

From: [Michael J. Black](#)
To: [Michael Isard](#)
Cc: [Leonid Sigal](#)
Subject: Re: New draft
Date: Monday, November 17, 2003 5:59:09 PM
Attachments: [paper.pdf](#)
[mods1.tar.gz](#)

Attached is the most recent modified text and one new figure.
I've also attached the pdf.

Michael

Tracking Loose-limbed People

November 17, 2003

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 (backwards and forwards 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 permit automatic initialization and aid recovery from transient tracking failures. We illustrate the method by automatically tracking complex human motion using four calibrated cameras. Our experimental apparatus includes a marker-based motion capture system aligned with the coordinate frame of the calibrated cameras. With this we are able, for the first time, to compute the ground-truth accuracy of a markerless vision-based person tracker. This novel dataset will be made available to the community.

1 Introduction

We present a method for the fully automatic detection and tracking of human bodies in 3D. Automatic detection and robust tracking is facilitated by the use of a loose-limbed body model [25] in which limbs are connected via learned probabilistic constraints. The problem of human pose estimation is formulated as one of inference in a graphical model and belief propagation is used to infer the pose of the body. Each node in the graphical model represents the 3D position and orientation of a limb (Figure 1). Directed edges between nodes represent the constraints between limbs and are used to form messages that are sent to neighboring nodes in space and time. Additionally, each node has an associated image likelihood defined over a rich set of image cues. The combination of non-Gaussian likelihoods and the

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 [15, 30].

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 [17], 2.5D [20], or 3D [2, 7, 24, 28] leading to a high-dimensional parameter space (25-50 dimensions is not uncommon). Searching such a space directly is infeasible and 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.

Automatic initialization and failure recovery are difficult because the mapping from image measurements to the high-dimensional space of human poses is so complex. While the full body pose is hard to recover directly, the location and pose of individual limbs is much easier to compute. Many good head detectors exist and limb detectors have been used for some time (e.g. [21]). With the kinematic tree, however, 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. In practice, low-level part detectors are noisy and unreliable; that is they produce both false positives and false negatives.

The use of a loose-limbed model and belief propagation provides an elegant framework for incorporating information from part detectors. We implemented simple leg, arm, hand, and head detectors using multi-view eigenspaces. Exploiting the multi-view nature we perform 3D searches for possible limb locations and, from the construction of the eigenspace, we know the corresponding 3D pose of the limb. We emphasize that this is much simpler than searching for the entire body pose. These bottom up processes present multiple, noisy, hypotheses about the presence of limbs which are treated as proposal distributions during non-parametric belief propagation. They not only enable

automatic initialization but also allow the recovery from tracking failures.

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 represent local priors over limb poses and encode information about joint limits. They are a relatively weak prior which proves ideal for tracking varied human motions.

We test the method by tracking a subject seen from four calibrated cameras in a standard lab environment with no special clothing. There is nothing restricting this approach to multiple cameras and our current work is exploring monocular tracking. An important contribution of this work is the use of a novel motion capture testbed that provides “ground truth” human motion from a commercial motion capture system. Details of the testbed are described elsewhere [?]. To our knowledge this is the first dataset in which ground truth motion is synchronized with the video data and this enables us to report quantitative results. We will make this dataset available to other researchers in the hope that quantitative comparison on methods will more quickly advance the state of the art.

In earlier, preliminary, work [25] 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 that previous work, the potential functions were constructed manually whereas here they are learned from training data and are also extended in time. Previously, we simulated bottom-up part detectors; here they are fully implemented and integrated. Finally here we also present the use of novel ground-truth data for quantitative evaluation of the method’s accuracy.

1.1 Previous Work

ROUGH ROUGH ROUGH. I’VE TRIED TO ADD SOME NEW THINGS THAT I THINK ARE RELEVANT.

The problem has received a great deal of attention but reliable tracking has proved elusive. In 3D, the high dimensionality of articulated body models makes human tracking particularly challenging. Current methods make restrictive assumptions about the imagining environment (e.g. special clothing), assume automatic initialization, assume a limited and specialized range of motions (e.g. walking), and are brittle. Here we develop fully automatic human detection and tracking system. We exploit four calibrated cameras and a known background but make use of few other assumptions.

There have been a number of attempts to go directly from silhouettes to full 3D pose. Some of these use a single image [22] or multiple images [11]. Extending these beyond silhouettes seems impractical due to the variability of human pose. 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 do exploit background/foreground separation but it is merely for computational efficiency; it is not fundamental to our technique.

[27] infers 3D pose from a single silhouette but relies on a reasonable initialization. They also assume extremely nice silhouettes that may be difficult to recover in general scenes.

[?] go from silhouettes directly to full 3D pose. In contrast, here we focus on individual body parts and rely on BP to form an inference of 3D pose. They require thousands of training images (synthetically generated) to learn the relationship between multi-view images and full human pose; finding limbs is a much simpler and lower dimensional problem.

closest work to ours: [32]

[4] fit a partial human model (upper body) to depth measurements. this is done automatically and it recovers from mistakes

This type of model is not new for finding or tracking articulated objects and dates back at least to Fischler and Elschlager’s pictorial structures [10]. Variations on this type of model have been recently applied by Burl *et al.* [1], Felzenszwalb and Huttenlocher [9], Coughlan and Ferreira [3] and Ioffe and Forsyth [12, 13, 14]. The main benefits are that it supports inference algorithms where the computational cost is linear rather than exponential in the number of body parts, it allows elegant treatment of occlusion, and it permits automatic initialization based on individually unreliable low-level body-part detectors [34].

The work described here, like the previous work above, exploits this notion of flexible “spring”-like constraints [9] defined over individually modeled body parts [12, 13], though we extend the approach to locate the parts in 3-space rather than the 2-dimensional image plane.

The standard approach exploits a hierarchical kinematic tree. To cope with the high dimensionality of the body model previous work has used specialized stochastic search algorithms that either exploit the highly redundant dynamics of typical human motions [24], or use hierarchical sampling schemes to exploit the tree-structured nature of the model [7, 18]. These schemes have been effective for tracking people wearing increasingly complex clothing in the increasingly complex cluttered backgrounds [28].

Search using such hierarchical models is naturally thought of as starting from the root (torso) and moving outwards to the hands and feet. Unfortunately, torsos are not

the most salient part of the human body. Hands, heads, and feet may, in practice be easier to find. For example, if the torso is occluded, then any hierarchical sampling algorithm would suffer significant performance loss.

Ioffe and Forsyth [13, 12] take a different approach in which they 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. Once assembled, the configuration is evaluated based on how likely it is to be a person. In contrast, our likelihood models are defined with respect to image features. We exploit a purely bottom up process to initialize our body model but then the limbs exploit the image likelihood model to “cooperate” in searching for the body pose.

Ramanan and Forsyth [21] propose a bottom-up method for finding and tracking people but their approach is limited to 2D.

They also use ground truth registered to the video data. They treat a “detection” as any overlap between the model limb and the true limb in 2D; this is a very weak criterion for tracking success. We argue that a better quantitative measure is the percentage of overlap. Moreover we measure this overlap in 3D rather than 2D.

Gao and Shi: Alternative representations are possible, for example in [?] the nodes of the graph are associated with joints instead of limbs, but this leads to a more complicated inference algorithm due to the lack of independent local evidence likelihood functions.

2 Loose-limbed body model

Following the definition in [25] 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 1a.

As is standard for graphical models we assume the variables in a node are conditionally independent of those in non-neighboring nodes given the values of the node’s neighbors¹.

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

¹Self-occlusion and self-intersection violate this assumption. These can be modeled by adding additional edges in the graph between the possibly occluding or inter-penetrating parts. In the limit this would lead to quadratic as opposed to linear computation time in the number of parts.

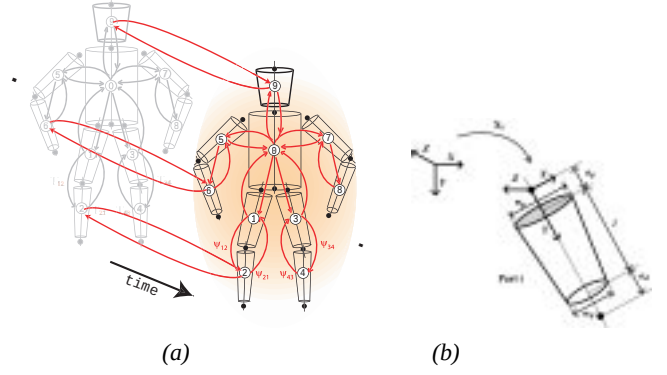


Figure 1: (a) Graphical model for a person. Nodes represent limbs and arrows represent conditional dependencies between limbs. (b) Parameterization of part i .

of the proximal and distal joints along the axis of the limb as shown in Figure 1b. 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 $\Theta_i \in \text{SO}(3)$ are the 3D position of the proximal joint 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:

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

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 computing the mean and covariance matrix respectively of the m -th Gaussian mixture component. δ_{ijm} is the relative weight of an individual component and $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$.

We learn the potentials from the ground truth motion capture data, details of this are discussed in Section 2.1 of the paper.

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:

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

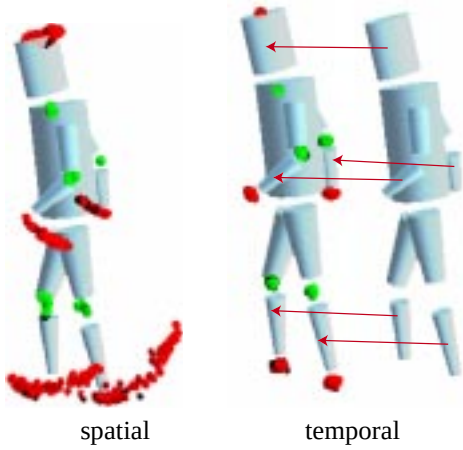


Figure 2: **Learned Spatial and Temporal Potentials.** Spatial and temporal potentials are illustrated by sampling from them. The potentials for the lower limbs (arms and legs) and head are shown. The spatial potentials show the distribution of limb positions and orientations conditioned on the neighboring 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 show are for the forward time direction.

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. Assuming independence we can learn the potential compatibility function

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

by learning the distribution over the $\mathbf{X}_{ij}$. We model this distribution using a mixture of Gaussians with $M_{ij} = 3$ components for the spatial, and $M_{ij} = 1$ for the temporal potentials. This distribution is learned using 4928 ground truth motion capture frames of walking data collected at 60 Hz.

We use a standard incremental Expectation-Maximization (EM) algorithm with K-means initialization for learning the Gaussian mixture model (GMM). While one can easily estimate the parameters of the GMM, $\hat{\mu}_m, \hat{\Lambda}_m, \hat{w}_m$, where $m \in (1..M_{ij})$, using the update rules

below,

$$\begin{aligned}\hat{\mu}_m &= \frac{\sum_{k=1}^N \mathbf{X}_{ij}^{(k)} P(m | \mathbf{X}_{ij}^{(k)}, \Theta^s)}{\sum_{k=1}^N P(m | \mathbf{X}_{ij}^{(k)}, \Theta^s)} \\ \hat{\Lambda}_m &= \frac{\sum_{k=1}^N (\mathbf{X}_{ij}^{(k)} - \hat{\mu})^T \times (\mathbf{X}_{ij}^{(k)} - \hat{\mu}) P(m | \mathbf{X}_{ij}^{(k)}, \Theta^s)}{\sum_{k=1}^N P(m | \mathbf{X}_{ij}^{(k)}, \Theta^s)} \\ \hat{w}_m &= \frac{1}{N} \sum_{k=1}^N P(m | \mathbf{X}_{ij}^{(k)}, \Theta^s)\end{aligned}\quad (4)$$

care must be taken to estimate mean and covariance for the quaternion orientations. Extending the approximation presented in [5] we define mean of the quaternion sample set

$$Q = [q_0, q_1, \dots, q_D]^T \quad (5)$$

where each sample is a unit quaternion such that $q_i = [q_{i,x}, q_{i,y}, q_{i,z}, q_{i,w}]$, where $q_{i,x}, q_{i,y}$, and $q_{i,z}$ are complex numbers and $q_{i,w}$ is real, to be

$$E_q[Q] = \left[\frac{1}{D} \sum_{i=0}^D \frac{q_{i,x}}{q_{i,w}}, \frac{1}{D} \sum_{i=0}^D \frac{q_{i,y}}{q_{i,w}}, \frac{1}{D} \sum_{i=0}^D \frac{q_{i,z}}{q_{i,w}}, 1.0 \right]. \quad (6)$$

This approximation suffers from singularity when $q_{i,w} \rightarrow 0$. To make the approximation most accurate we compute $E_q[Q]$ in a normalized quaternion space constructed such that $q_i \times [0, 0, 0, 1]^T$ is minimized for $\forall q_i \in Q$.

Similarly we approximate the covariance of the quaternion sample set by computing the deviations $\check{q}_i = q_i^{-1} * E_q[Q]$ and then

$$Cov_q[Q] = \frac{1}{D} \sum_{i=0}^D \left(\begin{bmatrix} \check{q}_{i,x} & \check{q}_{i,y} & \check{q}_{i,z} \\ \check{q}_{i,w} & \check{q}_{i,w} & \check{q}_{i,w} \end{bmatrix}^T \times \begin{bmatrix} \check{q}_{i,x} & \check{q}_{i,y} & \check{q}_{i,z} \\ \check{q}_{i,w} & \check{q}_{i,w} & \check{q}_{i,w} \end{bmatrix} \right) \quad (7)$$

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 defined over the quaternions we refer reader to [5].

NEED TO TALK ABOUT THE CONDITIONAL FIGURE HERE

2.2 Image Likelihoods

The inference algorithm outlined in the next section combines the body model described above with a probabilistic image likelihood model. In particular, 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

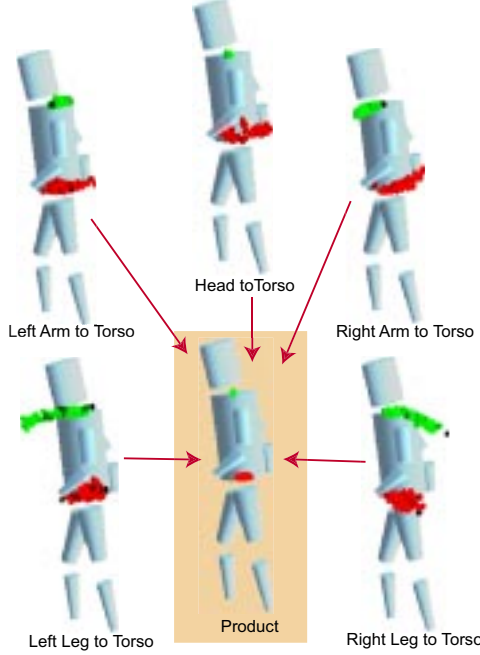


Figure 3: **Message Product.** The head, upper arms, and upper legs send messages to the torso. Samples from these messages are illustrated by showing the predicted torso location with green balls. The distribution over the orientation of the torso is illustrated by showing a red ball at the distal end of the torso for each sample. While any signal message represents uncertain information about the torso pose, the product of these messages tightly constrains the torso position and orientation.

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 as well as background subtraction information. Following related work [23], the likelihoods are estimated independently for each image view by projecting the 3D model of a limb into the corresponding image projection plane. These likelihoods are then combined across views, assuming independence, and are weighted by the observability of the limb in a given view (more weight is given to views in which the limb lies parallel to the image projection plane). For more information on the formulation of the image likelihoods see [26].

3 Non-parametric Belief Propagation

Having defined the model it remains to specify an algorithm which will perform inference and estimate a belief distribution for each of the body parts. If it were feasible

to discretize the $\mathbf{X}_i$ we could apply traditional belief propagation or a specialized inference algorithm as set out in [9]. However, the 6-dimensional configuration vector compels the use of continuous-valued random variables, and so we adopt the algorithm introduced in [15, 30] for just such types of model. It is a generalization of particle filtering [8] which allows inference over arbitrary graphs rather than just a chain. This generalization is achieved by treating the particle set which is propagated in a standard particle filter as an approximation to the “message” used in the belief propagation algorithm, and replacing the conditional distribution from the previous time step by a product of incoming message sets.

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

$$m_{ij}(\mathbf{X}_j) = \int \psi_{ij}(\mathbf{X}_i, \mathbf{X}_j) \phi_i(\mathbf{X}_i) \prod_{k \in A_i \setminus j} m_{kj}(\mathbf{X}_i) d\mathbf{X}_i, \quad (8)$$

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. (See [30] for an alternative algorithm with more general potential functions.) As discussed in [15] the samples may be stratified

1. Draw $N' = (N - 1)/M_{ij}$ samples from the proposal function:

$$\tilde{s}_{ij}^{n'} \sim f(\mathbf{X}_i), \quad n' \in [1, N'].$$

2. Compute importance corrections for $n' \in [1, N']$:

$$\eta_{ij}^{n'} = \frac{\phi_i(\tilde{s}_{ij}^{n'}) \prod_{k \in A_i \setminus j} m_{ki}(\tilde{s}_{ij}^{n'})}{f(\tilde{s}_{ij}^{n'})}.$$

3. Store normalized weights and mixture components for $n' \in [1, N'], m \in [1, M_{ij}]$:

$$(a) \quad n = (n' - 1)M_{ij} + m$$

$$(b) \quad \mu_{ij}^n = F_{ijm}(\tilde{s}_{ij}^{n'})$$

$$(c) \quad \Lambda_{ij}^n = G_{ijm}(\tilde{s}_{ij}^{n'})$$

$$(d) \quad \pi_{ij}^n = (1 - \lambda^0) \frac{\eta_{ij}^{n'} \delta_{ijm}}{\sum_{k=1}^{N'} \eta_{ij}^k}.$$

4. Assign outlier component: $\pi_{ij}^N = \lambda^0, \mu_{ij}^N = \mu_{ij}^0, \Lambda_{ij}^N = \Lambda_{ij}^0$

Figure 4: *The simplified PAMPAS non-parametric belief propagation algorithm.*

into groups with different proposal functions $f(\cdot)$, so some

samples come from the product of all incoming messages A_i into the node, some from $A_i \setminus j$ (i.e. A_i excluding j) and some from a static importance function $Q(\mathbf{X}_i)$ — we use a limb proposal distribution based on local image measurements. For reasons of space we present only a simplified algorithm to update message m_{ij} in Figure 4 which does not include the stratification but the full algorithm can be found in [15]. We use the Gibbs sampler described in [30] to form message products of $D > 2$ messages.

The algorithm must sample, evaluate, and take products over Gaussian distributions defined $\in SO(3)$ and represented in terms of unit quaternions. We adopt the approximation given in [5] 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.

4 Bottom-up Part Detectors

Occlusion of body parts, motions that violate the prior, changes in illumination, or a myriad of other situations may cause a person tracker to loose track of some, or all, parts of a body. We argue that reliable tracking requires bottom up process that constantly search for body parts and suggest their location and post to the tracker; we call these “shouters”².

One expects shouters to be noisy in that they will sometimes fail to detect parts and sometimes will find spurious parts. Furthermore it may be too much to ask for them to differentiate between left and right calfs for example. The key is that they provide guesses for the location of body parts. Our belief propagation framework is designed to exploit guesses such as these and can cope with missing and spurious hypotheses.

MENTION FORSYTH DOES RIBBON DETECTORS
DESCRIBE MECHNANISM BY WHICH WE SAMPLE FROM SHOUTERS.

We implemented simple detectors for the head, upper arms, hands, and lower legs; other body parts could be added as desired. Note that we do not need to detect all the body parts since the Bayesian inference will take whatever information the shouters provide and incorpore it with the image likelihoods and message.

Many approaches are available for body-part detection in single or multiple images. Here we implement a simple method using eigen-templates constructed via principal component analysis (PCA). This is sufficient for our pur-

²This term and the role of these processes grew out of long discussions with *withheld* and *withheld*.

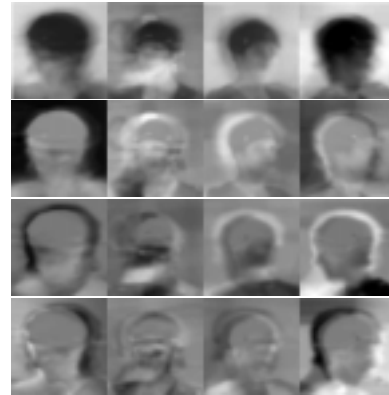


Figure 5: **Multi-view Eigenfeatures.** Top row: mean multi-view head. Next three rows: first three principal components.

poses and illustrates how bottom up information is incorporated into the loose-limbed model.

Multi-View Eigenspaces

Without loss of generality, consider building a detector to propose the 3D location and pose of heads. Given calibrated training images with known limb locations, we build a set of *multi-view* training images. Figure ?? shows several examples of multi-view training images which contain views of the head at one instant in time as seen from all four cameras. Each head view is normalized independently before constructing the multi-view image. We perform PCA on the training set of multi-view head images. We use a low-dimensional linear subspace accounting for approximately 80% of the variation in training data and build a probabilistic image likelihood model as described in [19]. Training images are projected onto the linear subspace and we store the coefficients along with the known 3D head pose in a database for later use.

This process is repeated for lower legs and lower arms.

3D Part Detection

For inferring the 3D pose of the body, the shouters must provide plausible 3D poses for the parts. One could detect parts individually in each image and then combine them to hypothesize 3D poses. Given the multi-view eigenspace defined above it is more convenient to search for parts in 3D using all camera views simulataneously.

To constrain the search space we exploit the fact that the background is known and static. We take a standard approach and model the background using a mixture model at each pixel [29]. We further exploit the training data (Appendix) to learn the range of possible 3D poses for each limb. For each part, we take the maximum extent of the

this
sec-
tion
is
too
long
(MJB)



Figure 6: **Knowing where to look.** We combine a learned uniform prior with standard foreground detection to limit the range of 3D locations that need to be considered for the detection of body parts. Colored regions show the projection of the possible range of locations in which we might find various limbs (as learned from training data).

3D limb position and define a uniform prior over that range. This range of 3D values can be projected into the images to define a uniform prior over image locations where a limb might be found. Note that this works well for most standard situations where the feet are more likely to be near the ground and the head in some range above them; this would not work well for videos of gymnastics where the head and feet could be anywhere in the 3D space.

Figure ?? shows an example of foreground detection and the 3D priors over part locations projected into each of the images. To search for possible head locations in 3D, we need only consider locations that are consistent with foreground pixels within head-prior region. This greatly reduces the search. Selecting any view, we examine all foreground pixels that could be a head. Each pixel defines a ray in 3D. We intersect this ray with the uniform prior over head positions in 3D and then search along the ray for possible heads. Note that a point on this ray is projected into the other views and if it does not correspond to a foreground pixel in all views, it is rejected. For 3D locations that correspond to foreground in all views, we construct the normalized multi-view image for the body part, project it onto the eigenspace and compute the probability that it is a head. We then take the ??? most likely such locations as a proposal distribution for belief propagation.

An example detection is shown in Figure ?. The four camera images are shown with a marker indicating the most likely head location as detected using the above methods. The multi-view image corresponding to that location is displayed along with the best matching pose in the database (for which the 3D pose is known).

5 Experiments

Figure 8 shows our really cool new boffo results.



Multi-view image corresponding to most likely 3D head location.



Best matching pose in the database.

Figure 7: **3D Part Detection.**

6 Discussion

We have implemented some simple bottom-up feature detectors (head and limbs). This approach could be generalized in many ways. Extremities such as feet and hands are an obvious extension. The use of other machine learning methods such as AdaBoost [31] could also be employed to detect such features. Here we focus on coupled feature detection in 3D but feature detection could be independently in multiple views and then a search could be performed to find plausible 3D locations. Similar ideas can be applied in the case of monocular tracking although obtaining initial 3D estimates for the parts is significantly more difficult. Finally, we have focused our bottom-up detection effort on the visible parts of our model. Note that we could have detected joint locations instead (e.g. elbow, knee, wrist, ankle) and used these for initialization [?]. To facilitate this, our graphical model could be extended to include joints as nodes [?].

The implementation here assumes rigid conical limbs with known length. More general deformable models could be employed. It would be straightforward to use the training data to learn distributions over limb lengths and other limb shape parameters from multiple subjects. Limb length (or shape) would then become a parameter of the nodes to be estimated.

There are a number of limitations of the loose-limbed scheme. Without a full body model, it is impossible to fully avoid poses where one limb penetrates another (e.g. lower arm and torso). For similar reasons, it is not straightforward to reason about occlusions between the limbs. These problems are much easier to address with a full kinematic tree body model.

One might think of the loose-limbed model as an intermediate stage between the bottom-up part detectors and a full kinematic model. In particular, one could add a node in the graphical model corresponding to the kinematic tree. This node would be connected to all the limb nodes. Messages from the limbs would be combined to produce a distribution over 3D body poses and the kinematic model in

Figure 8: Loos-limbed tracking.

turn would provide "top-down" messages constraining limb poses. This mechanism would address the problems of inter-penetration and self-occlusion mentioned above. The implementation of this idea is left for future research.

7 Conclusion

Relatively weak priors

Robustness to partial occlusion

Quantitative results for the first time (if you don't count Ramanan and Forsyth)

Future work – tracking through occlusion – self occlusion – moving camera – monocular – better likelihoods – learning appearance – tying to a kinematic model – hierarchical – links between joints (more nodes) – maybe not –

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. We implement simple part detectors and demonstrated that these bottom up processes could be incorporated into the inference process resulting in more stable tracking and re-initialization.

There are several advantages of this approach compared

with traditional particle filtering: the complexity of the search task is linear rather than exponential in the number of body parts; bottom-up initialization information can be incorporated in every frame; and forward-backward smoothing, either over a time-window or an entire sequence, is straightforward.

Acknowledgements. Thank Dave and Jon for the data. Stefan!! Ben!! Thank our beautiful subject. LS and MJB were supported in part by the DARPA HumanID Project (ONR N000140110886).

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* Vol. 3, pp. 453–468, 2002.
- [4] D. Demirdjian, T. Ko, T. Darrell. "Constraining human body tracking," *ICCV*, pp. 1071–1078, 2003.
- [5] J. Deutscher, M. Isard and J. MacCormick. Automatic camera calibration from a single manhattan image, *ECCV*, pp. 175–188, 2002.

- [6] J. Deutscher, A. Blake and I. Reid. Articulated body motion capture by annealed particle filtering, *CVPR00*, 2000.
- [7] J. Deutscher, B. North, B. Bascle and A. Blake. Tracking through singularities and discontinuities by random sampling, *ICCV*, pp. 1144–1149, 1999.
- [8] A. Doucet, N. de Freitas and N. Gordon. Sequential monte carlo methods in practice, *Statistics for Engineering and Information Sciences*, Springer Verlag, 2001.
- [9] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures, *CVPR*, Vol. 2, pp. 66–73, 2000.
- [10] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22(1), 1973.
- [11] K. Grauman, G. Shakhnarovich, T. Darrell. “Inferring 3D structure with a statistical image-based shape model,” *ICCV*, pp. 641–648, 2003.
- [12] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees, *ICCV*, Vol. I, pp. 690–695, 2001.
- [13] S. Ioffe and D. Forsyth. Probabilistic methods for finding people, *IJCV* 43(1):45–68, 2001.
- [14] S. Ioffe and D. Forsyth. Finding people by sampling, *ICCV*, pp. 1092–1097, 1999.
- [15] M. Isard. PAMPAS: Real-valued graphical models for computer vision, to appear: *CVPR*, 2003.
- [16] M.I. Jordan, T.J. Sejnowski and T. Poggio. Graphical models: Foundations of neural computation, *MIT Press*, 2001.
- [17] 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.
- [18] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. *ECCV* (2) 2000: 3–19.
- [19] Moghaddam B. and Pentland A., “Probabilistic Visual Learning for Object Representation,” *PAMI*-19 (7), pp. 696–710, July 1997.
- [20] 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.
- [21] D. Ramanan and D.A. Forsyth. “Finding and tracking people from the bottom up,” *CVPR*, pp.???, 2003.
- [22] R. Rosales and S. Sclaroff. “Specialized mappings and the estimation of human body pose from a single image,” *Proc. IEEE Workshop on Human Motion*. 2000.
- [23] H. Sidenbladh and M. Black. Learning image statistics for Bayesian tracking, *ICCV*, Vol. II, pp. 709–716, 2001.
- [24] H. Sidenbladh, M. Black and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion, *ECCV*, vol. 2, pp. 702–718, 2000.
- [25] *Withheld for review.*
- [26] *Withheld for review.*
- [27] C.Sminchisescu, A.Tealea, “Human pose estimation from silhouettes. A consistent approach using distance level sets,” *Int Conf Comp Graphics, Visualization and Comp Vision WSCG 2002*, Czech Republic.
- [28] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking, *CVPR*, vol. 1 pp. 447–454, 2001.
- [29] C. Stauffer and W.E.L. Grimson. “Adaptive background mixture models for real-time tracking,” *CVPR*, pp. 246–252, 1999.
- [30] E. Sudderth, A. Ihler, W. Freeman and A. Willsky. Nonparametric belief propagation, to appear: *CVPR*, 2003; (see also MIT AI Lab Memo 2002-020).
- [31] P. Viola and M. Jones. “Robust real-time face detection.” *ICCV*, Vol. II: 747, 2001.
- [32] Y. Wu, G. Hua, T. Yu. “Tracking articulated body by dynamic Markov network,” *ICCV*. pp. 1094–1101, 2003.
- [33] J.S. Yedidia, W.T. Freeman and Y. Weiss. Generalized belief propagation, *NIPS*, Vol 13, pp. 689–695, 2000.
- [34] S.X. Yu, R. Gross, and J. Shi. Object segmentation by graph partitioning Concurrent object recognition and segmentation by graph partitioning, *NIPS*, 2002.