Multiple People Tracking and Pose Estimation

Feifei Huo, Emile A. Hendriks
Vision Lab, Delft University of Technology, The Netherlands
f.huo@tudelft.nl, e.a.hendriks@tudelft.nl

Keywords: Multiple People Tracking, Pose Estimation, Probability Estimation

ABSTRACT
In this paper we present a combined probability estimation approach to detect and track multiple people for pose estimation at the same time. It can deal with partial and total occlusion between persons by adding torso appearance to the tracker. Moreover, the upper body of each individual is further segmented into head, torso, upper arm and lower arm in a hierarchical way. The simplicity of the feature and the simplified model allow close real time performance of the tracker. The experimental results show that the proposed method can deal with most of the inner-occlusion between persons, as well as certain self-occlusion. It's also much faster than the existing methods with comparable accuracy.

1. INTRODUCTION
Recently, multiple people tracking and pose estimation has drawn more and more attention due to its large applications in surveillance, pose recognition, understanding interactions between persons, etc [1]. Compared with a single person situation [2, 3], multiple person tracking and pose estimation has more challenges, such as dealing with occlusion between persons and self occlusion. The occlusion is always a difficult problem to deal with, due to its hard to predict behavior. For instance, persons may change their directions after occlusion or still move in the same direction. Apart from this, the appearance of people may also change, such as from frontal view to lateral view. Additionally when people are very close to each other, it is quite easy to cause shadows on the body. All of this makes it a challenge to cope with people occlusion in a tracking system.

2. RELATED WORK
There is a large amount of research on multiple people tracking and pose estimation. In [4], they propose an approach for automatic initialization and tracking of human poses. It combines bottom-up evidence with top-down priors to get efficient pose estimation. The proposed algorithm can handle most of the self-occlusion by using the appearance model and the occlusion map. However, the runtime is around 45 seconds per frame due to the search of possible part configurations. It is not very suitable for real time applications. In [5], the appearance model of a person is learned from the video data. Then the person is tracked by detecting the learned model in each frame. The advantage of the approach is that people can be accurately tracked from a single view in front of complex backgrounds. But the appearance model needs to be built before the online tracking.

In this paper, we introduce a fast multiple people tracking and pose estimation approach by combing several probabilities derived from image observation. The flowchart of the proposed system is shown in Fig. 1. In the first step, the system detects different body parts by using an initial pose. Then in the next step, the appearance model of torso and arm is build. In the third step, different body parts are segmented in a hierarchical way for body part detection and tracking. A combined probability approach is used to estimate the upper-body pose. In the last step, in order to deal with inner-person occlusion, torso color histogram is used to distinguish different persons. The main contributions of our paper are, tracking of multiple persons, dealing with inner occlusion between persons and body parts segmentation in a real time system.

Fig. 1. The flowchart of the proposed system.

3. METHODOLOGY
In our approach, a generative body template is used to represent the upper body configuration as shown in Fig. 2. It is composed of two parts, a 2D-upper-body model of torso and head [6], and four 2D rectangles for upper and lower
The parameters in the 2D upper-body-model are torso position and scale and the model is described as 

\[ h = \{x_t, y_t, \text{scale} \} \]

The upper and lower arms are modeled by image patches \( a_u = \{x_u, y_u, \theta_u\} \) and \( a_l = \{x_l, y_l, \theta_l\} \).

Here \((x_t, y_t)\) and \((x_u, y_u)\) are the x and y coordinates of torso center, shoulder and elbow, \( \theta_t \) and \( \theta_u \) are the angles of upper and lower arms with respect to the main torso direction. Assuming there is a single person in the current frame, the parameters for different body parts are put together into one state vector:

\[ X_t = \{h, a_u^l, a_l^l, a_u^r, a_l^r\} \tag{1} \]

where \(h\) represents torso and head parameters, \(a_u^l\) represents left upper arm parameters, \(a_l^l\) represents left lower arm parameters, \(a_u^r\) represents right upper arm parameters, \(a_l^r\) represents right lower arm parameters. We define the prior probability of upper body pose as:

\[
p(X_t) = p(h) \cdot p(a_u^l, a_l^l, a_u^r, a_l^r|h) \\
= p(h) \cdot p(a_u^l, a_l^l|h) \cdot p(a_u^r, a_l^r|h) \\
= p(X_{t,1}) \cdot p(X_{t,2}) \cdot p(X_{t,3}) \tag{2} 
\]

The prior probability \(p(h)\) is assumed to be Gaussian. The candidate sample set is generated on the foreground blob with different positions and scales. The prior probability of left arm conditioned on torso \(p(a_u^l, a_l^l|h)\) is derived from the loose connection between the start point of upper arm and left shoulder position. Here we use a Gaussian distribution to model \(p(a_u^l, a_l^l|h)\). It also can be other forms of distributions, such as uniform distribution, or more complex distribution learned from training data [11]. The prior probability of right arm conditioned on torso \(p(a_u^r, a_l^r|h)\) is also a Gaussian.

Although a similar upper body model has been used often [11, 12, 13], the main difference of our approach is that the model is described in a hierarchical way. Instead of putting all the parameters into one state vector \(X_t\), they are split into three state vector vectors:

\[ X_{t,1} = \{h\} \]
\[ X_{t,2} = \{a_u^l, a_l^l\} \tag{4} \]
\[ X_{t,3} = \{a_u^r, a_l^r\} \]

In this model, there are two assumptions: one is that both state vector \(X_{t,2}\) and \(X_{t,3}\) are depend on state vector \(X_{t,1}\); the other is that state vector \(X_{t,2}\) and \(X_{t,3}\) are independent. The first assumption is motivated by the fact that left arm and right arm should always be connected to the torso through shoulder joints. The second assumption is based on kinematic constrains. We assume that the movement of the person’s left arm is not related to the movement of a person’s right arm if the person is allowed to move freely. In certain cases, such as waking, running, jumping, etc, there are correlations between the movements of left arm and right arm. However, we do not want to limit the model to a certain motion.

A strong motivation of our model is that it can describe the desired probability as accurate as possible, but at the same time allow for real time processing. The proposed approach simplifies the model and reduces the state vector dimension. When particle filtering is used to estimate the state of a system, this advantage becomes more obvious, because the computation time is directly related to the dimension of the state. Since state vector \(X_{t,2}\) and \(X_{t,3}\) are dependent on state vector \(X_{t,1}\), a hierarchical way can be used to first segment torso and head, then left and right arms. The prior distributions of these three state vectors are indicated as \(p(X_{t,1}), p(X_{t,2}), p(X_{t,3})\). The image likelihood function of the candidate state vectors will be discussed in section 3.3.

3.1. Initialization step

In the initialization, the system automatically detects different body parts by using an initial pose. The initial pose is shown in Fig. 3. From the prior knowledge of this initial pose and from general geometrical properties of human beings, shoulder, hand and elbow positions are extracted in this initial frame, to be used for segmenting different body parts.

![Fig. 2. An upper body template.](image)

When people are appearing in the scene, the 2D-upper-body-model is used to fit the person’s head and torso on a foreground binary image. The foreground image is obtained by using background subtraction. The background image is built with a mixture of \(k\) Gaussian models [7]. In order to exclude shadows from the foreground image, a shadow
removing approach is also employed [8]. From the 2D model, we can roughly locate the person’s shoulders \((x_s, y_s)\). Then a simple skin color model [8] is utilized to detect person’s hands \((x_h, y_h)\). The initial pose is defined as people stretching both of their arms sideways. When the distance between person’s two hands is larger than a predefined threshold, the person is assumed performing the initial pose. In this case, the full arm length (FAL) can be obtained, which is the Euclidian distance between the shoulder \((x_s, y_s)\) and the hand \((x_h, y_h)\) on the same side of the body. The full arm length is equal to:

\[
FAL = \sqrt{(x_s - x_h)^2 + (y_s - y_h)^2}
\]

(5)

Since shoulder and hand are the two end points of one arm, the elbow can be considered as the middle point of the arm, which is \((x_e, y_e) = \text{mean}[(x_s, y_s), (x_h, y_h)]\). These body joints, shoulder, elbow and hand, are used to segment torso, upper arm and lower arm from the whole body configuration.

3.2. Appearance model building

From the previous initialization, torso, upper arm and lower arm are segmented. In order to deal with inner-person occlusion and self-occlusion, an appearance model for torso and arm is built. In our approach, a color histogram in HSV color space is used for modeling torso appearance. For the arm appearance modeling, a color vector in normalized RGB color space is used. It is a trade-off between accuracy and efficiency.

For torso modeling, the original image is first transferred from RGB to HSV color space. Then \(H, S\) components are used to construct a 2D color histogram in order to exclude the influence of lighting conditions. The color histogram of the torso model is represented as \(H_t\). We use the Bhattacharyya distance [9] to measure the appearance similarity between the torso appearance model \(H_t\) and candidate models \(H_c\). It is defined as:

\[
d(H_t, H_c) = \sqrt{1 - \sum_{i=1}^{m} \sqrt{H_t(i) \cdot H_c(i)}}
\]

(6)

where \(H(i)\) is the normalized histogram value of the \(i\)th bin. \(m\) is the total number of bins in the histogram. The Bhattacharyya distance is a number between 0 and 1. The value of 0 means a perfect match, while 1 is a total mismatch. The Bhattacharyya distance is used to derive the probability of the torso model appearing in the scene.

For arm modeling, the color vector is composed of normalized \(r\) and \(g\) component in RGB color space [4]. The color vector of the target arm model is described as \(V_t\):

\[
V_t = (r_1', g_1', r_2', g_2', \ldots, r_m', g_m')
\]

(7)

\(r_i', g_i'\) are the mean \(r\) and \(g\) component of the \(i\)th part in an image patch. \(m\) is the total number of partitions along the middle line of the candidate image patch \(a_i = (x_i', y_i', \theta_i)\) and \(a_t = (x_t, y_t, \theta_t)\) as shown in Fig. 4. Here an image patch is evenly segmented into \(m\) parts along the middle line. The value of \(m\) depends on the resolution of the image and the size of the image patch.

\[
\text{Fig. 4. The partition of an image patch.}
\]

Since the size of candidate image patches for upper arms and lower arms is much smaller than the torso, using color vector representation gives comparable accuracy as color histograms. However, calculating a color vector is faster than a color histogram. Specifically, when the number of candidate image patches is large, this advantage becomes important for real time application.

3.3. Body parts segmentation

After initialization and appearance modeling, a hierarchical method is used to segment different body parts in the subsequent video frames. For each frame, first head and torso region are segmented by using the 2D upper-body model, since head and torso give more robust vision cues, compared with other relative small body parts. From this 2D model, we can roughly locate the head, torso and shoulder position. The position of shoulder serves as the start point of the upper arm. In order to locate the upper arm, image patches \(a_s = (x_s, y_s, \theta_s)\) are generated around the shoulder position \((x_s, y_s)\). The length of these image patches (LIP) is equal to the half length of the full arm, denoted as \(LIP = FAL/2\). There is no large variation of the length, since most poses are in a 2D plane [10]. The width of the image patch (WIP) is kept constant. After the upper arm is located, the end point of the upper arm is supposed to be an elbow. Then the elbow position \((x_e, y_e)\) will be used as the starting point for lower arm searching. Image patches \(a_i = (x_i, y_i, \theta_i)\) are generated in the same way as for the upper arm.

The body part segmentation and tracking is formulated as the problem of state estimation. A combined probability estimation approach is used to track and segment each individual. For each person in one frame \(I_t\), three individual particle filters are used to estimate 2D poses in a hierarchical way. State vector \(X_{i,t}\) is estimated first. The image likelihood function \(p(I_t | X_{i,t})\) of a state candidate \(X_{i,t}'\) is denoted as:

\[
p(I_t | X_{i,t}') = \exp(-(1 - \rho_i) - (1 - \rho_d))
\]

(8)
where $\rho_d$ is the fitness coefficient of the 2D upper-body model on the foreground binary image [6]. Therefore, the first term gives the probability of how the torso and head model fits on the foreground image. $\rho_d$ is related to the torso appearance of the person. As mentioned in section 3.2, the torso appearance is modeled by a 2D color histogram $H_t$ in HSV color space and is measured by Bhattacharya distance. $\rho_d$ is given by:

$$\rho_d = 1 - d(H_t, H_c) = 1 - \sqrt{1 - \sum_{i=1}^{m} H_t(i) \cdot H_c(i)}$$  \hspace{1cm} (9)$$

After estimating state vector $X_{t,1}$, state candidates $X_{t,2}$, $X_{t,3}$ are generated. They are evaluated by image likelihood functions $p(I_t | X_{t,2})$ and $p(I_t | X_{t,3})$ respectively. $p(I_t | X_{t,2})$ is given by:

$$p(I_t | X_{t,2}) = \exp(-\alpha \cdot (1 - \rho_f) - \beta \cdot (1 - \rho_e) - \gamma \cdot (1 - \rho_e))$$  \hspace{1cm} (10)$$

where $\rho_f$ indicates how probable image patches fit on the foreground. It is described as:

$$\rho_f = \frac{NFP[a \in \{x,y,\theta\}]}{LIP \times WIP}$$  \hspace{1cm} (11)$$

$NFP$ counts the number of foreground pixels within image patch $a \in \{x,y,\theta\}$. $\rho_e$ gives the appearance similarity between the target color vector $V_t$ and the candidate image patch color vector $V_c$:

$$\rho_e = 1 - \frac{1}{m} \sum_{i=1}^{m} \sqrt{(r^i_t - r^i_c)^2 + (g^i_t - g^i_c)^2}$$  \hspace{1cm} (12)$$

$\rho_e$ is related to image patch edges. It is defined as the response to a parallel line filter, as in [4]. Instead of adding these three terms directly, we give each of them a weight $\alpha$, $\beta$ and $\gamma$. The value of the weight is not constant, but depends on the scene. For instance, if occlusion is detected, $\alpha$ is set to a small value, since $\rho_f$ will become unreliable. In the contrast, the value of $\beta$ and $\gamma$ will increase. So the weighting of different probabilities is related to the occlusion estimation and improves the robustness and accuracy of arm segmentation.

The calculation of $p(I_t | X_{t,3})$ is the same as $p(I_t | X_{t,2})$. After all the state candidates $X_{t,1}$, $X_{t,2}$ and $X_{t,3}$ are evaluated, the maximum a posterior poses at frame $t$ can be computed from:

$$\{X_{t,1}^\text{MAP}\} = \max_{X_{t,1}} p(I_t | X_{t,1})$$

$$\{X_{t,2}^\text{MAP}\} = \max_{X_{t,2}} p(I_t | X_{t,2})$$

$$\{X_{t,3}^\text{MAP}\} = \max_{X_{t,3}} p(I_t | X_{t,3})$$  \hspace{1cm} (13)$$

The posterior probability distribution $p(X_{t,1} | I_t)$ for the state vector $X_{t,1}$, at the current frame $t$, together with the temporal diffusion model $p(X_{t+1,1} | X_{t,1})$ forms the prior probability distribution $p(X_{t+1,1})$ for the next frame $t+1$. Since the movement of a person is not very predictable, $p(X_{t+1,1} | X_{t,1})$ is simplified as a Gaussian model. The posterior probability distributions $p(X_{t,2} | I_t)$, $p(X_{t,3} | I_t)$ are calculated in the same way.

$$p(X_{t+1,1} | X_{t,1}) \sim p(I_t | X_{t,1}) p(X_{t+1,1} | X_{t,1})$$  \hspace{1cm} (14)$$

$$p(X_{t,2} | I_t) \sim p(I_t | X_{t,2}) p(X_{t,2})$$  \hspace{1cm} (15)$$

3.4. Multiple people tracking

When there are multiple persons in the scene, we can simply track them separately. For the remaining of the paper, we assume to deal with a two persons’ scenario. However, this method can be easily extended to a situation with more than two persons. As soon as two persons are appearing in the scene (assume they do not have any occlusion in the initialization step), the appearance model of their torso $H_t$ is built (in section 3.2). When there is an overlap between these two persons, the similarity of torso color histogram is calculated for both of them. It will indicate which person is in the front and which person is occluded. The front person is always tracked. The body part segmentation of the front person is the same as described in section 3.3. An example frame of the front person being tracked is given in Fig. 5. The back (occluded) person is only tracked when he/she is appearing again. During occlusion, the joint locations of the back person maintain the same as in the last frame before occlusion.

![Fig. 5](image-url) A sample frame with occlusion between persons.

4. EXPERIMENTAL RESULTS

In order to test the proposed approach, we recorded two video sequences in our own lab and used one from the Multimedia and Geometry group in Utrecht University. They all have cluttered background. The video starts with empty background, and then people walk into the scene.
When people are standing in the middle of the scene and performing the initial pose (Fig. 3), the system automatically starts pose tracking. People are facing to the camera and allowed to have large movement of their arms. The recorded frame rate is 25 frames per second. We evaluated our approach against ground truth. The ground truth is obtained by manually labeling the body joints: head center, torso center, left elbow, left wrist, right elbow and right wrist. The joint error (in pixel units) is used to quantitatively measure the segmentation accuracy and is defined as follows:

\[
\text{Error} = \sqrt{(x_j - x_g)^2 + (y_j - y_g)^2}
\]

where \((x_j, y_j)\) is the obtained joint location, \((x_g, y_g)\) is the manually labeled ground truth.

For arm segmentation, we compared the method with occlusion estimation (line with circles in Fig. 6) to that without occlusion estimation (line with dots in Fig. 6).

For the first approach, the value of \(\alpha\), \(\beta\) and \(\gamma\) in \(p(U_{ij} | X_{ij}^c)\) will be changed when the occlusion is detected. For the second approach, the value of \(\alpha\), \(\beta\) and \(\gamma\) are kept constant and no occlusion is taken into account. The values of \(\alpha\), \(\beta\) and \(\gamma\) are chosen as 0.5, 0.2, 0.3. When occlusion is detected, they are modified to 0.1, 0.7 and 0.2 in the first approach. These values were empirically found to work well. The average joint error is used to measure the segmentation accuracy. The results are shown in Fig. 6. Fig. 6(a) is the segmentation results of person1 (front), while Fig. 6(b) is the results of person2 (back/occluded). From the results we can see that adapting the weight values of the front person improves the pose estimation in case of occlusion. For the back person (person2), since he is occluded, the joint locations just copied the ones in the last frame before occlusion. Therefore, the errors are the same for the two approaches, as shown in Fig. 6(b).

**Fig. 6.** Comparison of performance dealing with occlusion. The error is measured in pixels. (a) The segmentation results of person1. (b) The segmentation results of person2.

**Fig. 7.** Pose estimation results with (left column)/ without (right column) occlusion estimation.
In Fig. 7, some frames from our test data set are shown. The four frames in the left column show the segmentation results of using occlusion estimation. The right column shows the results without using occlusion estimation. From these frames, we can see that although the persons are wearing clothes with texture or similar color, the proposed approach can handle it quite well. The processing speed of our current implementation is 10 to 13 frames per second using a 2.66GHz Core Duo PC. With optimized implementation, the speed still can be increased. Here we only showed the experimental results from one video sequence. Similar results were obtained with the other videos.

5. CONCLUSION

In this paper, we introduced a combined probability approach to estimate human upper-body configuration in real time. It can deal with inner-occlusions between persons and certain self occlusion. We showed that using occlusion adaptive probability weights for arm segmentation improved the upper and lower arm position estimation. Because of the hierarchical segmentation approach, the processing can be done in real time and is suitable for practical applications. We will further extend multiple people pose estimation into pose recognition.

6. ACKNOWLEDGMENTS

This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

7. REFERENCES


