Fast Human Pose Estimation using Appearance and Motion via Multi-Dimensional Boosting Regression UCLA CSD-TR 050046

Alessandro Bissacco
UCLA Computer Science Department
4732 Boelter Hall
Los Angeles, CA 90095-1596
bissacco@cs.ucla.edu

Ming-Hsuan Yang
Honda Research Institute
800 California Street
Mountain View, CA 94041
mhyang@ieee.org

Stefano Soatto
UCLA Computer Science Department
4732 Boelter Hall
Los Angeles, CA 90095-1596
soatto@cs.ucla.edu

Abstract

We address the problem of estimating human pose in video sequences, where the rough location of the human has been detected. We exploit both appearance and motion information by defining suitable features of an image and its temporal neighbors, and learning a regression map to the parameters of a model of the human body using boosting techniques. Our work is intended to bridge the gap between efficient human body detectors that can estimate rough location but not pose in quasi-real time, and computationally expensive but accurate pose estimation algorithms based on dynamic programming. Our algorithm can be viewed as a fast initialization step for human body trackers, or as a tracker itself. In order to accomplish our task, we extend gradient boosting techniques to learn a multi-dimensional map from (rotated and scaled) Haar features to the entire set of joint angles representing the full body pose. Compared to prior work that advocated learning a separate regressor for each joint angle, our approach is more efficient (all joint angle estimators share the same features) and more robust (it exploits the high degree of correlation between joint angles for natural human pose).

1. Introduction

An important open problem in modern computer vision is full body tracking of humans in video sequences. In this work we focus in particular on estimating the 3D pose of a kinematic model of the human body from images. Such a task is extremely challenging for several reasons.

First there exist multiple plausible solutions to a query, since we are trying to recover 3D information from 2D images (this is especially true in the presence of partial occlusions). In order to disambiguate such cases, we can use prior knowledge on the most likely configurations, for example in a walking gait we expect the occluded arm to be parallel to the torso.

Second, humans are articulated objects with a fair amount of parts whose shape and appearance change due to various nuisance factors such as illumination, clothing, viewpoint and pose. This fact causes serious difficulties when, in the case of a discriminative approach (see [20] and references therein), we want to learn the map from images to poses or when using a generative approach (see [6] and references therein), we build a likelihood function as a matching score between a configuration hypothesis and a given image. Consequently, it is common to extract a feature representation which is most insensitive to the nuisance factors. For pose estimation, a frequent choice is to extract binary silhouettes, which can be computed from images using motion, a background model, or a combination of the two [1, 17, 9]. But using only silhouettes to model the appearance of people is seriously limiting, since important appearance information is discarded, which could help resolving ambiguous cases.

Finally, the space of admissible solutions, that is all possible positions and orientations of all body parts, is extremely large, and the search for the optimal configuration in this space is a combinatorial problem. To address this issue, most approaches proposed so far attempt to reduce the feasible space using both static and dynamic constraints. Static constraints restrict the search to the set of physically feasible body configurations. Dynamic constraints work by enforcing temporal continuity between adjacent frames, specified through a set of motions. A common approach is to learn a statistical model of the human dynamics and to use it in a sampling scheme [12], where given the body configuration in the current frame and the motion model we can compute a probability distribution which allows to make informed guesses on the limb positions in the next frame.

Although learned motion models have shown to greatly improve tracking performances for simple motions such as walking gaits, it is not clear how to efficiently combine different models in order to represent the ample variety of motions that can be performed by humans. Indeed, in the literature examples of effective tracking are limited to a small number of motions not too different from the training dataset. Moreover, each learned model represents a particular motion at a particular speed, so the system is unlikely to successfully track even an instance of the same motion if performed at a speed different from the one used for learning.

In general, there are always conditions where the tracker either provides an inaccurate estimate or loses track altogether. This is particularly true for fast motions, where the body limbs undergo large displacements from one frame to the next. Recent approaches which have shown considerable success for fast motions perform tracking by doing pose estimation independently at each frame [15]. Although we do not argue that this is necessarily the right approach to tracking, we believe in the importance of having an efficient pose estimator, which can take action whenever the tracking algorithm fails.

Therefore, the focus of this work is in building a fast body pose estimator for human tracking applications. Our pose estimator can be applied for:

- Automatically initializing a tracking module in the first frame and reinitializing it every time it loses track, or
- by running it at every frame, as a tracking algorithm.

The main distinction of our approach with respect to current state-of-the-art human pose estimators is that we aim to develop an algorithm which is fast enough to be run at every frame and used for real-time tracking applications. Unavoidably, to accomplish this we have to tradeoff estimation accuracy for execution speed.

Our work can also be seen as an important element towards the construction of an effective automatic body pose estimator system from video sequences. On one hand we have efficient body detectors [22] which can estimate presence and location of people in images. On the other hand we have accurate but computationally expensive dynamic programming approaches [6] which can find the optimal pose estimate of an articulated body model in a neighborhood of a proposed body configuration. Our method bridges the gap between these two approaches by taking an image patch putatively containing a human and computing an initial guess of her body pose, which can later be refined using one of the pose estimators available in the literature.

An important characteristic of our approach is that, in order to estimate the body pose, instead of restricting to binary silhouettes we exploit all information in the images, that is both appearance and motion. By doing so we can resolve some of the ambiguities that we would face if trying to directly map silhouettes to poses and which have led many researchers in this field to employ sophisticated mixture models [2, 20].

2. Related work

The two closely related problems of human pose estimation and tracking from images have attracted the interest of the computer vision community for decades. Here we will review the main approaches, emphasizing those directly relevant to our work. We refer the reader to the survey [8] for further investigations.

Estimating pose from a single image without any prior knowledge is an extremely challenging problem. It has been cast as deterministic optimization [6, 16], as inference over a generative model [11, 13, 10, 19], as segmentation and grouping of image regions [14], or as a sampling problem [11]. Proposed solutions either assume very restrictive appearance models [6] or make use of cues, such as skin color [23] and face position [13], which are not reliable and can be found only in specific classes of images (e.g. sport players).

A large body of work in pose estimation focus on the simpler problem of estimating the 3D pose from human body silhouettes [1, 17, 20, 9]. It is possible to learn a map from silhouettes to poses, either direct [1], one-to-many [17] or as a probabilistic mixture [2, 20]. However, as we mentioned in the introduction, silhouettes are inherently ambiguous as very different poses can generate similar silhouettes, so to obtain good results either we resort to complex mixture models [20] or restrict the set of poses [4], or use multiple views [9]. Shakhnarovich [18] demonstrates that combining appearance with silhouette information greatly improves the quality of the estimates. Assuming segmented images, they propose a fast hashing function that allows matching edge orientation histograms to a large set of synthetic examples. We use a similar basic representation of the body appearance, by masking out the image the background, and computing a set of oriented filters on the resulting patch.

Besides silhouettes and appearance, motion is another important cue that can be used for pose estimation and tracking [5, 24]. Most works assume a parametric model of the optical flow, which can be either designed [24] or learned from examples [5]. But complex motion models are not the only way to make use of motion information. As shown in [22], simple image differences can provide an effective cue for pedestrian detection. We follow this path, and integrate our representation of human body appearance with motion information from image differences.

Finally, recent work [15] advocates that tracking can be effectively performed by independently estimating pose at every frame. Our approach has a natural application in such a scenario, given that it can provide estimates in remarkably short order and, unlike [15], one does not need to learn an appearance model specific to a particular sequence.

3. Appearance and Motion Features for Pose Estimation

The input to our algorithm is a video sequence, together the bounding boxes of the human body for each frame as extracted by a human detector. We do not require continuity of the detector responses across frames, however our approach cannot provide an estimate for the frames in which the human body is not detected. Additionally, we require the binary silhouettes of the person, which can be extracted from the sequence using any background subtraction or segmentation scheme.

In this section we introduce our basic representation of people appearance and motion for the pose estimation problem. We use a set of differential filters tailored to the human body to extract essential temporal and spatial information from the images. We create a large pool of features, which later will be used in a boosting scheme to learn a direct map from image frames to 3D joint angles.

3.1. Motion and Appearance Patches

The starting point of our algorithm are patches containing the human body, extracted from the images frames using the bounding boxes provided by a human body detector. Patches are normalized in intensity value and scaled to a default resolution.

In order to improve learning speed and reduce the amount of training data required, we use the silhouette of the human body to mask out background pixels [1] . In figure 3 we can see some sample patches.

Motion information is represented using the absolute difference of image values between adjacent frames: $\Delta_i = \text{abs}(I_i - I_{i+1})$. As done before, from the image difference Δ_i we compute the motion patches by extracting the detected patch and masking out the background. We could use the direction of motion as in [22] by taking the difference of the first image with a shifted version of the second, but in order to limit the number of features considered in the training stage we opted for not using this additional source of information.

Normalized appearance I_i and motion Δ_i patches together form the vector x input to our regression function:

$$\mathbf{x_i} = \{I_i, \Delta_i\}$$

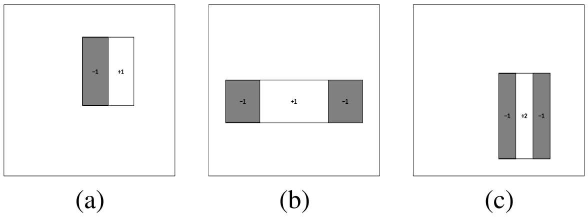
3.2. Features for Body Parts

Our human pose estimator is based on Haar-like features similar to the ones proposed by Viola and Jones in [22]. These filters measure the difference between rectangular areas in the image with any size, position and aspect ratio. They can be computed very efficiently from the integral image [22].

However, in the context of this work a straightforward application of these filters to appearance and motion patches is not doable for computational reasons.

For detection of either faces or pedestrians, a small patch of about 20 pixels per side is enough for discriminating the object from the background. But our goal is to extract full pose information, and if we were to use similar resolutions we would

[1] Given sufficient amount of data and training time, the boosting process would automatically select only the features whose support mostly lies in the foreground region.



(a) (b) (c)

Figure 1. Basic types of Haar features used in this work: edges (a), thick (b) and thin (c) lines. Each of these features can assume any position and scale within the estimation window (although for scale some restrictions apply, see text for details). Additionally, each feature can assume a set of 18 equally spaced orientations the range $[0, 180]$. It is intuitive to see how features (c) are suited to match body limbs, while features (a) and (b) can be used to match trunk, head and full body.

have limbs with area of only a few pixels. This would cause their appearance to be very sensitive to noise and would make it extremely difficult to estimate their pose. Consequently, we need to use a bigger patch. We chose the patch size by visual inspection, perceptually determining that a 64×64 patch contains enough information for pose estimation. Unfortunately augmenting the patch size greatly increases the number of basic features that fit in the patch (approximately squared in its area), therefore we need a strategy for selecting a good subset for training.

Another weakness of these basic features is that, by using vertical rectangles only, they are not suited to capture edges that are not parallel to the image axes. For pose estimation this is a shortcoming, since the goal is to localize limbs which can have arbitrary orientation. To allow for alignment of the filters to the body limbs, we extended the set of basic Haar features by introducing their rotated versions, computed at a few major orientations. In figure 1 we show some examples of the oriented Haar features. Notice that these filters are very similar to oriented rectangular templates commonly used for detecting limbs in pose detection approaches [6, 15]. Oriented features can be extracted very efficiently from integral images computed on rotated versions of the image patch. Notice that by introducing orientation in the features we further increase their number, so a good subset selection in the training process becomes crucial.

We experimented with various schemes for feature selection. Among the possible configurations, we found that one type of edge feature (fig. 1a) and two types of lines features (fig. 1b and 1c) are the best performers. Each feature can assume any of 18 equally spaced orientations in the range $[0, 180^o]$ (one every 10^o), and they can have any position inside the patch. To limit the number of candidates, we restrict each rectangle to have a maximum width of 12 pixels and minimum height of 6 pixels, thus making the filters roughly correspond to body limbs.

With this configuration, we obtain a pool of about 1 million filters for each of the motion and image patches. Since this number is still too high, we randomly select K of these features by uniform sampling. The result is a set of features:

$$\{f^k(\mathbf{x}_i)\}_{k=1,\cdots,K}$$

that map motion and appearance patches $\mathbf{x}_i = \{I_i, \Delta_i\}$ to real values.

4. Vector Gradient Boosting

In this section we introduce a novel approach for learning the regression map from motion and appearance features to 3D body pose.

We use the robust boosting approach to regression proposed in [7]. This algorithm is particularly suited to our problem since it provides an efficient way to automatically select from the large pool of filters the most informative ones to be used as basic elements for building the regression function.

Our contribution is to extend the gradient boosting technique [7] to multidimensional maps. Instead of learning a separate regressor for each joint angle, we learn a vector function from features to sets of joint angles representing full body poses.

The advantage of vector learning is that it allows the joint angle estimators to share the same set of features. This is beneficial because of the high degree of correlation between joint angles for natural human poses. The resulting pose estimator is sensibly faster than the collection of scalar counterparts, since it uses a number of features which grows with the effective dimension of the target space instead of with the number of joint angles. This has some similarities with the work of Torralba et al. [21], where detectors of a multiclass object classifier are trained jointly so that they share set of features.

In the following section we review the basic gradient boosting algorithm, next we derive its extension to multidimensional mappings.

4.1. Stochastic Gradient TreeBoost

Given a training set $\{y_i, \mathbf{x}_i\}_1^N$, with inputs $\mathbf{x} \in \mathbb{R}^m$ and outputs $y \in \mathbb{R}$, the goal of regression is to find a function $F^*(x)$ that maps $\mathbf{x}$ to y, such that the expected value of a loss function $\Psi(y, F(\mathbf{x}))$ is minimized:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y,\mathbf{x}} \Psi(y, F(x)) \tag{1}$$

We approximates $F^*(\mathbf{x})$ using an additive expansion:

$$F(\mathbf{x}) = \sum_{m=0}^{M} h(\mathbf{x}; \mathcal{A}_m, \mathcal{R}_m) \tag{2}$$

where the basic learners $h(\mathbf{x}; \mathcal{A})$ are assumed piecewise constant functions of $\mathbf{x}$ with values $\mathcal{A}_m = \{a_{1m}, \cdots, a_{Lm}\}$ and input space partition $\mathcal{R}_m = \{R_{1m}, \cdots, R_{Lm}\}$.

Gradient tree boosting [7] solves (1) with a two-step approach. At each step m it uses the previous estimate F_{m-1} to compute the "pseudo-residuals":

$$\tilde{y}_{im} = -\left[\frac{\partial \Psi(y_i, F(\mathbf{x_i}))}{\partial F(\mathbf{x_i})}\right]_{F(\mathbf{x}) = F_{m-1}(\mathbf{x})} \tag{3}$$

Then it constructs a regression tree which splits the $\mathbf{x}$ space into L-disjoint regions R_{lm} by least-squares fitting on the pseudo residuals $\tilde{y}_{im}$, and predicts a constant value a_{lm} in each region R_{lm}:

$$h(\mathbf{x}; \mathcal{A}_m, \mathcal{R}_m) = \sum_{l=1}^{L} a_{lm} 1(\mathbf{x} \in R_{lm}) \tag{4}$$

where $1(c)$ is 1 if condition c is true, 0 otherwise, and:

$$\mathcal{R}_m = \text{L-terminal node tree}(\{\tilde{y}_{im}, \mathbf{x_i}\}_{i=1,\cdots,N}) \tag{5}$$

The pseudo residuals $\tilde{y}_{im}$ and the tree predictions a_{lm} depend on the choice of the loss criterion Ψ. For the sake of robustness, we opted for least-absolute-deviation (LAD):

$$\Psi(y, F(\mathbf{x})) = |y - F(\mathbf{x})|.$$

We have:

$$\begin{aligned} \tilde{y}_{im} &= \text{sign}(y_i - F_{m-1}(\mathbf{x_i})) && (6) \\ a_{lm} &= \text{median}_{i:\mathbf{x_i} \in R_{lm}} \{y_i - F_{m-1}(\mathbf{x_i})\} && (7) \end{aligned}$$

The current approximation F_{m-1} is then updated by adding the regression tree scaled by a shrinkage parameter $0 < \nu < 1$ which controls the learning rate (smaller values lead to better generalization):

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \sum_{l=1}^{L} a_{lm} 1(\mathbf{x} \in R_{lm}) \tag{8}$$

In our setting, the regions are defined by thresholds θ on filter responses $f^k(\mathbf{x})$, where f^k is the k-th Haar filter computed on the appearance and motion patches $\mathbf{x}$. For the case of degenerate regression trees with a single node (decision stumps), we have:

$$h_s(\mathbf{x}; a_{1m}, a_{2m}, k_m, \theta_m) = \begin{cases} a_{1m} & \text{if} \quad f^{k_m}(\mathbf{x}) \le \theta_m \\ a_{2m} & \text{if} \quad f^{k_m}(\mathbf{x}) > \theta_m \end{cases} \tag{9}$$

An improvement both in terms of computation time and generalization performances can be easily obtained by training the regression tree on a subset of the training dataset, randomly drawn (without replacement) at each iteration. In particular,

Algorithm 1 Least-Absolute-Deviation Vector Gradient TreeBoost

```
1: $\mathbf{F}_0(\mathbf{x}) = \text{median}\{\mathbf{y_i}\}_{i=1,\cdots,N}$
2: for $m = 1$ to $M$ do
3:     $\{\pi(i)\}_1^N = \text{randperm}\{i\}_1^N$
4:     $\tilde{\mathbf{y}}_{\pi(i)m} = \text{sign}\left(\mathbf{y}_{\pi(i)} - \mathbf{F}_{m-1}(\mathbf{x}_{\pi(i)})\right) \quad , \quad i = 1, \cdots, \tilde{N}$
5:     $k_m, \theta_m = \arg\min_{k,\theta} \sum_{j=1}^{p} \min_{s_j \in \{-1,1\}} \sum_{i=1}^{\tilde{N}} \left|\tilde{y}_{\pi(i)m,j} - h_s(\mathbf{x}_{\pi(i)}; s_j, -s_j, k, \theta)\right|$
6:     $\mathbf{a}_{jm} = \text{median}\{\mathbf{y}_{\pi(i)} - \mathbf{F}_{m-1}(\mathbf{x}_{\pi(\mathbf{i})})\}_{(-1)^j\left(f^k(\mathbf{x}_{\pi(i)})-\theta\right)\geq 0 \ , \ i=1,\cdots,\tilde{N}} \quad , \quad j = 1, 2$
7:     $\mathbf{F}_m(\mathbf{x}) = \mathbf{F}_{m-1}(\mathbf{x}) + \nu \mathbf{h}_s\left(\mathbf{x}_{\pi(i)}; \mathbf{a}_{1m}, \mathbf{a}_{2m}, k_m, \theta_m\right)$
8: end for
```

at each step we compute $\{\pi(i)\}_1^N$, a random permutation of the integers $\{1, \cdots, N\}$. Then the base learner parameters, i.e. region parameters k_m, θ_m and regression values a_{lm} are computed from the $\tilde{N}$ subset $\{\mathbf{x}_{\pi(i)}, y_{\pi(i)}\}_1^{\hat{N}}$. The fraction $f = \frac{\tilde{N}}{N}$ determines the amount of randomness of the computation, with $f = 1$ giving deterministic boosting. Smaller values of f reduce the amount of computations but also reduce the amount of data for training the basic learner and therefore increase the variance of the estimates.

The resulting algorithm has good generalization performances and can be implemented very efficiently, since at each step for computing the best feature k_m and associated threshold θ_m, we use only the sign of the current residuals (6).

4.2. Vector Gradient TreeBoost

In this section we propose an extension to the Gradient TreeBoost algorithm in order to efficiently handle multidimensional maps.

Given a training set $\{\mathbf{y}_i, \mathbf{x}_i\}_1^N$ with vector inputs $x_i \in \mathbb{R}^n$ and outputs $\mathbf{y}_i \in \mathbb{R}^p$, our goal is to find the map $\mathbf{F}(x) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ minimizing the loss $\Psi(\mathbf{y}, \mathbf{F}(\mathbf{x}))$.

Vector Treeboost is derived by assuming that the map $\mathbf{F}(\mathbf{x})$ can be expressed as a sum of basic piecewise constant vector functions:

$$\begin{aligned}\mathbf{F}(\mathbf{x}) &= \sum_{m=0}^{M} \mathbf{h}(\mathbf{x}; \{\mathcal{A}_{m1}, \cdots, \mathcal{A}_{mp}\}, \mathcal{R}_m) = \\ &= \begin{bmatrix} h(\mathbf{x}; \mathcal{A}_{1m}, \mathcal{R}_m) \\ \cdots \\ h(\mathbf{x}; \mathcal{A}_{1p}, \mathcal{R}_m) \end{bmatrix} \end{aligned} \tag{10}$$

and by minimizing $E_{\mathbf{y},\mathbf{x}}\Psi\left(\mathbf{y}, \mathbf{F}(\mathbf{x})\right)$ using the Gradient Boosting scheme described in the previous section.

Notice that (10) differs from applying the expansion (2) to each element the vector map $\mathbf{F}(\mathbf{x})$ in that we restrict all the basic functions $h_i(\mathbf{x}) = h(\mathbf{x}; \mathcal{A}_i, \mathcal{R}_i)$ to share the same input space partition: $\mathcal{R}_i = \mathcal{R}$. For our application, this translates in requiring all the joint angle regressors to share the same set of features, thereby subtantially improving the efficiency of the representation.

Using Least-Absolute-Deviation as loss criterion and assuming decision stumps on the Haar features responses as basic functions, we obtain Algorithm 1. Here we give a brief outline of the main steps of the algorithm.

The approximation is initialized in line 1 with the median of the function. At line 3 a set of $\tilde{N}$ indices is drawn to select the training samples at current iteration. Line 4 computes the pseudo-residuals vectors as the sign of the current training residuals $\mathbf{y_i} - F(\mathbf{x}_i)$. Fitting to the pseudo-residuals elements $\tilde{y}_{i,j}$ then becomes a binary classification problem. Line 5 computes the regions (5) by finding the optimal feature and associated threshold value: for every feature f^k, we compute the classification error of a vector stump classifier $\mathbf{h}_s$ whose input are filter responses $f^k(\mathbf{x_i})$ and outputs are the pseudo-responses $\tilde{\mathbf{y}}_\mathbf{i}$, and pick the one with lowest error. Line 6 finds the two vector parameters $\mathbf{a}_1, \mathbf{a}_2$ of the basic stump learner $\mathbf{h}_s$, which are the constant predictions of the vector residuals in the two regions found in the previous step. Given that we use a L_1 norm as loss criterion, a_j are medians of the sample residuals. Line 7 adds the stump classifier $\mathbf{h}_s$ to the current vector function approximation $\mathbf{F_m}$, scaled by the learning rate ν.

As the name suggests, this algorithm is not limited to stumps but can be formulated for arbitrary decision trees. For the sake of clearness, we presented here only the simplest case. However, in our experiments we applied Classification and

Regression Trees(CART) [3] as basic functions $\mathbf{h}(\mathbf{x})$. These are decision trees modelling a piecewise constant function as in (4), where each node of the tree uses a feature f^k and a threshold θ to recursively split the current region of the input space in two, and the terminal leaves define the input space partition R_{lm}.

The most computationally expensive part of the algorithm is the selection of the optimal decision stump in line 4. However, since the pseudo residuals $\tilde{y}$ have only ± 1 value, if we precompute and sort the filter responses $f^k(\mathbf{x_i})$ we can find the solution by a simple counting procedure using only integer arithmetic. This allows us to use hundred of thousands of features per sample in the training stage.

5. Experiments

One of the major limitations for experimental validation of any approach to human pose estimation from images is the availability of labeled training data. Ground truth for human motion is difficult to collect, since it requires expensive motion capture systems that must be synchronized and calibrated with the imaging device. Moreover, motion capture data can only be collected in controlled indoor environments, often requiring the performer to wear a special suit and having her appearance altered by the motion capture sensors attached to her body.

Given these constraints, a thorough experimental validation of our approach is at this moment out of reach. Here we limit our tests to a feasibility study, using the only publicly available human motion sequence with 3D ground truth that we are aware of. The data is from the Brown Artificial Intelligence group, and has been used in [19]. It consists of 4 views of a person wearing motion capture setup walking in a circle , see 2 for sample frames. Even thogh motion capture information is available for a pair of cycles, image data is present for only one single gait loop, which makes it hard to divide it in training and testing subsets. We resolved this dilemma by randomly drawing 1000 frames to form the training dataset, while the remaining 900 frames will be employed for testing. Motion information for this data is represented as joint angles on a human body model, having 26 degrees of freedom angles and 12 parts.

The first step of our approach is to extract the human body patches and scale them to the default resolution of 64×64 pixels. We additionally mask out the background pixels from the patches using the binary silhouettes included in the dataset. Together with appearance, we extract the motion patches from frame differences, scaled and masked out as just described. Although eventually in real applications the patches will be provided by a human detector, we used the calibration information available in the datasets to draw the patches. Some sample output of this preprocessing stage are reported in figure 3.

First we scale each of the joint trajectories to unit L_1 norm :$\sum_{i=1}^{N} |y_{i,j}| = 1$. In this way each joint angle contributes equally to the cost function (1), however better visual results may be obtained by scaling the joint angles according to their contribution to the final image.

Given a set of motion and appearance patches $\mathbf{x}_i$ with associated normalized joint angles $\mathbf{y}_i$, we train the vector boosting regressor following Algorithm 1.

For each patch, we evaluated 200000 features, which is about the 20% of total number of the oriented filters that can be computed in a 64×64 window. At each iteration we used half of the training samples, $\tilde{N} = \frac{N}{2}$ to find the optimal parameters of the basic functions $\mathbf{h}$.

We experimented with different basic functions, and obtained best results using 5-node Classification and Regression Trees. We believe that decision stumps do not perform as well for this kind of problem because body part configurations for articulated objects such as humans are highly dependent, and an approximation of the pose map (10) as a sum of functions of single features cannot capture these dependencies. On the other hand, CART trees of n nodes can model functions having arbitrary interactions between $n - 1$ variables.

We set the learning rate $\nu = 0.05$, and run the boosting process until the improvement of the training residual is below a predefined threshold. Here we show results for a final regressors with 1035 basic functions.

Besides the computation speeed (on the order of milliseconds), notice that each basic functions has an economical representation, which consists in the descriptions of the Haar features (orientation and location of the rectangles), thresholds and output values. Compare this with approaches based on exemplar matching or kernel machines, which often need to retain a large part of the training examples.

In figure 3 we show some sample motion and appearance patches together with estimated pose, represented as the outline of a cilinder-based human model superimposed onto the original images. From these results it is clear how the lack of prior information adversarily affects the estimations of occluded parts.

In figure 4 we show the mean and standard deviation of absolute value of the error on the joint angle estimates for the test dataset.

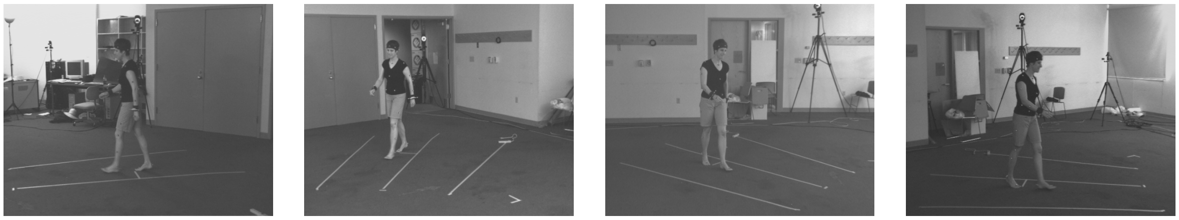


Figure 2. Sample frames from the dataset. For each pose, we have views from 4 different cameras.

6. Conclusions

In this work we proposed a novel approach to estimate 3D human poses from images. We derived an efficient algorithm which can run in real time and extract fairly accurate pose estimates from image patches containing humans. We introduced a set of oriented Haar features to extract low-level motion and appearance information from images. We proposed a multidimensional boosting regression algorithm which can handle efficiently the high dimensionality of the output space. We provided some preliminary experimental results showing the efficacy of our approach.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. *Proc. CVPR*, vol 2:pp. 882–888, 2004. 1, 2

[2] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. *CVPR*, 2005. 2, 3

[3] L. Brieman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, 1984. 7

[4] A. Elgammal and C. S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. *Proc. CVPR*, 2004. 3

[5] R. Fablet and M. J. Black. Automatic detection and tracking of human motion with a view-based representation. *Proc. ECCV*, 2002. 3

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. *Proc. CVPR*, vol 2:pp. 2066–2074, 2000. 1, 2, 4

[7] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38:367–378, 2002. 4, 5

[8] D. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, vol 73, no 1:pp 82–98, 1999. 2

[9] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. *Proc. ICCV*, 2003. 1, 2, 3

[10] G. Hua, M.-H. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. *Proc. CVPR*, vol 2:pp. 747–754, 2005. 2

[11] S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, vol 43, no. 1:pp. 45–68, 2001. 2

[12] M. Isard and A. Blake. Condensation - condiitional density propagation for visual tracking. *International Journal of Computer Vision*, 1:5–28, 1998. 2

[13] M. W. Lee and I. Cohen. Proposal driven MCMC for estimating human body pose in static images. *Proc. CVPR*, 2004. 2

[14] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. *Proc. CVPR*, vol 2:pp. 326–333, 2004. 2

[15] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. *Proc. CVPR*, 2005. 2, 3, 4

[16] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. *Proc. ECCV*, vol 4:pp. 700–714, 2002. 2

[17] R. Rosales and S. Sclaroff. Learning body pose via specialized maps. *In NIPS*, 2002. 1, 2

[18] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. *Proc. ICCV*, vol 2:pp. 750–757, 2003. 3

[19] L. Sigal, M. Isard, B. H. Sigelman, and M. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. *In NIPS*, pages 1539–1546, 2003. 2, 7

[20] C. Smichisescu. Learning to reconstruct 3d human motion from bayesian mixture of experts. *Proc. CVPR*, 2005. 1, 2, 3

[21] A. Torralba, K. P. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. *Proc. CVPR*, 2004. 4

[22] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Proc. ICCV*, pages pp. 734–741, 2003. 2, 3

[23] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997. 2

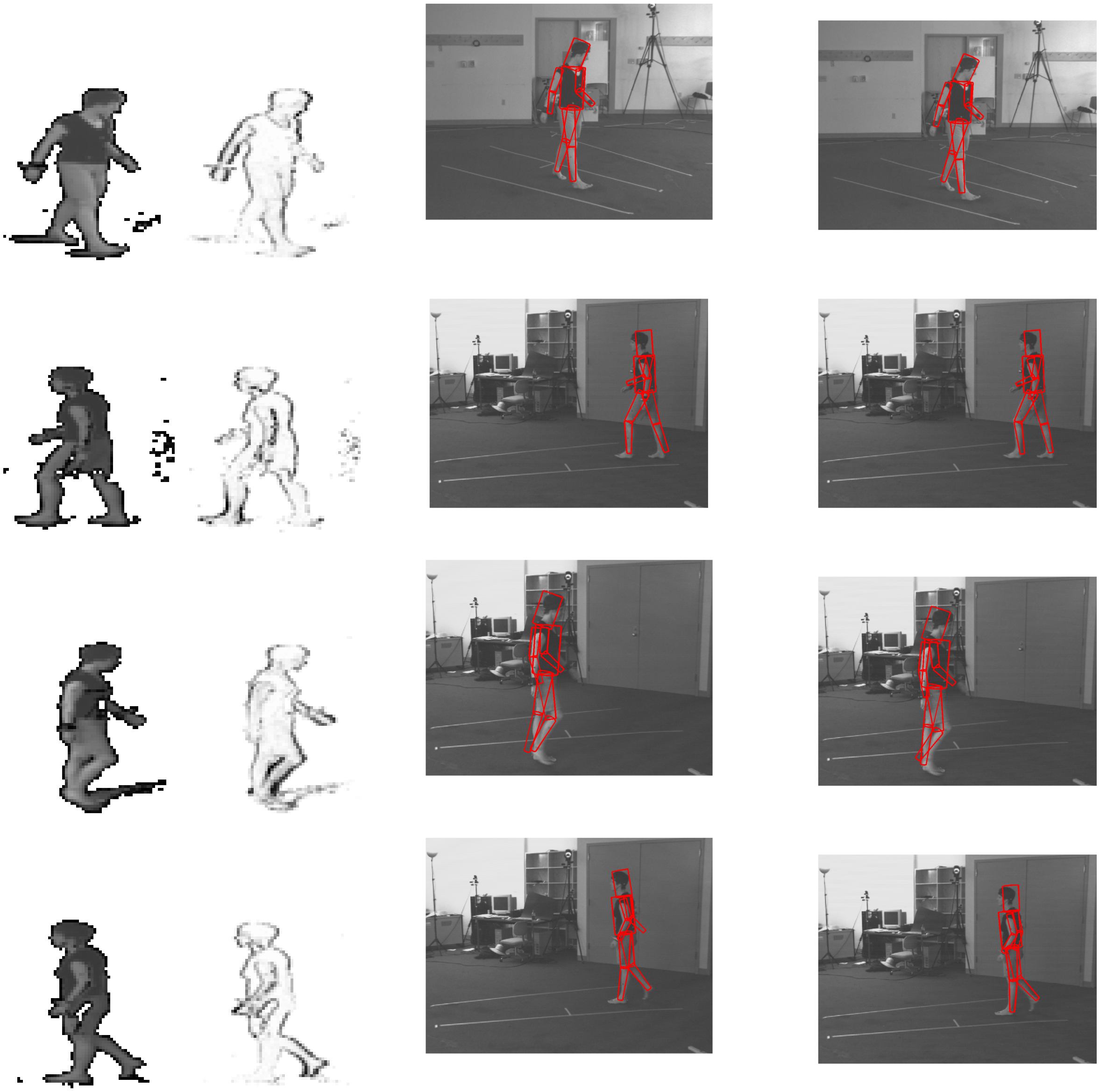


Figure 3. Sample estimation results. The first column shows appearance and motion patches extracted from the image frames. Second column displays the estimated pose, while the third column shows the provided ground truth. As we can see in the last row, we can have large estimation error when there are occlusions.

[24] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 1999. 3

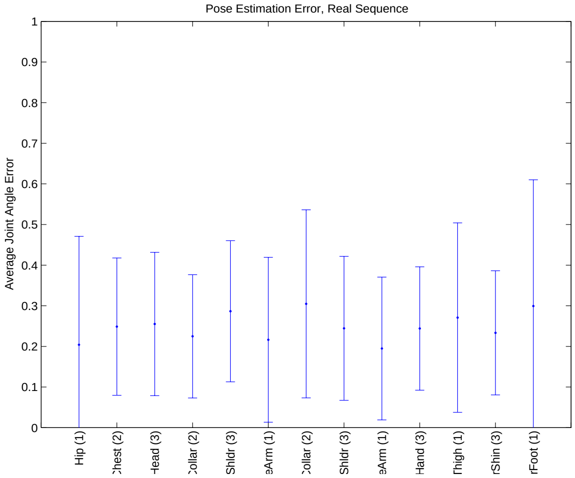


Figure 4. Joint angle estimation errors, mean values and standard deviations for each limb (in parenthesis the number of degrees of freedom of the limb). From the plot we see how most of the errors concentrates on the limb, since they are more prone to occlusions.