# Repeatable Local Coordinate Frames for 3D Human Motion Tracking: from Rigid to Non-Rigid

Chun-Hao Huang[1] Federico Tombari[1,2] Nassir Navab[1,3]

[1] Technische Universität München, [2] Università di Bologna, [3] Johns Hopkins University

## Abstract

*Local coordinate frame (LCF) is a key component deployed in most 3D descriptors for invariant representations of 3D surfaces. This paper addresses the problem of attaching a LCF to non-rigidly deforming objects, in particular humanoid surfaces, with the application of recovering correspondences between the template model and input data for 3D human motion tracking. We facilitate this by extending two current LCF paradigms for rigid surface matching to the non-rigid case. Such an adaptation is motivated by the assumption that interpolating locally rigid movements often amounts to smooth globally non-rigid deformations. Both approaches leverage spatial distributions, based on signed distance and principal component analysis, respectively. Furthermore, we advocate a new strategy that incorporates multiple LCF candidates. This way we relax the requirement of perfectly repeatable LCFs, and yet still achieve improved data-model associations. Ground truth for non-rigid LCFs are synthetically generated by interpolating locally-rigidly transformed LCFs. Therefore, the proposed methods can be evaluated extensively in terms of repeatability of LCFs, robustness on estimating correspondences, and accuracy of final tracking results. All the experiments demonstrate the benefits of the proposed methods with respect to the state-of-the-art.*

## 1. Introduction

Top-down human motion tracking is the process of recovering temporal evolutions of humanoid template surfaces using visual information, such as image silhouettes or 3D points. Due to its recent success in marker-less human motion capture (mocap), the field of applications ranges from computer vision [9] and computer graphics [24] to medical imaging [13]. In the 3D domain, input data is represented either by visual hulls from the 3D reconstruction, or by point clouds obtained from range sensors. The tracking process generally consists of two steps. One first estimates the correspondences between the data and the model as in Fig. 1(a). Given the estimated data-model associations, the template surface is deformed and fit to the observed 3D points accordingly.

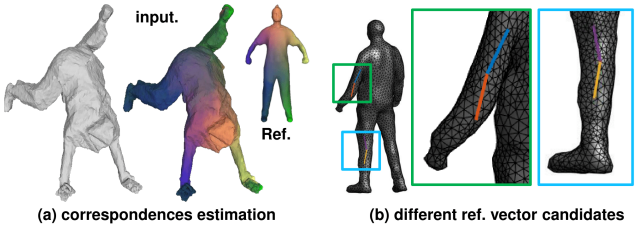



Figure 1: Coupled with forests as learning techniques, our LCF approach provides plausible correspondences as in (a). The invariance are achieved by considering multiple reference vectors that result in multiple LCFs as shown in (b).

When the input data is severely deformed with respect to the available template, discovering correspondences between them is not a trivial task. Several approaches [9, 22] employ machine learning strategies, *e.g.* random forests [3], to learn off-line from the training data, and then identify correspondences directly on-line during tracking. In order to learn with 3D meshes, the local geometry of each vertex has to be described in advance. In the Graphics community, several descriptors [1, 4, 21] have been designed for generic deformable shape matching purposes. However, the computational overhead as well as the sensitivity to noisy data, which is common in visual observations, prevent them from being applied for 3D human tracking. On the other hand, a different class of 3D descriptors has been developed in the Vision community, such as [6, 14, 15, 19, 23, 25] , with the goal of matching rigid surfaces in noisy point clouds. One essential trait of these methods is the employment of *Local Coordinate Frames* (LCF): by attaching a LCF to each point or vertex and describing the local neighborhood with respect to the LCF, feature representations can be invariant to rotations. The efficacy of this strategy is recently confirmed again in [9], where they represent meshes in volumes and apply regression forests to discriminatively dis-

cover dense correspondences. In this case, LCFs play the role of selecting neighboring voxels adaptively. Ideally, a good LCF is supposed to follow whatever transformations the meshes undergo, namely, as *co-variant* as possible, such that the consequent representations are as *invariant* as possible. Nevertheless, misalignments are inevitable. To dense data-model associations in human tracking, one has to properly handle the trade-off between efficient-but-loosely-attached LCFs and complex-but-repeatable LCFs.

In this aspect, the contribution of this paper is three-fold: firstly, we exploit the assumption that surfaces tend to deform smoothly in space, and, as a result, dense non-rigid deformation can be largely approximated by an ensemble of locally rigid motions [5]. Thus, we explore LCF methods used for rigid surface matching and adapt them for non-rigid humanoid surfaces, attaining more stable LCFs than the one in [9]. Secondly, we incorporate multiple LCFs resulting from distinct reference vectors as in Fig. 1(b) and propose a more reliable representation that yields more accurate correspondences. Finally, to evaluate our methods, we again leverage the locally rigid approximation and generate ground truth LCFs for non-rigid humanoid meshes by means of blending techniques.

## 2. Related work

This paper aims at developing a LCF-based approach used for correspondence estimation in 3D human motion tracking. In the following, the previous work is therefore briefly reviewed and discussed from the two different perspectives of LCFs and 3D human motion tracking.

### 2.1. Local Coordinate Frame

LCFs are usually proposed with their 3D descriptor counterparts. Here, we provide an overview, whereas a comprehensive review and evaluation is available in [16]. Constructing a LCF consists in defining three orthonormal vectors as $[x, y, z]$ axes. To this end, the local geometry has to be taken into account, involving all neighboring points $\mathbf{p}_i$ (hereinafter referred to as *support*) that lie within a sphere of a certain radius centered at the feature point $\mathbf{p}$. Current approaches can be broadly classified into eigenvalue-decomposition-based [14, 23], which establish three axes at once, and methods that identify them one by one separately [6, 9, 16, 17, 25].

The first category relies on computing the Principal Component Analysis (PCA) or EigenValue Decomposition (EVD) of the local spatial distribution of points. In [14], a covariance (or *scatter*) matrix is constructed as:

$$\mathbf{\Sigma} = \frac{1}{k}\sum_{i=0}^{k}(\mathbf{p}_i - \mathbf{p}')(\mathbf{p}_i - \mathbf{p}')^\top, \qquad (1)$$

where $\mathbf{p}'$ is the centroid of $k+1$ support points $\mathbf{p}_i$. Later in [23], the centroid $\mathbf{p}'$ is replaced with the feature point $\mathbf{p}$ itself for higher efficiency. The contribution of each support point $\mathbf{p}_i$ to the covariance matrix is also weighted by its Euclidean distance to $\mathbf{p}$. The three axes are provided by the three normalized eigenvectors obtained from scatter matrix decomposition. Conventionally, the one with the largest eigenvalue (principal direction) is defined as the $x$ axis, while the one with the smallest eigenvalue is considered as the $z$ axis. One of the major issue of these methods is that EVD defines only the directions of the axes but not their signs, which have to be disambiguated with additional efforts. For instance, the sign of $z$ axis is usually the one that yields positive inner product with the surface normal $\mathbf{n}$.

Another family of work defines three axes individually [6, 9, 16, 17, 25]. Typically, $y$ axis is attained as $z \times x$ to keep the orthogonality constraint. $z$ axis is either the surface normal vector itself [9], the averaged surface normal [25] across the 5-ring neighborhood, or the normal of a fitted plane within a smaller support [6, 16, 17], where the signs are again disambiguated by considering surface normals. More efforts are devoted to facilitate a stable $x$ axis. Some approaches [9, 16, 25] rely on higher order information (normals or curvatures) to identify prominent geometry and determine the orientations of $x$ axis. Instead of high order information which is prone to be noisy in visual 3D data, [6, 17] considers the signed distances of each support point to the tangent plane defined by $z$ axis, and yields more repeatability as demonstrated in [17].

### 2.2. Correspondences in human motion tracking

Different strategies exist for the correspondence task. Assuming close initializations, some approaches [5, 8, 20] discover the associations by spatial proximity and then refine the correspondences by iterating between the association step and the parameter estimation step. These generative approaches can be regarded as the extensions of Iterative Closest Point (ICP) algorithm [2]. Since in the tracking scenario, observations in successive frames usually lie in vicinity, these methods often initialize using the previous frames, and the results are satisfactory if the proximity assumption holds. Nonetheless, they also inherit the shortcomings of ICP, which are the slow convergence (because of its iterative nature), and the error accumulation (due to the heavy dependency on the previous outcome).

In contrast, discriminative approaches [9, 12, 22] utilize formerly observed meshes as training data off-line, in order to quickly establish the associations on-line. They make use of machine learning techniques such as randomized trees [9, 22] or Support Vector Machine [12]. Since the predictions are made frame-wise, they are in general less prone to drift. However, as a downside shared among all learning frameworks, they tend to fail on the unseen input data. In other words, the pool of training meshes is suggested to be

sufficiently large to include all possible deformations.

Note anyway that these two strategies are actually complementary to each other. One can always initialize discriminatively in each frame and then refine the results generatively with a few ICP iterations, as shown in [18] and [9] for 2.5D point clouds and full 3D visual hulls, respectively. In this work, we are particularly interested in [9] because it has demonstrated reasonably good correspondences using features that deploy LCFs for non-rigid surfaces.

# 3. Method

## 3.1. Preliminaries and Overview

We first provide a overview of [9] and then outline our method. A human surface is denoted as $\mathcal{M} = (\mathbf{M}, \mathcal{T})$, where $\mathbf{M} = \{\mathbf{x}_v\}_{v=1}^{N_v} \subset \mathbb{R}^3$ are the locations of vertices $v$, and $\mathcal{T}$ defines the triangles. To attach a LCF for each vertex $v$, Huang *et al*. consider its normal $\mathbf{n}_v$ as $z$ axis, and search for a reference vector in a local cuboid to establish $x$ axis. Formally, the surface is first voxelized into a volumetric field $\mathbf{N} : \Omega_3 \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$, where each voxel $\mathbf{v}$ holds either a unit-length normal $\mathbf{n}$ averaged from the containing triangles, or a number indicating inside or outside:

$$\mathbf{N}(\mathbf{v}) = \begin{cases} +\boldsymbol{\epsilon} & \text{if lies outside surfaces} \\ \mathbf{n} \in [-1, +1]^3 & \text{if overlap with surfaces} \\ -\boldsymbol{\epsilon} & \text{if lies inside surfaces.} \end{cases} \quad (2)$$

A vertex $v$ is first mapped to a voxel $\mathbf{v}_v$[1], by discretization of the space. We consider a cubic support of neighbors centered on the voxel $\mathbf{v}_v$, $\mathcal{S} \subset \Omega_3$, as depicted in Fig 2(a). A surface voxel $\mathbf{v}$ is selected based on the following criteria:

$$\hat{\mathbf{v}} = \arg\max_{\mathbf{v} \in \mathcal{S}} \left( \hat{d} + \mathbf{N}(\mathbf{v}_v)^\top \mathbf{N}(\mathbf{v}) \right). \quad (3)$$

where $\hat{d}$ is the distance between $\mathbf{v}_v$ and $\mathbf{v}$, normalized with respect to the size of cuboids. The projection of $(\hat{\mathbf{v}} - \mathbf{v})$ onto the plane define by $\mathbf{n}_v$ is then taken as $x$ axis of the LCF. Finally, the $y$ axis is obtained as $z \times x$.

We retain the volumetric framework to keep the property of organized data, *i.e*., accessing spatial neighbors simply by indexing without iteratively parsing the triangles like nearest neighbor search. We also consider $\mathbf{n}_v$ as $z$ axis, and obtain $y$ axis as $z \times x$. Differently, we pay more attention on the characteristic voxel/vector for $x$ axis. Eq. 3 favors the voxel that is far from the center voxel $\mathbf{v}_v$, and yet holds least normal changes. As an important trait of our approach, since it is well known that higher order information, *e.g*. normals and curvatures in visual 3D data, are particularly noise prone, we resort to rely on robust zeroth-order information such as spatial distributions.

[1] With a slight abuse of notations, in the remainder of this paper, $\mathbf{v}$ refers only to voxels that overlap with meshes (intersected with either vertices or triangles), since the other two cases, inside and outside, are both not of our interest. In particular, $\mathbf{v}_v$ refers to voxels containing mesh vertices.

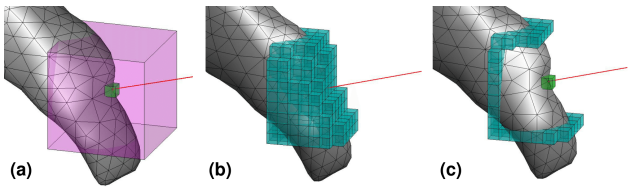



Figure 2: (a) visualizes the volumetric framework. Green: voxel $\mathbf{v}_v$ mapped by the current vertex $v$; pink: cuboid neighborhood; red line: normal vector. (b) The LCF method in [9] and EVD involve all the surface voxels (cyan) within the cuboid, whereas (c) SignDist. considers only those lying on the border. Best viewed in colors and in pdf.

## 3.2. LCF Proposals

Analogous to what proposed in [9], the state-of-the-art methods in the field of LCFs for rigid matching of 3D meshes and point clouds mainly rely on the neighboring points within a local support [6, 14, 15, 17, 23]. As reviewed in Section 2, the way they leverage spatial distributions can in general be classified into two categories: (1) EigenValue-Decomposition (EVD) [14, 15, 23], and (2) signed distance (SignDist.) [6, 17]. In the following, we propose an adaptation of both classes of methods to volumetric representations, so to be able to use them within the human motion tracking framework.

**EVD.** Methods within this class define the LCF as the principal directions of the point distribution within the support. Since the $z$ axis is already defined, we project all the $N_{\mathbf{v}}$ support surface voxels $\mathbf{v}$ onto the plane defined by $\mathbf{n}_v$, denoted as $\bar{\mathbf{v}}$. This way the resulting vectors defining the principal directions lie naturally on the $xy$ plane. It is given by the normalized eigenvectors of the covariance matrix:

$$\boldsymbol{\Sigma}_{\mathcal{S}} = \frac{1}{N_{\mathbf{v}} - 1} \sum_{\bar{\mathbf{v}} \in \bar{\mathcal{S}}} (\bar{\mathbf{v}} - \mathbf{v}_v)(\bar{\mathbf{v}} - \mathbf{v}_v)^\top, \quad (4)$$

where $\bar{\mathcal{S}}$ is the projection of all surface voxels falling within the support. The centroid of $\bar{\mathcal{S}}$ is replaced with the voxel $\mathbf{v}_v$ itself to speed up the computation, without decreasing much repeatability as in [23]. The eigenvector of largest eigenvalue is chosen as $x$ axis. Note that at this point, the sign of the $x$ axis is not uniquely determined, due to the inherent ambiguity of the sign of the eigenvectors obtained from the EVD process [23]. Because of this, the computed LCF might flip $180°$ along the $z$ axis. Later in Sect. 4, we will propose a specific feature to tackle this undesired effect, so to make the overall approach invariant to such ambiguity.

**SignDist.** This class of approaches look for a discerning point within the support. As contrasted in Fig. 2(b-c), the search involves typically only the peripheral points $\tilde{\mathbf{v}}$ lying on the intersection of the cuboid border and the surface, unlike EVD-based method where all points contribute to the covariance matrix. The discernibility is defined as the maximum signed distance to the tangent plane [6].

We propose to adapt this idea to volumetric representations as follows:

$$\hat{\mathbf{v}} = \arg\max_{\tilde{\mathbf{v}} \in \tilde{\mathcal{S}}} \left( (\tilde{\mathbf{v}} - \mathbf{v}_v)^\top \mathbf{n}_v \right). \tag{5}$$

where $\tilde{\mathcal{S}}$ is the intersection of the surface and the border of local cuboids. The $x$ axis is the projection of the vector directed from $\mathbf{v}_v$ towards $\hat{\mathbf{v}}$. Note that there is no guarantee that the discerning point $\hat{\mathbf{v}}$ from Eq. 5 is always repeatable: in particular, if different directions yield similar values of the signed distance, the $x$ axis will be ambiguous, hence the resulting LCFs could rotate about the $z$ axis. Purposely, we incorporate such an ambiguity in our new proposed features to make our method robust to this phenomenon, as explained in the next section.

## 4. Learning with multiple LCFs

As shown later in Section 5.2, the presented volumetric adaptations of EVD (Eq. 4) and signed distance (Eq. 5) generally yield a higher repeatability than the LCF originally employed in [9], despite the ambiguity associated to the $x$ axis of both approaches highlighted in the previous section. Recall that the employment of LCFs is to introduce invariance to 3D rotations of the deployed features, so as to make the subsequent learning tasks easier. In these regards, as intuitively easy to understand, computing locally defined features which are not rotation invariant might have a detrimental influence on learning.

However, a perfectly repeatable LCF is not only intractable but also unnecessary. Conventionally, people pursue repeatability to a certain extent, and let the learning algorithms deal with the noise resulting from misalignments. On the contrary, we persist in the ability of being invariant, not through a stable LCF but via the derived representations itself. Specifically, our main idea is, instead of devising a robust $x$ axis, to consider all possible distinctive candidates - this yielding multiple LCFs associated to the same voxel - and aggregate the corresponding feature values by averaging them. This way we reduce the noise left for the learning, and thereby increase the overall robustness of the method. In the remaining of this section, we take the task in [9] as an example, and demonstrate how to learn with multiple LCFs.

### 4.1. Learning framework

The task is to learn a mapping $\mathbf{Y} : \Omega_3 \rightarrow \mathbb{R}^3$ that takes a vertex voxel $\mathbf{v}_v$ as input, and regress to a 3D point $\mathbf{Y}(\mathbf{v}_v)$, denoted as $\mathbf{y}_v$. The mapping has a co-domain $\mathbb{R}^3$, but the range $\mathbf{Y}(\mathbf{v}_v)$ is only a 2-manifold subspace defined by the vertex positions on a template surface. Such a mapping facilitates correspondence search between input meshes and the template, because $\mathbf{y}_v$ implies the locations of possible matches of $v$ on the template. Using regression forests [7], the mapping is learned in advance with many voxelized meshes that share the same topology.

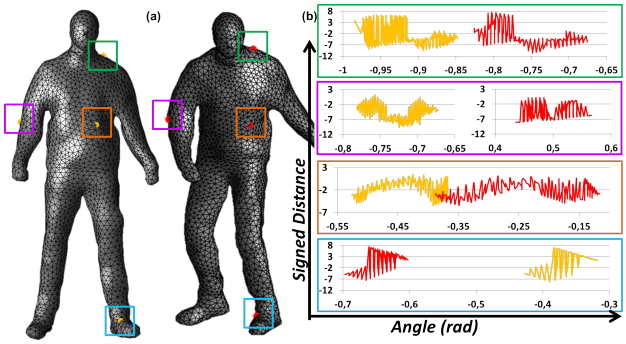



Figure 3: Signed distances in the local neighborhood. When subject changes the pose, the shape of the profile remains roughly unchanged, but presents a phase shift. See text for more explanations. Best viewed in colors and in pdf.

We follow the features in [9], describing the local geometry of each voxel $\mathbf{v}_v$ by looking at their neighbors, represented by an offset pair $\psi = (\mathbf{o}_1, \mathbf{o}_2) \in \Omega_3 \times \Omega_3$. To achieve pose invariance, the neighbors have to be aligned to a LCF, namely, orienting $\psi$ with a rotational matrix $\mathbf{R}$ defined by the three axes of LCFs. The feature vector $\mathbf{f}(\mathbf{v}_v; \mathbf{R}(\psi))$ takes the adjusted neighbors as parameters, and consider their dot product of normals, the subtraction of volumetric field $\mathbf{N}$ within local cuboids, and other operations in different feature channels. During training, at each branch node, many pairs of $\psi$ are randomly generated, aligned to LCFs, and each channel of the resulting feature vectors $\mathbf{f}$ are thresholded. Among all the combinations of neighbors $\psi$, channel indices, and thresholding values, the one that maximizes the information gain are saved. The trees keep growing recursively until stopping criteria are met. We refer interested readers to [7] and [9] for the theories of regression forests, and more explanations on the feature $\mathbf{f}$, respectively.

### 4.2. Averaged representation

The ambiguity of the $x$ axis in both the EVD as well as sign distance case leads to different LCFs, rotating around the $z$ axis (*i.e.* the vertex normal $\mathbf{n}_v$). Obviously, each LCF results in a different pair of neighbors $\mathbf{R}(\psi)$, and hence a different feature vector $\mathbf{f}$. Here, we first recall the source of ambiguities for $x$ axis as analyzed in Subsection 3.2, and then propose a way to incorporate them systematically into the averaged feature $\mathbf{f}'$.

| Sequence | Frames | Metric | Subject / #Vertex |
|---|---|---|---|
| *Crane* [24] | 173 | A, B C | S1 / 3407 |
| *Jumping* [24] | 149 | | |
| *Handstand* [24] | 149 | | S2 / 3848 |
| *Bouncing* [24] | 173 | | |
| *Cutting* [11] | 81 | | S3 / 5211 |
| *Hammer* [11] | 93 | | |

Table 1: Sequences used in our experiments. Depending on the evaluation tasks, we apply different error measures. A: deviated angle. B: repeatability score as defined in [17]. C: vertex index for correspondences.

**EVD.** The major concern of EVD-based methods is that the sign of the principal component is ambiguous. For this reason, every time we compute a feature vector $\mathbf{f}$, we consider two LCFs, obtained by using the same $x$ axis but with opposite signs, *i.e.* $x^+$ and $x^-$. In turn, this results in the computation of two feature vectors $\mathbf{f}^+$ and $\mathbf{f}^-$. The final representation is the average of the two: $\mathbf{f}' = (\mathbf{f}^+ + \mathbf{f}^-)/2$. In this way, we only require the direction of the $x$ axis to be repeatable, while we are completely independent from possible ambiguities about its sign.

**SignDist.** In this class of methods, the $x$-axis ambiguity stems from the variations of discerning points in the neighborhood. When described as the Point Signature [6], the profile of local geometry varies since the object move non-rigidly. Still, it is worthwhile taking a closer look.

Fig. 3(a) shows a subject in two poses, where we select several vertex correspondence pairs, highlighted in squares with different colors. Fig. 3(b) further depicts the profiles of these pairs in yellow (left mesh) and red (right mesh) curves, respectively. One can see that when the subject deforms, the shapes of signatures remain largely similar, but with a phase shift. The essence of SignDist-based method is to identify the shifts, and reflect them on the rotations of $x$ axes along $z$ axes. Nevertheless, as signatures appear to be noisy, it is difficult to capture the shifts solely depending on the voxel element yielding the global maximum of Eq. 5.

The possible remedy is, besides the voxel $\hat{\mathbf{v}}$ yielding the global maximum, to take into account also voxels yielding the highest local maxima of the signed distance function, *i.e.* a set of local maximizers $\{\hat{\mathbf{v}}^i\}_{i=1}^{L}$. In our implementation, such set is computed by retaining all border voxels whose signed distance is, after non-maxima suppression, above a certain threshold, up to a maximum number of $L$ elements. Given such a set of local maximizers, including the global one, and the corresponding feature vectors $\mathbf{f}^i$ computed from the associated LCFs, the final feature is then obtained as:

$$\mathbf{f}' = \frac{1}{L}\sum_i \mathbf{f}^i \tag{6}$$

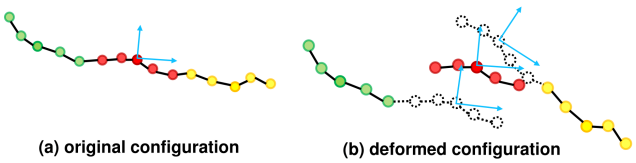



Figure 4: Ground truth LCFs are generated synthetically. A LCF is first computed in the original configuration in (a). When the mesh deforms, the new LCF is the linear combination of the predicted LCFs from the neighboring patches.

In this way, if the majority of maximizers are repeatable in presence of noise, the computed feature vector will also be repeatable, independently from their relative order.

## 5. Experimental results

In this section, we evaluate our approach under two aspects. First, we verify the repeatability of the proposed LCF methods by measuring how much they deviate from the ground truth. Secondly, we demonstrate the benefits of averaged LCF features for 3D human motion tracking by measuring how much they improve the correspondence task between input data and the template surface. The profiles of our sequences are summarized in Table 1.

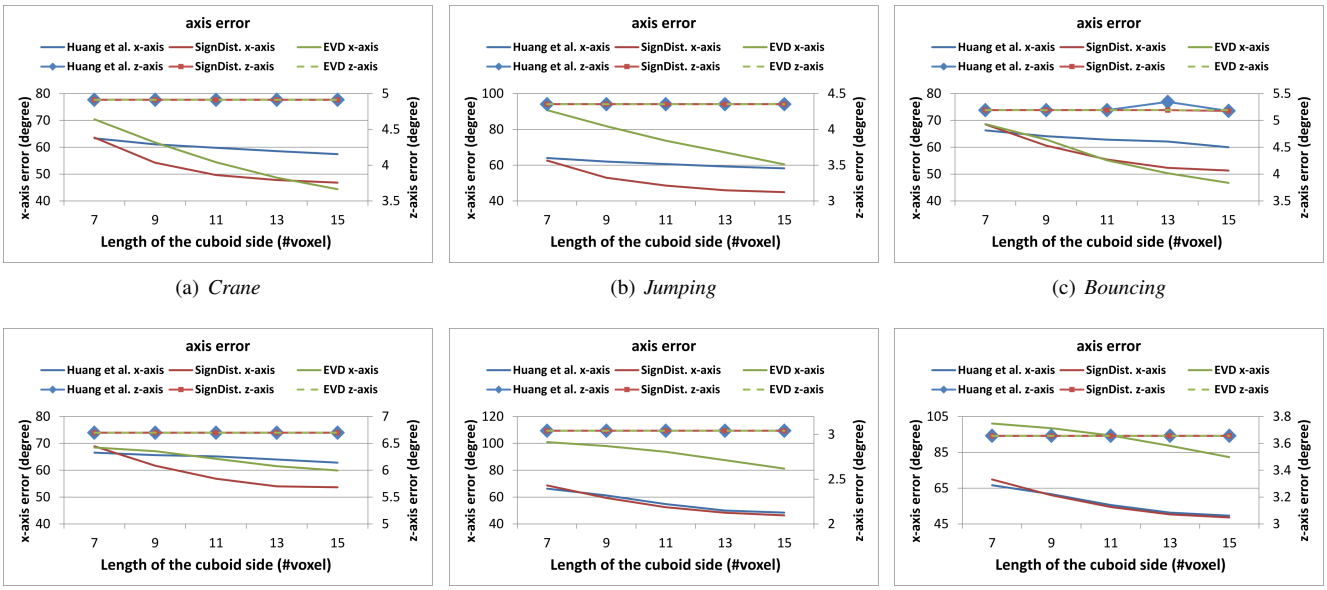
### 5.1. Ground truth generation

Due to the lack of ground truth LCFs from real visual data, we resort to synthetic transformations. For each method to be evaluated, we first compute a LCF for each vertex on the reference mesh $\mathcal{M}^0$, denoted as $LCF^0$ and depicted in Fig. 4(a). The mesh is then animated with a data-driven patch-based approach [5], and the goal is to see if the newly obtained LCFs follow such transformations. This deformation framework models global non-rigidness as a sparse set of control bases called *patches* that move locally rigidly. The animation is done by tracking with real visual hulls as input, so that we have realistic deformations.

Specifically, a mesh is decomposed into several small patches. Each patch $k$ has a rigid body motion $(\mathbf{R}_k, \mathbf{t}_k)$. When the mesh $\mathcal{M}^0$ deforms into $\mathcal{M}^t$, the new vertex position is the linear combination of its transformed location and all the predictions from the neighboring patches $\mathcal{N}_k$, visualized as dot circles in Fig. 4(b). The final position of each vertex is determined by interpolating their predicted locations from the neighboring patches, where the coefficients $\alpha_k$ encode the desired physical property and are normalized to sum up to 1. More details can be found in [5]. The attached $LCF^0$ follows the same operations:

$$LCF_k = \mathbf{R}_k LCF^0, \tag{7}$$

$$LCF' = \sum_{s \in k \cup \mathcal{N}_k} \alpha_s LCF_s. \tag{8}$$




(a) *Crane* (b) *Jumping* (c) *Bouncing*

(d) *Handstand* (e) *Hammer* (f) *Cutting*

Figure 5: Error reported by the three evaluated LCF methods with varying cuboid sizes, measured in degrees. Left vertical axes: error of $x$ axes; right vertical axes: error of $z$ axes.

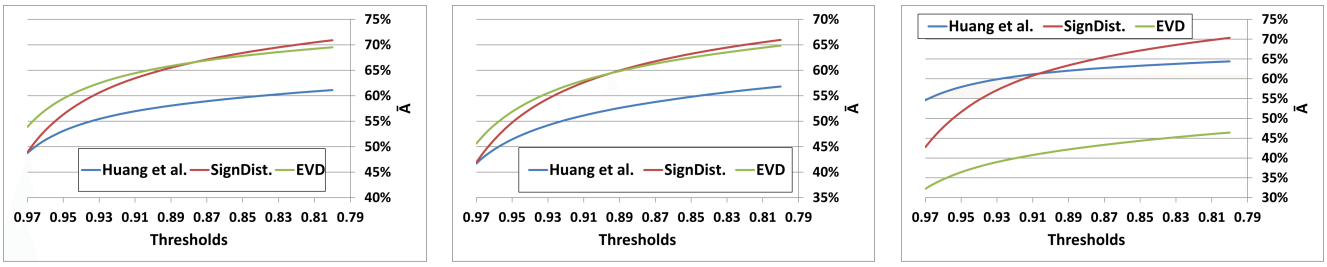
In Eq. 7, it is rotated according to either its own rigid body motion, or those from the neighboring patches. In Eq. 8, multiple predictions, *e.g.* blue LCFs in Fig. 4(b), are blended to yield the final local coordinate frame $LCF'$ (all axes normalized to unit length), which is considered as ground truth in our experiments.

## 5.2. LCF Repeatability

We compute our LCF methods, *i.e.* EVD (Eq. 4) and SignDist (Eq. 5), for all vertices on $\mathcal{M}^t$, and check respectively their discrepancies to the ground truth $LCF'$ by computing cosine scores as in [16]. Fig. 5 shows the results in degrees for each sequences, aggregated by averaging across every frame. For $z$-axes, all the approaches yield consistently low errors regardless the radius of the cuboid. For $x$-axes, on the other hand, the differences are more obvious. We see that the error suggests a monotonous decreasing trend when the support gets larger. EVD has relatively poor performance with small radius, due to the fact that the support is insufficient to fully characterize the local point distributions. SignDist. shares the same concern in smallest cuboid size 7, but the error drops faster than EVD and attains most of the time the best results also with respect to [9]. In the remainder of this section, we always use the largest cuboid size 15 for comparisons and analysis.

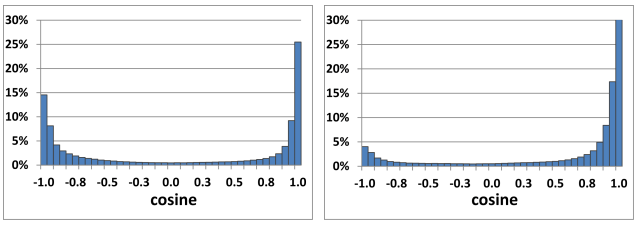
We also apply the repeatability score $\bar{A}$ in [17], which is defined as the number of points whose cosine scores are higher than a certain threshold $T_A$, measured in percentage and averaged across every frames in the sequence. The scores in varying thresholds are presented in Fig. 6. It actually reflects the angle errors in Fig. 5 faithfully. SignDist. attains better (S1 and S2) or comparable (S3) repeatability compared with [9]. This confirms that spatial distributions are more reliable in 3D data than higher order information such as normals used in [9]. On the other hand, we notice that EVD does not always present such a merit (S3), which is worth further investigation.

To understand why EVD performs badly in S3, we look into the cosine scores for $x$-axes, and plot the histogram in Fig. 7(a). One can clearly see that a considerable portion of vertices have scores less than $-0.8$, presenting a bimodal distribution. This indicates that EVD method indeed suffers from the sign-ambiguity and explains the low repeatability score. It verifies the motivations of averaged representation in Section 4, whose effectiveness is demonstrated in the next sub-section. For SignDist, the similar problem also exists but, as indicated in Fig. 7(b), is much less severe. In Fig. 8, we visualize for each vertex of a 3D mesh taken from the test sequences the angle error of $x$-axes in colors and show the distribution of cosine scores for each method. Overall, we can qualitatively observe how both SignDist. and EVD attain in average better cosine scores than [9]. Interestingly, while SignDist. has a more scattered error, EVD shows a more piece-wise distribution (compare, *e.g.*, the left leg with the right leg), this highlighting the previously discussed disambiguation problem affecting EVD-




(a) S1: *Crane*, *Jumping* (b) S2: *Bouncing*, *Handstand* (c) S3: *HammerTable*, *Cutting*

Figure 6: Repeatability scores $\bar{A}$ on three subjects in varying thresholds $T_A$. Cuboid size: 15.




(a) EVD (b) SignDist.

Figure 7: The distributions of cosine scores of $x$ axes for two different methods on *Hammer* and *Cutting*.

based LCFs. In addition, the plotted cosine distributions in Fig. 8 again confirm the aforementioned discussions.

### 5.3. Correspondences prediction

In addition to previous results, we investigate how different LCF strategies influence surface matching in terms of accuracy of the retrieved correspondences. The task is to find correspondences between input meshes and the reference template $\mathcal{M}^0$. For each subject, we learn separate regression forests using 3 LCF methods (length of cuboid side 15): Huang *et al*. [9], SignDist. (Eq. 5), and EVD (Eq. 4) respectively. For the latter two cases, we also consider the proposed extensions to averaged representation. To draw a fair comparison, all the other parameters remain the same: 20 trees, 15000 testing neighbor pairs $\psi$ at each branch node, and maximum tree depth 20. We consider the animated meshes $\mathcal{M}^t$ as input data, where ground truth vertex indices are available. The error measure here is the geodesic distances on template surfaces $\mathcal{M}^0$.

If the geodesic distance between the estimated and ground truth vertex indices is lower than a certain threshold, we consider it as a correct match. Fig. 9 shows the percentage of right matches in varying thresholds. We highlight half of the length of lower arms in orange dashed line for better interpretation of the estimated correspondences. Two observations can be remarked. First of all, among the 3 approaches, SignDist. always yields more than $90\%$ of matches that falls within the range of half of lower arms and achieve best results. Secondly, the averaged representations are consistently better than their counterparts. These observations confirm that the proposed LCF methods improve the state-of-the-art and furthermore, the strategy of averaged LCF results in more invariant representations, and hence better matching accuracy.

Although Heat Kernel Signature (HKS) [21] and Wave Kernel Signature (WKS) [1] are rarely used for human motion tracking, as a complement to our results above, we anyway evaluate them on our data since they are common baselines for generic deformable surface matching. We extract HKS/WKS descriptors for all vertices on the input mesh, and look on the reference surface for the vertex that has the closest signature in the feature space. The results are presented in Fig. 10. Clearly, our method (averaged SignDist.) attains better accuracy. The qualitative prediction results are also shown in Fig. 11. One sees that, as expected, HKS performs poorly on noisy data, whereas our method (averaged SignDist.) again achieves visually more plausible correspondences than HKS and [9]. We remark that the comparisons with HKS/WKS here is only to help one interpret how good the reported numbers are in the context of shape matching, not to draw thorough comparisons. We apply user-specific forests as decision mechanisms, whereas for HKS/WKS it is only nearest neighbor search.

Last but not least, since the end application is human mocap, we evaluate the tracking results as well. Given the data-model associations from the forests trained with different LCF approaches, we deform the template surfaces using the method similar to [10]. Differently, we do not run ICP refinements, so as to see the direct impact of correspondences in the end results. The metric is the silhouette overlap error that measures the discrepancy between the deformed reference surfaces and the silhouettes. As reported in Fig. 12, the averaged LCF method always yield lower errors than their counterparts. Moreover, both presented LCF methods improve the tracking compared to the LCF in [10].

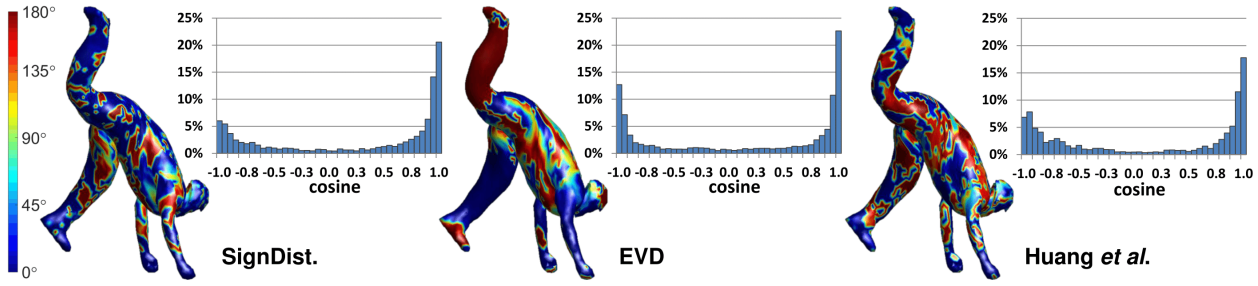
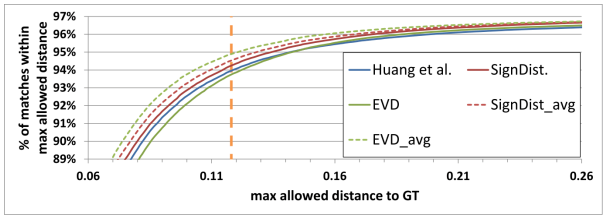
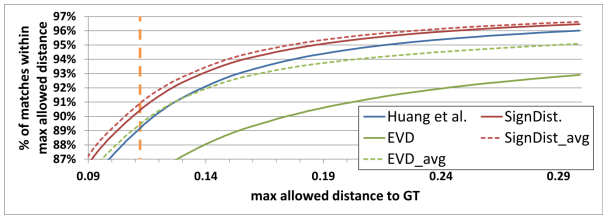


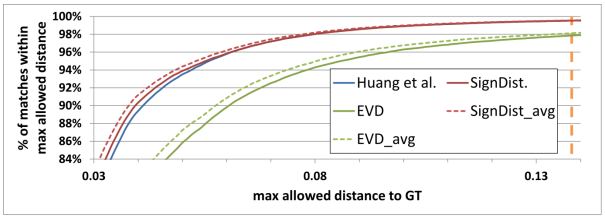

Figure 8: Qualitative results on *Handstand*. From left to right, the averaged cosine score is 0.626, 0.565, 0.538, respectively.




(a) S1: *Crane*, *Jumping*




(b) S2: *Bouncing*, *Handstand*




(c) S3: *Hammer*, *Cutting*

Figure 9: Comparison of the correspondence error from different LCF strategies for all subjects.

## 6. Conclusion

In this paper, we propose a method attaching LCFs to non-rigidly deforming surfaces, with the goal to facilitate correspondence tasks in 3D human motion tracking. The non-rigidness of human motions is approximated as the interpolations of several locally-rigid motions. We then adapt two LCF paradigms for rigid surface matching to the non-rigid case. In addition, we incorporate the sources of unrepeatability in learning, and present a more invariant representation, sparing the efforts of devising robust LCFs and yet maintaining the descriptiveness of features. Ground truth LCFs are also produced locally rigidly. Our methods are thereby evaluated thoroughly, and the experiments suggest that the proposed LCFs attain higher repeatability than the state-of-the-art approaches, the new representations from multiple LCFs yield improved correspondences than their counterparts, and, in turn, better tracking results.

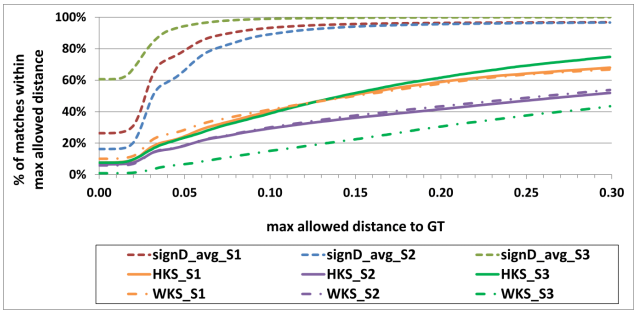



Figure 10: Comparison of the correspondence error from various descriptors for all subjects.

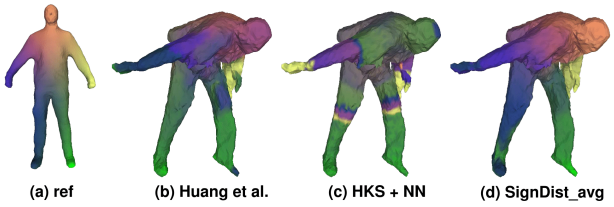
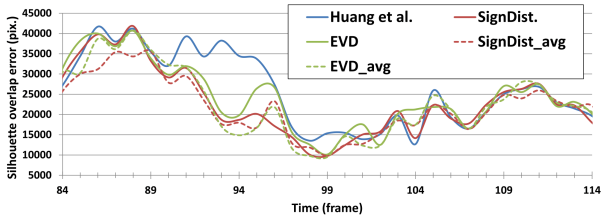



Figure 11: Visual comparisons of correspondences.




| Huang et al. | SignDist. | EVD | SignDist_avg | EVD_avg |
|---|---|---|---|---|
| 14065 | 13091 | 13705 | **12799** | 12863 |

Figure 12: Comparisons of tracking error on *Jumping*. The table shows the averaged silhouette overlap error across all frames and all cameras. Image resolution: $1920 \times 1080$.

## References


[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*. IEEE, 2011.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. In *PAMI*. IEEE, 1992.

[3] L. Breiman. Random forests. In *Machine learning*, volume 45, pages 5–32. Springer, 2001.

[4] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR*, pages 1704–1711. IEEE, 2010.

[5] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. In *ECCV*. Springer, 2010.

[6] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. In *IJCV*. Springer, 1997.

[7] A. Criminisi and J. Shotton. *Decision forests for computer vision and medical image analysis*. Springer, 2013.

[8] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*. IEEE, 2009.

[9] C.-H. Huang, E. Boyer, B. do Canto Angonese, N. Navab, and S. Ilic. Toward user-specific tracking by detection of human shapes in multi-cameras. In *CVPR*. IEEE, June 2015.

[10] C.-H. Huang, E. Boyer, and S. Ilic. Robust human body shape and pose tracking. In *3DV*. IEEE, 2013.

[11] C.-H. Huang, E. Boyer, N. Navab, and S. Ilic. Human shape and pose tracking using keyframes. In *CVPR*. IEEE, 2014.

[12] A. Kanaujia, N. Kittens, and N. Ramanathan. Part segmentation of visual hull for 3d human pose estimation. In *CVPR Workshop*. IEEE, 2013.

[13] A. Ladikos, C. Cagniart, R. Ghotbi, M. Reiser, and N. Navab. Estimating radiation exposure in interventional environments. In *MICCAI*, pages 237–244. Springer, 2010.

[14] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. Springer, 2010.

[15] J. Novatnack and K. Nishino. Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In *ECCV*. Springer, 2008.

[16] A. Petrelli and L. Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *ICCV*. IEEE, 2011.

[17] A. Petrelli and L. Di Stefano. A repeatable and efficient canonical reference for surface matching. In *3DimPVT*. IEEE, 2012.

[18] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric regression forests for human pose estimation. In *BMVC*, 2013.

[19] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*. IEEE, 2009.

[20] M. Straka, S. Hauswiesner, M. Rüther, and H. Bischof. Simultaneous shape and pose adaption of articulated models using linear optimization. In *ECCV*. Springer, 2012.

[21] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[22] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *CVPR*. IEEE, 2012.

[23] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*. Springer, 2010.

[24] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *TOG*. ACM, 2008.

[25] A. Zaharescu, E. Boyer, and R. Horaud. Keypoints and local descriptors of scalar functions on 2d manifolds. In *IJCV*. Springer, 2012.