Human Shape and Pose Tracking Using Keyframes

Chun-Hao Huang[§], Edmond Boyer[†], Nassir Navab[§], Slobodan Ilic[§]

[§]Technische Universität München
[†]LJK-INRIA Grenoble Rhône-Alpes
{huangc,slobodan.ilic,navab}@in.tum.de, edmond.boyer@inria.fr

Abstract

This paper considers human tracking in multi-view setups and investigates a robust strategy that learns online key poses to drive a shape tracking method. The interest arises in realistic dynamic scenes where occlusions or segmentation errors occur. The corrupted observations present missing data and outliers that deteriorate tracking results. We propose to use key poses of the tracked person as multiple reference models. In contrast to many existing approaches that rely on a single reference model, multiple templates represent a larger variability of human poses. They provide therefore better initial hypotheses when tracking with noisy data. Our approach identifies these reference models online as distinctive keyframes during tracking. The most suitable one is then chosen as the reference at each frame. In addition, taking advantage of the proximity between successive frames, an efficient outlier handling technique is proposed to prevent from associating the model to irrelevant outliers. The two strategies are successfully experimented with a surface deformation framework that recovers both the pose and the shape. Evaluations on existing datasets also demonstrate their benefits with respect to the state of the art.

1. Introduction

Marker-less human motion capture consists in tracking human shape and pose using visual information. This has become an important research area with many applications in motion analysis or digital content production. Perhaps the most widespread approach to solve this problem is to deform a pre-defined reference surface so as to fit data derived from image observations, *e.g.* silhouettes or 3D points. This model-based strategy has demonstrated a good success over the past few years [6, 11, 12, 15, 18, 21], because of its ability to enforce strong consistencies over time through the prior models of shape and deformation.

However, this strategy still relies on the assumption that image observations are complete and relevant, *i.e.* they do not describe another shape. In practice, it appears to be difficult to maintain such assumption when considering more realistic dynamic scenarios and with fewer constraints on the capture environment. As shown in Fig. 1, top row, background subtraction is often erroneous with the consequence of missing data, *e.g.* the missing arm. Another type of errors occurs when image observations describe a shape that is not in consideration, and can therefore mislead the surface as well, *e.g.* the chair in Fig. 1 bottom row. Our aim is to propose an alternative strategy that better handles such situations and hence contributes to the general objective of unconstrained human motion tracking in real environments.

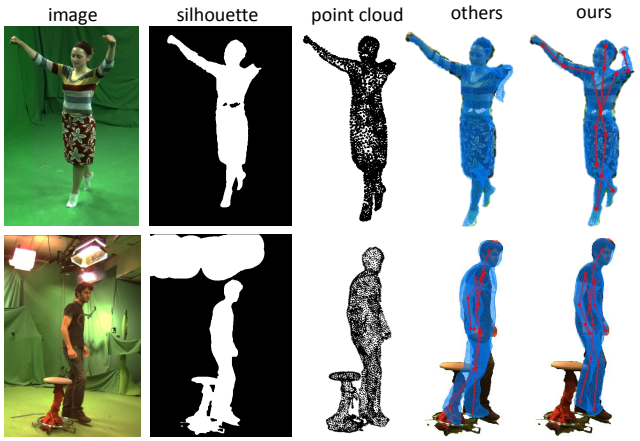


Figure 1. Our approach recovers the shape and the pose of the subject despite missing data (top row) and outliers (bottom row) whereas other approaches (top: [6] and bottom: [17] + [18]) fail.

Our framework is inspired by keyframe-based methods, *e.g.* [20] in camera/object tracking, and by non-sequential surface registration methods [4, 14]. In these works the tracking task is eased by reducing the discrepancy between the model and the input data to be matched. We exploit a similar idea that consists in having multiple reference shape models that can be fit to the observations. Numerous existing approaches rely on a one-fit-all strategy where a single

reference model is deformed to fit to all observations. This strategy is likely to fail when observations describe a shape significantly different from the model, due to the presence of missing data or outliers. Instead, we propose to build a set of reference models called *keyframes*, which correspond to several representative shapes and poses that have been explored during tracking. They are identified online using mean-shift clustering and without the need for offline pre-processing. At each frame, the *best* keyframe is chosen as the reference model. We combine this strategy with a robust surface deformation method. Comparisons with the state-of-the-art confirm the advantages of this approach with real and inaccurate data.

This paper has several contributions. First we introduce the notion of keyframes in 3D human motion tracking, and further propose a keyframe-based tracking framework that updates the keyframe pool incrementally. Second, a new outlier rejection method is presented with the benefit of high integrability into previous probabilistic surface deformation framework. Both contributions increase the robustness and significantly limit the impact of missing data and outliers. To evaluate our method, we recorded new sequences that include static outliers. To the best of our knowledge, none of the current public dataset presents such feature.

2. Related work

Existing methods that track human poses and shapes generally express the problem as maximum *a posteriori* (MAP) estimation which involves a data term modeling the likelihood of the estimation and a regularization term modeling the adequacy to the prior information. Methods differ then by the input data and the assumed prior knowledge.

2.1. Data term

Data terms measure how well the model explains the observations. In general, silhouettes, point clouds, and photometric information are considered for this purpose.

Silhouettes. Many approaches deform the model such that the contour of the projected surface coincides with the contour of the observed silhouettes, *e.g*. [7, 11, 18, 21]. In [11], Gall *et al*. consider also photometric information to establish 3D-2D model-data correspondences. Later in [15], additional image segmentation information are used to differentiate multiple interacting subjects. In these works, silhouette overlap error is often regarded as a standard error measure, which is sensible only when silhouettes are accurate and fully describe shapes. Also note that distances in 2D images do not necessarily reflect distances in 3D and small errors along the silhouette contour can correspond to large distances in 3D. As a result, some authors, *e.g*. [6,12] advocate for considering 3D points as input data.

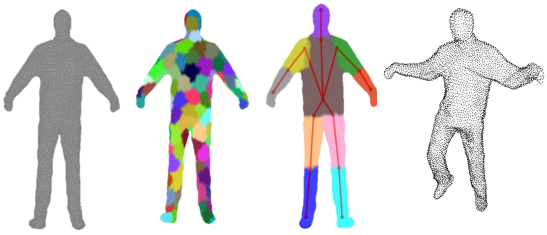
Point clouds. Given a set of points reconstructed from multiple silhouettes, *i.e*. [9, 10], some authors first estimate correspondences between the model and the 3D observations, and then deform the model accordingly. Although the reconstruction suppresses artifacts resulting from 2D noise, it also introduces new errors, such as missing body parts or fake geometry elements like ghost limbs. To robustly track in the presence of outliers, Huang *et al*. [12] train a linear support vector machine (SVM) that classifies the input data into different body parts. Outliers are then rejected based on the posteriors given by the SVM classifier. This approach depends heavily on the classifier, and is time-consuming since the SVM must be trained at each frame. In [6], Cagniart *et al*. model outliers as an additional component of a Gaussian Mixture Model (GMM) equipped with a uniform distribution defined *a priori*. The adjustment of this distribution is however difficult and has a strong influence on the results. We propose a more robust outlier rejection that does not depend on user defined parameters.

2.2. Regularization term

Evolving a surface with discrete observations is ambiguous by nature and some prior information on the model is usually required. This information varies from generative spatial shape models to discriminative models that are learned from already known shapes and poses.

Spatial shape models. Several works employ Laplacian coordinates [16] to preserve local shapes, *e.g*. [11,18], while others define a rigidity term that serves similar purposes, *e.g*. [5]. Note that all these methods refer to a single static reference shape to constrain local deformations. This reference shape model is usually in rest pose and built prior to the tracking [6, 7, 21]. However, the observations can significantly deviate from the reference model along time. The shapes and poses that were already recovered during tracking can help in that respect, which motivates the multiple-keyframe tracking framework presented in this paper.

Learned deformation models. A few works also make benefit from pre-collected information to help the tracking. They seek to learn the possible deformations in advance to regularize the results. In a non-sequential strategy, Budd *et al*. [4] and Klaudiny *et al*. [14] assume that the complete input sequence is available beforehand and they find the best order to traverse it using a minimum spanning tree algorithm. Duveau *et al*. [8] propose a supervised learning strategy that regularizes the results based on the learned distribution in a latent parameter space. These methods require a pre-processing step either to build a shape-similarity tree from the input sequence [4, 14], or to learn a low-dimensional representation from the gathered motion training data [8, 19].



(a) (b) (c) (d)

Figure 2. (a-c): reference model. (d): observations. (a) is a surface with N_v vertices. In (b) the surface is decomposed into N_p patches. (c) shows a rigged skeleton with N_j joints, and the associations between vertices and joints. (d) is a point cloud with N_y points. Typically, $N_v \approx 4$K, $N_p \approx 150$, $N_j = 15$, and $N_y \approx 9$K.

Our multiple-keyframe approach also exploits temporal information. As opposed to the mentioned approaches, we learn the reference models online in an unsupervised manner and do not require any preliminary step. Note anyway that our framework could also take advantage of already tracked sequences and generate the keyframe pool offline.

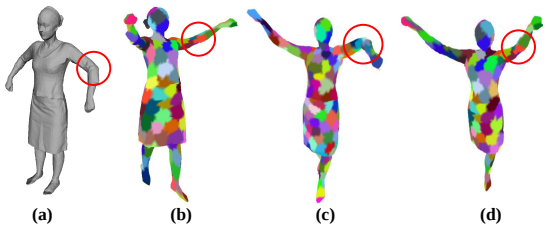
3. Overview

In this section we state our problem and give an overview of the proposed method. At every time frame t, a point cloud $\mathcal{Y}^t = \{y_i^t\}_{i=1:N_y}$ is reconstructed from silhouettes using `EPVH` [9]. y_i^t is a 6D vector that contains 3D spatial coordinates $\mathbf{y}$ and normal coordinates. The goal is to deform a reference model such that it fits the observations $\mathcal{Y}$. Our model comprises a reference triangle surface and an intrinsic skeleton. We adopt the patch-based model proposed in [5], where vertices are grouped into N_p patches, as shown in Fig. 2(b). Our skeleton is a tree structure of N_j nodes (3D joints) and the root is set at the pelvis, as shown in Fig. 2(c). The skeleton is rigged into the mesh using `Pinocchio` [2], which gives the associations between vertices v and joints j. Deformations are parameterized with respect to: (i) the shape of the surface; (ii) the pose of the skeleton. The shape parameters $\Theta = \{(\mathbf{R}_k, \mathbf{c}_k)\}_{k=1:N_p}$ are the orientation and position pair for each patch k and encode the deformation of the reference mesh model. The pose parameters $\mathcal{J} = \{\mathbf{x}_j\}_{j=1:N_j}$ are the 3D joint positions of the skeleton. Given the parametrization, the problem is formulated as the maximization of the joint probability distribution of the data and model:

$$\max_{\Theta,\mathcal{J}} P(\mathcal{Y}, \Theta, \mathcal{J}). \tag{1}$$

This above distribution can be decomposed into $P(\mathcal{Y}|\Theta) \cdot P(\mathcal{J}|\Theta) \cdot P(\Theta)$, which represents respectively the likelihood of the shape given the observations, the probability of the pose given the shape, and the prior knowledge on shape deformations. Hence Eq. 1 can be rewritten as:

$$\min_{\Theta,\mathcal{J}} [E_r(\Theta) + E_{bone}(\Theta, \mathcal{J}) - \ln\ P(\mathcal{Y}|\Theta)], \tag{2}$$



(a) (b) (c) (d)

Figure 3. Illustration of multi-keyframe advantages. (a) reference surface of *Skirt* [11] as the first keyframe. (b) second keyframe identified at $t = 95$. At $t = 102$, the left arm is missing in the observations, as in Fig. 1. Using (a) as the reference yields (c) while using (b) yields (d).

where $E_r(\Theta) = -\ln P(\Theta)$ is the rigidity energy in [6] and $E_{bone}(\Theta, \mathcal{J}) = -\ln P(\mathcal{J}|\Theta)$ is the bone-binding energy in [12], both of which behave like regularization terms. The likelihood $P(\mathcal{Y}|\Theta)$ is similar to [6] and uses Gaussian Mixture Model (GMM) where every patch explains every observations y_i according to:

$$P(y_i|\Theta) = \sum_{k=1}^{N_p+1} \Pi_k P(y_i|z_i = k, \Theta). \tag{3}$$

z_i is the latent variable for each y_i: $z_i = k$ means that y_i is generated by the mixture component associated with patch k. $\Pi_k = P(z_i = k|y_i, \Theta)$ represents the probability that patch k explains observation y_i. For each patch, when the closest vertex v_i^k with a compatible normal vector exists, the likelihood that y_i is generated by the k-th component is modeled as a multivariate Gaussian with a mean located at the position $\mathbf{x}_{v_i^k}$ of v_i^k and isotropic variance; otherwise the likelihood is a negligible number ϵ:

$$P(y_i|z_i = k, \Theta) = \begin{cases} \mathcal{N}(\mathbf{y}_i|\mathbf{x}_{v_i^k}, \sigma^2) & \text{if } v_i^k \text{ exists} \\ \epsilon & \text{otherwise.} \end{cases} \tag{4}$$

Solving Eq. 2 yields both the pose and the shape. In [6], the tracking over a complete sequence is achieved by deforming the model on a frame-by-frame basis. That is, using $(\Theta^{t-1}, \mathcal{J}^{t-1})$ as the initialization to solve Eq. 2 at frame t. To increase the robustness of this framework with respect to missing data and outliers better, we propose two methods that improve the deformation prior $P(\Theta)$ (Sec. 4) and the likelihood $P(\mathcal{Y}|\Theta)$ (Sec. 5), respectively.

4. Multiple keyframe tracking framework

The rigidity energy $E_r(\Theta)$ enforces neighboring patches to keep the original local configurations they have on the reference model, usually the shape at a given time t (*e.g.* $t = 0$). However, such local configurations do not always match with the current frame. This is particularly critical with missing data since patches without close observations tend to keep a *possibly* wrong prior reference configuration, as illustrated in Fig. 3.

Algorithm 1 Keyframe-based human motion tracking

1: $\mathcal{F} \leftarrow \{0\}$, $\Psi \leftarrow \{(\Theta_0^0, \mathcal{J}^0)\}$
2: Overall shape parameters $\Theta_0^t \leftarrow \{(\mathbf{I}, \mathbf{c}_k^0)\}_{k=1:N_p}$.
3: **for** t in timeFrames **do**
4: Choose the reference model f_{ref} based on $\mathcal{Y}^t$.
5: $\Theta_{f_{\text{ref}}}^{t-1} \leftarrow \Theta_0^{t-1} * (\Theta_0^{f_{\text{ref}}})^{-1}$
6: With $(\Theta_{f_{\text{ref}}}^{t-1}, \mathcal{J}^{t-1})$ as initialization, solve Eq. 2 to obtain $(\Theta_{f_{\text{ref}}}^{t}, \mathcal{J}^{t})$.
7: $\Theta_0^t \leftarrow \Theta_{f_{\text{ref}}}^t * \Theta_0^{f_{\text{ref}}}$
8: **if** new keyframe detected **then**
9: Update $\mathcal{F}$ and Ψ.
10: **end if**
11: **end for**

Therefore, to effectively handle missing data, we introduce a framework that exploits multiple reference models or keyframes. While already used in image tracking [20], keyframes have not yet, as far as we know, been applied to 3D human motion tracking problems. Multiple keyframes correspond to different instances of a shape that better represent the shape variability than a single pose at a given frame t. Our framework learns online keyframes and, at each frame, selects the best one as the reference model to be fitted to the observations. Let $\mathcal{F} = \{f_m\}_{m=1:n_f}$ denote the keyframe pool where f_m is the frame index and n_f is the total keyframe number, and let $\Psi = \{(\Theta_0^{f_m}, \mathcal{J}^{f_m})\}_{m=1:n_f}$ denote the corresponding parameter set. Our keyframe-based tracking method is summarized in Alg. 1, where Θ_0^f corresponds to the accumulated rotation and translation from $t = 0$ to $t = f$, and $(\Theta_0^f)^{-1}$ represents the inverse transformation. Allowing for different reference models enables different prior knowledge to be taken into account in the rigidity energy $E_r(\Theta)$. Two crucial steps in Alg. 1 are:

- *Line 4*: how to select the best reference model from the keyframe pool?
- *Line 8*: when to add a new keyframe in the pool?

We tackle the former issue with shape dissimilarity and the latter with key pose detection. More details are elaborated in the following two subsections.

4.1. Keyframe detection

We first explain how new keyframes are added (*i.e.* Line 8 in Alg. 1). The essence of the multiple keyframe strategy lies in its ability to identify and record, during tracking, new local patch configurations. When the observed shape takes a pose at t that is very different from the reference pose, it is worth considering adding a new keyframe corresponding to frame t. Such analysis can be performed offline if knowledge on the shape poses is available prior to tracking. However, we consider here the more generic situation with little prior knowledge and where keyframes are detected online during tracking.

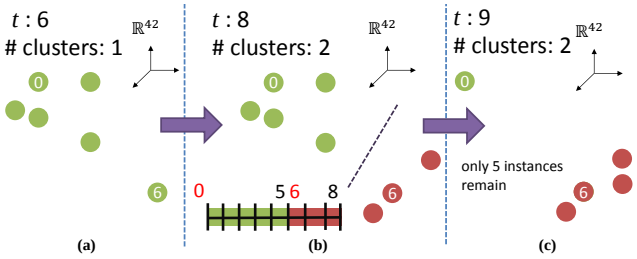


Figure 4. Illustration of Algorithm 2. (a): time frame $t = 6$, $\mathcal{F} = \{0\}$, and $f_{\text{last}} = 0$. (b): $t = 8$, $\mathcal{F}$ becomes $\{0, 6\}$, and f_{last} becomes 6. (c): $t = 9$, $\mathcal{F} = \{0, 6\}$, and $f_{\text{last}} = 6$. Only 5 instances $(9 - 6 + 2)$ are left for mean-shift.

Pose descriptor. In order to incrementally identify distinctive poses, the previously-obtained skeleton poses $\mathcal{J}$ are explored. For each new pose $\mathcal{J}^t$, the pelvis of the skeleton is aligned to the global origin. The 3D coordinates of the remaining 14 joints are then concatenated to form a 42-dimensional human body pose vector $\mathbf{v}$. Aligning the pelvis to the origin cancels the global position and similar poses at different locations get similar representations. Here the skeleton is not rotated into a canonical direction and $\mathbf{v}$ still encodes orientation, which may not be a desirable attribute. We address this issue later. Similar poses yield $\mathbf{v}$s that are close in $\mathbb{R}^{42}$ whereas different poses correspond to $\mathbf{v}$s that are distant. Using this descriptor, we cast the key pose detection as a clustering problem in $\mathbb{R}^{42}$. In this scenario, the number of clusters is supposed to be the output of the clustering technique and not a prior knowledge. Hence mean-shift clustering naturally appears as a well adapted solution.

Mean-shift based key pose detector. Assume n_f keyframes are already identified. Intuitively, there should be also n_f clusters of poses. When a new pose vector $\mathbf{v}$ is obtained, mean-shift is performed and returns a number of clusters n_c. If $n_c = n_f$, the number of clusters has not changed and we proceed to the next frame. If $n_c = n_f + 1$, it means that the shape pose has changed enough to justify a new cluster of poses. In general, poses in the same cluster also distribute closely in the time domain. Hence, starting from the current frame and going backward, the transition frame $\tilde{t}$ where the new cluster starts is determined (*e.g.* $\tilde{t} = 6$ in Fig. 4(b)) and added as a new keyframe to $\mathcal{F}$; its pose $(\Theta_0^{\tilde{t}}, \mathcal{J}^{\tilde{t}})$ being added to Ψ. Fig. 4 illustrates this principle, where f_{last} is the last element included in the keyframe set. The algorithm is sketched in Alg. 2, where Line 3 to 9 correspond to Line 8 to 10 in Alg. 1.

Note that, in general, $\tilde{t} \neq t$ but $\tilde{t} < t$, which means that new keyframes are created online with some delay. For ex-

Algorithm 2 Mean-shift-based key pose detector

1: Compute the bandwidth. Last keyframe $f_{\text{last}} \leftarrow 0$
2: **for** each new incoming pose $\mathcal{J}^t$ **do**
3: De-pelvis the skeleton to obtain $\mathbf{v}^t \in \mathbb{R}^{42}$.
4: De-pelvis all $\mathcal{J}^f$ in Ψ and obtain $\mathbf{V}_{\mathcal{F}} = \{\mathbf{v}^f\}$.
5: Do mean-shift clustering on $\{\mathbf{v}^{f_{\text{last}}+1} \dots \mathbf{v}^t\} \cup \mathbf{V}_{\mathcal{F}}$.
6: **if** the number of clusters = $n_f + 1$ **then**
7: Add transition frame $\tilde{t}$ to $\mathcal{F}$ and $(\Theta_0^{\tilde{t}}, \mathcal{J}^{\tilde{t}})$ to Ψ.
8: $f_{\text{last}} \leftarrow \tilde{t}$
9: **end if**
10: **end for**

ample, in Fig. 4(b), frame at $t = 6$ is detected as a keyframe with a 2 frame delay since in Fig. 4(a), $\mathbf{v}^6$ is still in cluster no. 1. When a new keyframe at $\tilde{t}$ is added, all the pose vectors before $\tilde{t}$, except the existing keyposes, are left out for further clustering (*e.g*. see Fig. 4(c)). This brings two advantages: first, the number of poses considered for clustering, *i.e*. $(t - f_{\text{last}} + n_f)$, is significantly reduced compared to the full set of poses; second, if a pose re-appears during tracking, the new collected $\mathbf{v}^t$ is very likely to be clustered with existing keyframes, avoiding this way the occurrence of duplicated keyframes.

Mean-shift bandwidth. One concern with Alg. 2 is the bandwidth of mean-shift. A small bandwidth leads to many clusters while a large bandwidth gives few clusters. Since the intrinsic scale of the pose variation varies among sequences, an automatic way to determine this bandwidth is desirable. We achieve this using virtual pose vectors. Specifically, the de-pelvised skeleton model at $t = 0$ is rotated for 360° with steps of 10°, creating $N_s = 36$ virtual pose vectors accordingly. Although they map to different points in $\mathbb{R}^{42}$, it is reasonable to cluster them together since they actually correspond to a single pose. We thus compute all $\binom{N_s}{2}$ pairwise distances and set the bandwidth as the half of their maximum. This way we ensure that they converge to the same mode with mean-shift. Recall that when a pose vector $\mathbf{v}$ is built from the estimated skeleton $\mathcal{J}$, only the position is canceled but not the orientation. Using the above bandwidth, we expect pose vectors that differ only by a rotation to be clustered together hence canceling the rotation as well. More analysis on the influence of the bandwidth are presented in the experiment section.

4.2. Choosing the best keyframe

Given a new set of observations $\mathcal{Y}$, the problem is now to determine the best keypose in $\mathcal{F}$ to be matched to these observations (Line 4 in Alg. 1). When the shape associated to such a keypose is close to the observed shape $\mathcal{Y}$, it simplifies the estimation of the shape parameters and reduces the chances to fall in local minima. Therefore, we apply a shape similarity criterion to select the best keypose. This criterion uses shape histograms [1] to describe 3D shapes and the L^2 distance between normalized histograms as the dissimilarity measure [13]. The keyframe that presents the smallest dissimilarity with $\mathcal{Y}$ is chosen as the reference frame.

5. Patch-based outlier modeling

Besides missing data, sometimes point clouds contain false segmented foreground as the chair in Fig. 1 bottom row. In order to be robust to such outliers, care must be taken when designing the likelihood function $P(\mathcal{Y}|\Theta)$. Note that the association of an observation y_i to a patch, *i.e*. Eq. 4, applies only for $z_i = k \in [1, N_p]$ and that $z_i = N_p + 1$ is a special case introduced to model the outliers y_i that are not explained by any patch. However, there is no physical outlier patch in the model to be associated to. In [6], Cagniart *et al*. use a uniform distribution to model $P(y_i|z_i = N_p + 1, \Theta)$ which basically assumes a certain proportion of the observations to be outliers and requires therefore some ad-hoc knowledge. Here we present a patch-based outlier modeling technique that takes into account spatial information and is based on the fact that the observations at frame t usually lie in the vicinity of the estimated surface at frame $t - 1$.

Before modeling the general outlier event $z_i = N_p + 1$, we first consider the outlier event just for patch k, denoted as O_k. The likelihood between y_i and O_k can be interpreted as how "bad" y_i is explained by patch k. Since Eq. 4 expresses how well patch k explains y_i under shape parameter Θ, we define the likelihood $P(y_i|O_k, \Theta)$ as:

$$P(y_i|O_k, \Theta) \equiv 1 - P(y_i|z_i = k, \Theta) = \begin{cases} 1 - \mathcal{N}(\mathbf{y}_i|\mathbf{x}_{v_i^k}, \sigma^2) & \text{if } v_i^k \text{ exists} \\ 1 - \epsilon & \text{otherwise.} \end{cases} \quad (5)$$

Eq. 5 also expresses, given shape parameter Θ and from the point of view of patch k, how likely y_i is to be an outlier. Checking over all patches how poorly they explain y_i, and assuming independence between O_k, we can approximate the overall outlier likelihood as:

$$P(y_i|z_i = N_p + 1, \Theta) \approx \prod_{k=1}^{N_p} P(y_i|O_k, \Theta). \quad (6)$$

From Eq. 6, we see that observations that are well explained by patches can not be outliers, where [6] considers equal chances for each observation to be an outlier. Fig. 5 illustrates this strategy. Fig. 5(a) and (b) shows the reference model represented in patches and body parts respectively. In [12], Huang *et al*. train a linear SVM on Fig. 5(b) obtained at $t - 1$ to classify $\mathcal{Y}^t$ into different rigid body parts, as shown in Fig. 5(c). Chair observations are classified as

Sequence	Views	Frames	Outlier	Mis. data	Err. metric	Keyframe pool	Bandwidth	Compared approaches
Skirt [11]	8	720	-	$\surd$	A	$0, 95, 198$	0.31	[6], [11] (1^{st} stage)
Dance [11]	8	574	-	$\surd$	A	$0, 201$	0.41	[6], [11] (1^{st} stage)
Basketball [6]	8	1330	dynamic	$\surd$	-	$0, 29$	0.41	-
Fighting [15]	12	500	dynamic	$\surd$	B	$0, 20, 59, 74$	330.25	-
WalkChair	9	148	static	$\surd$	A & C	$0, 32, 54$	0.50	[6], [12], [17] + [18]
HammerTable	9	93	static	-	A & C	$0, 21$	0.44	[6], [12], [17] + [18]
SideSit	9	97	static	-	C	$0, 21$	0.50	[6], [12], [17] + [18]

Table 1. Sequences used for evaluation. We apply three different error measures, depending on the provided ground truth. A: silhouette overlap error. B: distances in $\mathbb{R}^3$ between makers and associated vertices. C: distances in pixels with annotated joint positions.

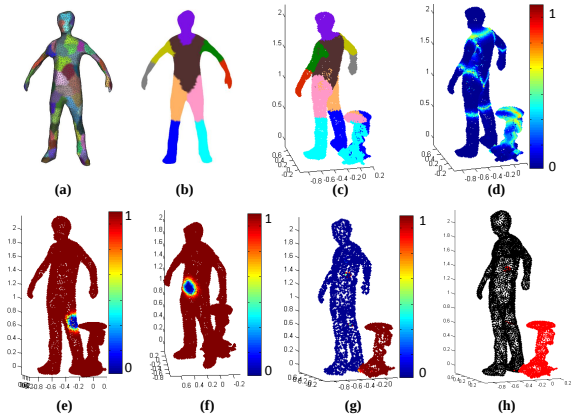


Figure 5. Comparison of outlier rejection in [12] (c-d) and ours (g-h). (a-b): reference surface colored in patches and body parts, respectively. (c): point cloud classified into body parts using the SVM trained on (b) from the previous frame, as suggested in [12]. (d): corresponding outlier likelihood from SVM. (e-f): two examples of Eq. 5. (g): point cloud colored based on Eq. 6. (h): points with outlier likelihood higher than 0.5 are colored in red.

body parts because of the linear assumption of SVM. If we consider the SVM output for the outlier likelihood, as in Fig. 5(d), we cannot distinguish between chair and human observations. However, our patch-based outlier modeling is able to assign high values to chairs, as in Fig. 5(g), and to identify them by simple thresholding.

Optimization We follow the optimization framework in [6, 12] which alternates between estimating associations, *i.e*. Eq. 3, and solving for the model parameters, *i.e*. Eq. 2. These two steps correspond to the E-step and the M-step in the Expectation-Maximization framework [3]. The advantage of our outlier strategy is that it easily integrates into this method. In practice, outliers are not removed once and for all with hard thresholding, but Eq. 6 is substituted in Eq. 3. This means that outliers are estimated during the EM optimization, and that there is no need to add any other sophisticated learning-based method to improve outlier rejection.

6. Experiment results

The method was evaluated on 4 publicly available sequences as well as on 3 new sequences: *WalkChair*, *HammerTable*, and *SideSit* that contain static occlusions. These sequences were recorded with 9 cameras at 1000×1000 resolution. The occlusion objects are considered as foreground by the background subtraction and they remain in silhouettes and hence appear in the resulting point cloud observations, as shown in Fig. 8(c) and (d). We manually annotate the joint positions in 5 cameras to evaluate the poses of the skeleton. Due to the lack of realistic dynamic 3D surface ground truth, we use silhouette overlap error to evaluate the shape parameter estimation. If the occlusion objects are separated from the human body in silhouettes, we manually remove them and consider only the human parts as ground truth. These sequences are available at the *4D Repository*[1]. To draw fair comparisons with other approaches, we do not refine the surfaces with silhouettes after tracking. These 7 sequences serve different purposes in the experiments and we summarize them in Table 1. Results are analyzed with respect to missing data and outliers, both qualitatively and quantitatively. In all the presented experiments, both the multiple keyframes strategy and the outlier rejection mechanism were used.

	Cagniart [6]	Gall [11]	Prev.	ours
Skirt	7283	6900	7466	**6715**
Dance	7881	7600	fail	**6940**

Table 2. Average silhouette overlap error with different approaches. Image resolution: 1004×1004. Note that comparisons with [11] concern only their first stage results. Prev. are the results obtained when using the previous frame as the reference model. See *Supplementary Material* for more discussion.

6.1. Robustness to missing data

Skirt and *Dance* demonstrate the effectiveness of the multiple keyframe strategy. The average silhouette overlap error for these sequences is shown in Table 2. In [11], Gall *et al*. refine the shape using silhouettes as a second stage of their method. Such refinement could fail if occlusion objects are close to the subject and appear in the silhouettes (*e.g*. Fig. 1 bottom row). We thus compare to their first stage results only. In *WalkChair*, missing data can be observed when the arms are too close to the torso. With the shape at $t = 0$ as the reference model result in Fig. 6(a) are

[1] http://4drepository.inrialpes.fr/

Figure 6. Results of frame $t = 119$ in *WalkChair*, with and without multiple keyframe strategy. Black dots are the observed point cloud where right arm gets merged into the torso. (a): estimated shape using surface at $t = 0$. (b): estimated shape at $t = 32$. (c): estimated shape using (b) as the reference surface.

obtained. The right arm stays in its configuration at $t = 0$ since not enough observations support the patches on the arm. Using another keyframe, as in Fig. 6(b), as the reference yields better result as shown in Fig. 6(c).

The influence of the mean-shift bandwidth. We report here on the influence of the mean-shift bandwidth on key pose detection and on the final tracking results. Tests were conducted on *Skirt*, *HammerTable*, and *Fighting* with varying bandwidths. Results on the keyframe numbers as well as on the corresponding errors are depicted in Fig. 7(a-c). The first two sequences were chosen for the repeating actions and the third one for its different numerical scale, see Fig.7(d). In general, small bandwidths lead to more keyframes (green curves) and it appears that the errors (blue curves) also decline as the bandwidth decreases. However, small bandwidths have higher chances to identify tracking failures as keyframes and therefore accumulate errors. This explains why the error slightly rises when the bandwidth gets really small. We noticed that due to the different numerical scales, resulting from different recording setups, there is no fixed bandwidth that guarantees the best performance over the three sequences. This indicates that manually fixing the bandwidth is difficult. However, our strategy considers the numerical scale of the sequence and adjusts the bandwidth accordingly. This provides optimal or close to optimal performance (red dots). Note also that in *Skirt*, the subject raises up both arms and slowly rotates herself for a while, which leads to many similar poses only differing in orientations. As a result of our approach to cancel rotations, and the way we perform mean-shift with $\mathbf{V}_{\mathcal{F}}$, duplicate key poses do not occur. Keyframe pools of all testing sequences are shown in the *Supplementary Material*.

6.2. Robustness to outliers

Static outliers. *WalkChair*, *HammerTable* and *SideSit* were used to demonstrate the robustness to outliers. Our approach was compared with [6], [12] and the tracking framework proposed by Straka *et al.* ([17]+ [18]). The silhouette overlap error as well as the discrepancies between the

	WlkChr.	*HmmrTbl.*
Cagniart [6]	18482	fail
Huang [12]	18063	fail
Straka [17, 18]	12219	17285
ours	**6803**	**3593**

Table 3. Average silhouette overlap error of *WalkChair* (*WlkChr.*), and *HammerTable* (*HmmrTbl.*) from different approaches.

	WlkChr.	*HmmrTbl.*	*SideSit*
Huang [12]	24.6 ± 10.7	fail	75 ± 40
Straka [17, 18]	20.6 ± 22.0	64.2 ± 53.9	84.4 ± 59.3
ours	$\mathbf{15.9 \pm 6.3}$	$\mathbf{10.1 \pm 3.0}$	$\mathbf{19.3 \pm 7.9}$

Table 4. Average joint 2D re-projection error in pixels of *WalkChair* (*WlkChr.*), *HammerTable* (*HmmrTbl.*), and *SideSit*.

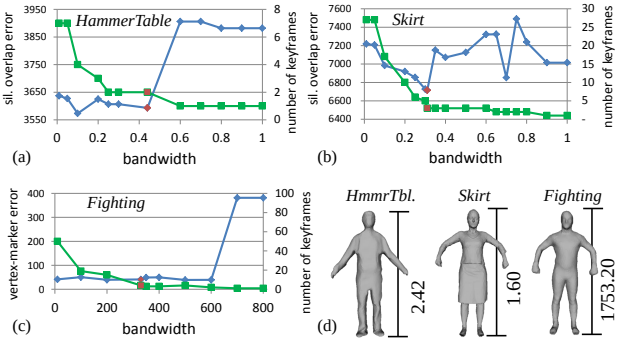


Figure 7. (a-c): performance and the number of keyframes v.s. bandwidth. (d): the heights of the subjects in 3 sequences. Blue curves are the corresponding error (left y-axis) while green curves represent the number of keyframes (right y-axis). (a): *HammerTable*. (b): *Skirt*. (c): *Fighting*. See text for more details.

projected skeletons and the manually annotated joint positions in 5 cameras were measured. As reported in Table 3 and Table 4, our method attains consistently lowest errors in both pose and shape estimations. It is worth noting that in *HammerTable*, around 41% of the observations are not from the human subject, but we still get decent results (see Fig. 8(c) and (d)). We notice also that when the occlusions are closely touching the subject, it confuses the method in [17] that deforms the skeleton model according to the observed skeletal graph in the point cloud, and thus the shape adaptation [18] cannot improve significantly the results. Please refer to the *Supplementary Material* and the accompanying video for more comparisons.

Dynamic outliers. Due to the lack of public datasets with dynamic outliers, we evaluate our approach with two multi-subject sequences where we track only one subject and consider the others as outlier observations. For *Basketball*, the human subject is tracked against the ball observations. For *Fighting*, the subject with markers is tracked and the observations from the other subject are considered as outliers. The results are shown in Fig. 8(a) and (b). First we con-

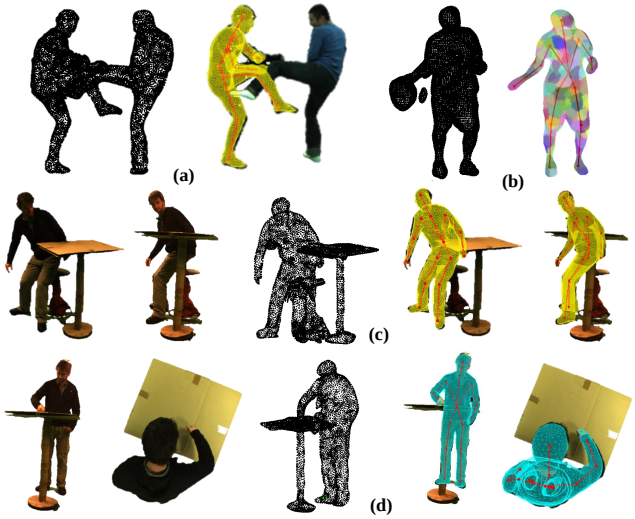


Figure 8. Results of (a) *Fighting*, (b) *Basketball*, (c) *SideSit*, and (d) *HammerTable*. Black dots represent the observed point clouds.

firm that without the outlier rejection strategy presented in Sec. 5, the method in [6] fails with this experimental setting, which is expected since it also fails to track against static outliers as in Table 3. Following the metric in [15], our approach attains 40.95 *mm* of average vertex position error with a standard deviation of 15.34 *mm* over 500 frames. Although the corresponding error in [15] is 29.61 *mm* and 25.50 *mm*, respectively, we would like to point out that their task is different from ours. Our objective is to robustly track in noisy environment whereas [15] simultaneously track two surfaces. Every observations is, in this case, associated to a patch or a vertex, and thus outliers are not to be considered. We observe anyway that, despite the presence of strong dynamic outliers (*i.e*. the ball and the second subject), our approach still provide reasonable results.

Acknowledgment. We would like to thank Matthias Straka from TU Graz for providing experimental results on their methods as well as valuable discussions.

7. Conclusion

We present an approach that captures human performances from multi-view video without markers. Considering realistic cases, we propose a multiple-keyframe-based tracking framework that uses mean-shift clustering to update a keyframe set online. A patch-based outlier modeling method is also presented to identify outliers more efficiently and effectively. Combining these two techniques into a surface deformation framework increases the robustness and enables the estimation of human shape and poses against missing data and outliers. The reliability of the proposed method is confirmed by the experiments on various public sequences as well as newly recorded sequences. Future directions include alleviating the requirement for background subtraction by considering photometric information.

References

[1] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *Advances in Spatial Databases*, 1999.

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

[3] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. Springer, 2006.

[4] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global non-rigid alignment of surface sequences. In *IJCV*, 2013.

[5] C. Cagniart, E. Boyer, and S. Ilic. Free-form mesh tracking: a patch-based approach. In *CVPR*, 2010.

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

[7] 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.

[8] E. Duveau, S. Courtemanche, L. Reveret, and E. Boyer. Cage-based motion recovery using manifold learning. In *3DimPVT*, 2012.

[9] J.-S. Franco and E. Boyer. Exact polyhedral visual hulls. In *BMVC*, volume 1, 2003.

[10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *PAMI*, 2010.

[11] 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*, 2009.

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

[13] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. In *IJCV*, 2010.

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

[15] Y. Liu, C. Stoll, J. Gall, H. P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR*, 2011.

[16] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Eurographics*, 2004.

[17] M. Straka, S. Hauswiesner, M. Ruether, and H. Bischof. Skeletal graph based human pose estimation in real-time. In *BMVC*, 2011.

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

[19] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *CVPR*, 2008.

[20] L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3d tracking in real-time. In *CVPR*, 2003.

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