MULTICUES 2D ARTICULATED POSE TRACKING USING PARTICLE FILTERING AND BELIEF PROPAGATION ON FACTOR GRAPHS

Philippe Noriega and Olivier Bernier

France Telecom R&D
2 Av. Pierre Marzin 22307 Lannion Cedex France

ABSTRACT

This paper describes a method for articulated upper body tracking in monocular scenes. The compatibility between model and the image is estimated using one particle filter for each limb and the compatibility between limbs is represented by interaction potentials. The joint probability is obtained by belief propagation on a factor graph. The body model is a loose limbed model including attraction potentials between adjacent limbs and constraints to reject poses resulting in collisions. Robust compatibility functions based on face color, edges and motion energy are used to evaluate the likelihood of the generated hypotheses. Experimental results show the upper body tracking efficiency of the proposed algorithm.

Index Terms— Belief propagation, factor graphs, particle filter, tracking

1. INTRODUCTION

Algorithms for body tracking must cope with non linear and high dimensional space in which the joint probability function is highly multimodal and sharp. In this context, deterministic methods can track in real time with stereo cameras [1], but they may fail for monocular view because of many local optimums due to ambiguities in monocular scenes [2]. In this context, learning based methods imply huge data bases even if robust locally-weighted regression between candidate poses is used [3]. To speed up the selection of a subset of learned nearest neighbors, the comparison process uses locally sensitive hashing and Hamming distance. Another approach consists in associating deterministic optimization and a learned base of poses [4]. The comparison between a test image and the learned base aims to initialize the optimization process near the modes of the likelihood. Learning based methods may fail owing to the wide pose space and to external parameters (clothing, hairstyle...). Stochastic algorithms are useful in monocular vision to resolve ambiguities resulting from 2D to 3D pose inference. In this case, a multi-hypothesis algorithm, such as particle filtering [5], is attractive but the high dimension of the pose state complexifies the solution. A key to this problem is to use a loose-limbed body model [6] where the likelihood of each limb can be evaluated independently. In this manner, a particle filter can be associated with each limb reducing the search space dimension to the number of *dof* of a limb [7]. Influence between limbs are taken into account by propagating limbs' likelihoods through a factor graph using belief propagation. A similar technique is used in monocular scenes [8] with only motion energy as cues. Another approach uses edges and grey level with Mean Field Monte Carlo [9]. In this paper, the number of cues is increased to enhance the robustness of the tracking. Particle filters and belief propagation are used [7] to simplify the problem by computing the estimation in a discrete space instead of using, for example, Gibbs sampler in a continuous one [6]. This paper presents a monocular multicues tracking algorithm using a loose-limbed body model and particle filters interacting through belief propagation on a factor graph.

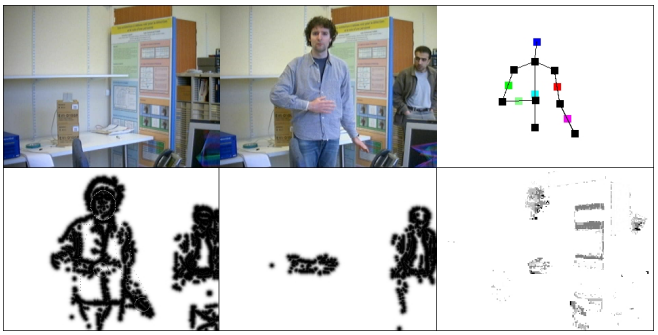

Fig. 1. Multicues tracking using belief propagation and factor graphs. Left to right and top to bottom: initialization image for background subtraction, current frame, estimated pose, fusion of contour distance map and robust background subtraction, motion energy distance map and face color map.

2. RECURSIVE BAYESIAN TRACKING FOR ARTICULATED BODY

The upper body is modeled by a graph including M limbs represented by nodes and links corresponding to articulations or non collision constraints between limbs (see figure 2). Basically, a Markov network can be used to represent this structure but the non-collision constraints between the head and

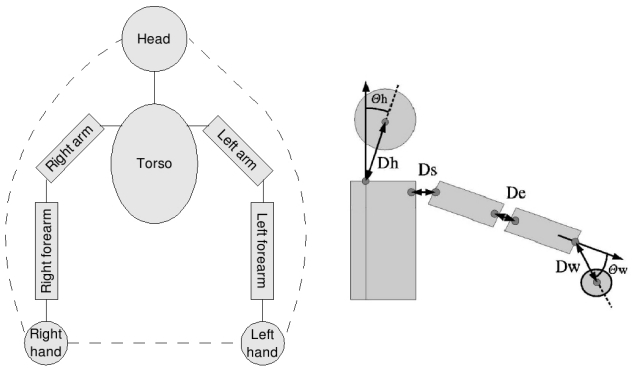

Fig. 2. Upper body model. Left: nodes correspond to limbs, articulation constraints are represented by solid lines and dashed lines are additional non-collision constraints between head and hands. Right: cardboard 2D body model.

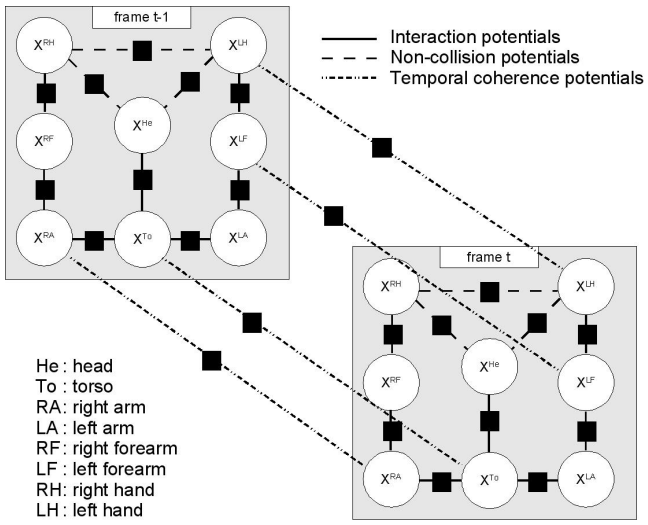

Fig. 3. Factor graph at time t. Circles corresponds to variable nodes (limb states) and black squares to function nodes (temporal coherence T^μ and interaction or non-collision potentials $\psi^{\mu\nu}$). For clarity, only two consecutive frames are shown and the function nodes corresponding to the observations Y^μ are omitted.

the hands generate a three nodes clique. A factor graph is constructed to simplify the model by using only pairwise dependences represented by factors [7]. The joint probability can be decomposed as products of these factors. The complete graph includes the previous states to take into account the temporal coherence (see figure 3). Given a limb μ, its state X_t^μ at time t and the image observations Y_t^μ, the model parameters are the observations compatibility potentials $\phi_\mu(X^\mu, Y^\mu)$, the time interaction potentials $\tau^\mu(X_t^\mu, X_{t-1}^\mu)$, and the interaction potential for the link between limbs μ and ν: $\psi_{\mu\nu}(X^\mu, X^\nu)$. Adopting these notations, the joint probability knowing all the observations from time 0 to K is:

$$P(X_{0:K}|Y_{0:K}) = \prod_{t=0}^{K} \Phi(X_t, Y_t)\Psi(X_t) \prod_{t=1}^{K} T(X_t, X_{t-1}) \,, \tag{1}$$

with:

- $\Phi(X_t, Y_t) = \prod_{\mu=1}^{M} \phi^\mu(X_t^\mu, Y_t^\mu)$,
- $\Psi(X_t) = \prod_{(\mu,\nu)\in\Gamma} \psi^{\mu\nu}(X_t^\mu, X_t^\nu)$, where Γ is the set of links,
- $T(X_t, X_{t-1}) = \prod_{\mu=1}^{M} \tau^\mu(X_t^\mu, X_{t-1}^\mu)$.

The marginal probabilities of the limbs state are obtained using the belief propagation algorithm on a factor graph [7]. 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, the messages are propagated to all nodes in the same frame for a fixed number of iterations (10 in our case) and then propagated only once from a frame to the following one. Therefore, 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.

The messages are represented by sets of weighted samples. From one frame to the next, they are calculated using a particle filter scheme consisting in a re-sampling step followed by a prediction step based on the time coherence functions. 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 restricted 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 contrary to [9], samples are drawn only once for each frame t. Moreover, unlike [6], the image based compatibility functions $\phi^\mu(X_t^\mu, Y_t^\mu)$ have to be evaluated only once for each sample and the link interaction potentials only once for each pair of samples for all connected limbs.

3. APPLICATION TO UPPER BODY TRACKING IN MONOCULAR SCENES

The model is applied to articulated upper-body tracking using monocular color images from a webcam. Color is used to track the head and the hands and the grey level image is used to detect motion and compute contour based cues.

3.1. Initialization

An accurate face detector [10] is used to detect the face in the color image. Once detected, the starting pose supposes that arms are 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 color histogram.

3.2. 2D body model and Link interaction potentials

The 2D body model is shown in figure 2 right. Head and hands are represented by circles and rectangular patches are used for torso, arms and forearms. Limbs and limbs' edges are discretized using respectively a grid of points inside them and regularly distributed points around them. A Gaussian of the distance between two link points is used to compute the link interaction potentials (see figure 2 for 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 θh , θw (see figure 3) above a fixed threshold. Three additional links are defined, which simply give a zero probability to solutions where hands and head intersect.

3.3. Time coherence function

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 forearms and hands, which can move fast and rapidly change speed, the time coherence functions are a mixture of two similar Gaussians, one centred on the previous parameter and the other centred on the prediction of the current parameter using previous limb 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 $\pi/8$.

4. IMAGE FEATURES

The image compatibility functions $\phi^{\mu}(X_t^{\mu}, Y_t^{\mu})$ are computed from scores S_f^{μ} representing the compatibility between a limb μ and cues f extracted from an image. Contrary to stereo [7], monocular images needs more cues to reach a sufficient level of robustness. Thus, multicues contour and color based terms are fused to provide a score: $S^{\mu} = \prod_f S_f^{\mu}$. Considering the highest score for all samples of the limb $\widehat{S^{\mu}}$, the image compatibility function is: $\phi^{\mu}(X_t^{\mu}, Y_t^{\mu}) = exp[-\lambda(\widehat{S^{\mu}} - S^{\mu})]$, with λ a parameter that depends on the range of the values returned by each cue.

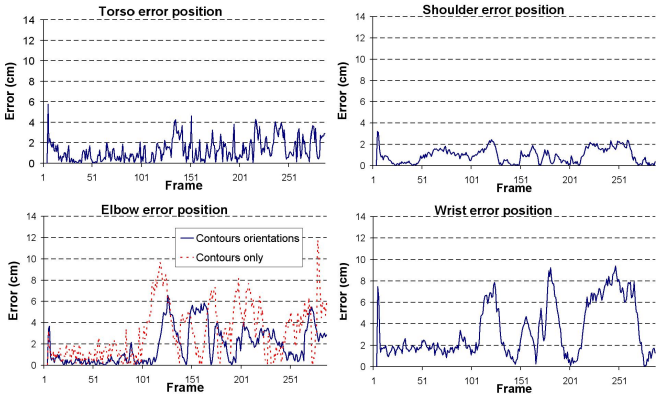

Fig. 4. Left to right and up to bottom, estimated error for the positions of torso, right shoulder, right elbow and right wrist. Elbow position computed using plain contour cue (dot line, $mean = 3.0cm$) is less accurate than the one computed using the orientation contours §4.2 (solid line $mean = 1.9\ cm$).

4.1. Face and hands detection

Considering the head position detected during initialization step (§ 3.1), a color model is provided computing the UV histogram of the head color from the YUV color space. During tracking, the points belonging to the head and the hands are compared with this model to evaluate their color compatibility. This score is completed with the Chamfer distance from contours provided by a Shen-Castan detector: the pixels corresponding to the projection of the points belonging to the hands model and head edges are scored computing the Gaussian of this distance. To avoid taking into account background contours, a robust background subtraction [11] is applied before contour detection.

4.2. Torso, arms and forearms detection

A more accurate contour based score can be estimated if the orientation of the contours are taken into account. A local squared area is defined on the center of a projected edge limb point. A pixel p belonging to an area Z_e is weighted by the contour magnitude $\|\overrightarrow{p}\|$ and a Gaussian spatial kernel G_e centered on the limb edge point p_e. For each pixel, the Gaussian difference between the limb and the contour orientation G_{θ} is computed:

$$S_{or} = \sum_{p_e} \sum_{p \in Z_e} M(\|\overrightarrow{p}\|) G_c[d(p, p_e)] G_{\theta}[\theta_{limb} - \theta_p] \ , \quad (2)$$

where $d()$ refers to the Euclidian distance and $M()$ is a function that penalize low and high magnitude contours.

To avoid limbs getting stuck in a possible wrong static configuration, a motion attraction term is added. A motion energy distance map is computed with the Chamfer distance from an adjacent frame difference. A motion energy score

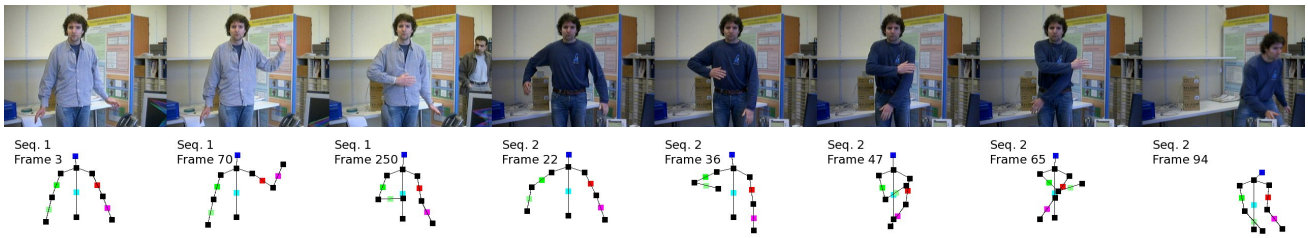

Fig. 5. Tracking results. Upper line: original frame, bottom computed front pose. In sequence 1, frame 3 is an initialization image. Frame 65 in sequence 2 shows a wrong right elbow position due to the position of the right arm in front of the torso of the same color. The last image shows a correct result even when the body is out of image plan rotation.

S_{mo} is computed summing the energy distance ε_{mo} of each pixel p_i corresponding to the projection of the points inside a limb. If $\Delta_{p_i} = d(p_i^{t-1}, p_i^t)$:

$$S_{mo} = \sum_{p_i} [1 - G_{mo}(\Delta_{p_i})]\varepsilon_{mo}(p_i) + \frac{G_{mo}(\Delta_{p_i})}{\varepsilon_{mo}(p_i) + 0.1} \quad (3)$$

The Gaussian kernel G_{mo} models the link between the inter-frame energy motion and the limb displacement.

5. EXPERIMENTAL RESULTS AND CONCLUSION

The system was tested on sequences grabbed with a web-cam. Quantitative results (see figure 4) were obtained for a sequence comparing the estimated pose with a ground truth provided by a magnetic motion captor. This sequence shows a person executing gestures in front of the camera (see figure 5, frames 3, 70 and 250). For the torso and the shoulder position, the error stays below 5 cm and 10 cm for the elbow. Results are less accurate for the wrist because the hand model is too coarse. In some cases, for example when the wrist goes out of the sleeve shirt, the hand model, a rigid circle (see § 3.2), does not correspond to the hand shape and the estimated hand position moves along the wrist. The pinhole camera model used to convert 3D points to pixels coordinates is very coarse and can explain part of the obtained errors. In frame 250, another person enters the scene. The system is not distracted in spite of numerous false positives on image cues (figure 1). The edge orientation cue (§4.2) provides a mean error for the test scene of 1.9 cm instead of 3.0 cm, the mean error resulting from plain edges cue.

The second tested sequence (see figure 5, frames 22, 36, 47, 65 and 94) shows challenging poses with rapid motion. In frame 65, the right arm is positioned in front of the torso of the same color and it does not produce enough cues to find the correct position.

The algorithm speed is around 3 fps on a bi-processor 3.4 GHz. Real-time can be reasonably reached using threads in the source code. Projected future works will include a more accurate 3D model to track the body in 3D including a learning base for occluded poses.

6. REFERENCES

[1] D. Demirdjian, T. Ko, and T. Darrell, "Constraining human body tracking," in *ICCV*. 2003, pp. 1071–1079, IEEE Computer Society.

[2] C. Sminchisescu and A. Telea, "Human pose estimation from silhouettes - a consistent approach using distance level sets," in *WSCG*, 2002, pp. 413–420.

[3] G Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *ICCV*. 2003, pp. 750–757, IEEE Computer Society.

[4] D. Demirdjian, L. Taycher, G. Shakhnarovich, K. Grauman, and T. Darrell, "Avoiding the "streetlight effect": Tracking by exploring likelihood modes," in *ICCV*, 2005, pp. 357–364.

[5] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *IJCV*, vol. 29, pp. 5–28, 1998.

[6] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *CVPR*, 2004, vol. 1.

[7] O. Bernier and P. Cheung-Mon-Chang, "Real-time 3d articulated pose tracking using particle filtering and belief propagation on factor graphs," in *BMVC*, 2006, vol. 01, pp. 27–36.

[8] J. Gao and J. Shi, "Multiple frame motion inference using belief propagation," in *FGR*, 2004, pp. 875–882.

[9] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic markov network," in *ICCV*, 2003, pp. 1094–1101.

[10] R. Féraud, O. Bernier, J. E. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," *PAMI*, vol. 23(1), pp. 42–53, 2001.

[11] P. Noriega and O. Bernier, "Real time illumination invariant background subtraction using local kernel histograms," in *BMVC*, 2006, vol. 3, pp. 979–988.