Supervised Particle Filter for Tracking 2D Human Pose in Monocular Video

Suman Sedai  Du Q. Huynh  Mohammed Bennamoun
School of Computer Science and Software Engineering
The University of Western Australia, AUSTRALIA
{suman, du, bennamou}@csse.uwa.edu.au

Abstract

In this paper, we propose a hybrid method that combines supervised learning and particle filtering to track the 2D pose of a human subject in monocular video sequences. Our approach, which we call a supervised particle filter method, consists of two steps: the training step and the tracking step. In the training step, we use a supervised learning method to train the regressors that take the silhouette descriptors as input and produce the 2D poses as output. In the tracking step, the output pose estimated from the regressors is combined with the particle filter to track the 2D pose in each video frame. Unlike the particle filter, our method does not require any manual initialization. We have tested our approach using the HumanEva video datasets and compared it with the standard particle filter and 2D pose estimation on individual frames. Our experimental results show that our approach can successfully track the pose over long video sequences and that it gives more accurate 2D human pose tracking than the particle filter and 2D pose estimation.

1. Introduction and Motivation

Image-based and video-based human pose estimation and tracking is a popular research area due to its large number of applications including surveillance, human computer interaction and content-based image retrieval. For example, in video-based smart surveillance systems, human poses can be used to analyze and identify human activities in a scene. Such systems involve detecting the human body parts and their movements in the image frames.

Human pose estimation and tracking systems are broadly based on three different approaches: discriminative methods, generative methods and hybrid methods. In discriminative methods, a direct forward mapping function between image features and pose parameters is obtained using supervised learning [1, 2, 10]. Although discriminative methods are fast, they can provide incorrect predictions for the new inputs if trained using small datasets. Moreover, the relationship between image features and the human pose is often multimodal. For example, when a human silhouette is used as the image feature, one silhouette can be associated with more than one pose, resulting in ambiguities. This ambiguity problem is not unknown to researchers. Technique to tackle the problem by learning the one-to-many relationships between image features and poses has been reported in the literature, e.g., [1, 10].

In generative methods, the output pose is estimated by searching the solution space for a pose that best explains the observed image features [6, 17]. In this approach, a generative model is constructed which measures how close the hypothesized pose is to the observed image features. A hypothesized pose that is most consistent with the observed image features is chosen as the output pose. The particle filter [9, 6] is a generative tracking method that estimates the pose in each frame using an estimate of the previous frame and motion information.

In hybrid methods, such as [14, 15], a discriminative mapping function can be used to generate a hypothesized pose. The correct pose can then be estimated by searching the space around the hypothesized pose using a generative method. Although the method of [14] can resolve ambiguities by validating the multimodal output by generative search, it is based on pose estimation in a single image and is unable to track the pose in a video sequence. In this paper, we propose a hybrid method that uses a discriminative model as well as the motion information to track the 2D pose of a human subject in each video frame. Our method involves firstly estimating the one-to-many mapping function from the silhouette space to the 2D pose space via supervised learning. The output of this learning process is a discriminative distribution, from which hypothesized 2D poses can be sampled. In the tracking step, we combine this distribution with the particle filter based on a stationary motion model. By doing so, ambiguities can be resolved more efficiently as the motion information also takes part during pose estimation in each frame. We name our approach a su-
pervised particle filter method. For the following sections of the paper, we will describe this tracking method in detail, the experiments that we conducted, and evaluation of our method against the standard particle filter and pose estimation.

The organization of the paper is as follows. Section 2 presents the supervised learning method for constructing the discriminative distribution. Section 3 presents the proposed supervised particle filter for human pose tracking. Section 4 provides experimental results and Section 5 concludes the paper.

2. Training Step

In the training step, we use supervised learning to estimate the multimodal mapping function between the shape descriptor space and the 2D pose space. These two spaces are detailed below.

2.1. Silhouette Feature Representation

Given an image, we first obtain the silhouette of a human subject using background subtraction [8]. We use the Histogram of Shape Context (HoSC) [2] descriptor for encoding the silhouette of the human subject. The HoSC descriptor is a vector quantization of the shape context (SC) descriptors, which are characterized by 12 angular bins and 5 radial bins computed at 400 points regularly sampled along the boundary of the silhouette. The vector quantization involves a soft-voting of each of the SC descriptors to each of the $m$ entries of the SC feature dictionary, which is obtained once and for all by k-means clustering of the combined set of shape context vectors for all the training silhouettes. The votings are accumulated to obtain an $m$-dimensional HoSC descriptor of the silhouette. We empirically found that the optimal size of the feature dictionary, $m$, to be 200. The dimensions of our HoSC descriptors were therefore set to 200. For the rest of the paper, our silhouette feature descriptors are denoted by $x$.

2.2. 2D Pose Representation

We represent the 2D pose by the image coordinates of the human body joints for the neck and the left and right wrists, elbows, shoulders, ankles, knees, and hips. With the head position included also in the vector, there are 14 joints altogether measured with respect to the pelvis joint (the root joint of the human skeletal model). The resultant 2D pose vector, denoted by $y = (u_1, v_1, \cdots, u_{14}, v_{14})^T$, is therefore of $p = 28$ dimensions.

In order to predict the 2D pose from the silhouette of different scales, we first scaled the pose vector of the training data by the factor of $1/h$ where $h$ is the height of the silhouette. The training of the mapping functions is performed in the common scale space. The relative joint locations predicted by the function are then scaled-back by multiplying the predicted joint locations by $h$. By this way, we can predict the 2D pose from the silhouette of different scales.

2.3. Mixture of Regressors

Our aim in the training step is to estimate the mapping $f : \mathbb{R}^m \rightarrow \mathbb{R}^p$ using supervised learning. Given $N$ training samples $T = \{x^{(i)}, y^{(i)}\}$, for $i = 1, \cdots, N$, the 2D pose vectors $\{y^{(i)} | \forall i\}$ are firstly divided into $K$ clusters using the k-means algorithm. Accordingly, the training set can be clustered into $K$ subsets $\{T_1, \cdots, T_K\}$ such that $(x^{(i)}, y^{(i)}) \in T_k$ if $y^{(i)}$ belongs to the $k$th cluster. The clustering allows $f$ to be modelled as piecewise mapping.

The multimodal relationship between the silhouette descriptors and the 2D pose vectors can be represented as a mixture of $K$ regressors. This relationship allows the pose vector $y$ to be estimated from any given silhouette vector $x$:

$$\hat{y} = \sum_{k=1}^{K} \beta_k(x) (f_k(x) + \epsilon_k),$$

where $\beta_k(x)$ is a $K$-class classifier which gives the probability that the $k$th regressor is selected to predict the given feature instance $x$; $f_k(x)$ is the regressor trained on the $k$th cluster of the 2D pose space; and $\epsilon_k$ is an error vector that determines the uncertainty in the prediction of $f_k$. We model $f_k$ using the Relevance Vector Machine (RVM), $\beta_k$ as the multinomial logistic function, and $\epsilon_k$ as random vectors drawn from a Gaussian distribution with zero mean and diagonal covariance matrix $\Lambda_k$. We can thus approximate the conditional distribution $p(y|x)$ as a mixture of Gaussian distributions:

$$p(y|x) \approx \sum_{k=1}^{K} \beta_k(x) N(f_k(x), \Lambda_k).$$

We call $p(y|x)$ henceforth a discriminative distribution.

Relevance Vector Machine As mentioned above, the Relevance Vector Machine regression [18] is used to approximate the mapping from the feature descriptor space to the pose space in each cluster $T_k$. Each regressor $f_k(x)$ is given by $f_k(x) = W_k \Phi_k(x)$, for $k = 1, \cdots, K$, where each $W_k$ is a $p \times N_k$ weight matrix and $N_k$ is the number of training samples in the set $T_k$; $\Phi_k(x)$ is an $N_k \times 1$ kernel vector whose $j$th element, $\Phi_k,j(x)$, is a Gaussian kernel centered at the training sample $x^{(j)}$ in the set $T_k$:

$$\Phi_k,j(x) = \exp \left( -c \| x - x^{(j)} \| ^2 \right).$$

The parameter $c$ in the equation above denotes the kernel width, which is determined empirically from the scatter matrix of the training feature vectors. Our goal here is to find the weight matrix $W_k$ such that $f_k(x)$ can generalize to the new observations. We use the method described in [18] to train the weight values...
from the training set \( T \). The RVM uses a Bayesian learning technique on the training data to find the weights of the regressors. It uses a Gaussian prior over the weight parameters, where the prior is controlled by an independent set of hyperparameters for each weight. The point estimate of the hyperparameters is obtained by optimizing the marginal likelihood (a.k.a type-II likelihood) [18]. The aforementioned error covariance matrix, \( \Lambda_k \), which denotes the average uncertainty in prediction over the training set \( T_k \), is computed from the error vectors \( e_k = y - f_k(x) \) as \( \Lambda_k = \frac{1}{N_k} \sum_i e_k(i) e_k(i)^T \).

**Training multi-class Classifier** The multi-class classifier \( \beta_k(x) \) which is modelled as a multinomial logistic regressor (softmax function) is simply

\[
\beta_k(x) = \frac{\exp(-v_k^T x)}{\sum_{j=1}^{K} \exp(-v_j^T x)},
\]

where the parameter vectors \( v_k \in \mathbb{R}^m \), \( \forall k \), are estimated from the training data \( \{ (x(i), c(i)) \}_{i=1}^{N} \), with \( c(i) \) being a \( K \)-dimensional vector where each component \( c_k(i) \) denotes the probability that feature vector \( x(i) \) belongs to the \( k^{th} \) cluster. We set \( c_k(i) = 1 \) if \( y(i) \) belongs to the \( k^{th} \) cluster, otherwise we set it to 0. The maximum likelihood estimation of these parameter vectors is then performed using the iteratively reweighted least squares method. We use the fast method based on the bound optimization technique described in [11] to train them.

3. Pose Tracking

3.1. Scaled prismatic model

In the tracking phase, we represent the 2D pose of the human body using the scaled prismatic model (spm) [12]. In this model, the 2D pose is represented by a set of links connected by joints, where each link is described by two parameters: the angle of the segment \( \theta_l \) relative to its parent segment and the length of the segment \( l_i \) in pixel unit. We take the torso segment as the root segment and the orientation of the torso segment is measured with respect to \( y \)-axis of image. To track the translation of the person in the image plane, we include the image coordinates, \((u_0, v_0)\), of the pelvis joint in the state vector also. Together with the 14 body segments the dimension of the 2D pose vector \( y = (u_0, v_0, \theta_1, l_1, \ldots, \theta_{14}, l_{14})^T \) becomes 30. We can see that this representation of \( y \) and its previous representations in Section 2.2 are interchangeable. The spm representation is useful for tracking because it describes the pose in terms of a kinematic chain of the body parts.

Before describing the details about the particle filtering and our supervised particle filtering techniques, we outline in the next subsection how we model the likelihood distribution.

3.2. Likelihood distribution

Given the image observation, denoted by \( r_t \), at time \( t \), the likelihood distribution \( p(r_t | y_t) \) measures how well a hypothesized pose vector \( y_t \) explains the image observation \( r_t \). In our method, the likelihood value is computed by matching the 2D human body model corresponding to the hypothesized pose with the observed image features. We use silhouette and edge features of the image to compute the likelihood of each pose state vector. By representing each body part of the 2D human body parts as isosceles trapezoidal planer patches as shown in Figure 1(a), the likelihood distribution is constructed using two cost measures: silhouette cost and edge cost as detailed below.

3.2.1 Silhouette Cost

The silhouette cost measures how well the image region projected by the hypothesized model fits with the interior of the observed silhouette. Given a hypothesized 2D pose vector \( y_t \) at time \( t \), we first generate a binary image \( H \) corresponding to the projection of \( y_t \) using the available camera calibration data. This binary image \( H \) has \( H(i,j) = 1 \) if the pixel \((i,j)\) corresponds to the hypothesized foreground and \( H(i,j) = 0 \) otherwise. An example of hypothesized foreground is shown in Figure 1(b). If \( Z \) is the observed silhouette image (Figure 1(c)), then the silhouette cost, \( C_{sil} \), is computed as

\[
C_{sil} = 1 - \frac{\text{Area}(H \cap Z)}{\text{Area}(H \cup Z)},
\]

where \( \cap \) and \( \cup \) denote, respectively, the intersection and union of the two sets of non-zero pixels; and \( \text{Area}(I) \) is the number of non-zero pixels in the binary image \( I \). When the image region for the hypothesized pose fits the foreground region bounded by the silhouette exactly, \( C_{sil} \) will give the lowest cost value of 0. When the two regions have zero overlap, \( C_{sil} \) will give the highest cost value of 1.
\subsection{Edge Cost}

The edge cost measures how well the boundary line of the body parts corresponding to the hypothesized model fits with the observed edge image. We detect edges using the Canny edge detector \cite{4}. Edges corresponding to the foreground subject are segmented from the background using the silhouette image. To compute the edge cost, the chamfer distance \cite{3} between the observed \((E_1)\) and hypothesized \((E_2)\) edge images is computed as \(C_{\text{edge}} = d(E_1, E_2)\) (refer \cite{3} for details). The final likelihood for a given hypothesized pose is approximated as follows:

\[
p(r_t | y_t) \approx \exp\left(- (C_{\text{sil}} + \omega C_{\text{edge}}) \right),
\]

where \(\omega\) is a constant used to normalize the magnitude of the edge cost. We need this normalization because the scale of the edge cost is higher than the scale of the silhouette cost and we want both edge and silhouette cost to equally contribute to the likelihood. We empirically found the optimal value of \(\omega\) to be 0.12 for our setting.

Note that \(r_t\) is not (and cannot be) explicitly defined. One may interpret that \(r_t\) represents a collection of observed image pixels associated with the true 2D pose \(\bar{y}_t\) for a specific camera view and that the likelihood distribution \(p(r_t | y_t)\) provides us information about how close \(y_t\) is to the true \(\bar{y}_t\) (large likelihood values indicate closeness).

\section{Particle Filter}

The particle filter is a Monte Carlo approximation to the sequential Bayesian estimation which propagates the posterior probability of the first order Markov process from time \(t-1\) to \(t\) through the following equation:

\[
p(y_t | R_t) = c p(r_t | y_t) \int p(y_t | y_{t-1}) p(y_{t-1} | R_{t-1}) dy_{t-1},
\]

where \(c\) is a normalization constant; \(y_t\) is the 2D pose state at time \(t\); \(r_t\) is the image observation, \(R_t = \{r_1, \cdots, r_t\}\) represents the history of observations up to time \(t\); \(p(y_t | y_{t-1})\) is a distribution that describes the transition based on the motion model; and \(p(r_t | y_t)\) denotes the likelihood distribution. The multidimensional integral of Eq. (6) can only be evaluated for the simple case where the posterior distribution of the state variable is Gaussian. When the state variable corresponds to the human pose, the posterior distribution is non-Gaussian and methods, such as the Kalman filter, generally fail \cite{5}. The particle filter, on the other hand, approximates Eq. (6) using a set of weighted samples \(S_t = \{y_t^{(i)}, \pi_t^{(i)}\}_{i=1}^n\) where each \(y_t^{(i)}\) is a particle and \(\pi_t^{(i)}\) is the corresponding particle weight such that the weights are normalized to ensure \(\sum_i \pi_t^{(i)} = 1\). The particle filter does not make any explicit assumption about the form of the posterior and is therefore applicable to any general systems. In order to estimate the pose using the particle filter, one must design two distributions: the dynamics distribution \(p(y_t^{(i)} | y_{t-1}^{(i)})\) which describes the motion of the human subject from one frame to the next, and the likelihood distribution \(p(r_t | y_t)\), which gives the probability that observation \(r_t\) can be generated by a pose sample \(y_t\). At each time step \(t\), given the particle set \(S_{t-1}\), a basic sequential importance resampling updates the particles in three steps \cite{7}.

First, sample \(n\) particles from the \(S_{t-1}\) with replacement. In the second step, each sampled particle is then drifted according to distribution that describes the motion of the state. In the third step, the normalized importance weights for the particles are computed based on the observation likelihood. The set \(S_t\) comprises the new particles and their corresponding importance weights. The weighted-particle set \(S_t\) represents the posterior distribution of the state vector. The 2D pose state can be taken to be the expected value of these particles.

The classical particle filter require a large number of particles to work robustly in high dimensional spaces. Variants, such as the annealed particle filter \cite{6}, have therefore been developed. Moreover, the computed importance weights may not always be correct when the observations are noisy and/or when the model is not exact. This leads to frequent mis-tracking. To resolve this problem, we use a supervised particle filter which takes into account the discriminative distribution obtained from supervised learning (Section 2.3) to guide the filtering, as detailed in the following subsection.

\section{Supervised Particle Filter}

Let \(x_t \in \mathbb{R}^m\), \(y_t \in \mathbb{R}^p\), and \(r_t\) be defined the same as before. Let \(p(y_t | x_t)\) be the discriminative distribution and \(p(r_t | y_t)\) be the conditional likelihood. Let \(X_t = [x_1, \cdots, x_t]\) and \(R_t = [r_1, \cdots, r_t]\) be the history observations up to time \(t\). Then the posterior density of the state at time \(t\) is given by the recursive Bayesian equation:

\[
p(y_t | R_t, X_t) = c p(r_t | y_t) p(y_t | R_{t-1}, X_t),
\]

where \(c\) is a normalization constant. In (7), we adopt the following conditional independence: \(p(r_t | y_t, X_t) = p(r_t | y_t)\). The prior distribution at time \(t\) is given by

\[
p(y_t | R_{t-1}, X_t) = \int p(y_t | y_{t-1}, x_t) p(y_{t-1} | R_{t-1}, X_{t-1}) dy_{t-1}.
\]

We assume that the conditional distribution \(p(y_t | y_{t-1}, x_t)\) can be expressed as the mixture of simpler conditionals \cite{13}, i.e.,

\[
p(y_t | y_{t-1}, x_t) = (1 - \alpha) p(y_t | y_{t-1}) + \alpha p(y_t | x_t),
\]

where \(0 \leq \alpha \leq 1\) is the mixing coefficient for the two distributions. By substituting Eq. (9) into Eq. (8), the prior
distribution can be expressed as
\[ p(y_t | R_{t-1}, X_t) = (1 - \alpha)p(y_t | R_{t-1}, X_{t-1}) + \alpha p(y_t | x_t), \]
where
\[ p(y_t | R_{t-1}, X_{t-1}) = \int p(y_t | y_{t-1}) p(y_{t-1} | R_{t-1}, X_{t-1}) dy_{t-1} \]
is the first component of the prior distribution which is obtained from the motion model and the posterior distribution at \( t-1 \). The second component of the prior is the discriminative distribution \( p(y_t | x_t) \). We approximate the posterior at time \( t-1 \) using the weighted-particle set \( S_{t-1} = \{ y_{t-1}^{(i)}, \pi_{t-1}^{(i)} \}_{i=1}^{n} \) and assume a stationary motion model given by \( p(y_t | y_{t-1}) = \mathcal{N}(y_t, \Omega) \), where \( \Omega \) is a covariance matrix. We can write the first component of the prior distribution obtained from the dynamical model as a Gaussian kernel distribution \( p(y_t | R_{t-1}, X_{t-1}) = \sum_{i=1}^{n} \pi_{t-1}^{(i)} \mathcal{N}(y_{t-1}^{(i)}, \Omega) \). The second component of the prior distribution is the discriminative distribution obtained from supervised learning. The final prior of Eq. (10) can be expressed as
\[ p(y_t | R_{t-1}, X_t) = (1 - \alpha) \sum_{i=1}^{n} \pi_{t-1}^{(i)} \mathcal{N}(y_{t-1}^{(i)}, \Omega) + \alpha p(y_t | x_t), \]
where the importance weight, \( \pi_{t}^{(i)} \), can be computed as a normalized likelihood value using Eq. (5):
\[ \pi_{t}^{(i)} = \frac{p(x_t | y_t)}{\sum_{j=1}^{n} p(x_t | y_j)}. \]

At each time step \( t \), given a weighted-particle set at time \( t-1 \), a discriminative distribution \( p(y_t | x_t) \), and the image observation \( r_t \), the weighted-particle set at time \( t \) is obtained using the procedure outlined in Table 1. Note that a sample obtained from \( p(y_t | x_t) \) is a pose vector in terms of 14 joint locations relative to the pelvis joint. Hence we convert it to the absolute joint locations by adding the position of a pelvis joint to each joint location, where the pelvis joint position is obtained by sampling from the first component of the prior. We then augment the pelvis joint location at the beginning of the vector and convert the vector to \( spm \) representation. Hence all the samples in the particle set are obtained in \( spm \) representation.

In this paper, we set \( \alpha = 0.5 \) to denote equal contribution of the dynamics motion model and the discriminative model to estimate the pose. It can be seen that a standard particle filter is a special case of a supervised particle filter when \( \alpha = 0 \). When \( \alpha = 1 \), the method does not use motion information and so the system becomes one that performs pose detection by sampling from the discriminative distribution alone and validates the samples using the importance weights computed from the likelihood distribution. Hence, the method of [14] can be seen as special case of the SPF when \( \alpha = 1 \). By choosing a value of \( \alpha \) between 0 and 1, the output from the discriminative estimation can be combined with the dynamics distribution to obtain a more accurate and robust pose estimate. Our method also provides a stable tracking since at each time step, even if the dynamical model produces wrong sample, the samples obtained from the discriminative distribution are used to estimate the posterior distribution.

### 4. Experiments

We trained and evaluated our proposed 2D human pose tracking method using the HumanEva I datasets [16] provided by Brown University. The datasets contain video frames and corresponding 3D poses of human subjects carrying out actions such as walking, jogging and boxing. For each image, the corresponding ground truth 3D pose is given by the 3D coordinates of the 15 human body joint locations relative to the location of the pelvis joint. To construct the ground truth 2D pose vectors, we projected the 3D joint locations into camera view C1. We only used images taken by one camera since our focus was on 2D pose track-
ing from monocular video sequences. The ground truth 2D pose vectors were in the format as described in Section 2.2.

For all the images in the training set, we extracted the 200-dimensional HoSC shape descriptor vectors (see Section 2.1). We trained the regressor that maps the HoSC descriptors to the relative 2D joint locations using the supervised learning approach described in Section 2.3. We set the number of clusters \( K = 5 \) to allow sufficient partitioning of the pose space and trained a RVM regressor for each cluster. In the tracking phase, the output of the regressor is a discriminative distribution defined in terms of a mixture of conditional Gaussians. The discriminative distribution is combined with the stationary motion model to track the pose using our proposed SPF algorithm. The final output of the supervised particle filter is a 30-dimensional spm pose vector which is then converted to a 30-dimensional joint location vector using inverse kinematics. The joint location vector contains absolute locations of \( n_{\text{hm}} = 15 \) body joints in image coordinate system. Let the ground truth pose vector be \( \hat{y} \in \mathbb{R}^{30} \) and an estimated pose vector be \( \hat{y} \in \mathbb{R}^{30} \). We computed the 2D RMS error [16] as 
\[
e(y, \hat{y}) = \frac{1}{n_{\text{hm}}} \sum_{i=1}^{n_{\text{hm}}} \| m_i(y) - m_i(\hat{y}) \| \]
where \( m_i(y) \in \mathbb{R}^2 \) is a function of 2D pose that returns the position of \( i^{\text{th}} \) joint in an image.

Table 2 shows the mean RMS errors for the three methods on the walking and jogging test sets. Also included in the table are the standard deviations of these errors. The first method is the supervised particle filter (SPF) for which we set \( \alpha = 0.5 \) in Eq. (12) to denote equal mixing of the discriminative and dynamics motion distributions in pose tracking. For the second method, we set \( \alpha = 1 \), so that only the motion model was used for tracking. This case is equivalent to the standard particle filter (PF). In the third method we set \( \alpha = 0 \) so that motion model was not used to track the pose; instead, pose detection was performed for each video frame (only pelvis joint is tracked). This case is equivalent to the method of [14]. The results show that our SPF gave the smallest error values among the three methods. It also gave the smallest standard deviations denoting that the poses estimated using of our method are more stable. It can be seen that the standard particle filter produced the largest errors because racking failed at the early stage due to the high dimension of the state space. Although the pose detection method produced smaller errors than the PF, the standard deviations of the errors were larger than those from SPF.

Figure 2 shows the pose estimation errors in every 15th frame of the walking sequence for all three cases. The pose detection (i.e. \( \alpha = 0 \)) method produced higher error than SPF for almost all of the frames. The standard particle filter, on the other hand, did not perform well at all. It is clear from the error plot that the SPF can accurately track the pose over the frames and can recover from miss-tracking. It can be seen that SPF can provide more stable pose estimates than the other two methods. Figure 3 shows an example of multimodal pose output given by discriminative distribution and the correct output given by our method. Figure 4 displays some of the output poses predicted using our method. Our experiments show that our proposed supervised particle filter can effectively track 2D poses in video sequences.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervised PF</th>
<th>Pose detection</th>
<th>PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>6.8 ± 2.8</td>
<td>8.70 ± 3.7</td>
<td>27.65 ± 12.1</td>
</tr>
<tr>
<td>Jogging</td>
<td>10.4 ± 4.5</td>
<td>15.0 ± 9.1</td>
<td>44.5 ± 18.1</td>
</tr>
</tbody>
</table>

Table 2. Comparison of our SPF with the standard particle filter and pose detection method. The table shows the mean RMS errors in pixels between the estimated and ground truth joint locations. The test sequence corresponding to subject S2 is used for evaluation.

![Figure 2](image_url) Comparison of our SPF with the standard particle filter and pose detection method for Walking and Jogging test sequences. The chart displays the mean RMS error in pixels of image frame taken every 15th from the corresponding test set.

![Figure 3](image_url) An image from the test set and the corresponding 2D pose estimated using Mixture of RVM regressors. The 2D pose estimates of two RVMs with the highest probability are displayed as a skeleton, with red denoting left limbs and blue denoting right limbs. The associated probabilities are displayed below each output 2D skeleton. These two probable solutions denotes the pose ambiguities associated with the silhouette. Our SPF uses them as input and predicts the correct 2D pose as shown in right-most image. (Figure best viewed in color).
5. Discussion and Conclusions

We have presented a supervised particle filter for 2D human pose tracking in video sequences. Our method exploits the discriminative distribution obtained from supervised learning with a dynamics motion model to obtain an accurate and stable tracking of the 2D pose of a human subject in each video frame. Our method does not require initialization and can resolve pose ambiguities using motion information during tracking. Although this paper uses the stationary motion model for pose tracking, we believe that by using a category of learned motion models, the pose can be tracked even more accurately. In this paper, a mixing coefficient is used to adjust the contributions from the motion distribution and from the discriminative distribution. We have experimented the case where the contributions from both distributions are equal. Mixing the two distributions according to their confidence levels can potentially improve the tracking performance even further. Currently we are developing a method to dynamically find the mixing proportion for more robust 2D human pose tracking.

References


