

# Combined Top-Down/Bottom-Up Human Articulated Pose Estimation Using AdaBoost Learning

Sheng Wang<sup>1</sup>, Haizhou Ai<sup>1</sup>, Takayoshi Yamashita<sup>2</sup>, Shihong Lao<sup>2</sup>

<sup>1</sup>Computer Science and Technology Department, Tsinghua University, Beijing, 100084, China

<sup>2</sup>Core Technology Center, Omron Corporation

E-mail:ahz@mail.tsinghua.edu.cn

## Abstract

*In this paper, a novel human articulated pose estimation method based on AdaBoost algorithm is presented. The human articulated pose is estimated by locating major human joint positions. We learn the classifiers on a normalized image for classifying each pixel position into a certain category. Two different kinds of classifiers, bottom-up joint position classifier and top-down skeleton classifier, are combined to achieve final results. HOG (Histogram of Oriented Gradient) feature is used for training both classifiers. Our human pose estimation system consists of three models, human detection, view classification, and pose estimation. The implemented system can automatically estimate human pose of different views. Experiment results are reported to show our proposed method can work on relatively small-size human images without using human silhouettes as a prerequisite, which is very efficient, robust and accurate enough for potential applications in visual surveillance.*

## 1. Introduction

Human articulated pose estimation from monocular images is an essential, yet challenging problem in computer vision. It has a vast area of potential applications, such as visual surveillance, human motion recognition, and human-computer interfaces.

There are many different approaches to tackling this problem. Regression based methods [1, 2] usually reduce the dimension to obtain a compact representation of the image and pose space, and find the mapping between them. Example based methods [6, 7] estimate the pose according to the pre-stored exemplars by matching the input image with them. Model based methods [4] use human articulated kinematics models to locate each

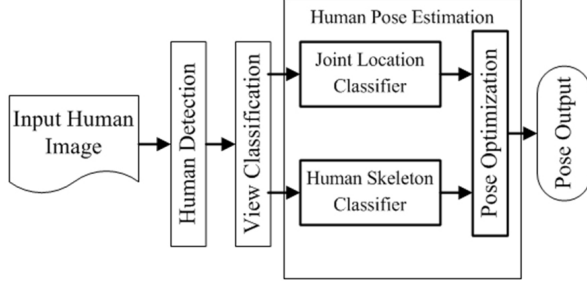
human part in which the kinematics models provide the basic restriction for human shape. Many of the pose estimation methods greatly depend on a clear silhouette of the human shape, which is hard to obtain from a single image.

In this paper, we develop a novel articulated pose estimation method based on AdaBoost algorithm to automatically find out the human joint positions without using silhouette as prerequisite. On the one hand we use bottom-up classification by locating each joint position separately, which capture the local feature of a human joint. On the other hand, we use top-down classification by learning a skeleton model as a whole, which captures the global feature of a human shape. By maximizing the objective function that incorporates the confidence value of both classifiers, we optimize the final pose estimation result. For training, we use HOG feature [3], because it captures the gradient information of an image. Our pose estimation method focuses on cropped human images with low resolution and noisy background, which is efficient, robust and accurate enough for potential applications in visual surveillance.

The rest of the paper is organized as follows: section 2 presents an overview of our pose estimation system; section 3 describes the design of the classifiers including the bottom-up joint location classifier and top-down skeleton classifier, and how we optimize the final pose estimation by combining the bottom-up and top-down methods. In section 4 we present the experiment results of our human pose estimation system. Our conclusion and future work are given in the last section.

## 2. System Overview

The proposed human pose estimation system consists of three modules, human detection, view classification, and pose estimation, as shown in Figure 1. We use a human detector [5] to locate humans, then we use



**Figure 1.** Human pose estimation system.

AdaBoosted classifiers to divide humans into three view classes: front/rear, left side and right side. The pose estimation is view specific which is trained separately because human pose varies dramatically under different views.

### 3. Articulated Pose Estimation Framework

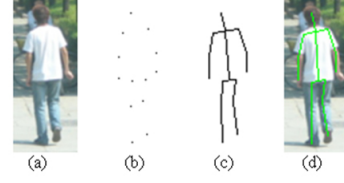
We use the 13-joint model as shown in Figure 2, including major human joints such as: head, shoulders, elbows, wrists, hips, knees, and ankles. For small human size image, 13 joints seem to be sufficient to render human pose. Connecting these joints we can get a clear human pose skeleton.

Inspired by [8], we treat the articulated pose estimation as a classification problem for each pixel into a certain category, that is, classify each pixel into a specific joint of the above 13 joints or non-joints, or into skeleton or non-skeleton. We use the following method to learn two kinds of classifiers: 3.1 Joint location classifiers; 3.2 Human skeleton classifiers.

Given a normalized image sample and a pixel position, the algorithm returns the confidence value of how likely this pixel belongs to a certain category. The image is denoted as an array  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  of grey value, and  $\mathbf{m} = (m_1, m_2, \dots, m_n)$  is its corresponding binary labeling with  $m_i \in \{-1, 1\}$ . If the position  $i$  belongs to the category then  $m_i = 1$ , otherwise  $m_i = -1$ .

The weak classifier is a function from the space  $\mathbf{X} \times \mathbf{U}$  to a real valued classification confidence space, where  $\mathbf{X}$  is the image space and  $\mathbf{U}$  is the coordinate space. Given a labeled sample set  $\mathbf{S} = (\mathbf{x}_1, \mathbf{m}_1), (\mathbf{x}_2, \mathbf{m}_2), \dots, (\mathbf{x}_N, \mathbf{m}_N)$  where  $\mathbf{x}_i \in \mathbf{X}$  is the pixel value vector of an image,  $\mathbf{m}_i$  is the corresponding labeling ground-truth that has the same dimension as  $\mathbf{x}_i$ . And all the training samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  have the same dimension.

In AdaBoost framework, one weak classifier  $h$  is built for each simple feature. A simple feature can be seen as a function  $f$  from the image space  $\mathbf{X}$  to real value ranged from 0 to 1,  $f : \mathbf{X} \mapsto [0, 1]$ . For a given



**Figure 2.** (a) A cropped human image. (b) The joint location of the human body in (a). (c) The human skeleton model by connecting the joints in (b). (d) The human skeleton rendered in the original image.

image  $\mathbf{x}$ , the feature value of  $\mathbf{x}$  is denoted as  $f(\mathbf{x})$ . We assign a weight to each pixel in each training image  $\mathbf{x}$ , denoted as  $D(\mathbf{x}, u)$ , where  $u$  is the position (coordinate) of the pixel.

The weak classifier  $h$  is learned as a piecewise function partition the domain of  $f$  into  $n$  disjoint blocks:  $block_j = [\frac{j-1}{n}, \frac{j}{n}]$ ,  $j = 1, 2, \dots, n$

$$h(\mathbf{x}, u) = \frac{1}{2} \ln\left(\frac{W_+^j(u) + \varepsilon}{W_-^j(u) - \varepsilon}\right), \text{ if } f(\mathbf{x}) \in block_j \quad (1)$$

where  $W_{\pm}^j(u)$  is the sum of the weight at position  $u$ :

$$\begin{aligned} W_{\pm}^j(u) &= P(f(\mathbf{x}) \in block_j, m(u) = \pm 1) \\ &= \sum_{f(\mathbf{x}) \in block_j, m(u) = \pm 1} D(\mathbf{x}, u) \end{aligned} \quad (2)$$

Different from most of classification problems using AdaBoost, we do not set a threshold to classify the pixel into a certain label; instead we get the confidence value of the pixel. We use the AdaBoost algorithm to choose weak classifiers to build one strong classifier that returns the confidence value. Assume  $h_t(\mathbf{x}, u)$  is the  $t$ -th weak classifier, and  $H(\mathbf{x}, u)$  is the strong classifier built by weak classifier  $h_t(\mathbf{x}, u)$ :

$$H(\mathbf{x}, u) = \sum_{t=0}^T h_t(\mathbf{x}, u) \quad (3)$$

The features used in AdaBoost training are HOG features because the HOG feature has proved effective in pedestrian detection [3] and pose estimation [7]. The feature makes statistics about magnitude of gradient in several orientations. To some extent it is better than a human silhouette, because it takes the appearance into consideration.

#### 3.1. Joint location classifiers

For joint location modeling, we build the classifier  $H_j^k(\mathbf{x}, u)$ , which is the likelihood of the position  $u$  in the image  $\mathbf{x}$  belonging to the  $k$ -th joint. For the joint location  $k$ , the training set is  $\mathbf{S} = (\mathbf{x}_1, \mathbf{m}_{1k}), (\mathbf{x}_2, \mathbf{m}_{2k}), \dots, (\mathbf{x}_N, \mathbf{m}_{Nk})$ , where  $\mathbf{x}_i$  is an

$n$ -dimensional vector of the image ( $n$  is the training image size),  $m_{ik}$  is the binary ground-truth for the  $k$ -th joint of image  $x_i$ ,  $m_{ik}$  is also an  $n$ -dimensional vector. For a pixel position  $u$ ,  $m_{ik}(u) = 1$  if and only if the pixel position  $u$  of the image  $x_i$  is the  $k$ -th joint location.

We train each joint location separately and do not consider the relationship between the joints at this stage. So for each joint  $k$  we train a strong classifier  $H_j^k(x, u)$ , which is a bottom-up learning that only captures the local feature of one joint.

For feature selection, the feature pool contains HOG feature in the whole image. In that way, although the joint location classifiers are trained independently, different human parts do have mutual influence. For instance, a HOG feature around the right leg might affect the position of the left arm. So actually the joint location classifiers alone can be used to estimate human pose simply by finding out the most likely pixel of each human joint.

### 3.2. Human skeleton classifiers

For human skeleton modeling, we build a classifier  $H_S(x, u)$ , which is the likelihood of the position  $u$  in the image  $x$  belonging to the human skeleton. The training set is  $\mathcal{S} = (x_1, m'_1), (x_2, m'_2), \dots, (x_N, m'_N)$ , where  $x_i$  is an  $n$ -dimensional vector of the image the same as those in the joint location training,  $m'_i$  is the binary ground-truth for the corresponding skeleton (see Figure 2c) linked by the joint location of  $x_i$ , and  $m'_i$  is also a  $n$ -dimensional vector. For the pixel position  $u$ ,  $m'_i(u) = 1$  if and only if the pixel position  $u$  of the image  $x_i$  is on the human skeleton.

The human skeleton model captures the relationship between joints. Therefore it can be used to guide the joint location. We consider it as a top-down model which captures the global information of the human body.

### 3.3. Pose optimization

To estimate the human pose, we locate the positions of major human joints  $\mathbf{u} = (u_1, u_2, \dots, u_m)$  by combining the top-down skeleton model and the bottom-up joint location model.

We find the joint positions which maximize the following objective function:

$$Z(\mathbf{x}, \mathbf{u}) = \left( \frac{\alpha}{m} \sum_{k=1}^m H_j^k(\mathbf{x}, u_k) \right) + \left( \frac{1-\alpha}{M} \sum_{i=1}^M H_S(\mathbf{x}, y_i) \right) \quad (4)$$

where  $\mathbf{x}$  is the gray value vector of the image,  $\mathbf{u} = (u_1, u_2, \dots, u_m)$  is the joint position,  $m$  is the number of

joints in the joint model ( $m = 13$ ), and  $u_k$  is the  $k$ -th joint coordinate.  $H_j^k(\mathbf{x}, u_k)$  is the confidence value of the  $k$ -th joint position  $u_k$ . For joint position  $\mathbf{u}$ , we connect these  $m$  joints which form a skeleton. And  $y_i$  is the  $i$ -th pixel coordinate of the skeleton, while  $M$  is the number of pixels in the skeleton.  $H_S(\mathbf{x}, y_i)$  is the confidence value of the skeleton at position  $y_i$ .  $\alpha$  is a real value,  $\alpha \in [0, 1]$ . It shows the relative importance of joint position model.

Given an image  $\mathbf{x}$  and a possible joint position  $\mathbf{u}$ , we can calculate the objective function  $Z(\mathbf{x}, \mathbf{u})$ . In practice, enumerating the objective function for all the possible joint positions is time consuming. Thus, we divide the process into two steps using greedy algorithm. First, for each joint position  $i$ , we select a few pixel positions  $u_k$  with largest  $H_j^k(\mathbf{x}, u_k)$  values as candidates. Next, for each limb (two arms and two legs), we compare the objective function of the candidates and acquire the optimized joint positions. The combinations of all the limbs can render the human pose quite accurately.

## 4. Experiments

We collected our training samples from a pedestrian sample set. These samples are resized to  $24 \times 58$  pixels. We hand-labeled the 13 joints of human body, and connected these joints to get the skeleton of that human shape as illustrated in Figure 2. All together 1000 front/rear and 1000 left side view pedestrian images are labeled as training samples for both the joint location classifiers and the human skeleton classifiers. We train front/rear and left side view classifiers separately, and use the left side view classifiers to estimate the right side view human pose by flipping the image. In addition, three view classifiers are trained to determine the human view.

Our program is coded in C++ using OpenCV functions on a 3G Hz 32-bit Pentium PC. We train the classifiers until the accuracy rate for all the pixels reaches over 99.9% on training set. Our human pose estimation appears to be very efficient, it takes about 120 ms to estimate the human pose.

### 4.1. Normalized testing images

We first evaluate our method on normalized images of front/rear and side view pedestrians.

Initially, we used the joint location classifiers alone to estimate the human pose, that is, select each joint position that with the highest confidence value. Then we incorporate the top-down skeleton model into the system by finding the joint location which maximizes the



**Figure 3.** Improvement after incorporating the skeleton model, the upper row is without skeleton model, the lower row is the result of the combined method



**Figure 4.** Results on normalized side view images

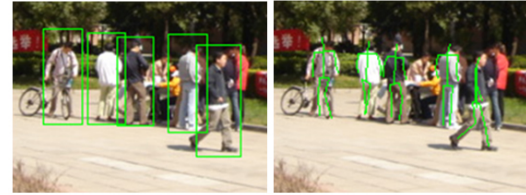
objective function in (4). The results of the improvement are shown in Figure 3. The upper row is the results of using the joint location classifiers alone while the lower row shows those of the combined method. For most of the testing images, by applying the joint location classifiers alone we can already achieve fine results. Nevertheless, the accuracy of pose estimation can be further improved after incorporating the skeleton model, since the skeleton model can capture the global shape of the human pose.

We also assess our method on right profile images, which are much more challenging than front/rear images due to the limbs occlusion. As show in Figure 4 the upper body is more difficult to estimate, for the reason that, upper body can easily get occluded in side view profiles and the ground truth labeled manually in training set may not be precise due to the occlusion.

#### 4.2. Pose estimation system

The results of our pose estimation system are shown in Figure 5. We use the bounding box (see Figure 5a) given by human detector as the initialization for articulated pose estimation. For each detected human pedestrian, we resize the image inside the bounding box to  $24 \times 58$  pixels. After determined the view of human body we estimate the pose. The results of the final pose estimation after detection are shown in Figure 5b. As we can see, our system can detect and estimate multi-view human poses, and can achieve quite good result, especially for small size human bodies in surveillance.

Our pose estimation system is position sensitive, which means the bounding box given by detection may affect the pose estimation. The deviation of the bounding box usually causes the joint location to be a few pixels away from its real position.



**Figure 5.** Human pose estimation: (a) Human detection (b) Pose estimation

## 5. Conclusion

In this paper, we have introduced a novel learning-based method for articulated human pose estimation. We use AdaBoost algorithm to learn the confidence value of the joint location of a human body. No human segmentation is required, and our method can work on small human size pictures with unclear background. We use the top-down skeleton model to guide the bottom-up joint localization, and obtain the skeleton of a human body. Our system shows promising results of multi-view pose estimation after human detection which is very efficient, robust and accurate enough for potential applications in visual surveillance.

## 6. Acknowledgment

This work is supported by Beijing Educational Committee Program (YB20081000303), and it is also supported by a grant from OMRON Corporation.

## References

- [1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *PAMI*, 28(1):44–56, 2006.
- [2] A. Bissacco, Y. Ming-Hsuan, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression, 2007. *CVPR*.
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection, 2005. *CVPR*.
- [4] P. F. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.
- [5] W. Gao, H. Ai, and S. Lao. Adaptive contour features in oriented granular space for human detection and segmentation, 2009. *CVPR*.
- [6] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *PAMI*, 28(7):1052–1062, 2006.
- [7] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. Torr. Randomized trees for human pose detection, 2008. *CVPR*.
- [8] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier, 2007. *ICCV*.