Real-Time 3D Articulated Pose Tracking using Particle Filtering and Belief Propagation on Factor Graphs

Olivier Bernier and Pascal Cheung-Mon-Chan
France Telecom R&D, 2 Av. Pierre Marzin 22307 Lannion Cedex France
olivier.bernier@francetelecom.com

Abstract

This article proposes a new statistical model for fast 3D articulated body tracking, similar to the loose-limbed model, but using the factor graph representation. Belief propagation is used to estimate the current marginal for each limb. All belief propagation messages are represented as sums of weighted samples. The resulting algorithm corresponds to a set of particle filters, one for each limb, where an extra step recomputes the weight of each sample by taking into account the interactions between limbs. To take into account fast moving limbs, proposal maps are used to steer samples to regions of high likelihood. Applied to upper-body tracking with disparity and colour images, the resulting algorithm estimates the body pose in quasi real-time (10Hz).

1 Introduction

Articulated body pose estimation and tracking, either for monocular, stereo or multiple camera sequences, is a challenging problem, especially if a real-time algorithm is needed. Methods can be classified between deterministic [1, 10, 11] and statistical ones [4, 5, 7, 9, 13, 15], the latter being generally more robust. [7] or [12] proposed a statistical graphical model to estimate the body pose from a few frames by using belief propagation. None of these techniques works in real-time, especially statistical ones. The main difficulty is the high dimension of the parameters space. Quasi real-time methods were proposed, using either a deterministic method [3] which can lose track for fast motions or occlusions (see [4]) or a-priori knowledge of the behaviour of the tracked person [2].

In this paper, we are interested in tracking the upper-body pose in real-time using a stereo camera. We propose a new statistical model for fast 3D articulated body tracking, similar to the loose-limbed model [12], but using belief propagation on factor graphs [8] and a specific message update order. All messages are represented as sums of weighted samples. A fully recursive estimation, equivalent to multiple particle filters interacting through belief propagation in the discrete state space of the samples, is obtained. The resulting method is quasi real-time (10Hz).

BMVC 2006 doi:10.5244/C.20.4

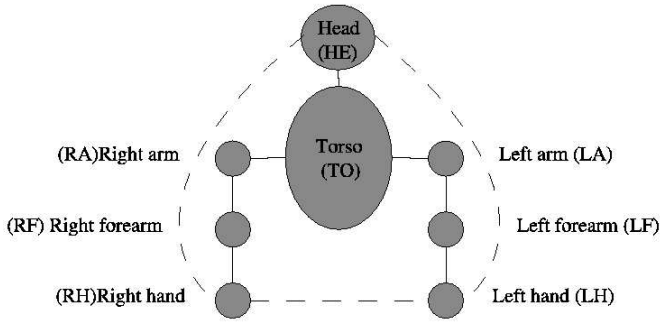

Figure 1: Upper body model: full lines represent joint links, dash lines represent non intersection constraints.

2 Recursive Bayesian Tracking for Articulated Body

The articulated structure of the upper body is represented by the graph shown in figure 1. The model consists of M limbs, each with its own state X^μ. Each limb μ generates an observation (image region) Y^μ. The model is further composed of links between limbs. These links represent joints, as well as other constraints such as non intersection constraints.

A Markov Network can be constructed from this structure [7, 12]. However, the resulting network presents a three nodes clique (corresponding to the head and hands). We simplify this model by using only pairwise interactions. The correct resulting structure can be expressed as a factor graph [8], which directly shows the decomposition of the joint probability as products of factors. To take into account the inter-frame coherence, this graph is extended to all frames by connecting the limbs in consecutive frames. The resulting factor graph is shown in figure 2. For clarity, only two consecutive frames are shown and the function nodes corresponding to the observations Y^μ are omitted. The model parameters are the observations conditional probabilities $P^\mu(Y^\mu|X^\mu)$, the time coherence functions $T^\mu(X_t^\mu, X_{t-1}^\mu)$ and the interaction potential for each link $\psi^{\mu\nu}(X^\mu, X^\nu)$.

Knowing an initial state of the body at time 0 and all observations, we can obtain the limbs state's marginal probabilities using the belief propagation algorithm on factor graphs [8]. As the graph includes cycles, the obtained marginal is an approximation of the true one. This approximation further depends on the messages update order. To simplify the algorithm, we choose to propagate the messages for all nodes in the same frame for a fixed number of iterations (10 in our case) and then to propagate only once from a frame to the following one. As a result, the estimation of a marginal at any time t does not depend on the observations after time t, and the estimation of the marginals can be computed recursively.

We represent all messages as sets of weighted samples. The messages sent from the previous frame to the current nodes are calculated using a particle filter scheme consisting in a re-sampling step followed by a prediction step based on the time coherence function. The loopy belief propagation algorithm is then reduced, for the current frame, to a loopy propagation algorithm for discrete state spaces, the space state for each limb being re-

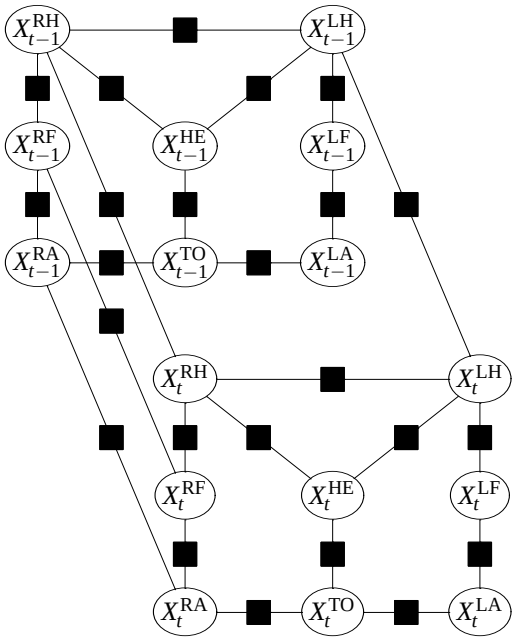

Figure 2: Factor graph at time t. Circles correspond to variable nodes (limb states), and black squares to function nodes (time coherence functions T^{μ} and interaction potentials ψ^{μ}). For clarity's sake, we have omitted the factor nodes between the limbs states at different frames for all limbs (except for the Left Hand, Right Hand, Right Forearm and Right Arm limbs).

stricted to its samples. Moreover the marginal probability is then simply represented as a weighted sum of the same samples. In this manner, a full recursive estimation is obtained. The algorithm is equivalent to a set of interacting particle filters, where the sample weights are re-evaluated at each frame through Belief Propagation to take into account the links between limbs. This algorithm is relatively fast because for each frame, as opposed to [12], the likelihood has to be evaluated only once for each sample, and the link interaction potential only once for each pair of samples for all connected limbs.

3 Application to upper body tracking using depth images

The model is applied to articulated upper-body tracking using depth images from a Bumblebee® stereo camera (*www.ptgrey.com*). The depth images are calculated using the corresponding commercial software. For tracking, both depth images and colour images are used. The colour image is needed to track the head and hands using the face colour (the hands are supposed to have a colour similar to the face).

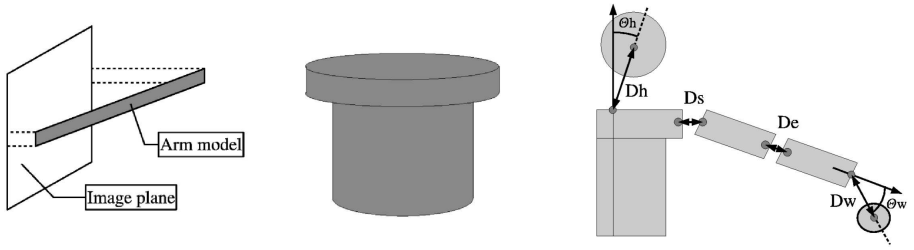

Figure 3: Left: arm and forearm model. Middle: torso model, the flat upper cylinder corresponds to the part between the shoulders. Right: links between limbs.

3.1 Initialization and preprocessing

An accurate face detector [6] is used to detect the face in the colour image. To speed-up face detection, only regions of valid disparity with moving skin coloured pixels are tested. Once the face is detected, the starting pose is supposed to be one with arms along the body, with the torso vertical and facing the camera. The tracker can easily recover the real pose as long as it is not too far from this hypothesis. The detected face is also used to initialize a face colour histogram (from the UV values in the YUV colour space).

For each pixel in the image, the 3D coordinates in the camera reference frame are calculated using the Bumblebee® software. Moreover, at each frame, the face colour histogram is used to compute a face/hand colour probability image. Both these information are sub-sampled by a factor of 16 (4x4 blocks) to reduce the computational complexity of the tracking (figure 5). Using the number of valid disparity pixels and the standard deviation of depth values, a confidence factor for depth information is also calculated for each 4x4 block.

3.2 Time coherence functions

The time coherence functions $T^{\mu}(X_t^{\mu}, X_{t-1}^{\mu})$ are simple Gaussians, independent for each parameter, centred on the value in the previous frame. For the forearms and hands, which can move fast and also rapidly change speed, the time coherence functions are a mixture of two similar Gaussians, one centred on the previous parameter and one centred on the prediction of the current parameter using the previous speed. The standard deviation is chosen to be 10 cm for hands positions, and 5 cm for other positions. For angles, the standard deviation is set to $\pi/4$ for forearms angles, $\pi/8$ otherwise.

3.3 Observation probabilities

The observations Y^{μ} are simply the estimated 3D points P (one for each pixel in the sub-sampled image), with associated confidence factors c and face colour probabilities h. The observations are supposed to be independent for each limb and each pixel. For each limb, the likelihood is calculated using a score $S^{\mu}(P,h)$:

$$P^{\mu}(Y^{\mu}|X^{\mu}) \propto \exp\{\sum_{i=1}^{N} S^{\mu}(P_i, h_i, X^{\mu})c_i\} \quad (1)$$

where N is the number of points in the sub-sampled images. As the depth data is very noisy, simple shape models were chosen. For the head, the score S^μ is a simple Gaussian distance to a sphere, multiplied by the head colour probability. For the torso, the score S^μ is the Gaussian distance to a shape composed of two flat cylinders (figure 3). We limit the orientations of the torso relative to the camera plane in the range $[-\pi/4, +\pi/4]$. Due to the depth estimation algorithm, hands appear flat. Consequently we chose for the hand a score S^μ which is the product of the distance to a flat 2D disk, parallel to the image plane, with the face colour probability. On a ring around this disk, the score is fixed to a small negative value multiplied by a Gaussian of the distance to the disk. This insures that the disk is isolated and not part of a larger surface. Similarly, for the arms and forearms, the score S^μ is the distance to a rectangular patch, parallel to the image plane in the direction of its smallest edge (figure 3), with a small negative factor for two strips on both sides of the patch. If the angle between the image plane and the rectangular patch is too important, the arm or forearm is not visible, and the probability of the limb is set to a fixed value. The size of each limb is fixed a priori, but the model is robust enough to accommodate various persons, as the articulations are not represented as hard constraints but as probabilities.

3.4 Links interaction potentials

For the links interaction potentials, a Gaussian of the distance between two link points is used (see figure 3, distances *Dh*, *Ds*, *De*, *Dw*). This Gaussian is zero centred for the shoulder-arm and arm-forearm joints, and on a reference distance for the head-torso and forearm-hand joints. Another constraint is added giving zero potential for angles θ_h, θ_w (see figure 3) above a fixed threshold, and for arms intersecting the torso (excluding the shoulder joint). Three additional links are defined, which simply give a zero probability to solutions where hands and head intersect.

4 Proposal distributions

The basic method to generate the samples for the message received by a variable node, from the previous frame, is to use the previous samples and the time coherence function. However these samples can be far from high likelihood region for the current image [4]. This is especially true for fast moving parts such as hands and forearms. Consequently we use a proposal distribution taking into account the current image. If we use a proposal distribution $Q^\mu(X_t^\mu | Y_t^\mu, X_{t-1}^\mu)$ to generate the samples $X_t^{\mu i}$ for a limb μ at time t, then the particle filter approximation of the message is:

$$m^\mu(X_t^\mu) = \sum_i w_t^{\mu i} \delta(X_t^\mu - X_t^{\mu i}) \tag{2}$$

$$w_t^{\mu i} \propto \frac{T^\mu(X_t^{\mu i}, X_{t-1}^{\mu i})}{Q^\mu(X_t^{\mu i} | X_{t-1}^{\mu i}, Y_t^\mu)} \tag{3}$$

The basic method consists in choosing for Q^μ the time coherence function T^μ, giving weights equal to 1. We can choose the proposal distribution Q^μ as the product of a distribution R^μ knowing the current image and the time coherence function T^μ, which gives the following weight:

Figure 4: Top left: original image. Bottom left: proposal map for hands. Top to bottom, and middle to right: proposal maps for arms and forearms for angles 0, $\pi/4$, $\pi/2$ and $3\pi/2$

$$w_t^{\mu i} \propto \frac{1}{R^{\mu}(X_t^{\mu i}|Y_t^{\mu})} \tag{4}$$

The trick is to use a very fast approximation R^{μ} of the true likelihood, a proposal map, i.e. a probability distribution on a discretized version of parameter space of each limb. This is similar to the proposal maps of [9] but for the parameters of the limbs instead of the position of the joints.

4.1 Proposal maps

For fast computation, only the parameters observable in the image (position and orientation in the image plane) are taken into account, and only for the arms, forearms and hands. For the other parameters, the samples are generated only from previous samples without considering the current image, using the basic method. The positions are discretized in the same way as the depth and head/hand colour information (4x4 pixels blocks), and the angles are discretized into 12 bins between 0 and π.

For both hands, a very simple probability, obtained from the product of the head/hand colour probability with the depth confidence, is used. For the arms and forearms a simple model is used, representing a small elongated rectangle of fixed size. The probability is obtained through a Hough transform, giving a score for each position and orientation of such a rectangle, and using the depth confidence of each 4x4 pixel block as data. As both hands use the same model, the same proposal map is also used. In the same way, all arms and forearms use the same proposal map. Figure 4 shows an example of the resulting proposal maps.

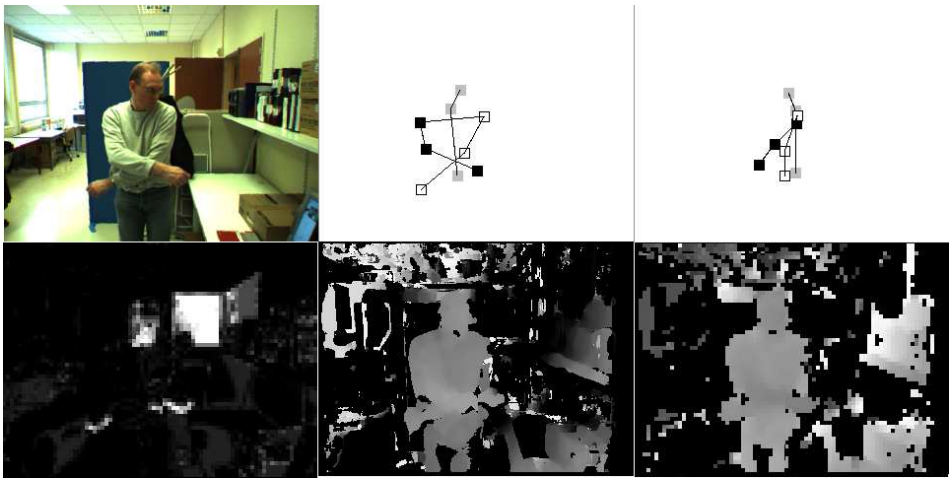

Figure 5: Tracking result. Top from left to right: current frame, frontal view of the estimated 3D model and side view from the right of the image. The black squares represent the right arm, the white ones the left arm, and the grey ones the centre of the head and the top and bottom of the torso. Each arm is represented by the position of its shoulder, elbow, and hand. Bottom from left to right: sub-sampled probability image using the face colour, inverted depth image (black correspond to invalid disparity, white to pixels closer to the camera), sub-sampled depth image (white shows pixels far from the camera).

4.2 Samples generation

From these proposal maps, we can draw samples of parameters using the conditional cumulative distribution functions of the corresponding probabilities. From samples drawn from R^{μ}, we can generate samples drawn from the product of R^{μ} and T^{μ} using rejection sampling. As the parameters of the obtained samples are discretized (see 4.1), a small-support uniformly distributed random variation is added to each parameter coming from a proposal map, to explore the continuous parameter space.

The rejection sampling scheme can be very slow as many samples must be drawn before one is accepted if the regions of high value for the proposal map R^{μ} are far from the regions of high value for the time coherence function T^{μ}. To avoid this problem the variance of each Gaussian in T^{μ} (see 3.2) is augmented after each rejection for the same sample. After the sample is accepted, the corresponding weight of the sample is divided by the ratio between the true time coherence function and the one obtained with the enlarged Gaussians. Moreover, a sample is automatically accepted after 100 rejections. This heuristic seems to give good results in practice.

5 Results

Figures 5 and 6 show tracking results for various positions, including a failure case due to an arm being totally occluded by a forearm. Usually, the system can recover the real position of the arm when the arm moves away from this position. This kind of occlusion

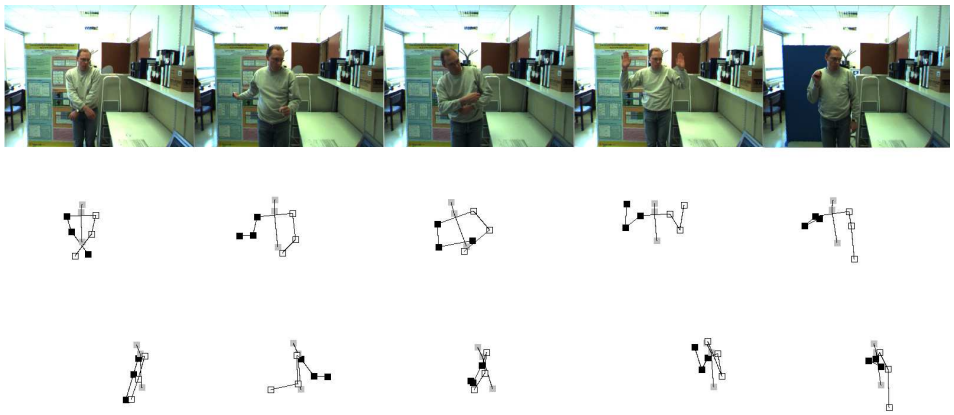

Figure 6: Tracking results. Middle line: front views, Bottom line: side view from the left of the image. Right: a failure case

should be taken into account directly in the model by adding a binary variable to the parameter state of arms and hands, indicating whether this part is occluded and should be matched to the image or not. Otherwise, good tracking results are achieved, even for fast movements and occlusions cases.

We also compare the results (front views only) using the proposal map R^{μ} for the arms and forearms and without using it (figure 7). The results are presented for each frame of a small sequence, showing a critical case with very fast movements. We do not provide results without the proposal map for the hands as the initialization phase may fail in this case even for a small deviation from the hypothesized initial position of the hands.

We use a 3.2 GHz bi-processor, with one thread devoted to depth estimation, and the other to tracking. The system runs approximately at 11-12Hz, using 300 samples for each limb, without the proposal map for arms and forearms, and approximately at 10Hz using the proposal map.

We evaluated the estimation noise on a small sequence (100 frames) without movements. In this case it can be approximated by the standard deviation of the estimated positions given by the tracker. The results are given in table 8. This table also shows that better results are obtained using true rejection sampling (see 4.2), or using 1000 samples for each limb. However, the true rejection sampling algorithm runs at 1 frame per second, with fast moving arms, and the algorithm with 1000 samples runs approximately at the same speed whatever the sequence content. Note that the estimation noise can be reduced by using an additional Kalman filter stage, with the drawback of introducing a small lag in the resulting estimation.

6 Conclusion

We have presented a fast method for articulated body-tracking, and applied it to upper-body tracking using depth images. The system can track the upper body in quasi real-time even with self-occlusions and fast movements, without a-priori knowledge of the

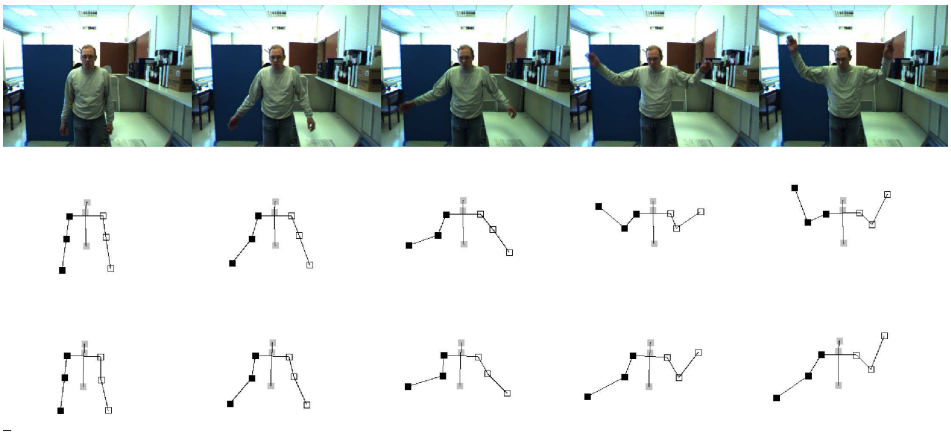

Figure 7: Tracking result for 5 consecutive frames. Top: original image. Middle: using the proposal map for the arms and forearms. Bottom: without the proposal map.

Limb	Proposed algorithm with 300 samples per limb	True rejection sampling with 300 samples per limb	Proposed algorithm with 1000 samples per limb
Head	10.6	10.1	8.8
Shoulders	16.5	14.5	11.3
Elbows	30.6	26.9	24.2
Hands	12.1	18.2	8.3
All limbs	18.4	18.5	13.8

Figure 8: Standard deviation in millimetres of the estimation noise for three algorithms (see text) for the head, shoulders, elbows, hands, and all combined

background or of the behaviour and appearance of the tracked person. This method can be generalized to other articulated body tracking problems, such as monocular tracking, or whole body tracking.

The next step is to take into account directly in the model the most recurrent occlusions (hand/hand, hand/arm and forearm, arm/forearm). Another extension would be to use learned time coherence functions, constraints and priors [14] to further restrict the sample space. And finally, for a robust real time system, a method for automatically detecting failures and reinitializing the tracking is needed.

References

[1] Christoph Bregler and Jitendra Malik. Tracking people with twists and exponential maps. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 8–15, June 1998.

[2] Fabrice Caillette, Aphrodite Galata, and Toby Howard. Real-time 3-d human body tracking using variable length markov models. In *British Machine Vision Conf.*,

September 2005.

[3] David Demirdjian, T. Ko, and Trevor Darrell. Constraining human body tracking. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1071–1078, October 2003.

[4] David Demirdjian, Leonid Taycher, Gregory Shakkhnarovich, Kristen Grauman, and Trevor Darrell. Avoiding the streetlight effect: Tracking by exploring likelihood modes. In *Proc. Int. Conf. on Computer Vision*, volume 1, pages 357–364, October 2005.

[5] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *Int. Journal of Comp. Vision*, 61(2):185–205, February 2005.

[6] Raphael Féraud, Olivier Bernier, Jean Emmanuel Viallet, and Michel Collobert. A fast and accurate face detector based on neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(1):42–53, January 2001.

[7] Jiang Gao and Jianbo Shi. Multiple frame motion inference using belief propagation. In *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 875–880, May 2004.

[8] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. on Information Theory*, 47(9):498–519, February 2001.

[9] Mun Wai Lee and Isaac Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 334–341, July 2004.

[10] Vladimir Pavlović, James M. Rehg, Tat-Jen Cham, and Kevin P. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Proc. Int. Conf. on Computer Vision*, volume 1, pages 94–101, September 1999.

[11] Ralf Plankers and Pascal Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9):1182–1187, 2003.

[12] Leonid Sigal, Sidharth Bhatia, Stefan Roth, Michael J. Black, and Michael Isard. Tracking loose-limbed people. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 421–428, July 2004.

[13] Cristian Sminchisescu and Bill Triggs. Covariance scaled sampling for monocular 3d body tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 447–454, December 2001.

[14] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. In *Proc. Int. Conf. on Computer Vision*, volume 1, pages 403–410, October 2005.

[15] Ying Wu, Gang Hua, and Ting Yu. Tracking articulated body by dnamic markov network. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1094–1101, October 2003.