Tracking Loose-limbed People

Leonid Sigal* **Sidharth Bhatia*** **Stefan Roth*** **Michael J. Black *** **Michael Isard†**

*Department of Computer Science, Brown University Providence, RI 02912
†Microsoft Research Silicon Valley, Mountain View, CA 94043
{ls,sibhatia,roth,black}@cs.brown.edu, misard@microsoft.com

Abstract

We pose the problem of 3D human tracking as one of inference in a graphical model. Unlike traditional kinematic tree representations, our model of the body is a collection of loosely-connected limbs. Conditional probabilities relating the 3D pose of connected limbs are learned from motion-captured training data. Similarly, we learn probabilistic models for the temporal evolution of each limb (forward and backward in time). Human pose and motion estimation is then solved with non-parametric belief propagation using a variation of particle filtering that can be applied over a general loopy graph. The loose-limbed model and decentralized graph structure facilitate the use of low-level visual cues. We adopt simple limb and head detectors to provide "bottom-up" information that is incorporated into the inference process at every time-step; these detectors permit automatic initialization and aid recovery from transient tracking failures. We illustrate the method by automatically tracking a walking person in video imagery using four calibrated cameras. Our experimental apparatus includes a marker-based motion capture system aligned with the coordinate frame of the calibrated cameras with which we quantitatively evaluate the accuracy of our 3D person tracker.

1 Introduction

We present a fully automatic method for tracking human bodies in 3D. Initialization and failure recovery are facilitated by the use of a loose-limbed body model [22] in which limbs are connected via learned probabilistic constraints. The tracking problem is formulated as one of inference in a graphical model and belief propagation is used to estimate the pose of the body at each time-step. Each node in the graphical model represents the 3D position and orientation of a limb (Figure 1). Directed edges between nodes represent statistical dependencies and these constraints between limbs are used to form messages that are sent to neighboring nodes in space and time. Additionally, each node has an associated likelihood defined over a rich set of image cues using a learned Gibbs model [19, 28]. The combination of highly non-Gaussian likelihoods and a six-dimensional continuous parameter space (3D position and orientation) for each limb makes standard belief propagation algorithms infeasible. Consequently we exploit a form of non-parametric belief propagation that uses a variation of particle filtering and can be applied over a loopy graph [13, 25].

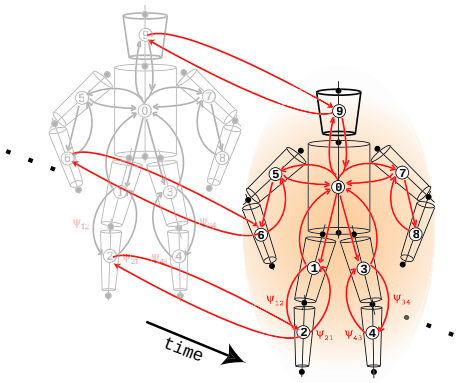


Figure 1: *Graphical model for a person.* Nodes represent limbs and arrows represent conditional dependencies between limbs. For clarity only a few temporal dependencies have been shown: in our model each part i at time t is connected by an edge to part i at times $t-1$ and $t+1$.

There are a number of significant advantages to this approach as compared to traditional methods for tracking human motion. Most current techniques model the body as a kinematic tree in 2D [14], 2.5D [17], or 3D [2, 5, 21, 23] leading to a high-dimensional parameter space (25–50 dimensions is not uncommon). Searching such a space directly is impractical and so current methods typically rely on manual initialization of the body model. Additionally, they often exploit strong priors characterizing the types of motions present. When such algorithms lose track (as they always do), the dimensionality of the state space makes it difficult to recover.

While the full body pose is hard to recover directly, the location and pose of individual limbs is much easer to compute. Many good head detectors exist and limb detectors have been used for some time (e.g. [18]). The approach we take here can use bottom up information from feature

detectors of any kind and consequently should be able to cope with a rich variety of input images. In our implementation we exploit background/foreground separation for computational simplicity but part detectors which perform well against arbitrary backgrounds are becoming standard [18, 26].

With a kinematic tree model, exploiting this partial, "bottom-up" information is challenging. If one could definitively detect the body parts then inverse kinematics could be used to solve for the body pose, but in practice low-level part detectors are noisy and unreliable. The use of a loose-limbed model and belief propagation provides an elegant framework for incorporating information from part detectors. Because the inference algorithm operates over a general graph rather than a forward chain as in traditional particle filter trackers, it is also straightforward to perform forward–backward smoothing of the limb trajectories without modifying the basic approach.

A loose-limbed body model requires a specification of the probabilistic relationships between joints at a given time instant and over time. We represent these non-Gaussian relationships using mixture models that are learned from a database of motion capture sequences. It is worth noting that these models encode information about joint limits and represent a relatively weak prior over human poses, which is appropriate for tracking varied human motions.

The model also requires an image likelihood measure for each limb. Using training data of known limb poses in images, we learn a novel likelihood model that captures the joint statistics of first and second derivative filter responses at multiple scales. We formulate and learn the likelihoods using a Gibbs model based on the maximum entropy principle [28].

We test the method by tracking a subject viewed from four calibrated cameras in an indoor environment with no special clothing. Quantitative evaluation is performed using a novel motion capture testbed that provides "ground truth" human motion from a commercial motion capture system that is synchronized with the video streams [1]. In particular we compare the accuracy of our method with that of a more standard kinematic tree body tracker using annealed particle filtering [5]. We find that the traditional approach looses track rapidly compared with the loose-limbed model when the image quality is poor.

Previous work

Disaggregated models are not new for finding or tracking articulated objects and date back at least to Fischler and Elschlager's pictorial structures [8]. Variations on this type of model have been recently applied by Burl *et al.* [1], Felzenszwalb and Huttenlocher [7], Coughlan and Ferreira [3], Ioffe and Forsyth [10, 11, 12] and Ramanan and Forsyth [18]. Ioffe and Forsyth [10, 11] first find body parts and then group them into figures in a bottom-up fashion. The approach exploits the fact that they have a discrete set of features that need to be assembled, but it prevents them from using rich likelihood information to "co-operate" with the body model when estimating the pose. Ramanan and Forsyth [18] propose an elegant estimation of appearance models jointly with the body's trajectory, but their inference algorithm relies on the fact that the 2D model has a relatively low-dimensional state-space for each body part. A similar approach to ours has been adopted in [27] for tracking a 2D human silhouette using a dynamic Markov network. A much simplified observation model was adopted in [27] and their system does not perform automatic initialization. They adopt a somewhat different inference algorithm and a comparison between the two methods merits future research.

In previous work [22] we presented the general loose-limbed body model and belief propagation algorithm but only addressed the problem of human pose estimation at a single time instant; here we extend the inference method over time to perform visual tracking. In [22], the potential functions linking limbs were constructed manually whereas here they are learned from training data and are also extended in time. Here we also propose a multi-view eigen-method to implement bottom-up body part detectors and we exploit a learned Gibbs likelihood model [19].

2 Loose-limbed body model

Following the framework in [22] the body is represented by a graphical model in which each graph node corresponds to a body part (upper leg, torso, etc.). Each part has an associated configuration vector defining the part's position and orientation in 3-space. Placing each part in a global coordinate frame enables the part detectors to operate independently while the full body is assembled by inference over the graphical model. Edges in the graphical model correspond to position and angle relationships between adjacent body parts in space and time, as illustrated in Figure 1.

In order to describe the body parts in a graphical model, we assume the variables in a node are conditionally independent of those in non-neighboring nodes given the values of the node's neighbors. Each part/limb is modeled by a tapered cylinder having 5 fixed and 6 estimated parameters. The fixed parameters $\Phi_i = (l_i, w_i^p, w_i^d, o_i^p, o_i^d)$ correspond respectively to the part length, width at the proximal and distal ends and the offset of the proximal and distal joints along the axis of the limb as shown in Figure 2. The estimated parameters $\mathbf{X}_i^T = (\mathbf{x}_i^T, \Theta_i^T)$ represent the configuration of the part i in a global coordinate frame where $\mathbf{x}_i \in \mathbb{R}^3$ and $\Theta_i \in \mathrm{SO}(3)$ are the 3D position of the proximal joint

[1] Available at http://www.cs.brown.edu/research/vision/motioncapture/.

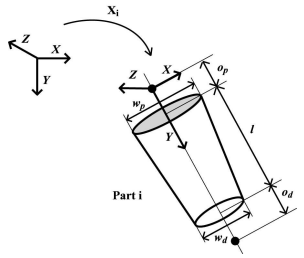


Figure 2: *Parameterization of part i.*

and the angular orientation of the part respectively. The rotations are represented by unit quaternions.

Each directed edge between parts i and j has an associated potential function $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ that encodes the compatibility between pairs of part configurations and intuitively can be thought of as the probability of configuration $\mathbf{X}_j$ of part j conditioned on the $\mathbf{X}_i$ of part i. The potential $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ is in general non-Gaussian and is approximated by a mixture of M_{ij} Gaussians:

$$\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) = \lambda^0 \mathcal{N}(\mathbf{X}_j; \mu_{ij}, \Lambda_{ij}) + (1-\lambda^0) \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}(\mathbf{X}_j; F_{ijm}(\mathbf{X}_i), G_{ijm}(\mathbf{X}_i)) \quad (1)$$

where λ^0 is a fixed outlier probability, μ_{ij} and Λ_{ij} are the mean and covariance of the Gaussian outlier process, and $F_{ijm}(\cdot)$ and $G_{ijm}(\cdot)$ are functions that return the mean and covariance matrix respectively of the m-th Gaussian mixture component. $\delta_{ijm} \geq 0$ is the relative weight of an individual component and $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$.

2.1 Learning limb conditionals

Given a ground truth parameter vector $\mathbf{X}_j$ for part j, we can construct the 3D object-to-world transform $M(\mathbf{X}_j)$ defining the pose of the limb j as

$$M(\mathbf{X}_j) = M(\mathbf{X}_i) M(\mathbf{X}_{ij}) \quad (2)$$

where $M(\mathbf{X}_i)$ is the pose of the neighboring spatial or temporal part, and $\mathbf{X}_{ij}$ encodes the position and orientation of part j in i's coordinate frame. We can approximate the potential compatibility functions:

$$\begin{aligned} \psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) &\approx p(\mathbf{X}_j | \mathbf{X}_i) \\ &= p(M(\mathbf{X}_j) | M(\mathbf{X}_i)) \\ &= p(M(\mathbf{X}_i) M(\mathbf{X}_{ij}) | M(\mathbf{X}_i)) \\ &\approx p(M(\mathbf{X}_{ij})) = p(\mathbf{X}_{ij}) \end{aligned} \quad (3)$$

by learning the distribution over $\mathbf{X}_{ij}$ to fix the $F_{ijm}(\cdot)$ and $G_{ijm}(\cdot)$ functions from (1). We model all spatial and temporal potentials using mixtures of Gaussians with

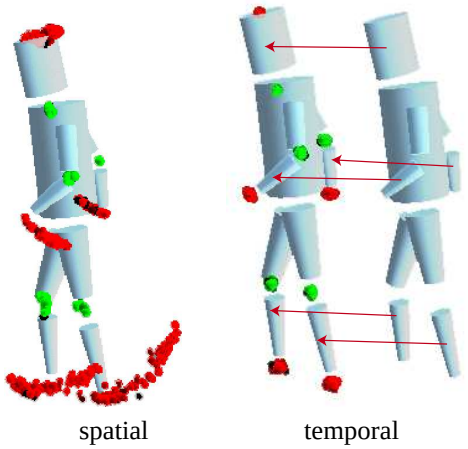


Figure 3: *Learned Spatial and Temporal Potentials.* Spatial and temporal potentials are illustrated by sampling from them. The potentials for the lower limbs (arms and legs) and head are shown. The spatial potentials show the distribution of limb positions and orientations conditioned on the neighboring limb. Green spheres indicate the joint position of a sample while the red spheres indicate the distal end of the limb for each sample. The spread of these samples illustrates the learned distribution encoded by the potentials. The temporal potentials shown are for the forward time direction.

$M_{ij} = 3$ components, and for example the first component $F_{ij1}(\mathbf{X}_j) = M^{-1}(M(\mathbf{X}_i) M(\hat{\mathbf{X}}_{ij1}))$ where $\hat{\mathbf{X}}_{ij1}$ is the mean of the first learned component of $p(\mathbf{X}_{ij})$. The distributions are learned using $S = 4928$ ground truth motion capture frames of walking data collected at 60 Hz.

We use a standard iterative Expectation-Maximization (EM) algorithm with K-means initialization for learning the Gaussian mixture model (GMM), however care must be taken to estimate the mean and covariance of the quaternion orientations. Given a set of S unit quaternions $Q = \{q_0, q_1, \ldots, q_S\}$ where $q_i = [q_{i,x}, q_{i,y}, q_{i,z}, q_{i,w}]$, we extend the approximation presented in [4] to approximate the mean of Q by

$$E_q[Q] \approx \frac{1}{S} \left[\sum_{i=0}^{S} \frac{q_{i,x}}{q_{i,w}}, \sum_{i=0}^{S} \frac{q_{i,y}}{q_{i,w}}, \sum_{i=0}^{S} \frac{q_{i,z}}{q_{i,w}}, S \right]. \quad (4)$$

This approximation suffers from a singularity when any $q_{i,w} \to 0$. To mitigate the effect of this singularity we compute $E_q[Q]$ in a normalized quaternion space constructed to minimize $\max_{q_i \in Q}(q_i \cdot [0,0,0,1]^T)$. Similarly we approximate the covariance of Q by computing the deviations $\check{q}_i = q_i^{-1} * E_q[Q]$ and set $Cov_q[Q] \approx$

$$\frac{1}{S} \sum_{i=0}^{S} \left(\left[\frac{\check{q}_{i,x}}{\check{q}_{i,w}}, \frac{\check{q}_{i,y}}{\check{q}_{i,w}}, \frac{\check{q}_{i,z}}{\check{q}_{i,w}} \right]^T \cdot \left[\frac{\check{q}_{i,x}}{\check{q}_{i,w}}, \frac{\check{q}_{i,y}}{\check{q}_{i,w}}, \frac{\check{q}_{i,z}}{\check{q}_{i,w}} \right] \right). \quad (5)$$

While our learning algorithm is general enough to learn

distributions that have couplings between positional and rotational components of the state space, resulting in block-diagonal covariance matrices, for computational purposes we restrict ourselves to the diagonal-covariance distributions. For sampling and evaluating the probability of the GMM we refer the reader to [4].

Figure 3 shows a few of the learned potential distributions. Samples are shown from several limb-to-limb potentials. For example, the lower leg distribution is shown conditioned on the pose of the upper leg. The proximal end of the shin (green circle) is predicted with high confidence given the thigh location, but there is a wide distribution over possible ankle locations, as expected.

3 Image Likelihoods

The inference algorithm outlined in the next section combines the body model described above with a probabilistic image likelihood model. We define $\phi_i(\mathbf{X}_i)$ to be the likelihood of observing the image measurements conditioned on the pose of limb i. Ideally this model would be robust to partial occlusions, the variability of image statistics across different input sequences, and variability among subjects.

To that end, we combine a variety of cues including multi-scale edge and ridge filters following [20]. However, we explicitly model the conditional dependencies between the various filter responses by learning the joint density using a Gibbs model [19, 28] of the form

$$p(\mathbf{f} \mid \mathbf{X}_i) \propto \exp\left(-\sum_i \langle \lambda^{(i)}, \xi^{(i)}(\mathbf{f}, \mathbf{X}_i)\rangle\right),$$

where $\mathbf{f}$ represents a vector of filter responses, the $\xi^{(i)}$ are functions selecting various marginals and the $\lambda^{(i)}$ are their learned weights. Since this likelihood is trained also in situations where the limb is partially or fully occluded, it is fairly robust to these conditions.

Separate foreground models are learned for the appearance of each limb. In addition, a pooled background model is learned from non-limb patches sampled from the training images. These are combined into a limb likelihood by taking the likelihood ratio [19]. These likelihood ratios along with background subtraction information are then combined across views, assuming independence of the views conditioned on the limb position.

4 Non-parametric BP

Inferring the body pose in our framework is defined as estimating belief in the graphical model. To cope with the continuous 6D parameter space of each limb, the non-Gaussian

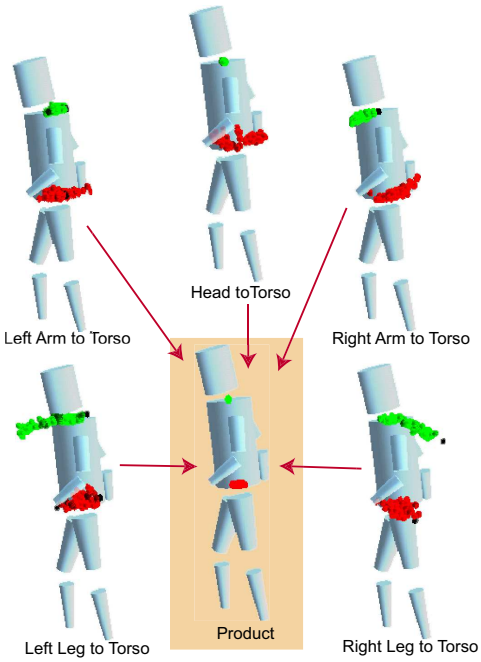


Figure 4: *Message Product.* The head, upper arms, and upper legs send messages to the torso. Samples from these messages are illustrated by showing the predicted torso location with green balls. The distribution over the orientation of the torso is illustrated by showing a red ball at the distal end of the torso for each sample. While any single message respresents uncertain information about the torso pose, the product of these messages tightly constrains the torso position and orientation.

conditionals between nodes, and the non-Gaussian likelihood, we use a form of non-parametric belief propagation [13, 25]. The approach is a generalization of particle filtering [6] which allows inference over arbitrary graphs rather than a simple chain. In this generalization the "message" used in standard belief propagation is approximated with a particle set, and the conditional distribution used in standard particle filtering is replaced by a product of incoming message sets. The two formulations [13, 25] have different strengths; we adopt the PAMPAS algorithm [13] because it maps better to our models where the potentials are small mixtures of Gaussians and the likelihoods are simple to evaluate up to an unknown normalization. NBP [25] is more suitable for applications with complex potential functions. We use the Gibbs sampler from [25] to evaluate products of $D > 2$ messages.

The message passing framework is illustrated in Figure 4 where the head, upper arms and upper legs all send messages to the torso. These messages are distributions that are represented by a set of weighted samples as in particle filtering. Belief propagation requires forming the product of these incoming messages. As Figure 4 shows, the individual limbs may not constrain the torso very precisely. The

product over all the incoming messages however produces a very tight distribution over the torso pose.

A message m_{ij} from node $i \rightarrow j$ is written

$$m_{ij}(\mathbf{X}_j) = \int \psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)\phi_i(\mathbf{X}_i) \prod_{k \in A_i \backslash j} m_{ki}(\mathbf{X}_i) \mathrm{d}\mathbf{X}_i,$$

where A_i is the set of neighbors of node i and $\phi_i(\mathbf{X}_i)$ is the local likelihood associated with node i. The message $m_{ij}(\mathbf{X}_j)$ can be approximated by importance sampling $N' = (N-1)/M_{ij}$ times from a proposal function $f(\mathbf{X}_i)$, and then doing importance correction. As discussed in [13] the N' samples may be stratified into groups with different proposal functions $f(\cdot)$, so some proportion $\lambda_B N'$ of samples come from the product of all incoming messages A_i into the node, $\lambda_I N'$ come from $A_i \backslash j$ (i.e. A_i excluding j) and $\lambda_S N'$ from an importance function $Q_t(\mathbf{X}_i)$ which is in general a function of the time-step t — we use a limb proposal distribution based on local image measurements desribed in Section 5. For algorithmic details see [13].

The basic algorithm leaves open the question of what proportions to use in the stratified sampler and what order to update the messages in. We use a 3-frame windowed smoothing algorithm for our tracking results where the estimates at time t are based on observations at times $(t-1, t, t+1)$. There are 30 nodes in the graph (10 body parts at each time-step) and 94 edges (18 between adjacent body parts within each time-step and two between each part at each consecutive time-step). All the messages are updated in batch, and this batch update takes place 4 times in 4 belief-propagation iterations. For the first iteration, the proportion of limb proposal samples is $\lambda_S = 0.50$ and this proportion halves for each subsequent iteration. In each iteration the proportion of samples taken from the belief estimate is $\lambda_B = 1/2(1-\lambda_S)$ and the remainder λ_I are taken from the incoming message product.

The algorithm must sample, evaluate, and take products over Gaussian distributions defined over $SO(3)$ and represented in terms of unit quaternions. We adopt the approximation given in [4] for dealing with rotational distributions by treating the quaternions locally linearly in $\mathbb{R}^4$ — this approximation is only valid for kernels with small rotational covariance and can in principle suffer from singularities if product distributions are widely distributed about the sphere, but we have not encountered problems in practice.

5 Bottom-up Part Detectors

Occlusion of body parts, changes in illumination, and a myriad of other situations may cause a person tracker to lose track of some, or all, parts of a body. We argue that reliable tracking requires bottom-up processes that constantly search for body parts and suggest their location and pose to the tracker; we call these "shouters" [2].

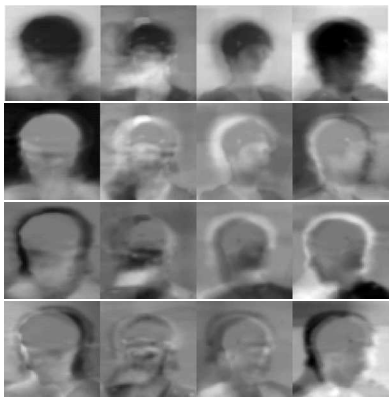


Figure 5: *Multi-view Eigenfeatures.* We learn the correlations between the projections of an object in our fixed cameras by concatenating the four views at each time-step into a single "multi-view" image vector. Top row: mean multi-view head. Next three rows: first three principal components.

One expects shouters to be noisy in that they will sometimes fail to detect parts or will find spurious parts. Furthermore they will probably not be able to differentiate between left and right arms. Both of these behaviours can be seen in Figure 6*a*. Even these noisy "guesses" provide valuable low-level cues, however, and our belief propagation framework is designed to incorporate this bottom-up information in a principled way. As described in Section 4 we use a stratified sampler for the messages to graph node i at time t that draws some samples from an importance function $Q_t(\mathbf{X}_i)$. This importance function is constructed by the node's shouter process, and draws samples from locations in pose space (3D location and orientation) near the detected body parts.

Multi-View Eigenspaces

There are many approaches to body-part detection in single or multiple images. We have implemented simple eigentemplate detectors for the head, upper arms, and lower legs; other shouters could be added as desired. Given calibrated training images with known body-part locations, we build a set of *multi-view* training images. Specifically, we construct a single training sample from the four camera views by concatenating the image regions of the part in each view. We perform PCA on these multi-view images as described in [16], keeping 9–40 principal components, depending on the detector, which describe approximately 80% of the variation in the training data. Figure 5 shows the first few principal components of our head detector model; each part detector is orientation independent.

[2] This term came from discussions with A. Jepson and D. Fleet.

Using the training data we construct a bounding box in 3-space where we expect each part to appear. We exploit the fact that the background is static, model it using a mixture model at each pixel, and perform standard foreground detection [24]. Selecting any camera view, we examine all foreground pixels within the projected bounding box region for the given body part. Each pixel defines a ray in 3D and we search along this ray for matches to our orientation-independent eigen-model, rejecting any location on the ray which does not project to a foreground pixel in every view. For each of the 10 most probable 3D locations we find the 5 closest multi-view matches from the training set and use their orientations to construct 50 candidate poses for the proposal mixture distribution $Q_t(\mathbf{X}_i)$.

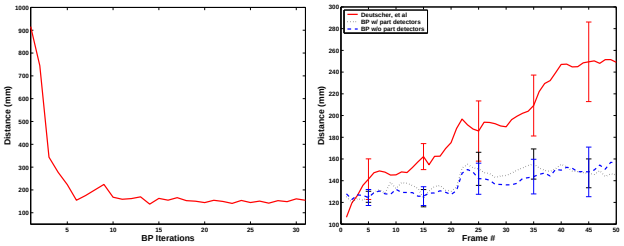


Figure 8: *Quantitative tracking evaluation. (left)* Automatic initialization error as a function of iterations of the belief propagation algorithm. *(right)* Tracking error over a 50-frame sequence. The dotted black and dashed blue lines show the error for the loose-limbed method with and without shouters respectively. The red solid line shows the error for the kinematic tree method [5].

6 Experiments and Evaluation

Figure 6*a* shows the automatic initialization of the 3D body model using bottom-up part detectors. Note that we use only detectors for the head, upper arms, and calfs and that these detectors are very inaccurate. While they give a rough 3D location and orientation for the limbs, they cannot differentiate left from right limbs reliably. Note also that the right calf was not detected. For body parts with no bottom-up detector, the initialization is random (Figure 6*b*). After several iterations of belief propagation, the algorithm "finds" the limbs and has a reasonable distribution over the limbs poses (Figure 6*c* and *d*). Figure 7 shows the results of tracking over 25 frames after automatic initialization.

There are no standard performance metrics for video-based 3D human tracking. Ramanan and Forsyth [18] report tracking success whenever there is *any* overlap between a limb and the ground truth; this seems overly generous. We propose a quantitative evaluation of accuracy based on the absolute distance of true and estimated marker locations on the limbs. We chose 15 markers corresponding roughly to the locations of the joints and "ends" of the limbs. For each limb we sample from the belief and compute the normalized likelihood of each sample to obtain a weight. These weights are then used to compute an expected absolute distance in mm for the markers associated with the limb. The expected deviations are then summed over all defined virtual markers to produce the final distance-based error measure.

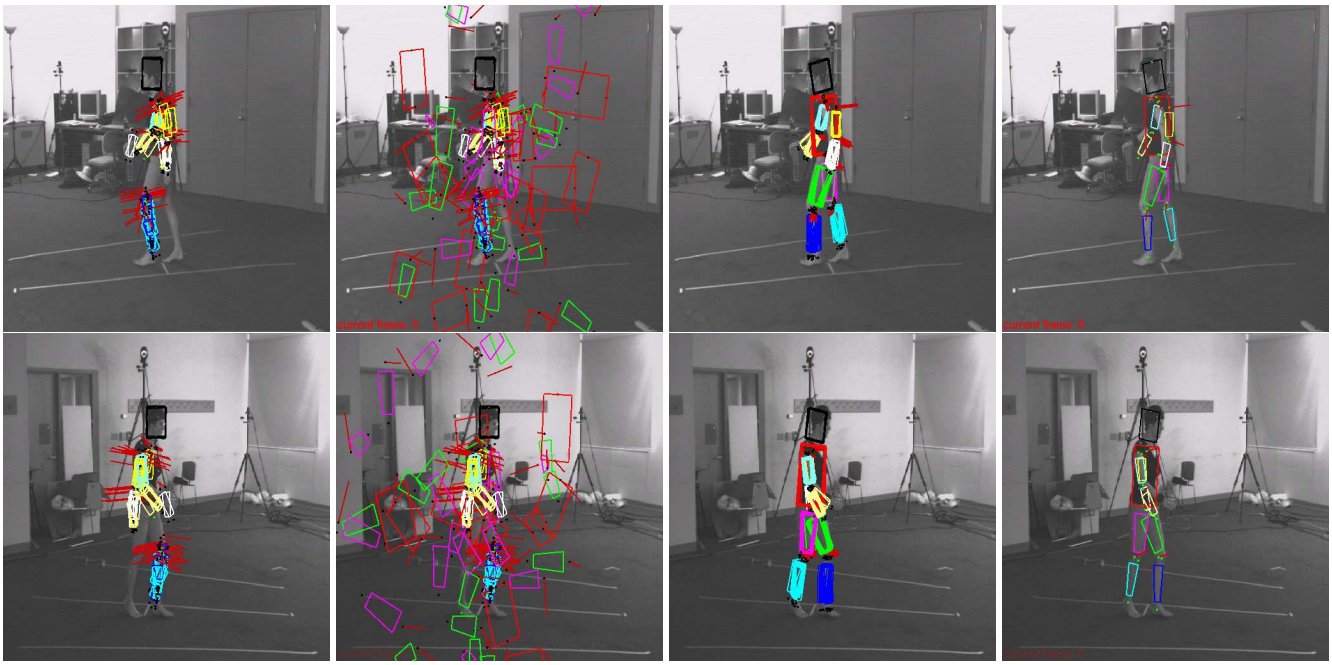
Figure 8 shows accuracy of the intialization (left) and tracking results (right). After a few iterations of belief propagation, the initialization error decreases and stays stable. We also observe that the error in the estimated pose increases only slightly over the tracking sequence.

To compare our method against the state of the art, we independently implemented a kinematic tree based tracker that uses annealed particle filtering [5]. This allows a quantitative performance comparison between the methods. Our implementation of [5] uses the same image likelihood as used with the loose-limbed model and both trackers are initialized with the same ground-truth pose obtained by the motion capture system. As suggested in [5] the annealing process uses 10 layers and 200 particles. The distance error was computed using the same markers but here we sampled from the particle set, computed the likelihood, and normalized to obtain the weights used in computing the expected absolute marker distance. We found that the performance of the tracker was much poorer than that obtained in [5] and posit that this was due to the more complex image data in our experiments and the lack of contrast between the foreground and background; see Figure 8.

In Figure 8 the performance of the loose-limbed tracker is shown with and without body part detectors. Due to the stochastic nature of the algorithms, the mean and the standard deviation of the error was computed over 10 runs. Initially the kinematic tree model appears more accurate. This is due to the fact that the "ground truth" has the same kinematic structure while the loose-limbed model is able to deviate from that. The annealed particle filter however steadily moves away from the ground truth and becomes lost. Both versions of the loose-limbed tracker outperform the competing algorithm after the first three frames. Loose-limbed tracking with part detection slightly outperforms the tracker that does not rely on the part detection. In most cases the variance of the latter is slightly higher. We predict that this difference would be more significant if we had better part detectors.

7 Conclusion

We present a probabilistic method for fully automatic 3D human detection and tracking. We show that a "loose-limbed" model with continuous-valued parameters can effectively represent a person's location and pose, and that inference over such a model can be tractably performed using belief propagation over particle sets. The belief prop-



a *b* *c* *d*

Figure 6: **Automatic Initialization from bottom-up detectors.** Each row corresponds to a different camera view (2 of the 4 views shown). The initialization used two frames but only the first frame is shown for brevity. *(a)* Samples from the shouter proposal distribution (head, upper arms, calves only). Notice that they completely fail to detect one calf while the other is detected by both left and right calf detectors. The upper arms are found but the edge is not well determined. The head is well localized. *(b)* Samples from the full proposal distribution. The body parts not constrained by shouters are sampled from a uniform distribution in position and orientation. *(c)* Samples from the belief distribution after 30 iterations of belief propagation. *(d)* Most likely limb poses after 30 iterations.

agation framework allows us to avoid distinguishing between initialization and tracking, but instead to use bottom-up part detectors to stabilize the motion estimation and provide "initialization" cues at every time-step.

The main advantages of our approach are: the complexity of the search task is linear rather than exponential in the number of body parts; bottom-up processes are integrated at every frame allowing automatic initialization and recovery from transient tracking failures; the conditional probabilities between limbs in space and time are learned from training data; a novel Gibbs likelihood model is learned from training data and models conditional dependencies between image measurements (filter responses); and forward-backward smoothing, either over a time-window or an entire sequence, is straightforward. Additionally, we exploit a novel data set with synchronized 3D "ground truth" and video data for the quantitative evaluation of performance. We also compared our method with the state of the art as proposed in [5].

While our preliminary results are promising, details of our implementation could be improved. We have used simple detectors for the head and limbs, but we expect that hands and feet would provide valuable additional cues. Other machine learning methods such as AdaBoost [26], might prove faster and more robust than the eigenfeatures we adopt. For greater applicability the method must be extended to use monocular image data and to allow a moving camera.

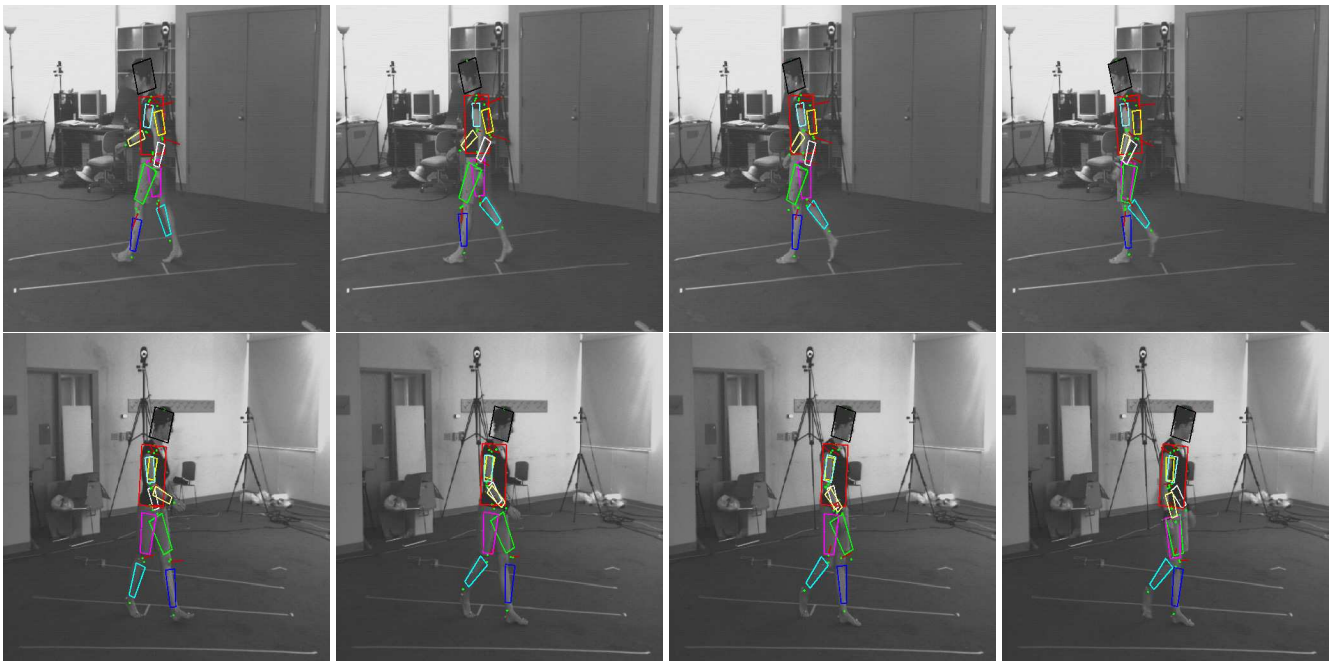
There are also limitations imposed by our use of a loose-limbed model. Since we assume independence of, for example, the left and right arms conditional on the torso location, it is cumbersome to fully avoid poses where one limb penetrates another. These problems are much easier to address with a full kinematic tree body model, and therefore one might think of the loose-limbed model as an intermediate stage between the bottom-up part detectors and a full kinematic model. The details and implementation of such a scheme are postponed to future research.

References

[1] M. Burl, M. Weber and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry, *ECCV*, pp. 628–641, 1998.

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

[3] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation, *ECCV* 3:453–468, 2002.



Frame 7 *Frame 12* *Frame 17* *Frame 22*

Figure 7: *Loose-limbed tracking.* The body model whose automatic initialization is shown in Figure 6 is tracked for a further 25 frames. Four sample frames are shown from 2 of the 4 camera views used.

[4] J. Deutscher, M. Isard and J. MacCormick. Automatic camera calibration from a single manhattan image, *ECCV*, pp. 175–188, 2002.

[5] J. Deutscher, A. Blake and I. Reid. Articulated body motion capture by annealed particle filtering, *CVPR*, II:126–133, 2000.

[6] A. Doucet, N. de Freitas and N. Gordon. Sequential Monte Carlo methods in practice, *Stats. for Eng. and Info, Sciences*, Springer Verlag, 2001.

[7] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures, *CVPR*, 2:66-73, 2000.

[8] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22(1), 1973.

[9] A.T. Ihler, E.B. Sudderth, W.T. Freeman and A.S. Willsky. Efficient Multiscale Sampling from Products of Gaussian Mixtures, *NIPS*, 2003.

[10] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees, *ICCV*, I:690–695, 2001.

[11] S. Ioffe and D. Forsyth. Probabilistic methods for finding people, *IJCV* 43(1):45–68, 2001.

[12] S. Ioffe and D. Forsyth. Finding people by sampling, *ICCV*, pp. 1092–1097, 1999.

[13] M. Isard. PAMPAS: Real-valued graphical models for computer vision, *CVPR*, I:613–620, 2003.

[14] S. Ju, M. Black and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. *Int. Conf. on Automatic Face and Gesture Recognition*, pp. 38–44, 1996.

[15] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. *ECCV* 2:3–19, 2000,

[16] Moghaddam B. and Pentland A., Probabilistic Visual Learning for Object Representation, PAMI 19(7):696–710, 1997.

[17] V. Pavolvić, J. Rehg, T-J. Cham and K. Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models, *ICCV*, pp. 94–101, 1999.

[18] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up, *CVPR*, II:467–474, 2003.

[19] S. Roth, L. Sigal, and M. Black. Gibbs likelihoods for Bayesian tracking, *CVPR*, 2004.

[20] H. Sidenbladh and M. Black. Learning the statistics of people in images and video, IJCV 54(1–3):183–209, 2003.

[21] H. Sidenbladh, M. Black and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion, *ECCV*, 2:702–718, 2000.

[22] L. Sigal, M. Isard, B.H. Sigelman, M. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation, *NIPS*, 2003.

[23] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking, CVPR, 1:447–454, 2001.

[24] C. Stauffer and W. Grimson. "Adaptive background mixture models for real-time tracking." CVPR, pp. 246–252, 1999.

[25] E. Sudderth, A. Ihler, W. Freeman and A. Willsky. Nonparametric belief propagation, *CVPR*, I:605–612, 2003.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, CVPR, I:511–518, 2001.

[27] Y. Wu, G. Hua, T. Yu. Tracking articulated body by dynamic Markov network, ICCV, pp. 1094–1101, 2003.

[28] S. Zhu, Y. Wu, and D. Mumford. FRAME: Filters, random field and maximum entropy: Towards a unified theory for texture modeling. *PAMI*, 27(2):1–20, 1998.