Real Time Motion Capture Using a Single Time-Of-Flight Camera

Varun Ganapathi Christian Plagemann Daphne Koller Sebastian Thrun

Stanford University, Computer Science Department, Stanford, CA, USA

{varung,plagemann,koller,thrun}@stanford.edu

Abstract

Markerless tracking of human pose is a hard yet relevant problem. In this paper, we derive an efficient filtering algorithm for tracking human pose using a stream of monocular depth images. The key idea is to combine an accurate generative model—which is achievable in this setting using programmable graphics hardware—with a discriminative model that provides data-driven evidence about body part locations. In each filter iteration, we apply a form of local model-based search that exploits the nature of the kinematic chain. As fast movements and occlusion can disrupt the local search, we utilize a set of discriminatively trained patch classifiers to detect body parts. We describe a novel algorithm for propagating this noisy evidence about body part locations up the kinematic chain using the unscented transform. The resulting distribution of body configurations allows us to reinitialize the model-based search. We provide extensive experimental results on 28 real-world sequences using automatic ground-truth annotations from a commercial motion capture system.

1. Introduction

If motion capture technology were to become convenient, cheap, and applicable in natural environments, then a whole range of applications would become possible, such as intuitive human-machine interaction, smart surveillance, character animation, virtual reality and motion analysis. It is likely that many such applications will become apparent once the technology is available.

The only viable solution today, that is, *marker-based* human motion capture, has so far mainly been used in the entertainment industry. The need for special-purpose cameras and inconvenient markers or suits as well as the high operation costs have inhibited broader use. As a result, there has been much interest in the area of *markerless* motion capture, and such systems are becoming more popular [12].

In recent years, algorithms have been proposed that capture full skeletal motion at near real-time frame rates; however, they mostly rely on multi-view camera systems and special controlled recording conditions that limit their applicability. Less expensive systems that use a narrow baseline camera system have not yet reached a similar level of maturity. Most monocular approaches so far aim at solving simplified versions of the full articulated motion capture problem, such as gesture disambiguation, or capture of restricted motion for certain parts of the body.

Time-of-flight sensors are a technology that offers rich sensory information about a large part of the scene and, at the same time, enables a convenient, non-invasive system setup. These sensors provide dense depth measurements at every point in the scene at high frame rates. The range data provided allows easy segmentation of the human body and can also disambiguate poses that would otherwise have similar appearance and therefore confuse most monocular systems. Range sensors, in general, lend themselves to a faithful generative model (as the robotics literature shows), because they are not sensitive to changes in lighting, shadows, and the variety of the problems that make it nearly impossible to generatively model intensity images. In the future, these sensors are likely to be as cheap as webcams are today. Thus, we approach the human motion capture task using time-of-flight sensors. Despite the advantages of depth sensors, however, several hard problems have to be dealt with, including the high dimensionality of the state-space (48 degree-of-freedom in our case) and the nonlinear, highly peaked likelihood function.

We propose in this paper a probabilistic filtering framework that employs a highly accurate generative model—which is achievable in this setting using an efficient GPU implementation—with a discriminative model. Our algorithm was developed specifically for fast operation at video frame rates, since natural communication requires a low-latency action-reaction cycle. The presented system requires $100 - 250$ms per camera frame to estimate the joint angles of a 48 degree-of-freedom human model.

Our primary contribution is a novel algorithm for combining discriminative part detections with local hill-climbing for this task. Our secondary contribution is the definition of a smooth likelihood function and a means of implementing it on readily available graphics hardware (GPUs) efficiently in order to obtain near real-time performance. In addition to this, we constructed an extensive set of real-world test sequences with annotated ground truth, which are published openly [6] for future benchmarks.

2. Related Work

The automatic analysis of human shape and motion from sensor data has been researched considerably as Moeslund *et al*. [9] illustrate in their survey covering more than than 350 papers. Several learning based approaches [1, 17, 14] attempt to directly map image structures, silhouettes or features computed from them directly to poses. While this is an interesting approach in general, it is not clear yet how to scale it robustly to the general problem setting in unconstrained environments due to the high dimensionality of the human pose space. Related to our approach, this line of research would fit well into the data-driven component of our algorithm described in Sec. 4.2. Similarly, several papers try to detect parts of the body which they assemble into a complete form, termed *pictorial structures*. Although operating on high quality point clouds obtained from laser scans instead of video, Rodgers *et al*. [13] take a similar approach that uses discriminative methods to populate the domains of discrete variables in a Bayesian network. In the multi-view vision setting, Sigal *et al*. [15] apply a similar strategy, but use nonparametric belief propagation for their inference method. Our approach differs in that we perform inference in the continuous domain and that we are working on temporal data with a noisier sensor. Our approach to body part detection employed in this work is described in [11].

Much work has also focused on sampling-based methods [5, 3, 16], including partitioned sampling, which updates subsets of parameters, and hierarchical sampling, which starts at the top of the kinematic chain and proceeds downwards. In our approach, we adopt the idea of hierarchical search along the kinematic chain, but replace random sampling with deterministic sampling because it is allows increased efficiency through pre-computation.

Recently there have been several attempts to track people using a time-of-flight (TOF) camera. Grest *et al*. [7] apply non-linear least squares to edge maps, that are associated from frame to frame. Knoop *et al*. [8] use a stereo camera and a TOF camera to fit a cylindrical 3D body model to the data via the iterative closest point (ICP) algorithm. Both papers focus on tracking of the upper body only. Due to their local nature, both algorithms are susceptible to losing track when the motion is too fast. Recently, Zhu *et al*. [18] have proposed an algorithm for upper-body tracking from one

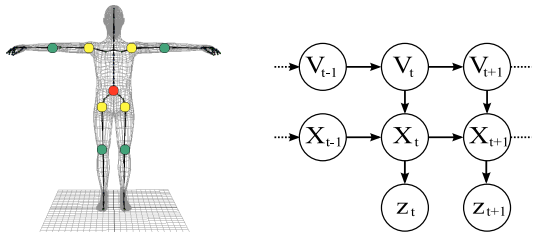


Figure 1. Left: The human body is modeled by a kinematic chain and a 3D surface mesh. Right: The dynamic Bayesian network modeling the poses X_t of the tracked person and the recorded range measurements z_t.

TOF camera. Their algorithm is based on hand-engineered heuristics for detecting joints of the upper body and is optimized for upper front-facing poses. These algorithms do not include any reinitialization component and operate through local optimization initialized from the previous frame.

The related problem of tracking using one or many video cameras has also received much attention. The classic work by Bregler *et al*. [4] tracks a person from a single camera using optical flow to obtain frame-to-frame correspondences, which are used to calculate motion derivatives, which are propagated up the kinematic chain. By assumption, their approach is limited to motions parallel to the image plane, and it is also susceptible to losing track. However, the central idea of propagating information from the image up the kinematic chain is one that we exploit in our algorithm, although we use the unscented transform to achieve higher quality linearization.

There is a growing trend in papers that use programmable graphics hardware (GPUs) to implement computer vision algorithms [10]. The parallel nature of computation on a GPU as well as their optimization for operating on images leads us to believe, that they are an ideal platform for computer vision algorithms, especially when real-time performance is paramount. We exploit GPUs to perform large numbers of likelihood evaluations efficiently.

3. Probabilistic Model

The objective is to track an articulated body over time using a stream of monocular depth images. We first define a probabilistic model for the variables of interest in this section and then describe how to efficiently perform inference in Sec. 4.

We model the body as a collection of 15 rigid body parts, which are constrained in space according to a tree-shaped kinematic chain (skeleton), see the left diagram in Fig. 1. A kinematic chain is a directed acyclic graph (DAG) with well-defined parent-child relations. The surface geometry of the model is represented via a closed triangle mesh, which deforms with the underlying kinematic chain

by means of a vertex skinning approach [10]. We denote the configuration of the body by $X_t = \{X_t^i\}_{i=1}^N$, where each i indexes a uniquely defined body part in the chain. The transformations X_i can be represented in various ways, such as using homogeneous matrices or in vector/quaternion form. Independent from the choice of representation, X^i denotes the position and orientation of a specific body part *relative* to its parent part. The chain is "anchored" to the world at the pelvis X_t^1 (which does not have a parent in the kinematic tree). In our model, we allow the pelvis to freely rotate and translate. The remaining body parts are connected to the their parent via a ball joint, which allows them to rotate in any direction, but not to translate. We obtain the absolute orientation $W^i(X)$ of a body part i by multiplying the transformations of its ancestors in the kinematic chain, $W^i(X) = X^1 \cdots X^{\text{parent}(i)} X^i$.

In order to determine the most likely state at any time, we must define a probabilistic model. The state at time t is the pose X_t and its first discrete-time derivative V_t. The measurement is the range scan z_t. We model our system as a dynamic Bayesian network (DBN), see the right diagram in Fig. 1, which encodes the Markov independence assumption that X_t and V_t are independent of $z_1, \ldots, z_{t-1}$ given X_{t-1} and V_{t-1}.

This DBN requires the specification of the conditional probabilities $P(V_t|V_{t-1})$, $P(X_t|X_{t-1}, V_t)$ and the measurement model $P(z_t|X_t)$. We make the assumption that the accelerations in our system are drawn from a Gaussian distribution with zero mean[1]. Thus, $V_t|V_{t-1} \sim \mathcal{N}(V_{t-1}, \Sigma)$ with the covariance matrix Σ being diagonal. We note that since X is a list of *relative* transformations, the velocities are also defined relatively. That is, if k is the index of the knee, V_t^k encodes the change in the angle between the shin and thigh at frame t. The covariance Σ was specified by hand following bio-mechanical principles and the known video frame rate, although it could easily be set by an automated procedure using the many available human motion data sets.

We define $P(X_t|V_t, X_{t-1})$ to be a *deterministic* CPD (conditional probability distribution), that applies the transformations in V_t to those in X_{t-1}. Formally, $X_t^i = V_t^i X_{t-1}^i$ with probability 1.

The measurement model defines the distribution on the measured range data given the state of the system. The measured range scan is denoted by $z = \{z^k\}_{k=1}^M$ where z^k gives the measured depth of the pixel at coordinate k. An example scan is shown in Fig. 6. We assume the conditional independence of each pixel given the state and scene geometry m,

$$P(z_t|X_t, m) = \prod_k P(z_t^k|X_t, m) \; .$$

[1]Note that it is common in the tracking literature to assume random accelerations rather than random velocities

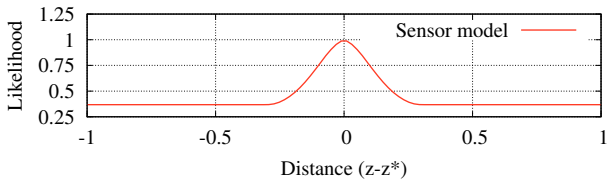


Figure 2. Our sensor model for individual range measurements approximates the mixture of a Gaussian (measurement noise) and a uniform distribution (outliers and mis-associations).

To generate the z^k for a specific body configuration X and model m, we transform the vertices corresponding to each part i by its corresponding transformation matrix $W^i(X)$. We then cast a ray from the focal point through pixel k and calculate the distance to the first surface it hits, which we denote z^{k*}. Note, that this is basically a ray-tracing operation common place in the computer graphics rendering literature.

Given the ideal depth value, we can then apply a noise model for the given sensor. Noise models for time-of-flight sensors have been heavily explored in the robotics literature. The standard approach is to explicitly model the different types of noise that are can occur. In principle, one can enumerate the effects, such as Gaussian noise of the sensor, the probability of a max range reading, outliers, and others. We approximate such a CPD using the function shown in Fig. 2. Let us define $smoothstep(x) = (\min(x, 1))^2(3 - 2\min(x, 1))$. We define

$$p(|z^k - z^{k*}|) \propto \exp(-smoothstep(|z^k - z^{k*}|)) \; . \quad (1)$$

We chose this function because it approximates a Gaussian mixed with a uniform distribution and it is a built-in function in the GPU shading language.

Our goal is to determine the most likely states $\hat{X}_t$ and $\hat{V}_t$ at time t given the MAP assignments of the previous frame, that is, $\hat{X}_{t-1}$ and $\hat{V}_{t-1}$. At each frame, we face the difficult, high dimensional optimization problem $\langle \hat{X}_t, \hat{V}_t \rangle =$

$$\text{argmax}_{X_t, V_t} \log P(z_t|X_t, V_t) + \log P(X_t, V_t|\hat{X}_{t-1}, \hat{V}_{t-1}) \; ,$$

for which we describe an efficient solution in Sec. 4.

As a central component of our optimization problem, the previously described measurement model $P(z_t|X_t, V_t)$ is inadequate due to its sensitivity to incorrect models m and to slight changes in the state X. Parts of the model that violate their silhouette in the measured image will be penalized heavily. For instance, slightly translating an object will result in all pixels at the edges evaluating an incorrect depth value, which would be penalized heavily. The sensor model partially accounts for this over-sensitivity through the use of a heavy-tailed distribution.

In the literature, it has frequently been observed that the true likelihood is often ill-suited for optimization, and sur-

rogate likelihoods are often used [5]. We develop a function that is more robust to the mis-association that occurs during optimization. Let us rewrite this likelihood $l(X)$ in terms of $z(X)$, the depth image obtained through ray-casting applied to X. $l(X) = \sum_k \log P(z^k|z(X)) = \sum_k \log P(z^k|z^k(X))$. We construct an alternate smoother likelihood

$$l_{smooth}(X) = \sum_k \max_j \log P(z^k|z^j(X)) + \lambda(j,k)$$

parametrized by a penalty function λ. Here, $\lambda(j,k)$ represents a cost for choosing a different pixel index than the one predicted by ray casting. In our case we define $\lambda(j,k) = -\infty$, if j is not an immediate pixel neighbor of k, $\lambda(j,k) = 0$, if $j = k$ and set $\lambda(j,k)$ to a constant in all other cases. We chose the constant -0.05 according to on the following reasoning: Given our sensor's field of view, the subject will be approximately two meters away in order to fit completely. At that distance, moving to a neighboring pixel results in a Euclidean distance of a approximately 0.05 meters perpendicular to the direction the camera is facing. Near the minimum, the log likelihood of the noise model is approximately quadratic. λ can be chosen to smooth the likelihood further though this could reduce accuracy. Thus, the total penalty function approximates Euclidean distance for close matches.

This section defined the probabilistic state space and measurement model. Inference in this model is non-trivial due to the high-dimensional nature of the space of the kinematic configuration space X and the associated velocity space V. This is particularly challenging for our real-time tracking objective, where exhaustive inference is infeasible.

4. Inference

We now describe how to perform efficient MAP inference at each frame. We attack this problem in two ways: (1) A model-based component locally optimizes the likelihood function by hill-climbing (HC) and (2), a data-driven part processes the measurement z to reinitialize parts of the filter state when possible. For the latter component, we derive an approximate inference procedure termed *evidence propagation* (EP) to generate likely states which are then used to initialize the model-based algorithm.

4.1. Model-Based Hill Climbing Search (HC)

To locally optimize the likelihood, we apply a coarse-to-fine hill-climbing procedure. We start from the base of kinematic chain which includes the largest body parts, and proceed toward the limbs. For a single dimension i of the state space, we sample a grid of values about the mean of $p(V_t^i|V_{t-1}^i)$. For each sample of V_t, we deterministically generate the state X_t from $\hat{X}_{t-1}$. The likelihood of this state is evaluated, and the best one of the grid chosen. The procedure can then potentially be applied to a smaller interval about the value chosen at the coarser level. For example, to optimize the X axis of the pelvis, we might sample values between -0.5 to 0.5 at intervals of 0.05 meters. The benefit of such a procedure is that it is inherently parallel. We can send a batch of candidates to the GPU, which evaluates all of them and returns their costs. We chose a deterministic sampling strategy over stochastic sampling because it allows more pre-computation and therefore increases the speed of likelihood evaluation.

4.2. Evidence Propagation (EP)

A variety of effects can cause the model-based search to fail. One problem is that fast motion causes significant motion blur. Additionally, occlusion can cause the estimate of the state of hidden parts to drift. Additionally, the likelihood function has ridges that are difficult to navigate. We therefore propose a data-driven procedure that identifies promising locations for body parts in order to find likely poses.

The three steps in this procedure are (I) to identify possible body part locations from the current range image, (II) to update the body configuration X given possible correspondences between mesh vertices and part detections and (III) to determine the best subset of such correspondences.

4.2.1 Body Part Detection

We consider the five body parts *head*, *left hand*, *right hand*, *left foot* and *right foot*. The 3D world locations of these parts according to the current configuration X of the body are denoted as $\mathbf{p}_i$, $i \in \{1,\ldots,5\}$. Note that we represent these parts by single vertices on the surface mesh of the body, such that all $\mathbf{p}_i$ are deterministic functions of X. The data-driven detections of body parts—which we describe in the following—are denoted as $\tilde{\mathbf{p}}_j$, $j \in \{1,\ldots,J\}$, where $J \in \mathbb{N}$ can be an arbitrary number depending on the part detector. Actual body parts as well as the detections have a class assignment c_i, $\tilde{c}_j$ to $\{head, hand, foot\}$.

We obtain the body part detections using the algorithm described in [11]. These detections are produced by a two step procedure. In the first step extremal points on the recorded surface mesh are determined from the range measurement z_t to form a set of distinct interest points. Discriminatively trained classifiers are applied to patches centered on the points to determine to what body part class they belong to. If the classifier is sufficiently confident, the feature is reported as a detection (see [11] for details).

4.2.2 Probabilistic Inverse Kinematics

We now define a probabilistic model, visualized in Fig. 3, consisting of the variables V_t, X_t, V_{t-1}, X_{t-1} and $\tilde{\mathbf{p}}_j$. As-

suming a correspondence between body part i and detection j, we apply the observation model

$$\tilde{\mathbf{p}}_j \sim \mathcal{N}\left(\mathbf{p}_i(X), \Sigma_o\right) \ . \tag{2}$$

We would now like to calculate a MAP estimate of X_t and V_t conditioned on $\hat{X}_{t-1}$ and $\tilde{\mathbf{p}}_j$. This is difficult because the intermediate variable $\mathbf{p}_i$ is a heavily non-linear function of X. In order to compute $\mathbf{p}_i(X)$ we must determine the world coordinates $W(X)$, which includes the absolute orientation of each body part. Then we transform $\mathbf{p}_i$ from its location in the mesh to its final location in the world.

To tackle this problem, we observe that X_t is a deterministic function of V_t and $\hat{X}_{t-1}$. Therefore, we can rewrite $\mathbf{p}_i$ as a non-linear function $\mathbf{p}_i(V_t, \hat{X}_{t-1}, \hat{V}_{t-1})$. Our approach will be to linearize the function $\mathbf{p}_i$. Because the distribution $P(V_t|\hat{V}_{t-1})$ is a linear Gaussian, linearizing $\mathbf{p_i}$ results in a linear Gaussian network approximation. MAP inference in this model is easy, so we can determine an estimate of argmax $P(V_t|\hat{X}_{t-1}, \hat{V}_{t-1}, \tilde{\mathbf{p}}_j)$. We linearize about this estimate and repeat the procedure until convergence.

There are many ways to linearize $\mathbf{p_i}$. We apply the unscented transform which is used in the unscented Kalman filter in a similar situation. The basic approach is to compute sigma points from the prior distribution on V_t, apply the non-linear function to them, and then approximate the result with a linear Gaussian. We omit the mathematical details, but it can be shown that this method provides an estimate of the distribution that is more accurate than linearization through calculating an analytic Jacobian.

To summarize, given a known correspondence between a point in the image and a point in the mesh, we can perform approximate MAP inference using the algorithm just described. The algorithm is related to existing methods for inverse kinematics using non-linear least squares, except that it performs linearization using the unscented transform, and it admits prior distributions on the variables.

4.2.3 Data Association and Inference

We now give the entire algorithm for determining $\hat{X}_t$ from $\hat{X}_{t-1}$ and z_t. In this section, we only assume we are given a set of part detections and their estimated body part classes, $\{\tilde{\mathbf{p}}_j, \tilde{c}_j\}$. The algorithm begins with an initial guess X_t^{best}, set to $\hat{X}_{t-1}$, which is repeatedly improved by integrating part detections.

The algorithm begins by updating X^{best} with the estimate from produced by hill-climbing as described in Sec. 4.1. Part detections are then extracted from the measurement z_t. In theory, we have to decide for each detection whether it is spurious, and if not, which specific body part it is associated with, constrained by the body part class of the detection. This results in a large number of possible combinations. Considering each such combination, which requires performing hill-climbing, would be far too time consuming for a real-time system. We therefore prune detections that are near any part location in X^{best}. This perhaps unintuitive heuristic is based on the observation that discriminative detections are needed most when the hill climbing approach has lost track. When such a loss has occurred, it is hoped the the discriminative algorithm will detect a part far from the location of any part in X^{best}. When the hill climbing algorithm is doing well, the part detections will be near their location in X^{best} and therefore can be ignored. We prune detections near *any* part in X^{best} because the part classifiers can often confuse classes. The next step is to expand the detections into a set of concrete correspondences. A candidate correspondence $\{(\mathbf{p}_i, \tilde{\mathbf{p}}_j)\}$ is created for each body part to all detections with a matching class, that is $c_i = \tilde{c}_j$. For instance, a correspondence is created for the right hand to all hand detections.

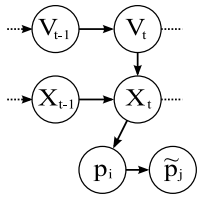


Figure 3. Dynamical model for the integration of body part detections .

At this point, we have a concrete list of possible correspondences from which we must choose a subset. We approach this problem in a greedy fashion. We iterate through each possible correspondence, and apply evidence propagation, initialized from X^{best} to find a new posterior mode X' which incorporates the current correspondence only. EP

1. Update X^{best} by local hill-climbing on the likelihood
2. Extract part detections from z_t
3. Prune hypotheses that are already explained
4. Produce set of correspondences $\{(\mathbf{p}_i, \tilde{\mathbf{p}}_j)\}$ by expanding hypotheses
5. Loop $i = 1$ to N
   (a) Let X' be the posterior mode of evidence propagation initialized from X^{best} conditioned on c^i
   (b) Update X' by local hill-climbing on likelihood
   (c) if likelihood of $X' > X^{\text{best}}$, set X^{best} to X_c

Figure 4. Tracking Algorithm. HC + EP

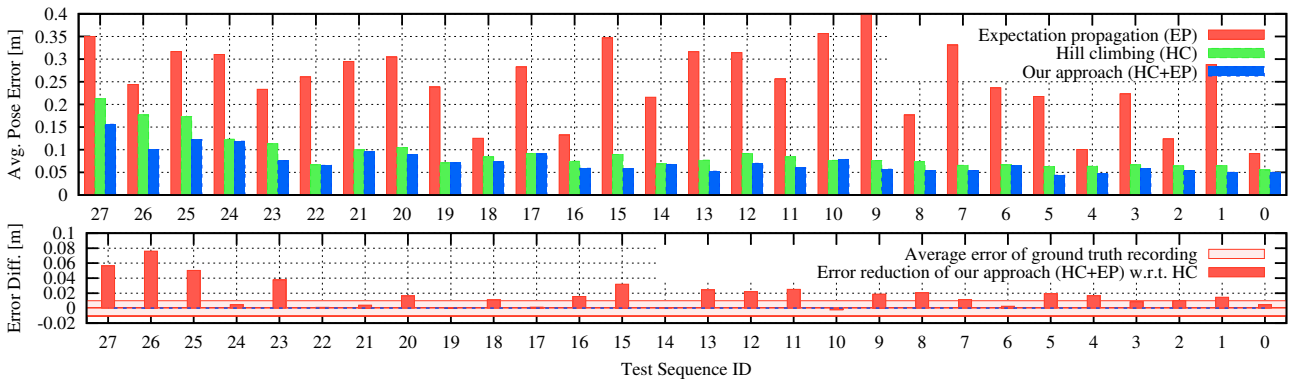


Figure 5. Tracking results on real-world test sequences, sorted from most complex (left) to least complex (right).

thus allows us to make a big jump in the state space. We then restart local hill-climbing from X' to refine it. If the final likelihood is better than X^{best}, X^{best} is replaced with the pose found. When this occurs, the candidate correspondence is considered to be accepted. The only effect of this on the subsequent iterations of the algorithm is through its update of X^{best}. The correspondence is *not* incorporated during subsequent states of EP, so that subsequent, possibly better, correspondences can override earlier ones. We have observed that this algorithm successfully rejects incorrect correspondences because the state resulting from their incorporation will be penalized by the likelihood function.

5. GPU-Accelerated Implementation

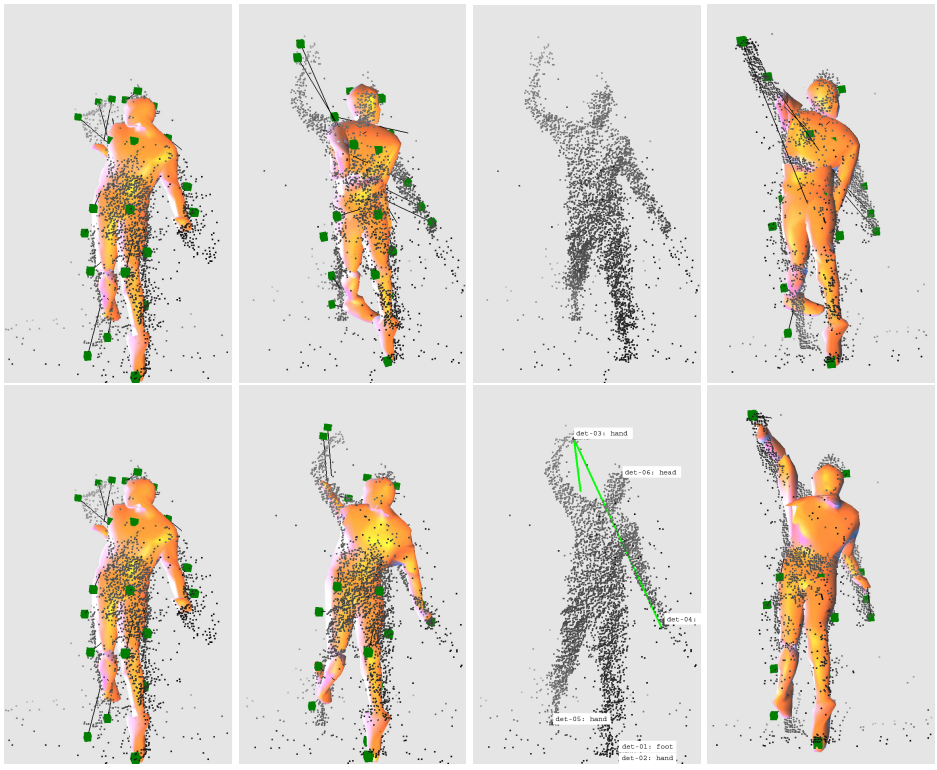
Several technical details enable the efficient evaluation of more than 50 000 candidates per second on the GPU. The main considerations are to maximize parallelism and to minimize uploading and downloading large amounts of data. The mesh is initially simplified using quadratic edge decimation and uploaded to the GPU along with the part assignments for each vertex. We use a custom skinning vertex shader that allows us to simply upload the transformation matrices for the entire configuration in order to render the entire body, without transferring any other data. Another shader calculates the measured ray length (since the Z-buffer has limited precision) and implements the actual cost function as described above. Because issuing operations to the GPU involves a certain latency, we process a batch of candidates simultaneously. We render a grid of candidates tiled on a single texture. This texture is then compared against the observation in parallel for each tile. By ensuring that the dimensions of the grid are a power of two, we can exploit the built-in functionality of the GPU to generate a mip-mapped texture. A mip-mapped texture is one which contains versions of itself at progressively lower resolutions, each calculated by averaging pixels at higher resolutions. By reading a particular mip-map level, we can directly read out the average costs of each candidate, one per pixel, which is the minimum amount of data we can transfer back to the CPU.

6. Experiments

The described algorithm was fully implemented in `C++` and evaluated on real sequences. The goal of this experimental evaluation is to show that

- our proposed system is able to estimate the pose and configuration of a human over time using only a stream of depth images,
- proposing candidates using EP on detected body parts significantly improves performance over just doing local hill-climbing,
- the smoothed energy function (Eq. 1) outperforms the typically used pixel-wise energy function, and
- the system runs close to real-time.

To this end, we have created a sophisticated test data set, that allows quantitative analysis of the tracking performance. The data set, which is available openly as a benchmark along with our results [6], consists of 28 real-world depth-image sequences of varying complexity—ranging from short sequences with single-limb motions to longer sequences including fast kicks, swings, self-occlusions and full-body rotations. Our definition of complexity, though subjective, increases with the length of the sequence, amount of occlusion, speed of motion, number of body parts moving simultaneously, and rotation about the vertical axis. The depth image stream is collected using a Swissranger SR4000 Time-of-Flight camera, which was set to record full-frame infrared intensity images and depth at 25 fps and a resolution of 176×144 pixels, which we subsample to 128×128 pixels in order to take advantage of the GPU more effectively. In addition to the stream of depth images, we recorded the locations of 3D markers attached to the subject's body using a commercial active marker motion



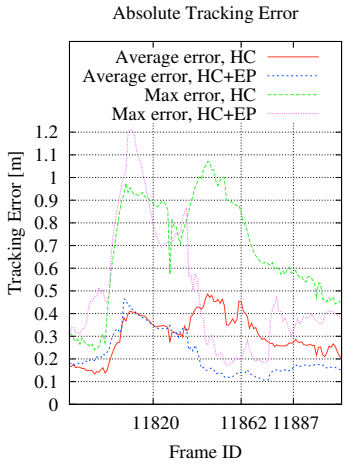


Figure 6. A typical situation in which data-driven evidence is crucial for tracking success (excerpt from Seq. 27). Left: Three exemplary frames from the *Tennis* sequence. *Model-Based search* (top row) loses track of the tennis swing, since the arm was occluded. Our combined tracker that integrates bottom-up evidence about body parts (bottom row) is able to recapture the fast moving arm. The right diagram shows the same situation in terms of actual tracking error (see text).

capture system. These measured marker locations serve as the ground truth in our error metric. Concretely, we consider the evaluation metrics

$$\epsilon_{\text{avg}} = \sum_{i=1}^{M} \frac{||\mathbf{m}_i - \tilde{\mathbf{m}}_i||}{M} \quad \text{and} \quad \epsilon_{\text{max}} = \max_i ||\mathbf{m}_i - \tilde{\mathbf{m}}_i|| \,,$$

where M is the number of visible motion-capture markers, $\mathbf{m}_i$ are the *true* 3D locations and $\tilde{\mathbf{m}}_i$ are the corresponding 3D locations on the estimated surface mesh of the tracked person. Through visual inspection, we found that individual marker errors $||\mathbf{m}_i - \tilde{\mathbf{m}}_i||$ of 0.1m or lower can be interpreted as perfectly tracked markers, since this corresponds to the approximate accuracy of the recorded *ground truth* data. On the other hand, marker errors of 0.3m or larger can be interpreted as tracking failures.

In order to evaluate the effectiveness of the combination of model-based and data-driven approaches, we show the results of two algorithms in addition to our proposed algorithm on this extensive dataset. The algorithm labeled EP consists of our overall algorithm with local hill-climbing removed. It simply proposes modes determined by EP, and keeps the one with highest likelihood. The algorithm labeled HC consists of just the model-based hill-climbing algorithm alone. We note that the mesh model for all algorithms was provided semi-automatically using a Cyberware Laser Scanner. The joint locations were determined using the SCAPE algorithm [2].

Figure 5 shows numerical results on our data set for all three algorithms. In the top panel, we show the errors of all algorithms. The results show that the data-driven method in isolation (EP) performs far worse than the other two approaches. Because of the high error of the data-driven algorithm alone, in the remaining plots we only consider the model-based algorithm and the combined algorithm. In the bottom panel, we visualize the difference in error between the combined approach (HC+EP) and the model-based approach (HC). In all the sequences, the combined approach performs best or equally well. The remaining error modes in our combined approach occur when both the local hill-climbing as well as the body part detectors fail. However, in several sequences the algorithm is able to recover even after longer periods of high tracking error.

It is interesting to note that on the harder sequences (left side), the difference in performance is more pronounced.

To analyze this in more detail, we consider a challenging excerpt from Sequence 27—a fast, partially occluded tennis swing as visualized in the left panel in Fig. 6. The right panel in this figure shows the trace of the error-metrics comparing the model-based and combined algorithm on the excerpt. In the beginning, the swinging arm becomes completely occluded, followed by moving swiftly forward. The ϵ_{max} measure ("Max error" in the diagram) reveals that once the hill-climbing tracker loses track, it never recovers. Our combined approach is able to again find the mode and continue. The left panel in Fig. 6 illustrates this using three tracked frames from the sequence. The top row illustrates that the model-based algorithm completely loses the arms, whereas our algorithm is able to find the arm after an occlusion and catch the trailing edge of the *Tennis* serve. The figure also illustrates through green lines from the detected body parts the associations that the algorithm considered, and how this enables it to use evidence propagation to pull itself back on track.

In terms of efficiency, both algorithms are close, though the hill-climbing approach is more efficient. The model-based algorithm ran at about 6 frames per second, whereas the combined algorithm ran at about 4-6 frames per second depending on how often evidence is integrated.

Finally, we also compared the effects of likelihood smoothing on the performance of the algorithms. We found that the smooth likelihood improved the performance in terms of average error across all sequences and frames by about 10 percent for the model based algorithm and 18 percent for the combined algorithm. The fact that it helped the combined approach more may be a result of the fact that the reinitialization is not always close enough to regain track with the non-smooth likelihood function.

7. Conclusions and Future Work

The ambitious goal of accurate real-time tracking of humans and other articulated bodies is one that has enticed researchers for many years due to the large number of useful applications. With the hybrid, GPU-accelerated filtering approach introduced in this paper, we believe to have made a large step forward, but there remain more challenges to overcome. Some examples include cluttered scenes, tracking more than one person at a time, improving the speed further, and fully automatic model initialization. Moreover, it would be interesting to integrate traditional imaging modalities such as color cameras or to apply the developed technology to stereo vision setups.

Acknowledgments

This work was supported by NSF grant number ISS 0917151, MURI contract N000140710747, and the Boeing company. We thank NVIDIA for donating state-of-the-art graphics hardware for this project.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ACM Trans. on Graphics*, 24(3):408–416, 2005.

[3] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 1998.

[5] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2000.

[6] V. Ganapathi and C. Plagemann. Project website and data sets: http://ai.stanford.edu/∼varung/cvpr10, March 2010.

[7] D. Grest, V. Krüger, and R. Koch. Single view motion tracking by depth and silhouette information. In *Scandinavian Conference on Image Analysis (SCIA)*, 2007.

[8] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *IEEE Conf. on Robotics & Automation (ICRA)*, 2006.

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

[10] M. Pharr and R. Fernando. GPU Gems 2: Programming techniques for high-performance graphics and general-purpose computation. 2005.

[11] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *IEEE Int. Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, 2010.

[12] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding (CVIU)*, 108(1):4–18, 2007.

[13] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller. Object pose detection in range scan data. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[14] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *In European Conference on Computer Vision (ECCV)*, volume 4, pages 700–714, 2002.

[15] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[16] C. Sminchisescu and B. Triggs. Hyperdynamics importance sampling. In *European Conference on Computer Vision (ECCV)*, 2002.

[17] Y. Sun, M. Bray, A. Thayananthan, B. Yuan, and P. H. S. Torr. Regression-based human motion capture from voxel data. In *British Machine Vision Conf. (BMVC)*, 2006.

[18] Y. Zhu, B. Dariush, and K. Fujimura. Controlled human pose estimation from depth image streams. *Proc. CVPR Workshop on TOF Computer Vision*, June 2008.