

Loose-limbed People: Estimating 3D Human Pose and Motion Using Non-parametric Belief Propagation

Leonid Sigal · Michael Isard · Horst Haussecker ·
Michael J. Black

Received: 31 October 2008 / Accepted: 23 February 2011 / Published online: 30 September 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract We formulate the problem of 3D human pose estimation and 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 body-parts. In particular, we model the body using an undirected graphical model in which nodes correspond to parts and edges to kinematic, penetration, and temporal constraints imposed by the joints and the world. These constraints are encoded using pair-wise statistical distributions, that are learned from motion-capture training data. Human pose and motion estimation is formulated as inference in this graphical model and is solved using Particle Message Passing (PAMPAS). PAMPAS is a form of non-parametric belief propagation that uses a variation of particle filtering that can be applied over a general graphical model with loops. The loose-limbed model and decentralized graph structure allow us to incorporate information from “bottom-up” vi-

sual cues, such as limb and head detectors, into the inference process. These detectors enable automatic initialization and aid recovery from transient tracking failures. We illustrate the method by automatically tracking people in multi-view imagery using a set of calibrated cameras and present quantitative evaluation using the HumanEva dataset.

Keywords Articulated pose estimation · Articulated tracking · Human pose estimation · Human motion tracking · Non-parametric belief propagation

1 Introduction

We present a fully automatic method for estimating the pose of the human body in three dimensions (3D) and for tracking this pose over time. As part of this method we introduce a representation for modeling the body that we call a *loose-limbed body model*. This model, in which limbs are connected via learned probabilistic constraints, facilitates initialization and failure recovery. The tracking and pose estimation 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 image frame. Each node in the graphical model represents the 3D position and orientation of a limb (Fig. 1). Undirected 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 function defined over multiple image features. 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 (Isard 2003;

This work was performed when L. Sigal and M.J. Black were at Brown University.

L. Sigal (✉)
Disney Research, Pittsburgh, PA 15213, USA
e-mail: lsigal@disneyresearch.com

M. Isard
Microsoft Research Silicon Valley, Mountain View, CA 94043, USA
e-mail: misard@microsoft.com

H. Haussecker
Intel Labs, Santa Clara, CA 95054, USA
e-mail: horst.haussecker@intel.com

M.J. Black
Max Planck Institute for Intelligent Systems, Spemannstr. 41,
72076 Tübingen, Germany
e-mail: black@tuebingen.mpg.de

Sudderth et al. 2003) that uses a variation of particle filtering and can be applied over a loopy graph.

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 (Ju et al. 1996), 2.5D (Cham and Rehg 1999), or 3D (Bregler and Malik 1998; Deutscher and Reid 2005; Sidenbladh et al. 2000; Sminchisescu and Triggs 2003) leading to a high-dimensional parameter space (25–50 dimensions are not uncommon). In 3D, searching such a high-dimensional space directly is often impractical and so current methods typically rely on manual initialization of the body model; exceptions are Gall et al. (2010), John et al. (2009). Additionally, they often exploit strong priors characterizing the types of motions present. When such algorithms lose track (as they eventually do), the dimensionality of the state space makes it difficult to recover. Alternatively, approaches that learn low-dimensional embeddings of the human motion have also been proposed (Elgammal and Lee 2004; Li et al. 2006; Lu et al. 2007; Urtasun et al. 2006), to allow inference to take place in the low-dimensional, and often non-linear, sub-space. However, learning meaningful low-dimensional manifolds, often requires strong assumptions about the types of body motion that are possible.

While the full 3D body pose is hard to recover directly, the location and pose of individual limbs is much easier to compute (particularly in 2D). Many good face/head detectors exist (Bhatia et al. 2004; Kinoshita et al. 2006; Viola and Jones 2001) and limb detectors, while less reliable, have been used for some time (e.g. Andriluka et al. 2009; Bhatia et al. 2004; Mori et al. 2004; Ramanan and Forsyth 2003; Siddiqui and Medioni 2006). The approach we take here can use bottom up information from feature detectors of any kind. In our implementation we exploit background/foreground separation and color coherency for computational simplicity but part detectors that perform well against arbitrary backgrounds are becoming standard (Andriluka et al. 2009; Ramanan and Forsyth 2003; Ramanan et al. 2005; Viola and Jones 2001).

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 (Yonemoto et al. 2000) to solve for the body pose, but in practice low-level part detectors are not sufficiently accurate. The use of a loose-limbed model and belief propagation provides a principled 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 (Deutscher and Reid 2005), it is also straightforward to perform temporal forward–backward smoothing of the limb trajectories without modifying the basic approach.

For 2D body pose estimation, pictorial structures models have become popular (Andriluka et al. 2009; Bhatia et

al. 2004; Eichner and Ferrari 2009; Felzenszwalb and Huttenlocher 2005; Fischler and Elschlager 1973; Ramanan and Forsyth 2003; Ramanan et al. 2005). These models incorporate bottom-up part detectors and use belief propagation for inference. While very similar to our loose-limbed model, the representations and inference methods do not easily generalize to 3D body pose estimation. Efficient 2D pose estimation relies on a discretization of the search space that is not practical in 3D; our approach represents a continuous state space over 3D limb location and orientation. Furthermore to deal with temporal consistency in tracking and to avoid interpenetration in of parts in 3D, our graphical models are no longer tree structured. Hence the loose-limbed model is a generalization of pictorial structures approaches to cope with the complexity of 3D body pose estimation.

A loose-limbed body model requires a specification of the probabilistic relationships between body parts 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 effectively encode information about joint limits and represent a relatively weak prior over human poses. The model also requires an image likelihood measure for each limb. We formulate our likelihood model based on foreground silhouette and edge features. The likelihoods for different features are defined separately and combined (assuming independence) across views and feature types. It should be noted, however, that our framework is general and can use any and all available features.

We test the method by tracking subjects viewed from a number of calibrated cameras in an indoor environment with no special clothing. There is nothing restricting this approach to multiple cameras and we have explored its use for monocular pose-estimation and tracking in Sigal and Black (2006a, 2006b). For clarity, however, in this work we will only concentrate on the multi-view case. Quantitative evaluation is performed using the HUMANEVA (Sigal et al. 2010) dataset which contains synchronized motion capture data and multi-view video. The motion capture data, obtained using a commercial Vicon motion capture system (Vicon Peak, Lake Forest, CA), serves as “ground truth” in the quantitative comparison. We also compare the accuracy of our tracking to the results obtained using a standard Bayesian tracking method that uses a kinematic tree body model and an Annealed Particle Filter (APF) for inference (Balan et al. 2005; Deutscher and Reid 2005).

2 Previous Work

There has been significant work in recovering the full body pose from images and video in the last 10–15 years. Here, we will briefly review only the most relevant literature to motivate our model. For a detailed review of the literature,

we refer the reader to the following survey papers (Forsyth et al. 2006; Gavrila 1999; Moeslund and Granum 2001; Poppe 2007a). Most approaches that deal with human motion can be classified into two categories: discriminative or generative.

Discriminative approaches attempt to learn a direct mapping from image features to 3D pose from either a single image (Agarwal and Triggs 2006; Navaratnam et al. 2007; Rosales and Sclaroff 2000, 2002; Shakhnarovich et al. 2003; Sminchisescu et al. 2005; Urtasun and Darrell 2008) or multiple approximately calibrated views (Grauman et al. 2003). These approaches tend to use silhouettes (Agarwal and Triggs 2006; Grauman et al. 2003; Rosales and Sclaroff 2000, 2002) and sometimes edges (Sminchisescu et al. 2005, 2006) as image features and learn a probabilistic mapping in the form of Nearest Neighbor (NN) search (Shakhnarovich et al. 2003), regression (Agarwal and Triggs 2006), Gaussian Process (GP) regression (Urtasun and Darrell 2008), mixture of Bayesian experts (Sigal et al. 2007; Sminchisescu et al. 2005), or specialized mappings (Rosales and Sclaroff 2002). While such approaches are computationally efficient and have been shown to work reliably in restricted domains, overall they tend to deal poorly with missing or corrupted image data. They also tend to generalize poorly to poses that are uncommon or unaccounted for during training.

Generative approaches, in contrast, attempt to model the image formation process. These approaches typically rely on a kinematic tree (Marr and Nishihara 1978; Nevatia and Binford 1973) representation of the body in 2D (Ju et al. 1996), 2.5D (Cham and Rehg 1999; Wang and Rehg 2006), or 3D (Bregler and Malik 1998; Cheung et al. 2003; Choo and Fleet 2001; Corazza et al. 2006; Deutscher and Reid 2005; Gall et al. 2007, 2010; Gavrila and Davis 1996; Hogg 1983; Horaud et al. 2008; John et al. 2009; Kakadiaris and Metaxas 1996; Kehl et al. 2005; Knossow et al. 2008; Rosenhahn et al. 2008; Sidenbladh et al. 2000; Sminchisescu and Triggs 2003; Wachter and Nagel 1999). While generative models employed for human pose and motion estimation are typically very weak (i.e. they cannot generate realistic images of articulated human motion) they still tend to be very effective for inference.

In such approaches the pose is defined by a set of parameters representing the global position and orientation of the root, usually the torso, and the joint angles representing the state of each limb with respect to the neighboring part higher up in the tree. Such centralized models are very expressive and are able to effectively encode prior knowledge that can both reduce the ambiguities in the observed pose and ensure that the recovered pose meets physical constraints.

If such models are initialized “close” to the true pose, gradient descent methods can be used to refine the pose (Cham and Rehg 1999; Choo and Fleet 2001; Kehl et al. 2005; Sminchisescu and Triggs 2003; Wachter and Nagel 1999;

Wang and Rehg 2006). Initializing the model automatically, however, is a key challenge. Consequently, inference with these models typically involves generating a number of hypothesis for the pose (e.g. stochastically) and evaluating the likelihood that a given hypothesis gives rise to the image evidence observed. This sort of search is computationally challenging for 3D human pose estimation because the parameter space is high dimensional (e.g. 25–50 dimensions). Many specialized inference approaches have been developed to deal with the exponential complexity of the search. Such inference methods typically take into account the structure (Deutscher and Reid 2005; MacCormick and Isard 2000) of these models and/or the dynamics (Sidenbladh et al. 2000) of human motion. However, none of these tractably infer the articulated pose without effective initialization relatively close to the solution. For this reason, these models are particularly valuable for tracking but are typically impractical for the pose estimation task.

More recently there have been a few attempts at building generative approaches that are able to initialize automatically, most notably (Gall et al. 2007, 2010; John et al. 2009). In John et al. (2009), a hierarchical strategy is used along with an efficient evolutionary search algorithm to estimate the pose of the body in the first frame of a sequence. The hierarchical structure of the search, however, assumes that the segments higher in the kinematic hierarchy can be localized well to reduce the search for subsequent body parts. In essence, the method partitions a high dimensional search into a number of smaller conditional searches, an idea first pioneered by MacCormick and Isard (2000). Gall et al. (2007, 2010) introduce a framework based on simulated annealing that allows automatic initialization by directly searching for the global optimum of the objective function. The results are very compelling (both qualitatively and quantitatively) and the approach is only guaranteed to converge to the global optimum in the limit. The method also assumes the ability to localize the body in space in terms of a rough bounding box, which may not be trivial in noisy scenarios. While such global methods are clearly desirable and the current results are promising, more work is needed to see how well this approach works in a wider variety of cases.

As an alternative, to address the complexity of inference in generative models, a class of disaggregated models has become popular. Disaggregated models for finding or tracking articulated objects date back to Fischler and Elschlager’s pictorial structures (Fischler and Elschlager 1973) and Hinton’s “puppets” (Hinton 1976). Variations on this type of the model have also been employed for generic object detection and recognition (e.g. Fergus et al. 2003; Opelt et al. 2006; Weber et al. 2000). The main idea is to model the human body as a collection of independent body parts that are constrained at the joints (ensuring proper articulated structure of the body). Based on this notion, Ioffe and Forsyth

(2001a, 2001b) 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 poses for parts 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. Consequently this also prevents them from effectively dealing with partial occlusions of the body.

An alternative, probabilistic, way of formulating disaggregated models stems from the theory of undirected graphical models. Assuming conditional independence between body parts (e.g. pose of the right arm is conditionally independent of the left given the torso), one can model the body using a corresponding undirected graphical model and formulate tracking and pose estimation as inference in this graph. Felzenszwalb and Huttenlocher (2005) introduce a clever inference scheme that allows linear¹ complexity exact inference in such graphical models using standard Belief Propagation. They use this method to recover mostly frontal 2D articulated poses. Their inference algorithm, however, requires a tree-structured topology for the graph, a particular form of potential function (that encodes the connectivity of the body parts at the joints), and discretization of the state-space. As a result, efficiency comes at the cost of expressiveness and the resulting models cannot account for occlusions, temporal constraints or long-range correlations between body parts, all of which introduce loops into the graphical structure. Furthermore, the inference algorithm relies on the fact that the 2D model has a relatively low-dimensional state-space for each body part, making it impractical to scale the approach to 3D inference. While later extended to deal, to some extent, with correlations between body parts in Lan and Huttenlocher (2005) and to jointly learn appearance in Ramanan and Forsyth (2003) the basic method still suffers the limitations discussed above.

Recently a variant that uses denser connected graphs has been introduced (Bergholdt et al. 2010). A*-search is used to find globally optimal solutions by employing a novel lower-bound as the admissible heuristic². This architecture, unlike tree-structured models discussed above, can account for long-range correlations between body parts, but so far only deals with relatively simple kinematic structures in 2D. In a similar spirit, (Tian and Sclaroff 2010) introduces a branch-and-bound approach to inference in loopy graphs, by using a tree structured solution as a lower bound, but again they deal with a planar 2D model of the body.

The *loose-limbed body model* described here can be viewed as scalable solution that provides a trade off be-

tween expressiveness and computational resources. This model permits expressiveness similar to that of kinematic tree models, while still allowing linear inference complexity similar to Felzenszwalb and Huttenlocher (2005). Our method makes no explicit assumptions about the topology of the graph (i.e. it can deal with cyclic graphs), allows for a richer class of potential functions, and can produce a continuous estimate of pose in 3D. To achieve tractable inference, however, we resort to approximate, instead of exact, inference using a variant of Non-parametric Belief Propagation (NBP) (Sudderth et al. 2003) called Particle Message Passing (PAMPAS) (Isard 2003). The comparison with closely related prior work discussed above is compactly summarized in Table 1.

A similar approach to ours was developed at roughly the same time for articulated hand tracking by Sudderth et al. (2004). However, in Sudderth et al. (2004) the authors only deal with tracking and do not address the pose initialization/estimation problem. Another closely related approach is that of Rodgers et al. (2006) for estimating articulated pose of people from range scan data. Also similar in spirit to our approach is the work of Wu et al. (2003) which tracks 2D human motion using a dynamic Markov network and (Hua et al. 2005) which uses data-driven Belief Propagation. A much simplified observation model, that relies solely on silhouettes, is used in Wu et al. (2003) and their system does not deal with pose initialization. In Hua et al. (2005) a much richer observation model is used, but the approach is still limited to 2D pose inference in roughly frontal body orientations; the subject is assumed to be facing towards the camera and wearing distinct clothes.

3 Loose-limbed Body Model

We represent the body using an undirected graphical model in which each graph node corresponds to a body part (upper leg, torso, etc.). *Graphical models* capture the way joint distributions over random variables can be decomposed into a product of factors, each depending on only a subset of the variables. This local decomposition of the joint distribution often leads to tractable inference algorithms. We test our approach with two such models consisting of 10 and 15 body parts (see Fig. 1), corresponding to a “coarse” and “fine” body representation respectively. The latter, in addition to modeling all major limbs of the body, also models hands and feet. The 15-part model also contains a more realistic parametrization of the torso that is modeled using 2 segments (pelvis and thorax with abdomen), allowing independent twist of upper and lower body.

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

¹The method is linear in the number of parts and exponential in the number of degrees of freedom for each part.

²A*-search requires a heuristic that approximates the distance from a current partial solution to a total solution; A*-search is provably optimal if this heuristic is admissible, i.e., it never overestimates the distance to the full solution.

Table 1 Comparison of the loose-limbed body model with other generative approaches. Our approach is summarized by the grayed column on the right. The loose-limbed body model enables continuous pose estimation and tracking with a rich set of constraints, while having a

tractable inference complexity that is linear in the number of body-parts in the model (the inference is still exponential in the number of degrees of freedom within each body part)

	Centralized Models		Disaggregated Models	
	Kinematic-tree		Pictorial Structures	Loose-limbed Body Model
Inference	Local Stochastic Search	Gradient Descent	Belief Propagation	Particle Message Passing
Convergence	Local optima	Local optima	Global optima	No guarantee
State-space	Continuous		Discrete	Continuous
Constraints	Kinematic Penetration Occlusion Temporal		(<i>simple</i>) Kinematic	Kinematic Penetration Occlusion ¹ Temporal
Applications	Tracking		Pose Estimation	Pose Estimation/Tracking
Model	3D/2D		2D	3D/2D ¹
Complexity	Exponential	N/A	Linear	Linear

¹This is addressed in Sigal and Black (2006b)

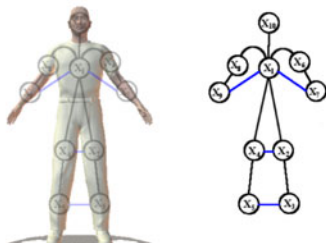
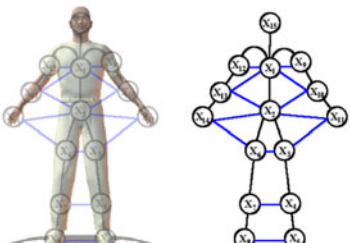
	10-part model	15-part model
Graph		
Number of nodes	10	15
Number of edges		
kinematic	9	14
interpenetration	4	13
Max node degree	7	8
Avg. node degree	2.6	3.6

Fig. 1 10-part and 15-part loose-limbed body models for a person. In each graphical model, nodes represent limbs and edges represent statistical dependencies between limbs. Black edges correspond to kine-

matic constraints, and blue to interpenetration constraints. The degree of the node is defined as the number of edges incident on that node and is a measure of graphical model complexity

to operate independently while the full body is assembled by inference over the graphical model. Edges in the graphical model correspond to correlations in position and angle relationships between adjacent body parts in space and possibly time, as illustrated in Fig. 1.

To describe the body by a graphical model, we assume that the variables in each node are conditionally independent of those in non-neighboring nodes given the values of the node's neighbors³. Each part/limb is modeled by a ta-

pered cylinder with an elliptical cross-section; this is modeled with 6 fixed and 6 estimated parameters. The fixed parameters $\Phi_i = [l_i, w_i^p, w_i^d, o_i^p, o_i^d, \epsilon_i^d]$ correspond respectively to the part length, width at the proximal and distal ends, the offset of the proximal and distal joints along the

³Self-occlusions of body parts in general violate this assumption. To deal with this, in Sigal and Black (2006b) we introduce occlusion-

sensitive likelihoods and additional graph edges to model occlusion relationships in addition to other constraints presented here. However, in the case of multiple views we find that kinematic and penetration constraints are typically sufficient to infer body pose. As the number of views decreases, or views become more degenerate, additional occlusion ambiguities will arise and occlusion constraints described in Sigal and Black (2006b) may be needed.

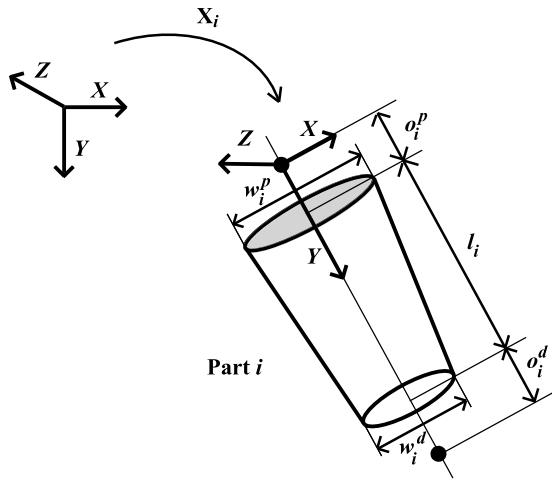


Fig. 2 Parametrization of a 3D body part

axis of the limb, as shown in Fig. 2, and eccentricity. Eccentricity models how circular the elliptical cross section of the tapered cylinder is, with 1 being perfectly circular. The offsets, o_i^p and o_i^d , are only used to limit the region in which the likelihood function is evaluated. In the vicinity of a joint, assumptions typically made by the likelihood function are often violated (Deutscher et al. 2000).

The estimated parameters $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{q}_i]^T$ represent the configuration of the part i in a global coordinate frame where $\mathbf{x}_i = [\mathbf{x}_{x,i}, \mathbf{x}_{y,i}, \mathbf{x}_{z,i}] \in \mathbb{R}^3$ and $\mathbf{q}_i \in \text{SO}(3)$ are the 3D position of the proximal joint and the angular orientation of the part respectively. The rotations are represented by unit quaternions $\mathbf{q}_i = [q_{x,i}, q_{y,i}, q_{z,i}, q_{w,i}]$, such that $\|\mathbf{q}_i\| = 1$. As a result, $\mathbf{X}_i \in \mathbb{R}^7$, lies on a 6D manifold. The overall pose of the body, \mathbf{X} , for a model with N_p parts is expressed by the collection of individual part locations and orientations, $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{N_p}\}$. This somewhat redundant representation, facilitates distributed inference using Belief Propagation.

Each undirected 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 joint probability of configuration \mathbf{X}_j of part j and \mathbf{X}_i of part i . We introduce two types of potential functions $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$ and $\psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j)$, corresponding to kinematic and penetration constraints between parts respectively. In general, these constraints are complex and non-Gaussian. While we only introduce kinematic and penetration potential functions, the framework is general and can handle a variety of other constraints (e.g. occlusions (Sigal and Black 2006b) and/or motion specific kinematics).

Formally, the joint distribution over all variables in our model, defined by the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with nodes \mathcal{V} , $|\mathcal{V}| = N_p$, corresponding to body parts and edges $\mathcal{E} = \{\mathcal{E}_K, \mathcal{E}_P\}$, corresponding to kinematic (\mathcal{E}_K) and interpenetration (\mathcal{E}_P)

constraints, can be written as follows:

$$p(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{N_p} | I) \propto \prod_{i \in \mathcal{V}} \phi_i(I | \mathbf{X}_i) \times \prod_{(i,j) \in \mathcal{E}_K} \psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j) \prod_{(i,j) \in \mathcal{E}_P} \psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j). \quad (1)$$

The conditional independence of the random variables in this graph is expressed by a neighborhood set encoded by the edges \mathcal{E} . A pair of node indices is in the graph $((i, j) \in \mathcal{E})$ if the node \mathbf{X}_j is not conditionally independent of \mathbf{X}_i given all other nodes in the graph. This formulation allows efficient inference, in the form of Particle Message Passing, details of which will be discussed in Sect. 7.

4 Constraints

The key to modeling the body using a *loose-limbed body model* is the formulation of local spatial (and temporal) coherence constraints for the body parts. In this section we define the potential functions used to probabilistically encode kinematic and interpenetration constraints between individual body parts.

Efficient inference in continuous graphical models (as described in later sections) poses a number of restrictions on the types of potential functions, $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$, that can be defined. In particular, ideally, one should (1) be able to easily sample from the product of potential functions efficiently and (2) be able to easily derive a conditional distribution⁴ of \mathbf{X}_i given \mathbf{X}_j or vice versa. The latter restriction is motivated by the inference framework, Particle Message Passing (PAMPAS), where one only needs to deal with the conditional distributions, not the full potential functions or joint distributions that give rise to these conditionals. For convergence, however, PAMPAS implicitly assumes that the joint distributions exist that give rise to these conditionals.

4.1 Kinematic Constraints

The kinematic potential functions $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$ are in general non-Gaussian and in our framework are approximated by a robust mixture of M_{ij} Gaussian kernels. Formally,

$$\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j) = \lambda^0 \mathcal{N}\left(\begin{bmatrix} \mathbf{X}_i \\ \mathbf{X}_j \end{bmatrix}; \begin{bmatrix} \bar{\mu}_i \\ \bar{\mu}_j \end{bmatrix}, \begin{bmatrix} \bar{\Lambda}_{ii} & \bar{\Lambda}_{ij} \\ \bar{\Lambda}_{ji} & \bar{\Lambda}_{jj} \end{bmatrix}\right) + (1 - \lambda^0) \times \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}\left(\begin{bmatrix} \mathbf{X}_i \\ \mathbf{X}_j \end{bmatrix}; \begin{bmatrix} \bar{\mu}_{im} \\ \bar{\mu}_{jm} \end{bmatrix}, \begin{bmatrix} \bar{\Lambda}_{im} & \bar{\Lambda}_{ijm} \\ \bar{\Lambda}_{jim} & \bar{\Lambda}_{jjm} \end{bmatrix}\right), \quad (2)$$

⁴This does not necessarily need to be a normalized distribution.

where λ^0 is a fixed outlier probability (in all experiments in this paper we use $\lambda^0 = 0.001$); $\delta_{ijm} \geq 0$ is the relative weight of an individual mixture component (designated by m) such that $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$. In addition, $\bar{\mu}_{im}$ and $\bar{\mu}_{jm}$ correspond to the means of the m -th mixture component for \mathbf{X}_i and \mathbf{X}_j ; $\bar{\Lambda}_{iim}$, $\bar{\Lambda}_{jjm}$ to the variances and $\bar{\Lambda}_{ijm}$ (and $\bar{\Lambda}_{jim}$) to co-variances of \mathbf{X}_i and \mathbf{X}_j according to the m -th mixture component; similarly $\bar{\mu}_i$, $\bar{\mu}_j$, $\bar{\Lambda}_{iim}$, $\bar{\Lambda}_{jjm}$, $\bar{\Lambda}_{ijm}$, $\bar{\Lambda}_{jim}$ correspond to means, variances and co-variances of the outlier process.

The Gaussian kernel density is closed under multiplication and conditioning, and allows the potential function to encode a rich class of constraints. This modeling choice results in a particularly convenient form for conditional distributions that can be written as follows,

$$\begin{aligned} \psi_{ij}^K(\mathbf{X}_j|\mathbf{X}_i) &= \lambda^0 \mathcal{N}(\mathbf{X}_j; \mu_{ij}, \Lambda_{ij}) + (1 - \lambda^0) \\ &\times \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}(\mathbf{X}_j; F_{ijm}(\mathbf{X}_i), G_{ijm}(\mathbf{X}_i)), \end{aligned} \quad (3)$$

where

$$F_{ijm}(\mathbf{X}_i) = \bar{\Lambda}_{jim} [\bar{\Lambda}_{iim}]^{-1} (\mathbf{X}_i - \bar{\mu}_{im})$$

and

$$G_{ijm}(\mathbf{X}_i) = \bar{\Lambda}_{jjm}^{-1} - \bar{\Lambda}_{jim} [\bar{\Lambda}_{iim}]^{-1} \bar{\Lambda}_{ijm}$$

are transformation functions that return the mean and covariance matrix respectively of the m -th conditional Gaussian mixture component. The mean and covariance of the Gaussian outlier process are⁵.

$$\mu_{ij} = \bar{\Lambda}_{ji} [\bar{\Lambda}_{ii}]^{-1} (\mathbf{X}_i - \bar{\mu}_{ij})$$

and

$$\Lambda_{ij} = \bar{\Lambda}_{jj}^{-1} - \bar{\Lambda}_{ji} [\bar{\Lambda}_{ii}]^{-1} \bar{\Lambda}_{ij}.$$

One of the challenges in modeling $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$ is that part of the state-space for $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{q}_i]^T$, corresponding to rotation in 3D, lies on Riemannian manifold. A distribution on a manifold in $\text{SO}(3)$ can be modeled using a *von Mises–Fisher* distribution (Banerjee et al. 2005) (or mixture thereof), which is a generalization of a Gaussian to an

arbitrary-dimensional spherical shell. The distribution of rotations on a 3-dimensional sphere embedded in \mathbb{R}^4 is written

$$\mathcal{M}(\mathbf{q}_i; \mu, \kappa) = \frac{\kappa}{(2\pi)^2 I_1(\kappa)} \exp(\kappa \mu^T \mathbf{q}_i), \quad (4)$$

where μ is the mean direction, $\kappa \geq 0$ is the concentration parameter (similar to variance) and I_1 denotes the modified Bessel function of the first kind of order 1. As with Gaussians the product of von Mises–Fisher distributions is in itself a von Mises–Fisher distribution. This means that PAMPAS (or any form of Nonparametric Belief Propagation) can be modified to take into account these distributions on angles. This, however, would lead to additional implementation complexity.

Instead, following Sigal et al. (2004b), Sudderth et al. (2004), we use a linearized approximation for densities that involve \mathbf{q}_i . Hence any distribution over rotations is modeled as a mixture of Gaussian distributions in \mathbb{R}^4 . Any sampled orientation from such a distribution may be projected back to $\text{SO}(3)$ by normalizing the corresponding 4-dimensional vector. This approximation works well for samples (orientations) that are tightly concentrated, and tends to overestimate the variance as they become more spread out over the sphere. Conveniently, since we model the distribution over orientation using a Gaussian mixture, the distribution over the entire state is jointly a Gaussian mixture as well and $F_{ijm}(\mathbf{X}_i) \in \mathbb{R}^7$, $G_{ijm}(\mathbf{X}_i) \in \mathbb{R}^{7 \times 7}$. We describe below how the parameters of the conditionals can be learned (it is these conditionals that are need for inference).

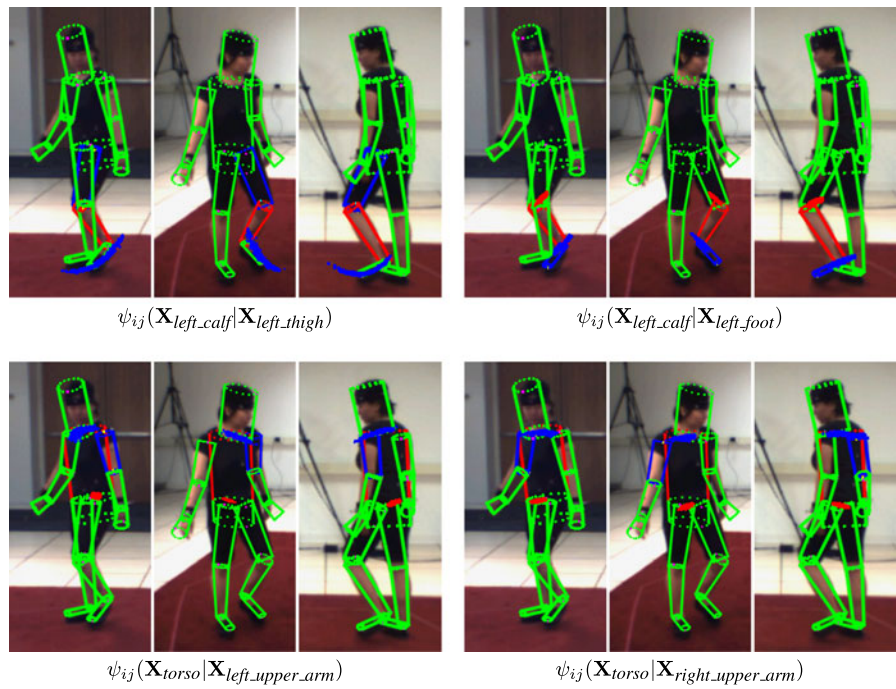
4.1.1 Deriving Kinematic Conditionals from Joint Distributions

We learn the potential functions from training data consisting of pairs of known, ground truth, state vectors for neighboring nodes i and j . We obtain the training data from motion capture sequences taken from the HUMANEVA dataset (Sigal et al. 2010). We could learn the potential function between the two nodes directly (e.g. using Expectation-Maximization (EM) for Gaussian Mixture Models (GMMs)) by simply learning the joint distribution $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j) = p(\mathbf{X}_i, \mathbf{X}_j)$. Since $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$ is modeled using a Gaussian mixture, we can derive the corresponding conditional distributions needed by PAMPAS analytically (as is illustrated in (3)).

This method of learning potential functions, however, has two disadvantages. First, learning the joint distribution, $p(\mathbf{X}_i, \mathbf{X}_j)$, in a high dimensional (\mathbb{R}^{14}) space is challenging. Second, the joint distribution will encode the prior information about both \mathbf{X}_i and \mathbf{X}_j . If we train on upright postures, for example, we will never be able to infer the pose of the person lying down (even if we have observed the full range of motion for all the joints). This is concerning, because instead of priors that only encode relative joint ranges

⁵In all experiments in this paper we use fixed μ_{ij} and Λ_{ij} . The exact value of these parameters is not important, what is important is that resulting distribution is relatively uniform across the domain of plausible limb positions and orientations (e.g., for positions the corresponding components of the mean, μ_{ij} , are set to the center of the 3D viewing volume and the diagonal elements of Λ_{ij} are set proportional to the maximal extent of this volume).

Fig. 3 Learned kinematic potential functions. Kinematic potential functions are illustrated by sampling from corresponding conditional distributions. The potential functions for the left lower leg and two potential functions for the torso are shown. The figure illustrates distributions of limb positions and orientations conditioned on the ground truth pose for the neighboring limb shown in blue. *Blue spheres* indicate the proximal joint position of a limb encoded by the sample, while the *red spheres* indicate the distal end of the limb for each sample. The spread of these samples illustrates the variance of the learned distribution. The ground truth pose for the limb is shown in red



between parts, the learned model will encode stronger prior knowledge that will favor postures in the training set, making it hard for the algorithm to generalize. Instead, we want to assume a uniform prior over both $p(\mathbf{X}_i)$ and $p(\mathbf{X}_j)$ and learn potential functions that only encode the kinematic joint constraints and limits. This amounts to learning conditional distributions $p(\mathbf{X}_j|\mathbf{X}_i)$ and $p(\mathbf{X}_i|\mathbf{X}_j)$ directly, which is allowed, so long as there exists a common joint distribution that gives rise to the two conditionals. Ideally such a learning procedure should ensure that the learned conditionals are consistent and symmetric, i.e. $p(\mathbf{X}_j|\mathbf{X}_i)$ and $p(\mathbf{X}_i|\mathbf{X}_j)$ have consistent modes. Our learning procedure, described in the next section, does not formally ensure this. Rather we rely on the training data to derive learned conditionals that are approximately symmetric and are marginalizations of some underlying joint distribution in the data.

4.1.2 Learning Kinematic Conditionals Directly

To learn the parameters of the conditional distributions directly, we first define a mapping from the state, \mathbf{X}_i , to a homogeneous 3D object-to-world matrix transformation, that we denote $H(\mathbf{X}_i) \in \mathbb{R}^{4 \times 4}$, and an inverse mapping from the homogeneous 3D object-to-world matrix back to the state, $H^{-1}(\cdot) \in \mathbb{R}^7$. The details are given in Appendix A. Using these mappings we can then define the relative states $\mathbf{X}_{ij} \in \mathbb{R}^7$, such that

$$\mathbf{X}_{ij} = H^{-1} \left([H(\mathbf{X}_i)]^{-1} \times H(\mathbf{X}_j) \right). \quad (5)$$

Intuitively \mathbf{X}_{ij} is the pose of the part j in part i 's coordinate frame for a particular pair of states. The conditional distri-

bution $\psi_{ij}(\mathbf{X}_j|\mathbf{X}_i)$ can then be expressed as a transformed distribution over \mathbf{X}_{ij} ,

$$p(\mathbf{X}_{ij}) = \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}(\mathbf{X}_{ij}; \mu_{ijm}, \Lambda_{ijm}). \quad (6)$$

We can learn a Gaussian mixture distribution for \mathbf{X}_{ij} using the Expectation-Maximization (EM) procedure.

We learn the parameters $\{\delta_{ijm}, \mu_{ijm}, \Lambda_{ijm}\}_{m=1}^{M_{ij}}$, where μ_{ijm} is the mean, Λ_{ijm} a covariance, and $\sum_{m=1}^{M_{ij}} \delta_{ijm} = 1$ are weights for the mixture components. We can then define the transformation functions $F_{ijm}(\mathbf{X}_i)$ and $G_{ijm}(\mathbf{X}_i)$ explicitly (details in Appendix A) that transform the learned mean and covariance of \mathbf{X}_i into the coordinate system of \mathbf{X}_j resulting in the conditional distribution in (3). Note that the weights of mixture components, δ_{ijm} , remain unchanged. The parameters of the outlier process in (3) are not learned and are set by hand.

While our learning algorithm is general enough to learn distributions that have couplings between positional and rotational components of the state space, resulting in full-covariance matrices, for computational purposes we restrict ourselves to the block-diagonal covariances.

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

4.2 Penetration Constraints

Another important constraint prevents interpenetration between limbs. Since the body consists of convex solid parts, they cannot physically penetrate each other. To model this we define a set of pair-wise potential functions (that encode interpenetration constraints) between the parts that are most likely to penetrate given the kinematics of the body. In the limit we could consider all pairs of parts, which would result in an inference algorithm that is quadratic in the number of parts. Instead, as a simplification, we only model the most likely penetration scenarios that arise in upright motions such as walking, running, dancing, etc.

Given a configuration, \mathbf{X}_i , of part i we want to allow potentially penetrating part j to be anywhere so long as it does not penetrate part i in its current configuration. This means that non-penetration constraints are hard to model using a mixture of Gaussians (Sigal et al. 2004b), since we need to model equal probability over the entire state space, and zero probability in some local region around the pose \mathbf{X}_i . Instead we model the penetration potential functions using the following unnormalized distribution

$$\psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j) \propto 1 - \varrho(\mathbf{X}_i, \mathbf{X}_j) \quad (7)$$

where $\varrho(\mathbf{X}_i, \mathbf{X}_j)$ is the probability that part i in configuration \mathbf{X}_i penetrates part j in configuration \mathbf{X}_j and is defined to be 1 if and only if i penetrates j in their respective configurations (0 otherwise). Notice that we can encode soft-penetration constraints by allowing $\varrho(\mathbf{X}_i, \mathbf{X}_j)$ to assume any value from 0 to 1 as a function of the overlap between parts. In our experiments, however, hard penetration constraints proved to be more effective.

There are a number of ways one can detect and measure 3D overlap between two body parts. Constructive solid geometry (CSG) (Foley et al. 1990; Wywill and Kunii 1985) could be used to detect intersections between the truncated cone primitives used for modeling body parts. Instead, we experimented with two simple approximations: spherical and voxel. The former approximates the truncated cones with a sparse set of spherical⁶ shells with corresponding non-constant radii. The set of shells approximating part i are then exhaustively intersected with the shells modeling part j . Since intersection of the two spheres can be computed using a simple Euclidean distance operator between the centroids, this process is very efficient. However, this approximation is only well suited for determining the presence or absence of the intersection between two parts, not the amount of intersection. If the amount of intersection is required, an alternative is to partition the space occupied by

one of the limbs into a set of 3D voxels and compute the approximate volume of the intersection by checking whether each voxel grid point lies within the potentially penetrating limb. Since we found hard penetration constraints to be more robust, we employ the simpler spherical approximation that avoids the additional computational complexity of the latter method.

5 Image Likelihoods

The inference algorithm, the details of which will be outlined in the next section, combines the body model described above with a probabilistic image likelihood model. We define $\phi_i(\mathbf{X}_i) \equiv \phi_i(I|\mathbf{X}_i)$ to be the likelihood of observing measurements of image I conditioned on the pose of limb i . Ideally this model should be robust to partial occlusions, the variability of image statistics across different input sequences, and variability among subjects. To that end, we combine multiple generic cues including silhouettes and edges.

5.1 Foreground Likelihood

Most algorithms that deal with 3D human motion estimation (Agarwal and Triggs 2006; Balan et al. 2005; Deutscher et al. 2000, 2002; Felzenszwalb and Huttenlocher 2005; Sigal and Black 2006b; Sigal et al. 2004b) rely on silhouette information for image likelihoods. Indeed this is a very strong cue (Balan et al. 2005) that should be taken into account when available. Here, as in most prior work, we assume that a foreground/background separation process exists that computes a binary mask $S_c(x, y)$, where $S_c(x, y) = 1$ if and only if pixel (x, y) in an image I belongs to the foreground for a given camera view $c \in [1, \dots, C]$.

Formally, we assume that pixels in the image (and hence the foreground binary mask) can be partitioned into three disjoint sub-sets (see Fig. 4(c)), $\Omega_{c,1}(\mathbf{X}_i)$, $\Omega_{c,2}(\mathbf{X}_i)$ and $\Omega_{c,3}(\mathbf{X}_i)$; where $\Omega_{c,1}(\mathbf{X}_i)$ is the set of pixels enclosed by the projection of the part i at pose \mathbf{X}_i in camera view c ; $\Omega_{c,2}(\mathbf{X}_i)$ contains pixels slightly outside part i that are likely to be statistically correlated with the part; and $\Omega_{c,3}(\mathbf{X}_i)$ are pixels that are far away and hence unlikely to be correlated with part i . Assuming pixel independence and independence of observations across camera views we write the likelihood of the image given the pose of the part as

$$\begin{aligned} \phi_{fg}(I|\mathbf{X}_i) \propto & \prod_{c=1}^C \left[\prod_{(x,y) \in \Omega_{c,1}(\mathbf{X}_i)} p_1(S_c(x, y)) \right. \\ & \times \prod_{(x,y) \in \Omega_{c,2}(\mathbf{X}_i)} p_2(S_c(x, y)) \\ & \left. \times \prod_{(x,y) \in \Omega_{c,3}(\mathbf{X}_i)} p_3(S_c(x, y)) \right], \end{aligned} \quad (8)$$

⁶3D ellipsoids can be used instead, for parts that have an elliptical cross section, with similar complexity.

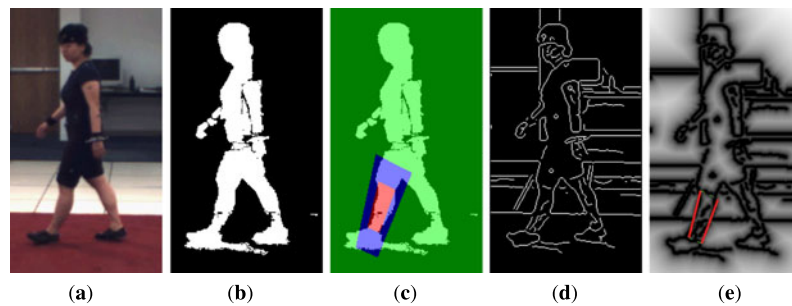


Fig. 4 Image likelihoods. Illustrated is the original image (a) and the likelihood features for computing the left lower leg likelihood; (b) illustrates the silhouette obtained by background subtraction; (c) shows the partition of the silhouette image pixels into three disjoint sub-sets where red, blue and green pixels correspond to $\Omega_{c,1}$, $\Omega_{c,2}$, and $\Omega_{c,3}$

where p_i , $i \in \{1, 2, 3\}$ are the region-specific probabilities. We learn p_1 and p_2 from a set of labeled images (this allows us to properly account for parts that have color distributions similar to background which would confuse the foreground/background separation process) and assign p_3 by hand. In general, $p_1(S_c(x, y) = 1) > 0.5$ and $p_2(S_c(x, y) = 1) < 0.5$, corresponding to the observation that pixels enclosed by projection of the part tend to be segmented as part of the foreground silhouette and pixels slightly outside typically correspond to the background. Reasoning about pixels that are outside of the immediate vicinity of the part's projection is often hard, because other parts or foreground objects may be present in the scene. To deal with this we assume equal probability for these regions, i.e., $p_3(S_c(x, y) = 1) = 0.5$. Furthermore, to simplify our likelihood model for all our experiments, we used the following learned⁷ values for all limb likelihoods (avoiding learning separate values for each part),

$$\begin{aligned} p_1(S_c(x, y) = 1) &= 0.8, \\ p_2(S_c(x, y) = 1) &= 0.3, \\ p_3(S_c(x, y) = 1) &= 0.5. \end{aligned}$$

Notice that since $S_c(x, y)$ is binary, $p_i(S_c(x, y) = 0) = 1 - p_i(S_c(x, y) = 1)$ for $i \in \{1, 2, 3\}$. The values learned, above, also consistent with those utilized by Felzenszwalb and Huttenlocher (2005) in a similar likelihood model.

5.2 Edge Likelihood

Even with perfect background subtraction, silhouettes provide ambiguous information about body pose; for example, the pose of occluded parts is unobserved. Ambiguity is reduced as the number of views increase, but with common

regions respectively. The edge image obtained by the Canny edge detector and the corresponding log of the distance transform for the edge image are shown in (d) and (e) respectively. In (e) the projected model edge pixels for which the edge likelihood is computed are shown in solid red

configurations (e.g., 4 cameras) the effects can still be significant. Hence, to reduce ambiguity and better localize parts, we also use a very simple edge-based likelihood measure; a more sophisticated model could be learned from examples (Andriluka et al. 2009; Sidenbladh and Black 2003).

We start by computing an edge distance transform, $E_c(x, y)$, by first running the Canny edge detector (Canny 1986) on the image (from camera c) and then computing a distance transform based on the resulting binary edge image. The edge based likelihood measure is then defined as follows, once again assuming independence across pixels and camera views,

$$\phi_{edge}(I|\mathbf{X}_i) \propto \prod_{c=1}^C \left[\prod_{(x,y) \in \Gamma_c(\mathbf{X}_i)} \exp(-E_c(x, y)^2) \right], \quad (9)$$

where $\Gamma_c(\mathbf{X}_i)$ corresponds to the two opposite edges of the trapezoid obtained by projection of the conic limb onto the image plane. This is illustrated in Fig. 4(e); for illustration purposes the log of the transform is shown.

5.3 Combining Features

To produce the final likelihood measure $\phi_i(I|\mathbf{X}_i)$, that takes into account both foreground and edge features, we must fuse the two likelihood terms while accounting for different *a priori* confidence we have in the two features. In particular, foreground features are in general much more reliable than edge features (Balan et al. 2005) (assuming a reasonably reliable foreground/background separation process). Taking this into account, results in the following weighted likelihood measure,

$$\phi_i(\mathbf{X}_i) = \phi_i(I|\mathbf{X}_i) = [\phi_{fg}(I|\mathbf{X}_i)]^{1-w_e} [\phi_{edge}(I|\mathbf{X}_i)]^{w_e}, \quad (10)$$

⁷We learn these values from a small set of manually labeled images.

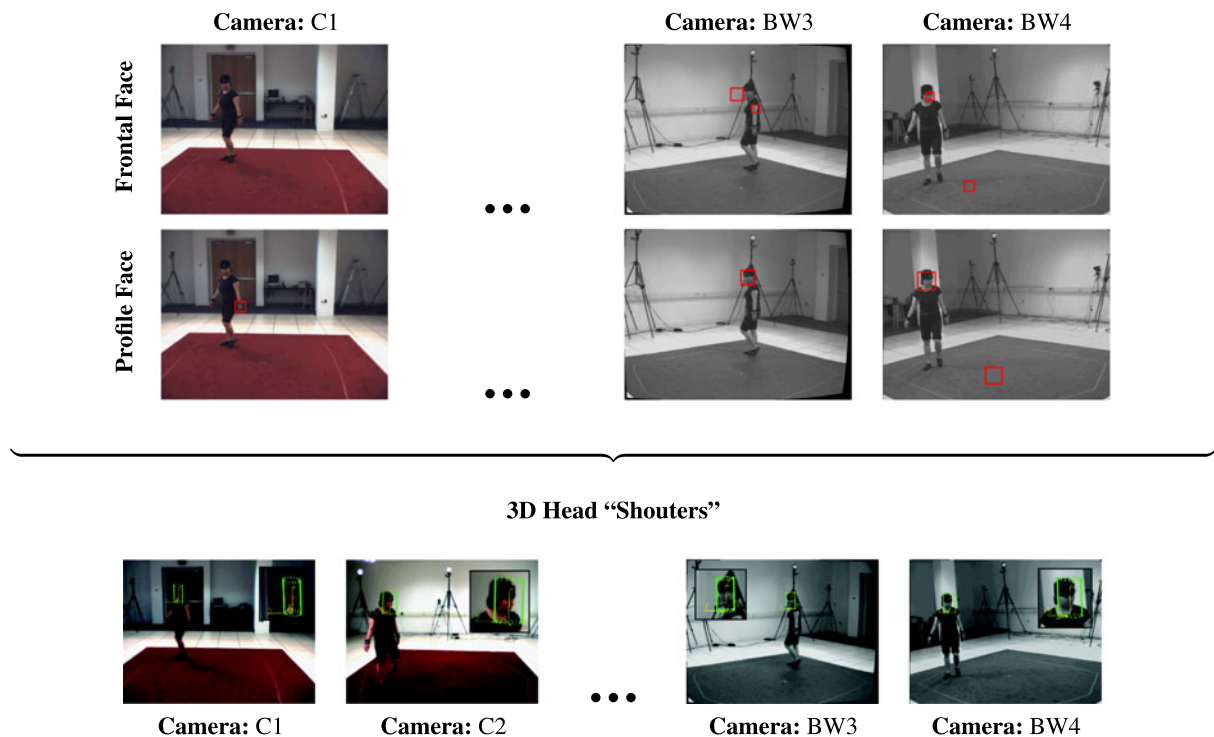


Fig. 5 Head detection. *Top two rows* show results of the Viola and Jones (2001) frontal and profile face detectors respectively (*red boxes*), run in high precision low recall mode. *Bottom row* shows 3D head estimates obtained by combining the face detection results from multiple

views. The *green bounding boxes* on the *bottom row* are projections of the 3D hypotheses for the head position and orientation; in *yellow* are the corresponding coordinate frames

where w_e is the relative confidence weight for the edge term. In practice we found $w_e = 0.1$ worked well and is used throughout.

6 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”⁸. This bottom-up process is also useful in bootstrapping the inference, by providing initial distributions over locations of a sub-set of parts. Further discussion of this in the context of Particle Message Passing can be found in Sect. 7.3.1.

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 extremities of the body. However, even these noisy guesses provide valuable low-level cues, and our belief propagation framework is designed to incorporate this bottom-up information in a principled way. As will

be described in detail in Sect. 7, we use a stratified sampler for approximating messages originating at graph node i and being sent to node j at time t . This sampler draws some fraction of samples from a static importance function $q_{ij}(\mathbf{X}_i) = f(\mathbf{X}_i)$. This importance function is constructed by the node’s shouter process, that we denote by $f(\mathbf{X}_i)$, and draws samples from locations in pose space (3D location and orientation) near the detected body parts.

6.1 Head Detection

We construct a head shouter based on the Viola and Jones face detector (Viola and Jones 2001). Specifically, we use separate detectors for frontal and profile faces; the implementation is from Intel’s OpenCV library (Intel Open Source Computer Vision Library). We apply these in multiple-views and combine the results to produce plausible estimates for the position and orientation of the head in 3D (see Fig. 5).

We first detect a set of 2D face candidates in all views, by running the two detectors at a number of scales (Fig. 5 (top)). We then pair up candidates from different views, assuming known extrinsic calibration estimated off-line for all cameras. The pose of the head can then be estimated by intersecting the frustums mapped by the two face candidates in the 3D space. The orientation about the head axis is refined, to about 45° precision, by considering the types of the

⁸The idea of “shouters” came about through discussions with A. Jepson and D. Fleet.

faces found in the two views. For example, a frontal face observed from one camera paired with a profile face found in a neighboring view, results in the overall head orientation pointing toward the camera that observed the frontal view in the first place; a frontal face observed from two different cameras results in a pose of the head where the face is pointing between the two cameras considered.

Once the 3D candidates are estimated, they are pruned by checking to ensure that the size is plausible (within limits for a human head) and that candidates project to (mostly) foreground regions in all the views. As a result of this process a set of plausible candidate poses for the head is constructed, $\{x_{head}^{(1)}, x_{head}^{(2)}, \dots, x_{head}^{(N_{head})}\}$, where N_{head} is the total number of plausible head candidates selected. The proposal function, $f(\mathbf{X}_{head})$, for the head is formulated using a kernel density estimate with Gaussian kernels centered on the candidates,

$$f(\mathbf{X}_{head}) = \sum_{n=1}^{N_{head}} \mathcal{N}(\mathbf{X}_{head}; x_{head}^{(n)}, \Lambda_{head}), \quad (11)$$

where the covariance Λ_{head} is a function of the overall head detector's precision. In general, Λ_{head} should be estimated from training data, however, since a labeled dataset with ground truth 3D head positions is not readily available, instead we set Λ_{head} by hand. We set diagonal elements of Λ_{head} , that account for variance in the estimated position and tilt of the head, to be relatively small while the twist (rotation about the head axis) is set to a considerably larger value to account for the 45° uncertainty discussed above; all off diagonal elements of Λ_{head} are set to 0.

6.2 Limb Detection

Unlike faces, limbs lack distinctive 3D shapes and textures that are consistent across people and clothing. Regardless, we build limb proposals based on color information (Mori et al. 2004), by assuming that limbs have roughly uniform color⁹. To this end, we first segment foreground regions of each view into a set of coherent color blobs (see third row of Fig. 6) using a mean-shift image segmentation procedure (Comaniciu and Meer 2002). We then fit ellipses to these regions (see fourth row of Fig. 6) and intersect frustums produced by the elliptical image regions in 3D. The intersection gives a rough estimate for the position and orientation of the limb (modulo the twist of the limb along its axis of symmetry, which is typically unobservable at standard video resolutions). Similar to head detection, we use the sizes of the estimated 3D limbs to prune the number of

candidates to a set of plausible limb positions and orientations $\{x_{limb}^{(1)}, x_{limb}^{(2)}, \dots, x_{limb}^{(N_{limb})}\}$, illustrated in the bottom of Fig. 6. As with the head, we form the proposal function for the limbs using a kernel density,

$$f(\mathbf{X}_{limb}) = \sum_{n=1}^{N_{limb}} \mathcal{N}(\mathbf{X}_{limb}; x_{limb}^{(n)}, \Lambda_{limb}). \quad (12)$$

As a result all limbs have the same proposal function and it is up to the inference and spatial (and possibly temporal) consistency constraints to interpret their identity in the context of the human body. While the inference algorithm proposed here can deal with this task, we found that this often requires many samples and results in slow convergence. Instead, since we typically are interested in dealing with mostly upright poses we modify the above proposal function as follows,

$$f(\mathbf{X}_i) = \sum_{n=1}^{N_{limb}} \mathcal{N}(\mathbf{x}_{z,i}; z_i, \Lambda_i) \mathcal{N}(\mathbf{X}_i; x_{limb}^{(n)}, \Lambda_{limb}), \quad (13)$$

where $\mathcal{N}(\mathbf{x}_{z,i}|z_i, \Lambda_i)$ weights detections as belonging to one of the body parts based on the vertical distance, z_i , from the floor¹⁰. Notice that the proposed weighting is simply a bias that helps to identify which proposed part positions are likely to belong to upper and lower extremities. These biases are the same for the left and right sides of the body and hence result in equivalent proposal functions for the two sides.

7 Inference

Pose estimation and tracking with the *loose-limbed body model* can be formulated as inference in the undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, formally introduced in Sect. 3. Belief Propagation (BP) is an efficient (and relatively standard) algorithm for inference in such graphical models. The BP algorithm operates in two stages: (1) it introduces auxiliary random variables $m_{ij}(\mathbf{X}_j)$ that can be intuitively understood as *messages* from node i to node j about what state node j should be in, and (2) computes the approximation to the marginal distribution over \mathbf{X}_i (often referred to as the belief). Messages are computed iteratively using the equations below:

$$m_{ij}^K(\mathbf{X}_j) = \int \psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j) \phi_i(\mathbf{X}_i) \times \prod_{k \in A_K(i) \setminus j} m_{ki}^K(\mathbf{X}_i) \prod_{k \in A_P(i) \setminus j} m_{ki}^P(\mathbf{X}_i) d\mathbf{X}_i \quad (14)$$

⁹Clearly this assumption can easily be violated by the various types and textures of clothing, however, one would hope that it will hold for at least some sub-set of limbs considered.

¹⁰This assumes the world coordinate system is either aligned with the floor or is known. While this assumption improves the efficiency and performance of our algorithm, it is not strictly necessary. One can use the more general form of the proposal function from (12) that assumes no knowledge of terrain.

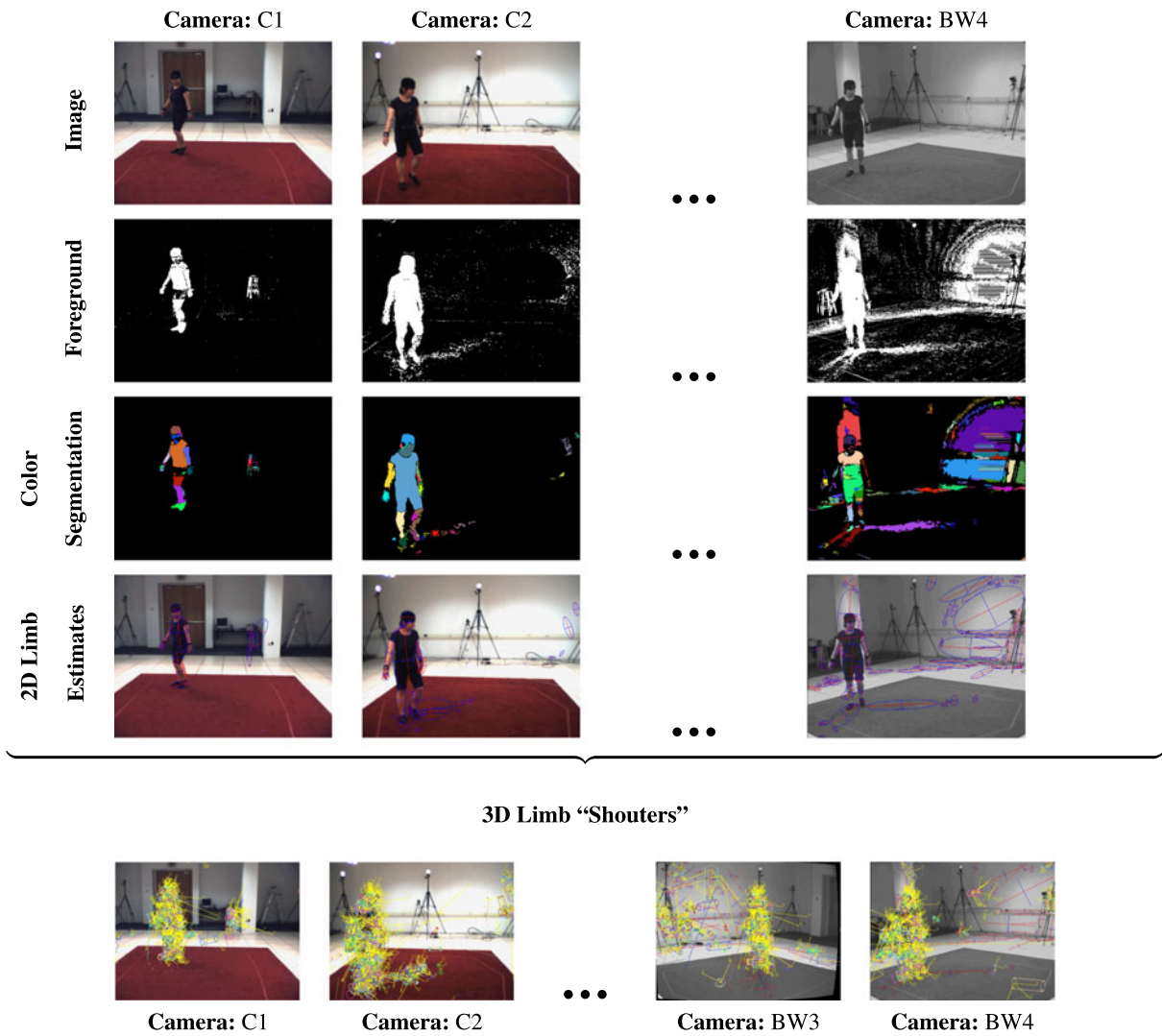


Fig. 6 Limb detection. *Top row* shows the original images from 3 out of 7 camera views. Results of foreground/background segmentation and mean-shift clustering for color segmentation of foreground regions are shown in the *second* and *third* rows respectively. Colors are

assigned to the region segments at random. The *fourth* row shows an elliptical 2D limb fit to the regions detected. The *last* row shows the resulting 3D limb estimates produced by combining the 2D estimates across different views

$$m_{ij}^P(\mathbf{X}_j) = \int \psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j) \phi_i(\mathbf{X}_i) \times \prod_{k \in A_K(i) \setminus j} m_{ki}^K(\mathbf{X}_i) \prod_{k \in A_P(i) \setminus j} m_{ki}^P(\mathbf{X}_i) d\mathbf{X}_i \quad (15)$$

where for notational simplicity we introduce a functions $A_K(i)$ and $A_P(i)$ that returns neighbors of node i connected to i by a given type of an edge/potential; in other words $j \in A_K(i) \Leftrightarrow (i, j) \in \mathcal{E}_K$, similarly $j \in A_P(i) \Leftrightarrow (i, j) \in \mathcal{E}_P$. Notice that since the loose-limbed body model formulation has two types of potential functions, $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$ and $\psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j)$ that have different representations, the messages in the two cases are different as well. Consequently, $m_{ij}^K(\mathbf{X}_j)$ is represented using a mixture of Gaussian kernel

densities and $m_{ij}^P(\mathbf{X}_j)$ by a mixture of continuous unnormalized functions. These representations stem from the choice of potential functions discussed in Sects. 4.1 and 4.2 respectively. The beliefs, where required, are given by

$$b_i(\mathbf{X}_i) \propto \phi_i(\mathbf{X}_i) \prod_{k \in A_K(i)} m_{ki}^K(\mathbf{X}_i) \prod_{k \in A_P(i)} m_{ki}^P(\mathbf{X}_i). \quad (16)$$

BP is guaranteed to converge to the exact marginals on tree-structured graphs (Jordan et al. 2001). In graphs that contain cycles (like our loose-limbed body model) BP, often referred to as *Loopy Belief Propagation* (LBP), provides an approximation to the marginals (exact inference is NP-hard (Cooper 1990)). LBP is not guaranteed to converge and in case of convergence, only converges to a fixed point (not

necessarily corresponding to a true marginal). In practice, LBP is widely used and has excellent empirical performance in many applications (Sun et al. 2002).

If the potential functions, $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$ and $\psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j)$, and the likelihoods, $\phi_i(\mathbf{X}_i)$, are all Gaussian then the marginal distribution at each node is also Gaussian (regardless of the graph topology) and the integration in the message equations can be performed exactly (Weiss and Freeman 2001). In our case however, multi-modal distributions arise naturally due to the projection ambiguities within the imaging process and non-linear nature of the human motion and pose. Since we model the potential functions and likelihoods using a Gaussian mixture model instead, it can then be shown that the representation of the messages and the marginals grows exponentially with each iteration of message passing (Koller et al. 1999) (the product of a mixture with n components and one with m components is a mixture with $m \times n$ components). Consequently we need to approximate the representation to obtain tractable inference. This gives rise to what are called *Non-parametric Belief Propagation* (NBP) algorithms (Isard 2003; Sudderth et al. 2003) that approximate messages using fixed-length kernel densities and integrals using Monte-Carlo integration.

NBP is a generalization of particle filtering (Doucet et al. 2001) 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 smoothed particle set, and the conditional distribution used in standard particle filtering is replaced by a product of incoming message sets. The two formulations of Isard (2003) and Sudderth et al. (2003) have different strengths; we adopt the PAMPAS algorithm because it corresponds better to our models where the potential functions are small mixtures of Gaussians and the likelihoods are simple to evaluate up to an unknown normalization. The method in Sudderth et al. (2003) is more suitable for applications with complex potential functions.

The message passing framework is illustrated in Fig. 7 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 (smoothed with a Gaussian kernel). Belief propagation requires forming the product of these incoming messages. As Fig. 7 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. In PAMPAS the belief propagation messages are approximated using Monte-Carlo importance sampling. This is achieved by sampling from the product of messages and then propagating these samples through an appropriate potential function.

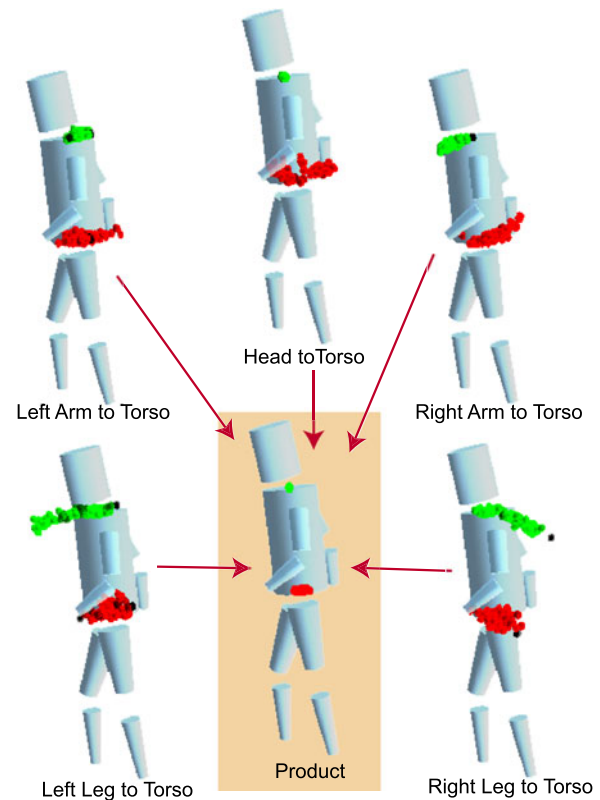


Fig. 7 Message product for the torso. In the 10-part body model, the head, upper arms, and upper legs send messages to the torso. Samples from these messages are illustrated by showing the predicted torso location with *green balls*. The distribution over the orientation of the torso is illustrated by showing a *red ball* at the distal end of the torso for each sample. While any single message represents uncertain information about the torso pose, the product of these messages tightly constrains the torso position and orientation

7.1 Particle Message Passing

The key observation, underlying both Particle Message Passing (PAMPAS) and the more general Non-parametric Belief Propagation (NBP) (Sudderth et al. 2003), is that integration required to perform message passing ((14) and (15)) can be approximated using Monte Carlo techniques. For convenience, we first formulate PAMPAS for a restricted set of graphs where the potentials $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ and likelihoods $\phi_i(\mathbf{X}_i)$ are expressed using finite Gaussian mixtures and then address the more general case where some potential or likelihood functions do not have this convenient form. The generalized version is then utilized for inference in our loose-limbed body model. In its original form, Particle Message Passing was introduced in Isard (2003); here we generalize the original formulation to make the approach appropriate for inference with the loose-limbed body model.

As in Isard (2003), for convenience we first introduce a probability density function called the *message foundation*

$$m_{ij}^F(\mathbf{X}_i) \equiv \frac{1}{Z_{ij}} \phi_i(\mathbf{X}_i) \prod_{k \in A(i) \setminus j} m_{ki}(\mathbf{X}_i), \quad (17)$$

where Z_{ij} is a normalizing constant. Intuitively, the message foundation approximates the distribution over \mathbf{X}_i that is then used to derive the compatible distribution for \mathbf{X}_j encoded by the message $m_{ij}(\mathbf{X}_j)$. We can use Monte-Carlo integration to approximate the messages by drawing N samples from the message foundation, $\{s_{ij}^{(n)} \sim m_{ij}^F(\mathbf{X}_i) | n \in [1, \dots, N]\}$, and then propagating these samples through a conditional $\psi_{ij}(\mathbf{X}_j | \mathbf{X}_i)$, resulting in the following mixture approximation to the message

$$m_{ij}(\mathbf{X}_j) = \frac{1}{\sum_{l=1}^N w_{ij}^{(l)}} \sum_{n=1}^N w_{ij}^{(n)} \psi_{ij}(\mathbf{X}_j | \mathbf{X}_i = s_{ij}^{(n)}) \quad (18)$$

where $w_{ij}^{(l)}$ is an unnormalized weight associated with each sample.

Assuming that $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ can be modeled using a joint distribution represented by the mixture of M_{ij} Gaussians (MoG), similar to (2) but without an outlier process for notational simplicity, the resulting mixture distribution,

$$\begin{aligned} m_{ij}(\mathbf{X}_j) &= \frac{1}{\sum_{l=1}^N w_{ij}^{(l)}} \sum_{n=1}^N w_{ij}^{(n)} \psi_{ij}(\mathbf{X}_j | \mathbf{X}_i = s_{ij}^{(n)}) \\ &= \frac{1}{\sum_{l=1}^N w_{ij}^{(l)}} \sum_{n=1}^N w_{ij}^{(n)} \sum_{m=1}^{M_{ij}} \delta_{ijm} \\ &\quad \times \mathcal{N}(\mathbf{X}_j; F_{ijm}(s_{ij}^{(n)}), G_{ijm}(s_{ij}^{(n)})), \end{aligned} \quad (19)$$

for the message is a Gaussian mixture as well with $M_{ij}N$ components. By assuming a MoG form for $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ we can model a large class of potential functions. For tractable inference, however, M_{ij} must remain small (on the order of tens of components).

In the simplest case, the weights are just $w_{ij}^{(n)} = \frac{1}{N}$. In general, we can sample from any *importance function*, $\{s_{ij}^{(n)} \sim q_{ij}(\mathbf{X}_i) | n \in [1, \dots, N]\}$ so long as we apply importance re-weighting resulting in non-uniform weight $w_{ij}^{(n)} \propto m_{ij}^F(s_{ij}^{(n)})/q_{ij}(s_{ij}^{(n)})$ (Doucet et al. 2000, 2001). As with any particle filter, the choice of importance function affects the convergence properties of the algorithm. Furthermore, samples can be stratified into a number of groups (Sigal et al. 2004b).

To compute the marginal distribution over \mathbf{X}_i , samples are drawn from the belief distribution $b_i(\mathbf{X}_i)$ directly or using importance sampling. These, possibly weighted, samples (sum of Dirac functions) serve as an approximate representation of the true marginal. If a continuous approxima-

tion of the marginal is required, kernel density estimation is used to smooth the particle set.

7.2 Sampling from a Product of Gaussian Mixtures

The key to inference using PAMPAS is sampling from the message foundation $m_{ij}^F(\mathbf{X}_i)$. For the moment, as in the previous section, assume that both the likelihoods, $\phi_i(\mathbf{X}_i)$, and the potentials, $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$, are expressed as mixtures of Gaussians. In this case sampling from $m_{ij}^F(\mathbf{X}_i)$ amounts to sampling from a product of Gaussian mixtures. In the next section we consider a more general case, where only a sub-set of potentials have this form.

Consider a case where a message is represented as a product of D mixtures, each with M_d , $d \in [1, \dots, D]$, components, resulting in a product that is expressed as a mixture with $\prod_{d=1}^D M_d$ Gaussian components. Hence, the brute force approach to sampling would require time exponential in the number of mixtures. This is only tractable for products of a small number of mixtures (typically $D < 3$) having relatively few mixture components. To make the sampling tractable, Sudderth et al. (2003) propose a Gibbs sampler, that can produce unbiased samples from the product in $O(KDM^2)$, as the number of iterations $K \rightarrow \infty$. In practice with a relatively small value of K a good sampling is achieved (we typically use $5 < K < 10$). In cases where $D < 3$ the brute force sampling is tractable, and we use the exact sampler instead.

The Gibbs sampler works by iteratively sampling labels $L = \{l_1, l_2, \dots, l_D\}$, where $l_d \in [1, \dots, M_d]$ corresponding to the Gaussian components in mixture d . L is initialized by randomly sampling the labels. We found that initializing the sampler by sampling l_d 's according to the probability of the mixture components in the mixture d , as in Sudderth et al. (2003), led to slower convergence in some cases. Given an initial set of labels L , we pick an integer $k \in [1, \dots, D]$ at random and sample l_k according to the marginal distribution on the labels.

Significant optimizations to the above algorithm can be made for the case where all mixture components have the same covariance. Similarly, for the specific case of mixtures that have diagonal covariance structure, an approximate sampling scheme was introduced in Ihler et al. (2003) that can sample from the product in time $O(KDM)$.

7.3 Sampling from More General Forms of the Message Foundation

It is impractical to assume that the likelihood $\phi_i(\mathbf{X}_i)$ can be explicitly modeled using a Gaussian mixture. In fact, in our case $\phi_i(\mathbf{X}_i)$ is too complex to sample from it directly. It is also possible that some sub-set of potentials $\psi_{ij}(\mathbf{X}_i, \mathbf{X}_j)$ cannot be modeled using a Gaussian mixture effectively (e.g. $\psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j)$). Hence, we must handle the

case where only a sub-set of the terms in the message foundation, $m_{ij}^F(\mathbf{X}_i)$, have the convenient Gaussian mixture form; for convenience let us call the product of those terms $m_{ij}^{FS}(\mathbf{X}_i)$. The rest of the terms that do not have the convenient form from which we can easily sample are combined into $m_{ij}^{FE}(\mathbf{X}_i)$, such that $m_{ij}^F(\mathbf{X}_i) = m_{ij}^{FS}(\mathbf{X}_i)m_{ij}^{FE}(\mathbf{X}_i)$. For example, in the case of the *loose-limbed body model*,

$$m_{ij}^{FS}(\mathbf{X}_i) = \prod_{k \in A_K(i) \setminus j} m_{ki}^K(\mathbf{X}_i) \quad (20)$$

and

$$m_{ij}^{FE}(\mathbf{X}_i) = \phi_i(\mathbf{X}_i) \prod_{k \in A_P(i) \setminus j} m_{ki}^P(\mathbf{X}_i). \quad (21)$$

The PAMPAS algorithm is easily modified to handle this case by setting the importance function $q_{ij}(\mathbf{X}_i) = m_{ij}^{FS}(\mathbf{X}_i)$. The new PAMPAS variant proceeds by sampling $s_{ij}^{(n)} \sim m_{ij}^{FS}(\mathbf{X}_i)$ from the importance function and then the importance re-weighting assigns the weight of $w_{ij}^{(n)} \propto m_{ij}^{FE}(s_{ij}^{(n)})$ to the sample. The resulting message is obtained as before, by conditioning on \mathbf{X}_i , e.g. $\psi_{ij}^K(\mathbf{X}_i = s_{ij}^{(n)}, \mathbf{X}_j) = \psi_{ij}^K(\mathbf{X}_j | \mathbf{X}_i = s_{ij}^{(n)})$.

7.3.1 Choice of Importance Functions

The previous section introduced a particular choice of importance function, $q_{ij}^{(1)}(\mathbf{X}_i)$ for approximating messages in BP using Monte Carlo,

$$q_{ij}^{(1)}(\mathbf{X}_i) = m_{ij}^{FS}(\mathbf{X}_i) = \prod_{k \in A_K(i) \setminus j} m_{ki}^K(\mathbf{X}_i), \quad (22)$$

where the bracketed superscript simply denotes the identity of the particular importance function, so that we can uniquely refer to a number of different alternatives for importance functions later on in the section. However, this choice of importance function is not always effective. In particular, consider inference in an undirected chain (e.g. Hidden Markov Model with 3 hidden random variables, $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$). The limitation is that for messages $m_{12}(\mathbf{X}_2)$ and $m_{32}(\mathbf{X}_2)$ the corresponding importance functions $q_{12}(\mathbf{X}_1) = m_{12}^{FS}(\mathbf{X}_1) = \emptyset$ and $q_{32}(\mathbf{X}_3) = m_{32}^{FS}(\mathbf{X}_3) = \emptyset$ are non-informative. While the messages correctly weigh the non-informative samples, in high-dimensional spaces this leads to poor approximation of the messages. One solution is to use a different importance function that facilitates placement of samples in high probability regions. One natural choice is to use the belief (or more precisely convenient terms of the belief) as an importance function. In other words, let

$$q_{ij}^{(2)}(\mathbf{X}_i) = m_{ij}^{FS}(\mathbf{X}_i)m_{ji}^K(\mathbf{X}_i) = \prod_{k \in A_K(i)} m_{ki}^K(\mathbf{X}_i). \quad (23)$$

In some cases this choice of importance function facilitates faster *mixing* between messages, leading to overall faster convergence of BP (Isard 2003).

In order to use either of the two importance functions, however, messages must be *initialized*. In discrete belief propagation, messages are often initialized by uniform distributions, that are then refined by Belief Propagation message passing. In the continuous case, and more specifically in the high-dimensional continuous case, having uniform messages leads to non-informative importance functions. This in turn leads to a poor approximation of the true messages, and may cause NBP to not convergence. Hence, for non-parametric BP inference to be effective, some or all messages must be initialized to semi-informative distributions¹¹. In most tracking applications (Sudderth et al. 2004) this is done by providing an initial pose, or distribution over poses at the first frame. Instead, we assume that there exists a static discriminative proposal process that provides reasonable starting values or distributions over those values for some of the variables. In other words that we use an informative proposal for some of the messages,

$$q_{ij}^{(3)}(\mathbf{X}_i) = f(\mathbf{X}_i). \quad (24)$$

In our framework this corresponds to finding plausible values for some variables corresponding to the pose of salient *limbs* or a *face* as discussed in Sect. 6.

7.4 Stratified Sampling

Stratified sampling (a.k.a. *proportional sampling*) (Cochran 1977), involves dividing the samples into a set of homogeneous groups, and sampling within each group according to some function. Here we consider stratified sampling in the context of Monte Carlo approximation to the messages in PAMPAS. The key observation is that instead of drawing all samples from one importance function that is believed to be most efficient, we stratify the sampling procedure to draw samples from multiple importance functions. As a result, the samples are more diverse overall, yet focused within each group.

For the stratified sampling to be effective, one must ensure that the number of groups (*strata*), is relatively small in relationship to the total number of samples, N . In addition, having widely disproportionate fractions of samples may cause sampling artifacts. We found stratified sampling to be effective in PAMPAS. The full stratified sampling PAMPAS procedure for kinematic messages is outlined in Algorithm 1. The procedure for the penetration messages is similar and only differs in steps (4)–(6) where a mixture of

¹¹Notice that this is equivalent to having an informative importance function for some of the variables in the graph.

Input: Graphical model $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with specified robust potentials $\psi_{ij}^K(\mathbf{X}_i, \mathbf{X}_j)$, $\psi_{ij}^P(\mathbf{X}_i, \mathbf{X}_j)$ and likelihood functions $\phi_i(\mathbf{X}_i)$
Set of possibly uninitialized messages $m_{ij}^K(\mathbf{X}_j)$ and $m_{ij}^P(\mathbf{X}_j)$
Number of samples to use for approximating the message, N
Output: Updated message $\tilde{m}_{ij}^K(\mathbf{X}_j)$ consisting of $NM_{ij} + 1$ Gaussian kernels

1. Collect all terms in the message foundation $m_{ij}^F(\mathbf{X}_i) = \frac{1}{Z_{ij}} \phi_i(\mathbf{X}_i) \prod_{k \in A_K(i) \setminus j} m_{ki}^K(\mathbf{X}_i) \prod_{k \in A_P(i) \setminus j} m_{ki}^P(\mathbf{X}_i)$, that have the Gaussian mixture form into $m_{ij}^{FS}(\mathbf{X}_i)$ term, i.e. let $m_{ij}^{FS}(\mathbf{X}_i) = \prod_{k \in A_K(i) \setminus j} m_{ki}^K(\mathbf{X}_i)$
2. Set importance functions, $q_{ij}^{(k)}$, and corresponding sampling fractions, γ_k , to be used in stratified sampling (where $k \in [1, \dots, 3]$)
 - (a) $q_{ij}^{(1)}(\mathbf{X}_i) = m_{ij}^{FS}(\mathbf{X}_i)$ $q_{ij}^{(2)}(\mathbf{X}_i) = m_{ij}^{FS}(\mathbf{X}_i) m_{ji}^K(\mathbf{X}_i)$ $q_{ij}^{(3)}(\mathbf{X}_i) = f(\mathbf{X}_i)$
 - (b) For the first iteration of BP let $\gamma_1 = 0$, $\gamma_2 = 0$, $\gamma_3 = 1$, for the rest let $\gamma_1 = 0.5$, $\gamma_2 = 0.5$, $\gamma_3 = 0$. where $f(\mathbf{X}_i)$ is the static proposal distribution.
3. For each of the importance functions $k \in [1, \dots, 3]$
 - (a) Compute a starting sample index, $N_s = \sum_{l=1}^{k-1} N\gamma_l$, (if $k = 1$, $N_s = 0$)
 - (b) Compute the number of samples to draw, $N_k = N\gamma_k$
 - (c) Draw N_k samples from the proposal function:

$$s_{ij}^{(N_s+n)} \sim q_{ij}^{(k)}(\mathbf{X}_i), n \in [1, \dots, N_k]$$

- (d) Compute the importance correction for $n \in [1, \dots, N_k]$

$$w_{ij}^{(N_s+n)} = \frac{m_{ij}^F(s_{ij}^{(N_s+n)})}{q_{ij}^{(k)}(s_{ij}^{(N_s+n)})}$$

4. Given a robust potential function for which the conditional can be derived, i.e.

$$\psi_{ij}^K(\mathbf{X}_j|\mathbf{X}_i) = \lambda_0 \mathcal{N}(\mathbf{X}_j; \mu_0, \Lambda_0) + (1 - \lambda_0) \sum_{m=1}^{M_{ij}} \delta_{ijm} \mathcal{N}(\mathbf{X}_j; F_{ijm}(\mathbf{X}_i), G_{ijm}(\mathbf{X}_i)),$$

store normalized weights and mixture components for $n \in [1, \dots, N]$, $m \in [1, \dots, M_{ij}]$:

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

$$(b) \ \mu_{ij}^{(n')} = F_{ijm}(s_{ij}^{(n)})$$

$$(c) \ \Lambda_{ij}^{(n')} = G_{ijm}(s_{ij}^{(n)})$$

$$(d) \ \pi_{ij}^{(n')} = (1 - \lambda_0) \frac{w_{ij}^{(n)} \delta_{ijm}}{\sum_{l=1}^N w_{ij}^{(l)}}$$

5. Assign outlier components: $\pi_{ij}^{(NM_{ij}+1)} = \lambda_0$, $\mu_{ij}^{(NM_{ij}+1)} = \mu_0$, $\Lambda_{ij}^{(NM_{ij}+1)} = \Lambda_0$
6. Let $\tilde{m}_{ij}^K(\mathbf{X}_j) = \sum_{n=1}^{NM_{ij}+1} \pi_{ij}^{(n)} \mathcal{N}(\mathbf{X}_j|\mu_{ij}^{(n)}, \Lambda_{ij}^{(n)})$.

Algorithm 1 PAMPAS stratified message update for a kinematic message

the continuous functions is formed instead of the mixture of Gaussian kernels, i.e.

$$\tilde{m}_{ij}^P(\mathbf{X}_j) = 1 - \sum_{n=1}^N \left[\frac{w_{ij}^{(n)}}{\sum_{l=1}^N w_{ij}^{(l)}} \mathcal{Q}(s_{ij}^{(n)}, \mathbf{X}_j) \right].$$

The stratified sampler we use draws all its samples from $q_{ij}^{(3)}(\mathbf{X}_i)$ for the first message passing iteration and then samples half of samples from $q_{ij}^{(1)}(\mathbf{X}_i)$ and half from $q_{ij}^{(2)}(\mathbf{X}_i)$

for the remaining iterations. We found the sampling from $q_{ij}^{(2)}(\mathbf{X}_i)$ sometimes leads to faster convergence, whereas sampling from $q_{ij}^{(1)}(\mathbf{X}_i)$ often leads to better results when the solution is close to convergence.

An illustration of PAMPAS (implemented by the stratified importance sampling discussed in this section) being utilized for pose estimation with a 10-part loose-limbed body model is seen in Fig. 8. In Fig. 8 marginals are illustrated in terms of: (1) the most likely sample drawn from

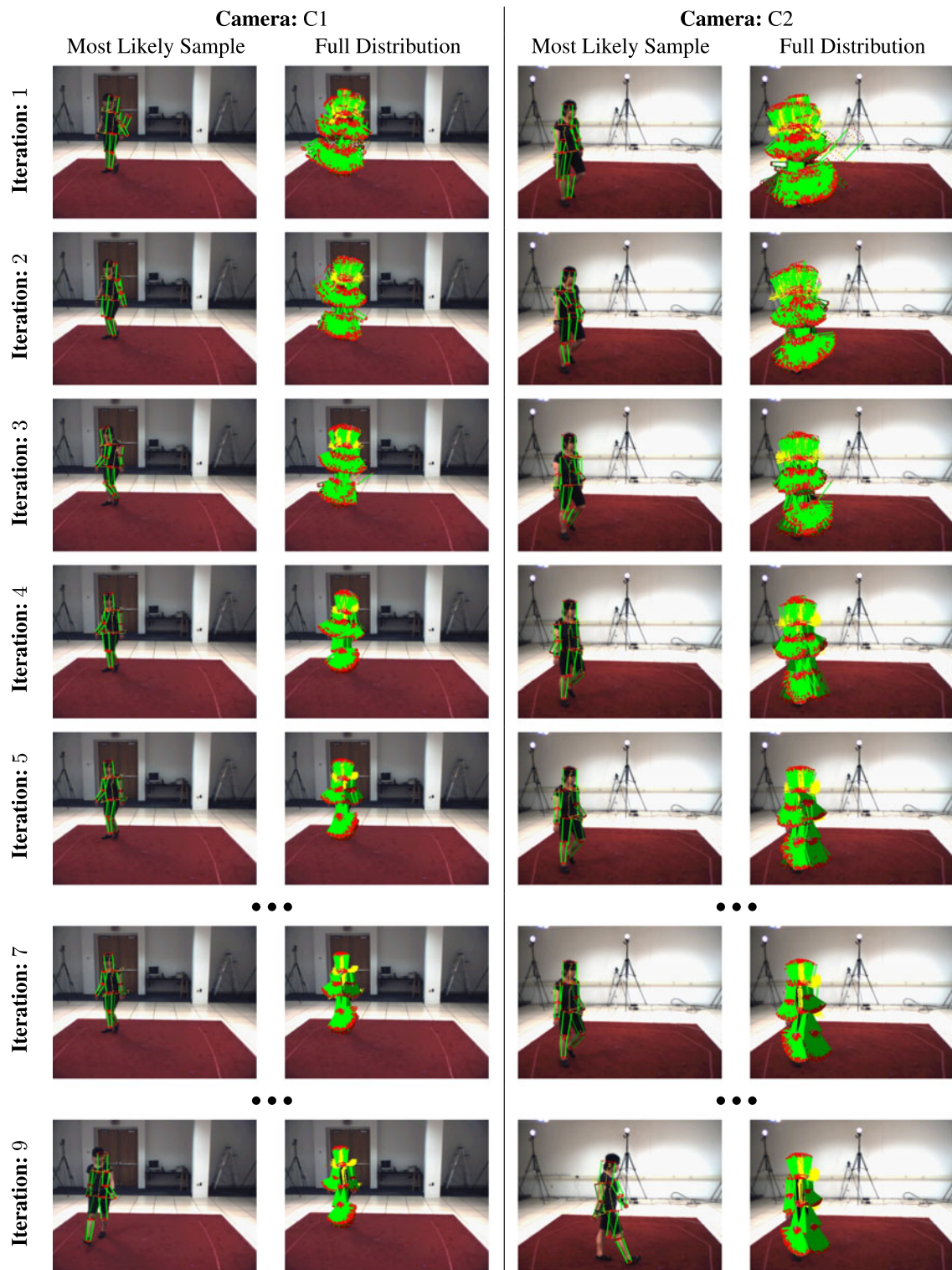


Fig. 8 Illustration of convergence of loose-limbed body model during pose estimation. Marginals for each limb estimated by PAMPAS are illustrated after 1–5, 7 and 9 message passing iterations. The marginals converge to a desired solution in roughly 5 iterations in

this case, after which they are refined without significantly affecting the mode of the marginal distribution. The error curve as a function of PAMPAS iterations for this frame can be found in Fig. 11 (Frame 450)

the marginal (left); and (2) the full distribution visualized by overlapping samples (right). In all images the dark and light green illustrate parts belonging to the left and right sides of the body respectively; yellow illustrates coordinate frames for the torso and the head. The marginals converge to the desired solution within the first 5–6 iterations.

For additional details about the implementation of the algorithm used for all experiments in the paper we refer the reader to Appendix B.

8 Tracking

The above model is formulated for pose estimation at a single time instant. Here we extend the model to enable the tracking of pose over time.

8.1 Tracking Using a Spatio-temporal Model

The most direct way of extending the pose estimation framework to tracking, is by replicating and chaining the spatial loose-limbed body model across time. The new spatio-temporal graphical model requires additional temporal constraints between limbs at time $t - 1$ and t , that we denote by $\psi^T(\mathbf{X}_{i,t-1}, \mathbf{X}_{i,t})$. Typically a single Gaussian potential is sufficient to model these temporal constraints. For example,

$$\psi^T(\mathbf{X}_{i,t-1}, \mathbf{X}_{i,t}) = \mathcal{N}(\mathbf{X}_{i,t} - \mathbf{X}_{i,t-1}; 0, \Lambda_T), \quad (25)$$

is equivalent to a zero velocity temporal prior. With this type of temporal constraint, inference can be performed as before using Particle Message Passing in either batch or sliding window fashion. We have explored this alternative in Sigal et al. (2004b). A similar approach has also been discussed in the context of generic object tracking by us in Sigal et al. (2004a).

The benefit of this type of spatio-temporal model is that temporal consistency is well maintained, the disadvantage is the additional computational cost resulting from the more complex model. In addition, if tracking fails, the spatio-temporal model is often harder to re-initialize, because of the tight coupling to the pose at the previous time instants.

8.2 Tracking Using Importance Sampling

An alternative approach, that we take in this paper, is to propagate pose information over time using importance sampling. This approach does not alter the model already introduced, and hence does not require additional computation. In essence, it assumes that we are solving the pose estimation problem at every frame, and the pose from the previous time step is only used as an initialization (or guess)

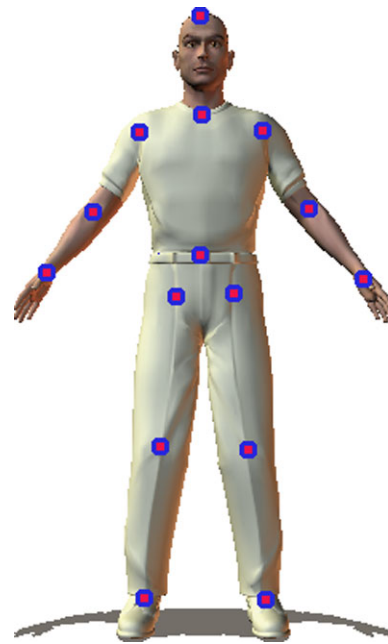


Fig. 9 Virtual marker-based evaluation metric. We define an evaluation metric based on the average distance from the estimated pose to the true pose for a set of 15 virtual markers corresponding to the 3D joint positions and limb ends

for where to start the inference at the next frame. As such, this approach is well suited for re-initialization if the pose estimate at the previous timeframe is wrong. The disadvantage is that temporal consistency is only loosely enforced, and the results often exhibit interframe jitter.

In particular, we define another importance function,

$$q_{ij}^{(4)}(\mathbf{X}_{i,t}) = \mathcal{N}(\mathbf{X}_{i,t}; \mathbf{X}_{i,t-1}, \Lambda_T). \quad (26)$$

Sampling from this importance function places the samples in the vicinity of the solution obtained at the previous time step. This is then refined using the observations from the current frame and the message passing. The covariance, Λ_T , can be learned from data, however, for the experiments in this paper we let $\Lambda_T = \text{diag}([0.1, 0.1, 0.1, 0.05\pi, 0.05\pi, 0.05\pi, 0.05\pi])$. Altering the fraction of samples that come from the different importance functions in the stratified sampling will have an effect on the diversity of poses considered at any given time instant. For the experiments presented in this paper, we use the simple generic importance sampling scheme discussed previously. To accommodate the additional importance function, $q_{ij}^{(4)}(\mathbf{X}_{i,t})$, step 2 (b) of the Algorithm 1 is altered such that for the first iteration of BP the sampling proportions are $\gamma_1 = 0, \gamma_2 = 0, \gamma_3 = 0.1, \gamma_4 = 0.9$ (for the remainder of iterations the sampling proportions are same as before— $\gamma_1 = 0.5, \gamma_2 = 0.5, \gamma_3 = 0.0, \gamma_4 = 0.0$).

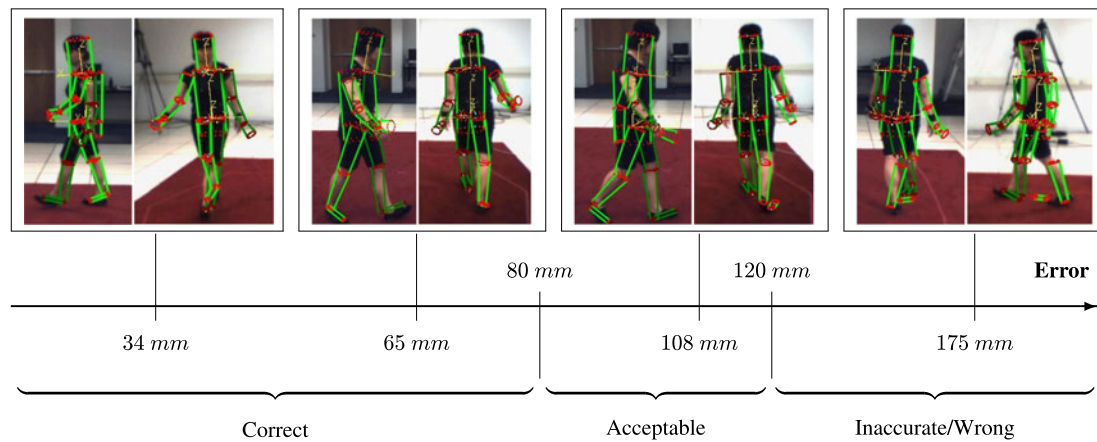


Fig. 10 Evaluation metric illustration. Intuition for the range of errors is provided. Typically an error of <80 mm corresponds to a correct pose and an error between 80–120 mm to a acceptable pose. As can be seen from the figure with an error of 108 mm the body is vertically

shifted down and the arms not well aligned, but the overall pose is still reasonable. Typically errors >120 mm correspond to wrong or inaccurate poses

9 Experiments and Evaluation

We evaluate the performance of our articulated pose estimation and tracking approach using the HUMANEVA-I dataset¹² (Sigal et al. 2010). The dataset consists of synchronized video streams from 3 color and 4 greyscale cameras at 60 Hz along with ground truth 3D body poses obtained using a commercial motion capture system. HUMANEVA-I contains 4 subjects performing a set of 6 predefined actions three times (twice with video and motion capture, and once with motion capture alone). The dataset is partitioned into training, validation and testing sub-sets. We learn potentials from training motion capture data that exhibits motions similar to those observed in the test set.

We assess the accuracy of recovered poses using the evaluation metric proposed in Sigal et al. (2010) which measures the sum of the Euclidean distances to $K = 15$ virtual markers corresponding to the locations of the major joints (see Fig. 9). Specifically, from our redundant body model $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ we derive the joint locations $\mathbf{X}_{mrk} = \{p_1, p_2, \dots, p_K\}$, where $p_k \in \mathbb{R}^3$ is the position of the marker k in the world. For every marker (except for the markers corresponding to the limb ends) we compute an average of the proximal and distal ends¹³ of the two limbs connected at the corresponding joint. The error in the overall estimated pose $\hat{\mathbf{X}}_{mrk}$ to the ground truth pose \mathbf{X}_{mrk} (in mm) is expressed as the average absolute distance between indi-

vidual markers,

$$Error(\mathbf{X}_{mrk}, \hat{\mathbf{X}}_{mrk}) = \sum_{k=0}^K \frac{\|p_k - \hat{p}_k\|}{K}. \quad (27)$$

Lower error corresponds to poses that more closely match the ground truth motion capture data. Qualitatively, errors of under 80 mm correspond to correct poses, 80–120 mm typically correspond to acceptable poses that are mostly right, and errors of >120 mm typically correspond to wrong or inaccurate poses. By “correct pose” we mean that all parts are appropriately recovered, but there may be misalignment at the joints (see Fig. 10) or slight global vertical shift of the body. These accuracy ranges are provided only to give some intuition about the algorithm performance. In general acceptable levels of accuracy are dictated by the application at hand (e.g., marker-less motion capture will generally demand higher fidelity; action recognition may require lower fidelity). To compute performance over a temporal sequence (for tracking), we average the error over all the frames in the sequence and report the mean and standard deviation for the sequence.

9.1 Static Pose Estimation

Figures 11 and 12 show the automatic pose estimation of the 3D body model using bottom-up part detectors. These results are for a single time instant (i.e. no temporal model). The approach is tested on a total of 198 frames; 128 frames using a 10-part model and 70 frames using a 15-part model. Note that we use only detectors for the head and outermost extremities, which, for the 10-part model, means lower arms and calves and for the 15-part model, hands and feet. In practice, lower arms and calves may be easier to detect in

¹²Dataset is available from <http://vision.cs.brown.edu/humaneva/>.

¹³This assumes that both proximal and distal markers correspond to the joint center. Alternatively, if this is not the case, there will be a constant offset between the proximal and/or distal ends of the limb and the required joint marker. This offset can typically be solved for in a least-squared sense using regression.

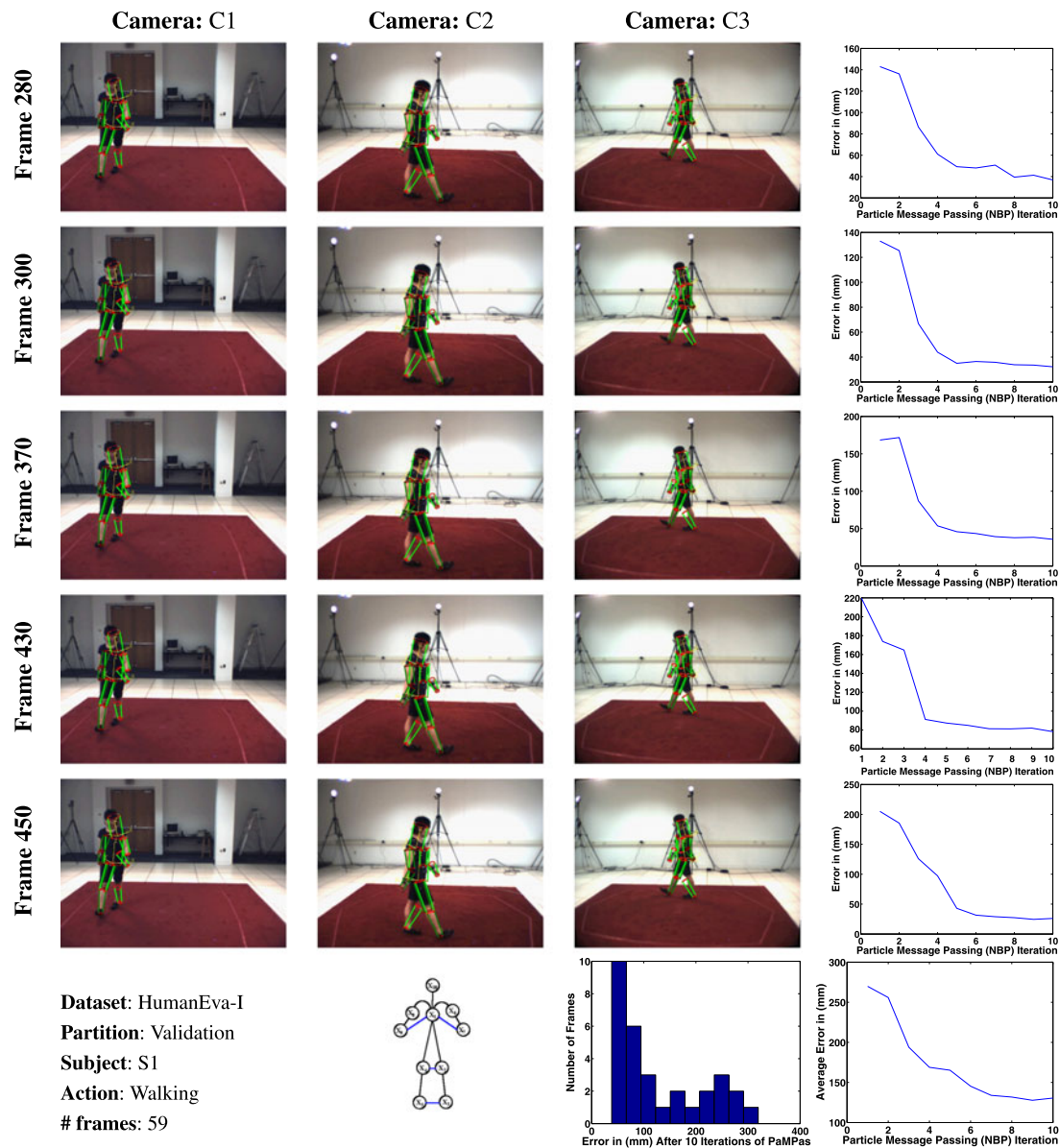


Fig. 11 Pose estimation using 10-part loose-limbed body model. Results of pose estimation at a single time instant are shown for a number of frames from HumanEva-I dataset. The *top five rows* show the final result in terms of the most likely sample from the marginal distribution over each part after 10 iterations of PAMPAS. The results are projected into 3 synchronized views for clarity (7 views are used for inference). The *right column of the first five rows* shows the error as a function of message passing iterations for the respective time instants. Notice that

typically the error decreases sharply for the first 4–5 iterations and then stays relatively low with minor variations that are due to sampling. The *last row* illustrates performance over all frames tested for the sequence (every 10-th frame was selected). The *bar plot* shows the distribution of errors, which are concentrated below 120 mm. The error as a function of message passing iterations averaged over all frames is shown in the *bottom right corner of the figure*

most cases, even for the 15-part model (shadows and self occlusions make detection of feet and hands challenging). This would lead to different message passing schedules for the two models (see Appendix B) and consequently, to keep the algorithmic details the same, we use hand and feet shouters for the 15-part model. It is important to note however, that the approach described here is not tied to any particular set of part detectors. After several iterations of belief propaga-

tion, the algorithm “finds” the limbs and has a reasonable distribution over the limbs poses. Notice that while we run PAMPAS for 10 message passing iterations, the solution often settles after 5–6 iterations.

We test the method on sequences of two subjects performing two different motions (walking and jogging). All experiments use the same values for all system parameters. In all experiments reported here, we use 7 camera views for

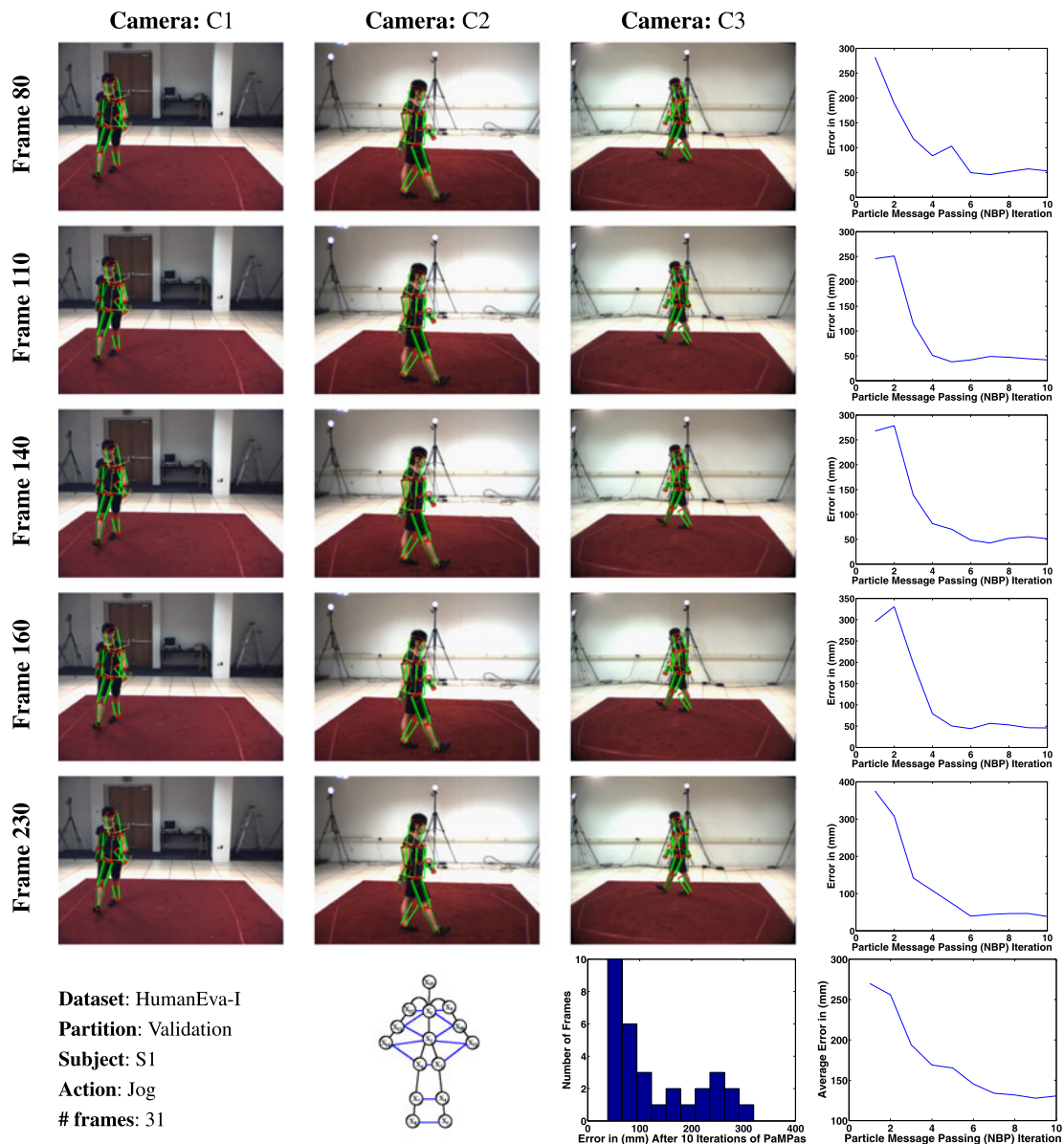


Fig. 12 Pose estimation using 15-part loose-limbed body model. See caption for Fig. 11

Table 2 Summary of pose estimation performance using loose-limbed body model. Examples of these results are illustrated and analyzed in more detail in Figs. 11 and 12

Subject	S1	S2	S1	S2	S1
Action	Walking	Walking	Jog	Walking	Jog
Number of frames	57	40	31	39	31
Model	10-part	10-part	10-part	15-part	15-part
Mean error in mm	89.7	161.6	83.7	158.5	130.5
Median error in mm	60.2	117.5	62.8	89.3	93.7
Standard deviation of error in mm	71.5	103.6	51.3	132.2	87.4
% of frames with error <80 mm	70.2	27.5	61.3	46.2	45.2
% of frames with error <120 mm	82.5	50.0	87.1	61.5	61.3
% of frames with error \geq 120 mm	17.5	50.0	12.9	38.5	38.7

Table 3 Summary of tracking performance using loose-limbed body model. A subset of these results are illustrated and analyzed in more detail in Figs. 13 and 14

Subject	S2	S1	S1	S2
Action	Walking	Walking	Jog	Walking
Number of frames	400	391	201	213
Model	10-part	15-part	15-part	15-part
Mean error in mm	74	66	77	69
Standard deviation of error in mm	9.95	19.0	20.2	18.8
Average for the model in mm	74	69		
Standard deviation for the model in mm	9.95	19.7		

inference (3 color and 4 greyscale). We also experimented with pose estimation and tracking using 4 and 3 views with similar results. The challenge with the HumanEva-I dataset used here is that, due to poor image quality, simple background subtraction employed by our system often produces poor segmentation of the foreground (see the right column corresponding to camera BW4 in Fig. 6). This would cause significant problems for standard voxel-based 3D tracking methods that require good background subtraction. In such methods noise in background silhouettes leads to holes and extrusions in the voxel-based representation. Our approach is able to deal with this and recovers joint positions that are <80 mm away from the true joint positions 50% of the time on average. Table 2 summarizes the performance.

The bar plot at the bottom of each of the Figs. 11 and 12 shows the histogram of errors for all tested frames. In most frames, the error falls below the 120 mm level (see Figs. 11, and 12) that we consider to be “acceptable”. The bottom 3 rows of the Table 2 show the percentage of frames where the error falls below the defined levels of <80 mm or <120 mm. The worst performance is on the sequence of subject S2 walking with the 10-part model, where an acceptable estimate the pose (below 120 mm) is found in only 50% of the frames.

9.2 Pose Estimation During Tracking

We also evaluate the performance of our approach in the context of tracking. A weak temporal consistency model is used to propagate results from one frame to the next (see description in Sect. 8.1) to help focus the inference. In this paradigm we assume that limbs at the next frame are sufficiently close to the correctly estimated pose at the previous frame. This provides a proposal distribution that focuses a fraction of samples in locations where the limbs were previously found. Since typically the previous frame estimates are sufficiently close to the solution at the current frame, we only run PAMPAS for 2 message passing iterations (instead of 10). Several results are illustrated in Figs. 13 and 14. The approach is tested on a total of 1205 frames; 400 frames

using a 10-part model and 805 frames using 15-part loose-limbed body model. The average performance over the sequence ranges between 59–77 mm in all cases (see summary of results in Table 3). The mean error and standard deviation is significantly reduced compared with the static pose estimation scenario.

9.3 Comparison with Annealed Particle Filter

We compare our tracking results with those obtained using a relatively standard tracking algorithm based on an Annealed Particle Filter (APF) (Deutscher and Reid 2005). In particular, we make use of the APF algorithm implemented¹⁴ and tested by Balan et al. (2005). In our comparison, the Annealed Particle Filter performs inference using a kinematic tree body model with 15 parts, comparable to our 15-part loose-limbed body model; the resulting state-space parameterization of the pose is $\in \mathbb{R}^{40}$, corresponding to the global position and orientation of the torso in 3D and 36 joint angles. The implementation of the APF uses a likelihood function that is comparable to the one described above and incorporates both silhouette and edge information (see Balan et al. 2005 for details). Unlike the original APF algorithm proposed by Deutscher et al. (2000), the variant of Balan et al. (2005) is also able to incorporate temporal and structural priors, that ensure that parts do not penetrate each other and that joint angles are within the allowable limits. In Fig. 15 we compare our model with three variants of the APF algorithm: the generic APF with interpenetration constraints and generic joint limits with (i) 250 and (ii) 500 particles, and (iii) an APF algorithm that in addition encodes action-specific joint limits and incorporates a temporal prior. In all cases the APF uses 5-layers for annealing and requires an initial pose at the first frame to start the inference; the initial pose was obtained from ground truth motion capture data.

The loose-limbed body model in both sequences outperforms the generic APF algorithms and performs comparably

¹⁴Implementation of APF is courtesy of Alexandru Balan and is freely distributed from <http://www.cs.brown.edu/alb/software.htm>.

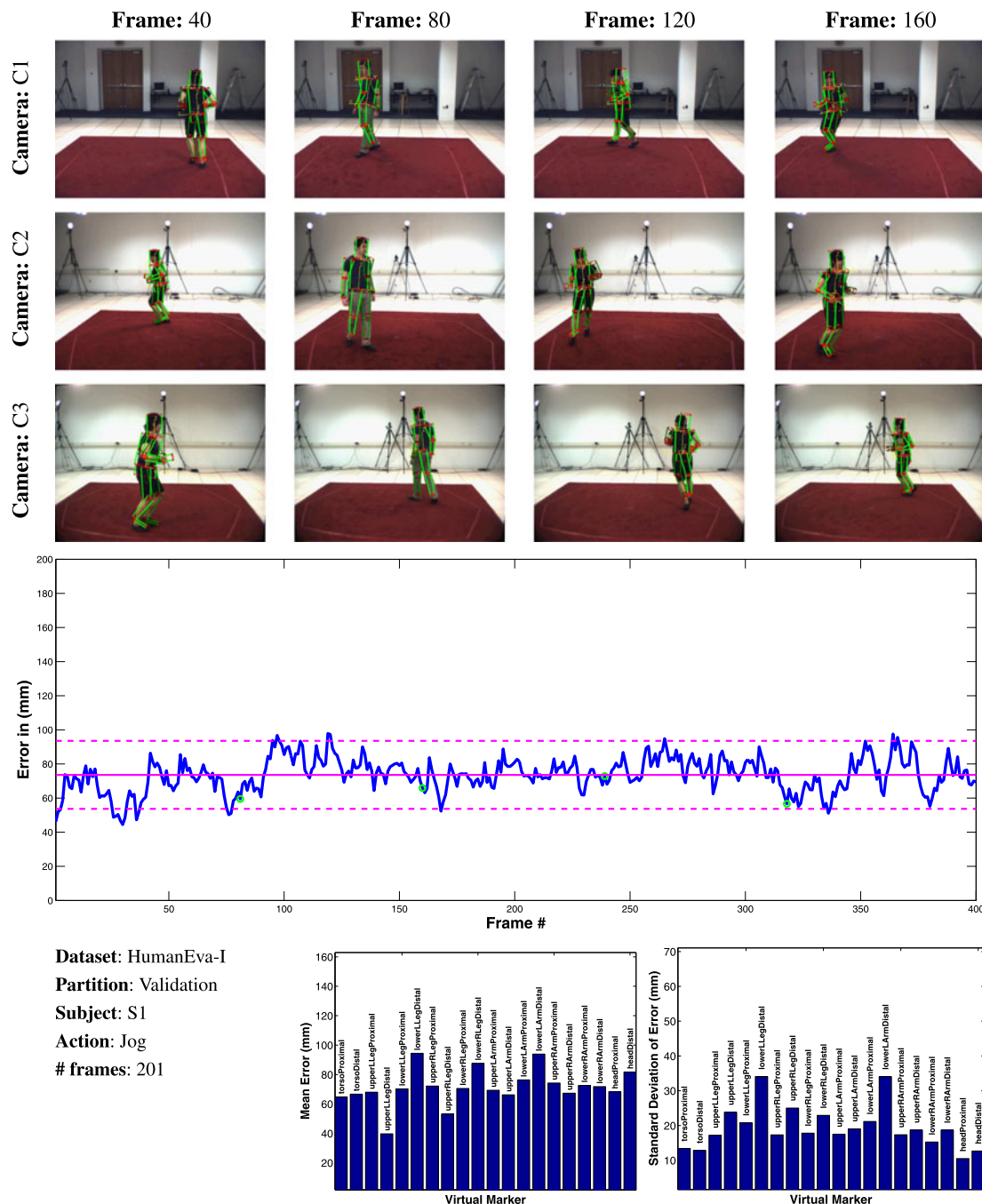


Fig. 13 Tracking using 15-part loose-limbed body model. Results of tracking pose over a multi-ocular sequence from the HumanEva-I dataset are shown for a number of frames. The *top three rows* show the final result in terms of the most likely sample from the marginal distribution over each part after 2 iterations of PAMPAS. The results are projected into 3 synchronized views for clarity (7 views were used for inference). The plot in the *second to last row* shows per frame error (in *blue*) for all frames used for testing. The

mean error computed over the entire sequence and $\pm 2\sigma$ are shown in *solid* and *dashed magenta* respectively. Frames selected automatically and temporally equidistantly to visually illustrate performance (*top three rows*), are designated by *green circles* on the graph. The *last row* illustrates an alternative analysis of the error by showing statistics for individual virtual markers, with the mean on the left and the standard deviation on the right, averaged over the entire sequence

to the action-specific APF variant (see Fig. 15). In all cases, however, the variance for the estimates obtained using APF are lower than those obtained using our loose-limbed body

model. This is not surprising, considering the nature of inference employed in the loose-limbed body model, where the pose at the previous time instant is simply a proposal for

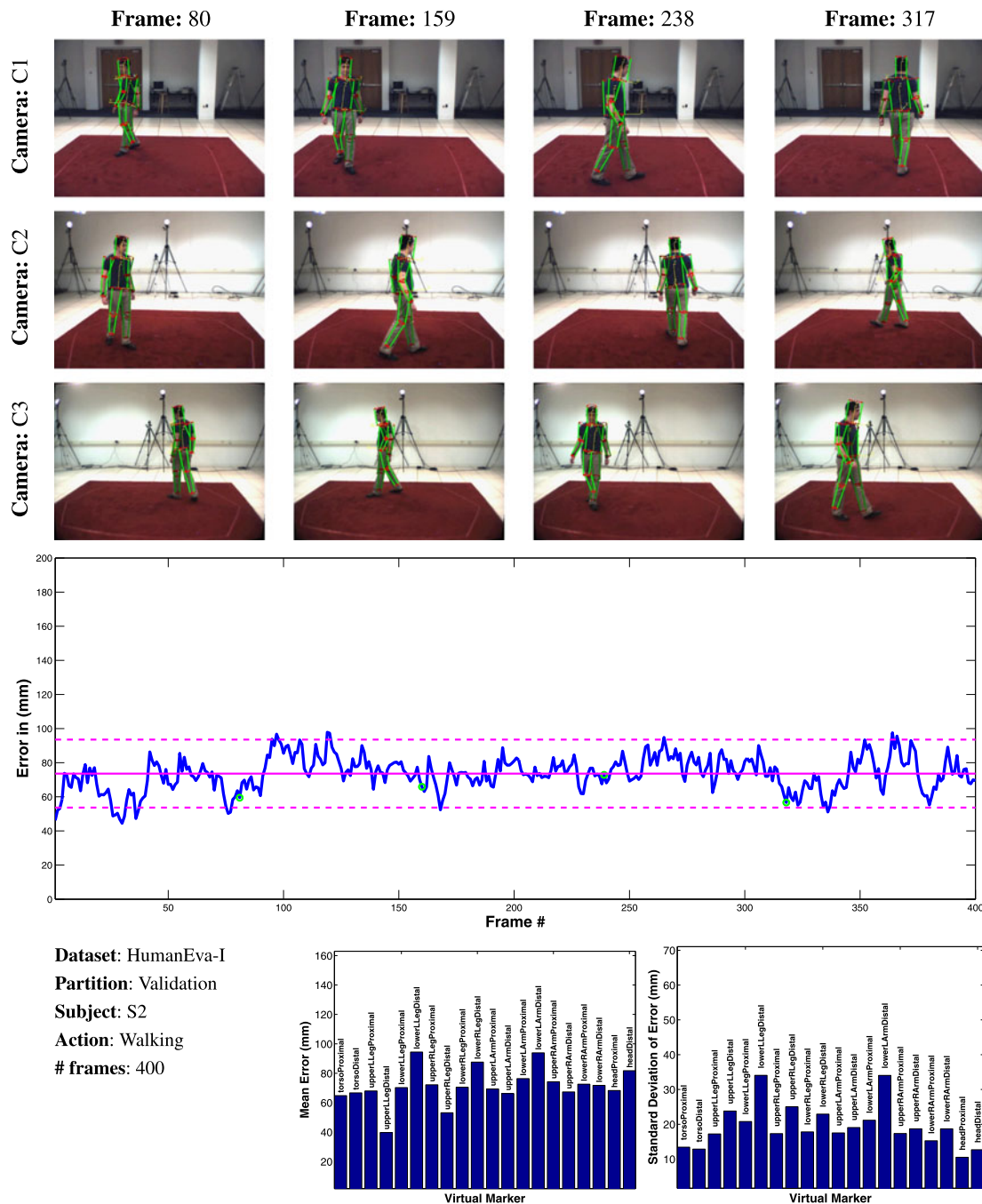


Fig. 14 Tracking using 10-part loose-limbed body model. See Fig. 13 caption

inference at the next time frame. At each frame the loose-limbed model is essentially solving the detection problem while incorporating proposals from the previous time instant. This type of inference aids recovery from intermittent tracking failures. In contrast the APF implementations here have fairly strong temporal dependencies from one frame to the next which smooth the posterior at the expense of introducing persistent failures (i.e. when failure occurs it usually persists for many, if not all, frames). More importantly, our

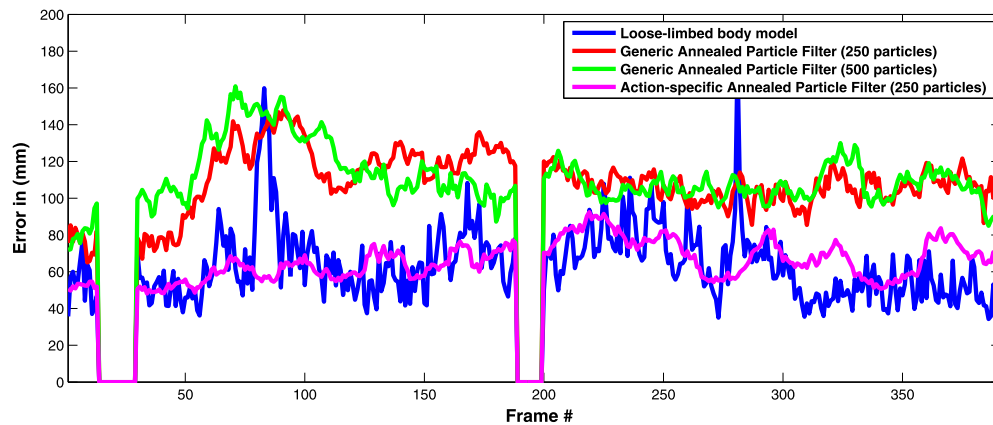
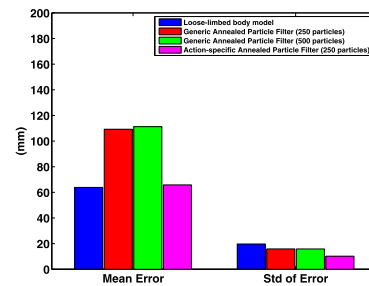
algorithm is fully automatic and is able to estimate the pose at the first frame as well as track it over time.

9.4 Comparison with Other Methods

There are a number of methods published in the literature that utilize HUMANEVA datasets (both HUMANEVA-I and HUMANEVA-II) for quantitative evaluation of performance. While direct comparisons are still difficult, we can draw

Dataset: HumanEva-I
Partition: Validation
Subject: S1
Action: Walking
frames: 391

	Error	
	Mean <i>mm</i>	Std <i>mm</i>
Loose-Limbed Model	66	19.0
Generic APF (250)	109	15.8
Generic APF (500)	111	15.8
Action-specific APF (250)	66	10.2



Dataset: HumanEva-I
Partition: Validation
Subject: S2
Action: Walking
frames: 213

	Error	
	Mean <i>mm</i>	Std <i>mm</i>
Loose-Limbed Model	69	18.8
Generic APF (250)	87	14.8
Generic APF (500)	81	12.2
Action-specific APF (250)	70	7.3

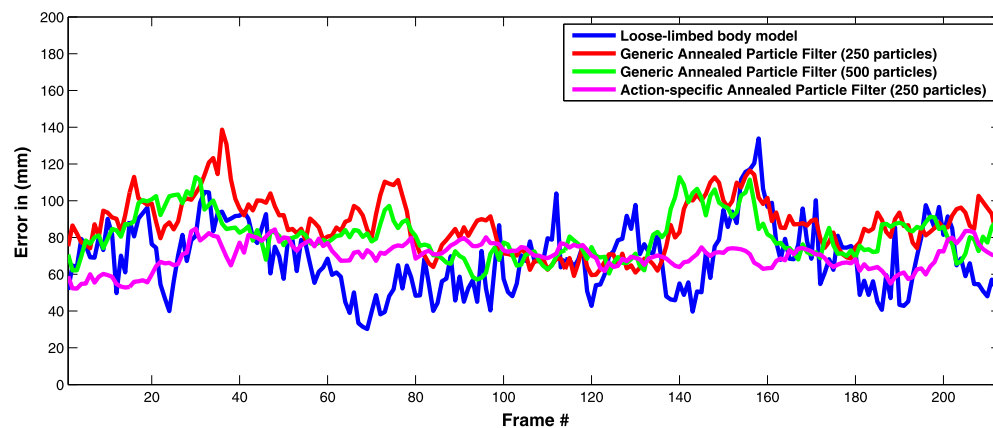
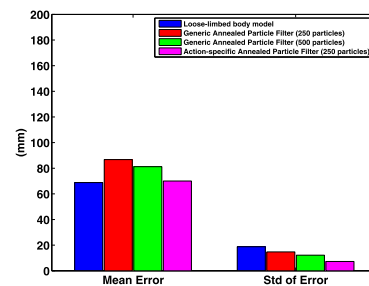


Fig. 15 Comparison with annealed particle filter. Tracking results produced by the loose-limbed body model and by an Annealed Particle Filter (APF) are illustrated and compared on two sequences. Three variants of the APF are implemented for comparison (see text for details). All methods use comparable 15-part body models and likelihood functions; for the APF this results in a kinematic tree model with 40 parameters. The *top row*, in each experiment, denotes the sequence used

(*left*) and the performance statistics for the various methods, averaged over the length of the entire sequence, in both table (*middle*) and bar plot (*right*) form. The *bottom plot*, for each experiment, illustrates the performance over the entire sequence. The two regions where the error goes to 0 correspond to frames for which the ground truth poses are invalid according to HumanEva (these regions are omitted for average error computations)

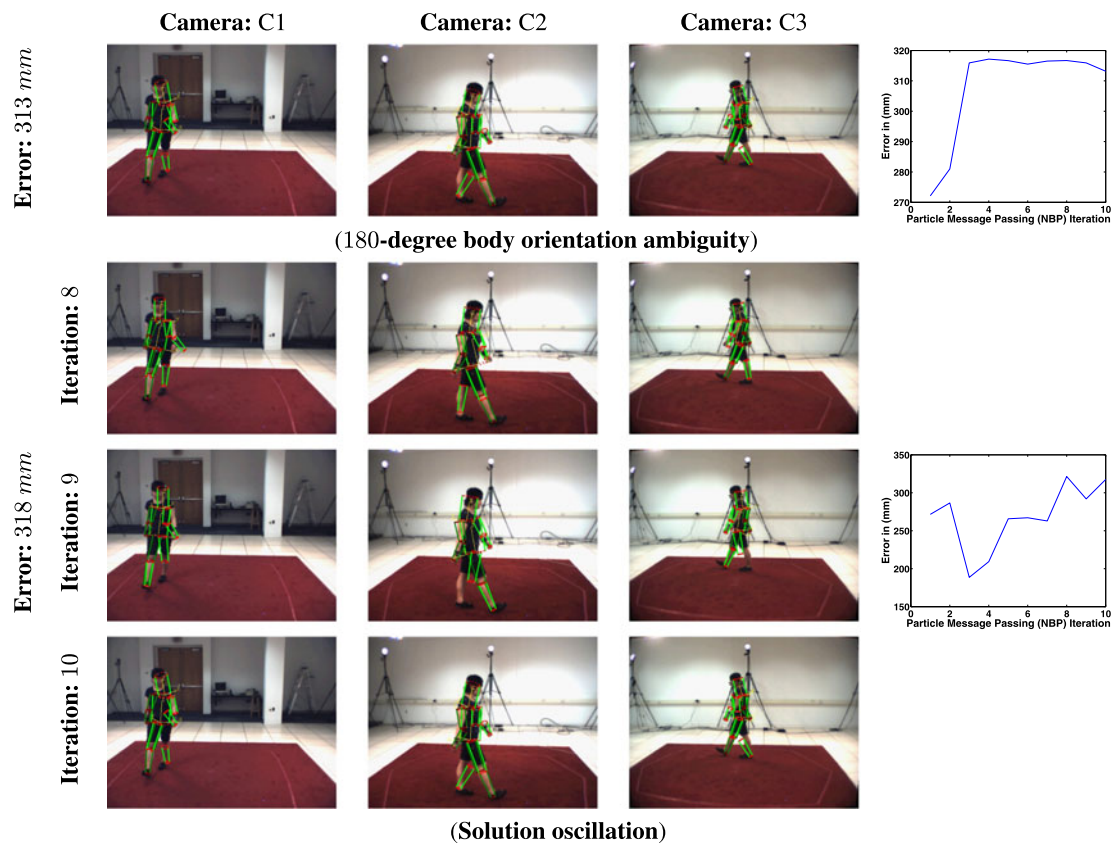


Fig. 16 Failure modes. One of the most common failure modes of our approach is due to the rotational symmetry of the body. Since the only detector that is sensitive to the overall orientation of the body is the head, in the absence of reliable head detection (a common scenario in practice), the overall pose of the body can potentially be recovered pointing in the opposite direction (*top*). In the figure, dark limbs correspond to the left side of the model. This is particularly common in the scenarios where articulations, which also provide hints as to the overall orientation of the body, are minimal. The *plot on the right* illustrates

the error as a function of message passing iterations and suggests that BP has converged. In this case it has converged to an incorrect solution (a local maximum of the joint probability function). In other cases (*bottom*) the lack of a clear body orientation (or lack of a good match to the image data in general) may lead to oscillations between solutions in the inference. In particular, notice how the legs assume similar configurations at iteration 8 and 10 and a competing configuration at iteration 9. This is a problem known in the general loopy graphical model literature

some conclusions. Several other 3D generative tracking approaches have been proposed and report errors (on the same walking sequences) on the order of 140–156 mm for tracking using multiple views (Xu and Li 2007) and 187 mm for tracking using monocular observations (Li et al. 2007); both of these utilize strong motion priors to constrain the inference and require manual initialization.

The best performance on this dataset, to date, has been reported using discriminative methods trained to work directly with monocular observations; the best reported performance on walking sequences is 26–31 mm (Lee and Elgammal 2007). The method in Lee and Elgammal (2007) assumes knowledge of the subject's identity and requires the first gait cycle from the sequence for training. Less restrictive discriminative models have also been proposed (Bo et al. 2008; Urtasun and Darrell 2008), which in comparison exhibit slightly inferior performance to Lee and Elgammal (2007) (e.g., in Bo et al. (2008) performance across entire HU-

MANEVA dataset ranges between 31.1 and 48.5 mm depending on the features) but still superior to our method. In all cases, however, these approaches are only able to recover relative pose of the body in 3D. Even though these methods tend to perform well on this dataset, there is evidence that this is in part due to overfitting which comes at expense of generalization (e.g. in Poppe (2007b) the error more than doubles once the models trained on HUMANEVA-I sequences are tested on HUMANEVA-II data).

9.5 Analysis of Failures

In the context of pose estimation, while our approach performs reasonably well in most frames, it does occasionally suffer from failures. In this section we analyze the common failure modes (see Fig. 16). Intuitively, our approach iteratively estimates the plausible domain for the position and orientation of limbs and the distribution over that domain.

Table 4 Runtime speed of inference. All numbers are reported per frame unless otherwise stated. These results were measured on a single processor 2.0 GHz machine with 1 GB of RAM

	10-part model	15-part model
Part Detectors	47 s	45 s
Message Passing	20 s/iter	84 s/iter
Belief Estimation	10 s	64 s
<i>Total</i>		
Pose Estimation	259 s	948 s
Tracking	99 s	274 s

Part detectors are critical in providing the initial guess to the plausible portion of the state space (domain) that should be considered. However, part detectors, are not always precise and hence the algorithm can become trapped in local optima. In particular, since the left and right limbs are indistinguishable, the only detector that gives clues as to overall orientation (view) of the body is the head detector. In the absence of reliable head estimates (a common scenario in practice due to the poor image quality and sparse placement of cameras), the model suffers from a 180 degree ambiguity. This ambiguity, illustrated in Fig. 16 (top), can be resolved to some extent by the articulation of the body itself. Joints that have asymmetric degrees of freedom, modeled in our case by kinematic constraints, can help to resolve this ambiguity in some cases. In other cases, however, where articulation is minimal, they do not provide reliable distinguishing power (see Fig. 16 (top)). Intuitively, the 15-part body model should help in these cases, because feet provide additional constraints on the overall orientation of the body. Unfortunately, floor shadows make it challenging to find feet reliably. Hence, we have observed limited performance benefit from this more refined model.

It is also worth mentioning that since we work with loopy graphical models, in general our method is not guaranteed to converge and in the case of convergence is only guaranteed to converge to a local optimum. If the model does not converge, which in our experience happens infrequently, it can oscillate between solutions as illustrated in Fig. 16 (bottom).

9.6 Analysis of Runtime Speed

The method described here is implemented in C++ and runs significantly slower than frame rate. The overall performance for a typical run of each one of the two body models and modes of operation is summarized in Table 4. The method however is easily amenable to parallel implementation on a multi-core architecture.

Part detectors present a fixed overhead for each frame, that roughly amounts to 45–50 seconds for 7 views. Notice that since part detectors operate on pairs of views, their runtime in general scales exponentially with the number of

views available. The rest of the time spent in PAMPAS, consists of a number of message passing iterations and a single belief estimation stage at the end. The majority of time in both stages is spent drawing samples from the product of messages (represented by Gaussian mixtures).

It is worth noting that the additional computation imposed by the 15-part model is not due to the change in topology or connectivity of the graphical model, rather it is due to the representation of messages (number of samples) we choose to employ for forming messages going out of the added nodes. Table 5 (Appendix B) lists the number mixture components needed to represent each message. Consider forming a message going from the left calf to the right calf in the 15-part model, for example. This involves drawing samples from the product of two messages represented by 2401-component and 801-component Gaussian mixtures; conversely, the most expensive message to form in the 10-part model involves sampling from the product of two messages represented by 2401-component and 201-component Gaussian mixtures (e.g., message from left to right thigh). Since the complexity of sampling is proportional to the number of components in the mixtures involved, this should result in 4 times slower formation of the message for the 15-part model, which matches the observed runtime (20 s/iter versus 84 s/iter). The belief estimation stage, in addition, requires computation of beliefs for 50% more nodes (parts).

10 Conclusion and Discussion

In this paper we have presented a probabilistic method for fully automatic 3D human pose estimation and tracking in multi-camera images. Like recent approaches in 2D pose estimation, we formulate the body in terms of a graphical model and use belief propagation for inference. In contrast to 2D methods, discretization of the 3D parameter space is not practical and we instead exploit a non-parametric form of BP. We find that a loose-limbed body 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 non-parametric belief propagation. The belief propagation framework allows us to avoid distinguishing between pose estimation and tracking, but instead to use bottom-up part detectors to enhance robustness of the motion estimation and provide “initialization” cues at every frame in a sequence.

The main advantages of our approach are: bottom-up processes are integrated at every frame allowing automatic initialization and recovery from transient tracking failures; it admits interpenetration constraints between body parts as well as temporal constraints which both introduce loops into the graphical model; and the conditional probabilities between limbs in space are learned from training data. In quantitative experiments we find that the method provides accu-

rate estimates of 3D body pose that meet or exceed the performance of a traditional kinematic tree-based model.

Fully automatic 3D body pose estimation is an important step toward practical systems that can function outside the laboratory setting. Our model, for example, can be used to initialize a standard kinematic tree model. The same optimization methods employed here can also be applied to 2D human pose estimation (Sigal and Black 2006b). For example, we have used similar methods to compute 2D pose in monocular sequences and then have used the 2D poses to initialize 3D models for 3D monocular person tracking (Sigal and Black 2006a).

Future work should develop more reliable part detectors with which to find body parts in complex cluttered environments (e.g. Andriluka et al. 2009). More powerful likelihood models, that go beyond silhouettes and edges, would improve performance. Current likelihoods are insensitive to rotations along the limb axes and hence limit the ability of our model to estimate these degrees of freedom in absence of articulations present in the corresponding lower extremities. For example, the twist of the upper arm can be estimated reliably when the elbow is bent, but not when the arm is straight; in the latter case the rotation is only constrained by learned prior joint limits. Richer likelihoods that take into account optical flow, illumination and shading (Guan et al. 2009), and the temporal persistence of the image appearance of body parts would likely help to resolve such ambiguities (recent examples of work along these lines include (Andriluka et al. 2009; Eichner and Ferrari 2009)). A multi-core, parallel, implementation of these methods should be pursued.

Acknowledgements This work was supported in part by NSF grants IIS-0534858 and IIS-0535075, NSF IGERT award 9870676, and gifts from Intel Corporation and Honda Research Institute. We would like to thank Alexandru Balan for annealed particle filter code; Ming-Hsuan Yang, Rui Li, Alexandru Balan, Stefan Roth and Payman Yadollahpour for help in data collection and post-processing. We also would like to thank Stan Sclaroff for making the color video capture equipment available for this effort. Finally, would like to thank Konstantin Rodyushkin, Alexander Kuranov, Victor Erubimov, Oleg Maslov and the remainder of Nizhny Novgorod Intel Research team for their contributions to the software.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Appendix A: Learning of Kinematic Conditionals

The details of the learning procedure described in Sect. 4.1.2 are provided here. Given the state $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{q}_i]^T$ that represents the configuration of the body part i , where $\mathbf{x}_i = [\mathbf{x}_{x,i}, \mathbf{x}_{y,i}, \mathbf{x}_{z,i}] \in \mathbb{R}^3$ and $\mathbf{q}_i = [q_{x,i}, q_{y,i}, q_{z,i}, q_{w,i}] \in$

$\text{SO}(3)$ ($\|\mathbf{q}_i\| = 1$) are the 3D position and the unit quaternion orientation of the part, we re-parameterize the state in terms of a homogeneous 3D object-to-world matrix transformation as follows,

$$\mathbb{X}_i = H(\mathbf{X}_i) = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where

$$\begin{aligned} a_{1,1} &= 1 - 2q_{y,i}^2 - 2q_{z,i}^2, \\ a_{1,2} &= 2q_{x,i}q_{y,i} - 2q_{w,i}q_{z,i}, \\ a_{1,3} &= 2q_{x,i}q_{z,i} + 2q_{w,i}q_{y,i}, \\ a_{2,1} &= 2q_{x,i}q_{y,i} + 2q_{w,i}q_{z,i}, \\ a_{2,2} &= 1 - 2q_{x,i}^2 - 2q_{z,i}^2, \\ a_{2,3} &= 2q_{y,i}q_{z,i} + 2q_{w,i}q_{x,i}, \\ a_{3,1} &= 2q_{x,i}q_{z,i} - 2q_{w,i}q_{y,i}, \\ a_{3,2} &= 2q_{y,i}q_{z,i} - 2q_{w,i}q_{x,i}, \\ a_{3,3} &= 1 - 2q_{x,i}^2 - 2q_{y,i}^2, \\ a_{1,4} &= \mathbf{x}_{x,i}, \quad a_{2,4} = \mathbf{x}_{y,i}, \quad a_{3,4} = \mathbf{x}_{z,i}. \end{aligned}$$

The corresponding inverse transformation $H^{-1}(\cdot)$ that maps back from the 3D object-to-world matrix to our state-space parameterization is somewhat more involved. If the trace, $\text{tr}(\mathbb{X}_i) \equiv a_{1,1} + a_{2,2} + a_{3,3} + 1$, where $a_{i,j}$ is the i -th row and j -th column of a 4×4 homogenized matrix \mathbb{X}_i , is ≥ 0 , then the following simple calculation would define the inverse,

$$\begin{aligned} \mathbf{X}_i &= H^{-1}(\mathbb{X}_i) \\ &= \begin{bmatrix} a_{1,4} & a_{2,4} & a_{3,4} & \frac{(a_{3,2} - a_{2,3})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} \frac{(a_{1,3} - a_{3,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} \\ \frac{(a_{2,1} - a_{1,2})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & \frac{\sqrt{\text{tr}(\mathbb{X}_i)}}{2} \end{bmatrix}^T. \end{aligned} \quad (28)$$

Otherwise, if $\text{tr}(\mathbb{X}_i) \leq 0$, one must look at the major diagonal element and apply the respective inverse transform as follows,

$$\mathbf{X}_i = H^{-1}(\mathbb{X}_i) = \begin{cases} \mathbf{b}_1 & \text{if } a_{1,1} \geq a_{2,2} \text{ and } a_{1,1} \geq a_{3,3}, \\ \mathbf{b}_2 & \text{if } a_{2,2} \geq a_{1,1} \text{ and } a_{2,2} \geq a_{3,3}, \\ \mathbf{b}_3 & \text{if } a_{3,3} \geq a_{1,1} \text{ and } a_{3,3} \geq a_{2,2} \end{cases}$$

where

$$\begin{aligned} \mathbf{b}_1 &= \begin{bmatrix} a_{1,4} & a_{2,4} & a_{3,4} & \frac{\sqrt{\text{tr}(\mathbb{X}_i)}}{2} \frac{(a_{1,2} - a_{2,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} \\ \frac{(a_{1,3} - a_{3,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & \frac{(a_{2,3} - a_{3,2})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} \end{bmatrix}^T, \end{aligned}$$

$$\mathbf{b}_2 = \begin{bmatrix} a_{1,4} & a_{2,4} & a_{3,4} & \frac{(a_{1,2} - a_{2,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & \frac{\sqrt{\text{tr}(\mathbb{X}_i)}}{2} \\ \frac{(a_{2,3} - a_{3,2})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & \frac{(a_{1,3} - a_{3,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & & & \end{bmatrix}^T,$$

$$\mathbf{b}_3 = \begin{bmatrix} a_{1,4} & a_{2,4} & a_{3,4} & \frac{(a_{1,3} - a_{3,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & \frac{(a_{2,3} - a_{3,2})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} \\ \frac{\sqrt{\text{tr}(\mathbb{X}_i)}}{2} & \frac{(a_{1,2} - a_{2,1})}{2\sqrt{\text{tr}(\mathbb{X}_i)}} & & & \end{bmatrix}^T.$$

It can be shown that indeed $\mathbf{X}_i = H^{-1}(H(\mathbf{X}_i))$.

Section 4.1.2 also omits the details of the transformations $F_{ijm}(\mathbf{X}_i)$ and $G_{ijm}(\mathbf{X}_i)$ for the mean and covariance of the learned Gaussian mixture components, μ_{ijm} and Λ_{ijm} respectively (see (6)). Formally, we can write

$$F_{ijm}(\mathbf{X}_i) = R(\mathbf{X}_i) * \mu_{ijm} + T(\mathbf{X}_i) \quad (29)$$

and

$$G_{ijm}(\mathbf{X}_i) = \left(\Lambda_{ijm}^{-1} * R(\mathbf{X}_i) \right)^{-1}, \quad (30)$$

where $T(\mathbf{X}_i) = [\mathbf{x}_i, 0, 0, 0, 0]^T$ and

$$R(\mathbf{X}_i) = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 & 0 & 0 & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{w,i} & -q_{x,i} & -q_{y,i} & -q_{z,i} \\ 0 & 0 & 0 & -q_{x,i} & q_{w,i} & -q_{z,i} & q_{y,i} \\ 0 & 0 & 0 & -q_{y,i} & -q_{z,i} & q_{w,i} & -q_{x,i} \\ 0 & 0 & 0 & -q_{z,i} & q_{y,i} & -q_{x,i} & q_{w,i} \end{bmatrix}.$$

Intuitively, for a given value of $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{q}_i]^T$, the top-left block will transform the translation component of the mean and covariance via a rotation matrix defined by the \mathbf{q}_i and the bottom-right block will transform the quaternion rotation component of the mean and covariance via the Grassman product.

Appendix B: PAMPAS Implementation Details

The algorithm outlined in Sect. 7.1 leaves open implementation details such as (1) how many particles to use for Monte Carlo approximation of each message and (2) what order to use in updating the messages. For completeness we address these implementation details and how they effect the inference below.

Number of Samples The number of particles/samples used to approximate messages has a significant effect on the run time of the algorithm. While the basic Particle Message

Passing algorithm assumes that all messages are approximated using the same number of N samples, we found this to be sub-optimal. In particular, we found that messages going out of the nodes that are highly connected (e.g. the torso) are often more compact and require fewer samples to represent adequately; alternatively, messages that correspond to outer nodes in the graph, that have fewer connections, need more samples to be adequately represented. Hence, we derived a heuristic to determine the number of samples required to represent the message based on the degree of the node sending the message. In particular, for all experiments we used the number of samples illustrated in Table 5 to approximate the corresponding messages.

Note that penetration messages, due to their non-Gaussian form, are simply treated in the PAMPAS framework as continuous functions. Automatically deriving the number of samples required for each message would clearly be of benefit, however, this is hard to do in general, since the number of samples must be a function of the overall graph topology, importance functions employed for Monte Carlo integration, and distributions of all involved variables.

Message Passing Scheduling While in theory the message passing schedule (order) in BP does not matter, in practice it has been shown that it can effect the convergence properties significantly. It is a well-known empirical observation that asynchronous message passing algorithms, where messages are updated sequentially, generally converge faster and more often than the synchronous variant, where all messages are updated in parallel. In practice, however, synchronous variants are often used, perhaps due to ease of implementation.

One of the standard asynchronous message schedules can be derived by computing a minimum spanning tree over the graph and updating messages according to the tree-structure rules (Wainwright et al. 2001). The spanning tree, however, may not be unique. In this case one must either choose a tree and a fixed asynchronous schedule for that tree, or for every iteration of BP randomly pick a minimum spanning tree and a corresponding schedule.

For simplicity, we use a fixed asynchronous message passing schedule with a predefined minimum spanning tree. This results in messages being sent from the outer extremities inward toward the torso and then back out (from the torso to outer extremities). We also first propagate the kinematic messages and then the penetration messages. For pose estimation we run PAMPAS for 10 message passing iterations per frame (with convergence often achieved in 5–6 iterations), and for tracking (discussed in Sect. 8.2) for only 2 message passing iterations per frame. The underlying assumption being that in the tracking framework the pose is likely to be relatively well constrained by the estimate from the previous time frame, and hence PAMPAS often converges faster.

Table 5 Implementation Details. Listed are the sampling proportions and representation for potentials for various body parts in the loose-limbed body model. Note that kinematic messages are represented using Gaussian mixture (MoG) densities with the specified number of components (equal to the number of sample times the number of mix-

tures in the potential plus one Gaussian outlier); penetration messages are represented using continuous functions that take unnormalized form of 1—Gaussian mixture with specified number of components (equal to the number of samples)

Node i	# of samples	# of Mixtures in potential	Message representation
torso	50	Kinematic: 4 Penetration: 1	$m_{ij}^K(\mathbf{X}_j) = 201$ MoG $m_{ij}^P(\mathbf{X}_j) = 1-(50$ MoG)
head, thighs, upper arms	200	Kinematic: 4 Penetration: 1	$m_{ij}^K(\mathbf{X}_j) = 801$ MoG $m_{ij}^P(\mathbf{X}_j) = 1-(200$ MoG)
calves, lower arms	800	Kinematic: 4 Penetration: 1	$m_{ij}^K(\mathbf{X}_j) = 2401$ MoG $m_{ij}^P(\mathbf{X}_j) = 1-(800$ MoG)
(In addition for the 15-part model)			
hands, feet	800	Kinematic: 4 Penetration: 1	$m_{ij}^K(\mathbf{X}_j) = 2401$ MoG $m_{ij}^P(\mathbf{X}_j) = 1-(800$ MoG)

In general, however, better convergence may be achieved by randomly choosing a spanning tree (assuming more than one exists, e.g. the root of the minimum spanning tree for the 15-part loose-limbed body model can either be the torso or the pelvis node) and a corresponding message passing schedule. More recently a new informative message scheduling approach (Elidan et al. 2006) has also been proposed.

Simulated Annealing Annealing involves gradually changing the objective function in a way that facilitates convergence. In the case of Particle Message Passing (PAMPAS) one can anneal the likelihood, the potentials or both.

The Markov-chain-based method of simulated annealing was developed initially in Kirkpatrick et al. (1982) and later adapted for articulated particle filtering in Deutscher et al. (2000) and Gall et al. (2006) as a way of handling multiple modes in a stochastic optimization context. The method employs a series of distributions, with probability densities given by $p_0(\mathbf{X})$ to $p_M(\mathbf{X})$, in which each $p_m(\mathbf{X})$, $m \in [0, \dots, M]$ differs only slightly from $p_{m+1}(\mathbf{X})$. In this context the samples need to be drawn from $p_0(\mathbf{X})$ and the $p_m(\mathbf{X})$'s such that in $p_M(\mathbf{X})$ the movement between all regions of the search space is allowed. The usual method is to set $p_m(\mathbf{X}) \propto [p_0(\mathbf{X})]^{\beta_m}$, for $1 = \beta_0 > \beta_1 > \dots > \beta_M$.

In our experiments, we found that annealing the likelihood as a function of BP iterations worked well. We set $\beta_m = \beta_{m+1}\kappa$, where m is the iteration of BP and $0 < \kappa < 1$ is a constant. Annealing of potentials at the same time (perhaps similarly as a function of BP iterations), is also possible, and would lead to stronger enforcement of joint constraints.

References

- Agarwal, A., & Triggs, B. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1), 44–58.
- Andriluka, M., Roth, S., & Schiele, B. (2009). Pictorial structures revisited: people detection and articulated pose estimation. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.
- Balan, A., Sigal, L., & Black, M. J. (2005). A quantitative evaluation of video-based 3D person tracking. In *IEEE workshop on visual surveillance and performance evaluation of tracking and surveillance* (pp. 349–356). October 2005.
- Banerjee, A., Dhillon, I. S., Ghosh, J., & Sra, S. (2005). Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6, 1345–1382.
- Bergtholdt, M., Kappes, J., Schmidt, S., & Schnorr, C. (2010). A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 87(1–2), 93–117.
- Bhatia, S., Sigal, L., Isard, M., & Black, M. J. (2004). 3D human limb detection using space carving and multi-view eigen models. In *IEEE Workshop on articulated and nonrigid motion, CVPR'04 CDROM proceedings*.
- Bregler, C., & Malik, J. (1998). Tracking people with twists and exponential maps. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (pp. 8–15).
- Bo, L., Sminchisescu, C., Kanaujia, A., & Metaxas, D. (2008). Fast algorithms for large scale conditional 3D prediction. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 679–714.
- Cham, T.-J., & Rehg, J. (1999). A multiple hypothesis approach to figure tracking. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 239–245).
- Cheung, G. K. M., Baker, S., & Kanade, T. (2003). Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 77–84).

- Choo, K., & Fleet, D. J. (2001). People tracking with hybrid Monte Carlo. In *IEEE international conference on computer vision (ICCV)* (Vol. 2, pp. 321–328).
- Cochran, W. G. (1977). *Sampling techniques*. New York: Wiley.
- Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603–619.
- Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42, 393–405.
- Corazza, S., Muendermann, L., Chaudhari, A., Demattio, T., Cobelli, C., & Andriacchi, T. (2006). A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach. *Annals of Biomedical Engineering* 34(6), 1019–1029.
- Deutscher, J., & Reid, I. D. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision* 61(2), 185–205.
- Deutscher, J., Blake, A., & Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 126–133).
- Deutscher, J., Isard, M., & McCormick, J. (2002). Automatic camera calibration from a single Manhattan image. In *European conference on computer vision (ECCV)* (Vol. 4, pp. 175–188).
- Doucet, A., Godsill, S. J., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 197–208.
- Doucet, A., de Freitas, N., & Gordon, N. (2001). Sequential Monte Carlo methods in practice. In *Statistics for engineering and information sciences*. Berlin: Springer.
- Eichner, M., & Ferrari, V. (2009). Better appearance models for pictorial structures. In *British machine vision conference (BMVC)*.
- Elgammal, A., & Lee, C. (2004). Inferring 3D body pose from silhouettes using activity manifold learning. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 681–688).
- Elidan, G., McGraw, I., & Koller, D. (2006). Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the twenty-second conference on uncertainty in AI (UAI)*, July 2006.
- Felzenszwalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision* 61(1), 55–79.
- Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 264–271).
- Fischler, M., & Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1), 67–92.
- Foley, J., van Dam, A., Feiner, S., & Hughes, J. (1990). *Computer graphics: Principles and practice*. Reading: Addison Wesley. ISBN:0-201-12110-7.
- Forsyth, D. A., Arikan, O., Ikemoto, L., O'Brien, J., & Ramanan, D. (2006). *Computational studies of human motion: Part 1, tracking and motion synthesis*. ISBN:1-933019-30-1, 178 pp.
- Gall, J., Potthoff, J., Schnoerr, C., Rosenhahn, B., & Seidel, H.-P. (2006). *Interacting annealing particle filters: Mathematics and a recipe for applications* (Technical Report MPI-I-2006-4-009). Saarbrücken, Germany, September 2006.
- Gall, J., Rosenhahn, B., & Seidel, H.-P. (2007). Clustered stochastic optimization for object recognition and pose estimation. In *LNCS: Vol. 4713. Annual symposium of the German association for pattern recognition (DAGM)* (pp. 32–41).
- Gall, J., Rosenhahn, B., Brox, T., & Seidel, H.-P. (2010). Optimization and filtering for human motion capture—A multi-layer framework. *International Journal of Computer Vision*, 87(1), 75–92.
- Gavrila, D. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1), 82–98.
- Gavrila, D., & Davis, L. (1996). 3-D model-based tracking of humans in action: A multi-view approach. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (pp. 73–80).
- Grauman, K., Shakhnarovich, G., & Darrell, T. (2003). Inferring 3D structure with a statistical image-based shape model. In *IEEE International conference on computer vision (ICCV)* (pp. 641–648).
- Guan, P., Weiss, A., Balan, A., & Black, M. J. (2009). Estimating human shape and pose from a single image. In *IEEE International Conference on computer vision (ICCV)*.
- Hinton, G. E. (1976). Using relaxation to find a puppet. In *Proceeding of the A.I.S.B. Summer conference* (pp. 148–157).
- Hogg, D. C. (1983). Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1, 5–20.
- Horáud, R., Niskanen, M., Dewaele, G., & Boyer, E. (2008). Human motion tracking by registering an articulated surface to 3-D points and normals. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.
- Hua, G., Yang, M.-H., & Wu, Y. (2005). Learning to estimate human pose with data driven belief propagation. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 747–754).
- Ihler, A. T., Sudderth, E. B., Freeman, W. T., & Willsky, A. S. (2003). Efficient multiscale sampling from products of Gaussian mixtures. *Advances in Neural Information Processing Systems*, 16, 1–8.
- Intel Open Source Computer Vision Library. Available at <http://www.intel.com/research/mrl/research/opencv/>.
- Ioffe, S., & Forsyth, D. (2001a). Human tracking with mixtures of trees. In *IEEE international conference on computer vision (ICCV)* (Vol. 1, pp. 690–695).
- Ioffe, S., & Forsyth, D. (2001b). Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1), 45–68.
- Isard, M. (2003). PAMPAS: Real-valued graphical models for computer vision. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 613–620).
- John, V., Ivekovic, S., & Trucco, E. (2009). Articulated human motion tracking with HPSO. In *International conference on computer vision theory and applications (VISSAPP)* (pp. 531–538).
- Jordan, M. I., Sejnowski, T. J., & Poggio, T. (2001). *Graphical models: Foundations of neural computation*. Cambridge: MIT Press.
- Ju, S., Black, M. J., & Yacoob, Y. (1996). Cardboard people: A parameterized model of articulated motion. In *International conference on automatic face and gesture recognition* (pp. 38–44).
- Kakadiaris, I. A., & Metaxas, D. (1996). Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (pp. 81–87).
- Kehl, R., Bray, M., & Gool, L. V. (2005). Full body tracking from multiple views using stochastic sampling. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 129–136).
- Kinoshita, K., Ma, Y., Lao, S., & Kawade, M. (2006). A fast and robust 3D head pose and gaze estimation system. In *International conference on multimodal interfaces (ICMI)* (pp. 137–138).
- Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1982). *Optimisation by simulated annealing* (Technical report). IBM Thomas J. Watson Research Centre, Yorktown Heights, NY, USA.
- Knossow, D., Ronfard, R., & Horaud, R. (2008). Human motion tracking with a kinematic parameterization of extremal contours. *International Journal of Computer Vision*, 79(2), 247–269.

- Koller, D., Lerner, U., & Angelov, D. (1999). A general algorithm for approximate inference and its application to hybrid Bayes nets. In *Proceedings of the 15th annual conference on uncertainty in artificial intelligence* (pp. 324–333).
- Lan, X., & Huttenlocher, D. (2005). Beyond trees: Common factor models for 2D human pose recovery. In *IEEE international conference on computer vision (ICCV)* (pp. 470–477).
- Lee, C.-S., & Elgammal, A. (2007). Modeling view and posture manifold for tracking. In *IEEE international conference on computer vision (ICCV)*.
- Li, R., Yang, M.-H., Sclaroff, S., & Tian, T.-P. (2006). Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers. In *European conference on computer vision (ECCV)* (Vol. 2, pp. 137–150).
- Li, R., Tian, T.-P., & Sclaroff, S. (2007). Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In *IEEE international conference on computer vision (ICCV)*.
- Lu, Z., Perpinan, M. C., & Sminchisescu, C. (2007). People tracking with the Laplacian eigenmaps latent variable model. *Advances in Neural Information Processing Systems (NIPS)*.
- MacCormick, J., & Isard, M. (2000). Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European conference on computer vision (ECCV)* (Vol. 2, pp. 3–19).
- Marr, D., & Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three dimensional structure. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200, 269–294.
- Moeslund, T., & Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3), 231–268.
- Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 326–333).
- Navaratnam, R., Fitzgibbon, A., & Cipolla, R. (2007). Semi-supervised joint manifold learning for multi-valued regression. In *IEEE international conference on computer vision (ICCV)*.
- Nevatia, R., & Binford, T. O. (1973). Structured descriptions of complex objects. In *Proc. 3rd international joint conference on artificial intelligence* (pp. 641–647).
- Opelt, A., Pinz, A., & Zisserman, A. (2006). A boundary-fragment-model for object detection. In *European conference on computer vision (ECCV)* (Vol. 2, pp. 575–588).
- Poppe, R. W. (2007a). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1–2), 4–18.
- Poppe, R. (2007b). Evaluating example-based pose estimation: experiments on the HumanEva sets. In *Workshop on evaluation of articulated human motion and pose estimation (EHuM2)*.
- Ramanan, D., Forsyth, D., & Zisserman, A. (2005). Strike a pose: Tracking people by finding stylized poses. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 271–278).
- Ramanan, D., & Forsyth, D. (2003). Finding and tracking people from the bottom up. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 467–474).
- Rodgers, J., Anguelov, D., Pang, H.-C., & Koller, D. (2006). Object pose detection in range scan data. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 2445–2452).
- Rosales, R., & Sclaroff, S. (2000). Inferring body pose without tracking body parts. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 721–727).
- Rosales, R., & Sclaroff, S. (2002). Learning body pose via specialized maps. *Advances in Neural Information Processing Systems*, 15, 1263–1270.
- Rosenhahn, B., Schmalz, C., Brox, T., Weickert, J., Cremers, D., & Seidel, H.-P. (2008). Markerless motion capture of man-machine interaction. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.
- Shakhnarovich, G., Viola, P., & Darrell, T. (2003). Fast pose estimation with parameter sensitive hashing. In *IEEE international conference on computer vision (ICCV)* (Vol. 2, pp. 750–757).
- Siddiqui, M., & Medioni, G. (2006). Robust real-time upper body limb detection and tracking. In *ACM international workshop on video surveillance & sensor networks (VSSN)*.
- Sidenbladh, H., & Black, M. J. (2003). Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1–3), 183–209.
- Sidenbladh, H., Black, M. J., & Fleet, D. (2000). Stochastic tracking of 3D human figures using 2D image motion. In *European conference on computer vision (ECCV)* (Vol. 2, pp. 702–718).
- Sigal, L., & Black, M. J. (2006a). Predicting 3D people from 2D pictures. In *LNCS: Vol. 4069. AMDO 2006—IV conference on articulated motion and deformable objects*, Mallorca, Spain, July (pp. 185–195).
- Sigal, L., & Black, M. J. (2006b). Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 2041–2048).
- Sigal, L., Zhu, Y., Comaniciu, D., & Black, M. J. (2004a). Tracking complex objects using graphical object models. In *LNCS: Vol. 3417. 1st international workshop on complex motion* (pp. 227–238). Berlin: Springer.
- Sigal, L., Bhatia, S., Roth, S., Black, M. J., & Isard, M. (2004b). Tracking loose-limbed people. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 421–428).
- Sigal, L., Balan, A., & Black, M. J. (2007). Combined discriminative and generative articulated pose and non-rigid shape estimation. *Advances in Neural Information Processing Systems (NIPS)*.
- Sigal, L., Balan, A., & Black, M. J. (2010). HumanEva synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1/2), 4–27.
- Sminchisescu, C., & Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, 22(6), 371–393.
- Sminchisescu, C., Kanaujia, A., Li, Z., & Metaxas, D. (2005). Discriminative density propagation for 3D human motion estimation. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 390–397).
- Sminchisescu, C., Kanaujia, A., & Metaxas, D. (2006). Learning joint top-down and bottom-up processes for 3D visual inference. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 1743–1752).
- Sudderth, E., Ihler, A., Freeman, W., & Willsky, A. (2003). Nonparametric belief propagation. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 605–612).
- Sudderth, E., Mandel, M., Freeman, W., & Willsky, A. (2004). Distributed occlusion reasoning for tracking with nonparametric belief propagation. *Advances in Neural Information Processing Systems*, 17, 1369–1376.
- Sun, J., Shum, H., & Zheng, N. (2002). Stereo matching using belief propagation. In *European conference on computer vision (ECCV)* (pp. 510–524).
- Tian, T.-P., & Sclaroff, S. (2010). Fast globally optimal 2D human detection with loopy graph models. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)*.
- Urtasun, R., & Darrell, T. (2008). Local probabilistic regression for activity-independent human pose inference. In *IEEE Computer*

- Society conference on computer vision and pattern recognition (CVPR).*
- Urtasun, R., Fleet, D. J., & Fua, P. (2006). Gaussian process dynamical models for 3D people tracking. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 238–245).
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 511–518).
- Wachter, S., & Nagel, H. (1999). Tracking of persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3), 174–192.
- Wainwright, M., Jaakkola, T., & Willsky, A. (2001). Tree-based reparameterization for approximate estimation on loopy graphs. *Advances in Neural Information Processing Systems (NIPS)*, 1001–1008.
- Wang, P., & Rehg, J. M. (2006). A modular approach to the analysis and evaluation of particle filters for figure tracking. In *IEEE Computer Society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 790–797).
- Weber, M., Welling, M., & Perona, P. (2000). Unsupervised learning of models for recognition. In *European conference on computer vision (ECCV)* (pp. 18–32).
- Weiss, Y., & Freeman, W. T. (2001). Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13, 2173–2200.
- Wu, Y., Hua, G., & Yu, T. (2003). Tracking articulated body by dynamic Markov network. In *IEEE international conference on computer vision (ICCV)* (pp. 1094–1101).
- Wywill, G., & Kunii, T. L. (1985). A functional model for constructive solid geometry. *The Visual Computer*, 1(1), 3–14.
- Xu, X., & Li, B. (2007). Learning motion correlation for tracking articulated human body with a rao-blackwellised particle filter. In *IEEE international conference on computer vision (ICCV)*.
- Yonemoto, S., Arita, D., & Taniguchi, R. (2000). Real-time human motion analysis and IK-based human figure control. In *Proceedings of the workshop on human motion (HUMO)*.