Human Posture Analysis under Partial Self-occlusion

Ruixuan Wang and Wee Kheng Leow

School of Computing, National University of Singapore,
3 Science Drive 2, Singapore 117543, Singapore.
{wangruix, leowwk}@comp.nus.edu.sg

Abstract. Accurate human posture estimation from single or multiple images is essential in many applications. Two main causes of difficulty to solve the estimation problem are large number of degrees of freedom and self-occlusion. Tree-structured graphical models with efficient inference algorithms have been used to solve the problem in a lower dimensional state space. However, such models are not accurate enough to formulate the problem because it assumes that the image of each body part can be independently observed. As a result, it is difficult to handle partial self-occlusion. This paper presents a more accurate graphical model which can implicitly model the possible self-occlusion between body parts. More important, an efficiently approximate inference algorithm is provided to estimate human posture in a low dimensional state space. It can deal with partial self-occlusion in posture estimation and human tracking, which has been shown by the experimental results on real data.

1 Introduction

Human posture estimation and human tracking try to accurately recover human posture from single or multiple images [8][9][13]. Accurately recovering posture is essential in many applications such as human-computer interaction, vision-based sport coaching, and physical rehabilitation.

Top-down approach is often used to estimate human posture, in which a 3D or 2D human body model is often required to generate a synthetic 2D image of the corresponding human posture. By measuring the similarity between the synthetic 2D image and the input image, the posture estimation can be updated iteratively. In general, there are many local minima in such an optimization problem, such that continuous local optimization methods are not effective [22]. To deal with the difficulty, prior motion models are often used to constrain the search space during optimization [17], although it is limited to estimating postures similar to those in the motion models. Another way is to find multiple local minima and choose the best one from them [3][23], but it requires more computation and also cannot guarantee to find the global minimum. In comparison, sampling method [12] may find the global minimum in a low state space, but directly sampling in the state space of body posture is infeasible because of the large number of degrees of freedom (e.g., 30) of human body.

By observing that the human body is in fact tree-structured, researchers often formulate the estimation problem by a tree-structured graphical model [24][11][26][7][18]. In the model, every body part is encoded by one node in the graph, and every edge connecting two nodes indicates that there are relationships between the two parts. Efficient inference algorithms exist (e.g., BP [28]) to recover the low dimensional (e.g., 6) pose of every body part. More importantly, sampling methods [7][24][11][26] can be used in the low dimensional pose space of each body part.

However, it is not accurate enough to formulate the problem by a tree-structured graphical model. In this model, it assumes that the image of each body part can be independently observed, while self-occlusion between body parts often happens in human motion. In such case, the image of one body part can not be independently observed because it may be partially or fully occluded by other body parts. Sigal *et al*. [20] tried to deal with partial self-occlusion by learning the likelihood of the observed image conditioned on the pose state of each body part. But learning is often a complex process and it is not easy to collect training images. What is more, such learned likelihood functions are limited to approximately estimating a small set of postures. Lee *et al*. [14] and Hua *et al*. [9] used detected part candidates to obtain proposal distributions for some body parts, which are then used to help approximately estimate postures even under partial self-occlusion. Good proposal distributions are essentially important in their methods. Sudderth *et al*. [25] explicitly modelled self-occlusion using factor graph in which one binary hidden variable is required for each image pixel. However, the large number of hidden variables inside the model make the inference algorithm more complicated.

In order to deal with partial self-occlusion in posture estimation, we use a more accurate graphical model by explicitly inserting a set of hidden variables between the state of human posture and the input image observation. Each hidden variable represents the 3D shape and the appearance of one body part, and the image observation of every body part depends on all the hidden variables. The possible self-occlusion between body parts can be implicitly modelled by the relative position between the 3D shapes of parts. In addition, the non-penetration between body parts can be explicitly modelled in the middle level of the model. More important, based on the new model, a novel and efficient approximate inference algorithm is developed to accurately estimate each body part's pose in a lower (i.e. 6) dimensional space. This algorithm is an annealed iteration process. In each iteration, conditional marginal distribution of each body part is estimated based on the estimation results of previous iteration. The relationships between body parts' states and the relationships between parts' states and the image observation are updated by an annealing factor in each iteration. Such annealed process can help to find the true posture with more probability even if the initial posture is far from the truth. This inference algorithm, without any learning process, can deal with partial self-occlusion in 2D posture estimation and human tracking, which has been shown by the experimental results on real data.

2 Related Work

In general there are two types of approaches to the related human posture estimation and articulated human tracking problems: top-down and bottom-up. Compared with top-down approach introduced above, bottom-up approach can avoid the need for explicit initialization and 3D or 2D body modelling and rendering. It directly recovers human posture from images by exemplar based method or non-linear mapping based method.

The exemplar based method [16][2] searches for exemplar images similar to the input image from a set of stored exemplars, and uses the known 3D posture of the exemplar as the estimated posture. Since multiple body postures may have very similar corresponding images, this method often outputs multiple 3D body posture estimations for the input image. Much computation can be saved by constructing a distance-approximating embedding [2], such that the similarity measurement between images can be efficiently computed in the embedded low space. Because the exemplars record only a limited number of body postures, this method may not obtain good posture estimations if the body posture in the input image is different from those in the exemplars.

The non-linear mapping based method learns a nonlinear mapping function that represents the relationships between body image features and the corresponding 3D body postures. During learning, a rich set of image features (e.g., silhouette [6], histogram of shape context [1]) are extracted from each training image as the input, and the output is the known 3D posture in the corresponding training image. Agarwal and Triggs [1] used relevance vector machine to learn a nonlinear mapping function that consists of a set of weighted basis functions. Rosales et al. [19] used a special combination of sigmoidal and linear functions to learn a set of forward mapping functions by one EM technique. In addition, by embedding the manifold of one type of human motion into a lower dimensional space, and learning the two non-linear mappings between the embedded manifold and both visual input (e.g., silhouette) space and 3D body pose space, 3D body pose can be estimated from each input image by the two mapping functions [6]. The mapping based method can directly estimate body posture from a single input image, but it is often limited to recovering the body postures which are similar to the 3D postures in the training images.

Recently, the combination of top-down and bottom-up approaches has also been used to estimate postures [10][14][9][15]. In general it firstly applies low-level feature detectors (e.g., rectangle detectors [10]) to generate a set of candidates of body parts, then applies some prior knowledge or constraints (e.g., kinematic constraints) to search for good candidates and find the best 2D posture. To list a few, Mori [15] used superpixels as the element to represent the input image. Based on the boundaries of superpixels and constraints (appearance and width consistency, kinematic constraints) between body parts, a rough 2D posture configuration was obtained. Hua [9] used the detected candidates of some body parts to form importance function for later belief propagation.

Note that both types of approaches can be used in human tracking problem. Compared to CONDENSATION [12] which efficiently combines top-down approach into a probabilistic framework for human tracking, Sminchisescu [21] recently proposed a probabilistic framework in which conditional density can be propagated temporally in discriminative (bottom-up), continuous chain models.

3 Problem Formulation

A human skeleton model (Figure 1(a)) is used to represent body joints and bones, and a triangular mesh model (Figure 1(b)) is used to represent the body shape. Each vertex in the mesh is attached to the related body part. The relative bone length and part width to a standard human model are used to represent each body part's shape size.

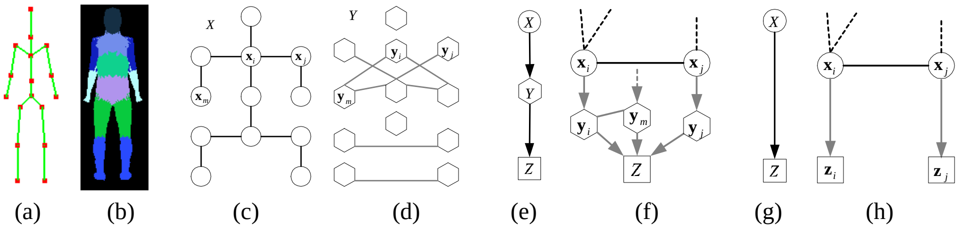


Fig. 1. Human body model and graphical model. (a) Human skeleton model. (b) Each vertex in the mesh model is assigned to one specific body part. (c) A tree-structured graph represents human posture $\mathcal{X}$. Each node represents one body part $\mathbf{x}_i$ and the edge between two nodes represents the potential relationship between them. (d) Each node in $\mathcal{Y}$ represents one 3D body part $\mathbf{y}_i$ and the edge between nodes represents the non-penetration relationship between them. (e) and (f) represent the graphical model we used. (g) and (h) represent the tree-structured graphical model.

Human body posture $\mathcal{X}$ is represented by a set of body parts' poses $\mathcal{X} = \{\mathbf{x}_i | i \in \mathcal{V}\}$ (Figure 1(c)), where $\mathcal{V}$ is the set of body parts. The pose $\mathbf{x}_i = (\mathbf{p}_i, \boldsymbol{\theta}_i)$ represents the i^{th} body part's 3D position $\mathbf{p}_i$ and 3D orientation $\boldsymbol{\theta}_i$. Given the shape size of each body part, a synthetic 3D body part $\mathbf{y}_i = f(\mathbf{x}_i)$ (Figure 1(d)) is generated for each part's pose $\mathbf{x}_i$, where f represents (but is not limited to) a deterministic process. By projecting the synthetic 3D body $\mathcal{Y} = \{\mathbf{y}_i | i \in \mathcal{V}\}$, a synthetic image observation can be generated. During posture estimation, each synthetic image observation will be used to compare with a real input image observation $\mathcal{Z}$. The relationship between $\mathcal{Y}$ and $\mathcal{Z}$ is represented by the observation function $\phi(\mathcal{Y}, \mathcal{Z})$. In addition due to the articulation, every pair of adjacent body parts $\mathbf{x}_i$ and $\mathbf{x}_j$ must be connected. Such kind of kinematic constraint is enforced by the potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$. Denote $\mathcal{E}$ as the set of adjacent body parts $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., $(i, j) \in \mathcal{E}$. Another kind of constraint is that body parts cannot penetrate each other, which can be enforced by potential function $\varphi_{im}(\mathbf{y}_i, \mathbf{y}_m)$. Denote $\mathcal{E}'$ as the set of part pair $\mathbf{y}_i$ and $\mathbf{y}_m$, i.e., $(i, m) \in \mathcal{E}'$.

A graphical model (Figure 1(e) and 1(f)) is used to represent all the relationships introduced above. Note that this model is different from the tree-structured graphical model that is generally used by other researchers [7][9]. In the tree-structured model (Figure 1(g) and 1(h)), it assumes that the image $\mathbf{z}_i$ of each body part i can be independently observed such that the relationship between $\mathbf{x}_i$ and $\mathbf{z}_i$ can be easily evaluated using local observation. However in general, self-occlusion between body parts often happens in human motion. In such a case, local observation $\mathbf{z}_i$ can not be observed independently and only the whole body's image observation $\mathcal{Z}$ can. In our graphical model, a middle level ($\mathbf{y}_i$) is inserted between $\mathbf{x}_i$ and $\mathcal{Z}$ in order to precisely model the image generation process. Each hidden variable $\mathbf{y}_i$ represents the 3D shape and appearance of one body part, and the image observation of every body part depends on all the hidden variables. This is different from tree-structured model in which every part's observation depends only on the part's state. In our graphical model, the possible self-occlusion between body parts can be implicitly modelled by the relative position between the 3D shapes of parts. In addition, the non-penetration relationship between 3D body parts can be enforced by potential function $\varphi_{im}(\mathbf{y}_i, \mathbf{y}_m)$, while such relationship cannot be modelled in tree-structured graphical model.

The problem is to infer $\mathcal{X}$ and corresponding $\mathcal{Y}$ from $\mathcal{Z}$. From the structure of the graphical model (Figure 1(e) and 1(f)), the posterior distribution $p(\mathcal{X}, \mathcal{Y}|\mathcal{Z})$ can be factorized as

$$\begin{aligned} p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) &\propto p(\mathcal{Z}|\mathcal{Y})p(\mathcal{Y}|\mathcal{X})p(\mathcal{X}) \\ &\propto \phi(\mathcal{Y}, \mathcal{Z}) \prod_{(i,m)\in \mathcal{E}'} \varphi_{im}(\mathbf{y}_i, \mathbf{y}_m) \prod_{i\in \mathcal{V}} \delta(f(\mathbf{x}_i) - \mathbf{y}_i) \prod_{(i,j)\in \mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \tag{1}$$

where $\delta(\cdot)$ is the Dirac's delta function because $\mathbf{y}_i$ is a deterministic function of $\mathbf{x}_i$. Now the objective is to find the maximum *a posteriori* estimation $\mathcal{X}^*$ and corresponding $\mathcal{Y}^*$ which make $p(\mathcal{X}, \mathcal{Y}|\mathcal{Z})$ maximum.

4 Inference Algorithm

Instead of directly inferring $\mathcal{X}$ and $\mathcal{Y}$ from (1), we calculate the conditional marginal distribution $p(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$. Unfortunately, due to the complex structure of the graphical model, the generally used efficient belief propagation algorithm cannot be used to calculate $p(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$. Here we develop an approximate inference algorithm to calculate the maximum $p(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ by introducing into it the idea of simulated annealing. This algorithm is an annealed iteration process. In each iteration, every $p(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ is estimated based on the estimation of the other body parts from the previous iteration and the real input image $\mathcal{Z}$. Since the estimation is not accurate in the first several iterations, the relationships between different body parts are relaxed and loose at first, and then become more and more restricted with respect to iteration. The update of relationships is realized by an annealing factor. In the following, we first explain how annealing factor is introduced to the iterations. After that, we will design the potential functions and observation functions.

Denote $\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ as the estimation of the true $p^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z}) = \{p(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})\}^{\lambda_n}$ at iteration n, where $n = 0, ..., N-1$ and $\lambda_{N-1} > \ldots > \lambda_1 > \lambda_0$. When λ_n increases (linearly or exponentially) with respect to iteration n, the MAP estimation $\mathbf{x}_i^*$ and $\mathbf{y}_i^*$ will emerge more and more clearly, because $\{p(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})\}^{\lambda_n}$ is much larger at $\mathbf{x}_i^*$ and $\mathbf{y}_i^*$ than at other $\mathbf{x}_i$ values. For two adjacent iterations, we have the following approximations:

$$\begin{aligned} \tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z}) &\approx \{p^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})\}^{\lambda_{n+1}/\lambda_n} \\ &\approx \{\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\hat{\mathcal{X}}_{-i}^n, \hat{\mathcal{Y}}_{-i}^n, \mathcal{Z})\}^{\lambda_{n+1}/\lambda_n}, \end{aligned} \tag{2}$$

$$\begin{aligned} p^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z}) &\approx \int_{\mathcal{X}_{-i}, \mathcal{Y}_{-i}} \{\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i, \mathcal{X}_{-i}, \mathcal{Y}_{-i}|\mathcal{Z})\} \\ &= \int_{\mathcal{X}_{-i}, \mathcal{Y}_{-i}} \{\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{X}_{-i}, \mathcal{Y}_{-i}, \mathcal{Z})p^{(n)}(\mathcal{X}_{-i}, \mathcal{Y}_{-i}|\mathcal{Z})\} \\ &\approx \int_{\mathcal{X}_{-i}, \mathcal{Y}_{-i}} \{\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\hat{\mathcal{X}}_{-i}^n, \hat{\mathcal{Y}}_{-i}^n, \mathcal{Z})\tilde{p}^{(n)}(\mathcal{X}_{-i}, \mathcal{Y}_{-i}|\mathcal{Z})\} \end{aligned}$$

$$= \tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i | \hat{\mathcal{X}}^n_{-i}, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z}), \tag{3}$$

where $\mathcal{X}_{-i}$ is the set of body parts' poses except $\mathbf{x}_i$, and $\mathcal{Y}_{-i}$ is the set of 3D body parts except $\mathbf{y}_i$. $\hat{\mathcal{X}}^n_{-i}$ and $\hat{\mathcal{Y}}^n_{-i}$ are the corresponding estimations at iteration n. In (2), $p^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ is approximated by $\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\hat{\mathcal{X}}^n_{-i}, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z})$. Although it needs to be theoretically explored for such approximation, the approximation (3) may be reasonable at least due to the following observations. During the first several iterations, the relationship between part $\mathbf{x}_i$ and the other parts $\mathcal{X}_{-i}$ are so loose that they are independent. The second observation is that when iteration n is large enough, $p^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ will become a Dirac's delta like function. In both cases, the $\tilde{p}^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\hat{\mathcal{X}}^n_{-i}, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z})$ can be used to exactly represent $p^{(n)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$.

From (1) and (2), we can get

$$\begin{aligned}
&\tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z}) \qquad (4)\\
&\propto \alpha\phi^{(n+1)}(\mathbf{y}_i, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z}) \prod_{m\in \Gamma'(i)} \varphi^{(n+1)}_{im}(\mathbf{y}_i, \hat{\mathbf{y}}^n_m)\delta(f(\mathbf{x}_i) - \mathbf{y}_i)\prod_{j\in \Gamma(i)} \psi^{(n+1)}_{ij}(\mathbf{x}_i, \hat{\mathbf{x}}^n_j)\\
&\propto \alpha\{\phi^{(n)}(\mathbf{y}_i, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z}) \prod_{m\in \Gamma'(i)} \varphi^{(n)}_{im}(\mathbf{y}_i, \hat{\mathbf{y}}^n_m)\delta(f(\mathbf{x}_i) - \mathbf{y}_i)\prod_{j\in \Gamma(i)} \psi^{(n)}_{ij}(\mathbf{x}_i, \hat{\mathbf{x}}^n_j)\}^{\lambda_{n+1}/\lambda_n},
\end{aligned}$$

where $\Gamma(i) = \{k|(i,k) \in \mathcal{E}\}$ is the neighbor of body part i, and similarly for $\Gamma'(i)$. α is a normalizing factor including potential functions related to the other body parts. From (4), conditional marginal distribution can be updated iteratively. Also, we can get

$$\phi^{(n+1)}(\mathbf{y}_i, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z}) \propto \{\phi^{(n)}(\mathbf{y}_i, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z})\}^{\lambda_{n+1}/\lambda_n}, \tag{5}$$

$$\varphi^{(n+1)}_{im}(\mathbf{y}_i, \hat{\mathbf{y}}^n_m) \propto \{\varphi^{(n)}_{im}(\mathbf{y}_i, \hat{\mathbf{y}}^n_m)\}^{\lambda_{n+1}/\lambda_n}, \tag{6}$$

$$\psi^{(n+1)}_{ij}(\mathbf{x}_i, \hat{\mathbf{x}}^n_j) \propto \{\psi^{(n)}_{ij}(\mathbf{x}_i, \hat{\mathbf{x}}^n_j)\}^{\lambda_{n+1}/\lambda_n}. \tag{7}$$

Observation functions $\phi^{(n+1)}(\mathbf{y}_i, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z})$ and potential functions $\varphi^{(n+1)}_{im}(\mathbf{y}_i, \hat{\mathbf{y}}^n_m)$ and $\psi^{(n+1)}_{ij}(\mathbf{x}_i, \hat{\mathbf{x}}^n_j)$ will be updated based on (5) (6) and (7) in the $(n+1)^{th}$ iteration.

4.1 Potential Functions

Potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ can be used to enforce relationships between body parts i and j. In our work, $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is used to enforce kinematic constraints and angle constraints between two adjacent body parts. In this case, assuming part i is one neighbor of part j, we can get

$$\psi^{(n)}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \propto \psi^{(n)}_{ij1}(\mathbf{x}_i, \mathbf{x}_j)\psi^{(n)}_{ij2}(\mathbf{x}_i, \mathbf{x}_j), \tag{8}$$

$$\psi^{(n)}_{ij1}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}(T(\mathbf{x}_i) - \mathbf{p}_j; 0, \Lambda^n_{ij}), \tag{9}$$

$$\psi^{(n)}_{ij2}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1 & \text{if } \theta_{ij} \in \Theta_{ij} \\ a^n_{ij} & \text{otherwise} \end{cases}, \tag{10}$$

where $\psi^{(n)}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ represents the probability of $\mathbf{x}_i$ given $\mathbf{x}_j$. $\psi^{(n)}_{ij1}(\mathbf{x}_i, \mathbf{x}_j)$ is used to enforce kinematic constraints, where T is a rigid transformation that is obtained from position $\mathbf{p}_i$ and orientation $\boldsymbol{\theta}_i$ in the pose $\mathbf{x}_i$ and the size information of the i^{th} body part, and $\Lambda^n_{i,j}$ is the variance matrix of the gaussian function $\mathcal{N}$ in the n^{th} iteration. $\psi^{(n)}_{ij2}(\mathbf{x}_i, \mathbf{x}_j)$ is used to enforce angle constraints, where θ_{ij} is the angle between the two body parts' orientation $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$, Θ_{ij} is the valid angle range between body part i and j, and a^n_{ij} is a value between 0 and 1. Note that Λ^n_{ij} and a^n_{ij} are tuned based on (7).

Potential function $\varphi_{im}(\mathbf{y}_i, \mathbf{y}_m)$ is used to enforce non-penetration constraints between two related body parts i and m where

$$\varphi^{(n)}_{im}(\mathbf{y}_i, \mathbf{y}_m) = \begin{cases} 1 & \text{if } d_{im} > D_{im} \\ b^n_{im} & \text{otherwise} \end{cases}, \tag{11}$$

d_{im} is the minimum distance between part $\mathbf{y}_i$ and $\mathbf{y}_m$, and D_{im} is the allowable minimum distance between the two parts. b^n_{im} is a value between 0 and 1. b^n_{im} is tuned according to (6) and becomes smaller with respect to the iteration, which means the non-penetration constraints will be more and more enforced.

4.2 Observation Functions

Observation function $\phi(\mathcal{Y}, \mathcal{Z})$ measures the likelihood of $\mathcal{Z}$ given $\mathcal{Y}$. In order to measure the likelihood, the 3D body $\mathcal{Y}$ is projected, and then the similarity between the projected image and the real input image observation $\mathcal{Z}$ is computed to estimate the likelihood. Since $\phi(\mathcal{Y}, \mathcal{Z})$ is estimated by the similarity of the two whole images, it can deal with self-occlusion where one body part is partially occluded by others.

In our work, edge and silhouette were used as the features for the similarity measurement. Chamfer distance was used to measure the edge similarity. For the silhouette similarity, in addition to the overlapping area of the projected image and the human body image region in the input image, the chamfer distance from the projected image region to the body image region in the input image was also used. The relative weight between edge and silhouette similarity is experimentally determined. Note that the edge similarity was a value between 0 and 1 by normalizing the chamfer distance, such that that the scaling problem between the edge similarity and the silhouette similarity was avoided.

4.3 Nonparametric Implementation

Because of the non-Gaussian property of potential functions and observation functions, analytic computation of the functions is intractable. We use Monte Carlo method to search for each body part's state by iteratively updating conditional marginal distribution $\tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$, called Annealed Marginal Distribution Monte Carlo (AMDMC). In our algorithm, each distribution $\tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ is represented by a set of K weighted samples,

$$\tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z}) = \{(\mathbf{s}^{(n+1,k)}_i, \pi^{(n+1,k)}_i)|1 \leq k \leq K\} \tag{12}$$

where $\mathbf{s}^{(n+1,k)}_i$ is the k^{th} sample of the i^{th} body part state $\mathbf{x}_i$ in the $(n+1)^{th}$ iteration and $\pi^{(n+1,k)}_i$ is the weight of the sample. Note that $\mathbf{y}_i = f(\mathbf{x}_i)$ is a deterministic function and so it is not necessary in the nonparametric representation.

In each iteration, every $\tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ is updated based on (4). The update process based on the Monte Carlo method is described in the following:

1. Update potential functions and observation functions based on (5)–(11).
2. Compute estimation $\hat{\mathcal{X}}^n_{-i}$ of the other body parts from initialization or previous iteration result, and get $\hat{\mathcal{Y}}^n_{-i} = f(\hat{\mathcal{X}}^n_{-i})$.
3. Use importance sampling to generate new samples $\mathbf{s}_i^{(n+1,k)}$ from related marginal distributions of previous iteration. The related marginal distributions include the neighbors' and its own marginal distributions of previous iteration. The new samples are to be weighted in the following step to represent marginal distribution.
4. Update marginal distribution. For each new sample $\mathbf{s}_i^{(n+1,k)}$, compute $\mathbf{y}_i^{(n+1,k)} = f(\mathbf{s}_i^{(n+1,k)})$ and then calculate the weight $\pi_i^{(n+1,k)}$, where

$$\begin{aligned}\pi_i^{(n+1,k)} &= \phi^{(n+1)}(\mathbf{y}_i^{(n+1,k)}, \hat{\mathcal{Y}}^n_{-i}, \mathcal{Z}) \prod_{m \in \Gamma'(i)} \varphi_{im}^{(n+1)}(\mathbf{y}_i^{(n+1,k)}, \hat{\mathbf{y}}_m^n) \\ &\times \prod_{j \in \Gamma(i)} \psi_{ij}^{(n+1)}(\mathbf{s}_i^{(n+1,k)}, \hat{\mathbf{x}}_j^n). \qquad (13)\end{aligned}$$

   $\pi_i^{(n+1,k)}$ is then re-weighted and normalized because we use importance sampling to generate sample $\mathbf{s}_i^{(n+1,k)}$. The updated marginal distributions will be used to update marginal distributions in the next iteration.

Human body posture can be estimated from the set of marginal distributions. The sample with the maximum weight in $\tilde{p}^{(n+1)}(\mathbf{x}_i, \mathbf{y}_i|\mathcal{Z})$ can be used to represent the estimation $\hat{\mathbf{x}}_i^{n+1}$ of i^{th} body part's pose.

Our algorithm can be viewed as the generalization of annealed particle filtering [5]. When body posture $\mathcal{X}$ is a single high dimensional state rather than a set of body parts' states, our AMDMC algorithm will become exactly the annealed particle filtering. Another related algorithm is the modified nonparametric belief propagation (mNBP) [27]. mNBP can also deal with partial self-occlusion, but it is based on the tree-structured model and there is no theoretical foundation on the modification. While mNBP is tested on synthetic image for posture estimation, our AMDMC is on real image sequences.

5 Experimental Results

Evaluation of our AMDMC algorithm and comparison with some related work were performed using real image sequences. The first experiment evaluated the algorithm's ability to accurately estimating 2D human posture under partial self-occlusion from a single image. The second experiment evaluated its ability to tracking 2D human body from a monocular image sequence. Note that the algorithm can be applied to estimate 3D human postures when multiple images or image sequences are available.

5.1 Preprocess

In the real image sequences in which a TaiChi motion was recorded, the static background inside each image was removed by a statistical background model. And the human model was modified to fit the human body size in every image sequence.

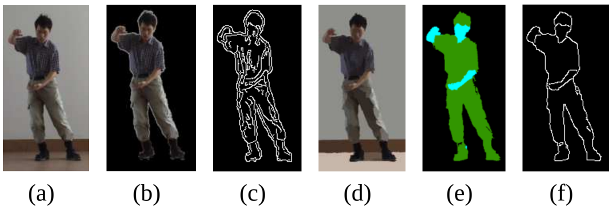


Fig. 2. Removing noisy edges inside each body part. (a) Input image. (b) Input image after background removal. (c) The extracted edge without removing noisy edges. (d) Over-segmentation result of (a) by mean shift. (e) Cluster every segment inside the body image region into one of the two clusters. (f) The edge extracted from (e).

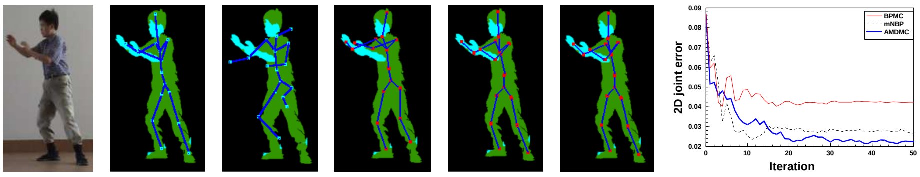
Note that the edge information plays an important role in posture estimation especially when limbs (e.g., arms) fall inside the torso image region. However, because of the variability in the appearance of human body, many noisy edges inside each body part will probably happen (Figure 2(c)). Mean shift [4] was used here to remove the noisy edges. Firstly from the first input image, the body image was manually divided into several clusters according to their appearance, and the histogram of each cluster was obtained. Then for subsequent input images, over-segmentation of the foreground image region was obtained by mean shift (Figure 2(d)), and every segment was classified into one cluster by comparing the histogram similarity between the segment and each cluster (Figure 2(e)). Of course because of similar appearance between symmetric body limbs, several body parts are often clustered together. In our experiment, just two clusters were used. One was for lower arms and the face, the other for the remained body parts. The noisy edges can be effectively removed by the above process (Figure 2(f)).

5.2 Posture Estimation under Partial Self-occlusion

Because of the depth ambiguity in a single image, 3D joint position cannot be estimated correctly. Therefore not 3D but 2D joint position error $E_{2D} = \frac{1}{nh}\sum_{i=1}^{n} \|\hat{\mathbf{p}}_{2i} - \mathbf{p}_{2i}\|$ was computed to assess the performance of our algorithm, where $\hat{\mathbf{p}}_{2i}$ and $\mathbf{p}_{2i}$ are the estimated and true 2D image position of the i^{th} body joint. $\mathbf{p}_{2i}$ was obtained by manually setting the image position of each body joint. h is the articulated body height and it is about 190 pixels in each 320×240 image.

In this experiment, 150 weighted samples were used to represent each conditional marginal distribution. The annealing related factor λ_{n+1}/λ_n was simply set to 0.9 in every iteration, and the total iteration number is 50. For each iteration, around 12 seconds was spent, most of which was used to generate synthetic image from mesh model and compute observation functions.

The skeletons in Figures 3(b) and 3(c) illustrate the 2D true posture and initial posture respectively. The skeletons in Figures 3(d), 3(e) and 3(f) illustrate the estimated posture using BPMC, mNBP, and our algorithm respectively. Both BPMC and mNBP [9] are efficient inference algorithms based on tree-structured model, and they have the same computation complexity as our AMDMC due to the similar structure of the graphical models. Note that the annealing factor was also used in BPMC and mNBP such that



(a) (b) (c) (d) (e) (f) (g)

Fig. 3. Posture estimation from a single image. (a) Input image. (b) 2D true posture represented by skeleton. (c) initial posture. Estimated posture by (d) BPMC, (e) mNBP and (f) our AMDMC. (g) AMDMC is better than BPMC and comparable to mNBP in posture estimation.

they can be compared with ours. Even the annealed BPMC was expected to obtain better result than the original BPMC, the pose estimation of some body parts (i.e., arms in Figure 3(d)) was not accurate when there was partial self-occlusion between body parts because local body image observation was used in BPMC's observation function. In comparison, the posture estimation was very close to the truth when using the mNBP and our AMDMC, in which the global body image observation was used in the observation function.

For this input image and the initial posture, Figure 3(g) represents the 2D joint position error E_{2D} with respect to the iteration number when using different inference algorithms. It shows that, after 10 to 15 iterations, the error has decreased to a relatively small value (i.e., 2.5% of body height h, or 5 pixels when h is 190 pixels) for both mNBP and AMDMC, but there was a higher error for BPMC.

Similar results have been reported on estimating body posture from a single image. Lee and Cohen [14] reported a higher 2D joint error which was 12 pixels when h was 150 pixels. Hua *et al.* [9] also reported a similar higher joint error. Both algorithms can deal with partial self-occlusion by detecting the possible positions of some body parts before posture estimation, while our AMDMC does not require any part detection in dealing with partial self-occlusion. In addition, Sigal *et al.* [20] provided an NBP algorithm which requires a complex learning process. They tested their algorithm on a simple walking posture using a multi-camera system. In comparison, our algorithm does not require any learning process and can deal with more complex postures.

5.3 Articulated Human Tracking

In the second experiment, a sequence of 88 real images were used for human tracking. The sequence was extracted from one original sequence of 350 images by sampling one from every four consecutive images. This will make human tracking more challenging because large posture difference between two adjacent frames will probably happen. In the tracking process, the initial posture for each image came from the estimated posture of previous frame image, while for the first frame image we manually set the initial posture. The annealing factor λ_{n+1}/λ_n was set to 0.85 for all 30 iterations. Because of the unknown true postures in the real images, the tracking result was visually compared with related algorithms. From Figure 4, we can see that both AMDMC and mNBP can accurately track human motion even under severe self-occlusion between two arms,

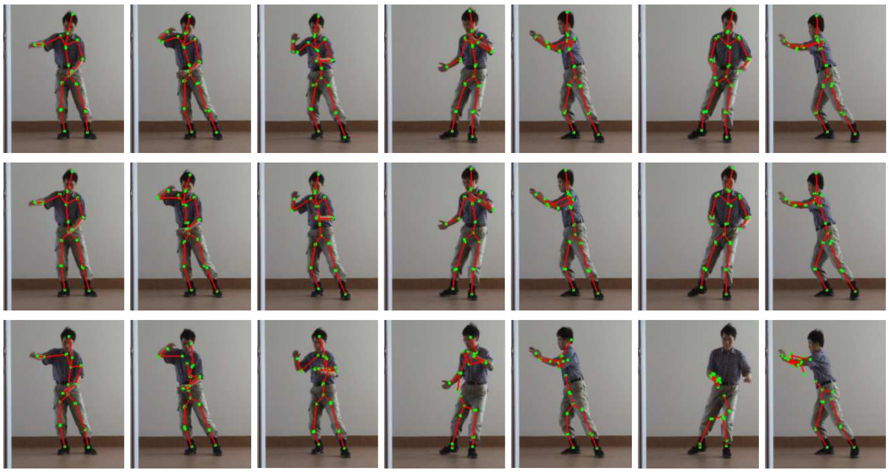


Fig. 4. Results of 2D articulated human tracking. The first row is the result by AMDMC; The second row is by mNBP; The third row is by BPMC.

while BPMC failed to track some body parts after a short sequence. The reason is clear. Because local image observation was used in BPMC to estimate each body part's pose, it is easy to fall into a local minimum in MAP estimation of the marginal distributions.

6 Conclusion

We presented a new graphical model and developed an efficient inference algorithm (AMDMC) to accurately estimate human postures under partial self-occlusion. The AMDMC algorithm can estimate human posture in a low dimensional state space by iteratively updating a set of body parts' marginal distributions. Experiments showed that the AMDMC algorithm can accurately estimate 2D human posture from a single image even if the initial posture was far from the truth and if there was partial self-occlusion between body parts. The AMDMC can be easily extended to articulated human tracking, which has been shown by the successful 2D articulated human tracking from a monocular image sequence. Compared to the existing techniques for posture estimation under self-occlusion, our AMDMC does not require any learning process or part detection beforehand. However in a monocular image sequence, it is difficult to discriminate the left body limbs from the right ones when human body viewed from the side. In such a case, prior motion knowledge or constraints must be explored in advance. Our future work is to extend the AMDMC algorithm to deal with more general cases in human posture and tracking by exploring motion models and human body constraints.

References

1. A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Proc. IEEE Conf. on CVPR*, pages 882–888, 2004.

2. V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *Proc. IEEE Conf. on CVPR*, pages 268–275, 2004.
3. T. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. IEEE Conf. on CVPR*, pages 239–245, 1999.
4. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002.
5. J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. IEEE Conf. on CVPR*, pages 126–133, 2000.
6. A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proc. IEEE Conf. on CVPR*, pages 681–688, 2004.
7. P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Int. Journal of Computer Vision*, 61(1):55–79, 2005.
8. D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.
9. G. Hua, M. H. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. In *Proc. IEEE Conf. on CVPR*, pages 747–754, 2005.
10. S. Ioffe and D. Forsyth. Finding people by sampling. In *Proc. IEEE Conf. on ICCV*, pages 1092–1097, 1999.
11. M. Isard. Pampas: Real-valued graphical models for computer vision. In *Proc. IEEE Conf. on CVPR*, pages 613–620, 2003.
12. M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. ECCV*, pages 343–356, 1996.
13. X. Lan and D. P. Huttenlocher. Beyond trees: common-factor models for 2D human pose recovery. In *Proc. IEEE Conf. on ICCV*, pages 470–477, 2005.
14. M. W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *Proc. IEEE Conf. on CVPR*, pages 334–341, 2004.
15. G. Mori. Guiding model search using segmentation. In *Proc. IEEE Conf. on ICCV*, 2005.
16. G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. ECCV*, pages 666–680, 2002.
17. A. H. R. Urtasun, D. J. Fleet and P. Fua. Priors for people tracking from small training sets. In *IEEE Int. Conf. ICCV*, 2005.
18. D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. IEEE Conf. on CVPR*, pages 271–278, 2005.
19. R. Rosales, V. Athitsos, and S. Sclaroff. 3D hand pose reconstruction using specialized mappings. In *Proc. IEEE Conf. on ICCV*, pages 378–385, 2001.
20. L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *Proc. IEEE Conf. on CVPR*, pages 421–428, 2004.
21. C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *Proc. IEEE Conf. on CVPR*, pages 390–397, 2005.
22. C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3D human tracking. In *Proc. IEEE Conf. on CVPR*, pages 69–76, 2003.
23. C. Sminchisescu and B. Triggs. Building roadmaps of minima and transitions in visual models. *Int. Journal of Computer Vision*, 61(1):81–101, 2005.
24. E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *Proc. IEEE Conf. on CVPR*, pages 605–612, 2003.
25. E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *NIPS*, 2004.
26. E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Visual hand tracking using nonparametric belief propagation. In *IEEE CVPR Workshop on Generative Model based Vision*, 2004.
27. R. Wang and W. K. Leow. Human body posture refinement by nonparametric belief propagation. In *IEEE Conf. on ICIP*, 2005.
28. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical report, MERL, 2002.