Human Pose Tracking Using Multi-level Structured Models

Mun Wai Lee and Ram Nevatia

Institute for Robotics and Intelligent System,
University of Southern California,
Los Angeles, CA 90089, USA
{munlee, nevatia}@usc.edu

Abstract. Tracking body poses of multiple persons in monocular video is a challenging problem due to the high dimensionality of the state space and issues such as inter-occlusion of the persons' bodies. We proposed a three-stage approach with a multi-level state representation that enables a hierarchical estimation of 3D body poses. At the first stage, humans are tracked as blobs. In the second stage, parts such as face, shoulders and limbs are estimated and estimates are combined by grid-based belief propagation to infer 2D joint positions. The derived belief maps are used as proposal functions in the third stage to infer the 3D pose using data-driven Markov chain Monte Carlo. Experimental results on realistic indoor video sequences show that the method is able to track multiple persons during complex movement such as turning movement with inter-occlusion.

1 Introduction

Human body pose tracking is important for many applications, including understanding human activity and other applications in video analysis. For example in surveillance applications, people are often the main object of interest in the monitored scenes. Analyzing the body poses of the people allows for inference of the people's activities and their interactions. In the general case, analyzing body poses involves estimating the positions of the main body components such as the head, torso, and limbs, and the angles of joints such as shoulders and elbows.

Existing research work on human pose estimation is motivated by different applications. In human motion capture and human computer interaction, one can simplify the problem by using multiple cameras and controlling the environment and the subject's appearance. In our work however, we focus on applications for video understanding and surveillance that deal with uncontrolled scenes with only a single camera; multiple persons may also be present. This is a difficult problem for many reasons including variations in individual body shapes and choice of clothing. Furthermore, the humans need to be segmented from the background and self-occlusions need to be considered. The presence of multiple persons in the scene makes the problem more complex as people may occlude each other.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part III, LNCS 3953, pp. 368–381, 2006.

Our aim is to recover body and limb positions and orientations, i.e. their poses, in 3D from monocular video sequences without any special markers on the body. We use a model-based approach to overcome the above difficulties. By modeling separately the human body and other aspects of the image generation process (including dynamics, image projection, and observation process model), fewer training images can be used to handle more varying environments. An analysis-by-synthesis approach is often used to evaluate hypotheses of the state that represents the pose parameters but efficient search for the state solution in a high dimension space is a key issue. For a sequence, we can use the dynamic model to reduce the search space; nonetheless, we must still estimate the initial state and re-initialize when tracking becomes unreliable. In this paper, we present a novel three-stage approach for 3D pose estimation and tracking of multiple people from a monocular sequence (See Fig. 1). To improve the search efficiency, a *hierarchical* approach is used since some parameters are easier to estimate than others. In addition, bottom-up detection of body components is used to reduce the search space. We focus on sequences on a meeting room environment. The camera is stationary and the resolution is such that a persons height is about 200 to 250 pixels.

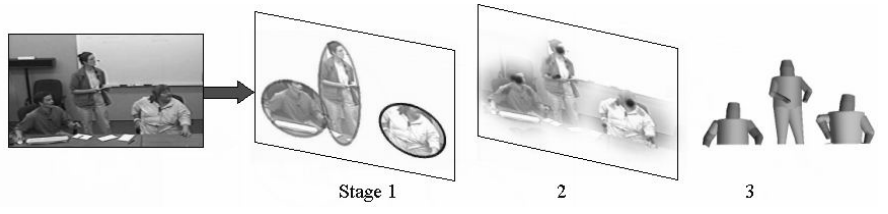

Fig. 1. Three stages approach. From left: (*i*) input, (*ii*) Stage 1: blobs tracking, (*iii*) Stage 2: 2D inference, (*iv*) Stage 3: 3D inference. The scene consists of three persons in a meeting room environment. The poses are estimated in a hierarchical coarse-to-fine manner.

In the first stage, moving people are detected and tracked as *elliptical blobs*. A coarse *histogram-based appearance model* for each person is learned during tracking so that when one person occludes another, we can determine the depth ordering from the appearance. This stage provides a coarse estimation of the persons' positions, sizes and the occluding layers.

In the second stage, part detection modules are used to locate the faces, shoulders and limbs. Inferences from these local component detections are integrated in a *belief network*; a *belief propagation* technique is used to estimate the marginalized belief of each state. Restricting the second stage to 2D inference enables us to use *grid-based representations* for the belief functions that can handle complex distributions efficiently. This approach is different from the *nonparametric belief propagation* method (NBP) [6] [10] [14] [17] , as we do not use Monte Carlo sampling or a mixture-of-Gaussians approximation.

In the third stage, a method based on *data-driven Markov chain Monte Carlo* (DD-MCMC) [21] is used to estimate the full 3D body poses in each frame.

A state candidate is evaluated by generating synthesized humans (we assume an orthographic projection and known camera orientation) and comparing it to the input image. With this generative approach, we can consider nonlinear constraints such as inter-occlusion and non-self-penetration. To search the state space efficiently, the Markov chain transition uses proposal functions generated from the previous stage (estimates of 2D joint positions).

With these three stages, the body poses of each frame are estimated with multiple hypotheses. The body trajectories of the people can be estimated by combining results of multiple frames. A human dynamic model is used to apply temporal constraints of body kinematics. A *dynamic programming* technique is used to compute the optimal trajectories of the persons motion.

Part of the implementation of Stage 3 is based on our earlier work on pose estimation for a single person in mainly upright and frontal poses [8]. This work extends the method to dynamic pose estimation of multiple persons in sequences and includes more difficult scenarios such as turning movements as well as occlusions among people.

1.1 Related Work

There has been substantial work on estimating 2D human pose [11] [12] [22]. Estimating 3D pose is more challenging as some degrees of motion freedom are not observed and it is difficult to find a mapping from observations to state parameters directly. Several learning based techniques have been proposed [1] [13], but these rely on accurate body silhouette extraction and having relatively large number of training images. Model-based approaches are popular because it is easy to evaluate a state candidate by synthesizing the human appearance. In [3], *particle filtering* is used for 3D pose tracking with multiple cameras by approximating the state posterior distribution with a set of samples. It is however difficult to extend this to tracking monocular view because of significant ambiguities in depth. In [15], a *mixture density propagation* approach is used to overcome the depth ambiguities of articulated joints seen in monocular view. A *hybrid Monte Carlo* technique is used in [2] for tracking walking people. Nonetheless, the issue of pose initialization is not addressed in these techniques.

The *non-parametric belief propagation* (NBP) method [10] [17] has been used for pose estimation [6] [14] and hand tracking [18]. A *mean field Monte Carlo* algorithm is also proposed in [20] for tracking articulated body. These techniques use a graphical model, with each node representing a body joint. Inference is made by propagating beliefs along the network. Our method uses belief propagation only at the second stage to bootstrap the 3D inference at the third stage where a complete analysis, including self-occlusion, is performed. Recently, bottom-up, local parts detection has been used as a data-driven mechanism for the pose estimation [6] [9] [12] and this has now been recognized as an important component in a body pose estimation solution and is the main motivation for this work.

2 Human Ellipse Tracking

We describe in this section the first stage of our approach which involves the tracking of humans whose shape is approximated as ellipses in the video. The objective here is to determine the number of people in the scene, estimate coarsely their positions and sizes and infer the depth ordering when they overlap with each other in the image.

Given a sequence, the static background is learned using an *adaptive mixture model* approach [16] and the foreground moving blobs are extracted by background subtraction. Human ellipses are detected and tracked by matching them with the foreground. A human blob is represented by a simple ellipse that has five parameters: positions, width, height and rotation. The matching between the ellipses and foreground is described by a *cost function* based on region matching of the estimated ellipses with the foreground (see Fig. 2). A track is initiated automatically by the presence of an unmatched foreground blob of sufficiently large size. At each time frame, the states are updated by performing a block search to minimize the cost function. We assume that the ellipse size changes slowly and that the ellipses are allowed to overlap each other. A color histogram is used to represent the appearance of a human blob and is learned by adaptive updating. When the ellipses overlap, we determine the depth order by comparing the overlapped region with the learned color histograms.

This is a simple method to track the human blobs as the first coarse stage to estimate human pose and is adequate for uncrowded scenes.

Fig. 2. Human blobs tracking. *Left*: extracted foreground. *Right*: estimated ellipses representing human blobs with inference of depth order.

3 Inference of 2D Joint Positions

The second stage aims to make efficient inference of 2D image position of body joints, using results from various body component detections, as well as the dependency between the various joints. We use a graphical model to represent the human body. For a single frame, this graphical model is a tree where each node corresponds to the image position of a body joint and each edge represents the pair-wise dependency between the adjacent joints, as shown in Fig. 3. We let the state of the i^{th} body joint be denoted by $r_i = (u_i, v_i)$. These states are approximates of the 3D pose.

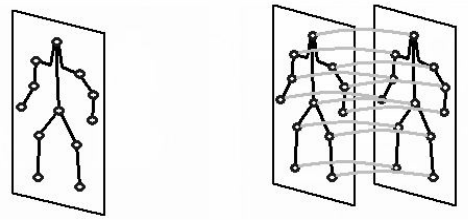

Fig. 3. *Left*: Graphical model in 2nd stage; *right*: extension to two frames showing temporal constraints. Note: observation nodes are not shown here.

3.1 Observation Function

For each node, there is a corresponding observation function, denoted by $\phi_i(r_i) = \phi_i(u_j, v_j)$. These functions are generated from various detection modules applied to the current frame. The observations include the outputs of human blob ellipse detector, face, torso, head-shoulder contour matching and skin blob extraction (Fig. 4); part detectors are described in more detail in [7][8]. Our proposed framework can be used with other detection modules proposed in the literature, for example in [11][12][14][16][22]. In general, these observations may contain localization noise, outliers, missed detections, and data association ambiguities in the presence of multiple persons. In [14], a mixture-of-Gaussians was used to approximate the observation function, but observations from multiple views were used to provide greater accuracy. For a monocular view, the observation function can be quite complex, and such an approximation scheme is inadequate. We therefore use a grid-based method to represent the observation function.

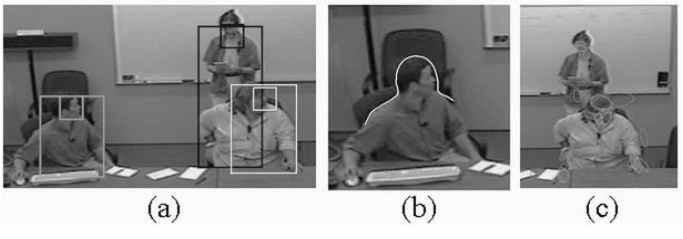

Fig. 4. Parts detection. (a) face-body tracker, (b) head-shoulder contour, (c) skin color blobs (for face and hand).

3.2 Potential Function

For each pair of adjacent nodes connected by an edge (i, j), there is a potential function denoted by $\psi_{i,j}(r_i, r_j)$ that encodes the joint distribution of neighbouring states. This potential function is shift invariant and can be simplified as a 2D function: $\psi_{i,j}(r_i, r_j) = \psi_{i,j}(u_i - u_j, v_i - v_j)$. We use an approximate grid representation for each potential function. In our current implementation, the grid is of the same size as the image.

The potential functions are however dependent on the image scale. We denote by s the scale representing the image height of a person when in a standing pose. The scale is estimated by the width of the detected human ellipse, denoted by $w_{ellipse}$, which is insensitive to occlusion of the lower body. Given the scale, the conditional potential function can be expressed as

$$\psi_{i,j}(r_i, r_j|s) = \hat{\psi}_{i,j}\left(\frac{u_i - u_j}{s}, \frac{v_i - v_j}{s}\right),$$

where $\hat{\psi}_{i,j}(\cdot)$ is a scale invariant function. Therefore, when learning the potential functions, the training data is normalized by scale.

The potential function, marginalized by scale, can be expressed as:

$$\psi_{i,j}(r_i, r_j) = \int \psi_{i,j}(r_i, r_j|s)p(s|w_{ellipse})ds,$$

where $w_{ellipse}$ is the width of the ellipse detected in the first stage and $p(s|w_{ellipse})$ is the posterior probability of scale given the width. In practice, the observed ellipse provides a fairly reliable estimate of the scale, so that the observation function can be approximated by $p(s|w_{ellipse}) = \delta(s - s')$, where $\delta(\cdot)$ is the delta function, $s' = \lambda w_{ellipse}$ is the estimated scale, and λ is a constant estimated from training data. The potential function can now be simplified as:

$$\psi_{i,j}(r_i, r_j) = \hat{\psi}_{i,j}\left(\frac{u_i - u_j}{s'}, \frac{v_i - v_j}{s'}\right).$$

3.3 Grid-Based Belief Propagation

Belief propagation is a statistical inference technique used to estimate the state belief in the graphical model as in [10][14][17][18]. At each iteration, each node passes messages to its neighbors. A message from ith node to the jth node is denoted by $m_{i,j}(r_j)$ and is expressed as:

$$m_{i,j}(r_j) = \int \psi_{i,j}(r_i, r_j)\phi_i(r_i) \prod_{k \in \Gamma_i \setminus j} m_{ki}(r_i)dr_i,$$

where r_i is an image position of the ith node and Γ_i is the set of neighbors of ith node. The belief of the ith node is given as:

$$b_i(r_i) \propto \phi_i(r_i) \prod_{k \in \Gamma_i} m_{ki}(r_i).$$

By using 2D grid representations for the observation functions, potential functions, messages and beliefs, the belief propagation computation is simplified. The message is expressed as:

$$m_{i,j}(u_j, v_j) = \sum_{u_i} \sum_{v_i} \psi_{i,j}(u_i - u_j, v_i - v_j)\phi_i(u_i, v_i) \prod_{k \in \Gamma_i \setminus j} m_{ki}(u_i, v_i).$$

This is a discrete convolution and can be rewritten as:

$$m_{i,j}(u_j, v_j) = \psi_{i,j}(u, v) \otimes \left[\phi_i(u, v) \prod_{k \in \Gamma_i \setminus j} m_{ki}(u, v) \right],$$

where the symbol $\otimes$ represents the convolution operation. The belief is now written as:

$$b_i(u_j, v_j) \propto \phi_i(u_j, v_j) \prod_{k \in \Gamma_i} m_{ki}(u_j, v_j).$$

We call these 2D belief functions as *belief maps* and they can be computed efficiently by using the *fast Fourier transform* for the discrete convolution. The maps are used as proposal functions in Stage 3 described later.

For a single frame, the graphical model is a tree. Each iteration involves a parallel updating of all the nodes. The number of iterations required for belief propagation is equal to the longest path between nodes, or the diameter of the graph. In our case, six iterations are sufficient.

The graphical model is extended to multiple frames, (see Fig. 3). Let r_i^t denotes the state of ith node at time t. The temporal potential function of this node between consecutive frames is denoted by $\psi_{T,i}(r_i^t, r_i^{t-1})$. This function is time invariant and can also be expressed as a grid representation:

$$\psi_{T,i}(r_i^t, r_i^{t-1}) = \psi_{T,i}(u_i^t - u_i^{t-1}, v_i^t - v_i^{t-1}).$$

The resulting graphical model now contains loops and the belief updating process becomes a *loopy belief propagation* which in general does not guarantee convergence. However, in practice, our network always converges during experiment. This is because the temporal potential function serves as a temporal smoother and this prevents oscillations. In our experiment, we observed

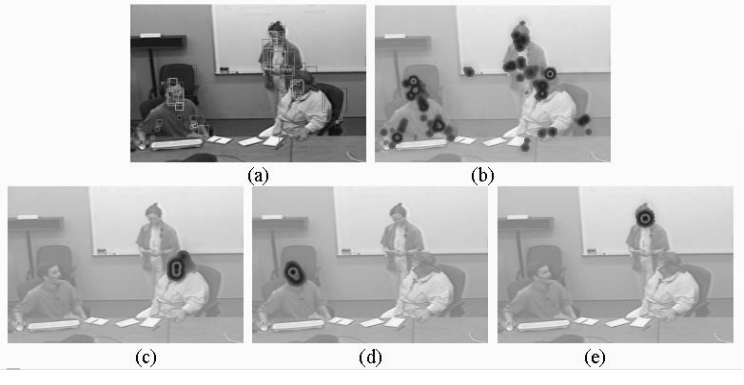

Fig. 5. Inference of 2D face positions. (a) Face detection from different cues with many false alarms, (b) initial face belief map before belief propagation, (c)-(e) face belief maps for each person after belief propagation. Dark regions indicate higher probability values.

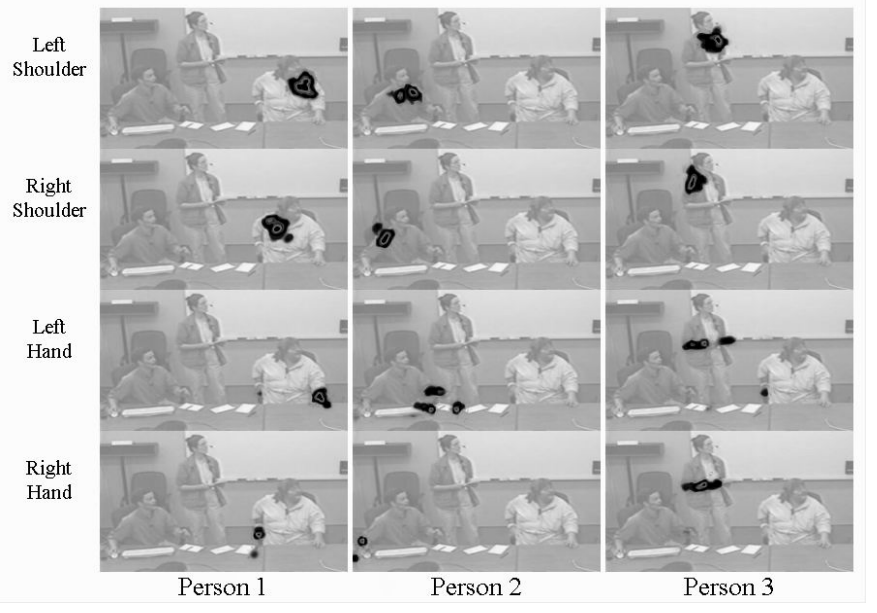

Fig. 6. Further examples of belief maps, each row for each body joint and each column for each person

that ten iterations are sufficient for convergence. Fig. 5 and Fig. 6 shows examples of belief maps for each person in the scene. In these maps, some ambiguities still exist; but the beliefs are much better compared to initial observation function.

The computations of messages and beliefs are deterministic. In comparison, nonparametric belief propagation uses Monte Carlo sampling which is less suitable in a multiple persons scene where the observations are more ambigious and distributions are complex.

4 3D pose inference

3D pose estimation is performed at the third stage. Estimating these object-centered parameters is important for providing view-invariant pose recognition and infering spatial relations between objects in the scene, for example during a pointing gesture. We use a model-based approach and MCMC inference technique. The belief maps generated in Stage 2 are used to generate data-driven proposal in this third stage.

In this section, we describe the key components including the human model, the observation, and the formulation of the prior distribution and likelihood function. We have extended a previous work [8] by formulating a joint prior distribution and a joint likelihood function for all persons in the scene.

4.1 Body Model and Likelihood Function

We use an articulated limb model of human body that defines the pose parameters as consisting of the torso position and orientation, and various limb joint angles. Additional latent parameters that describe the shape of the torso and limbs and the clothing type are also included to synthesize the human appearance more accurately for pose evaluation, as described in [7].

Pose estimation is formulated as the problem of estimating the state of a system. State for a sequence with T frames is represented by $\{\theta_1, \theta_2, ..., \theta_T\}$, where θ_t represents the states of all humans at the tth frame. It can be decomposed by $\theta_t = \{M_t, X_{1,t}, ..., X_{M,t}\}$ where M_t is the number of human at time it t (determined in Stage 1), $X_{m,t}$ is the state of the mth person. This state includes the pose, shape and clothing parameters.

The observed shape of a moving person tends to change due to clothing and posture. Therefore, the shape parameters are dynamic to allow deformation so that the synthesized human is aligned to the input more accurately. The observed images, denoted as $\{I_1, I_2, ..., I_T\}$, and are assumed to be conditionally dependent on the current states only.

The prior distribution of the state, denoted by $p(\theta_1, \theta_2, ..., \theta_T)$, can be decomposed into prior distributions and a series of conditional distributions.

$$p(\theta_1, \theta_2, ..., \theta_T) = \frac{1}{z} \prod_{m=1}^{M_1} p(X_{m,1}) \prod_{t=1}^{T-1} \prod_{m=1}^{M_t} p(X_{m,t+1}|X_{m,t})$$

where Z is a normalization constant (we simplified the above expression by assuming all tracks start from t=1). The prior distribution is learned from a training set of human poses in static image and sets of motion capture data. The conditional distribution is based on a zeroth-order dynamic model and is approximated by a normal distribution.

$$p(X_{m,t+1}|X_{m,t}) \approx \mathrm{N}(X_{m,t+1} - X_{m,t}, \Sigma), \tag{1}$$

where Σ is the covariance matrix of the dynamic model and is learned from motion capture data.

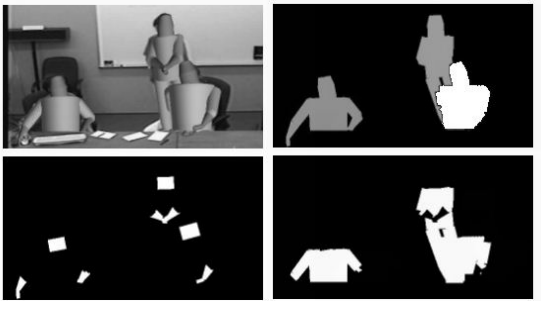

Fig. 7. Joint likelihood. *Top left*: predicted human poses; *top right*: predicted foreground regions; *bottom left*: predicted skin regions; *bottom right*: predicted non-skin regions.

A state candidate θ_t is evaluated by a likelihood function denoted by $p(I_t|\theta_t)$. We formulate the image likelihood function as consisting of four components, based on (*i*) region coherency, (*ii*) color dissimilarity with background, (*iii*) skin color and (*iv*) foreground matching, respectively.

$$p(I_t|\theta_t) = L_{region}(I_t, \theta_t) \times L_{color}(I_t, \theta_t) \times L_{skin}(I_t, \theta_t) \times L_{foreground}(I_t, \theta_t)$$

These likelihood components are described in detail in [7][8]; we have extended them to a joint likelihood measure for all humans in the scene that considers the inter-occlusion among them. Fig. 7 illustrates some of the synthesized variables that are generated when computing the likelihood measure.

4.2 Proposal Mechanisms

Different proposal mechanisms are used for the Markov chain transitions. We follow the procedure described in [8] and provide only a brief summary here for completeness. The MCMC approach uses a proposal function to generate state candidates. In theory, one can generate a candidate for the whole sequence of states $\{\theta_1, \theta_2, ..., \theta_T\}$ but such schemes have high computation complexity and difficult to implement. Instead, at each Markov transition, we update only the state of one person at one frame, $X_{m,t}$. From a current state, $X'_{m,t}$, a new state candidate, $X^*_{m,t}$, is generated by three types of evidence:

1. The estimation of previous state, $X_{m,t-1}$, can be propagated using a human dynamic model to generate candidates for the current state. We denote this proposal as $q(X^*_{m,t}|X_{m,t-1})$.

2. The candidates can be generated from the belief maps derived in the second stage. This is an adaptation of a bottom-up data-driven approach [21] that has now been used for a number of computer vision tasks [19][23]. In each belief map, the value at each pixel position represents the importance sampling probability of the corresponding joint's image position. The maps are used to generate pose candidates in a *component-based Metropolis-Hastings* approach. In [7], it is shown how this framework can be adapted for estimating 3D kinematics parameters by constructing reversible jumps using the belief maps and inverse kinematics computation; and it approximately satisfies the detailed balance requirement for MCMC. We denote this proposal function as $q(X^*_{m,t}|I_{m,t}, X'_{m,t})$, where $I_{m,t}$ represents the set of belief maps for the mth person derived in Stage 2.

3. Using backward-propagation, the next state estimates, can also be used to generate candidates for the current state. We denote this proposal as $q(X^*_{m,t}|X_{m,t+1})$.

The proposal distribution is denoted by $q(X^*_{m,t}|X_{m,t-1}, I_{m,t}, X_{m,t+1}, X'_{m,t})$, where $X'_{m,t}$ is the current Markov chain state. For simplicity, the distribution can be decomposed into its components:

$$\begin{aligned} &q(X^*_{m,t}|X_{m,t-1}, I_{m,t}, X_{m,t+1}, X'_{m,t}) \\ &= \alpha_1 q(X^*_{m,t}|X_{m,t-1}) + \alpha_2 q(X^*_{m,t}|I_{m,t}, X'_{m,t}) + \alpha_3 q(X^*_{m,t}|X_{m,t+1}) + \alpha_4 q(X^*_{m,t}|X'_{m,t}) \end{aligned}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the mixing ratios for the different components. The last component, $q(X^*_{m,t}|X'_{m,t})$, represents a proposal distribution derived from the current Markov state. It is implemented to involve both the random-walk sampler [5] and the flip kinematic jump [15] that is designed to explore the depth space [7].

4.3 Dynamic Proposals

Dynamic proposal mechanism involves generating a state candidate for the current frame, $X^*_{m,t}$, either from the estimates in the previous frame, $X_{m,t-1}$, or in the next frame, $X_{m,t+1}$. For the following discussion, we focus on the former.

The state estimation in the previous frame is represented by a set of state samples $\{X^1_{m,t-1}, X^2_{m,t-1}, ...\}$ generated by the Markov chain search. These samples are clustered to form a compact set of representative samples

$$\{X^{(1)}_{m,t-1}, X^{(2)}_{m,t-1}, ..., X^{(N)}_{m,t-1}\},$$

where N is the number of mixture components. We use N=50 in our experiments. These components are weighted according to their cluster sizes. To generate a candidate for the current frame, a sample $X^{(*)}_{m,t-1}$ is selected from the set of mixture components in the previous frame based on their normalized weights $\{w^{(1)}_{m,t-1}, w^{(2)}_{m,t-1}, ..., w^{(N)}_{m,t-1}\}$. Using a zeroth-order dynamic model, the state candidate is generated by sampling a normal distribution centered at $X^{(*)}_{m,t-1}$, with Σ as the covariance matrix from Equation (1).

4.4 Extracting Pose Trajectory

The previous section describes state estimation for each frame. The set of generated Markov samples can be represented compactly using a mixture model as described earlier. Using dynamic programming, an estimated trajectory of the each person can be obtained by "traversing" along the sequence and selecting a set of poses from these mixture components as in [8].

5 Experimental Results

In this section, we describe the experimental setup and discuss the result. We used a realistic sequence depicting a meeting room scene[1] [4]. We annotated the video manually to aid in evaluation by locating the image positions of the body joints. The depths of these joints, relative to the hip, were also estimated. The annotation data are used for evaluation only and not for training.

A set of training data is used to learn the prior distribution of state parameters, potential functions, dynamic model and observation models. These include motion capture data and annotated video sequences; and they are from sources

[1] The video was provided to us by the National Institute for Standards and Technology of the U.S. Government.

different from the test sequences and are of different scenes. The position of the table in the meeting room is annotated manually and provided to the system, and this information is used to infer occlusion of the body by the table.

5.1 Pose Tracking Results

The results of pose estimation are shown in Fig. 8. The initialization of each human model is automatic. The shape of the human model was initialized as the mean of the shape prior distribution.

As the results show, the proposed method is able to initialize and track the human poses robustly. The system is able to recover after partial self-occlusion or inter-occlusion. Estimation of the salient components including the face, torso and hands are fairly accurate. These help to boost the estimation of the other joints that are either less salient (e.g. elbows) or are temporarily occluded.

Some instances of temporary failure are observed due to lack of reliable observation, especially for the lower arms. Nonetheless, the results demonstrate the robustness of our approach in recovering from these partial failures.

For evaluation, we compare the estimated joint position with the annotated data. In the tth frame, we compute the 2D Euclidean distance error (in pixels) for the jth joint, denoted by e_t^j . A weighted average error, denoted by E_t, is defined by:

$$E_t = \left[\sum_{j=1}^{K} w_j e_t^j \Big/ \sum_{j=1}^{K} w_j \right],$$

where K is the number of joints used for evaluation and $\{w_j | j = 1, ..., K\}$ are the weights. The weights are chosen to approximate the relative size of the corresponding body parts, and the values are: 1.0 for torso and neck; 0.6 for shoulders, elbows and knees; 0.4 for wrists and ankles; 0.3 for head; and 0.2 for hand-tips. We ignore those joints that are always occluded, namely the lower

Fig. 8. Multiple persons pose tracking in meeting room scene

body joints of the sitting persons. The error for the pose estimation is 22.51 pixels or about 15.5cm (one pixel is approximately 0.69cm).

The experiment was performed on a 2.8GHz Intel PC in Windows XP and C++ programming code. For each frame, 1000 Markov state samples were generated in the third stage. The total processing for each frame took on average 5 minutes. We believe the computation can be improved significantly with later code optimization and the use of graphics hardware which we are currently exploring.

5.2 Discussion

We have shown how a novel three-stage approach using multi-level models can estimate and track poses accurately in highly realistic scene. This method allows us to perform a hierarchical estimation to overcome difficulties associated with realistic scene of multiple persons. By limiting the 2nd stage to 2D inference and using a grid-based representation, our method can efficiently integrate bottom-up observations with belief propagation using deterministic computation. Overall, the computation cost is slightly higher compared with that in [14] where nonparametric belief propagation is used to infer 3D pose directly (both methods run at several minutes per frame), but our system handles monocular views and considers inter-occlusion and non self-penetration constraints which we believe are essential for general applications related to event recognition and stored-video analysis.

The test sequences are different from the training data we used; this shows some generality of this model-based approach. A strength of this method is the ability to perform automatic initialization and recover from partial track failures due to occlusion. This is achieved without prior learning of the person's specific appearance or movement; these constraints are important in video understanding applications.

Acknowledgment

This research was funded, in part, by the Advanced Research and Development Activity of the U.S. Government under contract # MDA-904-03-C-1786.

References

1. A. Agarwal, B. Triggs: "3D human pose from silhouettes by relevance vector regression," *CVPR* 2004.
2. K. Choo, D.J. Fleet: "People tracking with hybrid Monte Carlo," *ICCV* 2001.
3. J. Deutscher, A. Davison, I. Reid: "Automatic partitioning of high dimensional search spaces associated with articulated body motion capture," *CVPR* 2001.
4. J.S. Garofolo, C.D. Laprun, M. Michel, V.M. Stanford, E. Tabassi: "The NIST Meeting Room Pilot Corpus," *Proc. 4th International Conference on Language Resources and Evaluation* (LREC-2004), Lisbon, Portugal, May 26-28 2004.

5. W. Gilks, S. Richardson, D. Spiegelhalter: *Markov Chain Monte Carlo in Practice.* Chapman and Hall, 1996.
6. G. Hua, M. Yang, Y. Wu: "Learning to estimate human pose with data driven belief propagation," *CVPR* 2005.
7. M. Lee, I. Cohen: "Proposal Maps driven MCMC for Estimating Human Body Pose in Static Images," *CVPR* 2004.
8. M. Lee, R. Nevatia: "Dynamic Human Pose Estimation using Markov chain Monte Carlo Approach," *Motion* 2005.
9. G. Mori, X. Ren, A. Efros, J. Malik: "Recovering Human Body Configurations: Combining Segmentation and Recognition," *CVPR* 2004.
10. M. Isard: "PAMPAS: Real-valued graphical models for computer vision," *CVPR* 2003.
11. D. Ramanan, D. A. Forsyth: "Finding and tracking people from the bottom up," *CVPR* 2003.
12. T. J. Roberts, S. J. McKenna, I. W. Ricketts: "Human Pose Estimation Using Learnt Probabilistic Region Similarities and Partial Configurations," *ECCV* 2004.
13. G. Shakhnarovich, P. Viola, T. Darrell: "Face pose estimation with parameter sensitive hashing," *ICCV* 2003.
14. L. Sigal, S. Bhatia, S. Roth, M. J. Black, M. Isard: "Tracking Loose-limbed People," *CVPR* 2004.
15. C. Sminchisescu, B. Triggs: "Kinematic Jump Processes for Monocular Human Tracking," *CVPR* 2003.
16. C. Stauffer, W.Grimson: "Adaptive background mixture models for real-time tracking," *CVPR* 1999.
17. E.B. Sudderth, A.T. Ihler, W.T. Freeman, A.S. Willsky: "Nonparametric belief propagation," *CVPR* 2003.
18. E.B. Sudderth, M.I. Mandel, W.T. Freeman, A.S. Willsky: "Distributed occlusion reasoning for tracking with nonparametric belief propagation," *NIPS* 2004.
19. Z.W. Tu, S.C. Zhu: "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *PAMI* 24(5), pp. 657-672, 2002.
20. Y. Wu, G. Hua, T. Yu: "Tracking articulated body by dynamic Markov network," *CVPR* 2003.
21. "S. Zhu, R. Zhang, Z. Tu: "Integrating bottom-up/top-down for object recognition by data driven Markov chain Monte Carlo," *CVPR* 2000.
22. J. Zhang, R. Collins, Y. Liu: "Representation and Matching of Articulated Shapes," *CVPR* 2004.
23. T. Zhao, R. Nevatia: "Tracking Multiple Humans in Crowded Environment," *CVPR* 2004.