Real-Time Upper Body Tracking with Online Initialization using a Range Sensor

Adolfo López-Méndez Marcel Alcoverro Montse Pardàs Josep R. Casas
Technical University of Catalonia (UPC)
{adolf.lopez, marcel.alcoverro.vidal, montse.pardas, josep.ramon.casas}@upc.edu

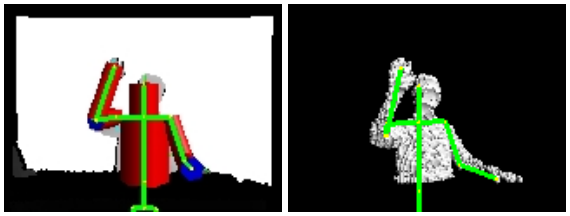
Abstract

We present a novel method for upper body pose estimation with online initialization of pose and the anthropometric profile. Our method is based on a Hierarchical Particle Filter that defines its likelihood function with a single view depth map provided by a range sensor. We use Connected Operators on range data to detect hand and head candidates that are used to enrich the Particle Filter's proposal distribution, but also to perform an automated initialization of the pose and the anthropometric profile estimation. A GPU based implementation of the likelihood evaluation yields real-time performance. Experimental validation of the proposed algorithm and the real-time implementation are provided, as well as a comparison with the recently released OpenNI tracker for the Kinect sensor.

1. Introduction

The technological evolution of sensors, such as cameras or microphones, has paved the way towards the research on new Human-Computer Interaction (HCI) paradigms based on human language. Among the possible research lines derived from this evolution, computer vision plays a major role with areas such as tracking, gesture, activity and object recognition. When attempting to interpret human activity and gestural language, human body tracking becomes a fundamental task, since it provides a markerless estimation of limb positions and even the anthropometric profile, i. e., limb sizes. In the last few years, the increasing computation power and especially Graphics Processing Units (GPU), as well as the eclosion of a wide variety of cameras, have brought human body tracking to a new level. Consequently, the applicability of human body tracking has gone beyond activity understanding. Recent experiments have proved its value in object recognition tasks [9] and in user authentication [8]. However, the paramount example of the current relevance of human body tracking is found in the mass market gaming: Microsoft's Kinect [15] allows people to use their body instead of the common controller, thus enhancing the gaming experience.

This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación, under project TEC2010-18094 and by the European Commission under contract FP7-215372 ACTIBIO.



a b

Figure 1. a) Human Body Model overlaid on an input image). b) Data Point Cloud and overlaid pose estimate

Human body tracking is a challenging problem that involves estimating a high-dimensional vector of pose parameters using video. Moreover, the pose estimation problem is strongly linked to the estimation of the anthropometric profile. Retrieving the upper body (torso and arms) deserves special attention due to its importance for the most common body language forms. However, although one could expect to be a simpler problem, occlusions of the lower body and partial occlusions of the torso make upper body pose and shape estimation a challenging problem in realistic environments.

In this paper, we present a novel method for upper body pose estimation with online initialization of pose and anthropometric profile. The main contributions of this paper are the following:

- Efficient use of geodesic saliency on range data and its introduction in a Hierarchical Particle Filter through Inverse Kinematics.
- An unsupervised method for online initialization of both pose and anthropometric profile (in terms of bone elongations) that does not require a specific (usually inconvenient) pose to start tracking.
- An optimized implementation of the likelihood evaluation that allows running our system in real-time.

Related Work Human body tracking is a hot topic within computer vision, as the survey by Moeslund et. al [19] and more recent approaches [10, 4, 5] show. In general, all these approaches differ in the body modeling framework, the method employed to measure how likely is a pose given an image or a set of images, and the technique used to find the optimal pose provided such a metric.

In this paper, however, we would like to emphasize the particular evolution of single-view approaches, currently linked to the technological improvements on range sensors: stereo cameras, Time-Of-Flight (TOF) cameras and Structured Light cameras. Bernier et al. [5] propose a body part assembly method based on Belief Propagation that relies on depth and color cues obtained with stereo cameras. Hauberg and Pedersen [13] also use stereo cameras in their approach, but they build their method upon a well-defined articulated body model. They use Particle Filters (PF) and novel spatial predictive distributions to tackle the pose estimation problem. A main drawback of stereo cameras is that the quality and quantity of depth data gets rapidly degraded in absence of textures and under generic lighting conditions. To tackle such a sensor problem, authors usually rely on TOF and Structured Light sensors. In [12, 17], Iterative Closest Point (ICP) algorithms are used to fit the surface of a model onto captured depth data. However, since ICP is essentially a local optimization method, it usually gets stuck on local minima. Ganapathi et al. [11] employ data from a TOF camera to discriminatively train a set of body part detectors. Their approach is based on propagating the belief of body part locations through the kinematic chains of an articulated model. Siddiqui and Medioni [22] propose an MCMC-based algorithm that integrates body part detectors in order to perform real-time upper body tracking. In their work, they also provide a method to estimate the anthropometric profile (*i.e.* bone elongations), but they do not perform such an estimation online.

More recently, Kinect [15], a particular depth sensor based on structured light technology, has been made available on the mass market. Together with the sensor, two innovative algorithms for human body tracking have stood out on their own among the state of the art [20, 14].

2. Hierarchical Particle Filter with Geodesic-Driven Proposals

In order to estimate the human pose, we adopt an analysis-by-synthesis approach whose central element is a human body model. This model comprises an articulated structure representing the underlying skeletal structure of the human body, and a set of cylindrical shapes with elliptical cross-sections that model the limb shapes (see Fig. 1). The rendering of the model is necessary for the likelihood evaluation, as it will be detailed in Section 2.1.

We follow the twists and exponential map formulation [6] to parameterize human poses as a global translation and rotation and a set of joint rotations. Global translation and rotation are associated to a root joint from which siblings grow to encode the full skeletal structure in the form of a Kinematic Tree, a directed graph structure that efficiently represents the hierarchical nature of the human body model.

Pose estimation with such a hierarchical representation of joints is attained by a Hierarchical Particle Filter (HPF) [4]. Let $\mathbf{x}$ be the vector containing the model parameters: global translation and rotation and joint rotations. Provided that at each time instant t, a set of observations $\mathbf{z}_t$ are produced by the state vector $\mathbf{x}_t$, the goal is to recursively estimate the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ given the observations up to time t. The noteworthiness of the HPF resides in tackling the problem by dividing the state space vector into hierarchical partitions, thus providing an efficient solution to the curse of dimensionality in Particle Filters [3] and a reasonable approach with the employed body representation.

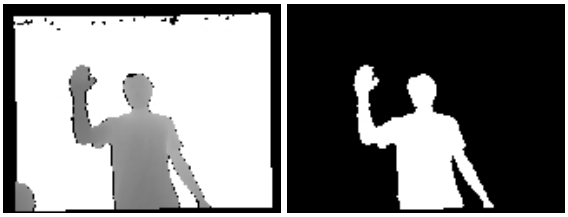
Let us define a set of L hierarchical partitions or layers, where the tuplet $\{\mathbf{x}^i_{t,l}, w^i_{t,l}\}$ denotes particles and weights in the l-th layer. In each one of these layers, the filtering step focuses only on a subset of variables χ_l. In our method, we opt for dividing the upper body model into 3 different partitions: torso + head, left arm and right arm. Therefore, in the first layer we sample and filter torso particles, in the second we do the same with left arm particles, using the torso filtered state, and in the last layer we filter and sample right arm particles, using the filtered state of the rest of model variables. This yields a sufficiently low number of dimensions per partition. In this context, sequential importance sampling is divided into a layered filtering where the importance weights in each layer l can be formulated as follows:

$$w^i_{t,l} \propto \frac{p(\mathbf{z}_{1:t}|\mathbf{x}^i_{t,l})p(\mathbf{x}^i_{t,l}|\mathbf{x}^i_{t,l-1})}{q(\mathbf{x}^i_{t,l}|\mathbf{x}^i_{t,l-1}, \mathbf{z}_{1:t})} \qquad for\ l > 0 \quad (1)$$

$$w^i_{t,0} \propto \frac{p(\mathbf{z}_{1:t}|\mathbf{x}^i_{t,0})p(\mathbf{x}^i_{t,0}|\mathbf{x}^i_{t-1,L-1})}{q(\mathbf{x}^i_{t,0}|\mathbf{x}^i_{t-1,L-1}, \mathbf{z}_{1:t})} \qquad for\ l = 0 \quad (2)$$

where the numerator contains the product of the likelihood and the prior, and the denominator contains the proposal distribution. Note that we have assumed Markovian state transitions, resampling in every layer and that the proposal distribution $q()$ factorizes such that we can perform sequential importance sampling within layers.

The common choice for $q()$ is the prior distribution $p(\mathbf{x}^i_{t,l}|\mathbf{x}^i_{t-1,l})$, yielding a direct proportionality between the weights and the likelihood term. Nonetheless, since this prior is normally modeled by a Gaussian distribution, the HPF particle propagation step usually becomes blind to data.



a b

Figure 2. a) Depth map (depth maps shown in this paper are conveniently clipped and scaled for better visualization in grayscale images). b) Extracted silhouette.

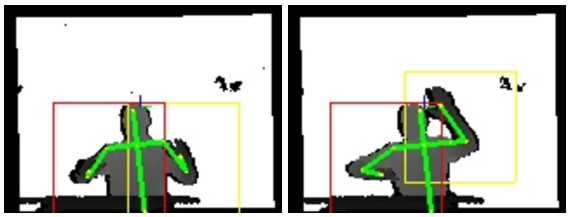


Figure 3. Examples of 2D Bounding Boxes for both right and left arms (overlaid on the input depth data). Only foreground depth pixels enclosed in the respective bounding boxes are used to evaluate the likelihood of each arm particles.

In our method, we exploit *geodesic saliency* on range data to obtain cues about the location of head and hands; these cues are introduced in the prior distribution, thus reducing the blindness of the filter. This approach is detailed in Section 2.2.

Finally, in the last layer, we compute the pose estimate. The usual choice for this estimate is the mean of the weighted samples. Nevertheless, the proposed modifications on the propagation step might yield increasingly multimodal distributions, hence we opt for providing the final pose as the particle with maximum weight (MAP estimate) to avoid drifts in the mean.

2.1. Likelihood Definition

In order to estimate the upper body pose, our method relies on a single range sensor providing a depth map $\mathbf{D}_t$ at each time instant t (see Fig. 2a). Using learnt depth background models, we perform foreground subtraction by simple thresholding. Assuming that only one human is to be tracked, a largest component filtering procedure is applied to remove spurious blobs. This procedure yields a mask for the actual depth data representing the human (see Fig. 2b). To effectively compare these data with the state space hypothesis represented by particles $\mathbf{x}_t^i$, we render the cylindrical shapes with elliptical cross-sections attached to the model bones on an image of the same resolution as the input depth map.

As no real likelihood is available, we define the likelihood term as a monotonic decreasing mapping of a sum of cost functions:

$$p(\mathbf{z}_t|\mathbf{x}_{t,l}^i) \approx \quad e^{-(\lambda_1 c_d(\mathbf{D}_t,\mathbf{x}_{t,l}^i)+\lambda_2 c_f(\mathbf{D}_t,\mathbf{x}_{t,l}^i)+\lambda_3 c_p(\mathbf{x}_{t,l}^i))} \tag{3}$$

where c_d denotes the cost function computed from depth data, c_f is the cost function computed with foreground silhouettes, c_p is a cost term that takes into account physical constraints such as interpenetration of limbs or wrong arm configurations, and λ_i are multipliers reflecting the importance of each cost term in the final likelihood evaluation. Cost c_d is the mean of pixel-wise squared differences between data and the model, thresholded by a maximum distance d_{max}^2 (each missed depth pixel adds d_{max}^2 to the cost). Cost c_f is a pixel-wise XOR between data and model silhouettes. Model silhouettes and depth values are efficiently obtained by means of the OpenGL depth buffer [21].

HPF likelihood functions in each partition should be properly peaked around the same region of the posterior restricted to those state variables in the partition of interest. To achieve such a property, we adopt a variant of the strategy proposed in [18]. Using the pixel positions of the joints in the previous frame, we define 2D bounding boxes of fixed size enclosing them all (see Fig. 3). In this way, we restrict the likelihood evaluation to a local region of the depth map that is likely to provide the most meaningful information for that partition. The reason for choosing a fixed size bounding box is the straightforward application on the implementation of this method in a GPU.

GPU implementation. Based on existing GPU implementations for PF algorithms aiming at articulated tracking [7], we propose an implementation for HPF-based human body tracking with range data. Specifically, we opt for implementing the likelihood evaluation using OpenCL [23] with OpenGL interoperability. The reason for such a combination is to take advantage of a general purpose programming language as OpenCL and the rendering capabilities of OpenGL.

We start at the first layer or partition of the HPF with the torso particles. In this layer, we compute a squared 2D bounding box of approximately 1/4 of the image resolution (due to GPU implementation reasons the sizes must be powers of 2), centered at the projection of the estimated body model centroid in the previous frame. For each particle, we render the model directly to a depth texture and then we use Quad primitives to map the pixels enclosed by the 2D bounding box to a bigger RGBA texture that we call the *mosaic texture*. The objective of this latter texture is to gather all the particle *tiles* that must be evaluated in one layer (see Fig. 4). After mapping a particle onto its corresponding *tile* on the *mosaic texture*, we load the bounding box offset

onto GPU global memory. We repeat this procedure until the *mosaic texture* is filled, yielding the maximum number of particles per layer.

We efficiently share the *mosaic texture* with OpenCL through its interoperability mechanisms, and we compute the pixel-wise depth and XOR costs in Equation 3. The OpenCL implementation uses different threads to perform these two costs, so that pixels are processed in parallel. Specifically, each thread looks for the corresponding pixel in the input depth map using its position in the *mosaic texture* and the offset previously loaded onto global memory. In this way, we compute both depth and XOR costs with one single read of the texture. The results of both costs and the evaluated pixels are stored in global memory (see Fig. 4). After obtaining the pixel-wise differences, we perform a modified 2D sum-reduction on OpenCL. This version of the well-known sum-reduction is constrained to provide the cost of every particle instead of the sum of all the pixel values.

The described method is repeated in the remaining arm layers.

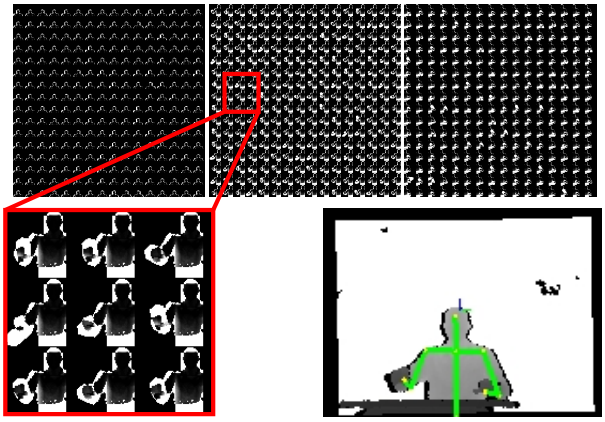


Figure 4. GPU computation of likelihoods. Top Row: Result of the pixel-wise sum of squared differences between depth maps on the *mosaic textures* for the 3 layers involved in the hierarchical evaluation of the likelihood (torso, right arm and left arm). These distance maps are stored in global memory to compute the final particle cost. Bottom Row: Right arm depth cost tiles in detail and input depth data with overlaid final estimation.

2.2. Geodesic-Driven Proposals

Our choice for the HPF proposal distribution is to draw samples from the prior. The simplest form of prior is a Gaussian $\mathcal{N}(\mathbf{x}^i_{t,l-1}, \mathbf{\Sigma}_{\chi_l})$. However, this strategy presents a main drawback: particles are drawn blindly with respect to data. Our objective is to introduce cues extracted from range data to build an appropriate data-driven proposal distribution that may reduce the blindness of the particle set. We achieve this goal by estimating the locations of relevant upper body end-effectors: head and hands.

In order to detect the upper-body end-effectors we employ a filtering strategy based on [2]. The key idea is to retrieve a set of salient geodesic extrema of the depth map on a per frame basis. This method starts by defining a graph $G = (V, E)$ where the vertices V are the depth pixels and edges E are defined using 8-connectivity neighborhoods constrained by a maximum distance d_{thr}: if a neighbor pixel represents a point that is farther away than d_{thr} in 3D world coordinates, the edge between both vertices is discarded. We then need to define a source point in order to obtain an approximate geodesic map on the human body surface. We choose the closest pixel to the center of mass of the masked depth information as source point. In fact, we find more convenient to lower the center of mass in order to obtain better geodesic extrema detection. Finally, we filter the geodesic map using a Component Tree and the strategies proposed in [2] to obtain the 3 most prominent geodesic extrema, that are likely to accurately represent head and hands location.

These geodesic extrema constitute a set of candidates for hands and head locations at each time instant t. Provided that the head location is, to a great extent, a result of rigid motion of the torso, we only consider geodesic extrema in the propagation of arm particles.

In order to efficiently incorporate these cues, obtained from range data processing, we model the prior distribution using Inverse Kinematics (IK), thus following the ideas of the work by Hauberg and Pedersen [13]. The main difference here is that in each frame we have a maximum of three end-effector candidates, but we do not know if they are head, left hand, right hand or simply a detection error. Thus, instead of modeling the spatial distribution of end-effectors, we efficiently incorporate a few, spatially sparse detections $\mathbf{g}_t$ as modes of a new prior distribution $p(\mathbf{x}^i_{t,l}|\mathbf{g}, \mathbf{x}^i_{t,l-1})$. Since we want to keep a direct proportionality between weights and likelihood, we draw samples from the modified prior distribution (see Equation 1), thus alleviating the inefficiency related to the blindness of the proposal distribution. Let F be the forward kinematics operator [13] such that $F(\mathbf{x}^i_{t,l})$ gives the end-effector location associated to l-th layer (in our case right or left hand). We formulate the distribution of an arm pose given the m-th location as:

$$\begin{aligned}
&\log p(\mathbf{x}^i_{t,l}|\mathbf{g}_t[m], \mathbf{x}^i_{t,l-1}) \triangleq \\
&-\frac{1}{2}(\mathbf{g}_t[m] - F(\mathbf{x}^i_{t,l}))\mathbf{\Sigma}^{-1}_{\epsilon}(\mathbf{g}_t[m] - F(\mathbf{x}^i_{t,l})) \\
&-\frac{\lambda}{2}\|\mathbf{x}^i_{t,l} - \mathbf{x}^i_{t,l-1}\|^2 \\
&+\sum_{n=1}^{N} \log \mathcal{U}_{[a_n,b_n]}(\mathbf{x}[n]) + C
\end{aligned} \tag{4}$$

The first term models the error between an extremity de-

tection and the end-effector position in the model obtained by forward kinematics. The second term is a smoothing constraint. The last terms are the kinematic constraints, formulated as in [13], and a constant.

To model the distribution conditioned by $\mathbf{g}_t$ we use a mixture model, where the closer the extremity detection is from the previous end-effector position, the higher the importance within the mixture:

$$h(\mathbf{x}^i_{t,l}|\mathbf{g}_t, \mathbf{x}^i_{t,l-1}) \propto \sum_{m=1}^{M} \frac{p(\mathbf{x}^i_{t,l}|\mathbf{g}_t[m], \mathbf{x}^i_{t,l-1})}{f(\|\mathbf{g}_t[m] - F(\mathbf{x}^i_{t,l-1})\|^2)} \quad (5)$$

where f is some increasing function (in practice we use an exponential or an adequate inverse of the Heaviside function in order to simply threshold the distance). Note that the mixture weights should be properly normalized.

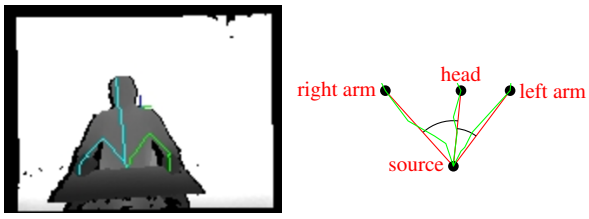
Finally, we use another mixture model to construct the prior (and thus the proposal distribution):

$$p(\mathbf{x}^i_{t,l}|\mathbf{g}_t, \mathbf{x}^i_{t,l-1}) \triangleq \alpha\mathcal{N}(\mathbf{x}^i_{t,l-1}, \mathbf{\Sigma}_{\chi_l}) + (1-\alpha)h(\mathbf{x}^i_{t,l}|\mathbf{g}_t, \mathbf{x}^i_{t,l-1}) \quad (6)$$

Efficient sampling of this distribution is achieved thanks to the form of the new modes expressed by Equation 4. Note that, minimizing Equation 4 is equivalent to finding an IK solution and, consequently, a sample with high probability. This is the reason why we adopt this mixture model formulation of the distribution of arm poses given a salient geodesic extrema location. To solve the IK problem we employ the swing twist formulation [16]. Then, we generate additional samples by random rotations around the swivel axis. This is an efficient way to easily get a number of samples in the typical set of the modes of the distribution in Equation 5, because all of them are solutions to the unconstrained IK problem. Furthermore, in this way we generate particles with highly correlated variations between shoulder rotations, which are difficult to generate by simply propagating the corresponding Euler angles.

Finally, we sample from the prior distribution in Equation 6 in a specific manner. We select between Gaussian diffusion in the angle space or IK sampling in a deterministic manner as a function of an initial α. If the chosen sample is drawn from Equation 5 and, as a result, gets a very low probability, the sample is rejected and we jump to Gaussian diffusion. In this way, α changes dynamically while we avoid that some erroneous detections or assignments misdirect the samples.

In the end, we obtain an approximated method to map a set of salient geodesic extrema from range data into modes of the HPF prior distributions in arm layers. These geodesic-driven proposals confer our method a hybrid bottom-up and top-down nature.



a b

Figure 5. Extrema identification during pose initialization: a) Example of the *fork* pattern produced by geodesic paths in the neighborhood of the source point. b) The fork pattern is matched by checking that the cosinus of the angles between paths are in a given interval; then we can identify the three key end-effectors

3. Online Initialization

Automated initialization of a body tracker and, specifically, a particle filter-based markerless motion capture system is a challenging problem but also a desired feature for such a technology, as the online usability of the algorithm gets boosted. We propose using the geodesic extrema detection (see Section 2.2) and the swing twist IK method [16] under some assumptions to initialize the pose as well as to provide an estimate of the anthropometric profile of the user in terms of bone elongations.

Pose Initialization The assumptions for pose initialization are that the human may stand approximately with his or her back in vertical position, staring in front of the camera, and that right and left hands will be visible to the range sensor. Under such assumptions, whenever three reliable geodesic extrema are detected, we classify them as head, right arm or left arm by tracking the geodesic path that leads to the corresponding geodesic extrema. If the geodesic map has been properly computed, the direction of the paths leading to end-effectors remains similar in the neighborhood of the source point: left hand to the north-east and right hand to the north-west (considering a non-mirrored image) and head is reached through the middle path, usually to the north (see Fig. 5). Then, we can label the 3 end-effectors depending on the extent to which the geodesic path directions match the *fork* pattern (see Fig. 5). To match such a pattern, we compute the slope of each path in the neighborhood of the source point. We then check that the relative slope between each pair of paths is within a given interval. To measure the relative slope, we check that the cosinus of the difference angles between each pair of paths is within the interval $[0.5, 0.9]$.

If the geodesic path analysis yields a location for head, right and left hands (namely $\mathbf{g}_0$) then a pose configuration is computed as follows:

1. Translate the model to match the head
2. Compute arm poses by means of IK

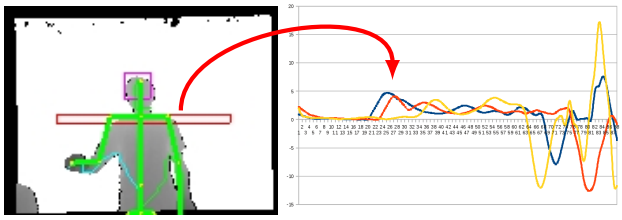


Figure 6. Shoulder breadth estimation: Derivatives of the silhouette width along the vertical axis of the depth image are used to locate the approximate shoulder positions during initialization. Plotted curves, starting at the head top location, show the first prominent local maxima of the derivative for 3 different users. Shoulder pixels are located in the neighborhood of this first prominent local maxima of the derivative.

3. Measure the probability of each computed pose with the following expression

$$\begin{aligned}&\log p(\mathbf{x}_0|\mathbf{g}_0,\mathbf{z}_0) \triangleq \\ &-\frac{1}{2}(\mathbf{g}_0 - F(\mathbf{x}_0))\boldsymbol{\Sigma}_\epsilon^{-1}(\mathbf{g}_t - F(\mathbf{x}_0)) \\ &+\sum_{n=1}^{N}\log \mathcal{U}_{[a_n,b_n]}(\mathbf{x}[n]) + C \\ &-(\lambda_1' c_d(\mathbf{D}_0,\mathbf{x}_0) + \lambda_2' c_f(\mathbf{D}_0,\mathbf{x}_0) + \lambda_3' c_p(\mathbf{x}_0))\end{aligned} \tag{7}$$

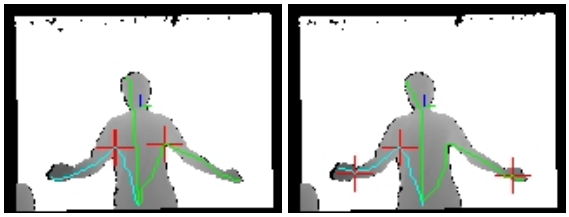
4. If $p(\mathbf{x}_0|\mathbf{g}_0,\mathbf{z}_0) > th$, where th is some threshold, accept the initial pose.

In comparison to Equation 4, here the first term measures the probability of the three end-effector locations. The last term incorporates a similar expression to the likelihood defined in Equation 3; in this case we dropped the subindex l, meaning that this expression is calculated for the three partitions. In practice, we average the costs of all the three partitions to obtain the initial probability.

Anthropometric Initialization For anthropometric estimation we assume the same for back inclination and also that, at some point, the target's upper arms will be approximately pointing towards the floor. When this situation happens, we trigger two measurements.

The first one is devoted to estimating the shoulder breadth. This measurement consists in analyzing the derivative of the summation of foreground pixels in each image row. Under the assumptions above, shoulders are found in rows closer to the first prominent local maximum of this derivative (see Fig. 6). Using this information, we extract several points from the leftmost and rightmost pixels of the rows around this maximum to compute an approximation of the shoulder breadth.

The second measurement concerns the arm length and is computed through the analysis of the geodesic path lengths.



a b

Figure 7. Points with high curvature along the projected geodesic path are marked with a red-cross. a) Correct detections below both shoulders b) Spurious detections near wrists. These detections are automatically rejected due to the proximity to the end-effectors.

In order to remove parts of the path located in the torso, we use second derivatives to analyze the curvature of the projection of the geodesic path on the image (see Fig. 7). Points with high curvature located between 25% and 50% of the path length (in real world units and starting from the source point) usually fall right below the shoulder joint, allowing us to use the resting path length as arm length measurement. If the arm length obtained by this method is within an interval of possible human arm lengths, the measurement is collected as valid. The total arm length is split into upper and forearm based on anthropometric studies [1].

In both cases, we perform anthropometric measurements during a short period of time. We perform outlier rejection by analyzing the variance of the collected measurements to finally provide the bone elongations needed to approximately fit the upper body model to the user.

The proposed method for estimating the anthropometric profile of the user barely relies on the model and the estimated pose. As a consequence, the robustness of the anthropometric estimates increases, because it is not affected by pose estimation errors.

4. Experimental Results

To validate our algorithm, we have conducted several experiments with range data recorded with a Kinect (640x480 pixels, 30fps). We consider two different scenarios where upper body motion is involved: desktop (Fig. 8) and workplace (Fig. 9). In desktop, users are sitting down in front of a table, while in workplace are standing up.

Desktop sequences involve 5 users performing several actions such as motion of one arm, picking a phone, or drawing some figure with both hands. These sequences comprise almost 4 minutes of data. Workplace sequences contain approximately 1 minute and a half of challenging motions performed by 2 subjects.

We pick several subsequences from this data and we manually annotate pixels belonging to joint positions. We perform these annotations in 1 of every 10 frames, obtaining around 470 annotated frames and more than 4700 frames to evaluate.

Tile Size	Device	64x3	256x3	1024x3
64x64	NVIDIA Quadro FX 3700	11.2	3.3	0.9
64x64	NVIDIA GeForce GTX 295	13.2	4.5	1.3
128x128	NVIDIA Quadro FX 3700	3.1	1.6	0.5
128x128	NVIDIA GeForce GTX 295	3.9	2.2	0.8
256x256	NVIDIA Quadro FX 3700	0.8	0.1	-
256x256	NVIDIA GeForce GTX 295	1.0	0.2	0.1

Table 1. Computational performance of the complete system (in frames per second) as a function of the number of particles (x3 layers) and the size of the tiles for different hardware platforms. The size of the tiles is proportional to the image resolution (e.g. a 64x64 tile is for 160x120 depth maps). *Mosaic textures* of 8192x8192 are not supported by the NVIDIA Quadro FX 3700.

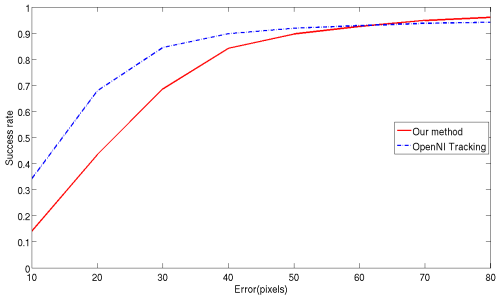


Figure 10. Success rate comparative between PrimeSense tracker and our method. Errors have been computed in 640x480 images.

In the first experiment, we evaluate the computational performance of the implemented system in two different hardware platforms: both CPUs are Intel 2.80GHz, 4GB RAM, but different GPUs are installed (see Table 1). Although we have only optimized the likelihood computation, the system runs online in a laptop with an ATI Mobility Radeon HD5800; to reach the online performance, we use a total of 192 particles and 160x120 depth maps (the original data at 1/16 resolution). With this configuration, we obtain a satisfactory performance for several upper body actions.

In the second experiment, we run our system offline with the recorded sequences in order to compare our results with the recently released PrimeSense body tracker for Kinect, which is accessible thanks to the OpenNI middleware [14]. This tracker requires a specific initial pose (hands up) called the *calibration pose*. In order to perform a comparison between both methods, all the annotated sequences have been recorded with this *calibration pose*. Nonetheless, since our method incorporates automated initialization of pose and anthropometrics, we can successfully launch our tracker without requiring such a specific pose. In these experiments, we use 256 particles in each hierarchical layer and depth maps of 160x120 pixels, yielding close to real-time performance.

Using the available annotations, we measure the accuracy/precision of both systems by means of a success rate: given a distance in pixels, we count the percentage of joints

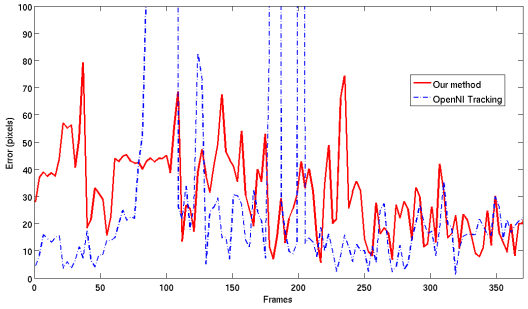


Figure 11. Right hand error (pixels) for the PrimeSense tracker and our method in a sequence where PrimeSense tracker has several misses. Errors have been computed in 640x480 images.

that have been tracked with an error below this distance. In the annotated sequences, hands are visible all the time, hence both algorithms should be able to provide estimates for the upper body joints. Since PrimeSense tracker may not provide joint locations that considers unreliable, we consider these cases as misses. The results, measured in 640x480 frames, are shown in Fig. 10. Our system shows a remarkable precision, although being less accurate than the PrimeSense tracker, that relies on accurate part detectors. However, we have observed that, while for a long tracking period and fast motions, the PrimeSense tracker outperforms our method, there are some cases, with limb self-occlusions, in which our method has a better performance. Specifically, when shoulders are occluded or a hand is partially occluded, the PrimeSense tracker can fail (see Fig. 11). These cases are more frequent in the desktop scenario, showing that the upper body tracking task is a difficult problem due to important occlusions of the lower body. In our method, the use of a body model and hierarchical layers helps in overcoming these cases. In overall, the PrimeSense method presents a mean tracking error of 20.74 pixels (excluding misses) while ours has an error of 28.95 pixels. The mean initialization error of our method is 33.31 pixels.

5. Conclusions and Future Work

In this work, we have proposed an upper body tracking approach with online and unsupervised initialization of both pose and anthropometrics. The system uses Geodesic-Driven proposals within the HPF formalism in order to improve the tracking performance. As we have shown, these geodesic cues are also useful during the initialization of the tracker. In addition, we have proposed a GPU implementation of the likelihood evaluation that yields real-time performance. Experiments with annotated data have shown the efficiency of the proposed system.

Future work involves improvement of the human silhouette extraction, research on additional body part detection methods for range data and a full optimization of the whole

Figure 8. Tracking results in desktop upper-body sequences recorded with Kinect. 256 particles per layer are used.

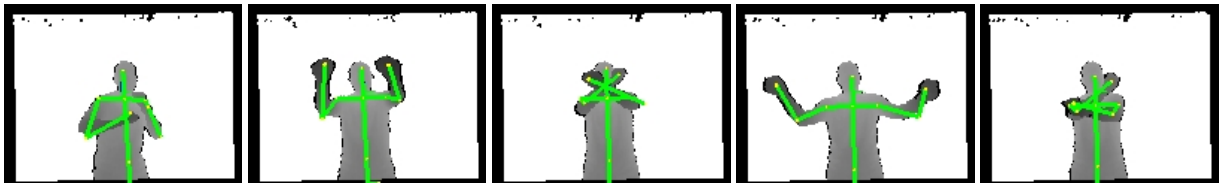


Figure 9. Tracking results in workplace upper-body sequences recorded with Kinect. 256 particles per layer are used.

system to further improve the computational performance.

References

[1] National Aeronautics And Space Administration. Man-Systems Integration Standards. Technical report. 6

[2] M. Alcoverro, A. López-Méndez, M. Pardàs, and J. R. Casas. Connected operators on range data for human body analysis. *CVPR Workshops*, june 2011. 4

[3] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. Adelaide. A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. 2

[4] J. Bandouch and M. Beetz. Tracking humans interacting with the environment using efficient hierarchical sampling and layered observation models. In *ICCV-HCI*, 2009. 2

[5] O. Bernier, P. Cheung-Mon-Chan, and A. Bouguet. Fast non-parametric belief propagation for real-time stereo articulated body tracking. *CVIU*, 113(1):29–47, 2009. 2

[6] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR*, pages 8 –15, jun 1998. 2

[7] R. Cabido, D. Concha, J. Pantrigo, and A. Montemayor. High Speed Articulated Object Tracking Using GPUs: A Particle Filter Approach. In *ISPAN*, pages 757 –762, dec. 2009. 3

[8] A. Drosou, K. Moustakas, D. Ioannidis, and D. Tzovaras. Activity related biometric authentication using spherical harmonics. *CVPR Workshops*, june 2011. 1

[9] J. Gall, A. Fossati, and van Gool L. Functional categorization of objects using real-time markerless motion capture. *CVPR*, june 2011. 1

[10] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel. Optimization and filtering for human motion capture - a multi-layer framework. *IJCV*, 0:1–18, 2009. 2

[11] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *CVPR*, pages 755 –762, june 2010. 2

[12] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. *LNCS*, 3663:285–292, 2005. 2

[13] S. Hauberg and K. Pedersen. Predicting articulated human motion from spatial processes. *IJCV*, pages 1–18, 2011. 2, 4, 5

[14] http://www.primesense.com/. 2, 7

[15] http://www.xbox.com/kinect. 1, 2

[16] M. Kallmann. Analytical inverse kinematics with body posture control. *Journal of Visualization and Computer Animation*, 19:79–91, 2008. 5

[17] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3D human body tracking with an articulated 3D body model. In *ICRA*, pages 1686 –1691, may 2006. 2

[18] A. López-Méndez, M. Alcoverro, M. Pardàs, and J. R. Casas. Approximate partitioning of observations in hierarchical particle filter body tracking. *CVPR Workshops*, june 2011. 3

[19] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006. 2

[20] J. Shotton, A. Andrew Fitzgibbon, M. Cook, and A. Blake. Real-time human pose recognition in parts from single depth images. *CVPR*, june 2011. 2

[21] D. Shreiner, M. Woo, , J. Neider, and T. Davis. *OpenGL programming guide*. Addison-Wesley, 2005. 3

[22] M. Siddiqui and G. Medioni. Human pose estimation from a single view point, real-time range sensor. In *CVPR Workshops*, pages 1 –8, june 2010. 2

[23] R. Tsuchiyama, T. Nakamura, T. Iizuka, A. Asahara, and S. Miki. The OpenCL Programming Book. *Group*, 2009. 3