Volumetric 3D Tracking by Detection

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

▶ **To cite this version:**

Chun-Hao Huang, Benjamin Allain, Jean-Sébastien Franco, Nassir Navab, Slobodan Ilic, et al.. Volumetric 3D Tracking by Detection. CVPR 2016 - IEEE Conference on Computer Vision and Pattern Recognition, Jun 2016, Las Vegas, United States. pp.3862-3870, 10.1109/CVPR.2016.419 . hal-01300191

HAL Id: hal-01300191
https://inria.hal.science/hal-01300191

Submitted on 20 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Volumetric 3D Tracking by Detection

Chun-Hao Huang *[1], Benjamin Allain *[2], Jean-Sébastien Franco[2], Nassir Navab[1], Slobodan Ilic[1], and Edmond Boyer[2]

[1]Technische Universität München
[2]Inria, LJK, Univ. Grenoble Alpes
{huangc,ilics,navab}@in.tum.de, {firstname.lastname}@inria.fr

Abstract

In this paper, we propose a new framework for 3D tracking by detection based on fully volumetric representations. On one hand, 3D tracking by detection has shown robust use in the context of interaction (Kinect) and surface tracking. On the other hand, volumetric representations have recently been proven efficient both for building 3D features and for addressing the 3D tracking problem. We leverage these benefits by unifying both families of approaches into a single, fully volumetric tracking-by-detection framework. We use a centroidal Voronoi tessellation (CVT) representation to compactly tessellate shapes with optimal discretization, construct a feature space, and perform the tracking according to the correspondences provided by trained random forests. Our results show improved tracking and training computational efficiency and improved memory performance. This in turn enables the use of larger training databases than state of the art approaches, which we leverage by proposing a cross-tracking subject training scheme to benefit from all subject sequences for all tracking situations, thus yielding better detection and less overfitting.

1. Introduction

3D visual shape tracking aims to recover the temporal evolution of a 3D template shape using visual information. It finds applications in many domains including computer vision, graphics, medical imaging, and has proven successful for marker-less motion capture in recent years. A standard tracking process consists in an alternation of the following two steps. First, finding associations from each primitive of the observed data, *e.g.* 3D points acquired from camera systems, to corresponding primitives of the template 3D surface, typically based on the proximity in Euclidean space (ICP) [5] or a feature space. Second, given such associations, recompute the pose of the template under the constraint of a deformation model, typically based on kinematic skeletons [14, 20, 24, 27], or the piecewise-rigid assumption [2, 8], among others.

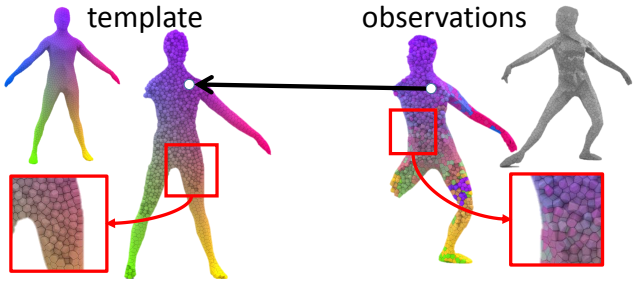


Figure 1: We represent 3D shapes using centroidal Voronoi tessellations. The volumetric cells of the observations are matched to cells of the template.

Recently, a number of alternative approaches and enhancements have been explored for both stages independently. On one hand, progress has been made in the deformation stage by introducing volumetric deformation models instead of purely surface-based ones. Thanks to its inherent local volume preservation property, this strategy has shown significantly improved robustness to various tracking situations, such as shape folding and volume bias of observed shapes. On the other hand, alternatives have also been proposed for the association problem by discovering them discriminatively using machine learning techniques [21, 24]. This in turn yields the possibility for 3D tracking techniques that are robust to partial tracking failure, while also improving the rate of convergence. Although surface-based features are used in many cases to describe local shapes and construct the associations, volumetric features have proven to be a promising direction for 3D shape description with surface-based templates [16], which we generalize to a fully volumetric pipeline.

*The first two authors contribute equally to this paper.

In this paper, we propose a unified volumetric pipeline, where the shape representation, deformation model, feature description, and primitive association are all built on a single volumetric representation, the centroidal Voronoi tessellation (CVT) [11]. Specifically, the observed and template shapes are all tessellated as a set of uniform and anisotropic cells (see Fig. 1), which bring benefits at all stages and yield a volumetric representation of regular cell shape and connectivity with controllable cell complexity.

While benefiting from local volume preservation properties inherent to this representation and the associated deformation model, we leverage the configurations of cells to build volumetric distance fields which we use to construct our volumetric feature space. On this basis, we propose a full framework to register a template shape to an observed shape, as two generic CVT cell sets. Because features are expressed in the volume, the proposed method is well suited to obtain fully volumetric detections, in turn helping the volumetric template tracking to be more robust. Thanks to its significantly low memory footprint, we use the representation to propose a multi-template learning framework, where large training sets can be assembled from multiple tracked action sequences for several human subjects. Specifically, every different subject's template is mapped to a generic, subject-agnostic template where the actual learning takes place, to benefit all subsequent tracked subjects. This framework consequently yields better or comparable detection and tracking performance than current state of the art 3D temporal tracking or tracking by detection methods.

2. Related work

3D tracking by detection. The tracking by detection strategy applied to human skeletal poses estimation (Kinect) [22] has shown robustness to tracking failure and reasonable convergence efficiency in real-world applications. It was first transposed to the problem of 3D shape tracking through the work of Taylor *et al*. [24] and presented similar targeted benefits, with the initial intention to substitute ICP-based optimization. The method achieves this goal by learning the mapping from input 3D points from depth sensors, to the human template surface domain, termed the Vitruvian manifold. This yields discriminative associations that replace the step of proximity search in ICP-based tracking methods. Variants of this work have explored changing the entropy function used to train random forests from the body-part classification entropy to the variance on surface embeddings for better data separation [20], or replacing surface-based features with 3D volume features computed on a voxel grid in a local coordinate frame [16]. Both increase the precision by finishing convergence with an ICP-based loop after the discriminative association stage. All these methods are nevertheless based on surface points, thus relying on heterogeneous shape representations, deformation models, target primitives and feature spaces. Our proposal builds a unified framework for all these purposes and takes advantage of volumetric tracking strategies as described below. Also, we introduce a multi-template strategy, where a template is assigned to each subject and mapped to a generic template, allowing to learn from all subject motions sequences for the benefit of any subsequent subject tracking task.

3D volumetric tracking. While many visual tracking techniques employ skeletons [14, 27] or surface-based representations [2, 17], volume-based representations have also been proposed to address various issues. On one hand, topology changes or online surface reconstructions are better handled if surfaces are implicitly represented in volumes as *e.g*. truncated signed distance field (TSDF) [13, 19, 18], with high memory cost due to regular grids storing empty space information. On the other hand, volumetric techniques have also been devised for robustness in long term tracking, as a way to alleviate the so-called *candy-wrapper artifacts*, namely, collapsing surfaces in animations. Without explicitly tessellating surface interiors, Zhou *et al*. [30] introduce internal nodes to construct a volumetric graph and preserve the volumes by enforcing Laplacian constraints among them. Instead, Budd *et al*. [7] and De Aguiar *et al*. [10] perform a constrained tetrahedralization on surfaces to create interior edges. Allain *et al*. [1] generate internal points by CVT decomposition and thereby propose a generative tracking strategy that yields high quality performance. These techniques are nevertheless based on ICP-variants, whereas we aim at detecting associations discriminatively.

3D features. In many cases, surface-based features are used for recognition or shape retrieval, such as heat kernel signatures (HKS) [23] and wave kernel signatures (WKS) [3]. Both exploit the Laplacian-Beltrami operator, the extension of the Laplacian operator to surface embeddings. These features are nonetheless known for their lack of resilience to artifacts present in noisy surface acquisitions, especially significant topology changes. MeshHoG [29] and SHOT [25] attach a local coordinate frame at each point to achieve invariant representations and reach better performance for noisy surfaces. More detailed reviews can be found in [6] and [15] for triangular surfaces and point clouds, respectively. In the context of discriminative 3D tracking, depth difference features have been used to build random forests on depth data [22, 24]. One common trait of the aforementioned features is that the computation involves only surface points. Huang *et al*. [16] show that features can be built based on local coordinate frames in a regular-grid volume. However, these features are only computed on surface vertices and do not address the need for fully volumetric correspondences as proposed in our work.

3. Preliminaries and method overview

Given a volumetric domain Ω defined by a shape in $\mathbb{R}^3$, CVT is a clipped Voronoi tessellation of Ω which holds the property that the seed location of each cell coincides with its center of mass. Cells are of regular size and shapes as in Fig. 1. A surface is expressed as the border of Ω, *i.e.* $\partial\Omega$.

Let $\mathcal{S}$ denote the set of all cell centroids. Both the template shape Ω_M and the observed data Ω_Y are expressed by their CVT samplings, $\mathcal{S}_M$ and $\mathcal{S}_Y$ with locations $\mathbf{M} \subset \Omega_M$ and $\mathbf{Y} \subset \Omega_Y$ using the method in [28]. We adopt a volumetric deformation framework [1] that groups cells into K clusters, each having a rigid transformation $\mathbf{T}_k \in SE(3)$. The collection of all transformations, $\mathbf{T} = \{\mathbf{T}_k\}_{k=1}^K$, encodes the *pose* of the shape. As a result, the problem amounts to estimating the best $\hat{\mathbf{T}}$ such that the deformed template cells $\mathbf{M}(\hat{\mathbf{T}})$ resembles $\mathbf{Y}$ as much as possible. Matching cells $i \in \mathcal{S}_Y$ with cells $s \in \mathcal{S}_M$ is therefore an indispensable task. To this end, each point in $\mathbf{Y}$ is first mapped to the template domain Ω_M, where the closest point in M is sought as the correspondence (as represented by the green line in Fig. 3). This mapping $\mathbf{r} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is accounted for by a regression forest which is learned off-line with many pre-tracked CVTs (§ 4.1). Given the detected associations, the best pose $\hat{\mathbf{T}}$ is estimated using an EM-ICP algorithm (§ 4.2).

4. Learning and tracking

4.1. Learning

We explain in this section how to learn the mapping $\mathbf{r} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ from the observation domain to the template domain with a regression forest [9], which is a set of T binary decision trees. An input cell is first described as a feature vector $\mathbf{f}$ in § 4.1.1. Taking $\mathbf{f}$ as input, during training each tree learns the split functions that best separate data recursively at branch nodes, while during testing the cell is routed through each tree, reaching T leaves that store statistics (a mode in $\mathbb{R}^3$ in our case) as predictions (§ 4.1.2). We first discuss the scenario with one single template and then generalize to multiple ones in § 4.1.3.

4.1.1 Feature

The feature $\mathbf{f}$ we use for building trees is designed with several principles in mind. In order to be discriminative for shape matching, our feature 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 SHOT [25]. For time and memory efficiency of forest training and prediction, we want our feature vector coefficients to be computable separately. This requirement is not met by the descriptors that rely on unit length normalization. In order 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 Euclidean distance from cell centroids s to the surface $\partial\Omega$: $d(s, \partial\Omega) = \min_{p \in \partial\Omega} d(s, p)$ because it naturally encodes the relative location with respect to the surface and it is invariant to rotations, translations and quasi-invariant to changes of poses. Finally, our feature needs to be robust to the topological noise present in the input data.

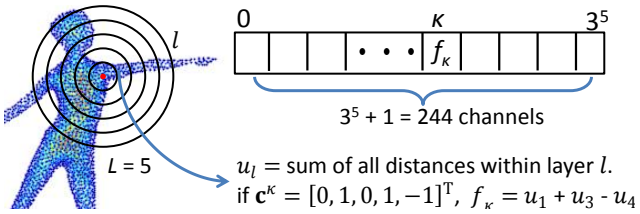


Figure 2: Right: the distance field defined by a CVT sampling $\mathcal{S}$, where each cell stores the distance $d(s, \partial\Omega)$. Blue to red colors means from close to far. Red dot: cell center s to be described. Left: illustration of our feature $\mathbf{f}$. $L = 5$ in this toy example. See text for more explanations.

Given a distance field defined by a CVT sampling $\mathcal{S}$, our feature is similar in spirit to Haar feature in the Viola-Jones face detector [26], except that the rectangular neighborhood is replaced with a sphere. As visualized in Fig. 2, 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 $\mathcal{C} = \{-1, 0, 1\}$. Formally, suppose $\mathbf{c}$ are L-dimensional vectors whose elements are the bootstrap samples of $\mathcal{C}$. Let $\mathbf{c}^\kappa$ denote one particular instance of $\mathbf{c}$, *i.e.*, $\mathbf{c}^\kappa \in \mathcal{C}^L$. 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 $\mathcal{C}^L$ and each element f_κ is defined as:

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

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 $d(s, \partial\Omega)$, $\mathbf{f}$ inherits the invariance to rigid-body motions. In addition, we normalize distances by their standard deviation in one surface, achieving scale invariance 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

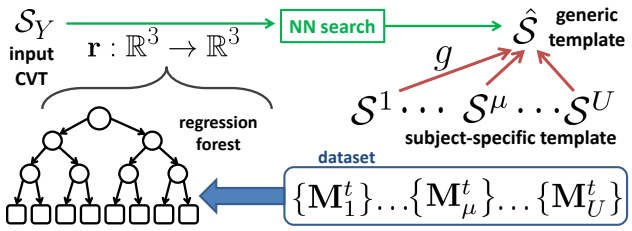


Figure 3: The schematic flowchart of the multi-template learning framework. Red arrows: mappings g^μ that associate the indices from each subject-specific template $\mathcal{S}^\mu$ to the common template $\hat{\mathcal{S}}$. $\mathbf{M}_\mu^t$ are the temporal evolutions of each template. Blue: training; green: prediction.

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.

4.1.2 Training and prediction

The aim of forests is to map an observed cell to the template domain Ω_M, typically chosen to be in the rest pose. Given a set of CVTs corresponding to the template Ω_M deformed in various poses, we associate each cell $s \in \mathcal{S}_M$ to its locations at the rest pose, denoted as $\mathbf{x}_s^0 \in \mathbf{M}^0$, forming a pool of sample-label pairs $\{(s, \mathbf{x}_s^0)\}$ as the dataset. Suppose $\mathcal{D}_N$ is the set of samples arriving at a certain branch node. The training process is to partition $\mathcal{D}_N$ recursively into two subsets $\mathcal{D}_L$ and $\mathcal{D}_R$ by simple thresholding on a chosen feature channel. Our splitting candidate $\phi = (\kappa, \tau)$ is therefore the pair of thresholds τ and feature attribute indices κ in Eq. 1. In branch nodes, many candidates ϕ are randomly generated and the one that maximizes the information gain G, $\phi^* = \arg\max_\phi G(\phi)$, is stored for the later prediction use.

We use the typical definition of information gain:

$$G(\phi) = H(\mathcal{D}_N) - \sum_{i \in \{L,R\}} \frac{|\mathcal{D}_i(\phi)|}{|\mathcal{D}_N|} H(\mathcal{D}_i(\phi)), \quad (2)$$

where H is the entropy, measured as the variance in Euclidean space, *i.e.* $H = \sigma^2$. We do not apply the more sophisticated measure [20] because (1) our continuous labels $\mathbf{x}_s^0$ lie in a volumetric domain Ω and (2) templates are usually chosen in canonical T or A poses. The Euclidean approximation holds more naturally here than in [16, 20], where the regression is performed along the surface manifold. 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 is performed in a leaf node to represent the distributions of $\mathbf{x}_s^0$ as a set of confidence-weighted modes $\mathcal{L} = \{(\mathbf{m}, \omega)\}$. $\mathbf{m} \in \mathbb{R}^3$ is the mode location and ω is a scalar weight.

In the prediction phase, a cell $i \in \mathcal{S}_Y$ traverses down the trees and lands on T leaves containing different collections of modes: $\{\mathcal{L}_1 \cdots \mathcal{L}_T\}$. The final regression output $\mathbf{r}_i$ is the cluster centroid with largest weight obtained by performing mean-shift on them. Each observed cell then gets a closest cell p in the reference $\mathcal{S}_M$: $p = \arg\min_{s \in \mathcal{S}_M} \left\| \mathbf{r}_i - \mathbf{x}_s^0 \right\|_2$. The correspondence pair (i, p) serves as input to the volumetric deformation framework described in § 4.2.

4.1.3 Multi-template learning

The above training scenario requires deformed CVTs of consistent topology such that one can easily assign each cell sample s a continuous label which is its rest-pose position $\mathbf{x}_s^0$. It hence applies only to one template. However, the amount of training data for one single template is often limited because a fully volumetric shape and pose modeling framework is still an open challenge. 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 strategy that learns across different CVT topologies.

Given U distinct CVT templates: $\{\mathcal{S}^\mu\}_{\mu=1}^U$[1], whose temporal evolutions are recovered with the method in [1], resulting in a collection of different templates deformed in various poses: $\{\{\mathbf{M}_1^t\} \cdots \{\mathbf{M}_U^t\}\}$ 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 in § 4.1.2 can again be applied. In other words, we align topologies by matching every template $\mathcal{S}^\mu$ to $\hat{\mathcal{S}}$. Fig. 3 depicts this multi-template learning scheme.

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

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 [4], 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

[1] The template suffix M is dropped to keep notations uncluttered.

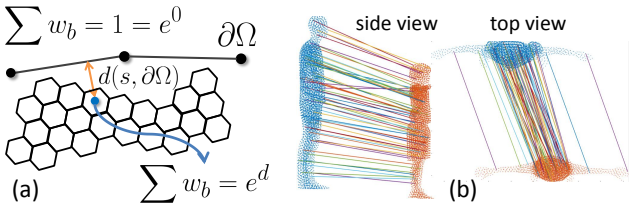


Figure 4: (a): illustration of our strategy adapting skinning weights to CVT cells. Distances $d(s, \partial\Omega)$ are reflected in normalizations. (b): result of matching two templates.

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 distance to the surface is d, its skinning weight is simply the one of its closest surface point[2], scaled by e^d. 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. 4(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` [4] to computes skinning weights, extend them from surface vertices to CVT cells, and match all cells to those of the common template $\hat{\mathcal{S}}$. The resulting skeletons are not used in our method. Fig. 4(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.

4.2. Tracking

We elaborate in this section on how to apply our regression forest to track a sequence of temporally inconsistent observations. The current state-of-the-art 3D shape tracking methods usually employ non-rigid ICP algorithms [2]. Instead of performing an extensive search over all possible associations, we directly use the correspondence pair (i, p) detected by the forest as initializations. This results in a faster pose estimation. We adopt the CVT-based deformation framework proposed in [1]. However, the approach we describe can easily be adapted to other ICP variants.

[2]When the shortest distance does not exactly correspond to a vertex but to a point in the middle of a triangle, we use barycentric coordinates as the coefficients to linearly combine the skinning weights of the three vertices.

4.2.1 Bayesian tracking

Bayesian tracking such as [2] consists in maximizing the *a posteriori* probability $P(\mathbf{T}|\mathbf{Y})$ of the pose parameters $\mathbf{T}$ given the observations $\mathbf{Y}$. It can be further simplified as $P(\mathbf{T}|\mathbf{Y}) \propto P(\mathbf{T}, \mathbf{Y}) = P(\mathbf{T}) \cdot P(\mathbf{Y}|\mathbf{T})$, where the deformation prior $P(\mathbf{T})$ discourages the implausible poses and the likelihood term $P(\mathbf{Y}|\mathbf{T})$ expresses the compatibility between the observations and the pose estimate $\mathbf{T}$. Since maximizing a probability $P(\cdot)$ is equivalent to minimizing $-\log P$, it leads us to the following problem:

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} \left(-\log P(\mathbf{T}) - \log P(\mathbf{Y}|\mathbf{T})\right). \quad (3)$$

In EM-ICP algorithms [8], the conditional likelihood $P(\mathbf{Y}|\mathbf{T})$ is expressed by introducing a set of latent selection variables $\{k_i\}_i$ that explicitly associate the cell k_i of the deformed template model to the observed cell i.

The prior on the latent association variables is usually uniform, which means that an observed point can be associated to any template point with the same probability. This leads to a long exhaustive search among all possible associations and produces a high number of residuals, slowing down the EM-ICP algorithms. Moreover, it is the source of wrong associations that guides the optimization to suboptimal local minimum.

4.2.2 EM-ICP with forest predictions

With a small number of possible associations provided by forests, our algorithm averts the need for an exhaustive search, and therefore highly decreases the running time of each optimization iteration. Moreover, it removes a lot of wrong association hypotheses. We integrate the predictions from the forest as a prior on the selection variable k. The selection variable k_i (for the observed cell i) follows a probability distribution where only the cell predicted by the forest has a non-zero probability.

Usually the forest outputs only one prediction per cell, which is at the mode with higher weight resulting from the mean-shift algorithm. However, because of the symmetry, the good match is often not the mode with highest weight. Thus, it makes sense to consider several modes instead of one in the prediction phase. The robust scheme described in the next section will usually select the good one.

4.2.3 Robustness

The detection forest sometimes outputs wrong correspondences, either due the symmetry of human bodies (left-right confusion) or other detection errors (*e.g.* hand-foot confusion). Therefore, the ICP algorithm needs to be robust to wrong correspondences. We achieve this goal by using a

Template / #Vertex / #Cell	Sequence	Frames
Ballet / 6844 / 5000	*Seq1* [1]	500
	Seq2	936
Goalkeeper / 5009 / 5000	*SideJump* [1]	150
	UpJump [2]	239
Thomas / 5000 / 4998	*Seq1*	1500
	Seq2	1400

Table 1: Sequences used in our experiments. For each subjects, the training set is the random 250 tracked CVTs sampled from first sequences and testing on the unseen second sequence. Unreferenced sequences are the ones proposed in this paper.

noisy observation model [8], where the noise variance is estimated by an EM algorithm.

5. Experimental results

We validate our approach with numerous multi-view sequences, whose profiles are summarized in Table 1. For each frame, a coarse visual hull is reconstructed by a shape-from-silhouette method [12], followed by [28] to draw CVT samplings (raw CVTs). Given a CVT template, we then perform an EM-ICP based method [1] on the raw CVTs to recover temporal coherent volumetric deformations (tracked CVTs). We evaluate our method in two aspects: detection accuracy (§ 5.1) and tracking results (§ 5.2). Unless otherwise specified, we follow the experimental protocol below.

Experimental protocol. We first explain the settings common to two experiments. For each subject, up to 250 tracked CVTs are randomly chosen from the first sequence as the training dataset, while the second sequences are completely left out for testing. We open $L = 8$ sphere layers for the feature computation. Each tree is grown with 30% bootstrap samples randomly chosen from the dataset and trees are grown up to depth 20.

Two experiments, however, differ in the input data for testing. To evaluate the quality of estimated associations, we feed the tracked CVTs into forests due to the availability of ground truth indices (§ 5.1), whereas raw CVTs are used as the input for tracking experiments in § 5.2. Some distinct experimental settings of the two are exposed in Table 2.

5.1. Matching

The contributions of CVT on improving the correspondences detection are evaluated with two experiments. First, we follow the learning framework in [16] but replace their voxel-based features with ours in § 4.1.1, denoted as *CVTfeature*. Next, we further change the regression domain from surfaces to volumes, as described in § 4.1.2 (*fullCVT*). We test on the tracked CVTs and report the results on all frames of training sequences (Tr) and testing ones (Te). The drop between them is a natural phenomenon for every machine learning algorithm and indicates the ability to generalize. If the Euclidean distances between the predicted cell index and the ground truth are smaller than a certain threshold, it is considered as correct.

Sect.	Forest	T	Testing data
§ 5.1	template-specific[3]	20	1. tCVTs of seq1 (Tr) 2. unseen tCVTs of seq2 (Te)
§ 5.2	multi-template	50	unseen rCVTs of seq2

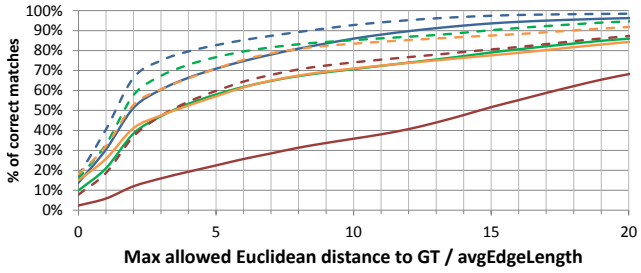
Table 2: Different experimental settings in two sections. tCVTs stand for tracked CVTs while rCVTs represent raw CVTs.

Single-template learning. To align the experimental setting, here the regression forests are subject-specific and consist of only $T = 20$ trees. Fig. 5 shows the percentage of correct matches in varying thresholds for *Thomas* and *Ballet*. Since *CVTfeature* and [16] 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 [16] (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. 7.

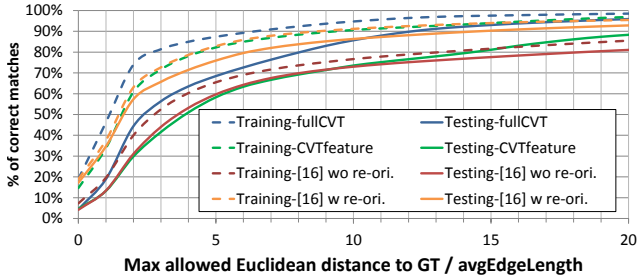
Discussion. It is worth a closer analysis to compare our approach against [16]. Compared to volumes of regular grids, CVT is certainly a more memory-efficient way to describe 3D shapes. In practice, [16] describes each mesh with 150^3 voxels, while we need only 5k cells[4]. Consequently, [16] is not able to include a sufficient amount of training shapes, leading to a major drawback that forests are limited to one single subject and learn merely pose variations. To further decrease the needed number of training meshes, [16] exploits skeletal poses to cancel the global orientation. This in turn makes every mesh in the training dataset face the same direction. During tracking the input data has to be re-oriented likewise using the estimated skeletal poses from the last frame. Our approach, on the other hand, considers distance fields of CVTs which is naturally invariant to rotations and hence does not require re-orientations. We anyway compare to [16] in both set-

[3]More precisely, forests in § 5.1 are all single-template based except for the one in "multi-template learning" paragraph.

[4]Further note that [16] 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) *Thomas*



(b) *Ballet*

Figure 5: Cumulative matching accuracy of different approaches. The x-axis is normalized with respect to 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.

tings. Orange curves in Fig. 5 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 [16] and needs just one forest for different subjects in the following experiment.

Multi-template learning. We use the sequences of *Goalkeeper* to verify the advantages of the multi-template learning strategy in § 4.1.3. 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. 6 the correctness of correspondences in *fullCVT* setting. Both curves represent the accuracy on testing *UpJump* sequence. The blue curve corresponds to a forest only trained with *Goalkeeper* tracked CVTs, whereas the green curve corresponds to a forest trained with tracked CVTs of *Ballet* and *Thomas*. For both forests, *UpJump* sequence is unseen during training. Compared with the forest of the blue curve, the one of the green curve is trained with twice the amount of meshes from different subjects, and yet it leads to better prediction accuracy on unseen testing poses. This suggests that including more variation of motions indeed results in better generalization to unseen data. It also confirms the necessity and efficacy of our multi-template

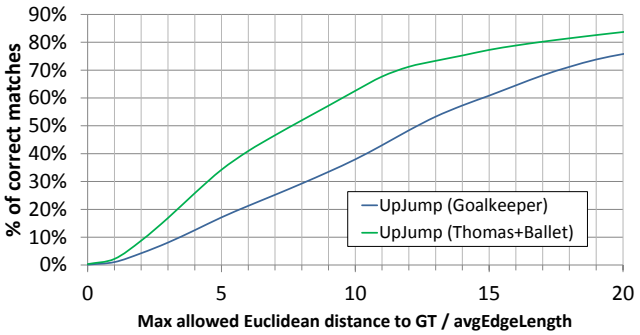


Figure 6: Cumulative matching accuracy of single and multi-template strategy on *Goalkeeper*.

strategy. We anyway point out that due to the lack of adequate amount of training data, these encouraging preliminary results need to be confirmed on datasets consisting of more subjects and sequences.

5.2. Tracking

We perform several experiments to evaluate our whole tracking-by-detection algorithm and compare with previous approaches using two quantitative metrics. We also show its resilience to large pose changes and its generalization capacities on an unknown subject.

Unlike § 5.1, here we apply the multi-template strategy in § 4.1.3 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 which have no temporal coherence. Correspondences are predicted by the forest and fed into the volumetric deformation framework described in § 4.2. The number of clusters K is 250 for *Ballet* and *Goalkeeper* and 150 for *Thomas*. Some visual results are shown in Fig. 8 and in the supplemental video[5]. With the help of regression forests, our approach is able to discover volumetric associations even in challenging poses found in *Thomas* and deform the templates successfully.

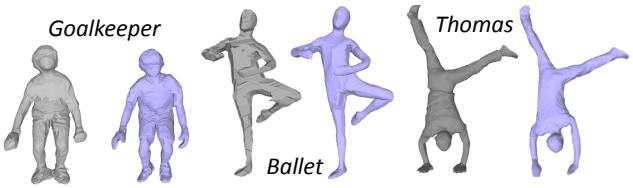


Figure 8: Qualitative tracking results. Gray: input observed visual hulls; purple: deformed templates.

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

Figure 7: Qualitative matching results on the raw CVTs. Templates are displayed at the upper left corner. Best viewed in pdf.

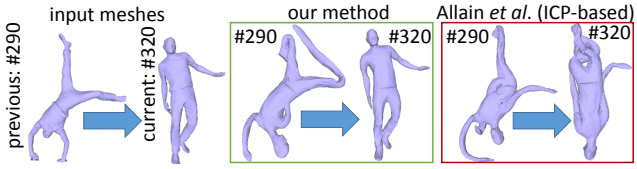


Figure 9: Tracking results of *Thomas* dataset at low frame rate.

Quantitative evaluation and comparison. We evaluate the tracking results with two complementary metrics: silhouette overlap error, which measures the consistency between the shape and observed silhouettes, and marker location error (using marker-based motion capture data), which sparsely evaluates the surface pose. Numerical results, which can be found in the supplemental paper, show similar or improved results with respect to volumetric ICP-based tracking [1] and surface-based tracking by detection [16].

Tracking at low frame rate. One of the expected benefits of our framework over purely ICP-based methods is improved resilience with large pose changes. We test this assertion by tracking the *Thomas* sequence at low frame rate (5fps). Figure 9 shows how our method recovers from tracking failures while [1] does not. This improvement is confirmed by the median silhouette overlap pixel error, which we found to be twice lower with our method (10054 pixels compared to 19998 pixels).

Testing with a new subject. We tested the generalization capacities of our framework with a subject (*Dancer* dataset [2]) which is not in the training data. For this purpose, one can either select an existing template from the training sequences, or build a template model by matching one of the samples from the test sequence to the common reference model using skinning weights, as we do in multi-template training. We use the latter, which is more subject specific and can be expected to yield better results. Most poses are correctly tracked in our experiment (see Fig. 10). Not unexpectedly for this type of approach, some failures occur on more complex poses unseen in training data and would probably be improved with a larger training set.

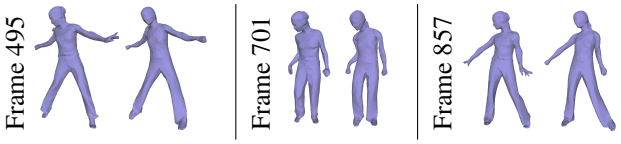


Figure 10: Tracking results with a new subject, *Dancer* dataset. Input mesh (left) and tracked mesh (right).

6. Conclusion

In this paper, we present a fully volumetric tracking-by-detection framework. Centroidal Voronoi tessellation is chosen to be the unified representation used in feature computations, predicting domains, and deformation models. Such informative and consistent representations have shown better detected correspondences than other discriminative strategies. We further devise a multi-template learning strategy to enrich the training variation. This leads to one single forest for different subjects and yields cross-subject learning of discriminative associations. The method opens several research directions, and thanks to low memory-footprint characteristics, it can be tested on much larger training sets for discriminative 3D tracking in the future. The methodology can easily be transposed to other volumetric features emphasizing other discriminative characteristics.

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

References

[1] B. Allain, J.-S. Franco, and E. Boyer. An efficient volumetric framework for shape tracking. In *CVPR*. IEEE, 2015.

[2] B. Allain, J.-S. Franco, E. Boyer, and T. Tung. On mean pose and variability of 3d deformable models. In *ECCV*. Springer, 2014.

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

[4] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. In *TOG*, 2007.

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

[6] 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*, pages 71–78. Eurographics Association, 2011.

[7] C. Budd and A. Hilton. Skeleton driven Laplacian volumetric deformation. In *CVMP*, 2009.

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

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

[10] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *TOG*, 2008.

[11] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM review*, 41:637–676, 1999.

[12] J.-S. Franco and E. Boyer. Efficient polyhedral modeling from silhouettes. *PAMI*, 31(3), Mar. 2009.

[13] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi. Locally rigid globally non-rigid surface registration. In *ICCV*. IEEE, 2011.

[14] 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.

[15] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3d local feature descriptors. In *IJCV*. Springer, 2015.

[16] 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.

[17] M. Klaudiny, C. Budd, and A. Hilton. Towards optimal non-rigid surface tracking. In *ECCV*. Springer, 2012.

[18] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015.

[19] 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.

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

[21] E. Rodola, S. R. Bulo, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *CVPR*. IEEE, 2014.

[22] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*. IEEE, 2011.

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

[24] 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.

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

[26] P. Viola and M. J. Jones. Robust real-time face detection. In *IJCV*. Springer, 2004.

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

[28] L. Wang, F. Hétroy-Wheeler, and E. Boyer. A hierarchical approach for regular centroidal Voronoi tessellations. In *CGF*, 2015.

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

[30] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum. Large mesh deformation using the volumetric graph Laplacian. In *SIGGRAPH*. ACM, 2005.