Full-Body Human Pose Estimation from Monocular Video Sequence via Multi-Dimensional Boosting Regression

Yonghui Du, Yan Huang*, Jingliang Peng

School of Computer Science and Technology, Shandong University, China
yonghuid@gmail.com, yan.h@sdu.edu.cn, jpeng@sdu.edu.cn

Abstract. In this work, we propose a scheme to estimate two-dimensional full-body human poses in a monocular video sequence. For each frame in the video, we detect the human region using a support vector machine, and estimate the full-body human pose in the detected region using multi-dimensional boosting regression. For the human pose estimation, we design a joints relationship tree, corresponding to the full hierarchical structure of joints in a human body. Further, we make a complete set of spatial and temporal feature descriptors for each frame. Utilizing the well-designed joints relationship tree and feature descriptors, we learn a hierarchy of regressors in the training stage and employ the learned regressors to determine all the joint's positions in the testing stage. As experimentally demonstrated, the proposed scheme achieves outstanding estimation performance.

1 Introduction

Human pose estimation is an important research topic with many potential applications, *e.g.*, image- and video-based event detection, interactive video gaming and human-computer interaction. Nevertheless, accurate and efficient human pose estimation has been a challenging problem. Challenges mainly come from the fact that the human body is an articulated object with many degrees of freedom and there are too many complicating factors like clothing, lighting and occlusion.

During the recent years, intensive research has been conducted in human pose estimation. However, most of the published schemes work on a single still image, and many of them conduct pose estimation for only the upper human body. There have been very few works on estimating human poses in video sequences. In particular, two-dimensional (2D) full-body human poses estimation in monocular video sequences is largely underrepresented in the research, to the best of our knowledge.

In this work, we propose a scheme to estimate two-dimensional (2D) full-body poses of a human in a monocular video sequence, which extends an existent

* Corresponding author.

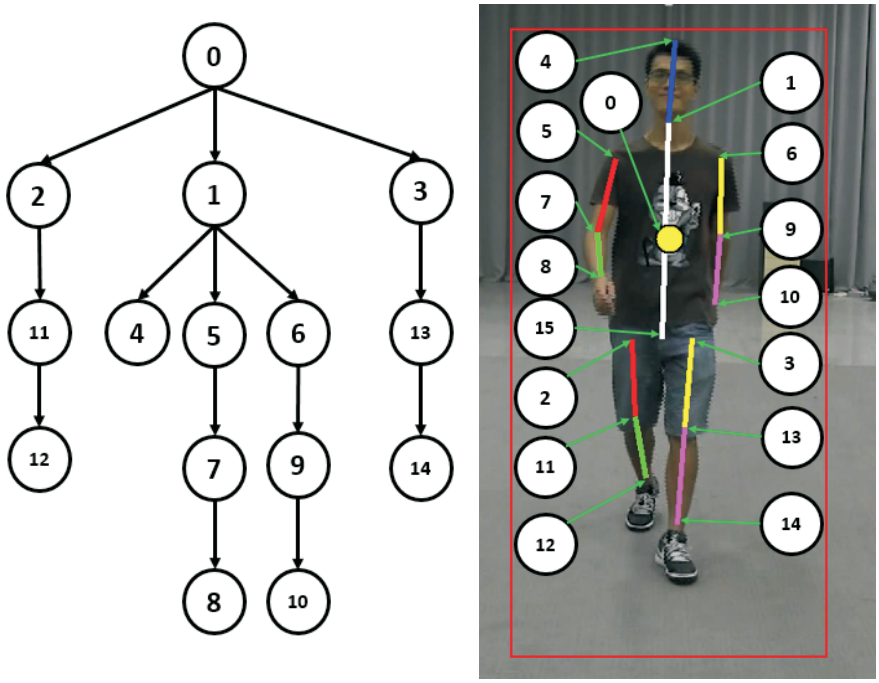

Fig. 1. Left: Joints Relationship Tree (JRT); Right: the joints marked in an image corresponding to the nodes of the JRT. The red rectangle represents the detected human region; a human body pose is represented as 10 sticks each connecting two joints in the image, which are head, torso, upper and lower arms and upper and lower legs; the yellow circle is at 1/2 width and 1/3 height of the human's bounding rectangle. Node 15 is at the center of Node 2 and Node 3.

scheme [1] for estimating upper-body human poses in still images. Compared with [1], major extensions of our work include: 1) we employ the support vector machine (SVM) method and the histogram of oriented gradients (HOG) descriptor to detect the human region in each frame, 2) we design a full-body JRT as the basic structure for pose representation and estimation, 3) we propose a motion feature descriptor and use it together with the spatial one to describe local image features, and 4) we construct a database of videos each with annotated full-body human regions and poses, which we use for both training and testing result verification.

The rest of this paper is organized as follows. Related work is introduced in Section 2, the proposed scheme is described in Section 3 and experimental results are given and analyzed in Section 4. Finally, this work is concluded in Section 5.

2 Related Work

Human pose estimation has been intensively researched in the past decade or so. We briefly review recent work in this section, while comprehensive surveys on earlier algorithms can be found in references [2, 3].

A large class of algorithms is based on the pictorial structures (PS) model [4–9], which represents the human body as a series of rigid parts and a set of relations between certain pairs of parts. Approaches extending the PS model have also been proposed, such as the deformable structures model [10, 11] and the cascade of pictorial structure models [12]. These structure-model-based algorithms require a large number of constraints and correspondingly intensive computation.

Methods have been proposed which use machine learning techniques for human pose estimation. Okada and Soatto [13] propose a piecewise linear regression method for human pose estimation. In their approach, they train several local linear regressors (which are based on pose clusters generated by K-means) and a support vector machine for estimating the human pose from the histogram of oriented gradients (HOG) feature vector. However, the accuracy of linear regression method is limited, due to the diversity of human poses. Dantone *et al.* [8] employ two-layered random forests as joint regressors, but this method needs a large search space, negatively affecting its computational efficiency. Hara and Chellappa [1] propose to use multidimensional output regression tree with dependency graph for upper-body human pose estimation. Their algorithm breaks a complex problem of human pose estimation down into a sequence of local pose estimation problems which are less complex. As a result, it achieves a good tradeoff between accuracy and efficiency.

Research has also been conducted recently on estimating the human pose from a single depth image. Some methods base on the random forest have been proposed, such as [14–16]. These methods work efficiently, assuming that the positions of human parts or labels of pixels are independent. Sun *et al.* [17] present a conditional regression forest model which takes the dependency relationships among the human parts into account. However, this method requires prior knowledge about torso orientation, human height, *etc.*

All the above-reviewed algorithms work on still images. By contrast, there has been far less work published on video-based human pose estimation. In particular, the work which can estimate two-dimensional (2D) full-body human poses in monocular video sequences is largely underrepresented in the research, to the best of our knowledge. Bissacco *et al.* [18] propose multi-dimensional boosting regression with appearance and motion to address the problem of three-dimensional (3D) human pose estimation in video sequences. They utilize both 2D videos and 3D motion data to train the regressors and the trained regressors map the Haar features of a detected human region directly to an entire set of 3D joint angles representing the full-body pose. Zuffi *et al.* [11] combine the dense optical flow with the deformable structures model [10] to address the upper-body human pose estimation in monocular video sequences. However, significant computation and memory use is introduced by the dense optical flow.

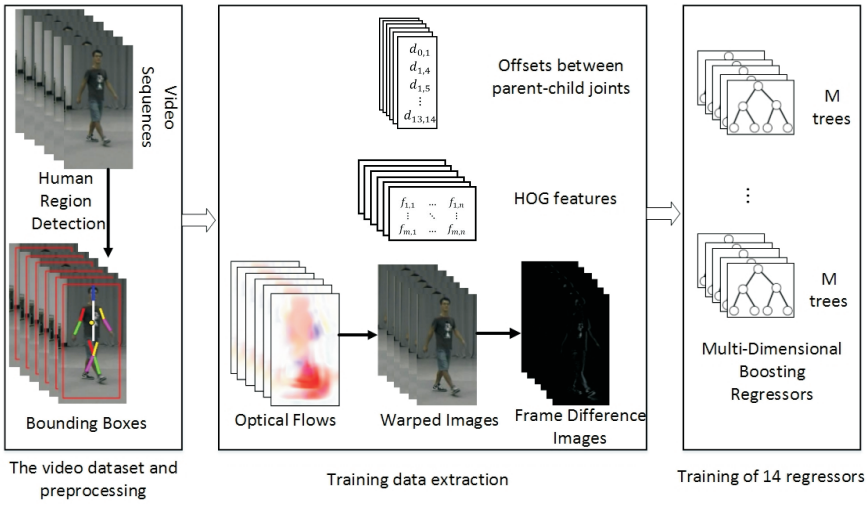

Fig. 2. The process of training. **Left**: The video dataset and preprocessing. In the preprocessing stage, we detect a rectangular region containing the human's image and annotate the full-body human poses. **Center**: Training data extraction. Here, $d_{i,j}$ represents the normalized offset vector between each parent-child joint pair based on the JRT in each video frame, and $f_{i,1} \cdots f_{i,n}$ represent the HOG feature vector for the i-th joint, computed from the subimage centered on its parent joint, in each video frame. The bottom part represents the optical flows computed, the images warped according to the optical flows, and the absolute frame difference images computed. More HOG features are further computed on the frame difference images. **Right**: Training of 14 regressors. For each edge in the JRT (See Fig. 1), we train a regressor that maps the local features around the parent joint to an offset between the parent and the child. The training of regressors is based on the extracted training data, *i.e.*, HOG features and offsets, as illustrated in **Center**.

3 Proposed Scheme

We assume as input to our scheme a monocular video sequence of a human. Our aim is to, for each video frame, estimate the full-body human pose that is represented as 2D positions of pre-defined human joints.

To the best of our knowledge, most of the researches based on regression for full-body human pose estimation are to learn a mapping function from the features computed from a local image region that contains an entire human body to a human pose. Those methods have a defect that the local image region should be large enough to contain the human body and thus may contain a large background region as well, increasing the complexity of human pose estimation. Therefore, we choose to extend the work by Hara and Chellappa [1] since it has achieved a good balance between accuracy and efficiency for estimating upper-body human poses in still images.

Fig. 3. Human regions detected on several video frames using the SVM and HOG approach.

At each video frame, we first detect a rectangular region containing the human's image, utilizing the SVM method and the HOG descriptor. Thereafter, we estimate the full-body human pose in this detected region. Specifically, we design a joints relationship tree (JRT) corresponding to the hierarchy of joints in a full human body (see Fig. 1). The root joint is always fixed at 1/2 width and 1/3 height of the human's bounding rectangle. For each of the other joints, its position relative to its parent is determined via regression on the features of a temporally and spatially local region around its parent. The hierarchy of regressors at all the non-root nodes are learned in a training stage (see Fig. 2). Details of the proposed scheme are given in the following subsections.

3.1 Human Region Detection

We utilize a linear SVM classifier on HOG features [19, 20] to detect the human region in each video frame. In the training stage, we randomly sample rectangular regions of $N \times M$ (we use 64×128 in experiments) in the video frames. If a sample contains a human or part of a human, we label it as positive; otherwise, we label it as negative. Next, we extract all the sample regions' HOG features, and train a linear SVM classifier on those HOG features. In the testing stage, We slide a rectangular window of $N \times M$ over each video frame, from top to bottom and from left to right. Each window is classified as positive or negative, utilizing the trained SVM classifier. The tight bounding rectangle of all the positive samples then give the detected human region in each frame. As examples, the results of human region detection for several video frames are illustrated in Fig. 3.

It is worth mentioning that Hara and Chellappa [1] do not conduct human region detection in their algorithm but directly use the annotated human region information in their test image datasets.

3.2 Joints Relationship Tree

We design a joints relationship tree for the full human body, which extends the dependency graph in reference [1]. The JRT we construct is illustrated in Fig. 1, where the left part shows the JRT structure and the right part marks the joints in an image corresponding to the nodes in the JRT. As shown in Fig. 1, there are totally 15 nodes in the JRT, which are numbered from 0 to 14 and correspond to a root location and 14 joints in a human body.

Denoting the image of the t-th frame as I_t, and the estimated position of the i-th joint in I_t as $J_{t,i}$. With each parent-child pair, (i,j), in the JRT, we associate a mapping function, $G_{i,j}(X_{t,i})$, which gives the normalized offset vector from $J_{t,i}$ to $J_{t,j}$ based on the spatially and temporally local feature vector, $X_{t,i}$, around $J_{t,i}$ in I_t. That is,

$$J_{t,j} = G_{i,j}(X_{t,i}) \cdot S_t + J_{t,i} \tag{1}$$

In Eq. 1, S_t is the normalizing factor used when training $G_{i,j}(\cdot)$. We define S_t as proportional to the width of the detected human region in I_t and $S_t = W_t/K$ ($K = 64$ in our method) where W_t is the width of the detected human region in I_t. The mapping function, $G_{i,j}(\cdot)$, is the regression function that we learn from manual annotations of the joint positions in the training video frames. How the feature vector, $X_{t,i}$, is computed and how the regression function, $G_{i,j}(\cdot)$, is learned are described in the following subsections.

3.3 Local Features

As described in Section 3.2, the regression between each parent-child joint pair is based on spatially and temporally local features around the parent. Therefore, for the i-th joint in the t-th frame, we need to compute a local feature vector, $X_{t,i}$, around $J_{t,i}$, as detailed below.

In order to characterize the spatially local features around $J_{t,i}$, we take a $K \times K$ ($K = 64$ in our method) appearance patch, $I_{t,i}$, centered on $J_{t,i}$ from I_t, and then compute the HOG feature vector [20], $H_{t,i}$, of $I_{t,i}$ as

$$H_{t,i} = \left[f_t(i,1), f_t(i,2), \cdots, f_t(i,n) \right] \tag{2}$$

where n is the dimensionality of the vector and $f_t(i,j)$ ($j = 1, \cdots, n$) is an item of the HOG feature descriptors.

As we known from previous researches, motion feature can enhance the performance of still image based pose estimation methods by utilizing the temporal correlation between temporally close frames. Motion features are particularly helpful in two cases that are often hard for still image based methods: 1) occluded body part, and 2) coloring/illumination similarity between a body part and its background. For these two cases, motions from previous frames will provide further hints on the affected joint's position in the problematic frame.

In order to characterize the temporally local features around $J_{t,i}$, we first compute Lucas-Kanade [21] optical flows between two frames. Denote the optical flow field from frame I_t to frame I_{t-n} as $U_{t,t-n}$ ($n < t$). We obtain the warped

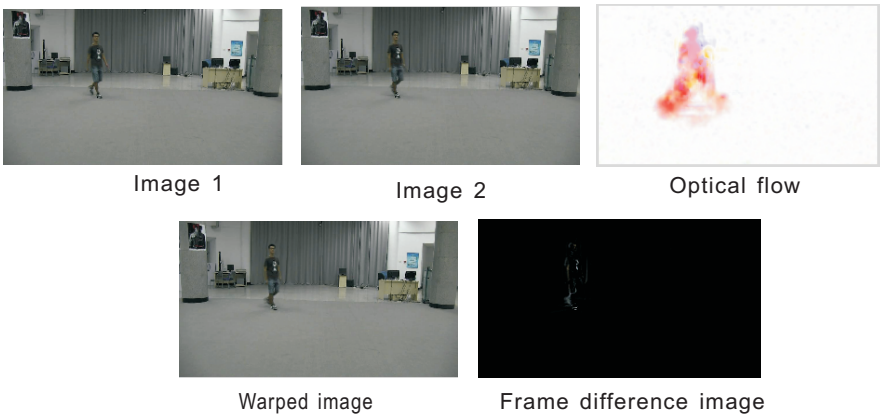

Fig. 4. The process of motion patch computation. The first row shows two adjacent video frames and their optical flow image obtained by the Lucas-Kanade optical flows algorithm; the second row shows the warped image from image 1 using the optical flow and the absolute frame difference image between image 2 and the warped image.

image I^t_{t-n} which is the frame I_{t-n} warped to the frame I_t by using bilinear interpolation with the flow field $U_{t,t-n}$. Then we compute an array, $M_{t,i}$, of motion patches as:

$$M_{t,i} = \begin{bmatrix} |I_{t,i} - I^{t,i}_{t-1,i}| \\ |I_{t,i} - I^{t,i}_{t-2,i}| \end{bmatrix} \tag{3}$$

where $I^{t,i}_{t-n,i}$ $(n = 1, 2)$ is the warped image patch from $I_{t-n,i}$ to $I_{t,i}$ by using the optical flow field $U_{t,t-n}$ from I_t to I_{t-n}. As an example, we illustrate in Fig. 4 the process of computing a motion patch from two adjacent video frames.

After the motion patch array, $M_{t,i}$, is obtained, we compute the HOG feature vectors, $H'_{t,i}$ and $H''_{t,i}$, from $M_{t,i}(1)$ and $M_{t,i}(2)$, respectively. Thereafter, we normalize $H_{t,i}$, $H'_{t,i}$ and $H''_{t,i}$. Still denoting the normalized HOG feature vectors as $H_{t,i}$, $H'_{t,i}$ and $H''_{t,i}$, we finally set $X_{t,i} = (H_{t,i}, H'_{t,i}, H''_{t,i})$. It is worth noting that $H_{t,i}$ is the spatial feature descriptor that has also been used by Hara and Chellappa [1], while $H'_{t,i}$ and $H''_{t,i}$ form the motion feature descriptor that we propose in this work.

In general, a frame closer to the t-th in time has a higher impact on the pose estimation accuracy for the t-th frame. Therefore, we make higher-resolution HOG descriptions for frames closer to the t-th, which is achieved by controlling cell and block sizes in the HOG computation. Specifically, we set the cell sizes to 8×8, 16×16 and 32×32 for the computation of $H_{t,i}$, $H'_{t,i}$ and $H''_{t,i}$, respectively, while setting the block size to 2×2 for all the three HOGs. In each cell, the number of orientation bins with signed gradients is set to 9. As a result, we get a 1,764-dimensional $H_{t,i}$, a 324-dimensional $H'_{t,i}$, a 36-dimensional $H''_{t,i}$, and a 2,124-dimensional $X_{t,i}$.

3.4 Multi-Dimensional Boosting Regression

We use a training set of manually annotated video sequences to learn the regression function between any parent-child pair in the JRT. All the human joints' 2D positions are marked on each frame of each training video sequence. We assume that all the training video sequences contain N frames in total, forming a training frame set, T.

Let us focus on learning the regression between one parent-child pair, (m, n), while the same process applies to all the other parent-child pairs as well. From each image, $I_i \in T$ ($1 \leq i \leq N$), we compute the local feature vector, X_i, around the m-th joint using the method described in Section 3.3, and compute the normalized offset between the two annotated joints positions, $J_{i,m}$ and $J_{i,n}$, as $Y_i = (J_{i,n} - J_{i,m})/S_i$. Now that we have a set of training samples $\{Y_i, X_i\}_{i=1}^N$, the learning is conducted via a standard multi-dimensional boosting regression process, as described in the following.

In general, given a set of training samples $\{Y_i, X_i\}_{i=1}^N$, where $Y \in R^v$ is the output vector and $X \in R^u$ is the input vector. The regression function can be theoretically sought by:

$$F^*(X) = \arg\min_{F(X)} \sum_{i=1}^{N} \omega_i \Psi(Y_i, F(X_i)) \tag{4}$$

where ω_i is the weight of the i-th training sample and $\Psi(\cdot)$ is the loss function.

In order to achieve the goal of Eq. 4, we may construct the strong regressor $F(X)$ as an ensemble of weak regressors $h(X; A_m, R_m)$:

$$F(X) = \sum_{m=0}^{M} h(X; A_m, R_m) \tag{5}$$

where $h(X; A_m, R_m) = \sum_{l=1}^{L} (A_{ml} \cdot 1_{R_{ml}}(X \in R_{ml}))$ is a regression tree with indicator function $1_{R_{ml}}(X \in R_{ml})$, vectors $A_m = \{A_{m1}, A_{m2}, \cdots, A_{mL}\}$ and input space partitioning $R_m = \{R_{m1}, R_{m2}, \cdots, R_{mL}\}$, where each A_{ml} is the average of the output vectors of the training samples that fall into space partition R_{ml}. In the training stage, the space partitioning is conducted iteratively. At each step, denoting the current space partitioning as $R_m = \{R_{m1}, R_{m2}, \cdots, R_{ml'}\}$ and the corresponding average output vectors as $A_m = \{A_{m1}, A_{m2}, \cdots, A_{ml'}\}$, we select one from the l' leaves with the largest sum of squared error E_{ml} for further partitioning, following the method in [1, 18]. E_{ml} is defined as:

$$E_{ml} = \sum_{X_i \in R_{ml}} \omega_i ||Y_i - A_{ml}||_2^2 \tag{6}$$

Further, we apply the Gradient TreeBoost algorithm [18] as follows:

$$F_m(X) = F_{m-1}(X) + \upsilon h(X; A_m, R_m) \quad (m \geq 1) \tag{7}$$

Algorithm 1 Multi-Dimensional Gradient Boosting Regression.

Input: A set of training samples $\{Y_i, X_i\}_{i=1}^N$
Output: The strong regressor $F_M(X)$
1: $F_0(X) = mean\{Y_i\}_{i=1,2,\cdots,N}$
2: **for** $m = 1$ to M **do**
3: $\tilde{Y}_i = Y_i - F_{m-1}(X_i), i = 1, \ldots, N$
4: $(A_m, R_m) = \arg\min_{A,R} \sum_{i=1}^N \omega_i ||\tilde{Y}_i - h(X_i; A, R)||_2^2$
5: $F_m(X) = F_{m-1}(X) + \upsilon h(X; A_m, R_m)$
6: **end for**
7: **return** $F_M(X)$

where υ $(0 < \upsilon < 1)$ is a shrinkage parameter and can control the learning rate, and $F_0(X) = mean\{Y_i\}_{i=1,2,\cdots,N}$. The parameters, (A_m, R_m), of a weak regressor is determined by:

$$(A_m, R_m) = \arg\min_{A,R} \sum_{i=1}^{N} \omega_i \Psi(Y_i, F_{m-1}(X_i) + \upsilon h(X_i; A, R)) \tag{8}$$

To summarize, the eventual regressor we construct is

$$F(X) = F_0(X) + \upsilon \sum_{m=1}^{M} h(X; A_m, R_m) \tag{9}$$

and the overall multi-dimensional boosting regression process is put in **Alg 1**.

4 Experimental Results

4.1 Dataset and Metric

Due to the unavailability of annotated monocular video database for full-body human pose estimation, we take video sequences with a DV camcorder to construct our own dataset. Each video sequence is taken of one person. The dataset contains 1,200 image frames in total, each sized at 960×540 pixels and annotated. In those videos the actors perform many different full-body actions such as walk, parade step, run, jump, one-hand wave, two-hand wave and so on. Some samples of the image frames in our dataset are shown in Fig. 5. To annotate a video frame, we run human region detection algorithm in it and manually mark the positions of the 14 joints (in accordance with the JRT structure) inside the detected human region. As illustrated in Fig. 1, a human body pose is then represented as 10 sticks each connecting two joints in the image, which are head, torso, upper and lower arms and upper and lower legs. Of all the 1,200 image frames, we apply the 5-fold cross validation and use 800 for training and 400 for testing.

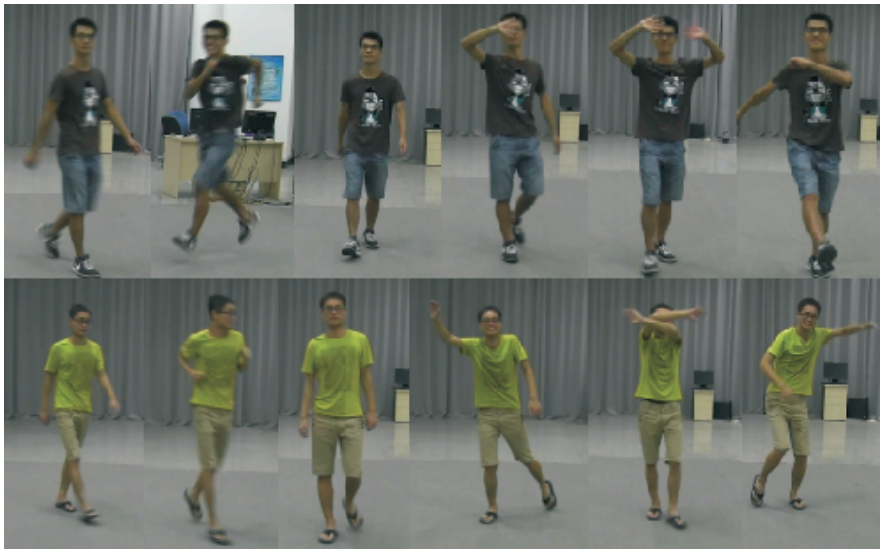

Fig. 5. Some samples of original image frames from videos in our dataset, corresponding to the motions of walk, parade step, run, jump, one-hand wave, and two-hand wave.

As the performance metric, we adopt the percentage of correctly estimated body parts(PCP) tool [1, 5, 22, 23]. With a PCP_t metric, it is considered correct if the estimated stick's endpoints lie within 100t% the length of the ground-truth stick from their ground-truth (annotated) locations.

4.2 Settings and Results

We need to train 14 regressors according to the JRT. In our experiments, for each boosting regression model, we set the number of trees as $M = 1000$, the number of leaves in each tree as $L = 5$ and the shrinkage parameter as $\nu = 0.1$.

In our experiments, we test four types of local feature vectors for the regression. In addition to the 2,124-dimensional HOG and optical-flow-temporal-difference(HOG-OFTD) features as introduced in Section 3.3, we also test other three feature vectors: one-scale-spatial (OSS), multi-scale-spatial (MSS) [1] and HOG-temporal-difference(HOG-TD) feature vectors. OSS computes the HOG of the local appearance patch with a cell size of 8×8 and a block size of 2×2, resulting in a 1,764-dimensional feature vector. MSS computes the HOG of the local appearance patch with a block size of 2×2 and cell sizes of 8×8, 16×16, 32×32, and concatenate the feature vectors for all these cell sizes to form a 2,124-dimensional feature vector. In order to test it for full body pose estimation, we run the original method in [1] on our proposed JRT structure but still use the MSS HOG features as used in [1]. HOG-TD computes the HOG of the local appearance patch with a cell size of 8×8 and a block size of 2×2, and the

Table 1. $PCP_{0.5}$ statistics for four types of features (OSS, MSS, HOG-TD and HOG-OFTD) on our dataset. Results are given for 10 human body parts: head, right and left upper arms and forearms, torso, right and left upper legs and lower legs. R and L stands for right and left, respectively; u.a and l.a standards for upper and lower arm, respectively; u.l and l.l standards for upper and lower leg, respectively.

	Head	R.u.a	R.l.a	L.u.a	L.l.a	Torso	R.u.l	R.l.l	L.u.l	L.l.l
OSS	98.18	78.73	46	79.64	34.54	100	96.18	76	92.36	76.36
MSS [1]	98.73	83.82	19.82	82.91	43.27	100	96.36	77.27	93.45	77.45
HOG-TD	**99.78**	**88**	21.78	83.78	36	100	**97.56**	77.11	98.22	88.22
HOG-OFTD	98.45	82.72	**58.32**	**86.11**	**53.44**	100	96.44	**80.22**	**98.75**	**90.74**

HOG of the motion patch, $M_{t,i} = |I_{t,i} - I_{t+1,i}|$, with a block size of 2×2 and cell size of 16×16, resulting in a 2,048-dimensional feature vector.

Using the four types of local feature vectors and $PCP_{0.5}$ as the performance metric, we obtain the statistics as given in Table 1 for head, upper arms and forearms, torso and upper and lower legs. From Table 1, we observe that HOG-OFTD yields the best performance on left upper arm, right and left forearms, left upper leg, right and left lower legs. Statistics in Table 1 demonstrates that, for most of the body parts, introducing the optical flow and frame difference as motion features leads to improved results over pure spatial features.

Further, we give in Table 2 the statistics about the average $PCP_{0.5}$ on our dataset. From Table 2, we see that HOG-OFTD leads to the best average estimation accuracy.

In the testing phase, the running time of regression on each tree of height h is O(h), since we follow a simple path down the regression tree. So the running time for each boosting regression model with M trees is O(Mh). We give in Table 3 the statistics about the average timing per frame excluding human detection on our dataset. We implemented our scheme in matlab language. Running our implementation on a desktop computer with an Intel Core(TM)i5 3.10GHz CPU and 4 GB memory.

Table 2. Average $PCP_{0.5}$ for four types of features (OSS, MSS, HOG-TD and HOG-OFTD) on our dataset.

Features	Average $PCP_{0.5}$
OSS	77.80
MSS [1]	77.31
HOG-TD	79.04
HOG-OFTD	**84.52**

Table 3. Time per image frame excluding human detection for two types of features (MSS, HOG-OFTD) on our dataset.

Features	Time/frame
MSS [1]	0.9sec.
HOG-OFTD	1.2sec.

Visual results of the full-body pose estimation using HOG-OFTD on a selected set of video frames are shown in Fig. 6.

From this figure, we see that the overall human poses are estimated with a good accuracy, though some failure cases exist locally. Those failure cases mainly happen in regions with self-occlusion and/or fast motion, for which insufficient information can be obtained and/or more randomness exists, adding to the difficulty of accurate estimation.

5 Conclusion and Future Work

In this paper, we propose a scheme to estimate 2D full-body human poses from monocular video sequences. At each frame, it detects the human region using an SVM and HOG human detection algorithm and then estimates the human pose in the detected region through multi-dimensional boosting regression. Specifically, we design a joint relationship tree reflecting the hierarchical structure of joints in a human body. In the training stage, we learn a regressor for each parent-child pair, which estimates the child joint's offset vector based on spatially and temporally local features around the parent; in the testing stage, we first fix the location of the root node relative to the detected human region, and then traverse the JRT in a depth-first order to estimate all the joints' positions utilizing the learned regressors. As experimentally demonstrated, the proposed scheme achieves outstanding estimation performance.

In the future, while further improving the estimation accuracy, we will increase the diversity of our datasets and seek to accelerate the computation for potential use in real-time applications.

Acknowledgement. This work is partially supported by Shandong Provincial Natural Science Foundation, China (Grant No. ZR2011FZ004), the National Natural Science Foundation of China (Grants No. 61472223, U1035004 and 61303083), the Scientific Research Foundation for the Excellent Middle-Aged and Youth Scientists of Shandong Province of China (Grant No. BS2011DX017) and the Program for New Century Excellent Talents in University (NCET) in China.

Fig. 6. Visual estimation results of our scheme with HOG-OFTD on selected frames.

References

1. Hara, K., Chellappa, R.: Computationally Efficient Regression on a Dependency Graph for Human Pose Estimation. Computer Vision and Pattern Recognition (2013) 3390–3397
2. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer vision and image understanding **104** (2006) 90–126
3. Poppe, R.: Vision-based human motion analysis: An overview. Computer vision and image understanding **108** (2007) 4–18
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision **61** (2005) 55–79
5. Eichner, M., Marin-Jimenez, M., Zisserman, A., Ferrari, V.: 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. International journal of computer vision **99** (2012) 190–214
6. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. Computer Vision and Pattern Recognition (2009) 1014–1021
7. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose priors for pictorial structures. Computer Vision and Pattern Recognition (2010) 422–429
8. Dantone, M., Gall, J., Leistner, C., Van Gool, L.: Human Pose Estimation using Body Parts Dependent Joint Regressors. Computer Vision and Pattern Recognition (2013) 3041–3048
9. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Strong Appearance and Expressive Spatial Models for Human Pose Estimation. The IEEE International Conference on Computer Vision (2013) 3487–3494
10. Zuffi, S., Freifeld, O., Black, M.J.: From pictorial structures to deformable structures. Computer Vision and Pattern Recognition (2012) 3546–3553

11. Zuffi, S., Romero, J., Schmid, C., Black, M.J.: Estimating Human Pose with Flowing Puppets. The IEEE International Conference on Computer Vision (2013) 3312–3319
12. Sapp, B., Toshev, A., Taskar, B.: Cascaded models for articulated pose estimation. Computer Vision–ECCV (2010) 406–420
13. Okada, R., Soatto, S.: Relevant feature selection for human pose estimation and localization in cluttered images. Computer Vision–ECCV (2008) 434–445
14. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient regression of general-activity human poses from depth images. The IEEE International Conference on Computer Vision (2011) 415–422
15. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. Computer Vision and Pattern Recognition (2011) 1297–1304
16. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.:, Real-time human pose recognition in parts from single depth images. Communications of the ACM **56** (2013) 116–124
17. Sun, M., Kohli, P., Shotton, J.: Conditional regression forests for human pose estimation. Computer Vision and Pattern Recognition (2012) 3394–3401
18. Bissacco, A., Yang, M.H., Soatto, S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. Computer Vision and Pattern Recognition (2007) 1–8
19. Pang, Y., Yuan, yuan an Li, X., Pan, J.: Efiicient HOG human detection. Signal Processing **91** (2011) 773–781
20. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition (2005) 886–893
21. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. IJCAI **81** (1981) 674–679
22. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. Computer Vision and Pattern Recognition (2008) 1–8
23. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. Computer Vision and Pattern Recognition (2011) 1385–1392