Computer Vision and Image Understanding 189 (2019) 102808

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Diffusion-based isometric depth correspondence[☆]

Emel Küpçü [*], Yücel Yemez

Koç University, Computer Engineering Department, Istanbul, Turkey

ARTICLE INFO

Communicated by: I Stamos

Keywords:
Isometric shape correspondence
Point cloud matching
Depth correspondence
Diffusion distance
Symmetry detection

ABSTRACT

We propose an iterative isometric point correspondence method that relies on diffusion distance to handle challenges posed by commodity depth sensors which usually provide incomplete and noisy surface data exhibiting holes and gaps. We formulate the correspondence problem as finding an optimal partial mapping between two given point sets, that minimizes deviation from isometry. Our algorithm starts with an initial rough correspondence between keypoints, obtained via any point matching technique. This initial correspondence is then pruned and updated by iterating a perfect matching algorithm until convergence in order to find as many reliable correspondences as possible. The resulting set of sparse but reliable correspondences then serves as a base matching from which a dense correspondence set is estimated. We additionally provide a global intrinsic symmetry detection technique which clusters a point cloud into its symmetric sides. We incorporate this technique into our point-based correspondence method so as to address the symmetrical flip problem and to further improve the reliability of our matching results. Our symmetry-aware correspondence method is especially effective on human shapes with global reflectional symmetry. We hence conduct experiments on datasets comprising human shapes and show that our method provides state of the art performance over depth frames exhibiting occlusions, large deformations, and topological noise.

1. Introduction

Depth sensors have become a commodity in the last half-decade, and this has opened up new opportunities in the field of computer vision and graphics as well as brought new challenges. Finding correspondences from depth is a key step for the success of various tasks in 3D computer vision, such as registration (Chang and Zwicker, 2011) and reconstruction (Liao et al., 2009).

Although the field of 3D shape correspondence has become quite mature in the last decade, finding reliable correspondences from depth, especially for non-rigid objects, is still an open problem. The first challenge is due to noisy data provided by commodity depth sensors, exhibiting holes and large gaps. This makes the estimation of geodesic distances on the surface geometry very difficult, and most isometric correspondence methods fail in this case. Moreover, when objects undergo non-rigid deformation, their topology can change drastically, which makes the computation of geodesic distances inconsistent between one pose and the other. Second, depth data is incomplete by acquisition since objects can be sensed only from one direction; hence correspondences exist only partially. Third, a major intricacy in correspondence estimation is the symmetrical flip problem, which is actually inherent to all isometric methods. Fourth, due to holes and gaps, surface information cannot reliably be triangulated; hence the need for point-based matching techniques that are currently less mature compared to existing mesh-based correspondence techniques.

In this paper, we present a mesh-free point-based method which can estimate reliable sparse/dense correspondences on non-rigid shapes undergoing large isometric deformations from noisy and incomplete depth data. We formulate the correspondence problem as finding an optimal partial mapping between two given point sets, minimizing deviation from isometry. We measure deviation from isometry based on a diffusion-based distance metric that we compute in a robust manner over noisy point clouds using an approximation of graph Laplacian. We experimentally show that our method outperforms state of the art techniques when tested on partial human body depth data.

The main contributions of this work are as follows:

- *A nested iterative prune and update correspondence algorithm:* We show that starting from an initial correspondence obtained by any point matching technique, it is possible to iteratively prune and update the initial matching, and obtain a set of sparse but reliable correspondences even with challenging noisy and incomplete data under occlusion. Our algorithm eliminates the unreliable matchings and outputs as many *reliable* correspondences as possible. Our sparse correspondence estimation algorithm is presented in

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cviu.2019.102808.

[*] Corresponding author.
E-mail address: ekupcu@ku.edu.tr (E. Küpçü).

https://doi.org/10.1016/j.cviu.2019.102808
Received 3 November 2018; Received in revised form 18 June 2019; Accepted 26 August 2019
Available online 29 August 2019

detail in Section 3.1. A preliminary version of this algorithm was published in (Küpçü and Yemez, 2017).

- *Coarse-to-fine dense point correspondence:* We adapt the idea of patchwise recursive dense matching on complete meshes originally proposed by Sahillioğlu and Yemez (2011) for computing dense correspondences on noisy and partial point clouds. To best of our knowledge, our work is the first to use a coarse-to-fine strategy for point correspondence. Our dense correspondence algorithm, which is robust and computationally efficient, is described in Section 3.2.
- *A clustering-based method for detecting symmetric sides of a point cloud:* Existing symmetry detection methods rather focus on finding symmetry axes whereas our method explicitly clusters a given point cloud into symmetric sides. Hence it is more resilient to noise and partial data, and can easily be incorporated into our point correspondence algorithm so as to make it symmetry-aware, as described in Section 4.
- *Symmetry-aware point correspondence:* We incorporate our clustering-based symmetry detection method into our point correspondence algorithm to alleviate the symmetric flip problem which is inherent to all isometric correspondence methods. Our algorithm is effective especially on (partial) human shapes with two intrinsically symmetric sides. Symmetry-aware correspondence methods existing in the literature are mostly mesh-based and rely on symmetry factored embeddings or descriptors whereas we make use of explicit symmetry cluster information to resolve ambiguities due to symmetrical flips.

2. Related work

Isometric deformations are the most common forms of non-rigidity. Most of the existing mesh-based isometric correspondence methods rely on geodesic distance information (Bronstein et al., 2006; Chen and Koltun, 2015; Sahillioğlu and Yemez, 2011, 2012, 2014; Van Kaick et al., 2011). However, conventional ways of computing geodesic distances such as shortest path algorithms become problematic on noisy surfaces with holes and gaps. A better alternative for noisy data is employing diffusion-based distance (Bronstein et al., 2010; Coifman and Lafon, 2006; Rodola et al., 2012; Sharma et al., 2011; Wang et al., 2011; Yoshiyasu et al., 2014). Diffusion distance takes into account all the paths existing between two surface points, thereby reducing the negative impact of noise, topological changes and incompleteness on estimation of distances.

The most common and generic approach for non-rigid point correspondence is to match individual surface points based on local shape descriptors (Johnson, 1997; Rusu et al., 2009; Tombari et al., 2010). However, since a local approach discards global shape cues such as isometry, it can easily yield incorrect correspondences especially when the shapes exhibit large variations in their local geometry, or when there are many points that are locally similar.

There exist a good number of isometric point correspondence estimation techniques that can handle unorganized point cloud data (Berger and Silva, 2012; Brunton et al., 2014; Guo et al., 2015; Huang et al., 2008; Kovnatsky et al., 2015; Li et al., 2008; Mateus et al., 2008; Rodolà et al., 2017; Tevs et al., 2009; Wei et al., 2016). These methods, except (Berger and Silva, 2012; Guo et al., 2015; Kovnatsky et al., 2015; Rodolà et al., 2017; Tevs et al., 2009; Wei et al., 2016), are not actually mesh-free techniques, performing poorly in the case of noisy and incomplete data since they rely on fitting intermediate mesh-based representations to point clouds, so that geodesic distances can be computed.

The mesh-free method presented by Guo et al. (2015) addresses the correspondence problem through piecewise rigid point registration by discovering parts in an iterative process. Their method hence relies on correct estimation of rigid parts as well as an approximation of geodesic distances by k-nearest neighbor graph distances, which are both problematic tasks especially in the case of occlusions as with depth data provided by commodity sensors. Two other related works on point-based correspondence are by Tevs et al. (2009) and by Berger and Silva (2012), where the former relies also on geodesic distances but within a probabilistic framework, and the latter uses medial diffusion to deal with incomplete data. Although Berger and Silva (2012) use a diffusion-based metric, the medial axis prior required in their technique is heavily dependent on shape topology and intolerant to large missing data, making it inapplicable to partial depth data with severe occlusions. Another promising point correspondence method is the learning-based technique proposed by Wei et al. (2016) for dense correspondence estimation on human models. The method learns how to extract features which are optimal for the correspondence problem over a training dataset and then uses nearest neighbor matching to find dense correspondences. As a learning-based method, it does not however necessarily generalize well to test data when trained on a limited dataset.

Another line of correspondence methods is based on the functional maps framework first introduced by Ovsjanikov et al. (2012). In theory, given the graph Laplacian of two shapes represented as point clouds or meshes, these methods can match complete models with resiliency against missing parts and noise (Kovnatsky et al., 2015), and can even handle the parts matching problem via matrix completion (Rodolà et al., 2017). These methods, however, cannot yet explicitly cope with partial and occluded depth data as provided by commodity sensors.

Probabilistic non-rigid registration techniques also provide us, as a byproduct, with point correspondences between point clouds (Billings et al., 2015; Chui and Rangarajan, 2003; Horaud et al., 2011; Jian and Vemuri, 2011; Ma et al., 2016; Mateus et al., 2008; Myronenko and Song, 2010; Tam et al., 2013). They find correspondences by optimizing a global objective to align point sets. Myronenko and Song (2010) and Ma et al. (2016) introduce non-rigid point registration methods that estimate parameters of transformations using Gaussian mixture models. In addition to global cues, Ma et al. (2016) incorporate local features to take into account the similarity of the neighboring structure of points. While point registration methods generate correspondences by matching all the points available, our focus is on finding partial mappings between point clouds based on isometric cues, with as many reliable correspondences as possible.

The symmetric flip problem is inherent to all isometric matching methods. There exist several shape correspondence methods in the literature, addressing explicitly this problem such as in (Dubrovina and Kimmel, 2011; Kim et al., 2010; Ovsjanikov et al., 2010, 2012; Sahillioğlu and Yemez, 2013; Yoshiyasu et al., 2014, 2016; Zhang et al., 2013), which are all mesh-based. The symmetric flip problem is also closely related to the problem of symmetry detection. We focus on intrinsic symmetry, which is addressed in various works with different applications (Jiang et al., 2013; Lipman et al., 2010; Mitra et al., 2006, 2013; Ovsjanikov et al., 2008; Raviv et al., 2007, 2010a,b; Sipiran et al., 2014; Wang et al., 2014; Xu et al., 2009, 2012). Rather than detecting symmetric parts in a scene or an object as in the works cited above, in this work we tackle the problem of clustering a given point cloud into "sides" which are globally and intrinsically symmetric such as in human shapes.

There exist only a few works in the literature, that address the problem of clustering into symmetric sides (Lipman et al., 2010; Xu et al., 2009). Xu et al. (2009) propose a voting based method to segment a model into two symmetric sides. They find point pair sets that vote for Voronoi boundaries between pairs so as to obtain the best intrinsic reflectional curve for each object in the scene. Similarly, Lipman et al. (2010) apply their intrinsic symmetry detection methodology to represent global reflectional symmetry with respect to a plane rather than a curve as in (Xu et al., 2009). To this end, they fit a plane to the centroids of the symmetry orbits based on an intrinsic symmetry dissimilarity measure which is calculated using the differences between the eigenvectors of the Laplacian matrix. For a given point cloud, a

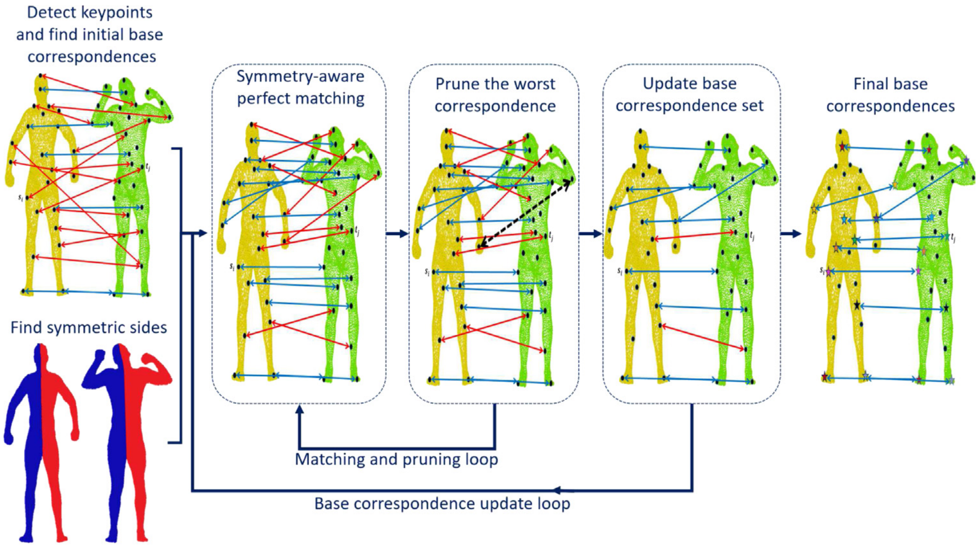


Fig. 1. Illustration of the main steps involved in our base correspondence algorithm. The output is a set of sparse correspondences. Red and blue lines indicate incorrect and correct correspondences, respectively.

symmetry plane – even if approximate – can always be found using their method. Küpçü and Yemez (2017) have used the plane fitting methodology of Lipman et al. (2010) to find symmetric sides of a point cloud and thereby to alleviate the symmetric flip problem for point correspondence estimation. However this plane fitting approach fails to reliably find symmetric sides for objects with complex motion such as human models with large deformations.

3. Correspondence algorithm

The inputs to our correspondence algorithm are the source P^S and target P^T point clouds obtained from depth frames of the object of interest. For initialization, we first apply a standard keypoint detection technique (any point-based 3D keypoint detection algorithm, such as SIFT (Lowe, 2004) or ISS (Zhong, 2009), can be used for this purpose). We represent the keypoint sets for the source and target with $S = \{s_1, s_2, \dots, s_{|S|}\}$ and $T = \{t_1, t_2, \dots, t_{|T|}\}$, respectively, where the number of detected keypoints usually varies with shape complexity.[1] Then, we match the detected points using any point-based matching algorithm such as standard descriptor matching (Aldoma et al., 2012; Tombari et al., 2010; Rusu et al., 2009) and point registration methods (Myronenko and Song, 2010; Ma et al., 2016). The resulting matching serves as the initial base correspondence set to be further improved. At the beginning of the correspondence algorithm, we also compute the graph Laplacian matrices of both point clouds, which we later use to calculate the diffusion-based distances between keypoints. We also segment the point clouds into their symmetric sides by using our clustering-based symmetry detection technique described in Section 4.

3.1. Base correspondence estimation

The initial keypoints S and T, detected previously, can be reconsidered for a better matching based on global isometric clues. For this purpose, we construct an isometric cost matrix C, where each entry c_{ij} represents the deviation from isometry of a candidate correspondence pair. To compute deviations from isometry, we need to rely on a set of known correspondences. We refer to this set as base correspondence, denoted by $B = \{(b_1^S, b_1^T), \dots, (b_{|B|}^S, b_{|B|}^T)\}$. The set B is initially set to B_0, that is, the correspondence obtained in the preprocessing step.

The isometric cost of matching a keypoint s_i on the source keypoint set S with t_j on the target T is then calculated by:

$$c_{ij} = \frac{1}{|B|} \sum_{(b_l^S, b_l^T) \in B} |d_S(s_i, b_l^S) - d_T(t_j, b_l^T)|. \tag{1}$$

where $d_S(.,.)$ and $d_T(.,.)$ denote diffusion distances between keypoints (Coifman and Lafon, 2006; Lafon, 2004). The resulting cost matrix C is bipartite, so we can apply the Blossom V algorithm, minimum-weight perfect matching algorithm of Kolmogorov (2009), to match all the keypoints from scratch, similar to Sahillioğlu and Yemez (2014). Each entry $c_{ij} \in C$ is normalized to be in the range $[0, 1)$ by $c_{ij} \leftarrow (1 - e^{-c_{ij}})$. The perfect matching algorithm that we employ requires a square cost matrix. Thus, we add virtual nodes to the smaller keypoint set and set the corresponding costs for non-existent pairs to infinity. The perfect matching algorithm results in a one-to-one mapping B, from which we then remove the pairs including the virtual nodes.

We iteratively modify this matching B to make it as reliable as possible. We achieve this goal with a nested loop of *pruning* and *update* iterations. While the inner loop prunes the unreliable correspondences based on an isometric error criterion, the outer iterations gradually update the base correspondence set (that was initially set as B_0). At the end of these iterations, we expect to end up with a partial one-to-one mapping that establishes a reliable (sparse) correspondence between keypoints.

We compute the isometric error, $E_{\text{iso}}(b_i^S, b_i^T)$, of a given correspondence pair $(b_i^S, b_i^T) \in B$, in terms of its deviation from isometry with respect to other available pairs in B:

$$E_{\text{iso}}(b_i^S, b_i^T) = \frac{1}{|B| - 1} \sum_{(b_l^S, b_l^T) \in B, i \neq l} |d_S(b_i^S, b_l^S) - d_T(b_i^T, b_l^T)| \tag{2}$$

If $(b_i^S, b_i^T) \in B$ is a correct matching pair, its isometric error is expected to be close to zero. Hence the correspondence set B can be pruned by eliminating the pairs having relatively larger errors compared to the others. The reliability of the isometric error defined in Eq. (2) depends on the correctness of B itself. Thus we perform pruning in an iterative scheme, one pair (the worst one) at a time, and each time we reinvoke the diffusion-based perfect matching algorithm with the pruned base set. At each iteration, we also remove the keypoints of the eliminated pair from the keypoint sets S and T. The pruning and perfect matching tasks are iterated until the gap between maximum and minimum isometric errors over the pairs becomes small enough according to a predesignated threshold value τ_1.

[1] We assume that the shapes to be matched contain repeatable and distinctive keypoints which might be missing for some certain shape categories such as those based on simple quadrics.

The modified correspondence set B resulting from the iterative pruning algorithm is smaller but usually much more reliable than the input correspondence. Hence it can be used to update the initial base correspondence for the next run of the iterative pruning algorithm in the outer loop. At the beginning of each outer iteration, the initial base correspondence is renewed with the current B, whereas the keypoint sets, S and T, are set back to their original content. Hence, the keypoints that are discarded during iterative pruning due to mismatches are reconsidered for other possible matches based on a more reliable estimation of isometric errors. The outer iterations terminate when the mean isometric error converges, i.e., when there is no further improvement on the base correspondence set B. We illustrate the main steps involved in our base correspondence algorithm in Fig. 1.

3.2. Dense correspondence estimation

In this section, we extend our base correspondence set B to a sequence of denser base correspondences $\{B^k\}$ with increasing levels of detail using a coarse-to-fine strategy adapted to partial point clouds as illustrated in Fig. 2.

We initialize the base correspondence set at the coarsest level $k = 0$ with $B^0 = B$, and then recursively subdivide the point clouds P^S and P^T into patches via clustering followed by subsampling. Initially, $P^{S,0} = P^S$ and $P^{T,0} = P^T$. We cluster the points within each patch P_l^{k-1} at level $k-1$ starting from $k = 1$ into subpatches P_i^k (dropping the superscripts S and T for notational simplicity). This is done by grouping points $p_j \in P_l^{k-1}$ according to their diffusion distance to the base points b_i^{k-1} of the previous level patch:

$$P_i^k = \{p_j \in P_l^{k-1} | i = \arg\min_m d(p_j, b_m^{k-1})\}. \tag{3}$$

Hence each base point b_i^{k-1} represents a subpatch P_i^k at the next level. We then sample at most N_S points as uniformly as possible from each patch. We denote the subsampled patch by $\bar{P}_i^k$. For subsampling, we sort all the points within a patch P_i^k in ascending order with respect to their diffusion distances to the representing base point b_i^{k-1}, and uniformly subsample the sorted list. Hence the sampled points tend to be equally distributed within each patch. If the number of points in P_i^k is already less than or equal to N_S, we set $\bar{P}_i^k = P_i^k$ and stop further recursion for that patch.

The points within each subsampled patch $\bar{P}_i^k$ are then matched with the points in the corresponding subsampled patch of the other point cloud. We use perfect matching (as described in Section 3.1) to match the sampled points in the corresponding patches. The isometric costs for perfect matching are computed using Eq. (1) based on the correspondence set B^0 regardless of the current level. We denote the resulting patch correspondence by B_i^k. We calculate the isometric error of each correspondence pair in B_i^k based on B^0 by:

$$E_{\text{iso}}(b_i^{S,k}, b_i^{T,k}) = \frac{1}{|B^0|} \sum_j |d_S(b_i^{S,k}, b_j^{S,0}) - d_T(b_i^{T,k}, b_j^{T,0})| \tag{4}$$

where $(b_j^{S,0}, b_j^{T,0}) \in B^0$.

The correspondences with isometric errors larger than a threshold are eliminated from the patch correspondence B_i^k. To define the threshold, we calculate the maximum error between base correspondences B^0 according to Eq. (2). We select the threshold as a factor ($\tau_2 > 1$) of the maximum error. The union of all patch correspondences B_i^k over i generates the base correspondence B^k to be used at the next level. The recursive sampling and matching process continues until no pair of points is left to match between the two corresponding patches. We set the resulting correspondences at the highest level as the final dense correspondences B^{k^*}.

We note that the base correspondences in B^0 are fine-tuned gradually throughout the patchwise recursive matching process. At the end of each level k, we consider every base correspondence pair $(b_i^{S,0}, b_i^{T,0})$ for a possible update. If the base point $b_i^{S,0}$ is already matched with $b_i^{T,0}$ at that level (or vice versa), then we continue without updating the base pair. If $b_i^{S,0}$ is matched with a different point x^T, and likewise $b_i^{T,0}$ is matched with x^S, then we have three candidate base pairs: $(b_i^{S,0}, b_i^{T,0})$ or $(b_i^{S,0}, x^T)$ or $(x^S, b_i^{T,0})$. Among those, we pick whichever yields the minimum isometric error computed via Eq. (4) over the dense correspondence B^k of level k (hence B^0 is simply replaced by B^k in Eq. (4) for computation of isometric error in this case).

Algorithm 1 Dense Correspondence Estimation

```
Require: Base correspondence $B$ and point clouds $P^S, P^T$
Ensure: Sequence of denser correspondences $\{B^k\}_{k=0}^{k^*}$
  Initialize with $B^0 = B$, $k = 1$, $P^{S,0} = P^S$, $P^{T,0} = P^T$;
  repeat
    For each $(b_i^S, b_i^T) \in B^{k-1}$
      Create patches $P_i^{S,k}$ and $P_i^{T,k}$ using Eq. (3);
      Find subsampled patch $\bar{P}_i^{S,k} \subseteq P_i^{S,k}$ if $|P_i^{S,k}| > N_S$;
      Find subsampled patch $\bar{P}_i^{T,k} \subseteq P_i^{T,k}$ if $|P_i^{T,k}| > N_S$;

      Find $B_i^k$: $\bar{P}_i^{S,k} \rightarrow \bar{P}_i^{T,k}$ relying on $B^0$ via
        diffusion-based perfect matching (Section 3.1);
      Eliminate outliers from $B_i^k$;
    Set $B^k = \bigcup_i B_i^k$;
    Fine-tune base correspondence set $B^0$ using $B^k$;
    Increment level $k$;
  until no unvisited point left to match
```

The above coarse-to-fine matching strategy has two advantages regarding computational cost and robustness. First, patchwise matching reduces the search space considerably, providing efficiency as well as robustness, since a point in a patch is very likely to match a point in the corresponding patch on the other point cloud, assuming that initial base matchings are mostly reliable. Second, it is indeed computationally very demanding to match all the points at once due to the large complexity of the perfect matching algorithm, and constraining the number of points sampled within each patch to a reasonably small number (such as $N_S = 50$) makes perfect matching affordable.

We provide the pseudocode of our dense correspondence estimation method in Algorithm 1.

3.3. Computational complexity

The complexity of our base correspondence estimation algorithm is dominated by the min-weight perfect matching algorithm with $O(N_0^2 \log N_0)$ cost (Kolmogorov, 2009), where $N_0 = \max(|S|, |T|)$, S and T are source and target keypoint sets. In the dense correspondence estimation part, we match $O(\frac{N_P}{N_S})$ patches with $O(N_S)$ points in each, where N_S is the number of points sampled from each patch, $N_P = \max(|P^S|, |P^T|)$, P^S and P^T are source and target point cloud sets. Again, the cost is dominated by the min-weight perfect matching algorithm, which is repeated for each patch and incurs a cost of $O(N_S^2 \log N_S)$ per patch. Therefore, the total complexity of the dense correspondence estimation part is $O(N_S^2 \log N_S) O(\frac{N_P}{N_S}) = O(N_P N_S \log N_S)$. In practice, we have $O(N_0) \sim O(N_S) \sim O(\sqrt{N_P})$. Thus, the overall complexity of our solution can be written as $O(N_P^{\frac{3}{2}} \log N_P)$, which is better than $O(Q^2 N_P)$ complexity of the PR-GLS algorithm (Ma et al., 2016) (one of our baselines), where $Q \sim N_S$ in practice. We note that the CPD algorithm (Myronenko and Song, 2010), which can be seen as precedent to the PR-GLS method, has $O(N_P)$ time complexity but less accuracy as we demonstrate experimentally.

4. Symmetry detection

In this section, we first describe our clustering-based method for detecting symmetric sides of a point cloud and then explain how to

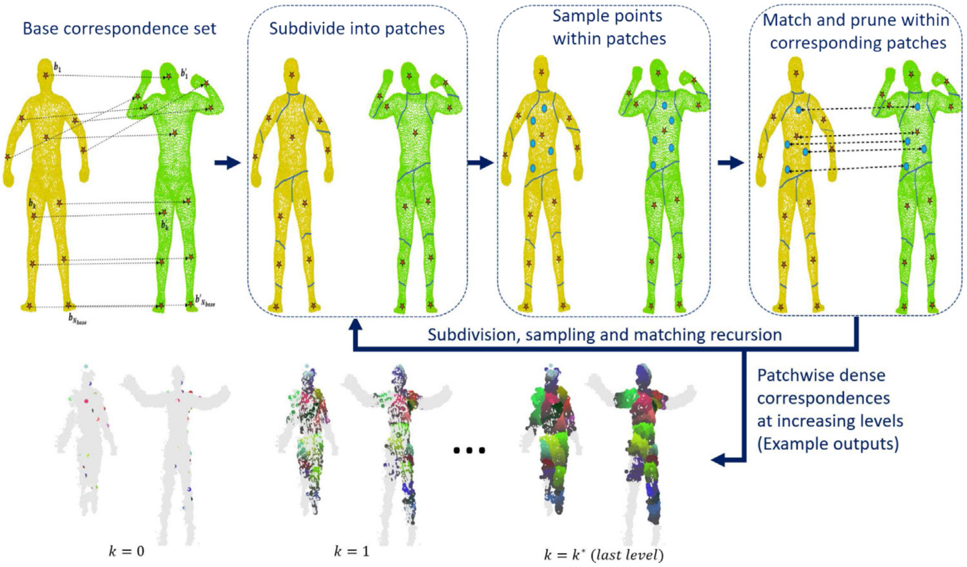

Base correspondence set
b_1
b'_1
b_k
b'_k
$b_{N_{base}}$
$b'_{N_{base}}$
Subdivide into patches
Sample points within patches
Match and prune within corresponding patches
Subdivision, sampling and matching recursion
Patchwise dense correspondences at increasing levels (Example outputs)
$k = 0$
$k = 1$
$k = k^*$ (last level)

Fig. 2. Illustration of our dense correspondence algorithm. The input base correspondences are extended to a sequence of denser correspondences via recursive subdivision, sampling and matching.

integrate this symmetry information into our point correspondence algorithm.

4.1. Clustering into symmetric sides

A shape P is said to be *intrinsically symmetric* with respect to a transformation $\mathcal{F} : P \rightarrow P$, if an intrinsic metric d is preserved by the transformation $\mathcal{F}$, i.e., $d_P(p,q) = d_P(\mathcal{F}(p),\mathcal{F}(q))$ for all $p,q \in P$ (Mitra et al., 2013). Since we deal with noisy and partial point clouds, we use the diffusion distance as our intrinsic metric for symmetry computations.

The first step in symmetry detection is to find this transformation $\mathcal{F}$, which provides us with the set of all symmetric point pairs in the point cloud. However since symmetries are never perfect on real shapes, we rather seek for an approximation of this transformation. Hence one can define it as the transformation that maximally preserves the diffusion distances between points, to minimize a global symmetry distortion function as given below:

$$\mathcal{F} = \arg\min_{\mathcal{G}} \sum_{p,q\in P} \max\{|d_P(p,q) - d_P(\mathcal{G}(p),\mathcal{G}(q))|, \\ |d_P(p,\mathcal{G}(q)) - d_P(\mathcal{G}(p),q)|\}. \qquad (5)$$

Note that the transformation $\mathcal{F}$ is partial in most cases, even if the object is globally symmetric as a whole, due to the incompleteness of real acquisition data, such as in the case of depth sensors. Thus, to address this optimization problem, we resort to an iterative algorithm with a sub-optimal solution that aims to maintain the global symmetry distortion as low as possible while trying to find as many symmetric point pairs as possible. We explain this algorithm in the sequel.

In practice, for efficiency, we sample a set of points $\bar{P} = \{\bar{p}_1, \dots, \bar{p}_{|\bar{P}|}\}$ from the point cloud P and use this set to find the mapping $\mathcal{F} : \bar{P} \rightarrow \bar{P}$. For sampling, we use a voxel grid based downsampling and filtering method, as implemented in the Point Cloud Library (Rusu and Cousins, 2011).

We first estimate an initial $\mathcal{F}$ based on the distribution of diffusion distances from each sampled point $\bar{p}_i \in \bar{P}$ to all other points $p \in P$. To this end, we compute the normalized diffusion distance histogram, $\boldsymbol{h}_i$, as a descriptor for each sampled point $\bar{p}_i$ as in (Raviv et al., 2007, 2010a,b). We then match each sampled point $\bar{p}_i$ with the point $\bar{p}_j$ that provides the minimum distance between histograms. The matching process explained above provides us with an initial *many-to-one* relation, i.e., a set of point pairs. We first make this relation *one-to-one*. For this purpose, we calculate the symmetric distortion $E_{sym}(\bar{p}_i,\bar{q}_i)$ of each point pair $(\bar{p}_i,\bar{q}_i)$ as in (Jiang et al., 2013) (see also Eq. (5)):

$$E_{sym}(\bar{p}_i,\bar{q}_i) = \frac{1}{|\mathcal{F}|-1} \sum_{(\bar{p}_j,\bar{q}_j)\in\mathcal{F}-(\bar{p}_i,\bar{q}_i)} \max\{|d_P(\bar{p}_i,\bar{p}_j) \\ - d_P(\bar{q}_i,\bar{q}_j)|, |d_P(\bar{p}_i,\bar{q}_j) - d_P(\bar{p}_j,\bar{q}_i)|\}. \qquad (6)$$

Note that we use $\mathcal{F}$ to denote the symmetry transformation as well as the set of symmetric point pairs, hence the relation induced by this transformation. In order to make the relation $\mathcal{F}$ one-to-one, we first pick the pair $(\bar{p}_*,\bar{q}_*)$ with minimum symmetric distortion, and eliminate each $(\bar{p}_*,\cdot)$, $(\cdot,\bar{p}_*)$, $(\bar{q}_*,\cdot)$ and $(\cdot,\bar{q}_*)$ except $(\bar{p}_*,\bar{q}_*)$ from $\mathcal{F}$. Then, we pick from the remaining the pair with the next minimum distortion, and likewise eliminate all other pairs including any of those points. We repeatedly perform this process until $\mathcal{F}$ becomes one-to-one.

The resulting one-to-one mapping may still include pairs with large distortion. Therefore, we prune the set using the same iterative procedure used for base correspondence estimation (see Section 3.1). At each iteration, we first re-calculate E_{sym} values of the pairs in the set (since it depends on $\mathcal{F}$ that is modified through iterations). Then, we eliminate the worst point pair from the set according to symmetric distortion. We continue re-calculating symmetric distortion and eliminating the worst pair until the minimum and the maximum distortions are close to each other within a given ratio τ_{sym}. The whole process of minimization of symmetric distortion outputs a mapping $\mathcal{F}$ that is one-to-one, but mostly partial, due to pruning.

We use the obtained symmetry set $\mathcal{F}$ to cluster the point cloud P into its symmetric sides, P_- and P_+. Each symmetric point pair $(\bar{p}_i,\bar{q}_i)$ votes for every point $p_j \in P$, in order to determine on which side of the shape p_j is. After the voting process, the points with negative-valued accumulation of votes are assigned to side P_- whereas the others to side P_+. The votes of the pairs take value depending on what we call a distance weight vector $\boldsymbol{w}(\bar{p}_i) = (w_1(\bar{p}_i), w_2(\bar{p}_i), \dots, w_{|P|}(\bar{p}_i))$, which is calculated for each sampled point $\bar{p}_i$ by $w_j(\bar{p}_i) = e^{-d_P(\bar{p}_i,p_j)}$. Hence a distance weight vector $\boldsymbol{w}(\bar{p}_i)$ assigns higher weights to the points closer to $\bar{p}_i$ and lower weights to the points further from $\bar{p}_i$ in the point cloud P.

We start the voting process with the pair $(\bar{p}_*,\bar{q}_*)$ having the least symmetric distortion E_{sym}. We define a vote vector $\boldsymbol{v} = (v_1, v_2, \dots, v_{|P|})$ and initialize it by $\boldsymbol{v} = \boldsymbol{w}(\bar{p}_*) - \boldsymbol{w}(\bar{q}_*)$, where each v_k represents the amount of votes for p_k. Thus initially, the points close to $\bar{p}_*$ take higher absolute votes with positive sign, whereas the points close to $\bar{q}_*$ take

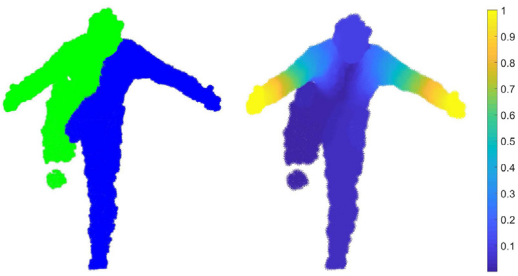


Fig. 3. A symmetry detection example from the Human dataset: signs of the symmetry weights (left), absolute values of the symmetry weights (right).

higher absolute votes with negative sign. The distances between point pairs are calculated by:

$$d_P((\bar{p}_k,\bar{q}_k),(\bar{p}_l,\bar{q}_l)) = \min(\frac{d_P(\bar{p}_k,\bar{p}_l)+d_P(\bar{q}_k,\bar{q}_l)}{2}, \frac{d_P(\bar{p}_k,\bar{q}_l)+d_P(\bar{q}_k,\bar{p}_l)}{2}) \qquad (7)$$

as adopted from (Xu et al., 2012). We then proceed with the point pair which is closest to $(\bar{p}_*,\bar{q}_*)$, say $(\bar{p}_i,\bar{q}_i)$, and we decide on the sides of the points of the pair, $\bar{p}_i$ and $\bar{q}_i$, based on the current votes accumulated in $\boldsymbol{v}$. This side assignment can be done more reliably for the closest pair compared to the farther pairs. To assign the sides, we compare the two values calculated by $\sum_{p_k\in P} w_k(\bar{p}_i)v_k$ and $\sum_{p_k\in P} w_k(\bar{q}_i)v_k$, respectively for $\bar{p}_i$ and $\bar{q}_i$. The point yielding the higher value, say $\bar{p}_i$, is assigned to side P_+ with $s(\bar{p}_i)=1$, and the other to side P_- with $s(\bar{q}_i)=-1$, where s is an indicator function taking value either 1 or −1. Then, the pair $(\bar{p}_i,\bar{q}_i)$ votes for all the points in P by incrementing the vote vector by $\boldsymbol{v}=\boldsymbol{v}+s(\bar{p}_i)(\boldsymbol{w}(\bar{p}_i)-\boldsymbol{w}(\bar{q}_i))$. We repeat this process, proceeding each time with the point pair that is the next closest to $(\bar{p}_*,\bar{q}_*)$, until no more point pair is left for voting. Finally we normalize the values in the vote vector $\boldsymbol{v}$ by the maximum absolute vote in the vector. We refer to the resulting vote values as *symmetry weights*, the signs of which provide us with two symmetric sides as given in Fig. 3 (left). We denote the symmetry weight vector by $\tilde{\boldsymbol{v}}$ and the symmetry weight for an individual point p_i by $\tilde{v}(p_i)$. Moreover, the absolute values of these weights indicate the confidence values for the side assignments of the points in P as shown in Fig. 3 (right). A high confidence value means that the symmetry side decision is strong for that point.

We note that the complexity of our clustering-based symmetry detection technique is $O(|P|^{\frac{3}{2}})$, which is executed only during preprocessing.

4.2. Symmetry-aware correspondence

We incorporate the symmetry weights $\tilde{v}$ into the cost calculation (Eq. (1)) of our base correspondence estimation algorithm so as to alleviate the symmetric flip problem.

Given two shapes (source and target) to be matched, the first step is to match the sides of the corresponding point clouds P^S and P^T. For example, if the shapes are frontal human models, then the left-hand side of the source (P^S_+ or P^S_-) is to be matched with the left-hand side of the target. For this purpose, we first register the two point clouds using the well-known rigid ICP method (Rusinkiewicz and Levoy, 2001) in order to make the corresponding sides as close to each other as possible. Then, we check whether the signs of the sides of the registered point clouds are matching. Let the symmetry weights of source P^S and target P^T be $\tilde{\boldsymbol{v}}^S$ and $\tilde{\boldsymbol{v}}^T$, respectively. We define a side matching score $\kappa_{S\to T}$ that decides on the corresponding sides of the point clouds by:

$$\kappa_{S\to T} = \sum_{p_i\in P^S} |\tilde{v}^S(p_i)+\tilde{v}^T(q_i)| - \sum_{p_i\in P^S} |-\tilde{v}^S(p_i)+\tilde{v}^T(q_i)| \qquad (8)$$

where $q_i = \mathrm{argmin}_{q_j\in P^T}\|p_i-q_j\|$ is obtained via registration. Then, we update the sign of the weights of P^S by $\tilde{\boldsymbol{v}}^S = \mathrm{sign}(\kappa_{S\to T})\tilde{\boldsymbol{v}}^S$. In practice, we register the point clouds both ways. We also calculate $\kappa_{T\to S}$, and update the signs of the weights based on the larger of the side matching scores $\kappa_{S\to T}$ and $\kappa_{T\to S}$.

Once the sides are matched, we next modify the cost function in Eq. (1) based on the symmetry weights of the point clouds so as to penalize matching of the points that are located on the non-corresponding sides of the shapes. So whenever a source point $s_i\in P^S$ and a target point $t_j\in P^T$ are found to be located on the non-corresponding sides, the following modified cost function c'_{ij} becomes in use:

$$c'_{ij} = c_{ij}(1+\frac{\alpha}{2}(|\tilde{v}^S(s_i)|+|\tilde{v}^T(t_j)|)) \qquad (9)$$

The penalization factor $\alpha > 0$ is set as $\alpha = 1.5$ in all our experiments. Note that for $\alpha = 0$, the symmetry-aware cost c'_{ij} becomes identical to c_{ij} in Eq. (1).

Due to the partial and noisy nature of the depth data, the estimated symmetry information of a given shape may deviate from the true symmetry on some body parts, especially on the parts that have topological misconnections or missing parts. We observe that the points at those parts with ambiguities usually have low confidence values. Therefore, we penalize matching of points only if they both have relatively high confidence values, i.e., the parameter α in Eq. (9) is reset to zero whenever $\min(|\tilde{v}^S(s_i)|,|\tilde{v}^T(t_j)|) \le \beta$, where β is set experimentally to 0.01.

5. Experimental results

5.1. Evaluation setup

We evaluate the performance of our algorithm on three partial human body depth datasets: The Berkeley motion human action dataset (MHAD) (Ofli et al., 2013; Teleimmersion Lab), the dataset that we collected (Human) and the dataset of Guo et al. (2015), all containing noisy and incomplete depth data of freely moving subjects, captured using Kinect v1. We convert all depth frames to 3D point cloud representations and discard color.

We pick 12 depth frames from the MHAD dataset and generate their 66 pair combinations. We select frames which exhibit large non-rigid deformations with respect to each other, and each from a different action set, such as jumping and throwing. Each frame in this dataset has approximately 24K points and 43 ground-truth marker positions.

The other dataset (Human) that we have collected contains RGB-D frames of a human subject exhibiting larger non-rigid motion compared to MHAD. The Human dataset includes 6 frames with 33 manually selected ground-truth keypoints out of approximately 24K points on each, and 15 model pairs.

We also use the dataset of Guo et al. (2015) to compare our dense correspondence results with the results of Guo et al. (2015). This dataset includes 9 frames with approximately 29K points on each. We manually selected 14 ground-truth keypoints on each frame.

The baseline methods to which we compare our *base* correspondence estimation algorithm are the SHOT descriptor matching (using a publicly available implementation[2]), and two state of the art non-rigid point registration methods with publicly available codes: the CPD method of Myronenko and Song (2010) and the PR-GLS method of Ma et al. (2016), both of which also provide point correspondences. The descriptor matching algorithm is based on the Euclidean distance between the SHOT descriptors (Tombari et al., 2010) in two directions: from target to source and from source to target. The intersection of the resulting correspondence sets is the final output and is referred to as *reciprocal correspondences* (Pajdla and Van Gool, 1995). We also employ the resulting correspondence set of each baseline as the initial base correspondence set B_0 in our matching experiments.

[2] http://pointclouds.org/.

We compare our *dense* correspondence results with the results of three state of the art methods. Besides CPD and PR-GLS, we additionally compare our method with the method of Guo et al. (2015), which is a recent point-based correspondence technique relying on geodesic distances. To experiment with this method, we use the code provided by the authors.

We employ two evaluation metrics: (1) The deviation from isometry, i.e., the isometric error, for a given correspondence pair is computed via E_{iso} measure given in Eq. (2). (2) The deviation from ground-truth, i.e., the ground-truth error, is calculated based on each pair's closest ground-truth correspondence. The ground-truth error, denoted by E_{grd}, is computed for each correspondence pair (b_i^S, b_i^T) by

$$\begin{aligned} E_{\text{grd}}(b_i^S, b_i^T) = \min(&|d_S(b_i^S, g_l^S) - d_T(b_i^T, g_l^T)|, \\ &|d_S(b_i^S, g_k^S) - d_T(b_i^T, g_k^T)|) \end{aligned} \tag{10}$$

where g denotes a ground-truth keypoint on the source or target, $l = \text{argmin}_m(d_S(b_i^S, g_m^S))$ and $k = \text{argmin}_m(d_T(b_i^T, g_m^T))$. Each of these measures is averaged over all the pairs of a given correspondence set and then eventually over the whole dataset. We note that Eq. (10) is used for the ground-truth error calculation of both sparse and dense correspondence results.

To analyze how much of the errors is caused by the symmetric flip issues, we define another ground-truth error measure $\tilde{E}_{\text{grd}}$ for each correspondence pair (b_i^S, b_i^T), which discards the symmetric flip errors:

$$\begin{aligned} \tilde{E}_{\text{grd}}(b_i^S, b_i^T) = \min(&|d_S(b_i^S, g_l^S) - d_T(b_i^T, g_l^T)|, \\ &|d_S(b_i^S, g_k^S) - d_T(b_i^T, g_k^T)|, \\ &|d_S(b_i^S, g_l^S) - d_T(b_i^T, \tilde{g}_l^T)|, \\ &|d_S(b_i^S, \tilde{g}_k^S) - d_T(b_i^T, g_k^T)|) \end{aligned} \tag{11}$$

where $l = \text{argmin}_m(d_S(b_i^S, g_m^S))$, $k = \text{argmin}_m(d_T(b_i^T, g_m^T))$, and $\tilde{g}_l^T$ and $\tilde{g}_k^S$ are the symmetrically flipped versions of g_l^S and g_k^T, respectively. To compute this measure, we manually marked the symmetrically flipped version of each ground-truth keypoint over all dataset models.

To make a fair comparison between our method and the baseline methods, we equalize the number of matchings for each model pair by selecting the best R correspondences resulting from each method, where R is the size of the correspondence set with the least number of pairs.

5.2. Implementation details

We normalize the coordinates of each point cloud in a given dataset such that all the points lie within the unit sphere centered at the origin. The error threshold coefficients for our base and dense correspondence algorithms are set experimentally as $\tau_1 = 2.1$ (see Section 3.1) and $\tau_2 = 1.2$ (see Section 3.2). The number of samples generated in each patch for dense correspondence estimation is fixed to $N_S = 50$. While finding the symmetric point correspondence of a point p via histogram matching in our global intrinsic symmetry method, we ignore the closest 10% of the sampled points in the candidate set. We experimentally set the symmetry error threshold coefficient as $\tau_{sym} = 2.4$ (see Section 4.1).

To calculate the diffusion distance $d_{X,t}(x, y)$ between two given points x and y at time scale t on a point cloud X, we compute the graph Laplacian directly on the point representation in a similar way as described in (Belkin and Niyogi, 2008). An important issue here is how to set time parameter t, that determines the scale. We define our diffusion distance metric as the average of diffusion distances over a set of time steps $\mathcal{T}$: $d_X(x, y) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} d_{X,t}(x, y)$. Hence the diffusion distance $d_X(x, y)$, which is independent of t, can be interpreted as the average of the lengths of all the paths existing between two surface points. Note that $d_X(x, y)$ then becomes an approximation of the commute-time distance metric (Bronstein et al., 2010), and has the advantage of bounding the time scale parameter from above (or below). In the experiments, we use the smallest $M = 60$ eigenvalues and the corresponding eigenvectors of the Laplacian matrix. The time step parameter t is incremented from 1 to 600 to compute the average diffusion distance.

Table 1
Quantitative evaluation of our sparse (base) correspondence estimation method with and without symmetry detection. (GT: Ground-truth, GT-SF: GT Error with symmetric flips ignored, SD: Symmetry Detection, Errors $\times 10^{-5}$).

Method	GT error		GT-SF Error	
	MHAD	Human	MHAD	Human
Without SD	0.43	0.32	0.09	**0.01**
With SD	**0.27**	**0.15**	**0.03**	0.02

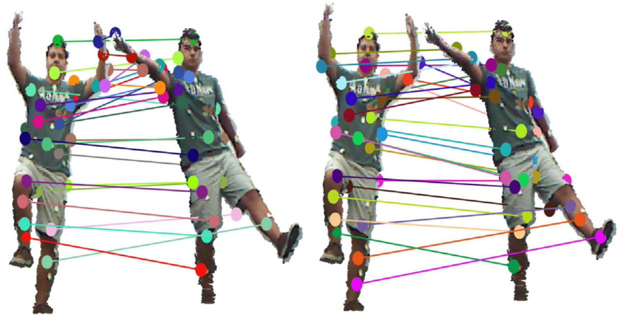


Fig. 4. Example base correspondence results without (left) and with (right) our intrinsic symmetry detection method. Symmetric flips reduced with our clustering-based symmetry detection method especially on the arms.

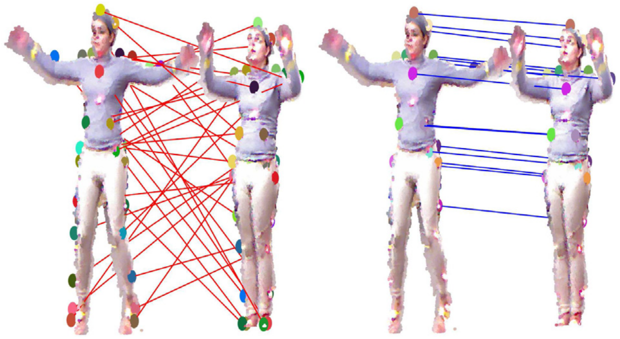


Fig. 5. Red lines: incorrect, blue lines: correct correspondences. Left: Initial random correspondences. Right: Results of our method with 100% precision.

5.3. Base correspondence results

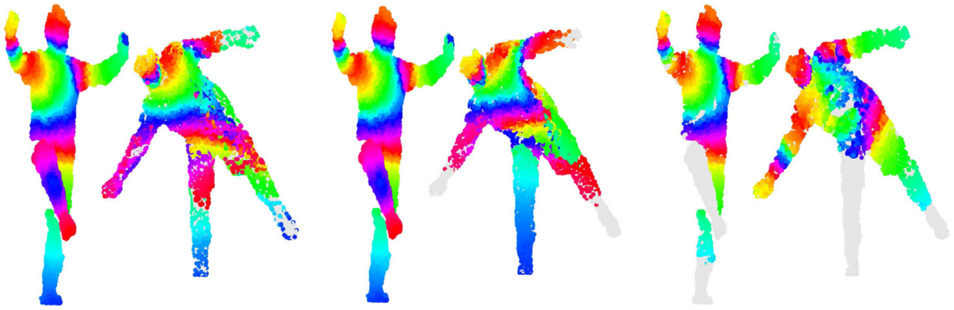
We start by evaluating the performance of our correspondence estimation method using our symmetry-aware technique described in Section 4 versus the case where the symmetry information is discarded. Table 1 shows that with our clustering-based symmetry detection method, we manage to reduce the symmetric flips on both datasets. Fig. 4 shows an example comparison with and without symmetry detection, where we observe that our symmetry-aware strategy corrects most of the symmetric flips, especially those occurring on the arms. All the correspondence results presented in the rest of this section are obtained with our symmetry detection method in use.

In Table 2, we provide the results of the three baseline methods compared to our method. We initialize our algorithm alternately with descriptor matching, CPD, PR-GLS and with random correspondences. We observe that our algorithm considerably improves the ground-truth and isometric error results of the baseline algorithms (using their output as our initial base correspondences) on the Human and MHAD datasets. The advantage of our method is pronounced especially when working with the challenging Human dataset that contains larger deformations, hence less local similarity and smaller surface overlap due to severe occlusions.

To test our method under more realistic situations, we also experimented with automatically detected keypoints, employing the ISS

Table 2
Quantitative evaluation of our sparse (base) correspondence estimation method in comparison to the baseline methods. (GT: Ground-truth, Errors ×10^{-5}).

Method	With ground truth keypoints				With detected keypoints			
	GT error		Isometric error		GT error		Isometric error	
	MHAD	Human	MHAD	Human	MHAD	Human	MHAD	Human
Descriptor Matching (DM)	3.79	7.09	7.48	6.33	5.55	6.74	5.22	6.52
CPD	0.48	4.79	4.89	6.54	2.53	3.39	4.03	6.01
PR-GLS	0.97	3.46	5.29	3.63	3.02	4.27	3.50	5.60
Our method (with DM)	0.49	0.21	0.57	0.61	1.76	1.76	0.72	0.88
Our method (with CPD)	**0.21**	**0.16**	**0.55**	**0.60**	**1.14**	**1.06**	**0.72**	0.88
Our method (with PR-GLS)	0.28	0.37	0.57	0.62	1.23	1.22	0.73	**0.88**
Our method (with Random)	1.99	0.39	0.71	0.65	2.61	1.86	0.74	0.96



(a) CPD (b) PR-GLS (c) Our method

Fig. 6. Example dense correspondence results on the Human dataset representing challenges: large deformation and gap.

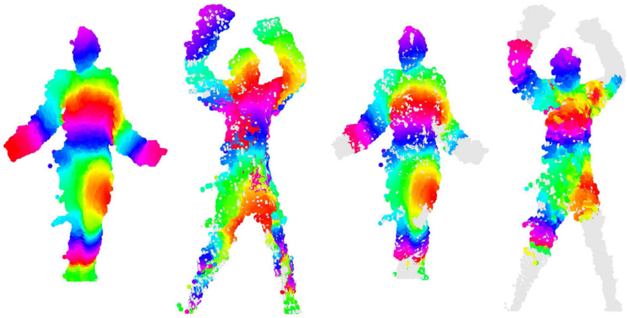
Table 3
Quantitative evaluation of our dense correspondence method in comparison to baseline methods. (Human^d: Human dataset downsampled to half resolution; Errors ×10^{-5}).

Method	Ground truth error			
	MHAD	Human	Human^d	Guo et al. (2015)
Our method	**1.25**	**1.32**	**1.18**	**0.91**
CPD	2.41	3.28	2.80	–
PR-GLS	–	–	3.79	–
Guo et al. (2015)	–	–	–	2.50

(Intrinsic Shape Signature) method (Zhong, 2009), instead of directly using the available ground-truth keypoints (see Table 2). The performances of all methods deteriorate in this case as expected, mainly due to possible inconsistencies of the detected keypoints on the source and target models. Yet our algorithm still significantly boosts the performances of all the baseline methods in terms of all error types and datasets. More interestingly, even when randomly initialized, our algorithm provides better results than the baseline methods in almost all cases, and its performance is only slightly affected by the choice of the initialization technique. Furthermore, we have cases where the initial correspondences are completely incorrect, and we still obtain up to 100% precision as in Fig. 5. For the rest of our experiments, to compute the initial base correspondence input of our algorithm, we use the CPD method which is both performant and computationally efficient.

5.4. Dense correspondence results

We demonstrate the visual performance of our coarse-to-fine dense correspondence algorithm on the bottom row of Fig. 2 on a sample pair from the MHAD dataset. This sample pair exhibits a large deformation yielding topological changes at the legs and arms. We display the results at increasing levels of detail, and represent the matched patches with the same color. The leftmost models show our reliable base correspondences, and the rightmost models are our final dense correspondence outputs. Note that the keypoints matched are detected automatically as in all our dense correspondence experiments.



(a) CPD (b) Our method

Fig. 7. Dense correspondence on a sample pair from the MHAD dataset. It represents large deformations and topological changes.

In Table 3, we provide quantitative evaluation of our dense correspondence algorithm in comparison to the baseline methods. Our algorithm is consistently better than the other methods in all cases. We note that the public code for PR-GLS fails to run on the MHAD and Human datasets due to their high resolution. Therefore, we downsampled the Human dataset (to generate Human^d) to be able to experiment with the PR-GLS method. Also, for a fairer comparison with the method of Guo et al. (2015), we used their dataset.

In Figs. 6 and 7, we visualize our dense correspondence results on sample pairs from the Human and MHAD datasets, respectively, in comparison to CPD and PR-GLS. We observe in Fig. 6 that our algorithm is capable of finding visually correct correspondences even on the parts exhibiting large deformation, such as arms, and parts with large gaps, such as legs. While the other methods suffer from symmetric flips on the legs and can even match the right leg to the right arm, our method successfully matches the left leg despite the unconnected part. Our nested iterative pruning strategy leaves the right leg unmatched since a reliable matching cannot be established. Similarly, on a challenging sample pair from the MHAD dataset in Fig. 7, our algorithm provides

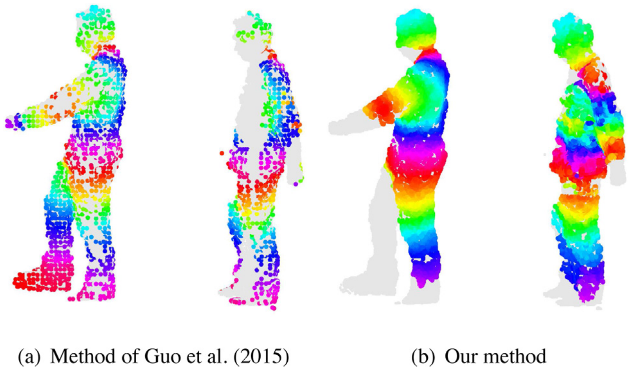


Fig. 8. Dense correspondence on a sample pair with side-views from the dataset of Guo et al. (2015) including large deformation and occlusions.

visually correct matchings on the arms exhibiting large deformation as well as on the legs undergoing significant topological change.

Fig. 8 displays our dense correspondence results on a sample pair from the (Guo et al., 2015) dataset in comparison to the method used in Guo et al. (2015). This sample pair is of particular interest since the point clouds of this pair are the side-views of the body, where the right arm and most of the right leg are severely occluded. Hence, this pair shows the benefits of our method well. First, the method of Guo et al. (2015) provides much sparser matching results compared to ours. Second, we successfully match the left arm in spite of large deformation.

6. Conclusion

We have proposed an isometric mesh-free diffusion-based method to find reliable sparse/dense correspondences between point clouds generated from partial depth data exhibiting noise, deformations, and occlusions. Our experiments show that our method provides state of the art performance on such challenging partial human body datasets, particularly on those exhibiting large deformations. We stress that our method focuses on finding as many reliable correspondences as possible, pruning whenever matching is not reliable. We have also designed a new global intrinsic symmetry detection method that addresses the symmetric flip problem inherent to isometric correspondence methods. Although our point correspondence framework is generic and can in theory be applied to any isometric shape, our symmetry detection technique is yet effective on human shapes which have global reflectional symmetry.

As future work, we plan to generalize our symmetry-aware correspondence method to non-human shapes with rigid and/or isometric deformations, that would certainly be useful for applications involving object detection and retrieval from depth images. In this respect, a possible research direction is to extend our clustering-based global symmetry detection method to also handle local intrinsic symmetries.

Acknowledgments

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grants 114E628 and 215E201. We thank Guo et al. (2015) for sharing their code and dataset with us.

References

Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S., Vincze, M., 2012. Point cloud library. IEEE Robot. Autom. Mag. 1070 (9932/12).

Belkin, M., Niyogi, P., 2008. Towards a theoretical foundation for Laplacian-based manifold methods. J. Comput. System Sci. 74 (8), 1289–1308.

Berger, M., Silva, C.T., 2012. Nonrigid matching of undersampled shapes via medial diffusion. Comput. Graph. Forum 31 (5), 1587–1596.

Billings, S.D., Boctor, E.M., Taylor, R.H., 2015. Iterative most-likely point registration (imlp): a robust algorithm for computing optimal shape alignment. PLoS One 10 (3), e0117688.

Bronstein, A.M., Bronstein, M.M., Kimmel, R., 2006. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. Proc. Natl. Acad. Sci. 103 (5), 1168–1172.

Bronstein, A.M., Bronstein, M.M., Kimmel, R., Mahmoudi, M., Sapiro, G., 2010. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. Int. J. Comput. Vis. 89 (2–3), 266–286.

Brunton, A., Wand, M., Wuhrer, S., Seidel, H.-P., Weinkauf, T., 2014. A low-dimensional representation for robust partial isometric correspondences computation. Graph. Models 76 (2), 70–85.

Chang, W., Zwicker, M., 2011. Global registration of dynamic range scans for articulated model reconstruction. ACM Trans. Graph. 30 (3), 26.

Chen, Q., Koltun, V., 2015. Robust nonrigid registration by convex optimization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2039–2047.

Chui, H., Rangarajan, A., 2003. A new point matching algorithm for non-rigid registration. Comput. Vis. Image Underst. 89 (2), 114–141.

Coifman, R.R., Lafon, S., 2006. Diffusion maps. Appl. Comput. Harmon. Anal. 21 (1), 5–30.

Dubrovina, A., Kimmel, R., 2011. Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures. Adv. Adapt. Data Anal. 3 (01n02), 203–228.

Guo, H., Zhu, D., Mordohai, P., 2015. Correspondence estimation for non-rigid point clouds with automatic part discovery. Vis. Comput. 1–14.

Horaud, R., Forbes, F., Yguel, M., Dewaele, G., Zhang, J., 2011. Rigid and articulated point registration with expectation conditional maximization. IEEE Trans. Pattern Anal. Mach. Intell. 33 (3), 587–602.

Huang, Q.-X., Adams, B., Wicke, M., Guibas, L.J., 2008. Non-rigid registration under isometric deformations. Comput. Graph. Forum 27 (5), 1449–1457.

Jian, B., Vemuri, B.C., 2011. Robust point set registration using gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 33 (8), 1633–1645.

Jiang, W., Xu, K., Cheng, Z.-Q., Zhang, H., 2013. Skeleton-based intrinsic symmetry detection on point clouds. Graph. Models 75 (4), 177–188.

Johnson, A.E., 1997. Spin-Images: A Representation for 3-D Surface Matching (Ph.D. thesis). CMU-RI-TR-97-47.

Kim, V.G., Lipman, Y., Chen, X., Funkhouser, T., 2010. Möbius transformations for global intrinsic symmetry analysis. Comput. Graph. Forum 29 (5), 1689–1700.

Kolmogorov, V., 2009. Blossom V: A new implementation of a minimum cost perfect matching algorithm. Math. Program. Comput. 1 (1), 43–67.

Kovnatsky, A., Bronstein, M.M., Bresson, X., Vandergheynst, P., 2015. Functional correspondence by matrix completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 905–914.

Küpçü, E., Yemez, Y., 2017. Reliable isometric point correspondence from depth. In: Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on. IEEE, pp. 1266–1273.

Lafon, S.S., 2004. Diffusion Maps and Geometric Harmonics (Ph.D. thesis). Yale University.

Li, H., Sumner, R.W., Pauly, M., 2008. Global correspondence optimization for non-rigid registration of depth scans. Comput. Graph. Forum 27 (5), 1421–1430.

Liao, M., Zhang, Q., Wang, H., Yang, R., Gong, M., 2009. Modeling deformable objects from a single depth camera. In: ICCV. pp. 167–174.

Lipman, Y., Chen, X., Daubechies, I., Funkhouser, T., 2010. Symmetry factored embedding and distance. In: ACM SIGGRAPH 2010 Papers. SIGGRAPH '10, ACM, New York, NY, USA, pp. 103:1–103:12.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60 (2), 91–110.

Ma, J., Zhao, J., Yuille, A.L., 2016. Non-rigid point set registration by preserving global and local structures. IEEE Trans. Image Process. 25 (1), 53–64.

Mateus, D., Horaud, R., Knossow, D., Cuzzolin, F., Boyer, E., 2008. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In: CVPR. pp. 1–8.

Mitra, N.J., Guibas, L.J., Pauly, M., 2006. Partial and approximate symmetry detection for 3D geometry. ACM Trans. Graph. 25 (3), 560–568.

Mitra, N.J., Pauly, M., Wand, M., Ceylan, D., 2013. Symmetry in 3d geometry: Extraction and applications. Comput. Graph. Forum 32 (6), 1–23.

Myronenko, A., Song, X., 2010. Point set registration: Coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. 32 (12), 2262–2275.

Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R., 2013. Berkeley MHAD: A comprehensive multimodal human action database. In: WACV. pp. 53–60.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L., 2012. Functional maps: a flexible representation of maps between shapes. ACM Trans. Graph. 31 (4), 30.

Ovsjanikov, M., Mérigot, Q., Mémoli, F., Guibas, L., 2010. One point isometric matching with the heat kernel. Comput. Graph. Forum 29 (5), 1555–1564.

Ovsjanikov, M., Sun, J., Guibas, L., 2008. Global intrinsic symmetries of shapes. Comput. Graph. Forum 27 (5), 1341–1348.

Pajdla, T., Van Gool, L., 1995. Matching of 3-d curves using semi-differential invariants. In: ICCV. p. 390.

Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., 2007. Symmetries of non-rigid shapes. In: Workshop on Nonrigid Registration and Tracking (NRTL).

Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., 2010a. Full and partial symmetries of non-rigid shapes. Int. J. Comput. Vis. 89 (1), 18–39.

Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Sapiro, G., 2010. Diffusion symmetries of non-rigid shapes. In: Proc. 3DPVT, vol. 2.

Rodola, E., Bronstein, A.M., Albarelli, A., Bergamasco, F., Torsello, A., 2012. A game-theoretic approach to deformable shape matching. In: CVPR. pp. 182–189.

Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D., 2017. Partial functional correspondence. Comput. Graph. Forum 36 (1), 222–236.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm. In: 3DIM. pp. 145–152.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: ICRA. pp. 3212–3217.

Rusu, R.B., Cousins, S., 2011. 3D Is here: point cloud library (PCL). In: IEEE International Conference on Robotics and Automation. pp. 1–4, http://www.pointclouds.org.

Sahillioğlu, Y., Yemez, Y., 2011. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. Comput. Graph. Forum 30 (5), 1461–1470.

Sahillioğlu, Y., Yemez, Y., 2012. Minimum-distortion isometric shape correspondence using EM algorithm. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2203–2215.

Sahillioğlu, Y., Yemez, Y., 2013. Coarse-to-fine isometric shape correspondence by tracking symmetric flips. Comput. Graph. Forum 32 (1), 177–189.

Sahillioğlu, Y., Yemez, Y., 2014. Partial 3-d correspondence from shape extremities. Comput. Graph. Forum 33 (6), 63–76.

Sharma, A., Horaud, R., Cech, J., Boyer, E., 2011. Topologically-robust 3D shape matching based on diffusion geometry and seed growing. In: CVPR. pp. 2481–2488.

Sipiran, I., Gregor, R., Schreck, T., 2014. Approximate symmetry detection in partial 3D meshes. Comput. Graph. Forum 33 (7), 131–140.

Tam, G.K., Cheng, Z.-Q., Lai, Y.-K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.-F., Rosin, P.L., 2013. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. IEEE Trans. Vis. Comput. Graph. 19 (7), 1199–1217.

Teleimmersion Lab, Berkeley Multimodal Human Action Database (MHAD). University of California, Berkeley. http://tele-immersion.citris-uc.org.

Tevs, A., Bokeloh, M., Wand, M., Schilling, A., Seidel, H.-P., 2009. Isometric registration of ambiguous and partial data. In: CVPR. pp. 1185–1192.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique signatures of histograms for local surface description. In: ECCV. pp. 356–369.

Van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D., 2011. A survey on shape correspondence. Comput. Graph. Forum 30 (6), 1681–1707.

Wang, C., Bronstein, M.M., Bronstein, A.M., Paragios, N., 2011. Discrete minimum distortion correspondence problems for non-rigid shape matching. In: SSVM. pp. 580–591.

Wang, H., Simari, P., Su, Z., Zhang, H., 2014. Spectral global intrinsic symmetry invariant functions. In: Proceedings of Graphics Interface 2014. Canadian Information Processing Society, pp. 209–215.

Wei, L., Huang, Q., Ceylan, D., Vouga, E., Li, H., 2016. Dense human body correspondences using convolutional networks. In: CVPR. pp. 1544–1553.

Xu, K., Zhang, H., Jiang, W., Dyer, R., Cheng, Z., Liu, L., Chen, B., 2012. Multi-scale partial intrinsic symmetry detection. ACM Trans. Graph. 31 (6), 181.

Xu, K., Zhang, H., Tagliasacchi, A., Liu, L., Li, G., Meng, M., Xiong, Y., 2009. Partial intrinsic reflectional symmetry of 3D shapes. ACM Trans. Graph. 28 (5), 138.

Yoshiyasu, Y., Yoshida, E., Guibas, L., 2016. Symmetry aware embedding for shape correspondence. Comput. Graph. 60, 9–22.

Yoshiyasu, Y., Yoshida, E., Yokoi, K., Sagawa, R., 2014. Symmetry-aware nonrigid matching of incomplete 3d surfaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4193–4200.

Zhang, Z., Yin, K., Foong, K.W., 2013. Symmetry robust descriptor for non-rigid surface matching. Comput. Graph. Forum 32 (7), 355–362.

Zhong, Y., 2009. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: ICCV Workshops, pp. 689–696.

Emel Küpçü received her Ph.D. from Koç University Computer Science and Engineering. She obtained her Master's degree from Brown University. She also worked as a research assistant at Marmara University for 5 years. Her current research is on computer vision and graphics applications employing noisy and occluded 3D point clouds captured with commodity depth sensors.

Yücel Yemez received his B.S. degree from Middle East Technical University, Ankara, Turkey, in 1989, and his M.S. and Ph.D. degrees from Boğaziçi University, İstanbul, Turkey, respectively in 1992 and 1997, all in Electrical Engineering. From 1997 to 2000, he was a postdoctoral researcher in the Image and Signal Processing Department of Télécom ParisTech (formerly Ecole Nationale Supérieure des Télécommunications), France. He is currently a professor of the Computer Engineering Department at Koç University, İstanbul, and an associate editor of the Graphical Models Journal, Elsevier. His current research is focused on various fields of computer vision, graphics and machine learning.