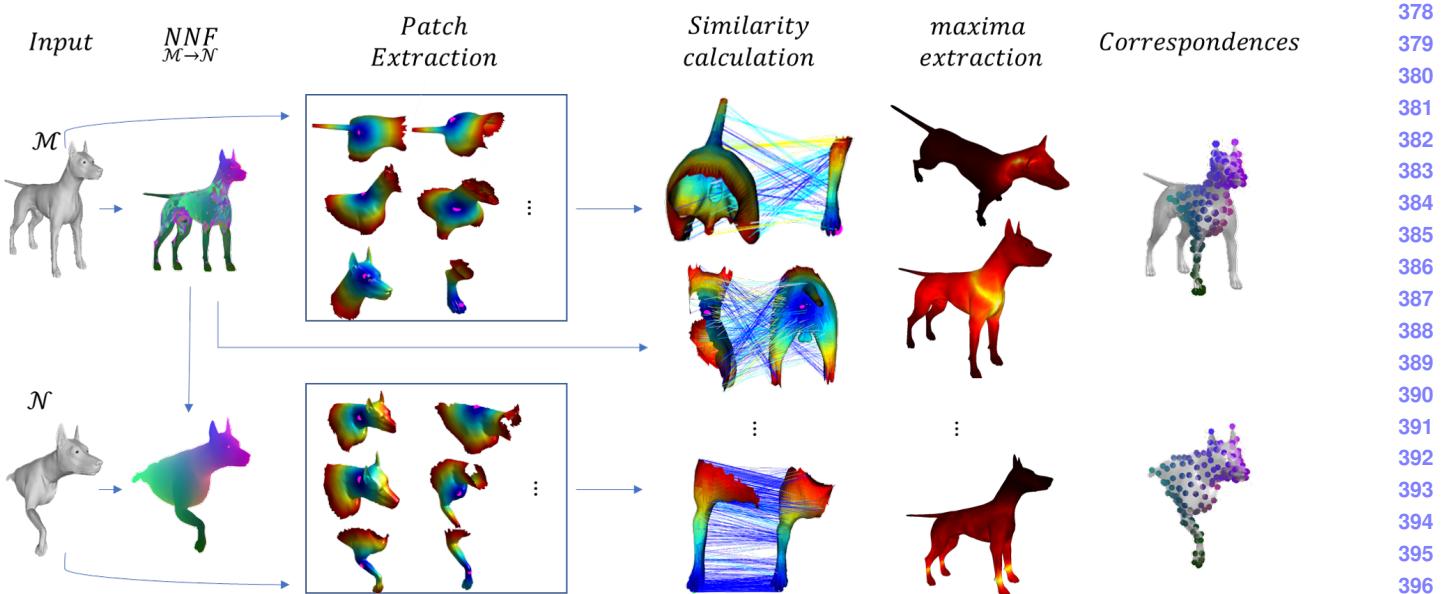


Figure 1. **Algorithm outline.** In the first step, we compute the nearest-neighbor field for \mathcal{N} and \mathcal{N} . Then, patches of the surfaces are extracted for every sample point in \mathcal{N} and for every vertex on \mathcal{M} . These patches cover the surface (i.e., may overlap) and represent 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) [11]. 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.

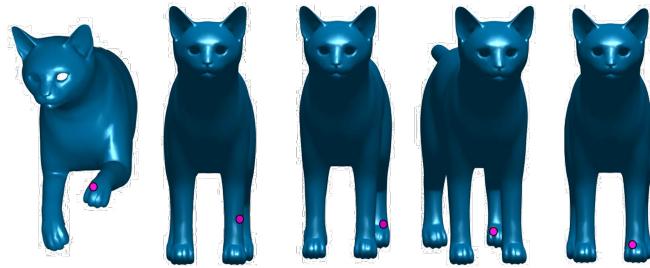
For each pair of patches of the same scale, $Q_i \subset \mathcal{M}$ and $P_i \subset \mathcal{N}$, we compute a similarity value. Recall that our goal is to reward a nearest-neighbor field with high diversity and low deformation. We will define the similarity function $DDIS$ that achieves it in Section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

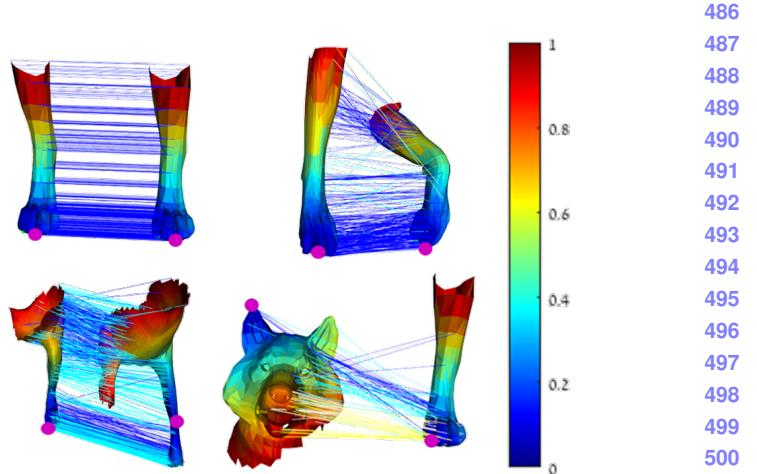
450
 451
 452
 453
 454
 455
 456
 457
 458
 459
5. Coherency-based correspondence refinement The result of Step 4 is a set of corresponding pairs of points. In most cases ($> 92\%$ on all our examples), the correspondences are correct. The goal of this step is to identify the incorrect ones and replace 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 , respectively;
 551 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

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

556 where $0 < \epsilon \ll Area(\mathcal{M})$ and is a 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 Having defined diversity between patches and deformation
 565 between points on these patches, we define the similarity
 566 between patches p and Q as:

$$567 \quad DDIS(P, Q) = \sum_{p_i \in P} \frac{1}{1 + r_i^*}. \quad (8)$$

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

5. Results

586 We have evaluated our method both qualitatively and
 587 quantitatively on several datasets: (1) the benchmark of
 588 *SHREC’16A—partial matching of deformable shapes* [8];
 589 (2) the even more challenging benchmark of *SHREC’16B—*
 590 *matching of deformable shapes with topological noise* [15].
 591 **AT: (3) Faust, (4) Archaeology** In all cases, our method
 592 wither outperformed the state-of-the-art methods or was
 593 competitive.

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

604 **AT: paragraph: explain that there are two distinct,**
605 yer related challenges: sparse and dense NA: The
606 SHREC’16 benchmarks accepts results of 2 distinct
607 types. (1) Sparse correspondences which aim to cover
608 the surface area uniformly, but sparsely. (2) Dense cor-
609 respondences which match every vertex on one shape to
610 the other shape. AT: paragraph: explain that you com-
611 pute only sparse and then use... to convert it into dense.
612 NA: Our method produces sparse correspondences, but
613 as our evaluation shows, these can be converted to dense
614 ones with minimal loss of quality. This is achieved by
615 feeding them as localized delta functions to the method
616 of [17] instead of standard shape descriptors.

617 **Qualitative results:** Figure 5 illustrates our results on two
 618 models from *SHREC’16A*. In this figure, the input model \mathcal{M}
 619 is color-coded according to its coordinates. The matches on
 620 the partial model \mathcal{N} (partial models and holes) are colored
 621 according to the match. Therefore, it is easy to visually
 622 verify if the match is correct or not.

623 It can be seen that our method obtains sparse corre-
 624 spondences of a high quality, since the dots on the model
 625 suit in color to the matching parts of \mathcal{M} . Furthermore,
 626 when comparing our dense-correspondence results to those
 627 of [25] (which computes dense correspondence directly),
 628 our method produces better results. It should be noted that
 629 symmetries are less of a problem in our method, due to dis-
 630 tance preservation between points in Equation (8) (note the
 631 legs). This is especially important when the model contains
 632 holes.

633 Figure 5 further demonstrates it, by color-coding the er-
 634 rors. The larger the error, the more reddish the color is. **AT:**
635 add this figure Figure 5 shows a couple of examples from
 636 the *SHREC’16B* dataset. On the topological noise bench-
 637 mark we can see bad matches are typically limited to the
 638 area of the topological noise, while PFM breaks completely
 639 on occasions.

640 **Quantitative results:** Next, we provide quantitative eval-
 641 uation of our method on the above datasets w.r.t previously
 642 reported results. The common error metric used is the nor-
 643 malized geodesic distance [12]. Specifically, let the corre-
 644 sponding point of $(q \in \mathcal{N}$, as found by the algorithm, be

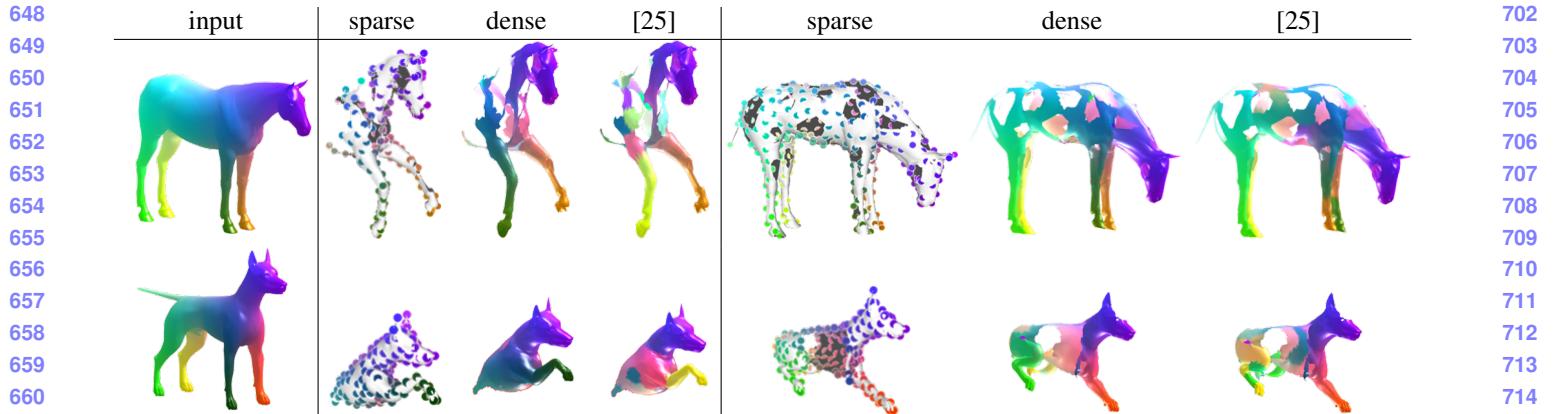


Figure 4. Results on Shrec'16A. Our results outperform those of [25], when run with the default parameters. **AT:** (1) create little images (2) remove boundaries (3) change the model color in sparse correspondences (4) the models must be in the same size

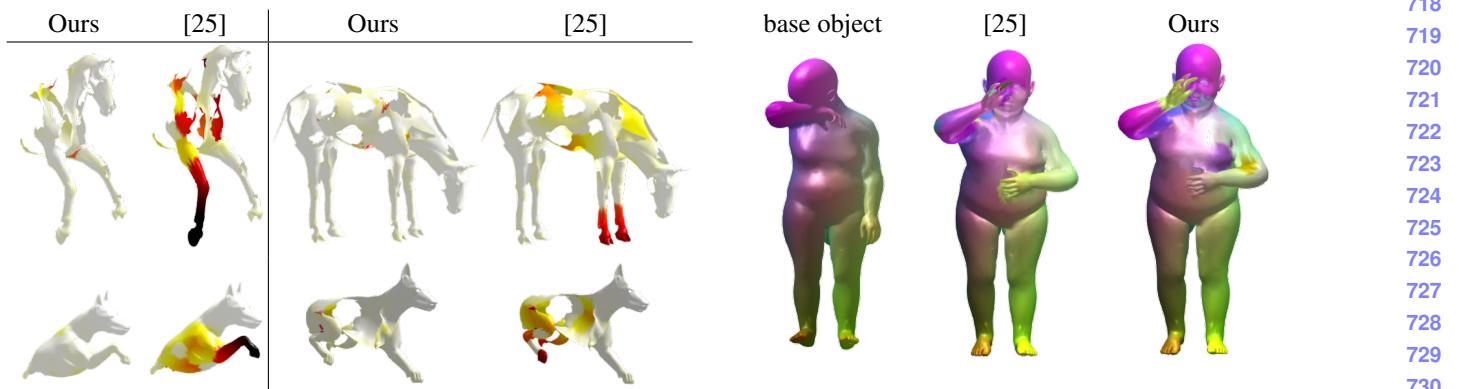


Figure 5. Errors on Shrec'16A. Our results outperform those of [25]

$p \in \mathcal{M}$) 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 5 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 [17, 25]. **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.

AT: Add a paragraph describing Figure 5 NA: Figure 5 Shows The mean geodesic error of a model as a function of it's partiality w.r.t. the neutral model. It can



Figure 6. Results on Shrec'16B. Our results outperform those of [25].

be seen that our method is pretty indifferent to the partiality of the model.

Figure 5 compares our results to results of state-of-the-art algorithms for SHREC'16B. Our method is competitive with that of [39], 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.

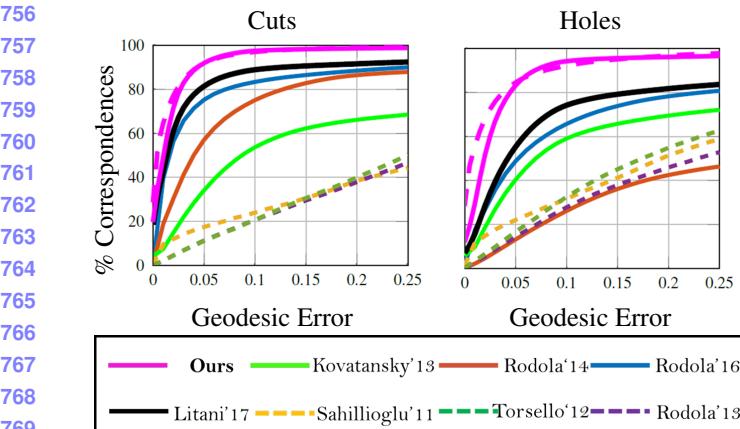


Figure 7. **Cumulative geodesic error curves on Shrec16A.** Our method (in magenta) outperforms other algorithms, both for the dense correspondence and for the sparse correspondence, on the two subsets of the dataset. **AT: 1. three images, (2) [Author'19] or Ours**

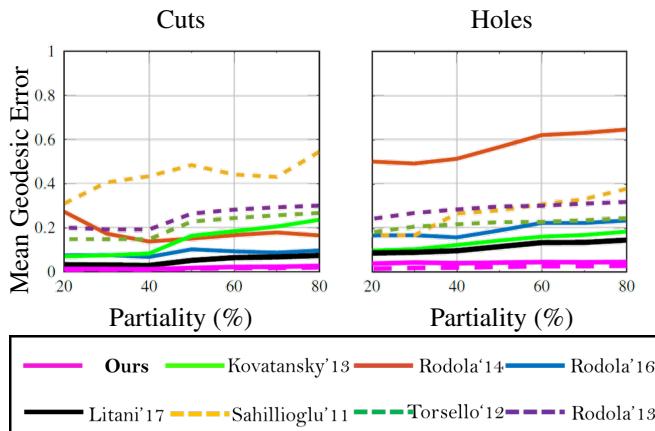


Figure 8. Error as a function of model partiality.

Drawbacks: xxx

AT: fifth paragraph: drawbacks with examples **NA:** Our method exhibits a few drawbacks. First and foremost is the asymptotic runtime - generating geodesic distances has an asymptotic complexity of $O(n^2 \log n)$ and computing 3D-DDIS similarity has an asymptotic runtime of $O(|S|n^2)$, which makes running on meshes with more than 15K vertices prohibitive. The algorithm also has six parameters which should be tuned. Additionally, surface patches such as the cat's tail, which are low in distinct shapes are hard to match. Finally, the method exhibits real difficulty dealing with topological noise.

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++

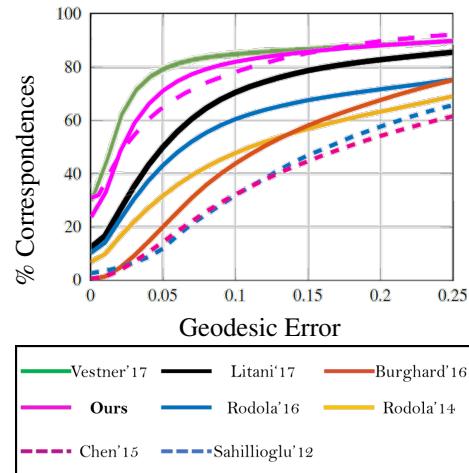


Figure 9. **Cumulative error curve on SHREC'16B. AT: same changes to the figure**

. For geodesics we have used Fast Marching Geodesics[13]. FPFH is calculated using the Point Cloud library[28]. The Nearest Neighbor field is computed with FLANN[20] 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 [17]. We had replaced the input dense descriptor field with localized smooth delta functions around our corresponding pairs. Instead of running 5 iterations we have found the best results are already achieved after only 1 iteration on SHREC16'A, and 2 on SHREC'16B. Since no default parameters are provided we have tuned the parameters loosely to achieve and aesthetically pleasing result on 15 models of cats from the SHREC16'A training set.

NA: Runtime and complexity 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 1/300 for a full model. (b) $|\mathcal{N}| \approx |\mathcal{M}|$. For simplicity sake we also treat each surface patch as though it contains all the points on its part. The complexity of different algorithm stages is thus:

- 1. NA: FPFH Calculation** Calculation of FPFH takes $O((|\mathcal{M}| + |\mathcal{N}|) \cdot k)$, where k is the number of neighbors for each point in a defined neighborhood of a radius R_F . In the radiiuses we use typically $k \ll n$
- 2. NA: NN search** Approximate nearest neighbor are calculated by building a kd-tree for \mathcal{N} which takes $O(d|\mathcal{N}| \log |\mathcal{N}|)$, where d is the feature dimension,

864
865
866
867
868
869
870
871
872
873874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

Figure 10. 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.

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

which is 33 for FPFH, and then performing $|\mathcal{M}|$ NN searches, each taking $O(\log|\mathcal{N}|)$, thus the overall complexity of this stage is $O(|\mathcal{M}|\log|\mathcal{N}|)$.

which share a nearest neighbor with κ other points will have their similarity contribution attenuated by $e^{(1-\kappa)}$. The comparison, illustrated in Figure 5 shows a significant improvement over the original formulation, stemming from a higher robustness to partiality.

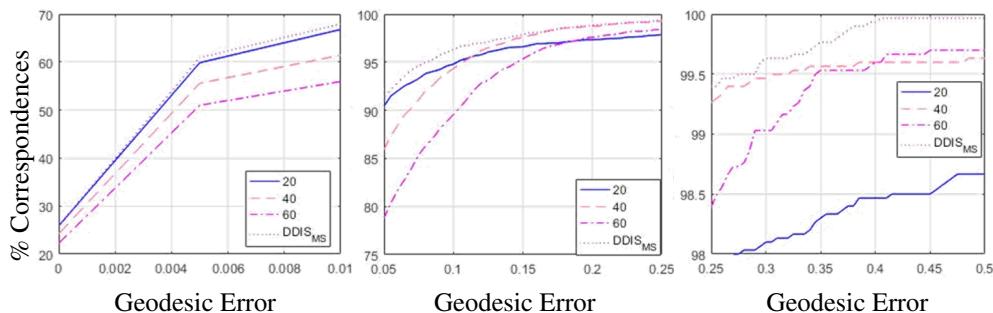
972
973
974
975
976
977
978
979
980
981
982
983
984

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

985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

SHREC partial matching dataset[8] 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.

1000

5.2. Error Metrics

1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012

The output of partial matching algorithms (as defined in[8]) 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 protocol [12]. 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

1013
1014
1015

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

1016
1017
1018
1019
1020

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.

1021
1022
1023

5.3. Sparse Correspondences on the SHREC16 Test set

1024
1025

We have tested the performance of DDIS in producing sparse correspondences on the SHREC16 Partial Matching

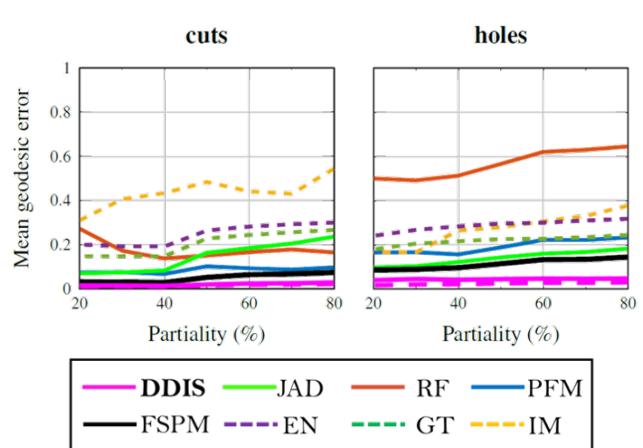
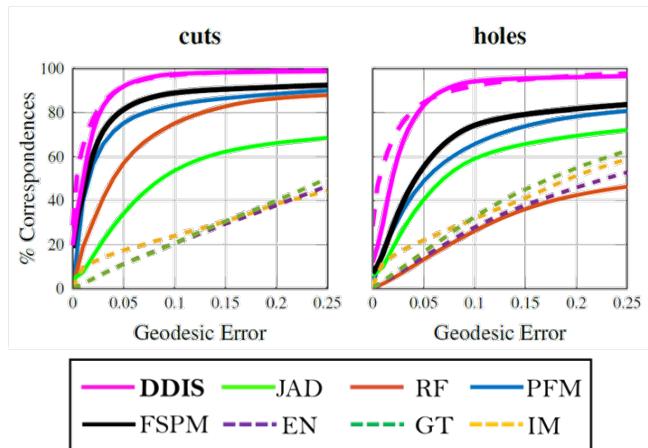
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
10381039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

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 [17] 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[17] as input instead of low level shape descriptors.

	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 with

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 $R_{FPFH} = 0.03 \cdot \sqrt{\text{Area}(\mathcal{M})}$, and NNF using FLANN. The average runtime on i7-4790 for 2 surfaces of 12000 vertices is 150s.



[htb]

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

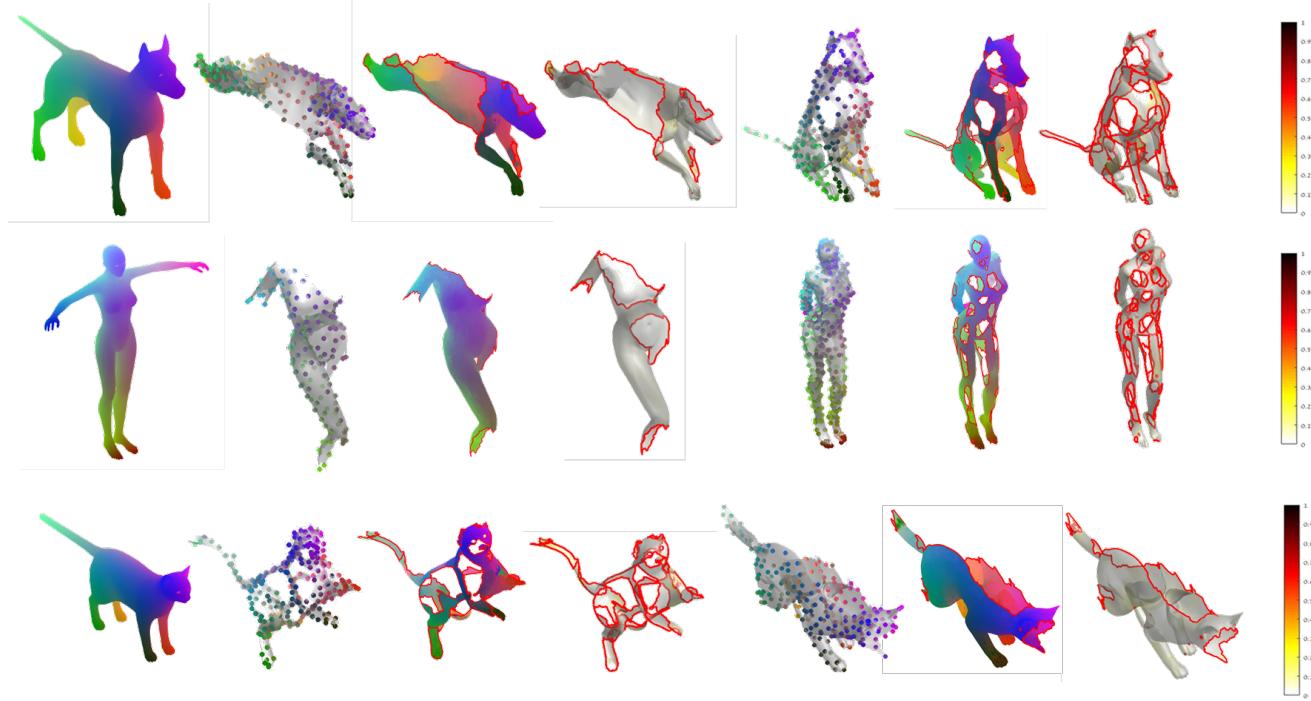


Figure 14. Good correspondences obtained by our method

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. Schlickwein, 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] 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
- [4] 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. CVPRW’08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008. 1, 2
- [5] 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*,

- 1188 84(2):163, 2009. 2 1242
- 1189 [6] 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 1243
- 1190 [7] 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 1244
- 1191 [8] 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, 9, 10 1245
- 1192 [9] 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 1246
- 1193 [10] 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 1247
- 1194 [11] 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 1248
- 1195 [12] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. 6, 10 1249
- 1196 [13] R. Kimmel. Fast marching methods for computing distance maps and shortest paths. 1996. 8 1250
- 1197 [14] 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 1251
- 1198 [15] 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 1252
- 1199 [16] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)*, 28(3):72, 2009. 2 1253
- 1200 [17] 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, 10 1254
- 1201 [18] 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 1255
- 1202 [19] F. Monti, D. Boscaini, and J. Masci. Geometric deep learning on graphs and manifolds using mixture model cnns. 2 1256
- 1203 [20] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331–340):2, 2009. 8 1257
- 1204 [21] 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 1258
- 1205 [22] J. Pokrass, A. M. Bronstein, and M. M. Bronstein. Partial shape matching without point-wise correspondence. *Nu- 1259*
merical Mathematics: Theory, Methods and Applications, 6(1):223–244, 2013. 2 1260
- 1206 [23] 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 1261
- 1207 [24] 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 1262
- 1208 [25] 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 1263
- 1209 [26] 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 1264
- 1210 [27] 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 1265
- 1211 [28] 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 1266
- 1212 [29] 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 1267
- 1213 [30] 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 1268
- 1214 [31] 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 1269
- 1215 [32] 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 1270
- 1216 [33] Y. Sahillioğlu 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 1271
- 1217 [34] A. Shtern and R. Kimmel. Matching the lbo eigenspace of non-rigid shapes via high order statistics. *Axioms*, 3(3):300–319, 2014. 2 1272
- 1218 [35] 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, 2009. Eurographics Association. 3 1273
- 1219 [36] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template matching with deformable diversity similarity. In *Proc. of IEEE 1274*
- 1220 [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [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] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183] [184] [185] [186] [187] [188] [189] [190] [191] [192] [193] [194] [195] [196] [197] [198] [199] [200] [201] [202] [203] [204] [205] [206] [207] [208] [209] [210] [211] [212] [213] [214] [215] [216] [217] [218] [219] [220] [221] [222] [223] [224] [225] [226] [227] [228] [229] [230] [231] [232] [233] [234] [235] [236] [237] [238] [239] [240] [241] [242] [243] [244] [245] [246] [247] [248] [249] [250] [251] [252] [253] [254] [255] [256] [257] [258] [259] [260] [261] [262] [263] [264] [265] [266] [267] [268] [269] [270] [271] [272] [273] [274] [275] [276] [277] [278] [279] [280] [281] [282] [283] [284] [285] [286] [287] [288] [289] [290] [291] [292] [293] [294] [295] [296] [297] [298] [299] [300] [301] [302] [303] [304] [305] [306] [307] [308] [309] [310] [311] [312] [313] [314] [315] [316] [317] [318] [319] [320] [321] [322] [323] [324] [325] [326] [327] [328] [329] [330] [331] [332] [333] [334] [335] [336] [337] [338] [339] [340] [341] [342] [343] [344] [345] [346] [347] [348] [349] [350] [351] [352] [353] [354] [355] [356] [357] [358] [359] [360] [361] [362] [363] [364] [365] [366] [367] [368] [369] [370] [371] [372] [373] [374] [375] [376] [377] [378] [379] [380] [381] [382] [383] [384] [385] [386] [387] [388] [389] [390] [391] [392] [393] [394] [395] [396] [397] [398] [399] [400] [401] [402] [403] [404] [405] [406] [407] [408] [409] [410] [411] [412] [413] [414] [415] [416] [417] [418] [419] [420] [421] [422] [423] [424] [425] [426] [427] [428] [429] [430] [431] [432] [433] [434] [435] [436] [437] [438] [439] [440] [441] [442] [443] [444] [445] [446] [447] [448] [449] [450] [451] [452] [453] [454] [455] [456] [457] [458] [459] [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] [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] [514] [515] [516] [517] [518] [519] [520] [521] [522] [523] [524] [525] [526] [527] [528] [529] [530] [531] [532] [533] [534] [535] [536] [537] [538] [539] [540] [541] [542] [543] [544] [545] [546] [547] [548] [549] [550] [551] [552] [553] [554] [555] [556] [557] [558] [559] [560] [561] [562] [563] [564] [565] [566] [567] [568] [569] [570] [571] [572] [573] [574] [575] [576] [577] [578] [579] [580] [581] [582] [583] [584] [585] [586] [587] [588] [589] [589] [590] [591] [592] [593] [594] [595] [596] [597] [598] [599] [600] [601] [602] [603] [604] [605] [606] [607] [608] [609] [610] [611] [612] [613] [614] [615] [616] [617] [618] [619] [620] [621] [622] [623] [624] [625] [626] [627] [628] [629] [630] [631] [632] [633] [634] [635] [636] [637] [638] [639] [640] [641] [642] [643] [644] [645] [646] [647] [648] [649] [650] [651] [652] [653] [654] [655] [656] [657] [658] [659] [660] [661] [662] [663] [664] [665] [666] [667] [668] [669] [669] [670] [671] [672] [673] [674] [675] [676] [677] [678] [679] [680] [681] [682] [683] [684] [685] [686] [687] [688] [689] [689] [690] [691] [692] [693] [694] [695] [696] [697] [698] [699] [699] [700] [701] [702] [703] [704] [705] [706] [707] [708] [709] [709] [710] [711] [712] [713] [714] [715] [716] [717] [718] [719] [719] [720] [721] [722] [723] [724] [725] [726] [727] [728] [729] [729] [730] [731] [732] [733] [734] [735] [736] [737] [737] [738] [739] [739] [740] [741] [742] [743] [744] [745] [745] [746] [747] [747] [748] [749] [749] [750] [751] [751] [752] [753] [753] [754] [755] [755] [756] [757] [757] [758] [759] [759] [760] [761] [761] [762] [763] [763] [764] [765] [765] [766] [767] [767] [768] [769] [769] [770] [771] [771] [772] [773] [773] [774] [775] [775] [776] [777] [777] [778] [779] [779] [780] [781] [781] [782] [783] [783] [784] [785] [785] [786] [787] [787] [788] [789] [789] [790] [791] [791] [792] [793] [793] [794] [795] [795] [796] [797] [797] [798] [799] [799] [800] [801] [801] [802] [803] [803] [804] [805] [805] [806] [807] [807] [808] [809] [809] [810] [811] [811] [812] [813] [813] [814] [815] [815] [816] [817] [817] [818] [819] [819] [820] [821] [821] [822] [823] [823] [824] [825] [825] [826] [827] [827] [828] [829] [829] [830] [831] [831] [832] [833] [833] [834] [835] [835] [836] [837] [837] [838] [839] [839] [840] [841] [841] [842] [843] [843] [844] [845] [845] [846] [847] [847] [848] [849] [849] [850] [851] [851] [852] [853] [853] [854] [855] [855] [856] [857] [857] [858] [859] [859] [860] [861] [861] [862] [863] [863] [864] [865] [865] [866] [867] [867] [868] [869] [869] [870] [871] [871] [872] [873] [873] [874] [875] [875] [876] [877] [877] [878] [879] [879] [880] [881] [881] [882] [883] [883] [884] [885] [885] [886] [887] [887] [888] [889] [889] [889] [890] [890] [891] [892] [892] [893] [894] [894] [895] [896] [896] [897] [898] [898] [899] [900] [900] [901] [902] [902] [903] [904] [904] [905] [906] [906] [907] [908] [908] [909] [910] [910] [911] [912] [912] [913] [914] [914] [915] [916] [916] [917] [918] [918] [919] [920] [920] [921] [922] [922] [923] [924] [924] [925] [926] [926] [927] [928] [928] [929] [930] [930] [931] [932] [932] [933] [934] [934] [935] [936] [936] [937] [938] [938] [939] [940] [940] [941] [942] [942] [943] [944] [944] [945] [946] [946] [947] [948] [948] [949] [950] [950] [951] [952] [952] [953] [954] [954] [955] [956] [956] [957] [958] [958] [959] [960] [960] [961] [962] [962] [963] [964] [964] [965] [966] [966] [967] [968] [968] [969] [970] [970] [971] [972] [972] [973] [974] [974] [975] [976] [976] [977] [978] [978] [979] [980] [980] [981] [982] [982] [983] [984] [984] [985] [986] [986] [987] [988] [988] [989] [990] [990] [991] [992] [992] [993] [994] [994] [995] [996] [996] [997] [998] [998] [999] [999] [999] [999]

- 1296 *Conf. on Computer Vision and Pattern Recognition*, pages 1350
1297 1311–1319, 2017. 1, 3, 9 1351
1298 [37] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of 1352
1299 histograms for local surface description. In *European conference 1353*
1300 on computer vision, pages 356–369. Springer, 2010. 1354
1301 2, 3 1355
1302 [38] A. Torsello. A game-theoretic approach to deformable shape 1356
1303 matching. In *Proceedings of the 2012 IEEE Conference on 1357*
1304 Computer Vision and Pattern Recognition (CVPR), CVPR 1358
1305 ’12, pages 182–189, Washington, DC, USA, 2012. IEEE 1359
1306 Computer Society. 1, 2 1360
1307 [39] M. Vestner, Z. Lähner, A. Boyarski, O. Litany, R. Slossberg, 1361
1308 T. Remez, E. Rodola, A. Bronstein, M. Bronstein, R. Kimmel, et al. 1362
1309 Efficient deformable shape correspondence via 1363
1310 kernel matching. In *3D Vision (3DV), 2017 International 1364*
1311 Conference on, pages 517–526. IEEE, 2017. 1, 2, 7 1365
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349