Robust Template-Based Non-Rigid Motion Tracking Using Local Coordinate Regularization

Wei Li, Shang Zhao, Xiao Xiao, James K. Hahn
Department of Computer Science, The George Washington University
{gw_liwei,edwinz,xxfall2012,hahn}@gwu.edu

Abstract

In this paper, we propose our template-based non-rigid registration algorithm to address the misalignments in the frame-to-frame motion tracking with single or multiple commodity depth cameras. We analyze the deformation in the local coordinates of neighboring nodes and use this differential representation to formulate the regularization term for the deformation field in our non-rigid registration. The local coordinate regularizations vary for each pair of neighboring nodes based on the tracking status of the surface regions. We propose our tracking strategies for different surface regions to minimize misalignments and reduce error accumulation. This method can thus preserve local geometric features and prevent undesirable distortions. Moreover, we introduce a geodesic-based correspondence estimation algorithm to align surfaces with large displacements. Finally, we demonstrate the effectiveness of our proposed method with detailed experiments.

1. Introduction

The recent development of low cost RGB-D cameras makes surface reconstruction and motion capture more accessible to general users. Compared with traditional commercial marker-based motion capture systems, such as Vicon®, the ability to track time-varying deformable surfaces other than the skeletal motions with commodity depth cameras provides more flexibilities and capabilities in various areas such as motion analysis, medical simulation and virtual reality. The major challenge of the motion tracking problem is to solve the misalignments between the deformed model and the partially captured scans. In this paper, we propose our template-based non-rigid registration algorithm to address the misalignments in a frame-to-frame motion tracking pipeline with single or multiple depth cameras.

There are several possible reasons for the failure of motion tracking. With sparse distributed depth camera setup, some regions of the surface may be occluded from the camera view preventing motion tracking. When those regions become visible again in subsequent frames, the calculated deformation may differ greatly from the actual deformation. Moreover, due to the low capture frequency of the commodity depth cameras, fast motions can cause large displacements in the camera space. This violates the assumption of small displacements in some correspondence estimation algorithms, which may lead to incorrect result. In other cases, even though the surface is visible to the camera and has reliable correspondences in the depth image, the surface is still not well registered. This is because the non-optimal constraints in the template model prevent large potential deformation from the rest pose. Error accumulation in the registered frames can also reduce the quality of alignments and result in undesirable distortion for long motion sequences.

In this paper, we utilize the differential representation of the deformation field to address the above issues in the framework of volumetric embedded deformation graph [28]. In this differential representation, we define the local deformation and the rotation-invariant regularization in the local coordinates of embedded nodes [16, 1]. We formulate an analytic solution for the regularization term by analyzing the net strain energy between each pair of neighboring nodes in the graph. This differential representation can help preserve local geometric features and prevent undesirable stretching, bending or torsional deformations. Unlike previous works which select a single reference frame to constrain the deformation of the entire surface, we adaptively select the reference frames for different regions of the surface in the regularization term based on the tracking status of the surface. We treat the registered motion sequences as feasible candidates for the deformable surface. Various tracking strategies are then applied to minimize misalignments and recover untracked regions. Moreover, we use the differential representation in multi-view motion tracking to transfer the registered motion from one view to another in order to reduce the effects of asynchronization and interference between different cameras. Finally, we introduce a geodesic-based correspondence estimation algorithm to solve large misalignments after the initial projective alignment. Unlike the previous spectral-based algorithms, our method focuses on evaluating geodesic features and estimating correspondences in the partially aligned scans with incomplete topologies.

2. Related Work

2.1. Template-Based Non-Rigid Registration

The deformation model for template-based non-rigid registration is usually based on differential geometry methods [7, 30, 12]. Botsch et al. [1] provide a comprehensive summary of the linear variational deformation methods. Local feature preservation and rotational invariance are important properties that lead to the popularity of these methods. As-Rigid-As-Possible (ARAP) [27] is one of the commonly used regularizations for deformable surfaces. This method estimates the local transformation from the neighboring vertices, and then minimizes the vertex displacement in the local space. Embedded deformation graph [28, 22, 33] samples a set of nodes on the surface and associates each node with an affine transformation which is implicitly solved in the optimization. This method evaluates the deformation in a coarser graph, which is more efficient than in a dense mesh. De Aguila et al. [2] generate tetrahedrons within the template model to preserve the volume during the deformation. DynamicFusion [22] uses dual quaternions to represent deformations for better quality of transformation blending. Some other methods [3, 4, 20] use multiple cameras to reduce occluded regions and improve motion tracking.

One major issue of these general deformation models is that the topology or the constraints of the model do not reflect the underlying kinematic structure of the scanned object. This often results in over-smoothed deformation or rubbery appearance. Therefore, other works are modeling the deformation field based on the piece-wise rigid articulated structure. Some of these works refine the original embedded deformation graph. Li et al. [14] adaptively refine the topology of the embedded graph by adding more nodes to the regions with larger misalignments. Guo et al. [5] apply an additional L0 optimization after the L2 optimization with the ARAP regularization to minimize the number of non-rigid node connections. The non-rigid regions are then assigned less constraint weights than the rigid regions in the regularization term. Some other works implicitly generate skeletal structure based on the analysis of the registered motion sequences. ArticulatedFusion [13] hierarchically clusters the surface into rigid segments in a bottom-top manner by minimizing a rigid registration energy function. Tzionas et al. [31] compute the deformation variance among a set of sample nodes on the surface, and apply spectral clustering on the corresponding affinity matrix to determine the rigid segments. There are also some works that explicitly define the kinematic structure of the human body for performance capture applications. DoubleFusion [35] separates the surface into two layers: the inner layer uses parametric model SMPL [17] to track the poses and correspondences, while the outer layer fuses far-body shapes. BodyFusion [34] presents a skeleton-embedded surface fusion (SSF) to join the surface graph with a skeleton structure. However, most of these works use single reference frame to constrain the deformation of the entire surface, which may cause large misalignments or tracking failure. We apply different reference frames and tracking strategies in the local coordinate of embedded nodes to address these issues.

2.2. Correspondence Estimation

The correspondence estimation is critical for the convergence of non-rigid registration. Most previous works estimate the correspondences under the Iterative Closest Point (ICP) framework [36]. Spatial partitioning data structures such as KD-trees [5] are often applied to accelerate the searching process. However, due to the high computational cost of building the data structures, these methods are often not suitable for dynamic surface tracking and interactive applications. Therefore, some other works search the correspondences in the image space of the measurements. Projective Depth Association (PDA) [23] is commonly used in surface tracking with depth cameras. This method searches for the matched point within a small window around the projection location in the depth image. However, this method is not efficient for large tangential motions which require a larger searching window. Tagliasacchi et al. [29] and Li et al. [13] accelerate the correspondence searching in depth images by using the Distance Transform (DT) of the foreground segmentation to locate the closest point on the silhouette of the surface. Moreover, Valstic et al. [32] constrain the silhouette of the canonical mesh to match that detected in the current image.

Another category is performing correspondence estimation in the embedded space. Jain et al. [9] perform principle component analysis (PCA) on the geodesic distance matrix to convert the points into embedded space to align shapes. The isometry-invariant property of the geodesic distance is demonstrated to be a significant feature for robust correspondence estimation on deformable surfaces. Motion2Fusion [3] improves the performance of the spectral embedded algorithm by developing a machine learning method to efficiently map the points to the embedded space. On the other hand, FunctionalMaps (FM) [24] define a consistent and linear mapping function between a pair of full shapes using Laplace-Beltrami eigenfunctions. Rodola et al.'s method [26] improves FM for the partial to full correspondence estimation with an additional permutation matrix. Instead, our work focuses on estimating reliable correspondences in the misaligned regions between the partially matched model and depth images using geodesic features. We use the geodesic distances precalculated in the canonical model to help evaluate the geodesic features along the incomplete topology of the depth images.

3. Motion Reconstruction

In this section, we discuss our algorithm of the non-rigid registration for motion reconstruction with single or multiple depth cameras. In Sec. 3.1, we construct a watertight canonical model from the static pose. In Sec. 3.2, we generate a volumetric embedded deformation graph to represent the local deformations and constraints of the canonical model. In Sec. 3.3, we reconstruct the motion for each frame by solving a non-linear least square optimization problem to minimize the misalignments between the deformed model and captured depth images. We discuss the issues of the traditional ARAP regularizations and derive our solution using the relative transformation between neighboring nodes. In Sec. 3.4, we describe our tracking strategies based on the tracking status of the surface regions. In Sec. 3.5, we introduce a geodesic-based correspondence estimation algorithm to solve large partial misalignments in the frame-to-frame non-rigid registration.

3.1. Canonical Model

We fuse the depth images captured from a static pose of the scanned object into a Truncated Signed Distance Field (TSDF) [8, 15] and extract the point cloud from the zero-crossing level-set of the TSDF. We then apply the Poisson reconstruction [11] on the point cloud to construct a watertight canonical model. The canonical model provides a unified topology of the deformed model across all frames in the motion tracking.

3.2. Embedded Deformation Graph

To deform the canonical model, we apply the volumetric Embedded Deformation Graph introduced in [28]. A set of embedded nodes is sampled in the volume of the watertight canonical model and a local rigid transformation $\mathbf{F} : \{(\mathbf{q}, \mathbf{g}), \mathbf{q} \in \mathbb{H}, \mathbf{g} \in \mathbb{R}^3\}$ is associated with each node. We represent rotations by using unit quaternions $\mathbf{q}$ ($\|\mathbf{q}\|_2^2 =$ 1) to reduce the number of parameters for computation and storage as in [3]. The algebras of $\mathbf{F}$ used in this paper are listed below:

$$\mathbf{F}\mathbf{v} = \mathbf{q}\mathbf{v}\mathbf{q}^* + \mathbf{g}, \mathbf{v} \in \mathbb{R}^3 \tag{1}$$

$$\mathbf{F}_1\mathbf{F}_2 = (\mathbf{q}_1\mathbf{q}_2, \mathbf{q}_1\mathbf{g}_2\mathbf{q}_1^* + \mathbf{g}_1) \tag{2}$$

$$\mathbf{F}^{-1} = (\mathbf{q}^*, -\mathbf{q}^*\mathbf{g}\mathbf{q}) \tag{3}$$

The point $\mathbf{v} \in \mathbb{R}^3$ in the canonical model $\mathcal{V}$ is deformed by the weighted transformations from the neighboring nodes $\mathcal{N}(\mathbf{v})$:

$$\mathcal{F}(\mathbf{v}; \mathbf{F}) = \sum_{i \in \mathcal{N}(\mathbf{v})} \hat{w}_i(\mathbf{v}) \mathbf{F}_i \mathbf{F}_i^{c,-1} \mathbf{v} \tag{4}$$

$\mathbf{F}^c : \{(\mathbf{q}^c, \mathbf{g}^c)\}$ is the initial node transformation in the canonical space. $\mathbf{q}^c$ is set to identity rotation by default, and $\mathbf{g}^c$ is the sample position of the embedded node. $\hat{w}_i(\mathbf{v})$ is the normalized weight using the Radial Basis Function (RBF): $\exp(-\frac{\|\mathbf{v}-\mathbf{g}_i\|^2}{2\sigma^2})$.

3.3. Energy Function

We formulate our non-rigid registration as a non-linear least square optimization problem. The complete energy function is:

$$E(\mathbf{F}) = w_{\text{data}} E_{\text{data}}(\mathbf{F}) + w_{\text{reg}} E_{\text{reg}}(\mathbf{F}) \tag{5}$$

The following subsections will describe each of these terms in details.

3.3.1 Data Term

The major objective of the template-based non-rigid registration is to minimize the distances between the deformed canonical model and the partially observed scan from the depth camera at each frame. We apply the point-to-plane distance metric [18, 25] in our data term:

$$E_{\text{data}}(\mathbf{F}) = \sum_{k \in \mathcal{C}} |(\mathcal{F}(\mathbf{v}_k; \mathbf{F}) - \mathbf{v}'_k) \cdot \mathbf{n}'_k|^2 \tag{6}$$

$\mathcal{C}$ is a set of chosen points in the canonical model that have reliable correspondences in the depth image. $\mathbf{v}'_k$ with normal $\mathbf{n}'_k$ is the correspondence point of $\mathbf{v}_k$ found in the point cloud back projected from the depth image.

3.3.2 Local Coordinate Regularization Term

To formulate the regularization term in our energy function, we first analyze the net strain energy between two embeded nodes i and j in the deformation graph. We consider a particle $\mathbf{x}$ around node j in the reference frame ϵ. $\mathbf{x}$ is located in a cubic volume Ω of size $[-r, r]$ centered at the origin of the local coordinate of node j (See Fig. 1). We measure the displacement vector of $\mathbf{x}$ from the reference frame to current frame in the local coordinate of node i. For linearly elastic materials, the rotation-invariant net strain energy between nodes i and j can thus be approximated by:

$$E_{ij} = \frac{1}{2} \int_{\Omega} \|\dot{\mathbf{F}}_{ij}\mathbf{x} - \dot{\mathbf{F}}^{\epsilon}_{ij}\mathbf{x}\|_2^2 d\mathbf{x} \tag{7}$$

where $\dot{\mathbf{F}}_{ij} = \mathbf{F}_i^{-1}\mathbf{F}_j : \{(\boldsymbol{\theta}_{ij}, \boldsymbol{\delta}_{ij}), \boldsymbol{\theta}_{ij} \in \mathbb{H}, \boldsymbol{\delta}_{ij} \in \mathbb{R}^3\}$ is the relative transformation from the local coordinate of

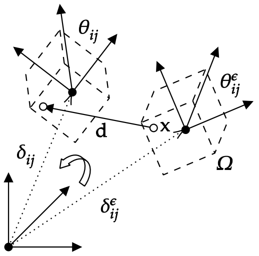

Figure 1: $\dot{\mathbf{F}}^{\epsilon}_{ij} : (\boldsymbol{\theta}^{\epsilon}_{ij}, \boldsymbol{\delta}^{\epsilon}_{ij})$ and $\dot{\mathbf{F}}_{ij} : (\boldsymbol{\theta}_{ij}, \boldsymbol{\delta}_{ij})$ are the relative transformations from the coordinate of node j to node i at reference frame ϵ and current frame respectively. Ω is a cubic volume centered at the origin in the coordinate of node j. $\mathbf{x}$ is a particle located in Ω. $\mathbf{d}$ is the displacement vector of $\mathbf{x}$ from reference frame to current frame observed in the coordinate of node i.

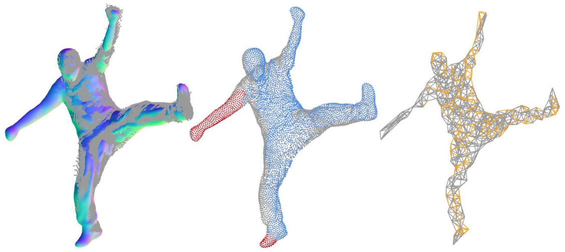

Figure 2: Left: depth image and registered deformed model. Middle: point clusters. Blue, red and gray points are tracked, untracked and occluded points respectively. Right: corresponding embedded graph. Orange and gray edges use the canonical frame and the latest tracked frame as the reference frame respectively.

node j to that of node i. Substituting Eq. 1 into Eq. 7, we obtain an analytic solution in the quaternion representation using properties of odd functions and Frobenius norms:

$$\begin{aligned} E_{ij} &= \frac{1}{2} \int_{-r}^{r} \int_{-r}^{r} \int_{-r}^{r} \| R(\boldsymbol{\theta}_{ij})\mathbf{x} - R(\boldsymbol{\theta}^{\epsilon}_{ij})\mathbf{x} + \boldsymbol{\delta}_{ij} - \boldsymbol{\delta}^{\epsilon}_{ij} \|_2^2 d\mathbf{x} \\ &= \frac{4r^5}{3} \| R(\boldsymbol{\theta}_{ij}) - R(\boldsymbol{\theta}^{\epsilon}_{ij}) \|_F^2 + 4r^3 \| \boldsymbol{\delta}_{ij} - \boldsymbol{\delta}^{\epsilon}_{ij} \|_2^2 \\ &= \frac{32r^5}{3} (1 - (\boldsymbol{\theta}_{ij} \cdot \boldsymbol{\theta}^{\epsilon}_{ij})^2) + 4r^3 \| \boldsymbol{\delta}_{ij} - \boldsymbol{\delta}^{\epsilon}_{ij} \|_2^2 \\ &\leq \frac{32r^5}{3} \| \boldsymbol{\theta}_{ij} - \boldsymbol{\theta}^{\epsilon}_{ij} \|_2^2 + 4r^3 \| \boldsymbol{\delta}_{ij} - \boldsymbol{\delta}^{\epsilon}_{ij} \|_2^2 \end{aligned} \tag{8}$$

where $R(\cdot)$ is the rotation matrix form of a quaternion. $\boldsymbol{\theta}_{ij} \cdot \boldsymbol{\theta}^{\epsilon}_{ij}$ is the dot product of $\boldsymbol{\theta}_{ij}$ and $\boldsymbol{\theta}^{\epsilon}_{ij}$. The equality of Eq. 8 holds when $\boldsymbol{\theta}_{ij} = \boldsymbol{\theta}^{\epsilon}_{ij}$. Evaluating the energy for all pairs of neighboring nodes in the graph, we obtain our regularization term so as to minimize the total elastic energy of the deformation graph:

$$E_{\text{reg}}(\mathbf{F}) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}(\mathbf{v}_i)} w_{ij} \| \mathbf{W}_{ij} (\dot{\mathbf{F}}_{ij} - \dot{\mathbf{F}}^{\epsilon}_{ij}) \|_2^2 \tag{9}$$

where w_{ij} is initialized to the RBF weight w^c_{ij} with respect to $\|\boldsymbol{\delta}^c_{ij}\|_2$. The diagonal weight matrix $\mathbf{W}_{ij}$ can be correctly derived from the coefficients of Eq. 8:

$$\mathbf{W}_{ij} = \begin{bmatrix} \sqrt{\frac{8}{3}} r \mathbf{I}_{4\times 4} & \\ & \mathbf{I}_{3\times 3} \end{bmatrix} \tag{10}$$

Here we set r to be the sampling radius of the embedded nodes. $\mathbf{I}$ is an identity matrix.

The commonly used ARAP regularization [27] is a special case of this method and has several disadvantages. First, the ARAP regularization treats nodes as particles instead of finite elements, thus it can only constrain the translation part $\boldsymbol{\delta}_{ij}$ of the relative transformation. Lacking constraints on the rotation part $\boldsymbol{\theta}_{ij}$ makes ARAP regularization unable to constrain torsional deformation around the node edges. Second, the ARAP regularization usually chooses the canonical frame or the key frame as the reference frame for all node edges in the graph, which is not optimal. For the nodes only affected by the regularization terms (e.g., occluded nodes), the transformations will drift back to the canonical poses. For visible nodes, the constraints may prevent potentially large deformation from the rest poses for joint regions. All of the issues mentioned above may cause large misalignments or tracking failure in the motion registration. In contrast, unlike ARAP, the local coordinate regularization with rotational constraints can uniquely determine the deformation field with sparse embedded nodes. This provides the capabilities and flexibilities in reconstructing the deformation field from multiple reference frames and views by manipulating the reference $\dot{\mathbf{F}}^{\epsilon}_{ij}$ in Eq. 9. Sec. 3.4 will discuss more about the strategies of selecting $\dot{\mathbf{F}}^{\epsilon}_{ij}$ in the frame-to-frame motion registration.

3.4. Tracking Strategies

To establish correspondences $\mathcal{C}$ between the canonical model and the captured depth image, we first render the deformed canonical model from the last frame to generate a new depth image in the camera space. This depth image is used to determine the visibility of the points in the canonical model to the camera. Then we perform PDA correspondence estimation in the depth image for the visible points in the deformed model. The points on the model can thus be clustered into three categories: occluded, tracked (visible and have correspondences), and untracked (visible but have no correspondences) (See Fig. 2). We can cluster the embedded nodes into the same categories by checking

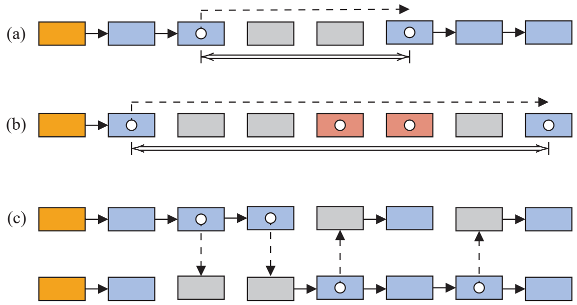

Figure 3: Illustration of the tracking strategies. Each row is a timeline of the tracking status for a node edge. Orange, blue, gray and red blocks stand for canonical, tracked, occluded, and untracked frames respectively.

the tracking status of the associated points with the node. If most of the associated points are well aligned, the estimated transformation of the node from the optimization can be considered as a reliable solution. If most of the associated points are occluded, the transformation of the node can only be guessed from previous frames. If most of the associated points have no correspondences, then we lose track of the node at this frame. This is often due to fast motion or occlusions, which needs better global correspondence estimation methods (See Sec. 3.5). We store both of the transformations $\mathbf{F}$ and the tracking status of the nodes into files after the registration of each frame. Based on the tracking status, we develop several strategies for the selection of the reference $\dot{\mathbf{F}}^\epsilon_{ij}$ to constrain the deformations in different regions.

Visible region If both nodes i and j are visible to the camera and their associated vertices have sufficiently reliable correspondences, we consider the edge between the nodes as being tracked (blue blocks in Fig. 3). For template-based motion tracking, we usually choose the relative transformation $\dot{\mathbf{F}}^c_{ij}$ from the canonical frame as the reference to constrain the deformation of tracked edges. But as mentioned in Sec. 3.3.2, this fixed ARAP constraint does not work well for non-rigid regions with large deformations. Another method always uses the relative transformation from the latest tracked frame $\dot{\mathbf{F}}^t_{ij}$ as the reference for the registration. However, this method can lead to error accumulation that changes the distribution of the vertices in the deformed mesh. Therefore, we add an L0 norm regularization term $E_{\mathbf{F}}$ in the energy function to adaptively adjust the reference frame $\dot{\mathbf{F}}^\epsilon_{ij}$ and encourage using the constraints from the canonical frame $\dot{\mathbf{F}}^c_{ij}$:

$$E_{\mathbf{F}}(\dot{\mathbf{F}}^\epsilon) = \lambda_{\mathbf{F}} \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}(\mathbf{v}_i)} w_{ij} \| \|\dot{\mathbf{F}}^\epsilon_{ij} - \dot{\mathbf{F}}^c_{ij}\|_2 \|_0 \tag{11}$$

where $\dot{\mathbf{F}}^\epsilon_{ij} \in \{\dot{\mathbf{F}}^c_{ij}, \dot{\mathbf{F}}^t_{ij}\}$. When the optimization converges, we add another L0 norm term E_w to adaptively adjust the weight w_{ij} in an additional optimization:

$$E_w(w) = \lambda_w \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}(\mathbf{v}_i)} \|w_{ij} - w^c_{ij}\|_0 \tag{12}$$

where $w_{ij} \in \{w^c_{ij}, 0.3w^c_{ij}\}$. Similar to [5], this method can refine the deformation field to be as articulated as possible and reduce the over-smoothed deformation in the joint regions.

Occluded region If the node edge is occluded or untracked (the gray and red blocks respectively in Fig. 3), we apply the relative transformation $\dot{\mathbf{F}}^\epsilon_{ij}$ from the latest tracked frame as the reference (the dashed lines in Fig. 3a and 3b). To optimize the query performance, we cache a copy of the relative transformations $\dot{\mathbf{F}}_{ij}$ across frames, and only update the cached $\dot{\mathbf{F}}_{ij}$ when the corresponding nodes are tracked in the new registered frame. This method allows the occluded surface to preserve its local deformation from previous visible frame while being transformed globally with the tracked surface. Due to the temporal coherence, there would be more opportunities for the occluded surface to be recovered when the surface becomes visible again in subsequent frames. To reduce the motion discontinuity between the occluded and tracked frames, we interpolate the transformations between tracked frames and perform another optimization for occluded nodes after the motion is reconstructed (the bidirectional arrows in Fig. 3a and 3b).

Untracked region Once the untracked nodes are recovered in subsequent frames, we apply a second-pass registration to recover the untracked nodes. Similar to the strategy used in the occluded region, we first calculate an interpolated reference frame $\dot{\mathbf{F}}^\epsilon_{ij}$. Then we only apply the regularization in the untracked region $\mathcal{P}$, and fix the position of the other nodes as hard constraints to initialize the poses:

$$E_{\text{reg}}(\mathbf{F}) = \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}(\mathbf{v}_i)} w_{ij} E^\epsilon_{ij}(\mathbf{F}), i \in \mathcal{P} \vee j \in \mathcal{P} \tag{13}$$

If the untracked surface with the interpolated deformation has sufficient overlaps with the depth image, the transformations of the untracked nodes can be recovered in the optimization.

Multi-view tracking A multi-view tracking system can significantly reduce occluded regions and second-pass tracking. However, for commodity depth cameras, such as Kinect V2, there are still some issues to address: the Kinect V2 lacks control of the camera shutters, therefore the depth images are captured at varying frequencies and phases that are not synchronized; the interference between multiple cameras also leads to larger measurement noises even for static objects. All of these issues make it difficult to integrate multi-view data into the same energy function because of the large misalignments in the global space. Therefore, we introduce a sequential registration pipeline for multiple depth cameras. We sort the frames captured from all the cameras by their timestamps, and register the sorted frames in chronological order. For occluded node edges, we choose the relative transformation from the closest tracked frame among all the camera views as the reference (See Fig. 3c). Therefore, the registered frames are first converted to the local space of neighboring nodes and then transfered to another view in the form of relative transformations to reduce the effects of global misalignments between different cameras.

3.5. Correspondence Estimation

Most of the misalignments can be solved by the frame-to-frame non-rigid registration using PDA correspondence estimation. However, due to the low capture frequency of the commodity depth cameras (e.g., 30Hz for Kinects) and the sparse camera arrangement, we can still fail to track some regions when fast motion or extended occlusion occurs. Since the captured objects are usually articulated, such as the human body, the deformation is almost isometric, which is more insensitive in the geodesic space along the surface. Therefore, we choose to convert the deformed points and the captured points to the geodesic space to perform global correspondence matching.

To be precise, we first sample a set of source points $\mathcal{P}$ on the surface of the canonical model $\mathcal{V}$. For each point $\mathbf{v} \in \mathcal{V}$, we calculate the minimum path distances between the point and the source points along the topology of the canonical surface. This defines our mapping from the Euclidean space to the geodesic feature space: $\mathbb{R}^3 \rightarrow \mathbb{R}^{|\mathcal{P}|}$. However, since the depth image is just a partial scan of the model, the connectivity on the captured surface is incomplete or ambiguous due to surface occlusion (See Fig. 4a) and contact (See Fig. 4c). Therefore, it is not feasible to compute the geodesic feature for the captured points directly from the depth image. To address this problem, we reduce the feature space to $\mathbb{R}^{|\mathcal{Q}|}$ where $\mathcal{Q}$ is a subset of $\mathcal{P}$ that excludes the source points with few reliable correspondences in the initial alignment. We consider the surface points around each source point within a fixed geodesic distance, and evaluate the proportion of surface points with projective correspondences. If the proportion is lower than a threshold, the source point is excluded from $\mathcal{Q}$. We then join the graph of the canonical model and the depth image through the aligned points. More specifically, we use the geodesic distances precalculated in the canonical model to initialize the geodesic distances of the aligned points in the depth image, and then estimate the distances in the unaligned regions based on those at the boundary of the overlapped region. This helps solve both the discontinuity and ambiguous connection in the partial graph of the depth image (See Fig. 4b and 4d). After these steps, we search for the best matched points between the unaligned sample points and captured points by comparing their L_1 distances in the sub-feature space $\mathbb{R}^{|\mathcal{Q}|}$ with a reciprocity test to reject false matching. These sparse correspondences are then applied to recover large misalignments in the registration.

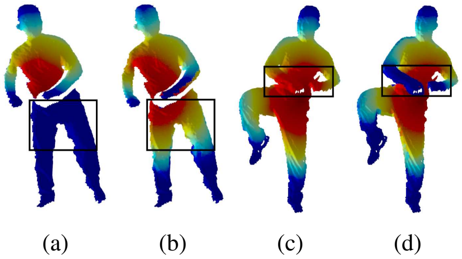

Figure 4: Illustration of the single source geodesic distance. (a) and (c) calculate geodesic distances directly from the depth image. (b) and (d) initialize geodesic distances from the canonical model.

3.6. Motion Smoothing

The reconstructed motion usually suffers from jittering deformation due to noises of measurements. Therefore, after the motion sequence is reconstructed, we smooth the motion transformations to filter out high frequency noises for visualization purpose. We calculate the weighted average of the translation $\mathbf{g}$ and the rotation $\mathbf{q}$ for frame t within a frame window of $[-n, +n]$ (See Eq. 14). $\hat{w}$ are the normalized RBF weights with respect to the time offset. The averaged quaternion $\bar{\mathbf{q}}^t$ is calculated as the eigenvector of the covariance matrix $\mathrm{cov}_{\mathbf{q}}^t$ with the maximum eigenvalue as in [6, 19]:

$$\bar{\mathbf{g}}^t = \sum_{k=t-n}^{t+n} \hat{w}_k \mathbf{g}^k, \mathrm{cov}_{\mathbf{q}}^t = \sum_{k=t-n}^{t+n} \hat{w}_k \mathbf{q}^k \mathbf{q}^{k,\top} \tag{14}$$

4. Result

This section contains experimental results of our motion tracking method with single-view and multi-view depth image sequences (600 to 3000 frames) from [5] (See Fig. 5, 6, and 7) and our capture system (See Fig. 8). We focus on comparisons between registration methods using the deformation graph as a general underlying structure without any prior knowledge of the scanned object, and demonstrate the advantages of using our local coordinate regularization.

Fig. 5 visualizes the importance of the rotation term $\boldsymbol{\theta}_{ij}$ in the local coordinate regularization. Due to the error of

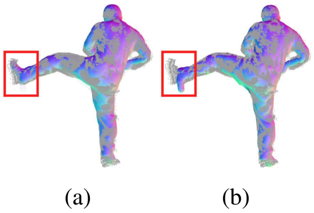

Figure 5: L2 regularizations with (a) and without (b) the rotation term θ_{ij}. Gray points are back projected from the depth image.

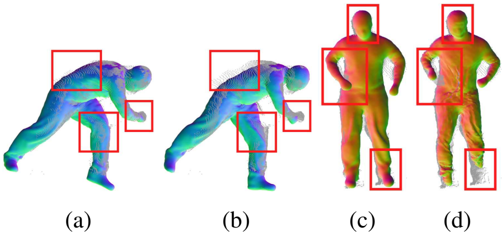

Figure 6: Comparison of using adaptive and single reference frames. (a) and (c) use adaptive reference frames. (b) only uses the canonical frame as the reference frame. (d) only uses the latest tracked frame as the reference frame.

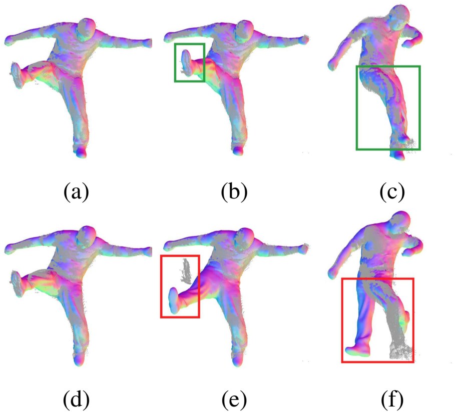

Figure 7: Comparison of different tracking strategies for the occluded regions in a motion sequence. (a), (b) and (c) use the latest tracked frame as the reference frame. (d), (e) and (f) use the canonical frame as the reference frame.

the correspondence estimation at the boundary of the surface, the registration may produce an undesirable torsional distortion in the region with fewer embedded nodes (e.g., the arms and the legs). Without the rotation term, there is no constraint to prevent this torsional distortion, and the misalignments may finally become unrecoverable (e.g., the left foot in Fig. 5b). Our regularization with the rotation term is thus more robust to these drift errors.

Fig. 6 shows the results using single and adaptive reference frame in the non-rigid registration. In Fig. 6d, we use the latest tracked frame as the reference frame for the registration. Due to the inaccuracy of alignments and error accumulation in the frame-to-frame registration, the deformed model no longer satisfies the ARAP constraints after several frames. In Fig. 6b, we use the canonical frame as the reference frame. The surface does not align well with the scans in the bending regions (e.g., the knees and the elbows). This is because the canonical reference $\dot{\mathbf{F}}_{ij}^{c}$ and the corresponding weight w_{ij}^{c} may not be optimal for large deformations. In contrast, our method with adaptive selection of reference frames can prevent error accumulation and fit the scans bet-

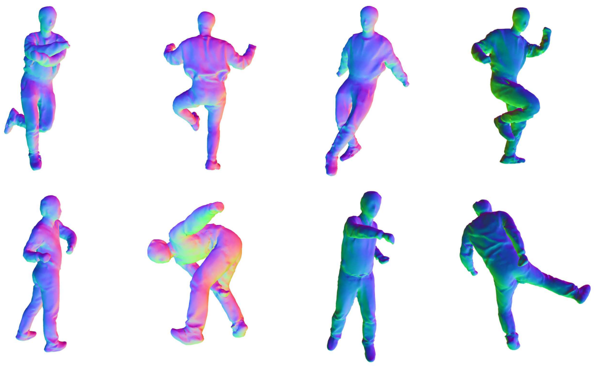

Figure 8: Results of our motion tracking method using four Kinect V2 cameras connected to one computer.

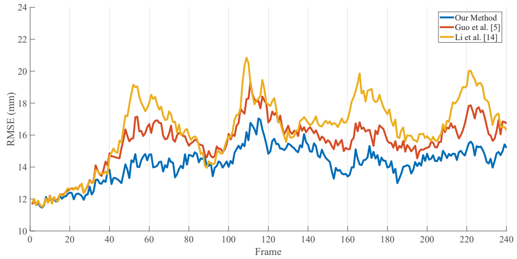

Figure 9: Quantitative comparison of single-view registration methods for kicking motions.

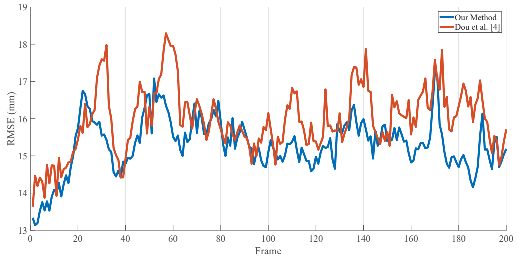

Figure 10: Quantitative comparison of multi-view registration methods for jumping motions.

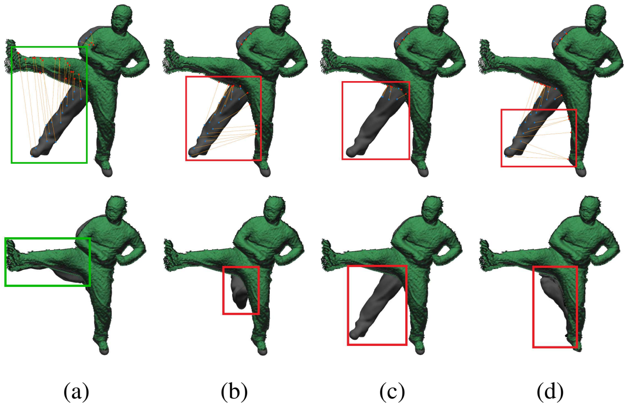

Figure 11: Comparison of correspondence estimation methods. The first and the second rows show the sparse correspondence pairs and the registration results respectively. The columns show our method, KD-trees, PDA and DT methods from left to right. The gray and green meshes are the canonical model and the captured scan respectively.

ter (See Fig. 6a and 6c). We evaluate the alignment errors and compare our method with other single-view (See Fig. 9) and multi-view (See Fig. 10) registration methods using deformation graphs. As can be seen, our method achieves lower errors for large bending motions (kicking motion) and fast motions (jumping motions).

In Fig. 7, we demonstrate our tracking strategy for the occluded regions. In the first row, we use the latest tracked frame (See Fig. 7a) as the reference frame in the occluded regions as in our method. In the second column, only the bottom of the right foot is visible to the camera, while the rest of the right leg is occluded. However, the right leg can keep its local pose and be rotated with the torso. The misalignments are then recovered after several iterations when there are more overlaps between the right foot and the depth image (See Fig. 7b). In the second row, we only use the canonical frame as the reference frame in the occluded regions. The right leg thus drifts back to the canonical pose (See Fig. 7e), which causes incorrect alignments in subsequent frames (See Fig. 7f). This tracking failure is a common issue in previous works [14, 37].

We compare our correspondence estimation method with the KD-trees [5], PDA [23] and DT [13] methods. We show the calculated correspondences for the sparse sample points in the misaligned regions and demonstrate the convergence of different methods in Fig. 11. The PDA method can only find a few correspondences around the overlapped regions due to a small searching window (See Fig. 11c). KD-trees and DT methods can find longer-range correspondences than the PDA method, but since the spatial distance based correspondences are not reliable for large non-rigid deformation, the estimation may lead to incorrect alignments (See Fig. 11b and 11d). In contrast, our geodesic-based method can find more robust correspondences following the topology of the surface (See Fig. 11a). Though the preprocessing time of our method is more than the DT and PDA methods (See Table 1), our method requires fewer iterations to converge and obtains better alignments (See Fig. 12).

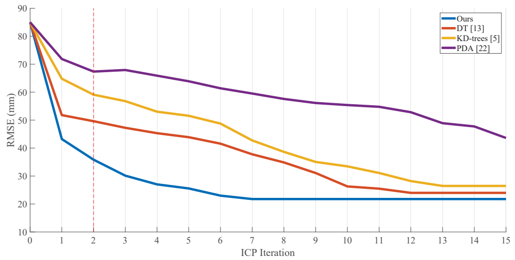

Figure 12: Convergence of different correspondence estimation methods for large misalignments. The first two iterations use estimated sparse correspondences. The rest iterations use dense correspondences with the PDA method.

Method	Ours	KD-trees	PDA	DT
Time (ms)	13	33	0	10

Table 1: Preprocessing time of correspondence estimation methods on Intel Core^{TM} i7-6900K with a single thread. Our method computes geodesic distances and corresponding KD-trees in non-overlapped regions using FLANN [21]; KD-trees build the spatial structure for the depth image; DT computes the distance map from the silhouettes of the scan.

5. Conclusion

We have presented a non-rigid registration algorithm using the local coordinate regularization to solve the misalignments in the motion tracking. We have demonstrated several tracking strategies for different regions of the surface based on the visibility and alignment of the surface. This method is shown to be effective and robust in various scenarios with single or multiple depth camera setup. We also propose a geodesic-based algorithm to efficiently locate reliable correspondences in partially aligned scans to recover the surface with large displacements. There are still difficulties in solving the ambiguous cases with self-intersecting surfaces for correspondence estimation, which would be another interesting topic.

Acknowledgements This study was supported by NSF grant CNS-1337722 and NIH grants R21HL124443 and R01HD09117.

References

[1] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230, 2007.

[2] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics (TOG)*, 27(3):98, 2008.

[3] M. Dou, P. Davidson, S. R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):246, 2017.

[4] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):114, 2016.

[5] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3083–3091, 2015.

[6] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.

[7] S. Helgason. *Differential geometry, Lie groups, and symmetric spaces*, volume 80. Academic press, 1979.

[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[9] V. Jain and H. Zhang. Robust 3d shape correspondence in the spectral domain. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 19–19. IEEE, 2006.

[10] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46. ACM, 2007.

[11] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013.

[12] H. Laga, Q. Xie, I. H. Jermyn, and A. Srivastava. Numerical inversion of srnf maps for elastic shape analysis of genus-zero surfaces. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2451–2464, 2017.

[13] C. Li, Z. Zhao, and X. Guo. Articulatedfusion: Real-time reconstruction of motion, geometry and segmentation using a single depth camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 317–332, 2018.

[14] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (ToG)*, 28(5):175, 2009.

[15] W. Li, X. Xiao, and J. Hahn. 3d reconstruction and texture optimization using a sparse set of rgb-d cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1413–1422. IEEE, 2019.

[16] Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or. Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics (TOG)*, 24(3):479–487, 2005.

[17] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015.

[18] K.-L. Low. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4, 2004.

[19] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[20] S. Meerits, D. Thomas, V. Nozick, and H. Saito. Fusionmls: Highly dynamic 3d reconstruction with consumer-grade rgb-d cameras. *Computational Visual Media*, 4(4):287–303, 2018.

[21] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[22] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[23] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[24] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012.

[25] S.-Y. Park and M. Subbarao. An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*, 24(16):2967–2976, 2003.

[26] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017.

[27] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[28] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *ACM Transactions on Graphics (TOG)*, volume 26, page 80. ACM, 2007.

[29] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, volume 34, pages 101–114. Wiley Online Library, 2015.

[30] A. B. Tumpach, H. Drira, M. Daoudi, and A. Srivastava. Gauge invariant framework for shape analysis of surfaces. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):46–59, 2015.

[31] D. Tzionas and J. Gall. Reconstructing articulated rigged models from rgb-d videos. In *European Conference on Computer Vision*, pages 620–633. Springer, 2016.

[32] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM,

2008.

[33] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems, 2015.

[34] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 910–919, 2017.

[35] T. Yu, Z. Zheng, K. Guo, J. Zhao, Q. Dai, H. Li, G. Pons-Moll, and Y. Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, 2018.

[36] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994.

[37] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4):156, 2014.