Tracking-by-Detection of 3D Human Shapes: from Surfaces to Volumes

Chun Hao Huang, Benjamin Allain, Edmond Boyer, Jean-Sébastien Franco, Federico Tombari, Nassir Navab, Slobodan Ilic

Tracking-by-Detection of 3D Human Shapes: from Surfaces to Volumes

Chun-Hao Huang, Benjamin Allain, Edmond Boyer, Jean-Sébastien Franco, Federico Tombari, Nassir Navab and Slobodan Ilic

Abstract—3D Human shape tracking consists in fitting a template model to temporal sequences of visual observations. It usually comprises an association step, that finds correspondences between the model and the input data, and a deformation step, that fits the model to the observations given correspondences. Most current approaches follow the Iterative-Closest-Point (ICP) paradigm, where the association step is carried out by searching for the nearest neighbors. It fails when large deformations occur and errors in the association tend to propagate over time. In this paper, we propose a discriminative alternative for the association, that leverages random forests to infer correspondences in one shot. Regardless the choice of shape parameterizations, being surface or volumetric meshes, we convert 3D shapes to volumetric distance fields and thereby design features to train the forest. We investigate two ways to draw volumetric samples: voxels of regular grids and cells from Centroidal Voronoi Tessellation (CVT). While the former consumes considerable memory and in turn limits us to learn only subject-specific correspondences, the latter yields much less memory footprint by compactly tessellating the interior space of a shape with optimal discretization. This facilitates the use of larger cross-subject training databases, generalizes to different human subjects and hence results in less overfitting and better detection. The discriminative correspondences are successfully integrated to both surface and volumetric deformation frameworks that recover human shape poses, which we refer to as 'tracking-by-detection of 3D human shapes.' It allows for large deformations and prevents tracking errors from being accumulated. When combined with ICP for refinement, it proves to yield better accuracy in registration and more stability when tracking over time. Evaluations on existing datasets demonstrate the benefits with respect to the state-of-the-art.

Index Terms—Shape tracking, random forest, centroidal Voronoi tessellation, 3D tracking-by-detection, discriminative associations.

1 Introduction

3D shape tracking is the process of recovering temporal evolutions of a template shape using visual information, such as images or 3D points. It finds applications in several domains including computer vision, graphics and medical imaging. In particular, it has recently demonstrated a good success in marker-less human motion capture (mocap). Numerous approaches assume a user-specific reference surface, with the objective to recover the skeletal poses [1], surface shapes [2], or both simultaneously [3]. A standard tracking process consists in an alternation of the following two steps. First, finding associations between the observed data, *e.g.* 3D points of the reconstructed visual hull, to the corresponding 3D template shape, typically based on the proximity in Euclidean space or a feature space. Second, given such associations, recovering the pose of the template under the constraint of a deformation model, typically based on the kinematic skeleton [1], [4], [5], [6], or the piecewise-rigid surface [2] parameterization, among others.

Most of these model-based methods can be viewed as extensions of Iterative-Closest-Point (ICP) framework [7], [8] to deformable shapes, which attempts to explain newly observed data using the previous outcomes. As long as the initialization is close to the optimum solution, it is able to produce outstanding results. However, they also suffer from inherent weaknesses of *generative* strategies, *e.g.* slow convergence. Moreover, when large deformations or many outliers occur, discovering associations becomes particularly difficult. Unreliable correspondences result in ambiguous situations that yield erroneous numerical solutions.

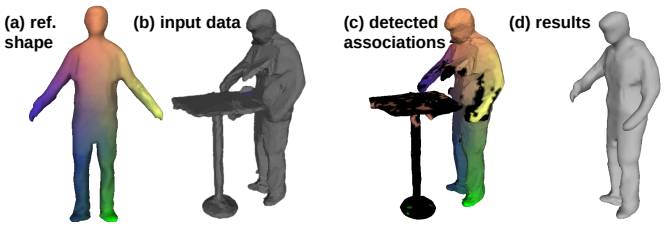


Fig. 1. Given a reference shape (a) and input data (b), our method discovers reliable data-model correspondences by random forests, color-coded in (c). This strategy detects user-specific shapes in a frame-wise manner, resulting in better sustainability. In (d) the reference model (a) is deformed with correspondences (c) to fit the input data (b).

Recently, a number of alternatives and enhancements have been explored for both association and deformation stages independently. On one hand, improvements have also been proposed for the association problem by discovering them discriminatively [6], [9], [16], This in turn yields the possibility for 3D tracking techniques that are robust to failure. In contrast to those generative ICP variants, these *discriminative* approaches that 'detect' rather than track models have shown better robustness over the past decade, for instance, in human pose estimation with 2.5D data from Kinect [6], [10]. These approaches usually consider foreground human subjects to be pre-segmented, which is not a

- *C.-H. Huang, F. Tombari and N. Navab are are with Computer Aided Medical Procedures, Technische Universität München, Germany, E-mail: see http://campar.in.tum.de/WebHome*
- *E. Boyer, J.-S. Franco and B. Allain are with Inria, LJK, France E-mail: see http://morpheo.inrialpes.fr/*
- *S. Ilic is with Siemens AG, Munich, Germany E-mail: see http://campar.in.tum.de/WebHome*

Manuscript received XX.XX, 2017; revised XX.XX, 2017.

favorable assumption in full 3D data that generally contains substantial amount of outliers like Fig. 1(b). Including non-human objects into the reference shape so that more points are explained, *i.e.* less outliers, is one workaround adopted by many existing multi-view methods [17], [18], with the downside that further post-processing is required to analyze only humans' movements. There is a growing need to facilitate robust frame-wise observation-model associations for reconstructed complete 3D human shapes. Although surface-based features are commonly used for this purpose in the context of shape matching [9], volumetric features have also proven to be a promising direction for 3D shape description with surface-based templates [11].

On the other hand, progress has also been made in the deformation stage by introducing volumetric deformation models instead of purely surface-based ones, mainly motivated by the observation that human movements are largely volume-preserving. It has shown significantly improved robustness to various tracking situations, such as shape folding and volume bias of observed shapes [12]. As volumetric deformation models are gradually used in capturing actors' motions due to their inherent local volume-preserving properties, facilitating volumetric discriminative correspondences can be favorable. We investigate this direction and make the following two contributions in this paper.

First, two volumetric features are designed for *human shape correspondence detection*, operating respectively on surface and volumetric meshes. Inspired by Taylor *et al.* [6], we apply regression forests to improve the associations, with two learning strategies devised for different shape parameterizations. In the case of surface mesh representations, we convert shapes to the volumetric Truncated Signed Distance Field (TSDF) [13], where each surface vertex is fed into user-specific forests to predict correspondences in *one shot*. Meanwhile, we also tessellate both the observed and template shapes as a set of uniform and anisotropic cells (see Fig. 2) from Centroidal Voronoi Tessellation (CVT) [14] and, again leverage the similar distance-transform representations to predict volumetric correspondences for all CVT cells.

Second, by integrating these one-shot associations into the respective deformation models, we further present a discriminative human mocap framework, as depicted in Fig. 1, termed tracking-by-detection of 3D human shapes. In contrast to the ICP-like methods [2], [3], [4], it does not require close initializations from a nearby frame to estimate correspondences and thus better handles large deformations. Experiments demonstrate that, when combined with a generative tracking approach, this *hybrid* framework leads to better or comparable results than purely generative ones, *e.g.* [2], [15], reducing error accumulations and hence increasing the stability. The regression entropy is also augmented with the classification one to identify outliers. Very few prior arts afford the tracking or matching situation where the input describes mainly irrelevant outliers. Notably, in the case of CVT, our method is a unified volumetric pipeline where the shape representation, deformation model, feature description, and points association are all built on a single CVT representation that brings benefits at all stages of the pipeline. This fully volumetric tracking-by-detection method shows improved accuracy and memory performance compared to the surface-based counterpart [11].

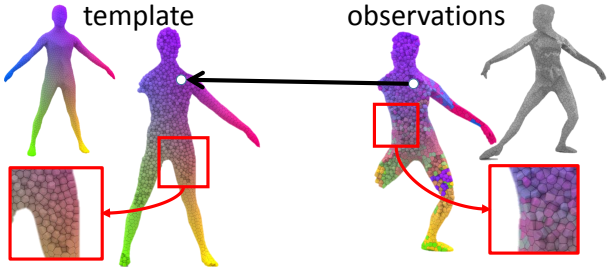


Fig. 2. Centroidal Voronoi tessellations yields volumetric cells of uniform shape and connectivity with controllable complexity. The cells of the observed shape are matched discriminatively to those of the template.

2 Related Work

Among the vast literature on human motion analysis [19], we focus on top-down approaches that assume a 3D template and deform it according to input data, either directly with pixels [4], [20], or with computed 3D points [2], [3], [15]. These methods typically decompose into two major steps: (1) data association, where observations are associated to the model, and (2) deformation stage, where motion parameters are estimated given the associations. As our primary objective in this paper is to improve the first part, existing approaches are discussed accordingly below.

2.1 Generative approaches

Methods of this category follow the association strategy in ICP while extending the motion model to more general deformations than the one in the original method [7], [8]. Correspondences are addressed by searching for closest points, with various distance measures such as point-to-point [2], point-to-plane [21], or Mahalanobis distances [20]. This strategy heavily relies on the fact that observations in consecutive frames are in vicinity. Klaudiny *et al.* [22], Huang *et al.* [3] and Collet *et al.* [17] generalize the idea from the previous frame to a certain key-frame in the considered sequences, finding the best non-sequential order to track, but the proximity assumption remains. On the other hand, since 3D data such as reconstructed point clouds often contain spurious fake geometries, another challenge consists in identifying online and dynamically irrelevant observations without any prior knowledge. Liu *et al.* [4] establish 3D-2D correspondences by considering both texture in images and contours in silhouettes and further include image segmentation information to differentiate multiple interacting subjects. Huang *et al.* [2], [3] relax the hard correspondence constraint to soft assignments and introduce an additional outlier class to reject noisy observations. Data is explained by Gaussian Mixture Models (GMM) in an Expectation-Maximization (EM) manner [23]. In [24], both source and target points are similarly modeled as GMMs and the registration problem is cast as minimizing the distance between two mixture models. Collet *et al.* [17] fuse information from various modalities attentively to generate high-quality textured meshes. Yet, to yield a temporal coherent mesh tessellation, the underlying tracking component is still ICP-based [25]. All these generative methods are highly likely to fail in large deformations. Furthermore, they are prone to error accumulations and, as a result of matching several successive frames wrongly (whether sequentially or not), they are prone to drift.

2.2 Discriminative approaches and 3D descriptors

Recently, discriminative approaches have demonstrated their strengths in estimating human [6], [26] and hand [27] poses from depth images. With the initial intention to substitute ICP-based optimization, Taylor *et al.* [6] propose a frame-wise strategy that yields decent dense correspondences without iterative refinements. The method replaces the step of proximity search in ICP-based tracking methods by learning the mapping from input 3D points from depth sensors, to the human template surface domain, termed the Vitruvian manifold. Later, Pons-Moll *et al.* [5] train forests with a new objective on surface manifolds, and increase the precision by finishing convergence with an ICP-based loop after the discriminative association stage. Both approaches operate frame-independently and are generally drift free. Following the same weak pair-wise features and random forest framework, Dou *et al.* [18] learn to match two successive depth frames to avoid depending on a specific template.

More informative descriptors and matching strategies have long been studied for shape recognition or retrieval with meshes [28] and point clouds [29]. The well known heat kernel signatures (HKS) [30] and wave kernel signatures (WKS) [31] exploit the Laplacian-Beltrami operator, the extension of the Laplacian operator to surface embeddings. Rodola *et al.* [9] later apply forests to learn the parameters of WKS during training. These features are nonetheless known for their lack of resilience to significant topology changes, an artifact frequently seen in noisy surface acquisitions. MeshHoG [32] and SHOT [33] attach a local coordinate frame at each point to achieve invariant representations and reach better performance for noisy surfaces. To enforce consistent matches over the whole shape, Chen and Koltun [34] and Starck *et al.* [35] formulate the matching problem as the inference of Markov random field (MRF).

Besides hand-crafted features, there is a recent trend that applies Convolutional Neural Network (CNN) [36] to discover the deep representation of non-rigid human shapes. Wei *et al.* [16] render depth images in several viewpoints, where the CNN feature transformation takes place, and average the descriptors from multiple views. Boscaini *et al.* [37] stay in 3D space but define the convolution function in the intrinsic manifold domain. While showing encouraging results in handling missing data, these methods do not consider matching human shapes in the presence of large amount of outliers, *e.g.* un-subtracted furniture in the background, and thus do not fit to our 'detection' purpose.

Another common trait of the aforementioned approaches is that the computation involves only surface points. We show in our early work [11] that surface features can be built based on local coordinate frames in a regular-grid volume. In this paper, we not only improve this feature but also propose a new one to address the need of fully volumetric correspondences. Both features, implicitly or explicitly, leverage distance-transform volumes to describe 3D geometry. Taking only surface vertices into account, the existing approaches rely on heterogeneous shape representations, deformation models, target primitives and feature spaces. Instead, our CVT-based tracking-by-detection proposal builds a unified framework for all these purposes and takes advantage of volumetric tracking strategies.

3 Overview

We implement discriminative associations using two different volumetric representations. In the first case, we convert the triangular surface meshes to the Truncated Signed Distance Field (TSDF) constructed with the regular 3D volumetric grid. In the second case, we use CVT representation which is not bound to the regular grids. As in Fig. 2, the interior space of a triangular surface is tessellated into a set of *cells* of uniform anisotropic shape whose seed location coincides with its centers of mass. Such an optimal discretization yields lower memory footprint than regular-grid volumes, in turn accommodating more training meshes. Moreover, we also associate CVT cells discriminatively and present volumetric correspondences.

Formally, a humanoid shape describes a continuous volumetric domain in 3D $\Omega \subset \mathbb{R}^3$ whose border $\partial\Omega$ defines a 2-manifold surface. The discretized mesh representation $\mathcal{M}$ contains a set of 3D points $\mathbf{M}$ and their connectivity $\mathcal{T}$, *i.e.* $\mathcal{M} = (\mathbf{M}, \mathcal{T})$, where $\mathbf{M}$ is drawn from the surface ($\mathbf{M} \subset \partial\Omega$) or the whole volume ($\mathbf{M} \subset \Omega$). The goal of 3D shape tracking is to register a source reference[1] mesh $\mathcal{X} = (\mathbf{X}, \mathcal{T}_X)$ to the observed target mesh $\mathcal{Y} = (\mathbf{Y}, \mathcal{T}_Y)$, such as fitting the shape in Fig. 1(a) to the one in Fig. 1(b).

Our method starts with surface meshes reconstructed by shape-from-silhouette method [38]. We refer only to points on surfaces as *vertices* $v \in \mathcal{V}$, where $\mathcal{V}$ is the set of their indices. Suppose the reference surface $\mathcal{X}$ and the input visual hull $\mathcal{Y}$ are located at $\mathbf{X} = \{\mathbf{x}_v\}_{v \in \mathcal{V}_X}$ and $\mathbf{Y} = \{\mathbf{y}_i\}_{i \in \mathcal{V}_Y}$[2] respectively, the registration typically boils down to two steps: (1) *association*: matching each points in $\mathcal{Y}$ with those in $\mathcal{X}$ to build the correspondence set $\mathcal{C} = \{(i, v)\} \subset \mathcal{V}_Y \times \mathcal{V}_X$; and (2) *optimization*: estimating the motion parameter $\mathbf{\Theta}$ by minimizing an energy E that describes the discrepancies between pairs in $\mathcal{C}$, *i.e.* $\hat{\mathbf{\Theta}} = \operatorname{argmin}_{\mathbf{\Theta}} E(\mathbf{\Theta}; \mathcal{C})$, such that $\mathbf{X}(\hat{\mathbf{\Theta}})$ resembles $\mathbf{Y}$ as much as possible.

To discover the correspondences $\mathcal{C}$ discriminatively, we adapt the *Vitruvian* strategy [6] from matching 2.5D against 3D to 3D against 3D. This amounts to *warping* the input mesh $\mathcal{Y}$ to the reference one $\mathcal{X}$, denoted as $\tilde{\mathcal{Y}} = (\tilde{\mathbf{Y}}, \mathcal{T}_Y) = (\mathbf{r}(\mathbf{Y}), \mathcal{T}_Y)$ where $\mathbf{r}$ is the warping function. A good $\mathbf{r}$ shall lead to a clean warp $\tilde{\mathcal{Y}}$ as in Fig. 3. Incorrect warped points, however, can still be told from huge edges. Specifically, this $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ mapping $\mathbf{r}$ is learned by a regression forest [39]. We convert each surface into an implicit representation, a distance field, which is usually defined volumetrically. As stated above, we investigate two ways to define the volumetric elements s. The first one is a voxel from a regular axis-aligned volume, *i.e.* $s \in \mathbb{N}^3$, while the second one is a cell from a volumetric mesh, *i.e.* $s \in \mathcal{S}$, where $\mathcal{S}$ is a group of CVT cells that tessellate only the surface interiors. Depending on the choice of s, our volumetric feature $\mathbf{f}$ is hence also realized in two different forms. Taking the feature $\mathbf{f}$ as input, multiple binary decision trees are trained with previously observed meshes. In the online testing phase, a input point obtains a prediction $\tilde{\mathbf{y}}_i = \mathbf{r}(\mathbf{y}_i)$ that indicates the locations of potential matches since the warp $\tilde{\mathbf{Y}}$ is

1. Several terms are used interchangeably in this paper: reference and template; correspondences and associations; point and primitive.

2. The observations are always indexed by i regardless of the parameterization.

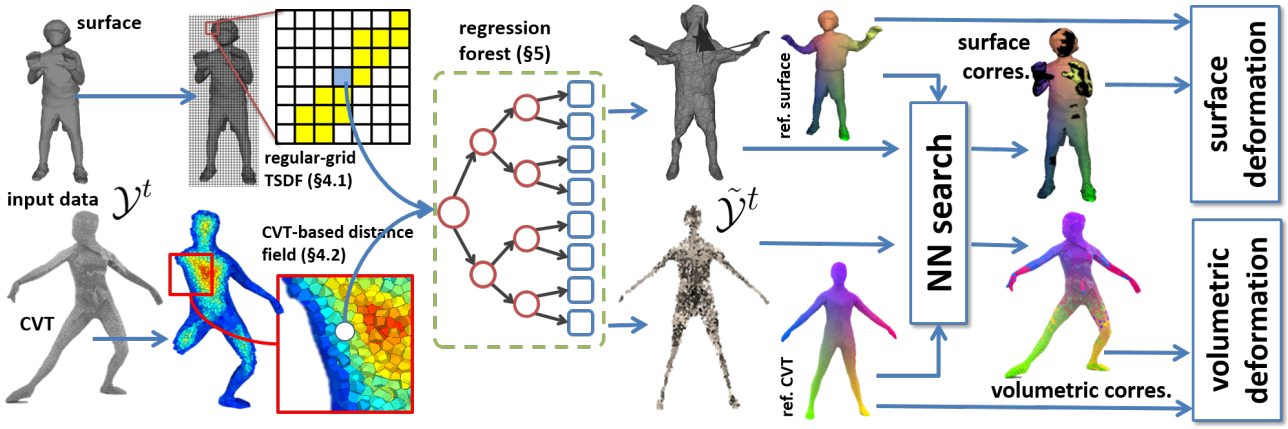


Fig. 3. The pipeline of our tracking-by-detection framework. Data-model associations are visualized in the same color. Upper row: surface-based associations (black means no correspondence found for that vertex); bottom row: volumetric associations.

learned to resemble $\mathbf{X}$. Thus, $\mathcal{C}$ can be built swiftly by doing nearest neighbor search between $\tilde{\mathbf{Y}}$ and $\mathbf{X}$ just once and the deformation parameter $\mathbf{\Theta}$ that encodes the *shape pose* of the template is estimated accordingly. Notably, in the case of CVT, since cells comprise a volumetric mesh, the whole pipeline (discovering $\mathcal{C}$ and estimating $\mathbf{\Theta}$) can instead be conducted in a fully volumetric fashion. Fig. 3 illustrates this *correspondence detection* process. The details of training, prediction and deformation models are provided in § 5.

4 Volumetric Features

The two volumetric features are introduced in this section. Although both taking a volumetric point s as input, the first one actually aims to match surface vertices v, denoted as $\mathbf{f}(v) \coloneqq \mathbf{f}(s_v)$ while the second one matches s directly, *i.e.* $\mathbf{f}(s)$. Both are designed to be incorporated into forest training and prediction. A great advantage of decision trees is to learn the most discerning attributes among a large feature bank. One does not have to prepare the whole high-dimensional vector $\mathbf{f}$ to draw predictions, because only a few learned attributes κ are needed to traverse the trees. As a result, features can be computed *on the fly* during testing. To make use of such property, the calculation of each f_κ is assumed to independent. We hence avoid the histogram-based descriptors that requires normalization, such as MeshHOG [32] or SHOT [33], and resort to offset comparison features used in [40] for $\mathbf{f}(s_v)$ and Haar feature in [41] for $\mathbf{f}(s)$.

4.1 Regular-voxel-based features

Our first approach to discriminative associations considers regular-grid volumes (upper row in Fig. 3, $s \in \mathbb{N}^3$). The warping function $\mathbf{r}$ is modeled as a composite one: $\mathbf{r} : \mathbb{R}^3 \to \mathbb{N}^3 \to \mathbb{R}^3$, where the former is voxelization and the regression trees account for only the latter. We first cast each mesh $\mathcal{M}$ into a volumetric scalar field $D : \mathbb{N}^3 \subset \mathbb{R}^3 \to \mathbb{R}$.

4.1.1 Truncated signed distance transform (TSDT)

Voxelizing a surface in general comprises two parts: (1) determining which voxel s that every vertex v maps to, and (2) testing the overlap between triangles and voxels. The first part can be viewed as a quantization mapping from Euclidean space to a discretized space $s : \mathbb{R}^3 \to \mathbb{N}^3$. The size of the volume is large enough to include all possible pose variations, and its center is aligned with the barycenter of the surfaces. The voxel size is chosen to be close to the average edge length of meshes, so that a single voxel is not mapped by too many vertices. To check the intersection of triangles with voxels, we apply *separating axis theorem* which is known to be efficient for collision detection [42].

Voxels occupied by the surface are referred to as s_{suf}. We further identify voxels located inside and outside the surface, denoted respectively as s_{in} and s_{out}. Together they define a directional truncated signed distance transform:

$$D(s) = \begin{cases} +\epsilon & \text{if } s_{\text{out}} \text{ and } d(s,\mathcal{M}) > \epsilon. \\ +d(s,\mathcal{M}) & \text{if } s_{\text{out}} \text{ and } d(s,\mathcal{M}) \leq \epsilon. \\ 0 & \text{if } s_{\text{suf}} \\ -d(s,\mathcal{M}) & \text{if } s_{\text{in}} \text{ and } d(s,\mathcal{M}) \leq \epsilon. \\ -\epsilon & \text{if } s_{\text{in}} \text{ and } d(s,\mathcal{M}) > \epsilon. \end{cases} \quad (1)$$

$d(s,\mathcal{M})$ denotes the shortest Euclidean distance from the voxel center to the mesh, which can be computed efficiently via AABB trees using `CGAL` library. If the distance is bigger than a threshold ϵ, we store only $\pm\epsilon$ to indicate the inside/outside information. It is empirically set to be three times the physical length of diagonal of voxels. In the earlier version of this work [11], we store averaged surface normals at each s_{suf}. However, such representations yield high memory footprint and thus limit the amount of training meshes we can incorporate later in § 5. The TSDT representation naturally encodes the spatial occupancies of a mesh and the required memory footprint is only one-third of the former (each voxel stores now just a scalar, not a vector). It shares a similar spirit with implicit surface representations, *e.g.* level-set, and has been widely employed in RGBD-based tracking or reconstruction [43], [44].

4.1.2 Pair-wise offset features

Next, we present the features $\mathbf{f}$ for describing TSDT, which are later used to train the forests. Since we are interested in predicting correspondences for vertices instead of triangles, from now on we concentrate only on those surface voxels s_{suf} occupied by mesh vertices v, denoted as s_v. The feature is thus defined as a function of s_v, *i.e.* $\mathbf{f}(v) \coloneqq \mathbf{f}(s_v)$.

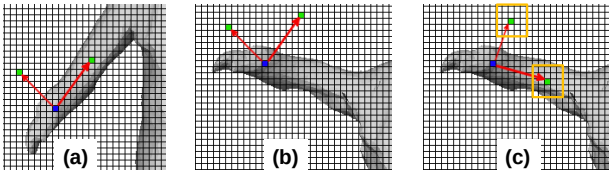


Fig. 4. The intuition of adjusting offsets. (a) original offset pair ψ. (b) $\eta = 0$ results in ψ without re-orientation, *i.e.* $\mathbf{R} = \mathbf{I}$. (c) $\eta = 1$. ψ is orientated by a rotation matrix $\mathbf{R} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ characterized by a LCF.

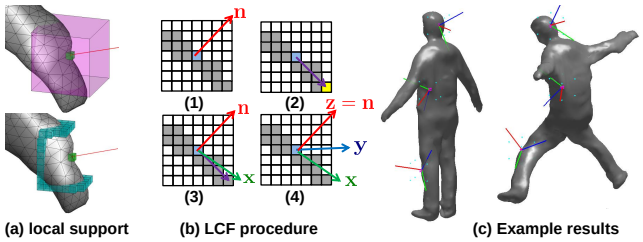


Fig. 5. Our method leads to quasi pose-covariant LCFs.

As depicted in Fig. 4, for each surface voxel s_v (blue), we shoot two offsets (red vectors) $\psi = (\mathbf{o}_1, \mathbf{o}_2) \in \mathbb{N}^3 \times \mathbb{N}^3$, reaching two neighboring voxels (green). To describe the local geometry, we take the TSDT values within a cuboid around two respective voxels (yellow squares), perform element-wise subtractions and sum them up. Let ε denotes this sum-of-difference operation. By definition, ε from different offsets ψ can be evaluated independently and thus fully parallelizable, which is an useful trait since this computation will be carried out multiple times during training with thousands of randomly generated ψ for the same s_v.

The feature vector $\mathbf{f}$ consist of ε resulted from many offset pairs ψ. More precisely, it is a function of s_v but takes an offset pair ψ, a binary variable η (whether to use *Local Coordinate Frame* (LCF) or not), and a rotational matrix $\mathbf{R} \in SO(3)$ (the orientation of LCF) as parameters. Every possible combination of offset pairs ψ and binary variables η results in one independent feature attribute κ, in notations: $f_\kappa(s_v) = \varepsilon(s_v; \mathbf{R}^\eta(\psi))$. The dimensionality of $\mathbf{f}$ is virtually infinite. Binary variables η determines the alignment of the offset ψ with respect to a LCF, whose transformation is specified by $\mathbf{R}$. The intuition behind this adjustment is to make features $\mathbf{f}$ invariant to poses, c.f. Fig. 4(b) and (c). Without re-orientations, ψ might land on different types of voxel pairs, c.f. Fig. 4(a) and (b), and hence cause different feature responses ε, despite the fact that the current voxels are located on the same position on the body. Both offset pairs ψ and binary variables η are learned during forest training, while the rotational matrix $\mathbf{R}$ is characterized by a LCF obtained as follows.

4.1.3 Local coordinate frame

Defining local coordinate frames for 3D primitives (voxels, vertices, points) has long been studied and usually comes with their 3D descriptor counterparts, see [45] for a comprehensive review. An ideal LCF is supposed to follow whatever transformations the meshes undergo, namely, as *co-variant* as possible, such that the consequent feature representations are as *invariant* as possible. Constructing a LCF boils down to defining three orthonormal vectors as $[x, y, z]$ axes. To do that, 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* [33], [46], [47], [48]. The way they leverage spatial distributions can in general be classified into two categories: (1) EigenValue-Decomposition (EVD) [33], [47], [49], and (2) signed distance (SignDist.) [46], [48]. Since it is impractical to repeat EVD process for all surface voxels s_v, in the following, we propose an adaptation of SignDist. approach to our volumetric representations [50]. This conclusion is drawn after an extensive study and comparison of three LCF approaches presented in our early work [50].

Specifically, for each s_v, we consider its surface normals $\mathbf{n}_v$ as z axis, and obtain y axis by $z \times x$. The task left is to identify a repeatable x axis. To this end, the class of SignDist. approaches look for a discerning point within the support (yellow voxel in Fig. 5(b)). We first open an local cuboid support (pink) around each s_v (green) as visualized in Fig. 5(a). The search involves only the peripheral voxels $\tilde{s}$ (cyan) lying on the intersection of support borders and the surface. The discernibility is defined as the maximum signed distance to the tangent plane [46]:

$$\hat{s} = \arg\max_{\tilde{s} \in \tilde{\mathcal{S}}} \left((\tilde{s} - s_v)^\top \mathbf{n}_v \right), \tag{2}$$

where $\tilde{\mathcal{S}}$ is the intersection of support borders and the surface. The x axis is the projection of the vector directed from s_v towards $\hat{s}$. Fig. 5(b) illustrates the full procedure. Note that there is no guarantee that the discerning point $\hat{s}$ from Eq. 2 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. Therefore, as shown in Fig. 5(c), this approach produces LCFs quasi-covariant to pose changes, and as a result, only quasi-pose-invariant features $\mathbf{f}$. We leave such noise for forests to take care of during learning.

4.2 CVT-based features

The feature $\mathbf{f}(s_v)$ above describes surface geometries in volumes but is devised to match only surface vertices v. A more intriguing question is: can one match these points s directly? In other words, instead of an auxiliary role of matching surfaces, can they also be associated to the template discriminatively and even participate in shape deformations (bottom row of Fig. 3)? We investigate this direction with a *volumetric* representation from centroidal Voronoi tessellations that haven shown some recent success in various applications [51], [52], *i.e.* s is a CVT cell.

We use it to sample a distance field where every cell s stores the Euclidean distance from the centroid to the surface $\partial\Omega$: $d(\mathbf{x}_s, \partial\Omega) = \min_{p \in \partial\Omega} d(\mathbf{x}_s, p)$, yielding a distance-transform like representation similar to the TSDT above.

4.2.1 Haar-like spherical feature

The offset feature $\mathbf{f}(s_v)$ above is nevertheless not applicable here since it relies on regular grids. We propose a new feature $\mathbf{f}(s)$ with the following principles in mind. It should be able to characterize the local neighborhood of any point of the volumetric shape. This rules out the descriptors that rely on surface normals such as MeshHOG [32] and SHOT [33].

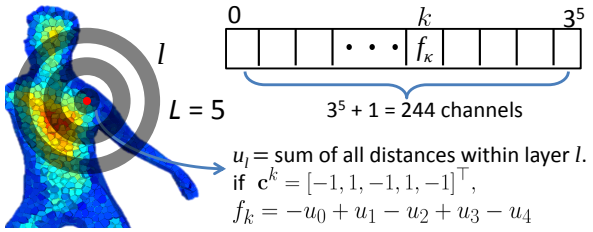


Fig. 6. CVT-based feature. Left: CVT cells $\mathcal{S}$ sample a distance field, where each cell stores the distance $d(\mathbf{x}_s, \partial\Omega)$. Blue to red colors means from close to far. Red dot: cell s to be described. Right: a toy example of our feature $\mathbf{f}$, where $L = 5$. Shadowed and transparent layers have coefficients $c_l = -1$ and 1 respectively. See text for more explanations.

To be able to match any deformed pose with the template, we would like our feature to be pose-invariant. Therefore, we build it on the distance transform because it naturally encodes the relative location with respect to the surface and it is invariant to rotations, translations and quasi-invariant to pose changes. Finally, our feature needs to be robust to the topological noise present in the input data.

Given a distance field sampled by CVT cells $\mathcal{S}$, our feature is similar in spirit to Haar feature in the Viola-Jones face detector [41], except that the rectangular neighborhood is replaced with a sphere. As depicted in Fig. 6, we open an L-layer spherical support region in the Euclidean space around each cell. An L-dimensional vector $\mathbf{u}$ is defined accordingly, where each element u_l is the sum of the distances of all cells falling within layer l. The feature value is the linear combination of all u_l, with coefficients c_l chosen from a set $\Upsilon = \{-1, 0, 1\}$. Formally, suppose $\mathbf{c}$ are L-dimensional vectors whose elements are the bootstrap samples of Υ. Let $\mathbf{c}^\kappa$ denote one particular instance of $\mathbf{c}$, *i.e.* , $\mathbf{c}^\kappa \in \Upsilon^L$. The feature value is then expressed as an inner product: $\mathbf{u}^\top \mathbf{c}^\kappa$, corresponding to one feature attribute κ. We consider all possible $\mathbf{c}^\kappa$ and also take the distance d itself into account. $\mathbf{f}$ is hence a vector of $(3^L + 1)$ dimensions, where 3^L is the cardinality of Υ^L and each element f_κ is defined as:

$$f_\kappa \triangleq \begin{cases} \mathbf{u}^\top \mathbf{c}^\kappa = \sum_l c_l^\kappa u_l, & \kappa < 3^L,\ c_l^\kappa \in \{-1, 0, 1\} \\ d(\mathbf{x}_s, \partial\Omega), & \kappa = 3^L \end{cases}. \quad (3)$$

Since each dimension f_κ is computation-wise independent, $\mathbf{f}$ is suitable for decision forests, which select feature channels κ randomly to split the data during training. Being derived from $d(\mathbf{x}_s, \partial\Omega)$, $\mathbf{f}$ inherits the invariance to rigid-body motions. As opposed to the early version of this work [53], we normalize the distances with respect to the averaged edge length of cells, achieving invariance to the body size to a certain extent. However, $\mathbf{f}$ is not invariant to pose changes as the contained cells in each layer vary with poses. Although considering geodesic spherical supports instead of Euclidean ones would overcome this issue and yield quasi-invariance to pose changes, the resulting feature would be highly sensitive to topological noise. Thus, we keep the Euclidean supports and let forests take care of the variations caused by pose changes in learning.

5 Correspondences Inference

Now that the features for both surface and volumetric associations, $\mathbf{f}(v)$ and $\mathbf{f}(s)$, are defined, we proceed on using them to train a regression forest, an ensemble of T binary decision trees, to learn the mapping $\mathbf{r} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ from the observation domain to the template domain. During training each tree learns the split functions that best separate data recursively at branch nodes, while during testing the input point is routed through each tree, reaching T leaves that store statistics as predictions. We discuss in § 5.1 a generic learning framework that applies to both shape parameterizations. A CVT-specific multi-template strategy is presented in § 5.2 to generalize the Vitruvian framework from single mesh connectivity to multiple ones.

5.1 Training and prediction

Broadly speaking, training a regression forest amounts to determining the following components: sample-label pairs, split functions, learning objectives and leaf-node statistical models. Readers are referred to [39] for a comprehensive analysis on different choices of these components.

5.1.1 Training data and split functions

First we elaborate the training scenario for surface representations. Since forests aim to map an observed 3D vertex back to the template domain $\partial\Omega_X$, usually chosen to be in the rest (T or A) pose, it requires meshes in various poses but with the same connectivity for training. To incorporate abundant training variations, we animate the template $\mathbf{X}^0 = \{\mathbf{x}_v^0\} \subset \partial\Omega_X$ to a variety of poses with a method similar to [54]. After voxelizing all animated meshes, we associate each surface voxel to their locations at the rest pose, obtaining a pool of sample-label pairs $\mathcal{D} = \{(s_v, \mathbf{x}_v^0)\}$. Each tree is trained with a randomly bootstrapped subset of $\mathcal{D}$. While the split function may be arbitrarily complex, a typical choice is a stump where one single dimension κ is compared to a threshold τ, *i.e. axis-aligned thresholding*. Our splitting candidate ϕ is hence the pair of testing channels κ and thresholds τ, $\phi = (\kappa, \tau)$, where κ is represented by offset pairs ψ and binary variables η in § 4.1. Let $\mathcal{D}_N$ denotes the samples arriving at a certain branch node. The training process is to partition $\mathcal{D}_N$ recursively into two subsets $\mathcal{D}_L$ and $\mathcal{D}_R$, based on randomly generated ϕ:

$$\mathcal{D}_L(\phi) = \{s_v \in \mathcal{D}_N | f_\kappa(s_v) = \varepsilon(s_v; \mathbf{R}^\eta(\psi)) \geq \tau\}, \quad (4a)$$
$$\mathcal{D}_R(\phi) = \{s_v \in \mathcal{D}_N | f_\kappa(s_v) = \varepsilon(s_v; \mathbf{R}^\eta(\psi)) < \tau\}. \quad (4b)$$

Similarly, given a set of CVTs corresponding to the template volumes Ω_X deformed in various poses, we associate each cell $s \in \mathcal{S}_X$ to its locations in the rest pose, denoted as $\mathbf{x}_s^0 \in \mathbf{X}^0 \subset \Omega_X$, forming a pool of sample-label pairs $\mathcal{D} = \{(s, \mathbf{x}_s^0)\}$ as the dataset. The split candidate ϕ is again the pair of thresholds and feature attributes, $\phi = (\kappa, \tau)$, where features are instead computed according to Eq. 3 but the thresholding criteria in Eqs. 4a and 4b follows.

5.1.2 Learning objectives and leaf predictions

At branch nodes, many candidates ϕ are randomly generated and the one that maximizes the information gain I, $\phi^* = \text{argmax}_\phi I(\phi)$, is stored for the later prediction use. We follow the classic definition of information gain:

$$I(\phi) = H(\mathcal{D}_N) - \sum_{i \in \{L,R\}} \frac{|\mathcal{D}_i(\phi)|}{|\mathcal{D}_N|} H(\mathcal{D}_i(\phi)), \quad (5)$$

where H is the entropy, measured as the variance in Euclidean space, *i.e.* $H = \sigma^2$ for both parameterizations. The tree recursively splits samples and grows until one of the following stopping criteria is met: (1) it reaches the maximum depth, or (2) the number of samples $|\mathcal{D}_N|$ is too small. A Mean-Shift clustering [55] is performed in a leaf node to represent the distributions of $\mathbf{x}^0$ as a set of confidence-weighted modes $\mathcal{H} = \{(\mathbf{h}, \omega)\}$. $\mathbf{h} \in \mathbb{R}^3$ is the mode location and ω is a scalar weight.

In the prediction phase, a 3D input point $i \in \mathcal{V}_Y$ or $i \in \mathcal{S}_Y$ traverses down the trees and lands on T leaves containing different collections of modes: $\{\mathcal{H}_1 \cdots \mathcal{H}_T\}$. The final regression output $\mathbf{r}_i$ is the cluster centroid with largest weight obtained by performing Mean-Shift [55] on them. Each observed point then gets a closest point p in the reference shape $\mathbf{X}^0$, either in surfaces, $p = \text{argmin}_{v \in \mathcal{V}_X} \left\|\mathbf{r}_i - \mathbf{x}_v^0\right\|_2$, or in CVTs, $p = \text{argmin}_{s \in \mathcal{S}_X} \left\|\mathbf{r}_i - \mathbf{x}_s^0\right\|_2$. The correspondence pair (i, p) serves as input to the subsequent deformation framework described in § 6.

Outliers such as false geometries, or un-removed background elements often exist in 3D data, drastically deteriorating tracking results. If their models are available, we also include them in the training process, so that forests can identify and reject them online. In this case, the goodness of a split ϕ is evaluated in terms of both classification and regression. We follow Fanelli *et al.* [56] and extend the entropy to be:

$$H(\mathcal{D}) = -\sum_c p(c|\mathcal{D}) \log p(c|\mathcal{D}) + (1 - e^{\frac{\delta}{\alpha}})\sigma^2(\mathcal{D}), \quad (6)$$

where $p(c|\mathcal{D})$ is the class probability of being foreground or background. It is the weighted sum of the aforementioned regression measure σ^2 and the classification entropy measure. Forests trained with Eq. 6 are often referred to as *Hough forests*. During training it learns simultaneously (1) how to distinguish between valid and invalid samples (outliers) and (2) how to match valid samples to the template. The regression part gets increasing emphasis when the current depth δ gets larger (*i.e.* the tree grows deeper), and the steepness is controlled by the parameter α.

5.2 Learning across multiple volumetric templates

So far we know how to utilize Vitruvian-based learning framework to match surface or volumetric data against the template. For the training purposes, one has to deform the reference mesh into various poses such that all meshes share a consistent topology $\mathcal{T}_X$ and one can easily assign each sample a continuous label which is its rest-pose position $\mathbf{X}^0$. In this regards, the trained forest applies only to one mesh connectivity $\mathcal{T}_X$. Nevertheless, the amount of training data for one single template is often limited. To avoid over-fitting, the rule of thumb is to incorporate as much variation as possible into training. This motivates us to devise an alternative that learns across different template connectivities $\mathcal{T}_X$. Due to the high memory footprint of regular voxel grids, this strategy is unfortunately less practical for the surface feature $\mathbf{f}(v)$ in § 4.1 and we implement it only with CVTs.

Given U distinct CVT templates: $\{\mathcal{S}^\mu\}_{\mu=1}^U$[3], whose temporal evolutions are recovered with the method in [51], resulting in a collection of different templates deformed in various poses: $\{\{\mathbf{X}_1^t\} \cdots \{\mathbf{X}_U^t\}\}$ as our dataset. To include all of them into training, we take one generic template $\hat{\mathcal{S}}$ as the reference. Intuitively, if there exists a mapping g that brings each cell $s \in \mathcal{S}^\mu$ to a new cell $g(s) = \hat{s} \in \hat{\mathcal{S}}$, one only needs to change the template-specific labels $\mathbf{x}_s^0$ to the corresponding $\mathbf{x}_{\hat{s}}^0$, which are common to all templates, and the training process above can again be applied. In other words, we align topologies by matching every template $\mathcal{S}^\mu$ to $\hat{\mathcal{S}}$. Fig. 7 depicts this multi-template learning scheme.

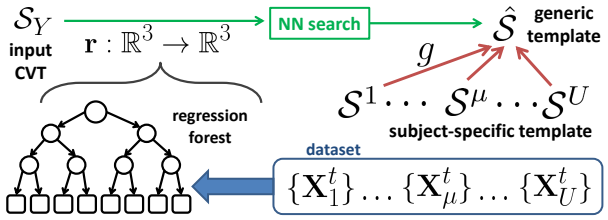


Fig. 7. The schematic flowchart of the multi-template learning framework. Red arrows: mappings g^μ that associate the indices from each subject-specific template $\mathcal{S}^\mu$ to the common one $\hat{\mathcal{S}}$. $\mathbf{X}_\mu^t$ are the temporal evolutions of each template. Blue: training; green: prediction.

Although various approaches for matching surface vertices exist, only a handful of works discuss matching voxels/cells. Taking *skinning weights* [57] as an example, we demonstrate in the following how to adapt a surface descriptor to CVTs. Note that our goal is not to propose a robust local 3D descriptor. With proper modifications, other descriptors can be used as well for shape matching.

5.2.1 Generalized skinning weights.

Skinning weights are originally used for skeleton-based animations, aiming to blend the transformations of body parts (bones). Usually coming as a side product of the skeleton-rigging process [58], it is a vector $\mathbf{w}$ of B-dimensions, each corresponding to a human bone b and B is the number of bones. The non-negative weight w_b indicates the dependency on that part and is normalized to sum up to one, *i.e.* $\sum_b w_b = 1$. As such, a skinning weight vector $\mathbf{w}$ is actually a probability mass function of body parts, offering rich information about vertex locations. To extend it from surface vertices to CVT cells, we first relax the unity-summation constraint as $\mathbf{w}$ is not used to average transformations of bones but only as a descriptor here. The intuition behind the adaptation is that, a CVT cell should have bone dependencies similar to the closest surface point. Therefore, for a cell whose normalized distance to the surface is d, its skinning weight is simply the one of its closest surface point, scaled by e^d. We tackle scale changes by normalizing the distance field with the averaged edge length of cells in the shape. Since the shortest distance usually hits a triangle rather than a single vertex, we use barycentric coordinates as the coefficients to linearly combine the skinning weights of the three vertices. Note that this does not violate the unity-summation constraint for surface vertices as their distance d is still zero. We illustrate this concept in Fig. 8(a). The mapping g is then determined by searching for the nearest neighbor in the skinning weight space: $g(s) = \arg\min_{\hat{s} \in \hat{\mathcal{S}}} \|\mathbf{w}_{\hat{s}} - \mathbf{w}_s\|_2$.

In practice, we use `Pinocchio` [58] to computes skinning weights, extend them from surface vertices to CVT

3. The template suffix X is dropped to keep notations uncluttered.

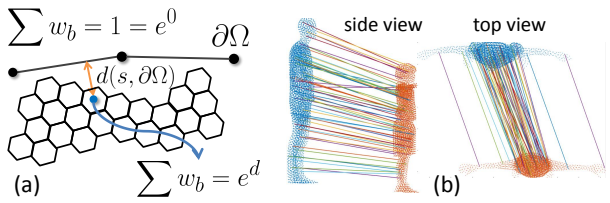


Fig. 8. (a): illustration of our strategy adapting skinning weights to CVT cells. Distances to the surface $d(\mathbf{x}_s, \partial\Omega)$ are reflected in the normalization constants e^d. (b): result of matching two templates.

cells, and match all cells to those of the common template $\hat{\mathcal{S}}$. The resulting skeletons are not used in our method. Fig. 8(b) visualizes one example of matching results. Our approach yields reasonable matches, regardless of the difference in body sizes. Due to the descriptiveness of skinning weights, symmetric limbs are not confused. Note that this computation is performed only between user-specific templates $\mathcal{S}^\mu$ and the generic one $\hat{\mathcal{S}}$ off-line once. Input data $\mathcal{S}_Y$ cannot be matched this way, because rigging a skeleton for shapes in arbitrary poses remains a challenging task.

6 Tracking

Recall that our goal is not only to detect the associations $\mathcal{C}$ but eventually to estimate the deformation parameter $\hat{\mathbf{\Theta}}$ via $\hat{\mathbf{\Theta}} = \text{argmin}_{\mathbf{\Theta}} E(\mathbf{\Theta}; \mathcal{C})$, such that the resulting $\mathbf{X}(\hat{\mathbf{\Theta}})$ best explains $\mathbf{Y}$. The choice of $\mathbf{\Theta}$ could be raw point positions [59], [60], skeletal kinematic chains [4], [61] and cage [62]. We opt for a patch-based deformation framework [2] for surfaces and a CVT cluster-based method [51] for volumetric meshes respectively. Both group the 3D points into a higher-level structure, where shape deformations are represented as the ensemble of their rigid-body motions $\boldsymbol{\theta}$. We briefly explain here the basic principles and how to apply the predicted correspondences in § 5 to track a sequence of temporally inconsistent observations.

6.1 Surface-based deformation

In [2], the reference surface is decomposed into several patches k. It serves as a intermediate deformation structure between vertex positions and anatomical skeletons. Without any prior knowledge of motion, patches are preferred to be distributed uniformly over $\mathcal{X}$. Given correspondences $\mathcal{C}$ from above, a data term is formulated as:

$$E_{data}(\mathbf{\Theta}; \mathcal{C}) = \sum_{(i,p)\in\mathcal{C}} w_{ip} \|\mathbf{y}_i - \mathbf{x}_p(\mathbf{\Theta})\|_2^2, \quad (7)$$

which is a standard sum of weighted squared distances.

Since evolving a surface with discrete observations (even with a good $\mathcal{C}$) is ambiguous by nature, regularization terms are usually introduced to exert soft constraints. Given a vertex v, the rigidity constraint enforces the predicted positions $\mathbf{x}_v(\boldsymbol{\theta}_k)$ and $\mathbf{x}_v(\boldsymbol{\theta}_l)$ from two adjacent patches P_k and $P_l \in \mathcal{N}_k$ to be consistent:

$$E_r(\mathbf{\Theta}) = \sum_{k=1}^{K} \sum_{P_l\in\mathcal{N}_k} \sum_{v\in P_k\cup P_l} w_{kl} \left\|\mathbf{x}_v(\boldsymbol{\theta}_k) - \mathbf{x}_v(\boldsymbol{\theta}_l)\right\|_2^2, \quad (8)$$

where $\mathbf{\Theta}$ is implicitly encoded in $\mathbf{x}_v(\boldsymbol{\theta}_k)$ and $\mathbf{x}_v(\boldsymbol{\theta}_l)$.

Given a fixed input $\mathbf{Y}$, the regression forest returns a fixed response $\tilde{\mathbf{Y}}$, and in turn a fixed $\mathcal{C}$. We therefore apply standard Gauss-Newton method directly to find the minimizer of final energy: $E(\mathbf{\Theta}; \mathcal{C}) = \lambda E_{data}(\mathbf{\Theta}; \mathcal{C}) + E_r(\mathbf{\Theta})$, where λ defines the softness of the template and is empirically set to 10 throughout our experiments. Note anyway that refining $\mathcal{C}$ like non-rigid ICP does is always possible. In this case, our method provides better initializations than using last frame results, reducing the needed ICP-iterations.

6.2 Volumetric deformation

On the other hand, a similar deformation framework can be formulated for CVTs as well, only that surface patches k are replaced by clusters of cells. We follow [51] which is essentially a non-rigid ICP method. As opposed to the extensive correspondence search, we again directly use the association pairs (i, p) detected by the forest as initializations. This results in a faster pose estimation.

7 Experiments

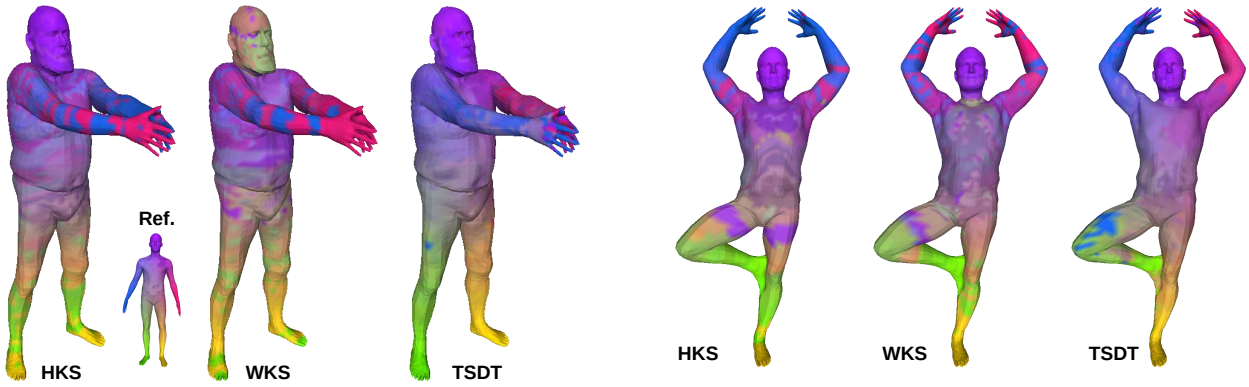
The presented method is evaluated extensively in this section. We verify the merits of the discriminative associations as well as the complete 3D tracking-by-detection pipeline, in both surface and CVT parameterizations. As summarized in Table 1 in the supplemental material, in total 15 datasets are considered for various evaluation purposes. Due to the availability of ground-truths, the input in § 7.1 is the non-rigid registration, whereas in § 7.2 it is the reconstructed visual hull from [38] or raw tessellated CVT from [63].

7.1 Discriminative associations

Recall that the goal of discriminative correspondences is to guide the shape deformation not to match non-rigid 3D shapes accurately. We aim only to show that (1) the presented features are more or at least equally informative for matching humanoid surfaces than the existing state-of-the-arts 3D descriptors, *e.g.* Heat Kernel Signature (HKS) [30], [64] or Wave Kernel Signature (WKS) [31] and (2) CVT-based associations are more reliable than the surface-based counterparts. We describe every vertices with HKS, WKS, and our pair-wise offset features $\mathbf{f}(v)$ in § 4.1. CVT cells are, on the other hand, described by the Haar-like spherical features $\mathbf{f}(s)$ in § 4.2. The forests learn to match these 3D primitives against their own learning template, either a generic reference surface (*FAUST*) or a subject-specific CVT template (*Goalkeeper*, *Ballet* and *Thomas*).

7.1.1 Surface-based correspondences

Surface correspondences are validated on the publicly-available dataset *FAUST* [65]. Following [16], [34], we use only the training set because of the availability of ground-truth vertex indices. It comprises 100 static 3D scans from 10 subjects in 10 poses. The accuracy on *FAUST* indicates how well the proposed method deals with human shape variations. Specifically, half the registrations (50 meshes) are chosen to train $T = 50$ trees and the other half are left out for testing. At branch nodes, 5000 splitting candidates ϕ are randomly generated and the best one is stored. The error measure is the geodesic distance between predicted vertices



(a) unseen poses (b) unseen shapes

Fig. 9. Qualitative results of surface matching on *FAUST*. Best viewed in pdf.

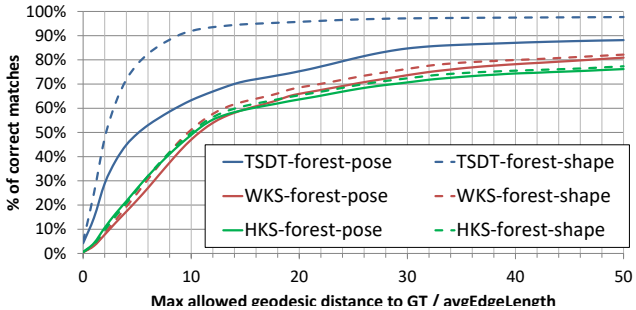


Fig. 10. Cumulative errors on *FAUST* [65].

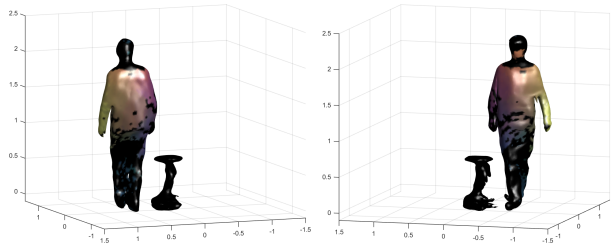


Fig. 11. Predicted data-model associations on noisy visual hulls with Hough forests. Black color means that the points are either outliers, or the inferred correspondences are rejected due to incompatible normals.

and ground-truths. If the distance is smaller than a certain threshold, we consider the point correctly matched. The percentage of correct matches in varying thresholds characterizes the performance of one algorithm and is commonly used in many matching papers [16], [34].

The results are shown in Fig. 10, where x-axis is normalized by the averaged edge length of the template. We partition the 100 meshes in two ways to test the generalization to unseen shapes or poses. The keyword *pose* means that the forest is trained with meshes in all 10 subjects but in only 5 poses, whereas *shape* represents the opposite. To compare fairly with other existing methods, we keep the Vitruvian-manifold label space unchanged (*i.e.* the same learning template) while replacing the voxel-based features with 30-dimensional scale-invariant HKS or WKS feature vectors. The proposed TSDT-forest combination yields overall best accuracy in Fig. 10, suggesting that the voxel-based TSDT feature is indeed more informative than H/WKS in the chosen parameter range. Comparing the blue solid curve to the dashed one, we notice that our approach handles unseen shapes better than unseen poses. This is due to the fact that our feature relies mainly on 3D geometry, in which pose variations cause more significant changes than shape variations. Although this phenomenon is not observed in the curves of H/WKS because they exploit the spectral domain for better pose invariance, they suffer from the confusion between symmetric parts as visualized in Fig. 9.

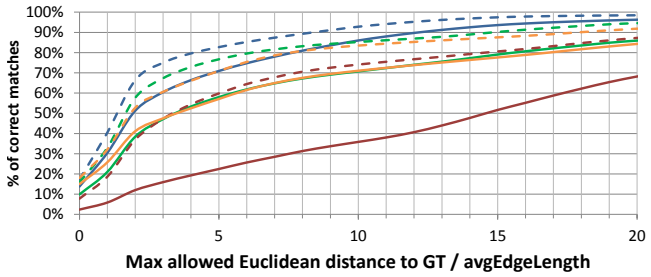
We further visualize in Fig. 11 the predicted associations on noisy reconstructed visual hulls with outliers, where no ground truths are available. Black colors means that the predicted correspondences are either rejected by simple normal compatibility check [2] like those on the body, or rejected because they are recognized as the chair. In this experiment, we include chair meshes into training data and follow Eq. 6 as the entropy measure to grow the trees. As a result, we can identify observations on the chair online and remove them, so that they do not affect the subsequent tracking stage. The task of trees here is throwing away the points of known outlier classes and in the meanwhile also predicting correspondences for the remaining points.

As one can see, our approach is capable of predicting reasonable associations for noisy visual hulls while rejecting outliers. This is of importance since they are the real input data of the final tracking-by-detection pipeline. HKS and WKS are known for their sensitivity to topological noises, *e.g.* the merging of arms and torso. We however would like to remark that, as oppose to our feature vector $\mathbf{f}(v)$ that has a dimensionality virtually longer than 5000 from the randomly generated splitting candidates at each branch node, HKS and WKS are only 30-dimensional in our experiment. To fully conclude that the presented voxel-based feature is certainly better than HKS or WKS requires a more fair and thorough comparison but is not the main goal of this paper.

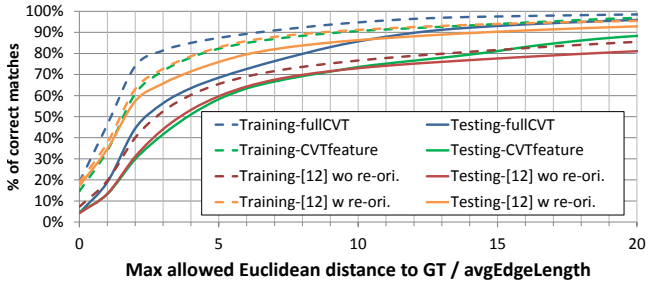
7.1.2 *Volumetric correspondences*

The discriminative CVT-based correspondence in § 4.2 is validated with 6 sequences from 3 subjects: *Goalkeeper*, *Ballet* and *Thomas*. We register each template to the corresponding raw CVT sequences using a EM-ICP based

Fig. 12. Qualitative results of volumetric matching on the raw CVTs. Best viewed in pdf.



(a) *Thomas*



(b) *Ballet*

Fig. 13. Cumulative matching accuracy of different approaches. The x-axis is normalized w.r.t. the average edge length of the templates. The number of trees T is 20 in this experiment. Dashed and solid lines are accuracies on training (Tr) and testing (Te) sequences respectively.

method [51] to recover temporal coherent volumetric deformations (tracked CVTs). For each subject, up to 250 tracked CVTs are randomly chosen from the first sequence to form the training set, while the second sequences are completely left out for testing. We open $L = 8$ sphere layers for the feature computation. Each tree is grown up to depth 20 with 30% bootstrap samples randomly chosen from the dataset.

The contributions of CVT on improving the correspondences detection are evaluated in two aspects. First, we keep using the Vitruvian manifold $\partial\Omega$ as the regressing domain but replace the voxel-based features $\mathbf{f}(v)$ with the spherical feature $\mathbf{f}(s)$, denoted as *CVTfeature*. Next, we further change the label space from surfaces $\partial\Omega$ to volumes Ω, termed *fullCVT*. We test on the tracked CVTs and report the results on all frames of both training sequences (Tr) and testing ones (Te). The drop between them indicates the ability to generalize. The same error measure as in the previous subsection is applied, only the geodesic distances are replaced by Euclidean ones. To yield a fair comparison with [11], here the forests are subject-specific and consist of $T = 20$ trees.

Fig. 13 shows the percentage of correct matches in varying thresholds for *Thomas* and *Ballet*. Since *CVTfeature*

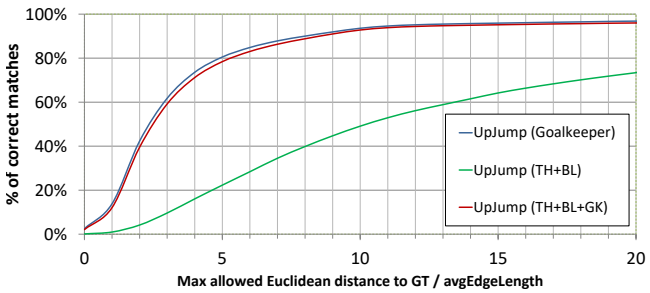
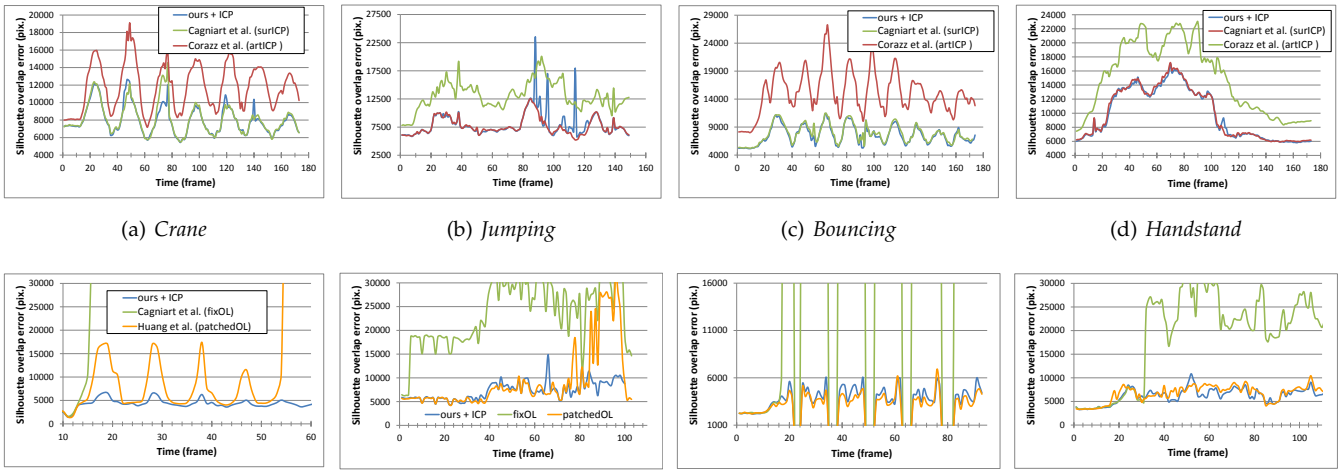


Fig. 14. Cumulative matching accuracy on *Goalkeeper*.

and [11] are regressing to surfaces whereas *fullCVT* regresses to volumes, we normalize the x-axis by the average edge length of templates to yield fair comparisons. While the results of *CVTfeature* are comparable to [11] (green vs. red or orange), *fullCVT* attains the improved accuracies (blue vs. red or green), demonstrating the advantages of our fully volumetric framework. Some visual results of the *fullCVT* approach on raw CVT input are shown in Fig. 12.

It is worth a closer analysis to highlight the advantages of CVT-based feature $\mathbf{f}(s)$ against the voxel-based one $\mathbf{f}(v)$. Our early work [11] applied $\mathbf{f}(v)$ that takes 150^3 voxels for $\mathbf{f}(v)$ to describe a human shape, while CVT needs only 5k cells[4]. Consequently, [11] is not able to include a sufficient amount of training shapes, leading to a major drawback that forests are limited to one single subject. To further decrease the needed number of training meshes, [11] exploits skeletal poses to cancel the global orientation. This in turn makes every mesh in the training dataset face the same direction and we learn merely pose variations. It follows that during tracking the input data has to be re-oriented likewise using the estimated skeletal poses from the last frame. The CVT-based feature $\mathbf{f}(s)$, on the other hand, considers distance fields of cells which is naturally invariant to rotations and hence does not require re-orientations. We anyway compare to [11] in both settings. Orange curves in Fig. 13 shows the results with re-orientation, which is better than the proposed strategy in *Ballet*. Nonetheless, without re-orienting data, the accuracy drops substantially during testing (compare red to orange). The efficiency on memory and the invariance of our features are two determining reasons why the presented method is better than [11]. With the multi-template learning strategy in § 5.2, it takes just one forest for different subjects in the tracking-by-detection experiment in § 7.2.

4. Further note that [11] stores a 3D vector in each voxel, whereas we store a scalar in each CVT cell. So the ratio is 3×150^3 to 5k.



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

(e) *Cutting* (f) *WalkChair1* (g) *HammerTable* (h) *WalkChair2*

Fig. 15. Pixel overlap error of 8 sequences, averaged over all cameras. Image resolution: (a-d): 1920×1080; (e-h): 1000×1000. Best viewed in pdf.

Next, we use the sequences of *Goalkeeper* to verify the merits of this multi-template learning framework, which is unfeasible for voxel-based feature $\mathbf{f}(v)$ due to the high memory footprint. It is a particularly difficult dataset because motions in the testing sequence *UpJump* have little overlap with those in the training *SideJump*. We report in Fig. 14 the correctness of correspondences for *UpJump* (unseen during training) in *fullCVT* setting. Three situations are taken into account: training with the tracked CVTs of all three subjects (red), training only with those from *Goalkeeper* (blue) and without *Goalkeeper* (green). For red and green curves, we choose the *Goalkeeper* template as the common one $\hat{\mathcal{S}}$ and follow the strategy in § 5.2 to align distinct CVT tessellations. Comparing the red curve to the blue one confirms the advantage of this cross-template approach, leading to a forest that applies to all three subjects without trading off much accuracy. Nonetheless, the training data of the testing subject is still indispensable, as the accuracy drops when there is no tracked CVTs of *Goalkeeper* (green vs. red or blue), even if the forest of green curve is trained with twice the amount of CVTs compared to the blue curve. This is partially due to the fact that template of *Goalkeeper* has much smaller size than the other two and suggests that the proposed Haar-like feature in Eq. 3 captures more shape than pose information.

7.2 Tracking-by-detection

Now we move on to evaluate the full tracking-by-detection pipeline. The predicted associations $\mathcal{C}$ of two parameterizations are fed into their respective shape deformation frameworks in § 6 and the tracking is carried out on a frame-by-frame basis. The fidelity of estimated shapes is verified by the widely-used silhouette overlap error.

7.2.1 Surface-based

An individual forest is trained for each subject with up to 200 meshes, depending on the number of vertices in the template. For *Baran* and *Vlasic*, we train standard regression forest; for *Lionel* and *Ben* we apply the adaptation in Eq. 6 ($\alpha = 5$) due to the un-properly segmented chairs and tables in input data. Growing $T = 20$ trees to depth 25 with 5000 testing offset pairs ψ takes about 3 hours. Although it is not the aim of this paper, we anyway augment the energies in § 6.1 with the skeleton energy in [2] and validate the estimated human poses in 2D.

TABLE 1
Average silhouette overlap error in pixels 4 sequences at low frame rate. Image resolution: 1920×1080.

	Crane	*Jumping*	*Bouncing*	*Handstand*
ours	7746.40	9148.94	6847.72	9279.57
surICP [2]	8295.58	16759.29	9400.76	11690.61

For sequences without outliers, we compare with surface-based ICP (surICP) [2] and articulated ICP (artICP) [15], both of which explain data with GMM using the Expectation-Maximization algorithm. We run an additional ICP step to reduce the errors (ours + ICP) for all testing sequences. The averaged overlap errors are shown in Fig. 15(a-d). In general, our method performs much better than artICP and attains comparable results with surICP. However, ICP-based methods often fail when large deformation occurs between consecutive frames, which is usually the case in videos of low frame rates. We simulate this by tracking only every three frames. As reported in Table 1, surICP now yields higher errors because local proximity search fails to estimate correspondences properly, while our approach is able to handle large jumps between successive input.

Four of our testing sequences, *Cutting*, *WalkChair1*, *HammerTable*, and *WalkChair2* contain tables or chairs in observations, which play the roles as static outliers. We compare with other outlier rejection strategies such as, fixed outlier proportion (fixOL) [2], removing outliers by body-part classifications with SVM (bpSVM) [66], and modeling outlier likelihood dynamically by aggregating over all patches (patchedOL) [3]. As shown in Fig. 15(e-h), conventional outlier strategy fixOL drifts quickly and soon fail to track (green curves). ICP with robust outlier treatment, patchedOL, is able to sustain noisy input to a certain extent. Once it starts drifting, the error only gets higher due to its ICP nature (yellow curves). When subjects and outliers are sperate

components in visual hulls, we cast them into separately TSDT, and feed them into the joint classification-regression forest. If they are connected to each other, forests inevitably associate some outliers to the humanoid template, leading to undesirable deformations as suggested by the spike in blue curves in Fig. 15(f). Nonetheless, since we rely less on previous frames for data associations, the results can always get recovered when they are separate again. In average, we still yield low errors throughout the whole sequences. We remark that such ability to recover is the essence of our discriminative approach, which is the biggest advantages over the existing generative methods. The recovered shapes and poses, superimposed on original images, are also presented in Fig. 2(c) in the supplementary material.

7.2.2 Fully volumetric tracking-by-detection

After evaluating the surface-based tracking-by-detection framework, now we turn to evaluate the volumetric one. We compare in two quantitative metrics against the whole pipeline in [11], which is the early version of our surface-based tracking-by-detection approach.

Unlike the matching experiment in the previous subsection, here we apply the multi-template strategy in § 5.2 to train one universal regression forest, with *Goalkeeper* chosen as the common template $\hat{\mathcal{S}}$. Training $T = 50$ trees up to depth 20 where each one is grown with around 200 CVTs (approximately one million samples) takes about 15 hours on a 24-core Intel Xeon CPU machine. For each subject, we track the testing sequence, which is not part of the training set. Tracking inputs are raw CVTs that have no temporal coherence. The number of clusters K is 250 for *Ballet* and *Goalkeeper* and 150 for *Thomas*. We evaluate our tracking approach with two different metrics. On one hand, evaluation with marker-based motion capture evaluates the correctness of the surface pose, but only for a sparse set of surface points. On the other hand, the silhouette overlap error evaluates the shape estimate but not the estimated pose. Hence these metrics are complementary.

Some visual results are shown in Fig. 3 in the supplemental material and video[5]. Our approach is able to discover volumetric associations even in challenging poses found in *Thomas* and deform the templates successfully. As shown in Table 2-4, we evaluate the results by computing the overlap error between the ground truth silhouette and the projection of the estimated surface. The metric we use is the pixel error (number of pixels that differ). Statistics are computed on all frames of all cameras. The *Ballet/Seq2* sequence has marker-based motion capture data: fifty markers were attached to the body of the subject, providing a sparse ground truth for surface tracking. First, each marker is associated to a vertex of the template surface. Then, for each marker, we measure the distance between its location and the estimated vertex location. Statistics on the distance are reported on Table 5. We observe that our approach attains slightly better performances than a state of the art ICP-based approach [51] and outperforms a surface-based tracking-by-detection [11] which mostly fails to correctly register the legs of the subject.

5. https://hal.inria.fr/hal-01300191

TABLE 2
Silhouette pixel error on sequence *Goalkeeper/UpJump*. Image size is 2048×2048.

method	mean	stddev.	median	max
Proposed	15221	6843	14754	57748
Huang *et al.* [11]	19838	14260	15607	109428
Allain *et al.* [51]	14773	6378	14355	43359

TABLE 3
Silhouette pixel error on sequence *Ballet/Seq2*. Image size is 1920×1080.

method	mean	stddev.	median	max
Proposed	2620	1041	2557	8967
Huang *et al.* [11]	5427	2809	4863	39559
Allain *et al.* [51]	2606	1008	2571	7642

TABLE 4
Silhouette pixel error on sequence *Thomas/Seq2*. Image size is 2048×2048.

method	mean	stddev.	median	max
Proposed	9991	7089	7968	78242
Huang *et al.* [11]	28731	23421	22991	354293
Allain *et al.* [51]	10199	7379	8022	81649

TABLE 5
Statistics of surface registration error at marker locations, on the *Ballet/Seq2* sequence.

method	mean (mm)	stddev. (mm)
Proposed	26.37	16.67
Huang *et al.* [11]	124.02	200.16
Allain *et al.* [51]	27.82	18.39

7.2.3 Discussion

Last but not lease, we make a short comparison between the two presented features. As discussed above, voxel-based volume in § 4.1 has the downside of high memory footprint, which limits the allowed training variations. Aligning the orientations is one way to reduce the training variation such that forests only need to learn the pose variations of one single subject. One has to repeat the same thing likewise during the testing phase. In [11], we rely on the skeletal poses of previous frames for this purpose and thus the forest predictions are not fully frame-independent, exposing tracking subject to the potential risk of drifting. To facilitate a fully 3D tracking-by-detection framework, the information of previous frames is preferred no to participate in the discriminative correspondence estimation. On the other hand, the spherical feature presented in § 4.2 attempts to incorporate rotational, pose, and even shape variations during training, yielding completely frame-wise forest predictions.

As reported in Fig. 13, without aligning rotations, the accuracies of correspondences drop substantially on the testing sequences for the method in [11]. This means that voxel-based framework and the corresponding features do not generalize well to unseen rotations. When deployed in tracking applications, such unreliable associations eventually result in failure. In particular, one can observe in Table 4 that [11] attains high silhouette overlap discrepancy, most likely due to the fact that the subject rotates himself in many orientations and thus confuses the forest. From these observations, we conclude that the CVT-based Haar-like feature and the derived fully volumetric tracking-by-detection framework is better than the voxel-based counterpart.

8 Conclusion

In this paper, we present two features for surface and CVT shape parameterizations respectively, both making use of volumetric distance fields to describe 3D geometry. Aiming to integrate with random forests, each feature attribute is computationally independent and can be obtained on the fly in testing. They facilitate the surface-based and CVT-based discriminative associations and in turn lead to the corresponding tracking-by-detection frameworks for 3D human shapes. While CVT-based approach is more robust to the surface counterpart, we show that both yield more stability compared with the respective generative ICP extensions. The reliability of the proposed method is confirmed by the experiments on numerous public sequences. Future directions include alleviating problems of topological changing and incorporating photometric information.

Acknowledgments

Several datasets proposed in this paper have been acquired using the Kinovis platform at Inria Grenoble Rhône-Alpes (http://kinovis.inrialpes.fr).

References

[1] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *TOG*, vol. 27, no. 3, p. 97, 2008.
[2] C.-H. Huang, C. Cagniart, E. Boyer, and S. Ilic, "A bayesian approach to multi-view 4d modeling," *IJCV*, vol. 116, no. 2, pp. 115–135, 2016.
[3] C.-H. Huang, E. Boyer, N. Navab, and S. Ilic, "Human shape and pose tracking using keyframes," in *CVPR*. IEEE, 2014, pp. 3446–3453.
[4] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of multiple characters using multiview image segmentation," *TPAMI*, vol. 35, no. 11, pp. 2720–2735, 2013.
[5] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon, "Metric regression forests for human pose estimation," in *BMVC*, 2013.
[6] 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, pp. 103–110.
[7] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *PAMI*, 1992.
[8] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
[9] E. Rodola, S. R. Bulo, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 4177–4184.
[10] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*. IEEE, 2011, pp. 1297–1304.
[11] 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, 2015, pp. 4027–4035.
[12] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," in *ACM SIGGRAPH 2008*, 2008.
[13] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Springer Science & Business Media, 2006, vol. 153.
[14] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM review*, vol. 41, pp. 637–676, 1999.
[15] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," *IJCV*, 2010.
[16] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *CVPR*, 2016.
[17] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *TOG*, vol. 34, no. 4, p. 69, 2015.
[18] 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," *TOG*, vol. 35, no. 4, p. 114, 2016.
[19] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2, pp. 90–126, 2006.
[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] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *ICRA*. IEEE, 1991.
[22] M. Klaudiny, C. Budd, and A. Hilton, "Towards optimal non-rigid surface tracking," in *ECCV*. Springer, 2012, pp. 743–756.
[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," in *Journal of the royal statistical society*, 1977.
[24] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE TPAMI*, vol. 33, no. 8, pp. 1633–1645, 2011.
[25] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," in *TOG*, vol. 28, no. 5. ACM, 2009, p. 175.
[26] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *TOG*, vol. 31, no. 6, p. 188, 2012.
[27] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *CVPR*, 2015, pp. 3213–3221.
[28] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky *et al.*, "Shrec 2011: robust feature detection and description benchmark," in *Eurographics 3DOR Workshop*. Eurographics Association, 2011, pp. 71–78.
[29] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *IJCV*, vol. 116, no. 1, pp. 66–89, 2016.
[30] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *CGF*, vol. 28, no. 5, pp. 1383–1392, 2009.
[31] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *ICCV Workshops*. IEEE, 2011.
[32] A. Zaharescu, E. Boyer, and R. Horaud, "Keypoints and local descriptors of scalar functions on 2d manifolds," *International Journal of Computer Vision*, vol. 100, no. 1, pp. 78–98, 2012.
[33] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*. Springer, 2010, pp. 356–369.
[34] Q. Chen and V. Koltun, "Robust nonrigid registration by convex optimization," in *ICCV*, 2015, pp. 2039–2047.
[35] J. Starck and A. Hilton, "Correspondence labelling for wide-timeframe free-form surface matching," in *ICCV*. IEEE, 2007, pp. 1–8.
[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.
[37] D. Boscaini, J. Masci, E. Rodolà, and M. M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *NIPS*, 2016.
[38] J.-S. Franco and E. Boyer, "Efficient polyhedral modeling from silhouettes," *PAMI*, vol. 31, no. 3, 2009. [Online]. Available: https://hal.inria.fr/inria-00349103
[39] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer, 2013.
[40] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu, "Regression forests for efficient anatomy detection and localization in ct studies," in *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*. Springer, 2011.

[41] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
[42] T. Akenine-Möller, "Fast 3d triangle-box overlap testing," in *SIGGRAPH 2005 Courses*. ACM, 2005.
[43] W. Kehl, N. Navab, and S. Ilic, "Coloured signed distance fields for full 3d object reconstruction," in *BMVC*, 2014.
[44] 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 *ISMAR*. IEEE, 2011.
[45] A. Petrelli and L. Di Stefano, "On the repeatability of the local reference frame for partial shape matching," in *ICCV*. IEEE, 2011, pp. 2244–2251.
[46] C. S. Chua and R. Jarvis, "Point signatures: A new representation for 3d object recognition," *IJCV*, vol. 25, no. 1, pp. 63–85, 1997.
[47] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *IJCV*, vol. 89, no. 2-3, pp. 348–361, 2010.
[48] A. Petrelli and L. Di Stefano, "A repeatable and efficient canonical reference for surface matching," in *3DimPVT*. IEEE, 2012.
[49] 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.
[50] C.-H. Huang, F. Tombari, and N. Navab, "Repeatable local coordinate frames for 3d human motion tracking: From rigid to non-rigid," in *3DV*. IEEE, 2015, pp. 371–379.
[51] B. Allain, J.-S. Franco, and E. Boyer, "An efficient volumetric framework for shape tracking," in *CVPR*. IEEE, 2015. [Online]. Available: https://hal.inria.fr/hal-01141207
[52] L. Wang, F. Hétroy-Wheeler, and E. Boyer, "On volumetric shape reconstruction from implicit forms," in *ECCV 2016-European Conference on Computer Vision*, 2016.
[53] C.-H. Huang, B. Allain, J.-S. Franco, N. Navab, S. Ilic, and E. Boyer, "Volumetric 3d tracking by detection," in *CVPR*, 2016.
[54] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," in *TOG*. ACM, 2004.
[55] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *TPAMI*, vol. 24, no. 5, pp. 603–619, 2002.
[56] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *IJCV*, vol. 101, no. 3, pp. 437–458, 2013.
[57] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM, 2000.
[58] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," in *TOG*, 2007.
[59] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Eurographics*, 2004.
[60] M. Zollhöfer, M. Nießner, S. Izadi, C. Rhemann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger, "Real-time non-rigid reconstruction using an rgb-d camera," *TOG*, vol. 33, no. 4, 2014.
[61] 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," *TPAMI*, vol. 38, no. 8, pp. 1517–1532, 2016.
[62] E. Duveau, S. Courtemanche, L. Reveret, and E. Boyer, "Cage-based motion recovery using manifold learning," in *3DimPVT*, 2012, pp. 206–213.
[63] L. Wang, F. Hétroy-Wheeler, and E. Boyer, "A hierarchical approach for regular centroidal Voronoi tessellations," in *CGF*, 2015.
[64] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *CVPR*. IEEE, 2010, pp. 1704–1711.
[65] F. Bogo, J. Romero, M. Loper, and M. Black, "FAUST: Dataset and evaluation for 3D mesh registration," in *CVPR*, 2014, pp. 3794–3801.
[66] C.-H. Huang, E. Boyer, and S. Ilic, "Robust human body shape and pose tracking," in *3DV*. IEEE, 2013.

Chun-Hao Huang received the MSc degree in computer and communication engineering from National Cheng-Kung University in 2010. After one year in Academia Sinica as a research assistant, he started his doctoral study in Technische Universität München in 2012 and obtained the Ph.D. degree in 2016. His research interests include 2D/3D conversion, human motion capture and other 3D-vision related topics. He received Studying Abroad Scholarship from Taiwan government and the best paper award runner-up in 3DV'13.

Benjamin Allain received the MSc degree in computer science from Ensimag - Grenoble INP, France, in 2012. Then he joined the Morpheo group at Inria Grenoble Rhône-Alpes and obtained his PhD degree in computer science from Université Grenoble Alpes in 2017. Since November 2016, he has been working as a research scientist at Smart Me Up, France. His research interests include non-rigid motion tracking of 3D shapes from multiview video sequences.

Edmond Boyer is a senior research scientist at the INRIA where he leads the MORPHEO research team. He obtained his PhD from the Institut National Polytechnique de Lorraine in 1996. He was research assistant at Cambridge university in 1998 before joining the INRIA. His fields of competence cover computer vision, computational geometry and virtual reality. He has done pioneering work in the area of geometric 3D reconstruction with focus on objects with complex shapes like the humans. Edmond Boyer is co-founder of the 4D View Solutions (http://www.4dviews.com/), one of the leading companies worldwide specialized in multi-view acquisition and processing. His current research interests are on 3D dynamic modeling from images and videos, motion perception and analysis from videos.

Jean-Sebastien Franco is assistant professor of computer science at the Ensimag (School of Computer Science and Applied Mathematics, Grenoble Universities), and a researcher at the Inria Grenoble Rhône-Alpes and LJK lab, France, with the Morpheo team since 2010. He obtained his Ph.D. from the Institut National Polytechnique de Grenoble in 2005 with the Inria MOVI / Perception team. He started his professional career as a postdoctoral research assistant at the University of North Carolinas Computer Vision Group in 2006, and as assistant professor at the University of Bordeaux, with the IPARLA team, INRIA Bordeaux Sud-Ouest. His expertise is in the field of computer vision, with several internationally recognized contributions to dynamic 3D modeling from multiple views and 3D interaction.

Federico Tombari is a research scientist and team leader at the Computer Aided Medical Procedures Chair of the Technical University of Munich (TUM). He has co-authored more than 110 refereed papers on international conferences and journals in the field computer vision and robotic perception, on topics such as visual data representation, RGB-D object recognition, 3D reconstruction and matching, stereo vision, deep learning for computer vision. He got his Ph.D at 2009 from University of Bologna, at the same institution he was Assistant Professor from 2013 to 2016. He is a Senior Scientist volunteer for the Open Perception foundation and a developer for the Point Cloud Library, for which he served, in 2012 and 2014, respectively as mentor and administrator in the Google Summer of Code. In 2015 he was the recipient of a Google Faculty Research Award. His works have been awarded at conferences and workshops such as 3DIMPVT'11, MICCAI'15, ECCV-R6D'16.

Nassir Navab is a professor of computer science and founding director of Computer Aided Medical Procedures and Augmented Reality (CAMP) laboratories at Technische Universitt Mnchen (TUM) and Johns Hopkins University (JHU). He also has secondary faculty appointments at TUM and JHU medical schools. He received the Ph.D. degree from INRIA and University of Paris XI, France, and enjoyed two years of postdoctoral fellowship at MIT Media Laboratory before joining Siemens Corporate Research (SCR) in 1994. At SCR, he was a distinguished member and received the Siemens Inventor of the Year Award in 2001. In 2012, he was elected as a fellow member of MICCAI society. He received the 10 year lasting Impact Award of IEEE ISMAR in 2015. He holds 45 granted US patents and over 40 European ones. He has served on the program and organizational committee of over 80 international conferences including CVPR, ICCV, ECCV, MICCAI, IPCAI, and ISMAR. He has also been co-author of more than 20 awarded papers in international conferences. His current research interests include robotic imaging, computer aided surgery, computer vision and augmented reality.

Slobodan Ilic is currently senior key expert research scientist at Siemens Corporate Technology in Munich, Perlach. He is also a visiting researcher and lecturer at Computer Science Department of TUM and closely works with the CAMP Chair. From 2009 until end of 2013 he was leading the Computer Vision Group of CAMP at TUM, and before that he was a senior researcher at Deutsche Telekom Laboratories in Berlin. In 2005 he obtained his PhD at EPFL in Switzerland under supervision of Pascal Fua. His research interests include: 3D reconstruction, deformable surface modelling and tracking, real-time object detection and tracking, human pose estimation and semantic segmentation.