Toward User-specific Tracking by Detection of Human Shapes in Multi-Cameras

Chun-Hao Huang[1], Edmond Boyer[2], Bibiana do Canto Angonese[1], Nassir Navab[1], Slobodan Ilic[1,3]

[1] Technische Universität München, [2] LJK-INRIA Grenoble Rhône-Alpes, [3] Siemens AG
{huangc,slobodan.ilic,bibiana.canto,navab}@in.tum.de, edmond.boyer@inria.fr

Abstract

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 find their common ground with the Iterative-Closest-Point (ICP) algorithm, which facilitates the association step with local distance considerations. 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. It allows for large deformations and prevents tracking errors from accumulating. The approach is successfully integrated to a surface tracking framework that recovers human shapes and poses jointly. When combined with ICP, this discriminative association proves to yield better accuracy in registration, more stability when tracking over time, and faster convergence. Evaluations on existing datasets demonstrate the benefits with respect to the state-of-the-art.

1. Introduction

Visual shape tracking is the process of recovering temporal evolutions of a template shape using visual information, such as image silhouettes 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, and the objective is to recover the skeletal poses [29], surface shapes [6], or both simultaneously [15].

Most of these model-based methods [6, 12, 15, 18, 29] can be viewed as extensions of Iterative-Closest-Point (ICP) framework [5], 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.*, the slow convergence. Moreover, when large deformations or outliers occur, discovering associations between data and models is particularly difficult. Unreliable associations result in ambiguous situations that yield erroneous numerical solutions and, consequently, break the tracking process.

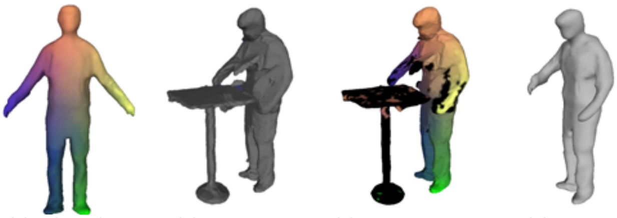


Figure 1. Given a reference surface (a), our method discovers reliable data-model correspondences by random forests, color-coded in (c). Instead of tracking, this strategy detects user-specific shapes in a frame-wise manner, resulting in more sustainability.

In contrast, *discriminative* approaches that 'detect' rather than track models have shown better robustness over the past decade, for instance, in human pose estimation with Kinect [22, 27]. These approaches operate frame-independently, and are generally drift free. In this paper, we explore this direction in order to get robust observation-model associations, regardless of the results from previous frames. We further present a discriminative 'tracking-by-detection' human mocap framework, as in Fig. 1. Inspired by Taylor *et al.* [27], we apply regression forests to improve the associations. Shape geometries are characterized by volumetric representations, and are fed into user-specific forests to predict correspondences in *one shot*. Contrary to generative methods, this prediction does not require close initializations from a nearby frame. In addition, it allows a single model to be used as a reference surface for several different sequences, again even if large deformations or outliers exist. We combine this strategy with a generative tracking approach that takes our one-shot associations as input. Experiments demonstrate that this *hybrid* discriminative-generative framework leads to better, or comparable results

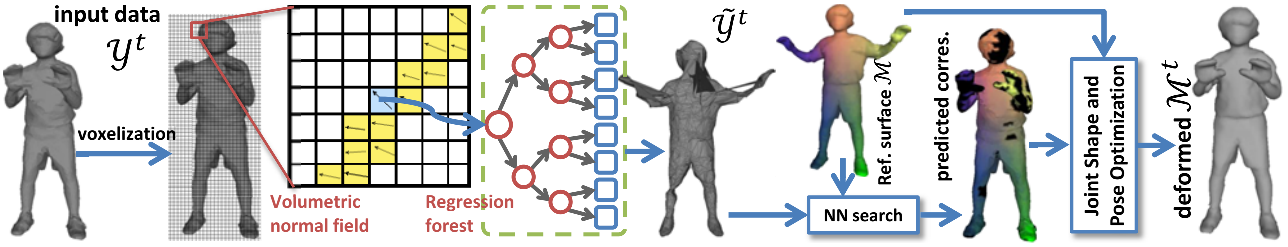


Figure 2. Pipeline of our framework. Correspondences are visualized in the same color. Black means no correspondence for that data point.

than purely generative approaches, *e.g*. [6,8], reducing error accumulations and hence increasing the stability.

The contribution of this work is two-fold: (1) a one-shot correspondence inference with complete 3D input (rather than 2.5D as in [22,27]), which leads to (2) a hybrid subject-specific human pose and shape capture method that relies little on former results (unlike other ICP-like methods [6, 12, 15, 18]), holding the ability to recover from drifting.

2. Related work

Among the vast literature on human pose estimation [19], we focus on top-down approaches that assume a 3D model and deform it according to input data, either directly with pixels as in [12,18,25], or with 3D points as in [6,8,15]. These methods typically decompose into two main steps: (i) data association, where observations are associated to the model, and (ii) deformation estimation, where deformation parameters are estimated given the associations. Our primary objective in this paper is to improve the first part. Existing approaches for this issue are discussed below.

Generative approaches. Methods in this category do not require any training. They follow the association strategy in ICP while extending the motion model to more general deformations than the one in the original method [5]. Associations are addressed by searching for closest points, with various distance measures such as point-to-point [6], point-to-plane [7], or Mahalanobis distances [25]. This strategy heavily relies on the fact that observations in consecutive frames are in vicinity. Klaudiny *et al.* [17] generalize the idea from the previous frame to a certain frame in the considered sequences, finding the best non-sequential order to track, but the proximity assumption remains. Gall *et al.* [12] establish 3D-2D correspondences by considering both texture in images, and contours in silhouettes. Later, Liu *et al.* [18] include image segmentation information in order to differentiate multiple interacting subjects. These approaches implicitly assume that observations only describe the tracking subjects, which does not necessarily hold in 3D data that often contain fake geometries. Cagniart *et al.* [6] introduce an additional outlier class to reject associations with noisy observations. Data is explained by Gaussian Mixture Models (GMM) in an Expectation-Maximization (EM) manner. Huang *et al.* [15] follow a similar concept, but aggregate the outlier likelihood over every Gaussian component and offer better robustness. 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.

Discriminative approaches. Recently, discriminative approaches have demonstrated their strengths in tracking human poses with depth images [4,27]. Taylor *et al.* [27] propose a single-frame, or so called one-shot strategy, which yields decent dense correspondences without iterative refinements. With the help of regression forests, they map each foreground pixel to a weighted point in 3D, and thereby search the closest point within a predefined surface. Later, Pons-Moll *et al.* [20] train forests with a new objective in metric space, and couple the one-shot strategy with ICP. In the case of full 3D, Kanaujia *et al.* [16] use shape context histograms as descriptors, segment visual hulls into body parts with a pre-trained Support Vector Machine (SVM), and build the skeletons with the method similar to [22]. Starka *et al.* [23] formulate the matching problem as the inference of Markov random field (MRF). Rodola *et al.* [21] apply forests to learn the parameters of wave kernel signatures [3] during training, and facilitate dense matching between two meshes.

To avoid computation overhead, we develop volumetric normal fields to describe meshes in a discretized volume Ω_3, and extend the comparison features from 2.5D [22, 27] to full 3D data. Our method can be viewed as a tracking-by-detection approach for human shape tracking.

3. Preliminaries and method overview

We first state the problem and briefly outline our method. A 3D reference surface is denoted as $\mathcal{M} = (\mathbf{M}, \mathcal{T}_{\mathcal{M}})$, where $\mathbf{M} = \{\mathbf{x}_v\}_{v=1}^{N_v} \subset \mathbb{R}^3$ are the locations of vertices v, and $\mathcal{T}_{\mathcal{M}}$ defines the triangles. Evolving $\mathcal{M}$ typically amounts to parameterizing $\mathbf{M}$ as a function of deformation parameters Θ, namely, $\mathbf{M}(\Theta)$. Unlike [12, 18, 27], our interest lies in recovering not only poses but also shapes of the subjects. We adopt a surface deformation framework that groups vertices into N_p patches [6], and assign each of them a rigid body motion θ. Thus, Θ is the collection of all θ, $\Theta = \{\theta_k\}_{k=1}^{N_p}$, encoding the global shape of the surface. We refer the readers to [6] for detailed explanations.

Given an observed visual hull $\mathcal{Y}^t = (\mathbf{Y}^t, \mathcal{T}_{\mathcal{Y}}^t)$[1], where $\mathbf{Y}^t = \{\mathbf{y}_i\}_{i=1}^{N_y} \subset \mathbb{R}^3$, the goal is to determine the optimal $\hat{\Theta}^t$ such that $\mathbf{M}^t = \mathbf{M}(\hat{\Theta}^t)$ resembles $\mathbf{Y}^t$ as much as possible. It typically boils down to two sub-problems:

1. finding correspondence pairs $\mathcal{C} = \{(i, v)\}$ between the vertex sets of $\mathcal{Y}$ and the vertex sets of $\mathcal{M}$[2], and

2. minimizing an energy E that describes the discrepancies between vertices in $\mathcal{C}$: $\hat{\Theta} = \arg\min_{\Theta} E(\Theta; \mathcal{C})$.

Standard ICP-based approaches [6, 8] alternate between these two steps, refining $\mathcal{C}^t$ and Θ^t iteratively. They require $\mathbf{M}^{t-1}$ to be close to $\mathbf{Y}^t$, and is usually slow to converge.

We develop a different strategy that warps the input data $\mathcal{Y}$ to the reference mesh $\mathcal{M}$, denoted as $\tilde{\mathcal{Y}} = (\tilde{\mathbf{Y}}, \mathcal{T}_{\mathcal{Y}})$, and visualized as a triangular mesh in Fig. 2. If the warping is perfect, this mesh will look clean and resemble $\mathcal{M}$ as much as possible. Incorrect mapping, on the other hand, can be told from huge edges. Vertex positions $\tilde{\mathbf{Y}}$ represent the locations of potential matches between $\mathcal{Y}$ and $\mathcal{M}$. Thus, $\mathcal{C}$ can be built directly by doing nearest neighbor search between $\tilde{\mathbf{Y}}$ and $\mathbf{M}$ just once, as illustrated in Fig. 2.

Specifically, we consider this $\mathbb{R}^3 \to \mathbb{R}^3$ mapping as a composite one: $\mathbb{R}^3 \to \Omega_3 \to \mathbb{R}^3$. The former mapping is voxelization (Sec. 4.1), while the latter is regression (Sec. 5). A forest is trained with many voxelized meshes off-line (Sec. 5.1). During runtime, $\mathbf{y}_i$ is first mapped to a voxel $\mathbf{v}_i$, and then regressed to a 3D point $\tilde{\mathbf{y}}_i \in \tilde{\mathbf{Y}}$. Given the properly estimated $\mathcal{C}$, poses and shapes are then recovered as in [14] (Sec. 5.2). Fig. 2 shows our pipeline.

4. Normal volume and features

Before entering the forest, we cast our data into a volumetric field $\mathbf{N} : \Omega_3 \subset \mathbb{R}^3 \to \mathbb{R}^3$, and design a set of features to describe surface geometries in volumes.

4.1. Volumetric normal field (VNF)

Voxelizing a mesh in general consists of two parts: (1) finding voxel positions for every vertex, 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 $\mathbf{v} : \mathbb{R}^3 \to \Omega_3$. The size of the volume is large enough to include all possible pose variations, and its center is aligned with the mean 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 in graphic community [1].

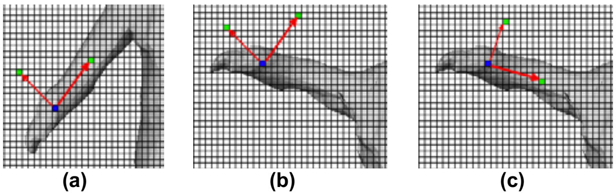


Figure 3. (a) original offset pair ψ. (b) $\eta = 0$ results in ψ without re-orientation, *i.e.* $\mathbf{R} = \mathbf{I}$. (c) $\eta = 1$. ψ is re-orientated by a rotation matrix $\mathbf{R} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ characterized by a LCF.

After determining voxels occupied by the surface, referred to as $\mathbf{v}_{\text{suf}}$, we further identify voxels located inside and outside the surface, denoted respectively as $\mathbf{v}_{\text{in}}$ and $\mathbf{v}_{\text{out}}$. Together they define a truncated normal field:

$$\mathbf{N}(\mathbf{v}) = \begin{cases} +\boldsymbol{\epsilon} & \text{if } \mathbf{v}_{\text{out}} \\ \mathbf{n} \in [-1, +1]^3, \ \|\mathbf{n}\| = 1 & \text{if } \mathbf{v}_{\text{suf}} \\ -\boldsymbol{\epsilon} & \text{if } \mathbf{v}_{\text{in}}. \end{cases} \quad (1)$$

$\mathbf{n}$ is the average surface normal from the containing triangles, whereas $\boldsymbol{\epsilon}$ is an indicator vector ($[2, 2, 2]$ in our experiments). Given a mesh, our VNF naturally encodes both spatial occupancies (surface locations) and first-order derivatives (normals). It shares a similar spirit with implicit surface representations, *e.g.* level-set, but does not require any expensive distance transform computations.

4.2. Volumetric features

Here, we present the features $\mathbf{f}(\mathbf{v}_{\text{suf}})$ for describing the above VNF, which are later used to train the forests. Conventionally, only one feature dimension κ is selected to separate data at each branch node of the forest [9]. One does not have to prepare the whole high-dimensional feature vector for predictions, because only a few dimensions are needed. The calculation of $\mathbf{f}$ is suggested to be dimensionally independent. We therefore avoid descriptors that requires normalization, like MeshHOG [30], or SHOT [28], and resort to comparison features used in [10, 22].

As depictcted in Fig. 3, for each surface voxel $\mathbf{v}_{\text{suf}}$ (blue), we shoot two offsets (red vectors) $\psi = (\mathbf{o}_1, \mathbf{o}_2) \in \Omega_3 \times \Omega_3$, and reach two neighboring voxels (green). To describe the local geometry, we concatenate the dot product of their normals, and the difference of VNF within local cuboids to form the feature vector $\mathbf{f}$. Such a computation is repeated for multiple offsets at $\mathbf{v}_{\text{suf}}$. By definition, each dimension of $\mathbf{f}$ can be evaluated independently. See *Supplementary Material* for more details.

Note that $\mathbf{f}$ is a function of $\mathbf{v}_{\text{suf}}$, but takes an offset pair ψ, a binary variable η (whether using *Local Coordinate Frame*

[1] Both the reference model and input data are described as triangular meshes. Although the terms mesh and surface are used interchangeably in the text, we refer only to input data as visual hulls.

[2] Whenever it is clear from the context, we will drop the time dependent variable t, in order to keep notations uncluttered.

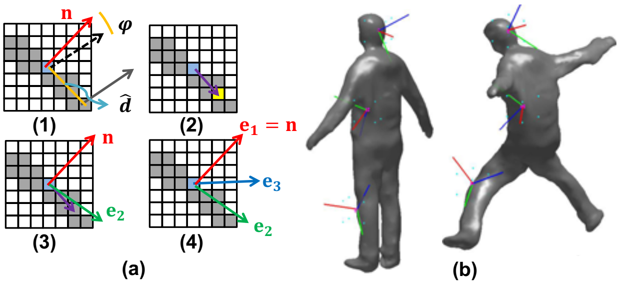


Figure 4. (a) The way we compute LCF. We search neighboring surface voxels to get a reference vector (purple arrow); see text for details. (b) Our method leads to LCFs covariant to pose variations.

(LCF) or not), and a rotational matrix $\mathbf{R} \in SO(3)$ (the orientation of LCF) as parameters, denoted as $\mathbf{f}(\mathbf{v}_{\text{suf}}; \mathbf{R}^{\eta}(\psi))$. η determines the alignment of the offset ψ w.r.t 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. 3(b-c). Without re-orientations, ψ might land on different types of voxel pairs (c.f. Fig. 3(a-b)), and hence cause different feature responses, 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 when training the forest, while the rotational matrix $\mathbf{R}$ is characterized by a local coordinate frame obtained as follows.

Local coordinate frame. Constructing a LCF requires 3 orthonormal vectors: $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$. They usually come from doing PCA on a local region, which is impractical to repeat for all surface voxels $\mathbf{v}_{\text{suf}}$. Instead, we apply the following method, as illustrated in Fig. 4(a). The first eigenvector $\mathbf{e}_1$ is always the normal vector (red arrow). $\mathbf{e}_3$ is simply $\mathbf{e}_1 \times \mathbf{e}_2$ if the second eigenvector $\mathbf{e}_2$ is known. First we open a local cuboid around the current voxel (blue). Among neighboring gray surface voxels, we find the one that has least normal changes, and yet located far from the current voxel. In practice, given the notations in Fig. 4(a), we select the one that maximizes $(\hat{d} + \cos\varphi)$, where φ is the angle between normals, and $\hat{d}$ is the normalized distance. Given this reference voxel (yellow), a reference vector (purple) is decomposed into two components. The component perpendicular to the normal is then chosen as $\mathbf{e}_2$ (green). Fig. 4(b) shows that this approach produces LCFs quasi-*covariant* to pose changes, and as a result, quasi-pose-invariant features $\mathbf{f}$.

5. Correspondence inference and tracking

Now we proceed to the second mapping: $\Omega_3 \rightarrow \mathbb{R}^3$, where the surface voxel $\mathbf{v}_{\text{suf}}$ traverses a pre-trained regression forest according to the feature responses, and attains a point in $\mathbb{R}^3$ that lies on the surface embedding defined by the vertices of the reference model $\mathbf{M}$.

5.1. Random forest

A forest is an ensemble of T binary decision trees, each separating data with split functions at branch nodes and storing statistic models at leaf nodes. The training objectives, split functions, and statistical models vary from task to task. We refer readers to [9] for a comprehensive review.

Training. To incorporate abundant training variations, we animate $\mathcal{M}$ to a variety of poses with a method similar to [26]. After voxelizing all the animated meshes, we associate each surface voxel to their locations at rest pose, and obtain a pool of sample-label pairs $\mathcal{S} = \{(\mathbf{v}_{\text{suf}}, \mathbf{x}_v)\}$. Each tree is trained with a random subset of $\mathcal{S}$. Our splitting candidate is $\phi = (\psi, \eta, \kappa, \tau)$ (offset pairs ψ, binary variables η, testing feature channels κ, and thresholds τ). Let $\mathcal{S}_N$ denotes the samples arriving at a certain branch node. The training process is to partition $\mathcal{S}_N$ recursively into two subsets $\mathcal{S}_L$ and $\mathcal{S}_R$, based on randomly generated ϕ:

$$\mathcal{S}_L(\phi) = \{\mathbf{v}_{\text{suf}} \in \mathcal{S}_N | f_\kappa(\mathbf{v}_{\text{suf}}; \mathbf{R}^{\eta}(\psi)) \geq \tau\} \tag{2a}$$

$$\mathcal{S}_R(\phi) = \{\mathbf{v}_{\text{suf}} \in \mathcal{S}_N | f_\kappa(\mathbf{v}_{\text{suf}}; \mathbf{R}^{\eta}(\psi)) < \tau\}. \tag{2b}$$

Here f_κ denotes the κ^{th} dimension of the feature $\mathbf{f}$.

Whether ϕ is a good split or not depends on whether it produces more homogeneous subsets. In supervised learning, this is often measured by information gain:

$$G(\phi) = H(\mathcal{S}_N) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}_i(\phi)|}{|\mathcal{S}_N|} H(\mathcal{S}_i(\phi)), \tag{3}$$

where $H = \sigma^2(\cdot)$ is the entropy, measured by the variance of all $\mathbf{x}_v$ in the sample set as in [13]. The split that maximizes the information gain, $\phi^* = \arg\max_\phi G(\phi)$, is stored for the later prediction use. The tree recursively splits samples and grows until one of the following stopping criteria is true: (1) it reaches the maximum depth, or (2) the number of samples $|\mathcal{S}_N|$ is too small. Once a leaf node is reached, we perform mean-shift clustering to represent the distributions of $\mathbf{x}_v$ 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.

Outliers such as false geometries, or un-removed background often exist in visual hulls, drastically deteriorating tracking results. If their models are available, we also include them in the training process, so that forests can reject them online. In this case, the goodness of a split ϕ should be evaluated in terms of both classification and regression. We follow Fanelli *et al.* [11] and adapt the entropy as:

$$H(\mathcal{S}) = -\sum_c p(c|\mathcal{S}) \log p(c|\mathcal{S}) + (1 - e^{\frac{D}{\alpha}})\sigma^2(\mathcal{S}), \tag{4}$$

where $p(c|\mathcal{S})$ is the class probability of being foreground or background. Eq. 4 is the weighted sum of the aforementioned regression measure σ^2 and the classification entropy measure. The regression part gets increasing emphasis when the current depth D gets larger (*i.e.*, the tree grows deeper), and the steepness is controlled by the parameter α.

Subject coordinate frame. To achieve global rotation invariance, we always rotate meshes into a canonical orientation before the voxelization. In the training phase, since each animated mesh is accompanied with a skeletal pose, we compute two unit-length vectors from the skeleton, and align them to x- and z-axis respectively. The first one is the common perpendicular vector of bone *RShoulder-Torso* and bone *LShoulder-Torso*, while the second one is the sum of them. See *Supplementary Material* for illustrations.

Recall that the volume center is aligned with the means of meshes, which brings translation invariance. Together these two steps characterize a subject coordinate system, simulating the PCA process on a whole mesh. During tracking, we approximate the skeletal pose using results of the previous frame, and repeat the same process.

Prediction. In the prediction phase, a voxel lands on T leaves containing different collections of modes, denoted together as $\mathcal{G} = \{\mathcal{H}_1 ... \mathcal{H}_T\}$. A standard way of aggregation is doing mean-shift and keeping the cluster with largest weight. This is usually done independently for each sample. For instance, in Fig. 5(a), the green vertex aggregates with only green leaves, and black vertices aggregate with their respective black leaves as well. The consequent $\tilde{\mathcal{Y}}$ is however, often noisy, where vertices of same triangles are mapped to locations distant from each other, as in Fig. 5(b).

We utilize the triangles $\mathcal{T}_\mathcal{Y}$ to ameliorate this problem. All $\mathbf{y}_i$ are first mapped to voxels, sent into the forest once, and return in total $N_y \times T$ predictions. For each vertex i, we consider not only its own predictions, but also those from the neighbors $\mathcal{N}_i$, *e.g.* all green and black leaves in Fig. 5(a). We sort the set $\mathcal{G}_i \cup \mathcal{G}_{\mathcal{N}_i}$ descendingly according to their weights ω, and do mean-shift only with the first half of them. The new mode location with highest confidence is the final output $\tilde{\mathbf{y}}_i$. This strategy respects the mesh connectivity and results in more structured forest predictions. Compare Fig. 5(c) to (b) to see the improvements.

Nearest neighbor search. Given the regression results $\tilde{\mathbf{y}}_i$, each vertex i in input gets a closest vertex p in the reference vertex set $\mathcal{V}_\mathcal{M}$, *i.e.* $p = \arg\min_{v \in \mathcal{V}_\mathcal{M}} \|\tilde{\mathbf{y}}_i - \mathbf{x}_v\|_2$. Similar to [6], we reject the searched correspondences if their normals differ from each other too much. Our advantage over [6] is that $\mathbf{M}$ and its normals are fixed throughout tracking. One does not have to re-compute normals online, and other speed up algorithms like kd-tree are also feasible. Each correspondence pair (i, p) is associated to a weight $w_{ip} = \exp(-d^2(i,p)/2l^2)$, where $d(\cdot)$ stands for Euclidean distance, and l is the averaged edge length on $\mathcal{M}$. Since the forest gives relatively good initial guess of the matches. It is much easier to find the right matches in this way.

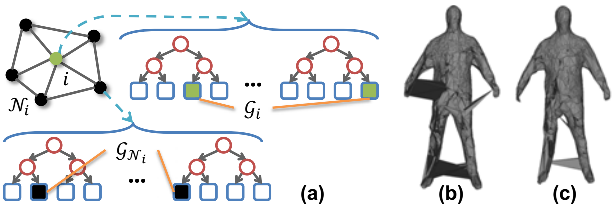


Figure 5. (a) Traditional approach aggregates leaves of each vertex (green) independently, while we also take the 1-ring neighbors (black) into account. (b) Forest output $\tilde{\mathcal{Y}}$ by aggregating each $\mathcal{G}_i$ separately. (c) Forest output $\tilde{\mathcal{Y}}$ by aggregating $\mathcal{G}_i \cup \mathcal{G}_{\mathcal{N}_i}$.

5.2. Energy functions

Recall that our goal is to estimate deformation parameter Θ, whose resulting $\mathbf{M}(\Theta)$ best explains $\mathbf{Y}$. Given correspondences $\mathcal{C}$ from above, a data term is formulated as:

$$E_{data}(\Theta; \mathcal{C}) = \sum_{(i,p) \in \mathcal{C}} w_{ip} \|\mathbf{y}_i - \mathbf{x}_p(\Theta)\|_2^2, \tag{5}$$

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. We follow the simultaneous estimation framework [14], which applies the following regularization terms:

Preserving local rigidity. Without any prior knowledge of motion, patches are preferred to be distributed uniformly on $\mathcal{M}$. Given a vertex v, the rigidity constraint enforces the predicted positions $\mathbf{x}_v(\theta_k)$ and $\mathbf{x}_v(\theta_l)$ from two adjacent patches P_k and $P_l \in \mathcal{N}_k$ to be consistent:

$$E_r(\Theta) = \sum_{k=1}^{N_P} \sum_{P_l \in \mathcal{N}_k} \sum_{v \in P_k \cup P_l} w_{kl} \|\mathbf{x}_v(\theta_k) - \mathbf{x}_v(\theta_l)\|_2^2, \tag{6}$$

where Θ is implicitly encoded in $\mathbf{x}_v(\theta_k)$ and $\mathbf{x}_v(\theta_l)$.

Pose posteriors. It is noteworthy that the method described so far applies to not only humans but all types of surfaces. Nevertheless, since our application is capturing human motions, skeletal poses are also of our interest. We employ the skeleton binding energy that keeps the relationship between skeletons and surfaces:

$$E_b(\Theta, \mathcal{J}) = \sum_{k=1}^{N_P} w_k \|\mathbf{T}_{\theta_k}(\boldsymbol{\beta}_k^0) - \boldsymbol{\beta}_k\|_2^2. \tag{7}$$

Here $\boldsymbol{\beta} \in \mathbb{R}^3$ stands for the β-coordinate proposed by Straka *et al.* [25]. Θ is encoded in $\mathbf{T}_{\theta_k} \in SE(3)$. Our

Sequence	Views	Frames	Outliers	Err. metric	Compared approaches	Subject / # Vertices
Goalkeeper [2]	48	176	-	-	-	S1 / 4980
Crane [29]	8	173	-	A	surICP [6], artICP [8]	S2 / 3407
Jumping [29]	8	149	-	A		
Handstand [29]	8	173	-	A		S3 / 3848
Bouncing [29]	8	174	-	A		
Cutting	9	91	√	A	fixOL [6], bpSVM [14], patchedOL [15]	S4 / 5211
WalkChair1	9	130	√	A		
HammerTable [15]	9	93	√	A & B	[6], [14], [15], [24] + [25]	
WalkChair2 [15]	9	148	√	A & B		S5 / 5233

Table 1. Sequences used for evaluation. We apply two different error measures, depending on the provided ground truths. A: silhouette overlap errors averaged over all views. B: distances to annotated joint positions in pixels.

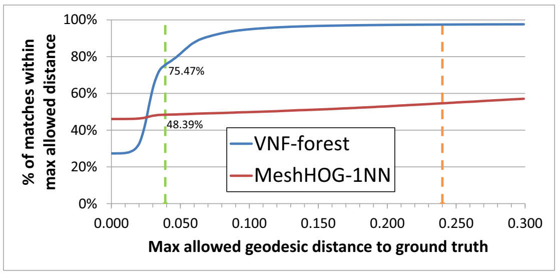


Figure 6. Matching error comparison of our method and MeshHOG of S2. We yields more locally consistent correspondences.

pose parameter $\mathcal{J} = \{\mathbf{x}_j\}_{j=1}^{N_j}$ is a set of 3D locations for $N_j = 15$ joints, implicitly encoded in $\boldsymbol{\beta}$.

Combining Eq. 5-7, we have the final energy:

$$E(\Theta, \mathcal{J}; \mathcal{C}) = \lambda E_{data}(\Theta; \mathcal{C}) + E_r(\Theta) + E_b(\Theta, \mathcal{J}). \quad (8)$$

λ defines the softness of the surface, and is empirically set as 10 throughout our experiments. Given an input $\mathcal{Y}$, the regression forest returns a fixed response $\tilde{\mathbf{Y}}$, and hence a fixed $\mathcal{C}$. We therefore apply standard Gauss-Newton method directly to find the minimizer of Eq. 8. Note anyway that refining $\mathcal{C}$ like ICP is always possible. In this case, our method provides better initializations than using last frame results, reducing the number of needed ICP-iterations.

6. Experiment results

The proposed method was evaluated extensively on 9 sequences, whose profiles are summarized in Table 1. An individual forest is trained for each subject with up to 200 meshes, depending on the number of vertices per mesh. For S1 $-$ 3 we train standard regression forest; for S4 & S5 we apply the adaptation in Eq. 4 due to the un-properly segmented chairs and tables in input data ($\alpha = 2$). Growing $T = 20$ trees to depth 25 with 15000 testing offset pairs ψ takes about 8 hours. The performance of our method is analyzed in different aspects, both qualitatively and quantitatively. We evaluate shapes with the widely-used silhouette overlap error, and evaluate poses with the 2D distances between ground truths and the projected joint locations.

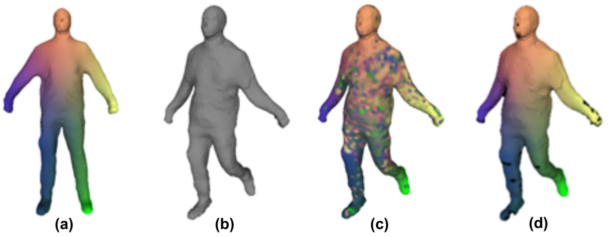


Figure 7. (a) references surface of S2 (vertex indices encoded with colors). (b) test mesh. (c) matching results with MeshHOG. (d) matching results of the proposed VNF-forest framework.

Descriptiveness. First we briefly demonstrate the discriminability of the VNF-forest combination. The task is finding correspondences on the reference surface of S2. We consider animated meshes as testing data, where ground-truth vertex indices are available. The error metric is the geodesic distance between predicted vertices and ground-truths, and we compare with MeshHOG [30] which is the extension of image-based histogram of oriented gradient (HOG) on surface manifolds.

Fig. 6 depicts the proportion of vertices whose errors are less than a certain distance. The VNF-forest framework gives overall 27.34% exact matches, while searching the vertex with closest MeshHOG response gives 46.07%. However, when increasing the tolerance of correctness, our method presents a significant improvement, and yet MeshHOG gains only a little. With our method, 75.47% of vertices obtain correspondences no farther than three times averaged edge length (green dashed line), and 48.39% for MeshHOG. Moreover, almost half of matches from MeshHOG lie outside the range of roughly the length of the lower arm (orange dashed line) from the ground truths. Such a phenomenon is confirmed in Fig. 7. Taking Fig. 7(b) as input, MeshHOG returns results as in Fig. 7(c), whereas our method gives visually smoother matches like Fig. 7(d). The proposed VNF-forest framework certainly provides more

	ours + ICP		[24] + [25]		bpSVM [14]		patchedOL [15]		fixOL [6]
	sil. error		sil. error		sil. error		sil. error		sil. error
WalkChair1	**7432**		-		fail		8931		24976
Cutting	**4048**		-		fail		20385		fail
	sil. error	joint error	sil. error	joint error	sil. error	joint error	sil. error	joint error	sil. error
HammerTable	4019	13.3 ± 6.9	17285	64.2 ± 53.9	fail	fail	**3593**	$\mathbf{10.1 \pm 3.0}$	fail
WalkChair2	**6144**	$\mathbf{15.8 \pm 6.1}$	12219	20.6 ± 22.0	18063	24.6 ± 10.7	6803	15.9 ± 6.3	18482

Table 2. Pixel overlap error of 4 sequences over all frames and all cameras. Image resolution: 1000 × 1000.

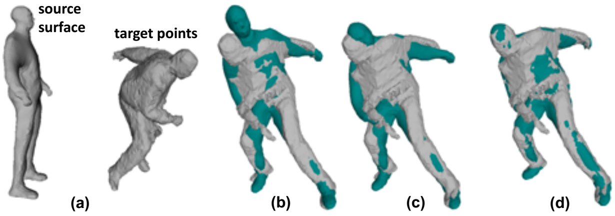


Figure 8. (a) the source surface and target points. (b) registration results (green) of EM-ICP [6] after 30 iterations. (c) results of VNF-forest framework (no iterations). (d) results of VNF-forest framework plus 14 iterations of EM-ICP.

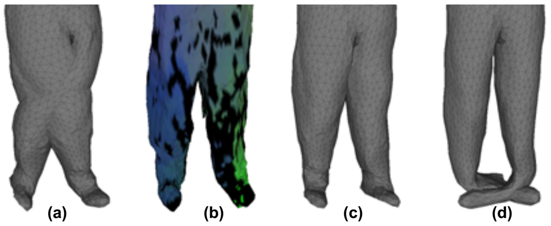


Figure 9. (a) results of frame 91 in *bouncing*. (b) visual hulls of frame 92 and correspondences from the forest in colors. (c) results of frame 92 from our approach. (d) results from surICP [6].

locally consistent associations, which, as shown below, are sufficiently accurate to guide plausible deformations.

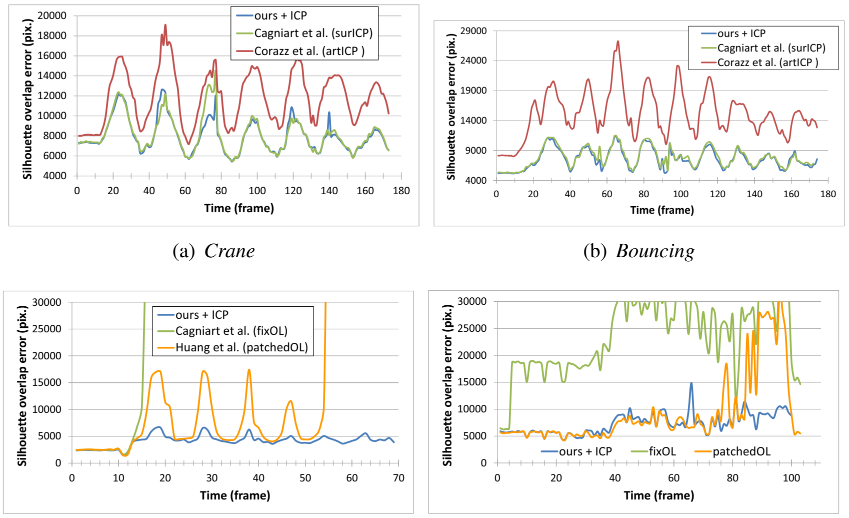
Registration. Next we show the capacity of the proposed method on registration tasks. With the source surface and distant target points as in Fig. 8(a), ICP-like approach only produces results like Fig. 8(b), where many discrepancies still remain after 30 iterations. Instead, our VNF-forest framework brings the source mesh closer to the target points in one shot as in Fig. 8(c). After 14 ICP-iterations it refines the results as in Fig. 8(d). In the following experiments if reference surfaces are not aligned with the first frames, we also register them before the tracking starts.

Tracking without outliers. For sequences without outliers, we compare with surface-based ICP (surICP) [6] and articulated ICP (artICP) [8], 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 silhouettes overlap errors are shown in Fig. 10(a-b). In general, our method performs much better than artICP, and yields comparable results with surICP. However, our method requires less ICP-iterations to converge (see *Supplementary Material*). It follows that, compared with using results of previous frames as initializations, our method is capable of providing better ones. Examples of the estimated correspondences, the deformed surfaces and the skeletons are shown in Fig. 12(a).

We further investigate what happens when there are tracking errors in previous frames. Fig. 9(a) shows the results of frame 91 in sequence *bouncing*. Note that left/right legs are crossed due to their close interaction with each other. In the next frame, when they are separate in the visual hull, the forest discovers correspondences more correctly, as in Fig. 9(b) (c.f. the colors of $\mathcal{M}$ in Fig. 2), and lead us to results in Fig. 9(c) without self-intersections. ICP strategy discover wrong associations and gives results in Fig. 9(d) under the same softness parameter λ. Errors propagate to the next frame, and gradually deteriorate the tracking, unless luckily regularization terms implies the opposite.

Tracking with outliers. 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) [6], removing outliers by body-part classifications with SVM (bpSVM) [14], and modeling outlier likelihood dynamically by aggregating over all patches (patchedOL) [15].

As shown in Fig. 10(c-d), 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 VNF, and feed them separately into the joint classification-regression forest. If they are connected to each other, forests inevitably associate some outliers to vertices on the model, and cause undesirable deformations, as the spike in blue curves in Fig. 10(d). Nonetheless, since we rely less on previous frames for data associations, the results can always get recovered when they are separated again. In average, we still yield low errors



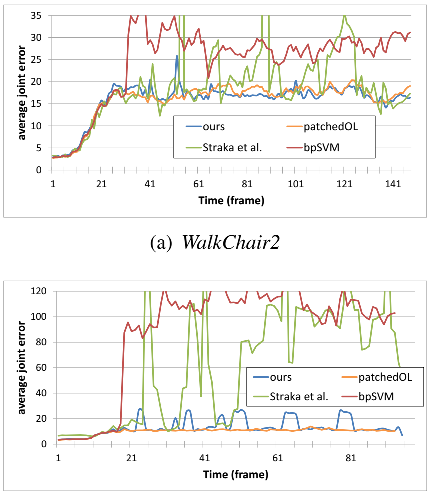
(a) *Crane*

(b) *Bouncing*

(c) *Cutting*

(d) *WalkChair1*

Figure 10. Pixel overlap error of 4 sequences, averaged over all cameras. Image resolution: (a-b): 1920 × 1080. (c-d): 1000 × 1000



(a) *WalkChair2*

(b) *HammerTable*

Figure 11. Averaged joint errors. Image resolution: 1000 × 1000.

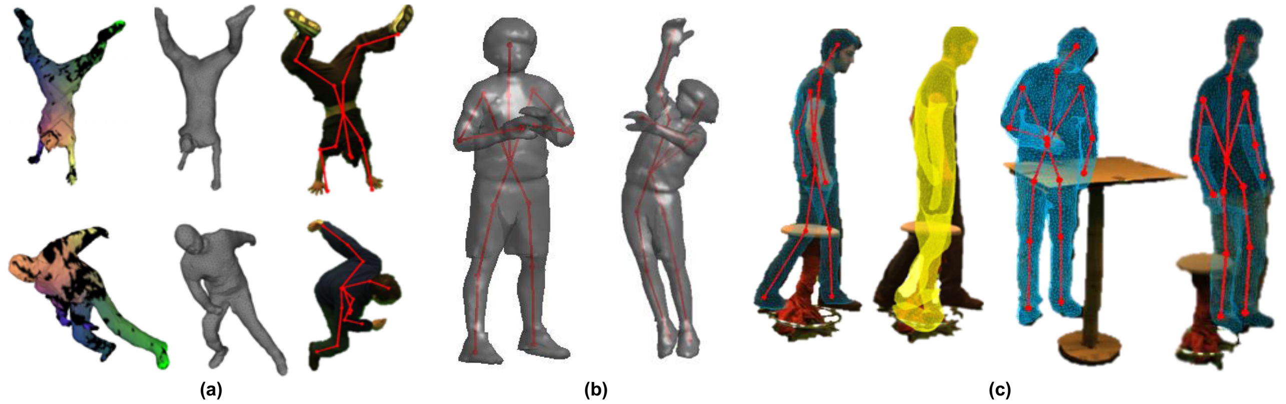


Figure 12. Qualitative results. (a) input visual hulls and the estimated correspondences visualized in different colors. Black color means no correspondence found for that vertex. Top: *Handstand*; bottom: *jumping*. (b) estimated shapes and poses of *Goalkeeper*. (c) estimated shapes and poses of *Cutting*, *WalkChair2*, and *WalkChair1*, overlayed on images. Blue: our results. Yellow: Cagniart *et al.* [6].

throughout the whole sequences, as in Table 2. We remark that such ability to recover is the essence of our discriminative approach, which is the biggest advantages over the existing generative methods.

We also verify the efficacy on pose estimations, and compare to another single-frame approach from Straka *et al.* [24, 25]. The error metric is the discrepancy between estimated joint positions and the annotated ground truths. Fig. 11 plots the results of sequences *HammerTable* and *WalkChair2*, where we confirm again considerably better accuracy than bpSVM [14] (red curves), Straka *et al.* [24, 25] (green curves), and comparable results with patchedOL [15]. The recovered shapes and poses are also presented in Fig. 12(c), superimposed on original images.

7. Conclusion

We present a hybrid human shape tracking approach equipped with an alternative data-model association strategy. With the help of regression forests, we learn from pre-animated meshes to discover better correspondences on input visual hulls. The one-shot property effectively prevent tracking errors from accumulating, yielding more stability compared with other generative ICP extensions. It contributes to locally consistent correspondences, which speeds up the convergence of ICP when used as the initializations. 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.

Acknowledgment. This work is funded by Taiwan Ministry of Education Scholarship n°1001121US089, and partially funded by Technische Universität München - Department of Informatics - Chair for Computer Aided Medical Procedures & Augmented Reality.

References

[1] T. Akenine-Möller. Fast 3d triangle-box overlap testing. In *SIGGRAPH 2005 Courses*. ACM, 2005.

[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] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*. IEEE, 2011.

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

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

[7] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *International Conference on Robotics and Automation*. IEEE, 1991.

[8] 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.

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

[10] 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.

[11] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. *IJCV*, 2013.

[12] 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.

[13] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *PAMI*, 2011.

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

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

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

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

[18] 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*. IEEE, 2011.

[19] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 2006.

[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. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *ICCV*. IEEE, 2007.

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

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

[26] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*. ACM, 2004.

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

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

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

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