

Correspondence Estimation for Non-rigid Point Clouds with Automatic Part Discovery

Hao Guo · Dehai Zhu · Philippos Mordohai

Received: date / Accepted: date

Abstract We propose an approach for estimating non-rigid correspondences between two shapes that can handle articulation and deformation of the surfaces to be matched. It operates on open or closed surfaces represented by point clouds, and, therefore, it can be applied on other representations that can be converted into point clouds. Our method is capable of automatically discovering the articulated parts of the surface without requiring knowledge of the topology or the number of rigid parts. Processing begins by estimating potential sparse correspondences between the source and the target surface. These are used to align the largest corresponding parts of the two surfaces. Fragments of the surface that are not consistent with this alignment generate part hypotheses on which the algorithm is applied recursively. We present qualitative and quantitative results on four datasets comprising open and closed surfaces.

Keywords Non-rigid correspondence · point clouds · shape matching

1 Introduction

Rigid shape correspondence estimation is a well-understood problem with several mature algorithms, mostly variants of the Iterative Closest Point algorithm [3, 9, 25], that are effective in many practical scenarios. Non-rigid shape correspon-

Hao Guo
China Agricultural University, China
E-mail: guohaolys@cau.edu.cn

Dehai Zhu
China Agricultural University, China
E-mail: zhudehai@263.net

Philippos Mordohai
Stevens Institute of Technology, USA
E-mail: mordohai@cs.stevens.edu

dence, on the other hand, is intrinsically a much harder problem and has only received the attention of the research community in the past few years. A diverse set of approaches that make different assumptions about the inputs and the desired transformations between two or more shapes have achieved encouraging results. We briefly review some of the most relevant methods in the next section. One way to categorize them is based on the assumptions they make on the input shapes. These assumptions range from noise-free manifold meshes with genus zero to entirely unorganized point clouds under severe occlusions.

Our approach falls towards the latter end of the above spectrum. It aims at computing dense point-wise correspondences between point clouds that are assumed to come from the same non-rigid shape. We explicitly model articulation as the dominant source of non-rigidity, while deformation is handled by allowing correspondences to vary smoothly locally during the final densification stage. Due to this assumption, our method cannot handle shapes that deform without discernable rigid parts, such as a piece of cloth or dough being manipulated. The proposed algorithm has been designed to work on unorganized point clouds and thus does not rely on topology information. The inputs may be degraded due to occlusion or self-occlusion and as result may correspond only partially. We do not assume a part in whole relationship between the source and target point clouds. That is, both shapes can have parts that are not observed in the other shape.

According to the classification of [36], our method is extrinsic, in the sense that it aims at estimating transformations in Euclidean space without embedding the inputs into canonical forms. On the other hand, we use (approximate) geodesic distance to measure distances between points in the same point cloud, under the assumption that the shapes are locally isometric.

Addressing non-rigid shape correspondence as a combinatorial matching problem between two sets of vertices leads to a prohibitively large number of solutions that have to be evaluated. To reduce the search space researchers have taken different approaches. Many of these approaches, including ours, rely on detecting reliable sparse features and matching them across the two input shapes. These sparse correspondences are used to hypothesize potential correspondences between the two shapes. In the initial alignment stage, we use the sparse correspondences to detect the largest subsets of points in the input point clouds that are approximately consistent with a single rigid transformation. The main novelty of our method is that points that are not consistent with this transformation, due to deformation, articulation or occlusion, are clustered and the clusters are passed as inputs to the initial alignment stage recursively to discover alignments between parts. Special care is taken to ensure that connectivity at the joints is preserved. Each of these clusters is considered a *part* by our algorithm. Clearly, if an articulation is not observed in either instance of the shape to be matched, the relevant parts and joints are not detected. Joints can only be detected by our algorithm if articulation is observed in the input shapes. Finally, starting from the most reliable correspondences, a propagation technique generates dense correspondences for all points.

In Section 8, we present results on four diverse datasets. The SCAPE data [2] are in the form of complete, watertight meshes, which are trivially converted to point clouds by keeping only the vertices before using them as inputs to our method. The SHREC 2011 data [4] test correspondence between complete shapes undergoing isometric deformations as well as complete to partial shape correspondence. One of the reasons for using these data is that they enable comparisons to recently published methods. On the other hand, we also use the shoulder scans of the University of Washington [1] and a set of Kinect range scans we collected. The two latter datasets comprise open surfaces with boundaries that are inconsistent across views. The current implementation of our approach is able to successfully estimate correspondences in all tested scenarios, despite large deformations in all datasets and large occlusions in most of them.

2 Related Work

In this section, we review approaches that are closely related to ours. For a broader perspective we refer readers to recent, comprehensive surveys on 3D shape correspondence, including non-rigid registration, [36, 38]. Our method operates on point clouds without requiring topology information, but we include relevant methods that require meshes as inputs in this brief survey.

We begin with approaches that are capable of processing point cloud data. Brown and Rusinkiewicz [6] address multi-

view, non-rigid registration of point clouds in the presence of small deformations and distortions, but their approach cannot handle large motions such as those due to articulation.

Mitra et al. [21] address symmetry detection with an approach that can also be used for non-rigid shape correspondence. Corresponding rigid parts in two deformed instances of the same shape can be detected by finding significant clusters in transformation space based on a Hough transform proposed earlier by the same authors [20]. Each cluster corresponds to a rigid part that has undergone a geometric transformation. The method of Huang et al. [11] is similar to ours, since we also initially seek sparse correspondences and detect rigid clusters. (Our approach, however, does not model large deformation within the clusters. Instead, in our approach clusters would have to be decomposed in parts to be matched.) Processing in [11] starts by detecting feature correspondences on the input point clouds and progressively merging them to form rigid parts, allowing the method to handle large articulation. The key differences include the use of feature descriptors based on the principal curvatures of fitted patches, the use of spectral matching to find the largest set of consistent feature correspondences and the way clusters are merged as long as the registration error for each merged cluster remains below a threshold. We experimented with spectral matching for finding large sets of consistent correspondences, but RANSAC (RANdom SAmple Consensus) proved to be a superior alternative. Finally, we propose a novel way of detecting new clusters (parts) while explicitly preserving part connectivity at the joints. This technique removes almost entirely the symmetric flip problem, which is discussed in detail in [28].

Li et al. [17] address non-rigid registration of range scans as a non-linear optimization problem that jointly considers point correspondences, correspondence reliability estimation and detection of non-overlapping areas. The input range scans are converted to meshes and a piece-wise affine model is used to represent non-rigid deformations after the two shapes have been rigidly aligned. Unlike [11] and our approach, a single rigid transformation is computed to align the shapes.

Chang and Zwicker [8] address range scan registration using a linear skinning model. They convert the point cloud into a grid-based representation and use Expectation Maximization (EM) to optimize registration error and the assignment of points to parts, which is solved as a discrete labeling problem initially and then refined to produce soft memberships of grid cells to parts. An additional term penalizes inconsistencies at the estimated joints. Spin images [13] are used to detect initial point correspondences and clustering is used to initialize the parts. Quantitative results are not presented, however.

Tevs et al. [37] present a method that is similar to ours in that it operates on point clouds and uses the k -nearest neighbor graph to approximate local connectivity and geodesic distances. A novel RANSAC algorithm with importance sampling is used to detect the most likely isometric deformation between the two shapes despite potential topological noise. The emphasis is on augmenting the set of feature correspondences by analyzing the consistency of geodesic distances of each potential correspondence to a large number of previously detected correspondences. Qualitative results are shown on multiple datasets, but with limited amounts of motion between the point clouds to be matched.

A somewhat different approach was presented by Ma et al. [19] who cast correspondence estimation between two sets of points as the estimation of a vector field representing the motion from the source to the target shape. EM is used to distinguish between inlying and outlying vectors among a large set of putative correspondences. Fitting of parametric and non-parametric models on 2D and 3D data is shown. Our algorithm is also faced with a large fraction of outliers in the set of putative correspondences and tackles the problem by establishing consensus for locally rigid motions without higher-order models, such as those imposed by the epipolar geometry.

We now turn our attention to methods that require topology information (meshes) focusing on those that can handle large articulations. This form of the non-rigid correspondence problem was addressed by SHREC (SHape REtrieval Contest), in particular by the 2010 [5] and 2011 [4] instances of the contest. We use data from SHREC 2011 for some of our experiments. Starck and Hilton [34] address non-rigid shape correspondence by introducing local feature descriptors that are invariant to isometric deformations. Dense correspondences are obtained using a Markov random field defined on the edges and nodes of the mesh and guided by the feature correspondences. Zhang et al. [40] define a cost function based on the distortion of the deformed meshes and use shape extremities as features to generate correspondence hypotheses. An efficient mechanism for selecting the most likely solutions is also proposed. Lipman and Funkhouser [18] introduce the Möbius Voting algorithm for discovering point correspondences between approximately or partially approximately isometric surfaces. It uses the Möbius transformations defined by triplets of potential point correspondences to compute mappings between the input surfaces. Each mapping casts a vote for point correspondences that are consistent with it and the probability of a given correspondence can be estimated from the number of votes it receives. Sharma et al. [32] initially determine seed matches between two meshes using local heat-kernel based descriptors. The seeds are grown to progressively densify the correspondences and an EM algorithm is used to generate the final dense correspondences. The approach is designed to be

robust against changes in topology. Zeng et al. [39] address 3D surface tracking using triplets of correspondences using a Markov Random Field for regularization. Unlike previous work on conformal matching [18], this approach can handle manifolds with boundaries, even if the boundaries are inconsistent, due to occlusion for example. Rodola et al. [24] pose shape correspondence estimation as a quadratic assignment problem and solve it by combining spectral relaxation, which produces dense matches, with game-theoretic matching, which produces accurate but sparse matches.

Kim et al. [16] propose the Blended Intrinsic Maps (BIM) algorithm which operates on two closed meshes and estimates a mapping between them by blending multiple conformal maps. Each conformal map models a part of the surface but fails when the transformation is not globally conformal. Conformal map hypotheses are generated by exhaustively searching all triplets of a small set of feature points, which are detected as the maxima of the average geodesic distance function. Sahillioglu and Yemez [30] present the rank-and-vote-and-combine (RAVAC) algorithm for estimating partial correspondences between meshes that have common, but potentially also dissimilar, parts. RAVAC begins by detecting shape extremities which are used as features to hypothesize partial isometric mappings between the shapes. Unlike BIM, RAVAC poses no restrictions on topology. Sipiran and Bustos [33] propose a hierarchical representation of meshes via the decomposition tree as well as a hierarchical matching algorithm. The main assumption is that, when two near-isometric shapes are decomposed hierarchically, their constituent regions can be matched. This is similar to our method, but we do not a priori decompose the input shapes independently of each other.

Several authors have investigated the case where there are more than two shapes to be matched. Recent methods that operate on collections of 3D shapes from the same class include that of Huang and Guibas [12]. It solves a semi-definite programming problem to enforce the cycle-consistency constraint, which states that shape primitives must be mapped to themselves after a cycle of pairwise transformations has been traversed. Budd et al. [7] introduce the shape similarity tree which is the minimum spanning tree in shape similarity space. Sahillioglu and Yelmez [29] use shape extremities as matching primitives and dynamic programming to infer correspondences that minimize the isometric distortion over all pairwise shape maps without requiring initial pairwise correspondences. Rodola et al. [23] address class-specific, non-rigid shape correspondence by training a random forest on a set of exemplars from each class. A different approach based on classification was recently proposed by Kanezaki et al. [14]. Classifiers are trained, similarly to metric learning, to predict whether a reference shape and an unknown shape are from the same class or not. Kim et al. [15] address a similar problem in large collections of shapes based on the

notion of fuzzy correspondences to capture important similarity relations within each class of shapes, despite its potential heterogeneity. By examining an entire collection instead of model pairs, salient point-wise correspondences can be discovered and used for shape retrieval. Our method in its current form focuses on the pairwise shape correspondence problem.

A key difference between methods that operate on point clouds and those that operate on meshes is that the latter often use shape extremities as features. While shape extremities, such as fingertips, are very useful features and allow these methods to only consider small numbers of features, they typically cannot be reliably detected on partial 3D scans. In the presence of occlusion, occluding and occluded boundaries can appear as extremities, while true extremities may be hard to detect. Due to the self-imposed requirement that our method work on partial point clouds, we use feature descriptors from the point cloud processing literature. We compute these descriptors uniformly on the input point clouds, bypassing the detection stage. We use the Fast Point Feature Histogram descriptor of Rusu et al. [26] for all results shown in this paper, but it is possible that other descriptors [10, 13, 31] could have been effective.

3 Overview of the Approach

In this paper, we address the problem of estimating a dense point-wise correspondence between a *source point cloud* S_o and a *target point cloud* T_o . We begin by downsampling the input point clouds using an octree with leaf size d_r . At most one point per leaf is retained and all subsequent steps are applied on the downsampled point clouds denoted by $S_d = \{s_i\}$ and $T_d = \{t_j\}$. Typically, $5r$ is used as the leaf size, where r is the resolution of the point cloud defined as the average distance between neighboring points.

Starting from the downsampled point clouds, normals are estimated [22] and local descriptors are computed [26]. The local descriptors are used to establish potential *sparse correspondences*, which may contain several errors initially. Errors are reduced by requiring that the sparse correspondences be *reciprocal*, i.e. if the most similar point to s_i is t_j , we require that the most similar point to t_j is s_i . Otherwise, the correspondence is rejected.

We, then, use RANSAC to estimate a rigid alignment that is supported by the largest number of points. The overlapping area is designated as the *largest rigid part (LRP)*. Points that are not consistent with the LRP are clustered separately in each point cloud. The result is a set of clusters in the source and target point clouds that have not been matched yet. We seek correspondences for these clusters taking into account the way they are attached to previously registered parts. This procedure is recursively applied to all clusters, which can be subdivided into multiple rigid parts,

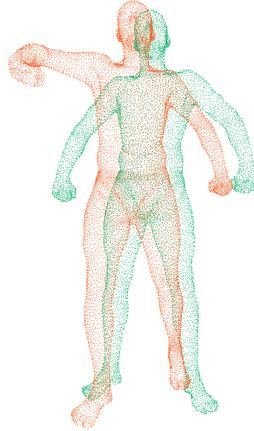


Fig. 1 Raw data: two point clouds from SCAPE

until the remaining unregistered points form very small clusters. Finally, correspondences are propagated to all unmatched points from their neighbors.

Sections 4 through 7 provide more details on each step illustrated on an alignment example from SCAPE [2]. Results on four datasets, namely SCAPE, SHREC 2011, the shoulder scans of the University of Washington [1] and a Kinect dataset collected for this paper, are shown in Section 8.

4 Sparse Correspondences

The input to this stage is two point clouds and the desired output is a set of sparse correspondences which will be used to guide subsequent steps. We expect that some of the correspondences generated at this stage will be wrong. These errors are rejected by the RANSAC-based alignments that are estimated in Sections 5 and 6. Figure 1 shows the source and target point cloud for two scans from SCAPE [2].

After downsampling, un-oriented normals are estimated for the point clouds using the technique of Mitra et al. [22]. Then, Fast Point Feature Histogram (FPFH) descriptors are computed according to Rusu et al. [26]. The FPFH is a local weighted average of the Simplified Point Feature Histograms (SPFH) computed at each point p of the point cloud. Specifically, for each neighbor p_i of p in a sphere of radius equal to $10r$, a Darboux frame uvn is defined as follows. Let n and n_i be the estimated normals at p and p_i and let n_m be the normal that forms the smallest angle with the line connecting p and p_i , that is $n_m = n$, if $n \cdot (p_i - p) > n_i \cdot (p_i - p)$ or $n_m = n_i$, otherwise. Without loss of generality, we will assume that $n_m = n$ here. Then, $u = n_i$, $v = (p_i - p) \times u$ and $w = u \times v$ and the following angles are computed and histogrammed:

$$\begin{aligned}\alpha &= u \cdot n \\ \phi &= u \cdot (p_i - p) / \|p_i - p\| \\ \theta &= \arctan(w \cdot n_i, u \cdot n_i)\end{aligned}$$

Each of these angles is histogrammed independently in 11-bins resulting in a 33-D histogram, which is the SPFH. The FPFH descriptor for a point p in point cloud P is defined as:

$$f(p) = SPFH(p) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_k} \cdot SPFH(p_k), \quad (1)$$

where $p_k \in N_k(p, P)$ are the k -nearest neighbors of p , $SPFH$ is the Simplified Point Feature Histogram (SPFH) at p and w_k is the Euclidean distance between p and p_k . The SPFH is a local descriptor that aggregates pairwise geometric relationships between the reference point and its nearest neighbors.

Given these descriptors, we find the most similar point in the target point cloud for every source point using the Euclidean distance in the 33-D descriptor space $d_f(s_i, t_j) = \|f(s_i) - f(t_j)\|$. The resulting initial set of correspondences is denoted by $K = \{(s_i, t_i)\}$ with $s_i \in S_d \subset S_o$ and $t_i \in T_d \subset T_o$.

Correspondences are sought between source and target points in both directions, source to target and target to source. Reciprocal correspondences that have been selected in both directions are retained, while all other correspondences are discarded. The set of reciprocal correspondences is denoted by $K_f = \{(s_k, t_k)\}$. Figure 2 shows a visualization of the reciprocal sparse correspondences that are passed to the next stages of processing.

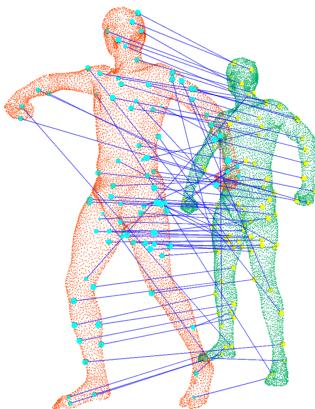


Fig. 2 Reciprocal sparse correspondences for the data of Fig. 1

5 Initial Alignment

Based on the reciprocal correspondences from the previous stage, we compute an initial rigid alignment between the two point clouds using RANSAC. Each minimal hypothesis gives rise to a rigid transformation, which is evaluated according to the number of inliers. Inliers here are reciprocal correspondences that are within ε after the transformation is applied. Typical values for the parameters of RANSAC are: 10,000 iterations and $\varepsilon = 10r$, where r is the resolution (average distance between neighboring points) of the original point cloud before downsampling. The transformation supported by the highest number of inliers is chosen and it is refined using least squares fitting on all inliers. See Fig. 3 for an example.

$$X_0 = \arg \min_{R, t} \sum_{i=1}^{|K_r|} \|(Rs_i + c) - t_i\|^2 \quad (2)$$

where X_0 is the rigid transformation of the largest rigid part (LRP) and K_r is the set of all inliers after RANSAC and $(s_i, t_i) \in K_r$. Note that in order to avoid switching between homogeneous and inhomogeneous coordinates when applying rigid transformations, we will treat X_0 as a function that operates on points and applies a rotation R_0 followed a translation c_0 to them.

In order to identify all points in the LRP, we first transform the source point cloud S_d onto the coordinate system of the target using X_0 . Thus, we obtain $S_l = X_0(S_d)$. We then cluster points of the transformed source that are within $3r$ of points of the target point cloud using region growing to obtain a set of clusters in regions where the two point clouds overlap.

$$G_s = \{C_{RGS1}, C_{RGS2}, \dots, C_{RGSN}\}. \quad (3)$$

Each of these clusters is grown from a seed s_i , an unclustered point, in S_l . A new point s_k is added to the cluster under two

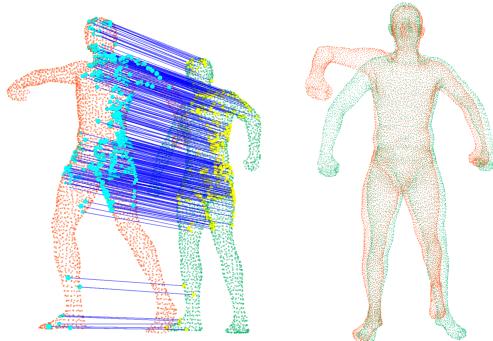


Fig. 3 Left: correspondences consistent with the alignment of the LRPs. Right: initial alignment of LRP.

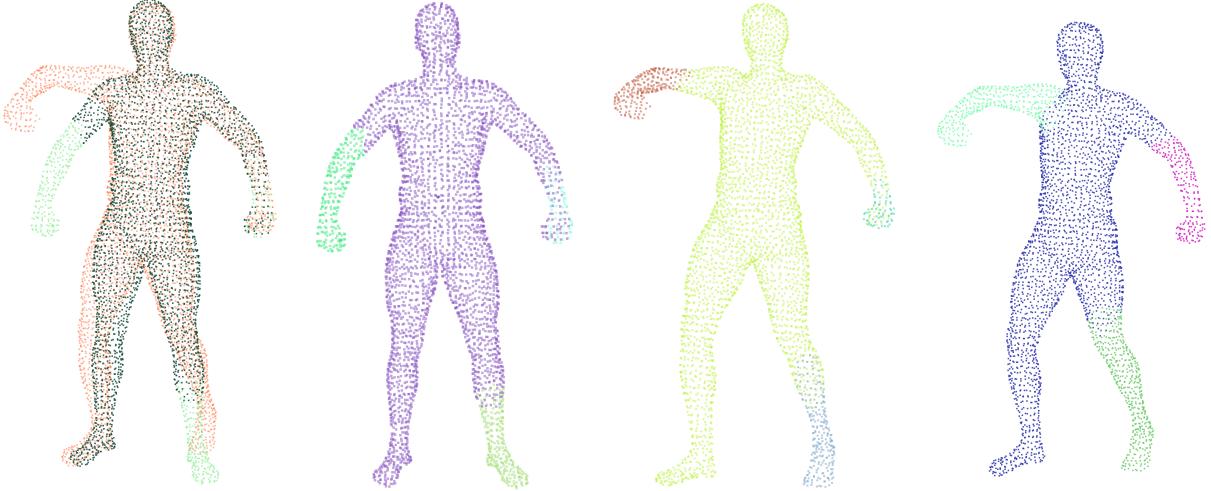


Fig. 4 Left to right: segmentation of source and target according to the corresponding LRP; segmentation of the source; segmentation of the target; segmentation of the target after extension of the unmatched parts. Note that the rightmost figure shows the extension of the clusters that is used during part discovery. The extension of the LRP is in the opposite direction.

conditions: if it is the single nearest neighbor of a point s_m which is already in the cluster; and if it is sufficiently close to a point t_j in T_d . The latter condition is implemented by testing whether $d_e(s_k - t_j) \leq 3r$, where d_e is the Euclidean distance. We use very tight neighborhoods in this step to obtain compact clusters. Then, the largest overlapping cluster in the transformed source point cloud S_l , in terms of number of points, is denoted by C_{sl} .

The same procedure is applied on the target point cloud to obtain a set of clusters G_t and the largest overlapping cluster C_{tl} . We verify that C_{sl} and C_{tl} correspond to each other, or we pick the two corresponding clusters with the maximum total number of points if that is not the case. The selected corresponding overlapping clusters define the LRP in both point clouds.

C_{tl} in the target point cloud is extended by adding the neighbors of the already included points. Since we seek neighbors in the same point cloud, we use geodesic distances in this search, approximated by graph distances in the 20-nearest neighbor graph. The geodesic radius used for all extension steps, here and in the following section, is $50r$. The goal is to ensure that the correspondences for all points in the source point cloud are included in C_{tl} . Using the extended LRP in the target, we recompute the correspondences with the LRP of the source by finding the nearest point in the target for each point in the source. We further require that the FPFH descriptors of corresponding points are within a distance threshold θ_f . Points that fail the descriptor similarity test are removed from the set of correspondences. This gives us a set of reliable correspondences that we refer to as *kernel correspondences* following Huang et al. [11]. These correspondences can be seen at the left part of Fig. 3, while the

right part shows the estimated alignment of the LRP of the point clouds. The LRP can now be segmented in both point clouds, as shown in the three leftmost parts of Fig. 4.

6 Part Discovery

In this section, we attempt to discover parts, if they exist, in the point clouds. A part is a connected component, above a minimum size, that has not been matched to the other point cloud yet. Parts are formed by growing connected components starting from unmatched points. We seek correspondences for unmatched parts that are connected to matched parts and the process is repeated recursively. After the initial alignment, parts that are connected to the LRP are matched first, e.g. the thighs, while in the next step, the focus shifts to parts connected to the parts which were just matched, e.g. the lower legs.

We begin by forming clusters using region growing, connecting points not in the largest rigid part with their k -nearest neighbors ($k = 20$). We use S_l and T_l to denote the inputs at level l . These are the entire downsampled point clouds at level 0, but they can also be other corresponding clusters as the initial alignment and part discovery steps are applied recursively. C_{sl} and C_{tl} are the LRPs of the input point clouds at level l . Points in the LRPs are excluded from this clustering step. Let $\{C_{s(l+1)1}, C_{s(l+1)2}, \dots, C_{s(l+1)K}\}$ denote the clusters in $S_l - C_{sl}$ and $\{C_{t(l+1)1}, C_{t(l+1)2}, \dots, C_{t(l+1)K}\}$ denote the clusters in $T_l - C_{tl}$, at level $l + 1$ after dropping clusters with less than a minimum number of points.

Clusters in the target are extended as described in Section 5 to ensure that all source points can be matched. The rightmost subfigure of Fig. 4 shows the extension of the clus-

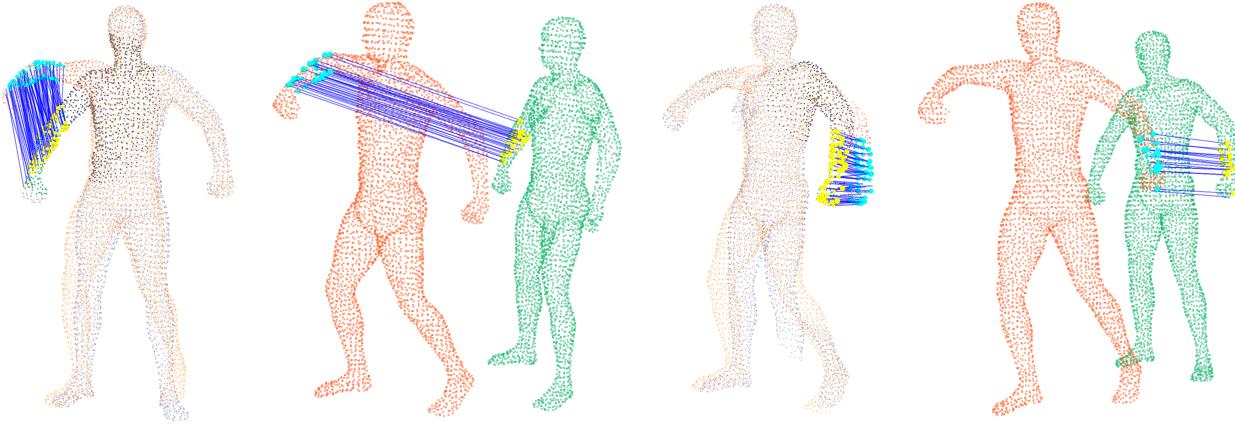


Fig. 5 First and third subfigures: correspondences found by propagation from the LRP and used for estimating the transformation of the first and third clusters. Second and fourth subfigures: new correspondences detected for these clusters after aligning the corresponding clusters. The second cluster is the left leg (not shown here).

ters (compare with the third subfigure). The extension is now done from the clusters towards the LRP. The same geodesic radius ($50r$) is also used here.

$$EC_{t(l+1)k} = \{G_r(t_i, T_d) | t_i \in C_{t(l+1)k}\}, \quad (4)$$

where $G_r(t_i, T_d)$ is the geodesic neighborhood of radius r in the downsampled target point cloud T_d .

For every cluster $C_{s(l+1)k}$ in the source point cloud we find the region $A_{s(l+1)k}$ in the cluster which is connected with the largest rigid part C_{sl} :

$$A_{s(l+1)k} = \{s_i | s_i \in C_{s(l+1)k}, G_r(s_i, S_d) \cap C_{sl} \neq \emptyset\}, \quad (5)$$

where $G_r(s_i, S_d)$ is the geodesic neighborhood of radius s_i in the downsampled source point cloud. We also find a similar region $B_{s(l+1)k}$ in the largest rigid part C_{sl} .

$$B_{s(l+1)k} = \{s_i | s_i \in C_{sl}, G_r(s_i, S_d) \cap C_{s(l+1)k} \neq \emptyset\}. \quad (6)$$

The same process is repeated in the target point cloud. Given two overlapping regions $B_{s(l+1)k}$ and $B_{t(l+1)m}$, in the source and target point clouds respectively, we can identify which clusters correspond and avoid the symmetry problem. Symmetry leads to errors when, for example, the left arm of the source is matched to the right arm of the target. We can easily find $A_{s(l+1)k}$ and $A_{t(l+1)m}$ that correspond in the two point clouds and then estimate an alignment for the entire clusters $C_{s(l+1)k}$ and $EC_{t(l+1)m}$.

Correspondences are propagated to unmatched points in the A regions. Specifically, given a cluster $C_{s(l+1)k}$, its corresponding cluster $EC_{t(l+1)m}$, and a sample $s_i \in A_{s(l+1)k}$ that does not have a corresponding point in the target yet, we can find its corresponding point by propagating existing correspondences.

$$K_{s(l+1)k} = \{(s_i, t_i) | t_i = \arg \min_{t \in EC_{t(l+1)m}} e_{K_l}(s_i, t), s_i \in A_{s(l+1)k}\}, \quad (7)$$

where the consistency error e_K is defined as:

$$e_{K_l}(s, t) = \sum_{(s_k, t_k) \in K_l, s_k \in C_{sl}} [d_g(s, s_k) - d_g(t, t_k)]^2. \quad (8)$$

K_l indicates the kernel correspondences form level l . Note that correspondence propagation is done in geodesic neighborhoods and $d_g()$ denotes the geodesic distance between two points. Finally, we use all correspondences in $K_{s(l+1)k}$ to estimate the transformation $T_{(l+1)k}$ between $C_{s(l+1)k}$ and $EC_{t(l+1)m}$. This is done as in Section 5 for the LRP. After the source cluster has been transformed, we find new correspondences based on the same criteria used for the LRP. Specifically, we require that these correspondences are within a certain Euclidean distance θ_c after the transformation and the FPFH descriptors of corresponding points are within a certain Euclidean distance θ_f . We further require that the correspondences be reciprocal. The output is a set of kernel correspondences for the part. Figure 5 shows two examples of matching clusters and the new correspondences found in this stage.

The procedure described in this section is recursive. After a cluster has been discovered in the source point cloud, we find the potentially corresponding part in the target point cloud and pass them as inputs to the initial alignment algorithm of Section 5. Recursion terminates when an entire cluster can be matched between the two point clouds by a single rigid transformation, or when the remaining unmatched clusters contain less than a minimum number of points. This scheme for discovering and matching parts that are close to previously matched parts allows us to enforce joint consistency across the two shapes. The limitation is that parts that appear disconnected from the LRP due to occlusion or self-occlusion cannot be matched.

7 Dense Correspondences

After rigid transformations for all parts have been estimated as above, we perform a final propagation step to compute dense correspondences for all points. The input to this stage is the kernel for all clusters in the source point cloud

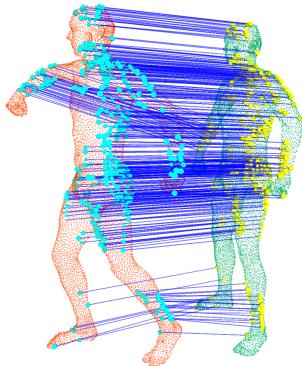
$$K_k = \{(s_i, t_i) | (s, t) \in K_l, l = 0, 1, \dots, L\}. \quad (9)$$

Based on this, propagation is done as follows: for a sample s_i in the source that does not have a correspondence yet, we seek the nearest point s_j with a correspondence (s_j, t_j) . We then assign to s_i a target point that is most consistent with the kernel correspondences according to minimum geodesic distance

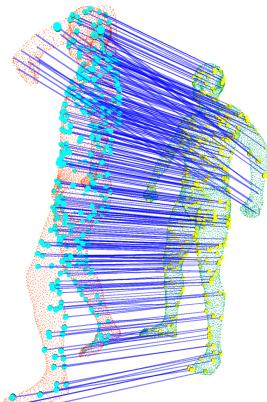
$$K_a = \{(s_i, t_i) | t_i = \arg \min_{t \in G_r(t_j, T_d)} e_{K_k}(s_i, t), s_i \in S_d\} \quad (10)$$

where the consistency error e_K is defined as:

$$e_{K_k}(s, t) = \sum_{(s_k, t_k) \in K_k} [d_g(s, s_k) - d_g(t, t_k)]^2. \quad (11)$$



(a) All kernel correspondences



(b) 10% of final dense correspondences

Fig. 6 Final results on aligning two SCAPE models

We thus obtain the final set of correspondences comprising the kernel K_k and the correspondences from the propagation K_a . See Fig. 6 for a visualization of all kernel correspondences and a sampling of the dense correspondences.

8 Experimental Results

We present qualitative and quantitative results on four diverse datasets, namely SCAPE [2], SHREC 2011 [4], the shoulder scans [1] and a Kinect dataset introduced here. All datasets are processed with constant parameter values. Specifically, the octree leaf size for downsampling d_r is set equal to $5r$, where r is the resolution of the point cloud, the radius of the neighborhood for SPFH computation is $10r$, k in (1) is set to 30, the minimum number of points for a cluster to be processed further is 20, while the thresholds on Euclidean and FPFH distance are set to $\theta_c = r$ and $\theta_f = 400$.

In accordance with the literature, we use the average of normalized geodesic distances when presenting quantitative results. For each shape, we find the largest geodesic distance between any two points and use it to normalize all other distances. In the following tables we report the averages of such distances over the relevant number of points for each experiment.

Results on SCAPE. The first dataset we validate our method on is SCAPE [2]. Specifically, we use the 12 pairs of shapes used by Sahillioglu and Yemez to validate their RAVAC algorithm [30] and compare against their results, as well as those obtained by the Möbius Voting (MV) method [18] and the blended intrinsic maps (BIM) method [16] as reported in [30]. Figures 3-6 show results on estimating correspondences between meshes 0 and 2 of the SCAPE dataset, while Fig. 7 shows results on meshes 1 and 53.

	Ave. Nr. Corresp.	D_{grd}
MV [18]	~250	0.203
RAVAC [30]	~250	0.043
Our method	260	0.015

Table 1 Average normalized geodesic distance from ground truth for sparse correspondences on the 12 pairs from SCAPE used in [30]. Results from [18, 30] are copied from the latter. The results of our method are on all kernel correspondences before the propagation of Section 7 is applied. All methods generate approximately the same number of correspondences.

We present quantitative results in Tables 1 and 2. Due to the availability of meshes for these data, we use the software of Surazhsky et al. [35] to compute geodesic distances. (It should be noted that due to the normalization, the difference between these exact geodesic distances and approximations computed using graph distances is negligible.) Table 1

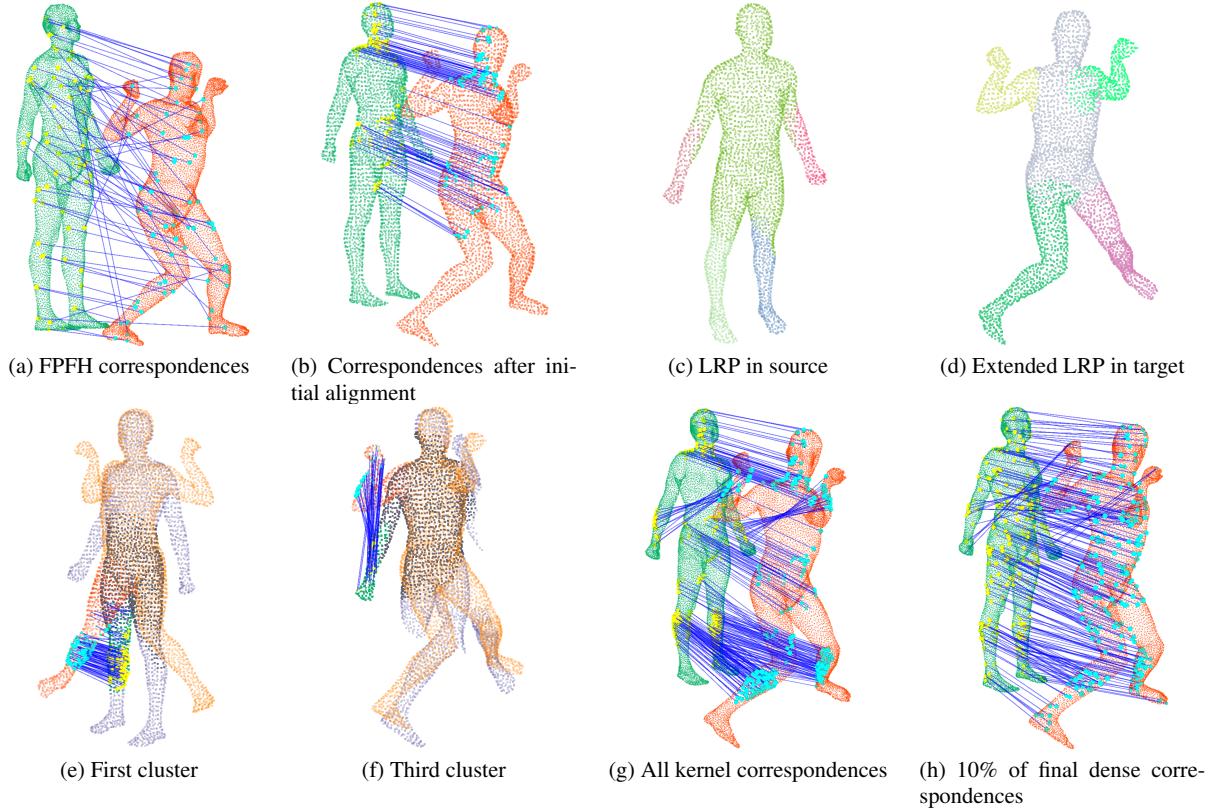


Fig. 7 Visualization of entire pipeline on models 1 and 53 from SCAPE. Only the vertices of the meshes are used as inputs to our algorithm.

contains the average distance from ground truth for sparse correspondences, the kernel for our algorithm, and Table 2 shows results on all vertices. Our method is superior to all other methods, except the dense results of BIM. BIM has several advantages over our approach on these data since it exploits topology information from the mesh and assumes that the shape has genus zero. It also requires a very small number of repeatable features to be detected, which is not always possible on point clouds or open surfaces. BIM is not applicable on the following experiments, except those on matching SHREC 2011 shapes under isometric deformations.

	D_{grd}
BIM [16]	0.042
RAVAC [30]	0.051
Our method	0.045

Table 2 Average normalized geodesic distance from ground truth for dense correspondences on the 12 pairs from SCAPE used in [30]. Results from [16,30] are copied from the latter. The results of our method are after the propagation of Section 7.

Results on SHREC 2011. We performed two types of experiments on the SHREC 2011 dataset [4]. In the first set of

experiments we used the five instances of the human model that have undergone isometric deformations to form ten pairs, for which we estimate correspondences. We call these experiments *iso-iso* matching. In the second set of experiments we form pairs of shapes by taking an instance that has undergone an isometric deformation and one that has undergone an isometric deformation, but has also been cropped. We refer to these experiments as *iso-part* matching. As above, we only keep the vertices of the models and discard all topology information. Figure 8 shows representative results for both cases.

	Test	Ave. Nr. Corresp.	D_{grd}
Our method: kernel	iso-iso	346	0.015
Our method: dense	iso-iso	4345	0.047
Our method: kernel	iso-part	276	0.014
Our method: dense	iso-part	1581	0.023

Table 3 Average normalized approximate geodesic distance from ground truth for kernel and dense correspondences on SHREC 2011

Table 3 summarizes the geodesic distance errors between manually clicked ground truth and the estimated correspondences. It includes errors for both kernel and dense matches in the two types of experiments. Errors on kernel matches

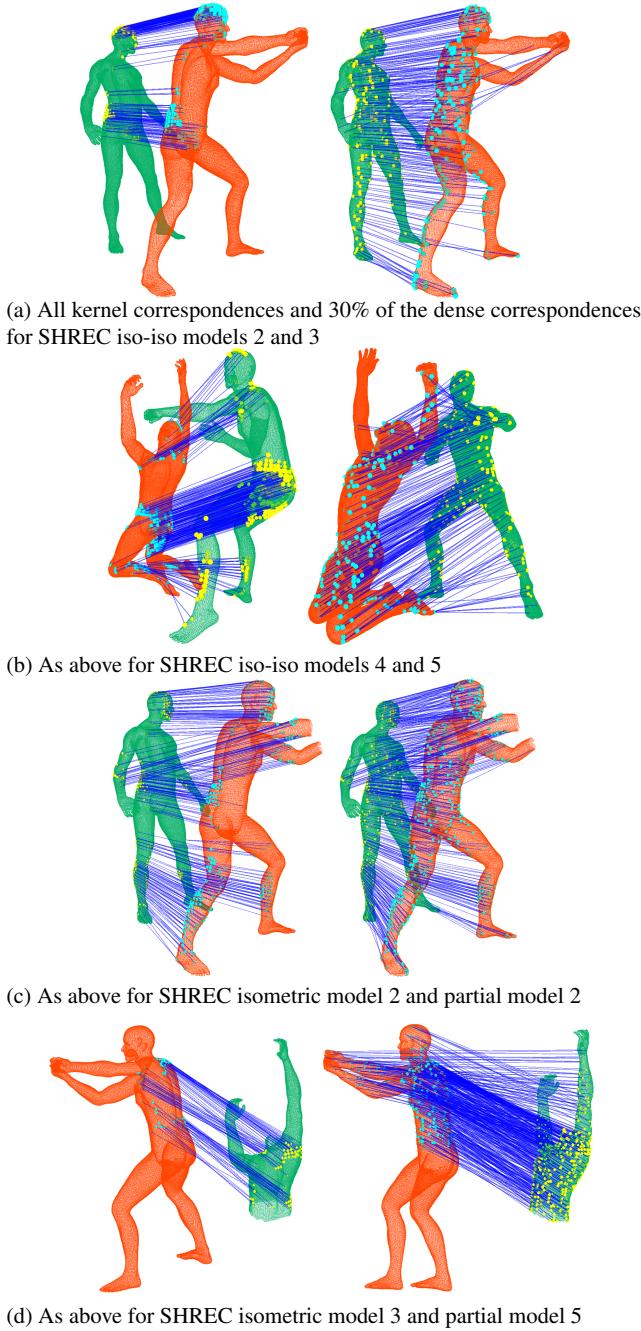


Fig. 8 Estimated correspondences on iso-iso and iso-part pairs of models from SHREC 2011

are similar, while errors on dense matches are lower in *iso-part* matching tighter alignment is possible for the un-occluded parts. Sahillioglu and Yemez [30] published results on the same matching scenarios for their method, as well as for Möbius Voting (MV) [18]. The results are not directly comparable, since [30] reports error at the extremities and for the top-5 matches while our method relies on consensus and requires larger numbers of correspondences. In the iso-iso experiments, MV had an error of 0.053 on extremities and

0.002 on the top-5 matches, while RAVAC had errors of 0.003 and 0.044 on extremities and top-5 matches, respectively. Each method is outstanding according to its own criterion. Our method results in lower errors than both RAVAC and MV if the secondary criterion is considered in each case despite the fact that we consider hundreds of correspondences. The only error reported for the iso-part scenario is 0.049 for RAVAC on extremities. Our method achieves much lower error even on dense matches.

Results on the shoulder data. Here we present results on the shoulder scans from the University of Washington [1]. Figure 9 shows alignment results on these data while Table 4 contains quantitative results. For these data we approximate the geodesic distances to the ground truth by graph distances in 20-nearest neighbor graphs.

	Ave. Nr. Corresp.	D_{grd}
Our method: kernel	92	0.014
Our method: dense	494	0.030

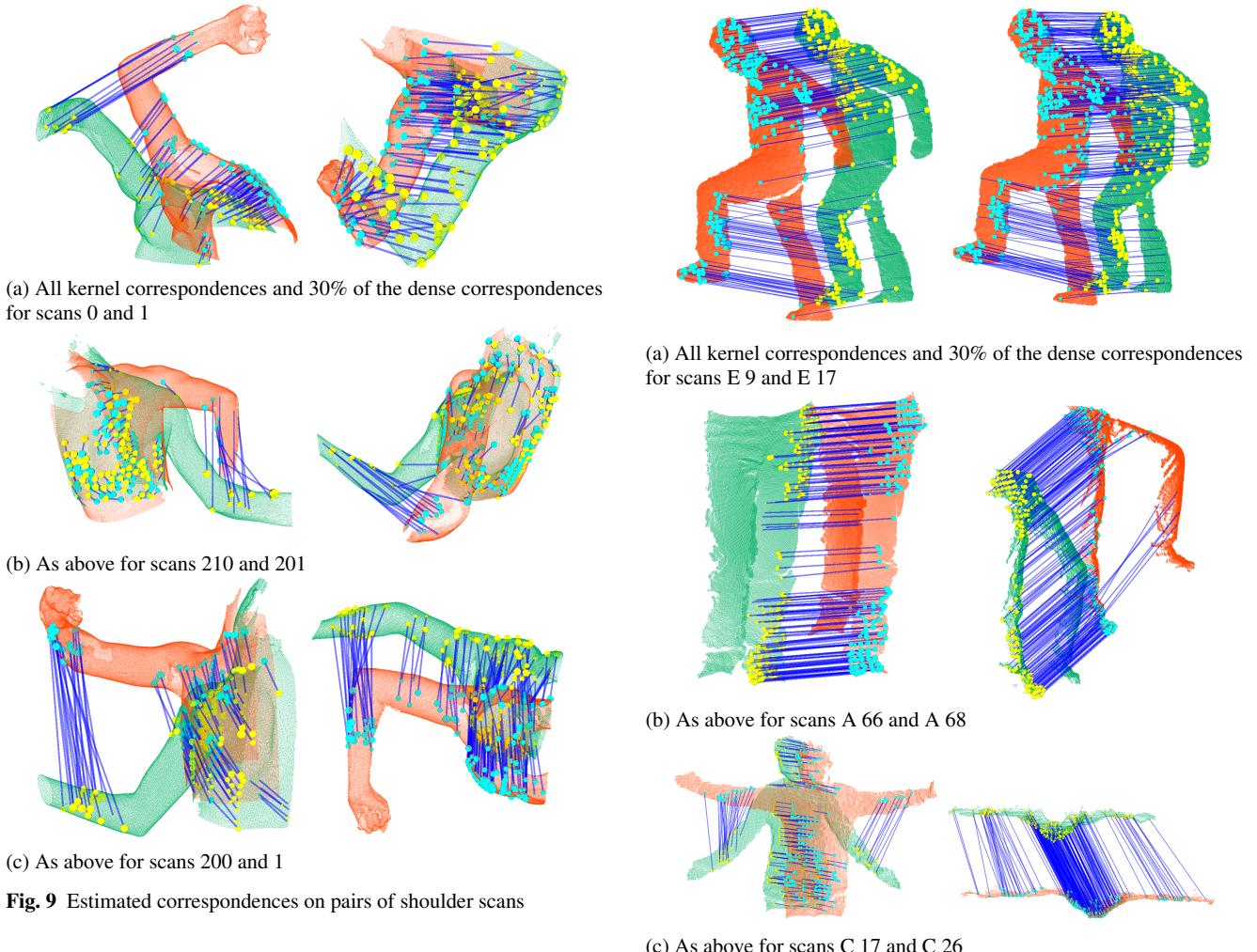
Table 4 Average normalized approximate geodesic distance from ground truth for kernel and dense correspondences on 17 random pairs of shoulder scans

Results on Kinect data. The final dataset contains Kinect range maps collected for this paper. There are three subsets with the dominant motion concentrated on the upper or lower body or both. Ground truth on these data was manually labelled. Figure 10 shows alignment results on these data while Table 5 contains quantitative results. As with the previous dataset, we approximate the geodesic distances to the ground truth by graph distances in 20-nearest neighbor graphs. We report the average accuracy over kernel and dense correspondences for the entire dataset since the statistics of the three subsets are very similar. It is worth pointing out that large deformations pose no insurmountable difficulties to our algorithm, which is also not affected by the fact that the shapes are partial with irregular boundaries.

	Ave. Nr. Corresp.	D_{grd}
Our method: kernel	387	0.012
Our method: dense	1133	0.017

Table 5 Average normalized approximate geodesic distance from ground truth for kernel and dense correspondences on 26 random pairs of Kinect scans.

Timing Results. Running times vary depending on the number of points in the input data, the local point density and also on the number of parts that are discovered. Each part

**Fig. 9** Estimated correspondences on pairs of shoulder scans

requires the estimation of an alignment with a potentially corresponding part followed by clustering to discover which points are consistent with the alignment. Timing results are reported on an Intel Core i7-2670QM CPU at 2.20 GHZ. The Point Cloud Library (PCL) [27] is used for many of the supporting tasks, like FPFH descriptor computation, but other parts of the code are not optimized.

Execution times per task and per dataset are shown in Table 6. The computation of surface normals and FPFH descriptors depends on the local density of the point cloud since they are done in r -neighborhoods. Geodesic distance computation and correspondence densification primarily depend on the number of input points, while establishing part correspondences and kernel matches depend on the number of parts and the number of points. BIM, according to results in [16], takes approximately 80 seconds to register two SCAPE meshes and between 358 and 1945 seconds for SHREC 2011 (TOSCA) shapes depending on the number of points. A processor with the same clock speed as ours was used. RAVAC [30] takes 22 seconds on SCAPE and 557 seconds on an iso-iso pair from SHREC 2011 on a slightly faster processor than ours.

Fig. 10 Results on Kinect data

9 Conclusions

We have presented a novel approach for non-rigid registration of point clouds that is fully automatic, does not require topology information and is able to discover the number of moving parts observed in the two input point clouds. Our approach can handle large articulated motions and does not suffer from ambiguities due to symmetry because it ensures that corresponding parts in the two shapes are attached consistently at the joints of previously matched parts. We have shown results on open and closed surfaces including comparisons with state of the art methods that demonstrate that

Dataset	FPFH/Init.	Geod.	Kernel	Dense	Total
SCAPE	1.12	13	71	348	433
SHREC	26.4	18	273	484	802
Shoulder	9.00	0.82	1.27	4.00	15.09

Table 6 Timing results per dataset in seconds. The four main stages of processing are: FPFH computation and initial alignment; geodesic distance computation; kernel match estimation for all parts; and dense correspondence estimation.

our method is competitive even though it makes less restrictive assumptions about the data.

There are three limitations of our method. The first is that if the correspondence of the first rigid part is wrong, our method cannot recover. This can be addressed by keeping track the top few alignment hypotheses generated by RANSAC, as in [28]. In practice, we did not observe any failures due to this, since real point clouds are unlikely to be perfectly symmetric and incorrect hypotheses do not compete evenly with the correct one. The second limitation is that our current implementation cannot handle parts that appear disconnected due to occlusion or self occlusion, such as a forearm that is visible but is attached to an occluded upper arm for example. Disconnected parts do not allow the algorithm to detect joints between them and the rest of the shape. The implementation will be augmented to handle this case in our future work by continuing the search for part correspondences in the presence of a joint in only one of the shapes. The occluded joint would have to be inferred based on part proximity. The third limitation is that large free-form deformations which cannot be approximated as piece-wise rigid cannot be handled.

Acknowledgements

Hao Guo is supported by a China Scholarship Council (CSC) scholarship. This work has been partially supported by National Science Foundation award #1217797 and a Google Research Award. The authors are grateful to Qin Ma and Ke Wang from the Key Laboratory of Agricultural Information Acquisition Technology of Ministry of Agriculture, China Agricultural University, for their help in data acquisition. We are also grateful to Yusuf Sahillioglu for providing details on how the SCAPE data were used in [30].

References

- Allen, B., Curless, B., Popovic, Z.: Articulated body deformation from range scan data. *ACM Trans. on Graphics* **21**(3), 612–619 (2002)
- Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: Shape completion and animation of people. *ACM Trans. on Graphics* **24**(3), 408–416 (2005)
- Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. *PAMI* **14**(2), 239–256 (1992)
- Boyer, E., Bronstein, A.M., Bronstein, M.M., Bustos, B., Darom, T., Horaud, R., Hotz, I., Keller, Y., Keustermans, J., Kovnatsky, A., Litman, R., Reininghaus, J., Sipiran, I., Smeets, D., Suetens, P., Vandermeulen, D., Zaharescu, A., V., Z.: Shrec 2011: Robust feature detection and description benchmark. In: Eurographics Workshop on 3D Object Retrieval (2011)
- Bronstein, A., Bronstein, M.M., Castellani, U., Dubrovina, A., Guibas, L.J., Horaud, R.P., Kimmel, R., Knossow, D., Von Lavante, E., Mateus, D., Ovsjanikov, M., Sharma, A.: Shrec 2010: Robust correspondence benchmark. In: Eurographics Workshop on 3D Object Retrieval (2010)
- Brown, B.J., Rusinkiewicz, S.: Global non-rigid alignment of 3-d scans. *ACM Trans. on Graphics* **26**(3), 21 (2007)
- Budd, C., Huang, P., Klaudiny, M., Hilton, A.: Global non-rigid alignment of surface sequences. *IJCV* **102**(1-3), 256–270 (2013)
- Chang, W., Zwicker, M.: Range scan registration using reduced deformable models. *Computer Graphics Forum* **28**(2), 447–456 (2009)
- Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. *Image and vision computing* **10**(3), 145–155 (1992)
- Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: *ECCV*, pp. Vol III: 224–237 (2004)
- Huang, Q.X., Adams, B., Wicke, M., Guibas, L.J.: Non-rigid registration under isometric deformations. *Computer Graphics Forum* **27**(5), 1449–1457 (2008)
- Huang, Q.X., Guibas, L.: Consistent shape maps via semidefinite programming. *Computer Graphics Forum* **32**(5), 177–186 (2013)
- Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI* **21**(5), 433–449 (1999)
- Kanezaki, A., Rodola, E., Cremers, D., Harada, T.: Learning similarities for rigid and non-rigid object detection. In: *3DV* (2014)
- Kim, V.G., Li, W., Mitra, N.J., DiVerdi, S., Funkhouser, T.A.: Exploring collections of 3d models using fuzzy correspondences. *ACM Trans. on Graphics* **31**(4) (2012)
- Kim, V.G., Lipman, Y., Funkhouser, T.: Blended intrinsic maps. *ACM Trans. on Graphics* **30**(4), 79 (2011)
- Li, H., Sumner, R.W., Pauly, M.: Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum* **27**(5), 1421–1430 (2008)
- Lipman, Y., Funkhouser, T.: Möbius voting for surface correspondence. *ACM Trans. on Graphics* **28**(3), 72 (2009)
- Ma, J., Zhao, J., Tian, J., Yuille, A.L., Tu, Z.: Robust Point Matching via Vector Field Consensus. *IEEE Transactions on Image Processing* **23**(4), 1706–1721 (2014)
- Mitra, N.J., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. *ACM Trans. on Graphics* **25**(3), 560–568 (2006)
- Mitra, N.J., Guibas, L.J., Pauly, M.: Symmetrization. *ACM Trans. on Graphics* **26**(3), 63 (2007)
- Mitra, N.J., Nguyen, A., Guibas, L.: Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry and Applications* **14**(4–5), 261–276 (2004)
- Rodola, E., Bulo, S.R., Windheuser, T., Vestner, M., Cremers, D.: Dense non-rigid shape correspondence using random forests. In: *CVPR*, pp. 4177–4184 (2014)
- Rodola, E., Torsello, A., Harada, T., Kuniyoshi, Y., Cremers, D.: Elastic net constraints for shape matching. In: *ICCV*, pp. 1169–1176 (2013)
- Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: *3-D Digital Imaging and Modeling*, pp. 145–152. IEEE (2001)
- Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3d registration. In: *ICRA*, pp. 3212–3217 (2009)

27. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (PCL). In: ICRA (2011)
28. Sahillioglu, Y., Yemez, Y.: Coarse-to-fine isometric shape correspondence by tracking symmetric flips. Computer Graphics Forum **32**(1), 177–189 (2013)
29. Sahillioglu, Y., Yemez, Y.: Multiple shape correspondence by dynamic programming. Computer Graphics Forum **33**(7), 121–130 (2014)
30. Sahillioglu, Y., Yemez, Y.: Partial 3-d correspondence from shape extremities. Computer Graphics Forum (2014)
31. Salti, S., Tombari, F., Di Stefano, L.: SHOT: Unique signatures of histograms for surface and texture description. CVIU **125**, 251–264 (2014)
32. Sharma, A., Horaud, R., Cech, J., Boyer, E.: Topologically-robust 3d shape matching based on diffusion geometry and seed growing. In: CVPR, pp. 2481–2488 (2011)
33. Sipiran, I., Bustos, B.: A fully hierarchical approach for finding correspondences in non-rigid shapes. In: ICCV, pp. 817–824 (2013)
34. Starck, J., Hilton, A.: Correspondence labelling for wide-timeframe free-form surface matching. In: ICCV (2007)
35. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. ACM Trans. on Graphics **24**(3), 553–560 (2005)
36. 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.: Registration of 3d point clouds and meshes: A survey from rigid to non-rigid. IEEE Transactions on Visualization and Computer Graphics **19**(7), 1199–1217 (2013)
37. Tevs, A., Bokeloh, M., Wand, M., Schilling, A., Seidel, H.P.: Isometric registration of ambiguous and partial data. In: CVPR, pp. 1185–1192 (2009)
38. Van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D.: A survey on shape correspondence. Computer Graphics Forum **30**(6), 1681–1707 (2011)
39. Zeng, Y., Wang, C., Wang, Y., Gu, X., Samaras, D., Paragios, N.: Intrinsic dense 3d surface tracking. In: CVPR, pp. 1225–1232 (2011)
40. Zhang, H., Sheffer, A., Cohen-Or, D., Zhou, Q., Van Kaick, O., Tagliasacchi, A.: Deformation-driven shape correspondence. Computer Graphics Forum **27**(5), 1431–1439 (2008)