This WACV 2022 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Registration of Human Point Set using Automatic Key Point Detection and Region-aware Features

Amar Maharjan, Xiaohui Yuan
University of North Texas, Denton
amarmaharjan@my.unt.edu, xiaohui.yuan@unt.edu

Abstract

Non-rigid point set registration is challenging when point sets have large deformations and different numbers of points. Examples of such point sets include human point sets representing complex human poses captured by different types of depth cameras. In this work, we present a probabilistic, non-rigid registration method to deal with these issues. Two regularization terms are used: key point correspondences and local neighborhood preservation. Our method detects key points in the point sets based on geodesic distance. Correspondences are established using a new cluster-based, region-aware feature descriptor. This feature descriptor encodes the association of a cluster to the left-right (symmetry) or upper-lower regions of the point sets. We use the Stochastic Neighbor Embedding (SNE) constraint to preserve the local neighborhood of the point set. Experimental results on challenging 3D human poses demonstrate that our method outperforms the state-of-the-art methods. Our method achieved highly competitive performance with a slight increase of error by 3.9% in comparison with the method using manually specified key point correspondences.

1. Introduction

Non-rigid point set registration plays an important role in many computer vision applications such as human movement tracking [26] and surface matching [11]. However, registration of point sets becomes challenging when there are significant deformations and a different number of points between the points sets [17].

Many existing methods leverage key point (landmark) correspondence to deal with large deformations [11, 18]. Key points, usually sparse, are the points that represent important regions in the point sets such as points at the head, hands, and feet in a 3D human point set. In these methods, the key point correspondences between the point sets are obtained either manually [18] or by matching feature descriptors [23, 24] of the identified key points in feature space. However, manually preparing key point correspondences is often difficult; on the other hand, finding key point correspondences via feature descriptors matching usually results in a large number of incorrect correspondences [20] due to noise, incomplete data, or an inconsistent number of points (different spatial resolutions) in the point sets.

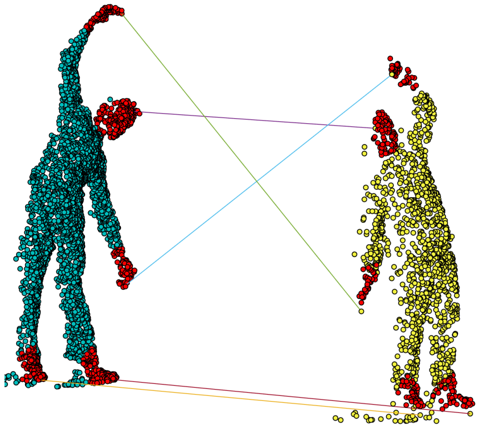

Figure 1. Key points correspondences from our cluster-based region-aware feature descriptor. Both point sets are noisy and the number of points in the left set is twice the number of points in the right set.

In this paper, we present a non-rigid registration method to deal with large, articulated deformations of point sets with different numbers of points such as human point sets acquired with different types of devices. We leverage two important constraints: key point correspondences and local neighborhood preservation. Our method detects the key points in the point sets based on the geodesic distance [21, 1, 12]. We then extract feature descriptors based on cluster statistics of the detected key points to compute key point correspondences. We use cluster regions statistics in our feature descriptor as they are robust in noisy and incomplete data, and cope well even when the number of points between point sets varies significantly. Our feature descriptor encodes left-right and upper-lower cluster regions information that are critical in point set with symmetry such as 3D human point sets. We remove outliers and retain good key point correspondences (inliers), which are used as a constraint in our optimization for the registration. Additionally, we aim to preserve local neighborhood structure during registration using Stochastic Neighbor Embedding (SNE) [9]. Fig. 1 shows an example of key point correspondences, which are identified using our feature descriptor.

The rest of this paper is organized as follows: Section 2 discusses the related work. In Section 3, we present our proposed method. Section 4 discusses our experimental results and compares the proposed method with state-of-the-art methods. Section 5 concludes this paper with a summary.

2. Related Work

Non-rigid point set registration methods based on density estimation are known for their robustness to noise and outliers. In such methods, points from one point set, template, are treated as centroids of Gaussian Mixture Models (GMMs) and transformed so that they align with points of other point set as close as possible. Typically, solutions to these methods are obtained using EM or EM-like algorithm. One of the earliest methods based on density estimation is proposed by Chui et al. [5], which used an EM algorithm together with a deterministic annealing scheme for non-rigid registration. Myronenko and Song [19] presented a popular method, Coherent Point Drift (CPD), where the Gaussian centroids are constrained to move coherently to preserve the topological structure. Many methods extended the CPD to deal with large deformation [8, 6, 16]. Ge et al. [8, 7] extended the CPD called Global Local Topological Preservation (GLTP) to deal with highly articulated non-rigid deformation by adding Local Linear Embedding [22] constraint, which preserves local neighborhood structure. Later, they extended the GLTP to handle complex non-rigid and articulated deformations by incorporating additional constraints to preserve the local neighborhood scale using the Laplacian coordinate (LC). Recently, Hirose [10] presented an algorithm, Bayesian CPD (BCPD), which formulates the CPD in a Bayesian setting and used the prior distribution of displacement vectors for motion coherence instead of using the motion coherence theory. However, these methods may need specific template pose such as T-pose in human point set and are susceptible to local minima in case of large deformation.

Previous methods have been proposed to identify correspondences between surfaces with symmetries [13, 28, 27]. Liu et al. [13] proposed a method to find surface correspondences using symmetry axis curves. In the method, the first symmetry axis curves are aligned that are identified on surfaces. Then, correspondences are obtained by extrapolating correspondences which are found on the axis curves on the surfaces. Yoshiyasu et el. [28] proposed a nonrigid shape matching method for finding correspondences on 3D surfaces that exhibit intrinsic reflectional symmetry. An oriented local depth map was used that is sensitive to local reflectional symmetry. Yoshiyasu et al. [27] presented a method to establish correspondences between shapes that have symmetric (left-right) and rotational (front-back) flips. The proposed symmetry-aware embedding embeds surfaces into lower-dimensional (3D) unlike previous embedding methods that embed in higher dimensional spaces. However, these methods are unclear on point clouds with noises and different resolutions (number of points) between the points clouds which are typical in widely available depth cameras.

Several other methods have proposed to use initial putative correspondences and then refine the correspondences by removing incorrect correspondences as the points transformed during optimization or iterative process. The refined or true correspondences are used to get dense correspondences. Ma et al. [15] proposed a method which creates a set of putative correspondences using feature descriptors, such as shape context [2] for 2D and MeshHOG [30] for 3D, and then identifies correct correspondences by interpolating a smooth vector field between the point sets. Later, the method is extended by adding manifold regularization [25, 14] to preserve the intrinsic structure of the point set. However, the feature descriptors used to compute initial correspondences depend on similar neighborhood structures, suffer from the symmetric flip problem, and are not robust when the resolutions of the point sets are significantly different.

3. Proposed Method

Given two point sets $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$ in a D dimensional space, where M and N denote the number of points in $\mathbf{X}$ and $\mathbf{Y}$, respectively. We detect the key points on both point sets and compute the correspondences using our novel cluster-based region-aware feature descriptor. Finally, we use the key point correspondences for global structure and local neighborhood preservation constraints in our non-rigid point set registration to deal with large deformations and different number of points between the points sets.

3.1. Key Point Detection and Feature Descriptor

Given a point set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ containing N number of points, our task is to detect n number of points in $\mathbf{X}$ which have largest geodesic distance from the mean of $\mathbf{X}$. We treat each point, $\mathbf{x}_i \in \mathbf{X}$, as a node in an undirected weighted graph, G, and its neighbors are identified as all the points of $\mathbf{X}$ which are within δ_k distance from $\mathbf{x}_i$. We add an edge between two neighbors $\mathbf{x}_i$ and $\mathbf{x}_j$ and the weight of the edge is computed as: $w_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$. Note that it is important to make sure that the graph G is one single connected graph. So, we detect a graph that has the highest number of nodes as the main graph, G_M, and connect with the other smaller graphs. Two graphs are connected by identifying a node from each graph and the connection of which gives the shortest distance. An edge is hence formed to connect these two nodes and the weight to the edge follows the aforementioned Euclidean distance. We treat smaller graphs as outliers, which have less than δ_o number of nodes and do not connect with the main graph G_M. We set δ_o = 50 in our experiments. We detect n key points, $E_k = \{e_1, \ldots, e_n\}$, of the point set $\mathbf{X}$ using geodesic extrema (see more on [21, 1]). Fig. 2 shows first seven geodesic extrema for different challenging poses.

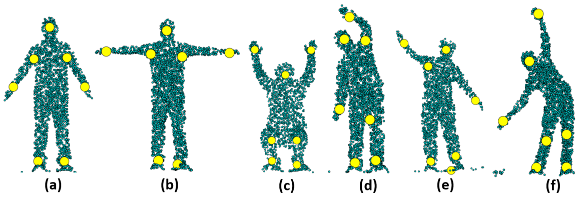

Figure 2. Detected (first seven) key points are shown as yellow dots.

Our key observations on computing key point correspondences based on existing feature descriptors [23, 24] are that the quality of these feature descriptors degrades on 1) point sets with noises, 2) point sets with different resolutions (number of points). The existing feature descriptors are unreliable in computing key point correspondences as they assume the same or similar local neighborhood structure around the key point. Therefore, our new feature descriptor is based on a cluster or region around the key points. We extract points (cluster) around each key point with radius r_c and compute the eigenvalues and three principal component axes/lengths of the clusters.

3.2. Left-right (Symmetry) Cluster Regions

To distinguish between symmetric clusters (regions) such as the left hand (foot) and right hand (foot), we propose to use distance (geodesic) differences between distances from cluster mean to two points: points that are left and right side of the mean of a point set (in the horizontal direction), p_m. To choose these points, we take points that are $-\delta$ and δ away from p_m for left and right points, denoted by p_l^m and p_r^m, respectively. The key idea is that the geodesic distance difference from left cluster mean p_c^l, i.e., mean of a cluster which is left side of p_m, to p_l^m should be less than geodesic distance from p_c^l to p_r^m. Similarly, for a right cluster mean p_c^r (mean of a cluster which is right side of p_m), geodesic distance from p_c^r to p_l^m should be greater than geodesic distance from p_c^r to p_r^m. So, we exploit this geodesic distance difference to encode the left and right context of the clusters of the point sets. Fig. 3 shows geodesic path from cluster mean to the left (p_l^m) and right (p_r^m) side points of the mean of the point set, p_m. We compute the left-right (symmetry) distance difference metric, s, as follows:

$$s = \exp\left(d_g(p_c^m, p_l^m) - d_g(p_c^m, p_r^m)\right) \qquad (1)$$

where $d_g(p_i, i_j)$ is a function that returns geodesic distance between two points p_i and p_j.

In practice, the computation of s is sensitive to noise and voids on the point sets. To circumvent this, we take neighboring points above and below p_m for neighboring means and compute s using Eq.(1) for all the neighboring means. We compute the average of all the symmetry feature information computed on the neighboring means to get the final symmetry feature information, $s_f = \frac{1}{N_s}\sum_k^{N_s} s_k$, where $s_k = \exp\left(d_g(p_c^m, p_l^k) - d_g(p_c^m, p_r^k)\right)$ is a left-right (symmetry) metric using cluster mean p_c^m and a neighboring mean p_k. We set N_s equals to 15.

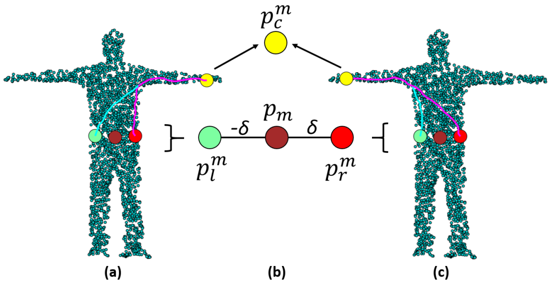

Figure 3. Identifying left and right cluster regions. The geodesic paths from cluster mean (p_c^m) to left (p_l^m) and right (p_r^m) points are shown in cyan and magenta, respectively.

3.3. Upper-lower Cluster Regions

Similar to left-right symmetry, we also compute the feature of a cluster that encodes information whether the cluster belongs to the upper half or lower half of a point set. We propose to use (geodesic) differences between distances from the cluster mean, p_c^m, to two neighboring points to distinguish between upper and lower cluster regions. The two points that are upper (above) and lower (below) side of the mean of the point set (in the vertical direction, y-axis), p_m. To choose these points, we take points that are $-\delta$ and δ away from p_m for upper (p_a^m) and lower (p_b^m) points (in y-axis), respectively. The key insight is that the geodesic distance difference from upper cluster mean p_c^m (cluster above p_m) to p_a^m should be less than p_c^m to p_b^m. Similarly, for a lower cluster region p_c^m (cluster below the mean of point set), geodesic distance from p_c^m to p_a^m should be greater than p_c^m to p_b^m. Fig. 4 shows geodesic paths from cluster means to the upper and lower points. We add following upper-lower distance metric in our feature descriptor: $u = \lambda$ if $d_g(p_c^m, p_a) < d_g(p_c^m, p_b)$ otherwise $-\lambda$, where $d_g(p_i, p_j)$ is a function that returns geodesic distance between two points p_i and p_j, p_c^m is a cluster mean, p_a^m is an upper point, p_b^m is a lower point, and λ is some constant value. We set λ to 1.0.

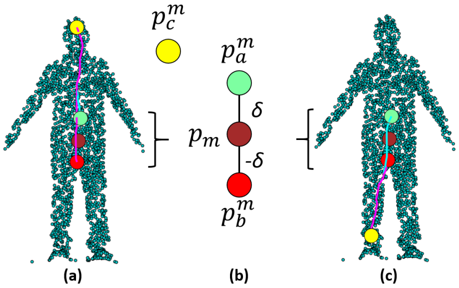

Figure 4. Identifying upper and lower cluster regions. The geodesic paths from cluster mean (p_c^m) to upper (p_a^m) and lower (p_b^m) points are shown in cyan and magenta colors, respectively.

3.4. Key Point Correspondence

To get the key point correspondences, we first compute the cost matrix: $C(i, j) = c_{ij} = \|F_i - F_j\|_2$, where F_k is our cluster-based feature descriptor for key point p_k. Then we use Hungarian method to get the initial correspondences between the key points which minimizes the following cost function: $H(\pi) = \sum_{i=1}^{K} C(i, \pi(i))$, where π is the bijection function $\pi : \mathbf{X} \rightarrow \mathbf{Y}$. Since our initial key point correspondence list may contain incorrect correspondences, we prune incorrect correspondences and keep only those correspondences which have higher normalized weight values, w_{ij}, between the correspondences of two key points $p_i \in \mathbf{X}$ and $p_j \in \mathbf{Y}$ as follows: $w_{ij} = \frac{\|F_i - F_j\|_2}{\sum_{(i,n)\in\kappa_i} \|F_i - F_n\|_2}$, where F_i and F_j are features of p_i and p_j respectively, and κ_i is a set that contains key point correspondence tuples from key point $p_i \in \mathbf{X}$ to every key points $p_n \in \mathbf{Y}$. We include correspondences from our initial list whose normalized correspondence weight is less than a threshold ω. In our experiments, ω equals 0.01.

3.5. Non-rigid Point Set Registration

We treat our non-rigid point set registration as a density estimation problem based on the Gaussian mixture model (GMM) [18]. In our method, moving points from one point set (template) are GMM centroids and align with another point set (input). In addition, we add two regularization terms for local neighborhood preservation and global structure constraints using Stochastic Neighbor Embedding (SNE) [9] and key points correspondences [17], respectively. Expectation-Maximization (EM) optimization is used to find out the best parameter settings to align both point sets as close as possible. We assume noise follows the uniform distribution, i.e., $p_u = \frac{1}{N}$, and have the probability density function of point $\mathbf{x}_n$ given $\mathbf{Y}$ as follows:

$$p(\mathbf{x}_n) = (1 - \gamma) \sum_{m=1}^{M} p(\mathbf{x}_n|\mathbf{y}_m)p(\mathbf{y}_m) + \gamma p_u \tag{2}$$

where $p(\mathbf{y}_m) = \frac{1}{M}$, $\gamma \in [0, 1]$ denotes the rate of noise and outliers in the observed dataset $\mathbf{X}$. To maintain the global structure of the point set, our optimal transformation function minimizes the distances between the corresponding key points of the point sets (see Section 3.4). So, our global structure constraint is defined as follows:

$$E_G = \sum_{m,n}^{M,N} \mathbf{A}_{m,n} \|\mathbf{x}_n - \tau(\mathbf{y}_m)\|^2 \tag{3}$$

where $\mathbf{A}_{M\times N}$ is key point coefficient matrix, $\mathbf{A}_{m,n} = 1$ if $(\mathbf{x}_n, \mathbf{y}_m) \in L$; otherwise, 0, L is a set containing all pairs of key point correspondences, and $\tau(.)$ is transformation function.

We employ Stochastic Neighbor Embedding (SNE) [9] constraint to keep points within a neighborhood relatively close and points far apart remain distant after transformation. Let r_{ij} be the probability that two points $\mathbf{y}_i$ and $\mathbf{y}_j$ are neighbors before transformation and s_{ij} be the probability that these two points become neighbors after transformation τ. A constraint on the local structure is represented as the minimization of a cost function which is the sum of Kullback-Leibler (KL) divergences between r_{ij} and s_{ij} distributions over neighbors of each point [9]:

$$E_L = \sum_{ij} r_{ij} \log \frac{r_{ij}}{s_{ij}} = \sum_{i} KL\left(\mathbf{R}_i \| \mathbf{S}_i\right), \tag{4}$$

where

$$r_{ij} = \frac{\exp(-\beta_2 \|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k\neq i} \exp(-\beta_2 \|\mathbf{y}_i - \mathbf{y}_k\|^2)},$$

$$s_{ij} = \frac{\exp(-\|\tau(\mathbf{y}_i) - \tau(\mathbf{y}_j)\|^2)}{\sum_{k\neq i} \exp(-\|\tau(\mathbf{y}_i) - \tau(\mathbf{y}_k)\|^2)},$$

β_2 is precision parameter, and $\mathbf{R}_i = [r_{i1}, ..., r_{iM}]$ and $\mathbf{S}_i = [s_{i1}, ..., s_{iM}]$ are probability distributions.

Following the GMM framework in [19], the objective function of our method integrates local and global constraints as follows:

$$\begin{aligned} Q(\theta, \sigma^2) \quad &= \quad \frac{1}{2\sigma^2} \sum_{n,m=1}^{N,M} p^{i-1}(\mathbf{y}_m|\mathbf{x}_n) \|\mathbf{x}_n - \tau(\mathbf{y}_m)\|^2 \\ &+ \frac{N_P D}{2} \ln \sigma^2 + \frac{\lambda_1}{2} E_{MC} + \frac{\lambda_2}{2} E_L + \frac{\lambda_3}{2} E_G \end{aligned} \tag{5}$$

where

$$p^{(i-1)}(\mathbf{y}_m|\mathbf{x}_n) = \frac{\exp^{(-\frac{1}{2}\| \frac{\mathbf{x}_n - \tau(\mathbf{y}_m)}{\sigma_{(i-1)}} \|^2)}}{\sum_{k=1}^{M} \exp(-\frac{1}{2}\| \frac{\mathbf{x}_n - \tau(\mathbf{y}_k)}{\sigma_{(i-1)}} \|^2) + C}, \tag{6}$$

$$C = \gamma(2\pi\sigma^2_{(i-1)})^{D/2}M/((1-\gamma)N),$$

and $E_{MC} = tr(\mathbf{W^T G W})$ is motion coherence constraint to maintain topological structure of the point set [29, 19]. Function $tr(\cdot)$ computes the trace of a matrix, and $N_P = \sum_{n,m=1}^{N,M} p^{(i-1)}(\mathbf{y}_m|\mathbf{x}_n) \leq N$. $\mathbf{G}_{M\times M}$ is a kernel matrix with elements $g_{ij} = G(\mathbf{y}_i, \mathbf{y}_j) = \exp(-\frac{1}{2}\|\frac{\mathbf{y}_i - \mathbf{y}_j}{\beta})\|^2$. $\mathbf{W}_{M\times D} = (\mathbf{w}_1, \ldots, \mathbf{w}_M)^T$ is a coefficients matrix, λ_1, λ_2, and λ_3 are regularization weights for motion coherence, local structure, and key point correspondence constraints, respectively.

To obtain the coefficient matrix $\mathbf{W}$, we take derivative of Eq. (5) with respect to $\mathbf{W}$ and set it equal to zero:

$$\begin{aligned}&(diag(\mathbf{P1})\mathbf{G} + \sigma^2\lambda_1\mathbf{I} + \sigma^2\lambda_2\mathbf{JG} + \sigma^2\lambda_3 diag(\mathbf{A1})\mathbf{G})\mathbf{W} = \\ &\mathbf{PX} - diag(\mathbf{P1})\mathbf{Y} - \sigma^2\lambda_2\mathbf{JY} - \sigma^2\lambda_3 diag(\mathbf{A1})\mathbf{Y} + \sigma^2\lambda_3\mathbf{AX}\end{aligned} \tag{7}$$

where $\mathbf{J} = (diag(\mathbf{R1}) - 2\mathbf{R} + diag(\mathbf{1}^T\mathbf{R}))$, $\mathbf{1}$ refers to column vector of all ones, $\mathbf{I}$ refers to identity matrix, and $diag(\mathbf{v})$ refers to the diagonal matrix created from the vector $\mathbf{v}$.

We define the transformation function, τ, as the initial position, $\mathbf{y}_m$, plus a displacement function $\mathbf{f}(\mathbf{y}_m)$, $\tau(\mathbf{y}_m) = \mathbf{y}_m + \mathbf{f}(\mathbf{y}_m)$. We adopt the following transformation function which moves neighborhood points coherently and helps in maintaining topological structure of the point set [19]: $\mathbf{T} = \tau(\mathbf{Y}, \mathbf{W}) = \mathbf{Y} + \mathbf{GW}$.

Similarly, to obtain σ^2, we take derivative of Eq. (5) with respect to σ^2 and set to zero

$$\begin{aligned}\sigma^2 &= \frac{1}{N_P D}(tr(\mathbf{X}^T diag(\mathbf{P}^T\mathbf{1})) - 2tr(\mathbf{P}\mathbf{X}^T\mathbf{T}) \\ &\quad + tr(\mathbf{T}^T diag(\mathbf{P1})\mathbf{T}))\end{aligned} \tag{8}$$

where $N_P = \mathbf{1}^T\mathbf{P1}$.

4. Experimental Results

In our experiments, we use a human dataset that contains different complex human poses captured by Microsoft Kinect II. The dataset has human subjects with different body shapes and sizes having poses such as stretching, squatting, standing. Each human point set consists of around 12K points. In all our experiments, the parameter values we used are as follows: λ_1 = 2.0, λ_2 = 1.0, λ_3 = 150.0, β_1 = 1.0, β_2 = 15.0, initial $\sigma^2 = \frac{\xi}{NMD}\sum_{n=1}^{N}\sum_{m=1}^{M}\|\mathbf{x}_n - \mathbf{y}_m\|^2$, ξ = 0.1, and maximum number of iterations of EM is 50. We normalized the point sets with zero mean and unit variance before registration. We downsampled uniformly at random for each point set to 2500 in our experiments and the experiments are repeated three times. We detect seven key points on each point sets used in our experiments. For quantitative evaluation, we use normalized Euclidean distance between the corresponding points as follows: $\varepsilon = \frac{1}{N}\sum_i \|\mathbf{x}_i - \mathbf{y}_j\|_2$, where $\mathbf{x}_i \in \mathbf{X}$ and $\mathbf{y}_j \in \mathbf{Y}$ is the estimated corresponding point of $\mathbf{x}_i$ after registration.

To choose the cluster radius, r_c, we select squat and stretch poses and evaluate our method using different radii. For the squat pose, we register both hands upward pose with half squat pose. For the stretch pose, we register t-pose with left (right) stretch poses. Table 1 shows the avg. registration error w.r.t. different radii. We set 0.09 for r_c.

Table 1. Registration error w.r.t. different cluster radii.

	Radius				
Poses	0.03	0.06	0.09	0.12	0.15
Squat	16.96 (1.54)	16.50 (1.47)	16.38 (1.35)	16.40 (1.37)	16.75 (1.56)
Stretch	21.96 (3.56)	20.87 (2.99)	20.26 (2.91)	20.16 (2.81)	20.13 (3.23)

All our experiments are conducted on Intel Core i7-7700 CPU @ 3.60GHz 64-bit Windows 10 machine with 16GB of memory. We evaluate our method in the following aspects: 1) different degrees of deformation, 2) ablation study on the influence of global and local constraints in our method, 3) a different number of points between the point sets, and 4) generalization to point sets other than human. We compare our results with the following state-of-the-art methods: CPD [19], PR-GLS [16], BCPD [10], Landmarks based Global Local Preservation (LGLP) [18], and GLTP [8, 7]. We used the publicly available source code which are distributed by the authors of these methods except LGLP, which is our previous work. For LGLP, we use seven key point correspondences, which are established manually.

4.1. Degree of Deformation

We evaluate our proposed method with different degrees of deformations: small, medium, and large. We select three types of human poses to evaluate the deformation degrees: *squat*, *stretching*, and *both hands moving upward*. For the squat, we select both hands upward pose (template) vs three other squat poses: squat starting pose, half squat pose, and full squat pose. For stretching, we select t-pose (template) is registered with three levels of stretching (left and right) poses from small to large deformations, respectively. Finally, for both hands moving upward pose, we select the pose with both hands down as a template vs three other poses where both human hands are moving upward.

Table 2 shows the average registration errors (stds.) of our and other state-of-the-art methods for all three deformation levels. Our method has the lowest average registration errors and standard deviations for all three deformation levels among CPD, PR-GLS, and BCPD. Our method has

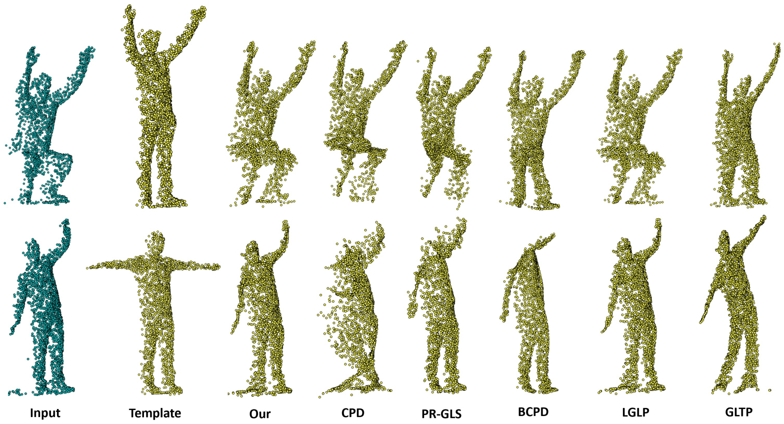

Figure 5. Exemplar registration results on squat (top row) and stretch (bottom row) poses. First two columns show input and template point sets, respectively. Third to seventh columns show results from our method, CPD, PR-GLS, BCPD, LGLP, and GLTP, respectively.

Table 2. Registration error w.r.t. deformations.

Method	Deformation		
	small	medium	large
Our	16.02 (2.84)	18.06 (2.70)	20.09 (3.05)
CPD	*19.14 (4.88)*	*29.33 (8.20)*	*30.55 (6.61)*
PR-GLS	21.85 (15.54)	40.26 (24.81)	40.43 (22.70)
BCPD	23.28 (10.43)	34.80 (22.82)	36.48 (22.18)
GLTP	25.03 (11.86)	40.97 (14.11)	53.53 (14.67)
LGLP	15.78 (2.43)	17.66 (2.49)	19.57 (2.56)

only slightly higher registration errors (stds.) than LGLP, which establishes the key point correspondences manually. In general, registration errors increase when the degree of deformation increases for all the methods, as can be seen in the table. But, error differences between our method and the third best method (CPD) are much higher in large deformation than in small deformation. When combined with all three deformations, the average registration errors for our method, CPD, PR-GLS, BCPD, LGLP, and GLTP are 18.06, 26.34, 34.18, 31.52, 17.67, and 39.84, respectively. In comparison with the method using manually specified key point correspondence [18], our method achieved highly competitive performance with a slight increase of error by 2%. The results suggest that our method successfully identifies important key point correspondences to accommodate large degrees of deformation.

Fig. 5 shows exemplar registration results on squat (top row) and stretch (bottom row) poses. Inputs and templates are shown in the first two columns, respectively. The third to seventh columns show results from our method, CPD, PR-GLS, BCPD, LGLP, and GLTP, respectively. In squat results, both our method and LGLP show better results. CPD and PR-GLS also show good results but have issues in leg regions. BCPD and GLTP show better results in the upper part of the body such as the torso and hands but fail to register accurately in lower body regions such as both legs. In stretching results, our method and LGLP show a better result than CPD, PR-GLS, BCPD, and GLTP. CPD does not even maintain the human shape. PR-GLS and BCPD show a better result than CPD but fail to preserve local regions such as the head, left arm. GLTP is able to maintain the overall shape of the human subject but fails to register correctly in the articulated region of the left hand.

4.2. Ablation Study

This section discusses our study on the influence of global and local constraints on the performance of our method. Human point sets with three degrees of deformations are used in our evaluation. In our analysis of the impact of local constraint, λ_3 was set to zero so that the global constraint is suppressed in the objective function. Similarly, to understand the impact of global constraint, we suppress the influence of local neighborhood constraint in our objective function by setting λ_2 equals to zero.

Table 3 lists the average registration error with respect to the three degrees of deformations. The second row shows the registration errors without using key point correspondences constraint while the second row shows the registration error without using local neighborhood constraint. When only key point correspondences (global constraint) are used, our method performs better than using only local neighborhood constraints in all the three degrees of deformation. Also, using only key point correspondences (but not local neighborhood constraints) generates similar registration errors when we use both global and local constraints in our method. When we take the average of all the registration errors of the three degrees of deformation, our method's error without using key point correspondences and without using local neighborhood constraints are 19.92 and 18.30, respectively. This means that we get 10.3% more error if we do not use the key point correspondences (but use local neighborhood constraint). Similarly, we get 1.3% more

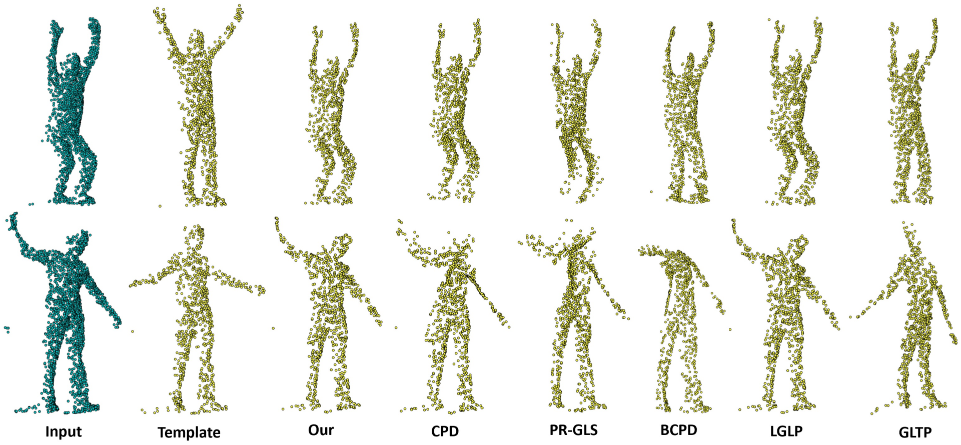

Figure 6. Exemplar registration results on using different number of points between the point sets. First two columns show input and template point sets, respectively. The input point set contains *2500* points while template contains only *1000* points in it. Third to seventh columns show results from our method, CPD, PR-GLS, BCPD, LGLP, and GLTP, respectively.

error if we do not use local neighborhood constraints (but use key point correspondences). Hence, using both global and local constraints is necessary for dealing with large deformation. The influence of the global constraint (key point correspondences) is greater than that of the local constraint.

Table 3. Ablation study: Registration error w.r.t. deformations.

Method	Deformation		
	small	medium	large
our method	16.02 (2.84)	18.06 (2.70)	20.09 (3.05)
w/o global	16.51 (3.34)	20.98 (5.51)	22.28 (3.19)
w/o local	16.11 (3.00)	18.17 (2.81)	20.62 (3.04)

Figure 7 shows registration results of right stretch pose from first three columns and squat pose from fourth to sixth columns without using global constraint. In each pose type, point sets are shown in the following order: input, template, and registration result. In both cases, correct human shapes are not achieved due to the lack of global constraint.

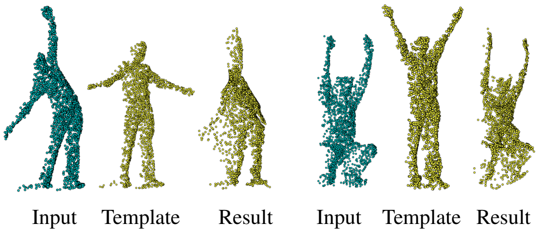

Figure 7. Illustration of inaccurate registration results without using global constraint (key point correspondences).

4.3. Robustness to Different Number of Points

We also evaluate the robustness of our method w.r.t. the different number of points between the points sets. We prepared two types of poses for the experiments (squat and stretching). The template point sets are always fixed with 2,500 points but each input point set is downsampled into five different levels: 2,200, 1,900, 1,600, 1,300, and 1,000 (see Table 4). First, we fix the hands-up pose as a template and register it with three squat poses as inputs, similar to the one used in the degree of deformation levels. Second, we fix the t-pose as the template and register with three different left (right) stretching poses as input point sets.

Table 4. Registration error by changing the number of points in one point set. The other point set has 2,500 points in all experiments.

Method	Number of points in one point set				
	2200	1900	1600	1300	1000
Our	17.34	15.06	13.18	11.36	9.27
	(3.53)	(2.79)	(2.61)	(2.24)	(1.82)
CPD	*23.13*	*20.17*	*17.33*	*14.35*	11.51
	(7.50)	*(6.07)*	*(5.04)*	*(4.22)*	(3.24)
PR-GLS	24.43	21.04	17.97	14.76	*11.18*
	(6.39)	(5.73)	(4.85)	(5.97)	*(2.67)*
BCPD	34.14	35.57	51.92	50.79	37.69
	(18.98)	(21.90)	(116.30)	(105.17)	(15.58)
GLTP	36.34	33.18	27.22	21.98	17.67
	(16.55)	(14.95)	(12.28)	(9.65)	(8.08)
LGLP	16.78	14.49	12.64	10.50	8.56
	(3.15)	(2.37)	(2.08)	(1.59)	(1.36)

Table 4 shows the average registration error against five levels of a different number of points. In all the cases, our registration method has the lowest average registration errors and smaller standard deviations among CPD, PR-GLS, BCPD, and GLTP. Our method's registration results are very close to LGLP, which fixed the key point correspondences manually. When combined with all five levels, the average registration errors for our method, CPD, PR-GLS, BCPD, LGLP, and GLTP are 13.24, 17.30, 17.88, 42.02, 12.43, and 27.28, respectively. This demonstrates that our method is robust to handle different numbers of points between the point sets. However, our method exhibits a higher average error in comparison to LGLP by 6.5%, which used manually established key point correspondences.

Figure 6 shows exemplar registration results on using a different number of points between the point sets. The first two columns show the input and template point sets. Input point sets contain 2,500 points and template point sets contain only 1,000 points. Third to eighth columns show registration results from our method, CPD, PR-GLS, BCPD, LGLP, and GLTP, respectively. In the top row for squat poses, our method, CPD, and LGLP show good results. PR-GLS result shows points around leg regions are not registered correctly. BCPD result shows a better human shape but the squat pose is not maintained. In the bottom row for stretching poses, only our method and LGLP show good results. CPD, PR-GLS, BCPD, and GLTP maintain the mid or lower body part such as the torso or legs but fail to preserve stretching arms, head regions.

4.4. Generalization to Other Point Sets

We evaluate our method to deal with non-human point sets that do not necessarily contain left-right (symmetry) and upper-lower regions. To conduct the experiments, we select two categories of point sets. First, we select the point sets representing knives from the Tools 2D database [4]. The knife point sets do not have left-right (symmetry) regions such as left and right hands (feet) in the human point sets. Second, we select cat point sets from the Nonrigid world 3D database [4, 3]. The cat point sets have left-right (symmetry) regions but lack upper-lower regions such as hands and feet on the human point sets. We use seven key points for the experiments.

Figure 8 shows the registration results on non-human point sets. In both rows, the first column shows the input point sets while the second column shows the template point sets. The third column shows the registration results (transformed template point sets). In tools point sets, our method identified four key point correspondences. Out of four correspondences, two correspondences are accurate (left and right tips of the knives shown in magenta and green connected lines). Despite establishing a small number of correct key point correspondences, our method is able to achieve good registration results for the tool point sets. In cat point sets, our method is not able to establish any key point correspondences. Due to this lack of key point correspondences, the transformed template point set failed to maintain different regions such as the tail, right foot and do not maintain the overall shape of the cat. These results suggest that our method can deal with non-human point sets such as tools point sets but need further improvements to establish key point correspondences on point sets such as animals.

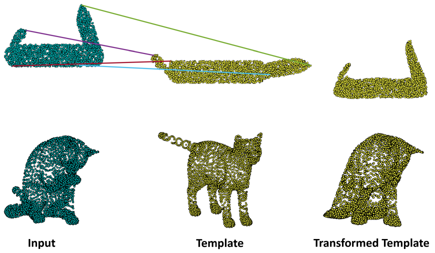

Figure 8. Illustration of registration results on non-human point sets. Identified key point correspondences (top row) are represented by the connected lines. No correspondences are identified between the cat point sets (bottom row).

5. Conclusion

This paper presents a probabilistic non-rigid point set registration method to deal with large deformation and a different number of points between point sets such as data acquired with different types of devices. Two important constraints, key point correspondences, and local neighborhood preservation are used as regularization terms in the registration method. Key points are identified on the point sets using geodesic extrema. The correspondence between key points is computed using our novel cluster-based, region-aware feature descriptor.

Our proposed method is evaluated and compared with state-of-the-art methods on challenging 3D human point sets with a large degree of deformation in poses such as squat and stretching as well as different sizes of the point sets. The experimental results demonstrate that the key point correspondences established between the point sets by our method are highly reliable. Because of highly reliable key point correspondences identified by our method, its average registration error is only about 3.9% higher than the method that uses manual key point correspondences when combined with all the experimental results. Our method outperforms other state-of-the-art methods such as CPD, PR-GLS, BCPD, and GLTP. In particular, when combined with all the quantitative experimental results, our method's average registration error is 28.28%, 39.88%, 57.44%, and 53.37% lower than CPD, PR-GLS, BCPD, and GLTP, respectively.

Further, our ablation study on global and local constraints suggests that both constraints are necessary for dealing with large deformation. The influence of the global constraint (key point correspondences) is greater than that of the local constraint in our method. Finally, in our future work, we plan to deal with non-human point sets such as animal point sets to correctly establish key point correspondences between the point sets.

References

[1] A. Baak, M. Muller, G. Bharaj, H. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *2011 International Conference on Computer Vision*, pages 1092–1099, 2011.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[3] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[4] Alexander M. Bronstein, Michael M. Bronstein, Alfred M. Bruckstein, and Ron Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 78(1):67–88, Jun 2008.

[5] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No.PR00737)*, pages 190–197, 2000.

[6] S. Ge and G. Fan. Non-rigid articulated point set registration with local structure preservation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–133, June 2015.

[7] Song Ge and Guoliang Fan. Topology-aware non-rigid point set registration via global—local topology preservation. *Machine Vision and Applications*, 30(4):717–735, Jun 2019.

[8] S. Ge, G. Fan, and M. Ding. Non-rigid point set registration with global-local topology preservation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 245–251, 2014.

[9] Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 857–864. MIT Press, 2003.

[10] O. Hirose. A bayesian formulation of coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[11] Tao Jiang, Xiaosong Yang, Jianjun Zhang, Feng Tian, Shuang Liu, Nan Xiang, and Kun Qian. Huber-l_1-based non-isometric surface registration. *Vis. Comput.*, 35(6-8):935–948, jun 2019.

[12] Longbo Kong, Xiaohui Yuan, and Amar Man Maharjan. A hybrid framework for automatic joint detection of human poses in depth frames. *Pattern Recogn.*, 77(C):216–225, may 2018.

[13] Tianqiang Liu, Vladimir G. Kim, and Thomas Funkhouser. Finding surface correspondences using symmetry axis curves. *Comput. Graph. Forum*, 31(5):1607–1616, Aug. 2012.

[14] J. Ma, J. Wu, J. Zhao, J. Jiang, H. Zhou, and Q. Z. Sheng. Nonrigid point set registration with robust transformation learning under manifold regularization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2018.

[15] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, April 2014.

[16] J. Ma, J. Zhao, and A. L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on Image Processing*, 25(1):53–64, Jan 2016.

[17] Amar Maharjan and Xiaohui Yuan. Point set registration of large deformation using auxiliary landmarks. In *Urban Intelligence and Applications*, pages 86–98, Singapore, 2020. Springer Singapore.

[18] Amar Maharjan, Xiaohui Yuan, Qiang Lu, Yuqi Fan, and Tian Chen. Non-rigid registration of point clouds using landmarks and stochastic neighbor embedding. *Journal of Electronic Imaging*, 30(3):1 – 15, 2021.

[19] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, Dec 2010.

[20] A. Parra Bustos and T. Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2868–2882, 2018.

[21] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *2010 IEEE International Conference on Robotics and Automation*, pages 3108–3113, 2010.

[22] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

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

[24] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Computer Vision – ECCV 2010*, pages 356–369, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[25] Gang Wang, Zhicheng Wang, Yufei Chen, Xianhui Liu, Yingchun Ren, and Lei Peng. Learning coherent vector fields for robust point matching under manifold regularization. *Neurocomputing*, 216:393–401, 2016.

[26] M. Ye, Y. Shen, C. Du, Z. Pan, and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1517–1532, 2016.

[27] Yusuke Yoshiyasu, Eiichi Yoshida, and Leonidas Guibas. Symmetry aware embedding for shape correspondence. *Computers & Graphics*, 60:9 – 22, 2016.

[28] Y. Yoshiyasu, E. Yoshida, K. Yokoi, and R. Sagawa. Symmetry-aware nonrigid matching of incomplete 3d surfaces. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4193–4200, 2014.

[29] A. L. Yuille and N. M. Grzywacz. The motion coherence theory. In *Second International Conference on Computer Vision*, pages 344–353, 1988.

[30] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–380, June 2009.