Boosted Deformable Model for Human Body Alignment [*]

Xiaoming Liu Ting Yu Thomas Sebastian Peter Tu
Visualization and Computer Vision Lab
GE Global Research, Niskayuna, NY 12309
{liux,yut,sebastia,tu}@research.ge.com

Abstract

This paper studies image alignment, the problem of learning a shape and appearance model from labeled data and efficiently fitting the model to a non-rigid object with large variations. Given a set of images with manually labeled landmarks, our model representation consists of a shape component represented by a Point Distribution Model and an appearance component represented by a collection of local features, trained discriminatively as a two-class classifier using boosting. Images with ground truth landmarks are the positive training samples while those with perturbed landmarks are considered as negatives. Enabled by piece-wise affine warping, corresponding local feature positions across all training samples form a hypothesis space for boosting. Image alignment is performed by maximizing the boosted classifier score, which is our distance measure, through iteratively mapping the feature positions to the image, and computing the gradient direction of the score with respect to the shape parameter. We apply this approach to human body alignment from surveillance-type images. We conduct experiments on the MIT pedestrian database where the body size is approximately 110×46 *pixels, and demonstrate our real-time alignment capability.*

1. Introduction

This paper presents a novel approach to image alignment, a key component for many vision tasks such as object tracking, mosaicing, registration, etc. Image alignment is defined as the process of moving and deforming a *template* so that its *distance* to the underlying image is minimized. This is a well-studied problem, and there are approaches for tackling it in a variety settings [2, 3, 17, 18].

Approaches to image alignment can be categorized based on the *direction* of warping, *i.e.*, whether the template is warped to the image or vice versa. In Active Appearance Model (AAM) [3, 18], image observations are warped to the mean shape over which the residual is computed and this drives the fitting. In contrast, for Active Shape Model (ASM) [2] fitting is performed by warping the mean shape and image evidence at the warped landmarks. The latter approach is more efficient, and object edge information can be explicitly utilized in the alignment.

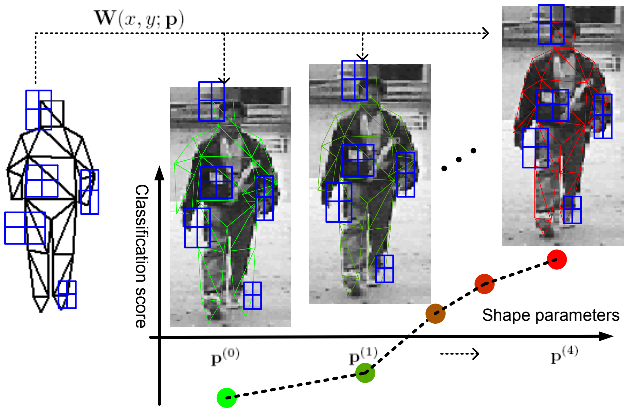

Figure 1. **Boosted Deformable Model.** An landmark-based eigenspace is used as the shape model, and a set of local features is discriminatively trained to distinguish correct v.s. incorrect alignment. During the fitting, the (x, y) location of each feature is mapped to the observation space based on the current shape parameter $\mathbf{P}$. The responses of the local features on the observation jointly determine the updates for $\mathbf{P}$, until convergence is achieved.

In terms of appearance modeling, AAM employs a simple eigenspace model, whereas ASM learns the local appearance for each landmark. On the other hand, the recently proposed Boosted Appearance Model (BAM) [14] learns the local features that have globally optimal discrimination properties. BAM has been shown to have superior performance than AAM for face alignment. However, BAM is expensive as it must warp the image to the mean shape.

This paper presents a novel image alignment framework called Boosted Deformable Model (BDM), summarized in Figure 1, which combines the advantages of both approaches. We illustrate the usefulness of BDM in address-

[*] This work was supported by award #2006-IJ-CX-K045 awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

ing the human body alignment problem. The *template* representation of BDM is computed using a set of images with manually labeled landmarks. It consists of a shape component represented by a Point Distribution Model (PDM) and an appearance component modeled by a collection of Histograms of Oriented Gradient (HOG) features. These HOG features are learnt discriminatively using a two-class classifier via boosting. We align the HOG feature locations across all training samples using a piece-wise affine warping between body shapes, and this forms a hypothesis space from which boosting chooses features. Unlike ASM where local measures are used, BDM uses global measures such as discriminative body alignment to select the features. Body alignment is performed by maximizing the boosted classifier score, our *distance* measure, through iteratively warping the HOG locations to the image, and computing the gradient of the score with respect to the shape parameter. Since BDM warps landmarks, as opposed to the entire image in BAM, the landmarks can be evaluated using the original image data at the warped locations. We have applied BDM to the MIT pedestrian database [21], with excellent results.

The proposed approach has three main contributions:

◇ Our appearance model consists of a set of local features trained discriminatively by collecting samples from original unwarped image observations. Discriminative learning enables us to fully take advantage of the labeled data and model the alignment process as moving from any negative shape toward the positive shape for a particular image. Using unwarped image samples allows learning to use both interior object appearance and boundary information.

◇ Our fitting algorithm deforms/warps the spatial distribution of learnt local features, such that the classification score, *i.e.*, the total response of the warped features on the given image, is maximized. It has the advantage that the collection of all features jointly determines the updating of the shape parameter. Finally, fitting is efficient because no image warping is involved in the optimization.

◇ In terms of applications, we focus on efficient body alignment from surveillance-type low resolution images. This is distinct from most prior work on human modeling, where the usual goal has been to recover complex body articulation from high resolution images with less concern on efficiency.

Body alignment is the fundamental basis for many high-level vision tasks, but at the same time it is an extremely difficult problem due to highly deformable body configurations and dramatic variations in body appearance. There is a rich line of work in articulated pose estimation [9,11,19,23–25,28] with three common characteristics. First, the human subjects of interests, such as sports players, often express a wide variety of poses. Second, they normally perform on relatively high resolution images. Third, they are usually far from being efficient for real-time applications. This paper takes a different route of research. To be specific, we are interested in standing/walking individuals with mostly front and back views in the context of surveillance imagery. This implies that targets will have a reasonable amount of pose variation, whereas the image resolution is relatively low and real-time execution is often required. This makes our problem challenging and very unique.

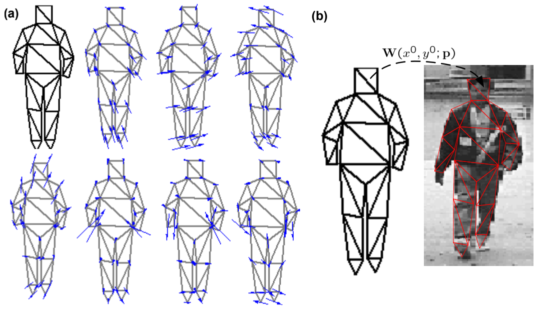

Figure 2. **Shape model and warping.** (a) The mean shape and top 7 shape bases of the PDM; (b) Given a pixel coordinate (x^0, y^0) in the mean shape $\mathbf{s}_0$, $\mathbf{W}(x^0, y^0; \mathbf{p})$ indicates the corresponding pixel in the image observation.

2. Shape and Appearance Model Learning

An image alignment algorithm is composed of a modeling component and a fitting component. In this section, we introduce the modeling component of the BDM, which consists of models for shape (set of landmarks or PDM) and appearance (collection of HOG features).

2.1. Shape Model Learning

We use a PDM consisting of several landmarks to represent the shape of the body [2]. Given a database, each image is manually labeled with a set of 2D landmarks, $\{x_i, y_i\}_{i=1,\cdots,v}$. The collection of landmarks of one image forms a shape observation, $\mathbf{s} = [x_1, y_1, x_2, y_2, ..., x_v, y_v]^T$. Eigenanalysis is then applied to an entire set of observations to learn the PDM. Then a particular shape instance is represented as,

$$\mathbf{s}(\mathbf{p}) = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i, \quad (1)$$

where $\mathbf{s}_0$ is the mean shape, $\mathbf{s}_i$ is the i^{th} shape basis, and $\mathbf{p} = [p_1, p_2, ..., p_n]^T$ is the shape parameter. Figure 2(a) shows an example of PDM. By design, the first four shape bases represent global translation and rotation. As shown in Figure 2(b), a warping function from the mean shape coordinate system to the coordinates in the image observation is defined as a piece-wise affine warp:

$$\mathbf{W}(x^0, y^0; \mathbf{p}) = [1\ x^0\ y^0]\mathbf{a}(\mathbf{p}), \quad (2)$$

where (x^0, y^0) is a pixel coordinate within the mean shape domain, $\mathbf{a}(\mathbf{p}) = [\mathbf{a}_1(\mathbf{p})\ \mathbf{a}_2(\mathbf{p})]$ is a unique 3×2 affine

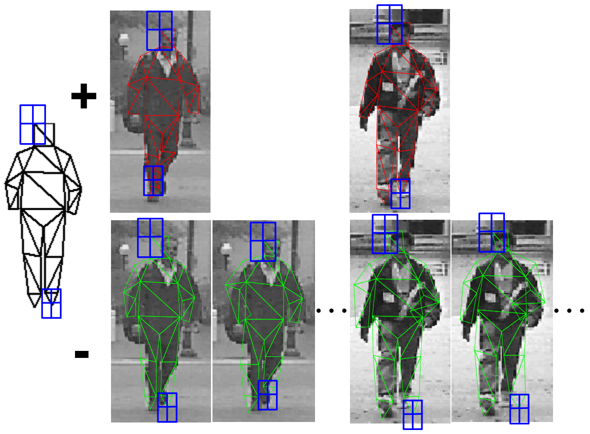

Figure 3. **Training data and the hypothesis space.** Each image has one positive shape ($\mathbf{p}_i^0$) and a number of negative shapes ($\mathbf{p}_i^j$). The pixel set in the mean shape domain defines the hypothesis space, where each HOG feature can find its corresponding locations in two shape classes via the piece-wise affine warp. Boosting selects a local feature set that best distinguishes two shape classes.

transformation matrix that relates each triangle pair in $\mathbf{s}_0$ and $\mathbf{s}(\mathbf{p})$. Given a shape parameter $\mathbf{p}$, $\mathbf{a}(\mathbf{p})$ needs to be computed for each triangle. However, since the knowledge of which triangle each pixel (x^0, y^0) belongs to is known a priori, the warp can be efficiently performed via a table lookup (see [18] for detailed description).

2.2. Appearance Model Learning

Given an image region, $\mathbf{I}$, we want to define a function $F(\mathbf{I};\mathbf{p})$ as our appearance model, which takes the image region and a shape parameter as input, and outputs a score. In our case we use how a shape instance represents shape of the underlying human body to determine the appearance model. Specifically, if the shape instance $\mathbf{s}(\mathbf{p})$ is the ground truth shape of the image region $\mathbf{I}$, where we denote $\mathbf{p}$ to be a *positive shape*, $F(\mathbf{I};\mathbf{p})$ returns a positive score. Otherwise, $\mathbf{p}$ denotes a *negative shape* and $F(\mathbf{I};\mathbf{p})$ is negative. In other words, $F(\mathbf{I};\mathbf{p})$ indicates whether or not $\mathbf{p}$ represents a true shape parameter for the underlying image region.

With this formulation, the appearance model is actually a two-class classifier. An important aspect of training a classifier is the choice of features. One could use holistic or local features. Holistic features such as eigenfaces [26], have been commonly used in face recognition. On the other hand, local features such as Haar [20,27], HOG [4,12], and SIFT [16] are popular for representing objects with large variations. Since the human body is highly deformable in both appearance and shape, we adopt a local feature representation. In particular, we use a linear combination of several local features to define the appearance model:

$$F(\mathbf{I};\mathbf{p}) = \sum_{m=1}^{M} f_m(\mathbf{I};\mathbf{p}) \tag{3}$$

where $f_m(\mathbf{I};\mathbf{p})$ is a function operating on one local feature of $\mathbf{I}$.

Given this formulation of the appearance model, we can use machine learning tools such as boosting. Boosting refers to a simple yet effective method of learning an accurate prediction function by combining a set of weak classifiers [6]. Note that $f_m(\mathbf{I};\mathbf{p})$ in Equation 3 can be viewed as a weak classifier operating on $\mathbf{I}$. To realize a boosting framework, we need to specify three key elements: hypothesis space construction, weak classifier design, and learning procedure. These are described in the following sections.

2.2.1 Hypothesis space construction

Hypothesis space denotes the set of potential features from which the final features are chosen. We need to define how the positive and negative training samples are obtained. Note that the goal of discriminative modeling is to learn the appearance difference between the positive and negative samples. For a human body image $\mathbf{I}_i$, the ground truth shape parameter $\mathbf{p}_i^0$ corresponding to the manual labels is its positive shape. The negative shapes $\mathbf{p}_i^j$ are obtained by perturbing $\mathbf{p}_i^0$ in the shape space. Then the training samples can be obtained in two ways. The first approach is to warp the image $\mathbf{I}_i$ to the mean shape using positive and negative shape parameters, and use the warped images $\mathbf{I}_i(\mathbf{W}(x^0, y^0; \mathbf{p}_i^0))$ and $\mathbf{I}_i(\mathbf{W}(x^0, y^0; \mathbf{p}_i^j))$ as the positive and negative training samples, respectively. This is how BAM is learned [14]. It has one drawback that only the appearance information within the boundary is utilized since no background content is warped. However, the human body has much larger appearance variability than faces. Furthermore, edge information has been shown to be more useful than interior texture for human body representation [4]. Therefore, instead of warping images to the mean shape, we overlay the positive and negative shapes on the image region, and the local features are directly evaluated on the image data.

For BAM, the hypothesis space can be easily constructed by going through every pixel in the mean shape domain, since all training samples are defined in that domain. In contrast, for BDM feature correspondence needs to be built across all training samples, which is achieved by the warping function defined in Equation 2. For each pixel (x_k^0, y_k^0) in the mean shape, we compute its corresponding pixels in all training samples based on their associated shape parameters. Hence, one feature in the hypothesis space can be represented as:

$$\mathcal{F}(x_k^0, y_k^0, w, h) = \{\mathbf{W}(x_k^0, y_k^0; \mathbf{p}_i^j)\}_{i=1,\cdots,K,\, j=0,\cdots,J}, \tag{4}$$

where (w, h) is the size of the local feature, K is the number of training images, and J is the number of perturbations per image. This feature vector lists the feature locations of K positive training samples and $K \times J$ negative samples. The

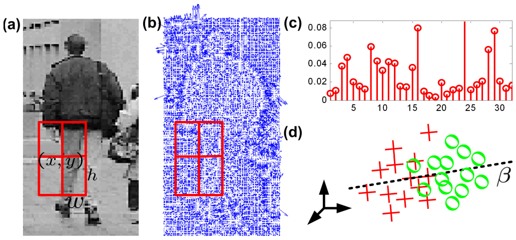

Figure 4. **Weak classifier.** (a) The parametrization of a block; (b) The gradient map; (c) The HOG of a block; (d) The HOG features of two classes and the LDA projection.

number of features in $\mathcal{F}$ is the multiplication of the number of pixels in the mean shape and the number of different (w, h) combinations (see Figure 3).

2.2.2 Weak classifier design

We adopt the HOG feature [4, 12] in our weak classifier design for the following reasons: (a) the gradient computation captures the edge information around body silhouettes; (b) the cell array structure makes HOG location sensitive, which is necessary for alignment; and (c) the histogram feature obtained through binning allows HOG to be rotation robust within certain angles, which suits well our application domain where the body configuration is constrained.

Using a block with 2 × 2 cells, HOG can be parameterized by (x, y, w, h), where (x, y) is the center of the block and (w, h) is the size of a cell (see Figure 4). For each cell, the b-bin histogram of the gradient magnitude at different orientations is computed. The histograms of all cells are concatenated to form a $4b$-dimensional HOG feature vector, which is then normalized to be a unit vector. When a multidimensional feature vector is used in the weak classifier, the conventional method of computing the threshold in the decision stump classifier cannot be directly applied. We employ the idea of boosted histograms proposed by Laptev [10]. Weighted Linear Discriminative Analysis (LDA) is applied to the HOG features of positive and negative samples, and results in the optimal projection direction β. Thus, HOG can be converted to a 1-D feature by computing its inner product with β.

In summary, we use the following weak classifier:

$$f(\mathbf{I}; \mathbf{p}) = \frac{2}{\pi} \tan^{-1}(\beta^T \mathbf{h}(\mathbf{I}; x, y, w, h) - t), \qquad (5)$$

where $\mathbf{h}$ is the HOG feature of image $\mathbf{I}$ evaluated at (x, y, w, h) and t is a threshold. Since (x, y, w, h) is one feature drawn from the hypothesis space, we have $(x, y) = \mathbf{W}(x^0, y^0; \mathbf{p})$, and

$$f(\mathbf{I}; \mathbf{p}) = \frac{2}{\pi} \tan^{-1}(\beta^T \mathbf{h}(\mathbf{I}; \mathbf{W}(x^0, y^0; \mathbf{p}), w, h) - t). \quad (6)$$

Similar to [14, 15], we use the $\tan^{-1}()$ function, instead of the commonly used decision stump, because of its differentiability with respect to the shape parameter $\mathbf{p}$.

Algorithm 1: The GentleBoost algorithm.

```
Input: Training data and their class labels $\{x_i, y_i\}_{i=1,\cdots,K}$
Output: A strong classifier $F(x)$
Initialize weights $w_i = 1/K$, and $F(x) = 0$
for $m = 1, 2, ..., M$ do
    (a) Fit $f_m(x)$ by weighted least-squares of $y_i$ to $x_i$:
        $f_m(x) = \arg\min_{f \in \mathcal{F}} \epsilon(f) = \sum_{i=1}^{K} w_i (y_i - f(x_i))^2$
    (b) Update $F(x) = F(x) + f_m(x)$
    (c) Update the weights by $w_i = w_i e^{-y_i f_m(x_i)}$
    (d) Normalize the weights such that $\sum_{i=1}^{K} w_i = 1$
return $F(x) = \sum_{m=1}^{M} f_m(x)$
```

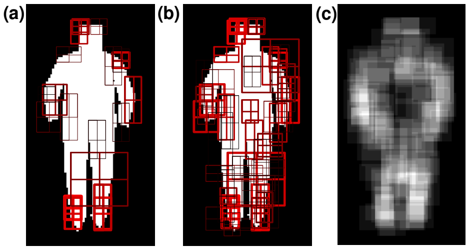

Figure 5. **Feature selection results.** The top 20 (a), 50 (b) HOG features and the density map of the top 200 features (c). The rank of the features is illustrated by the redness and thickness of lines.

2.2.3 Learning procedure

We employ the GentleBoost algorithm [7] (Algorithm 1) due to its superior performance over other boosting variants in object detection applications [13]. The boosting-based learning algorithm proceeds with the following iterative steps: 1) select features by evaluating the classification error of each feature in the hypothesis space and 2) update weights of training samples so that the later learning stages focus on the challenging samples.

Given a feature (x_k^0, y_k^0, w, h) from the hypothesis space, HOG can be evaluated for each training sample efficiently by accessing its integral histograms [22] using the feature locations in Equation 4. Weighted LDA is then applied to compute β, and finally t is obtained through binary search along the span of LDA projections of all training samples, such that the weighted least square error is minimal. Due to the large hypothesis space, Step (a) in Algorithm 1 is the most computationally intensive step in the learning process.

The learning process results in a number of weak classifiers, each represented by $5 + 4b$ parameters $\mathbf{c}_m = (x_m^0, y_m^0, w_m, h_m, \beta_m, t_m)$. We consider the set of weak classifiers $\{\mathbf{c}_m\}_{m=1,\cdots,M}$ as our appearance model representation. Together with PDM, they form the Boosted Deformable Models (BDM). Figure 5 shows the feature set trained from a body dataset with 100 images. The density map clearly shows that most features are indeed selected from the body boundary, rather than the inner body.

3. Model Fitting

3.1. Problem Definition

Given an image region $\mathbf{I}$ and the learned BDM, we obtain the formal problem we are trying to solve: *find the shape parameters* $\mathbf{p}$ *to maximize the score of the strong classifier*

$$\max_{\mathbf{p}} \sum_{m=1}^{M} f_m(\mathbf{I}, \mathbf{p}). \tag{7}$$

In the context of body alignment, solving this problem means that given the initial shape parameters $\mathbf{p}^{(0)}$, we look for the new shape parameter that leads to the maximal score from the strong classifier. Because coordinate warping is involved in the objective function, this is a nonlinear optimization problem. We choose to use the gradient ascent method to solve this problem iteratively. Notice that the key idea of the above problem definition, *i.e.*, alignment through maximizing a two-class classifier score, is the same as the work of [1, 14].

3.2. Algorithm Derivation

Combining Equation 6 and Equation 7, the function to be maximized becomes

$$F(\mathbf{I};\mathbf{p}) = \sum_{m=1}^{M} \frac{2}{\pi} \tan^{-1}(\beta_m^T \mathbf{h}(\mathbf{I}; \mathbf{W}(x_m^0, y_m^0; \mathbf{p}), w_m, h_m) - t_m). \tag{8}$$

Taking the derivative with respect to $\mathbf{p}$ gives

$$\frac{dF}{d\mathbf{p}} = \frac{2}{\pi} \sum_{m=1}^{M} \frac{\frac{\partial \mathbf{h}}{\partial \mathbf{p}}^T \beta_m}{1 + (\beta_m^T \mathbf{h} - t_m)^2}, \tag{9}$$

Since the HOG feature position depends on $\mathbf{p}$, we have

$$\frac{\partial \mathbf{h}}{\partial \mathbf{p}} = \frac{\partial \mathbf{h}}{\partial x_m} \frac{\partial x_m}{\partial \mathbf{p}} + \frac{\partial \mathbf{h}}{\partial y_m} \frac{\partial y_m}{\partial \mathbf{p}}. \tag{10}$$

As an example, we show how to compute the derivative of the HOG feature vector with respect to one of the cell location parameters x_m, *i.e.*, $\frac{\partial \mathbf{h}}{\partial x_m}$. The other partial derivative, $\frac{\partial \mathbf{h}}{\partial y_m}$, can be computed similarly. The $4b$-dimensional HOG feature $\mathbf{h} = [h_1, h_2, \cdots, h_{4b}]^T$ for an image $\mathbf{I}$ is computed from b integral images of the magnitude of the gradient at each orientation $\{\bar{\mathbf{I}}_1, \bar{\mathbf{I}}_2, \cdots, \bar{\mathbf{I}}_b\}$. For example the first b-bin of $\mathbf{h}$ is computed via

$$\begin{aligned} h_j = {} & \bar{\mathbf{I}}_j(x_m - w, y_m - h) + \bar{\mathbf{I}}_j(x_m, y_m) \\ & - \bar{\mathbf{I}}_j(x_m - w, y_m) - \bar{\mathbf{I}}_j(x_m, y_m - h), \end{aligned} \tag{11}$$

where $j \in [1, b]$. Hence, the derivative of $\mathbf{h}$ with respect to x_m can be computed by

$$\begin{aligned} \frac{\partial h_j}{\partial x_m} = {} & \frac{\partial \bar{\mathbf{I}}_j}{\partial x}\Big|_{(x_m - w, y_m - h)} + \frac{\partial \bar{\mathbf{I}}_j}{\partial x}\Big|_{(x_m, y_m)} \\ & - \frac{\partial \bar{\mathbf{I}}_j}{\partial x}\Big|_{(x_m - w, y_m)} - \frac{\partial \bar{\mathbf{I}}_j}{\partial x}\Big|_{(x_m, y_m - h)}, \end{aligned} \tag{12}$$

where $\frac{\partial \bar{\mathbf{I}}_j}{\partial x}|_{(x_m, y_m)}$ is the partial derivative of $\bar{\mathbf{I}}_j$ with respect to the horizontal axes x and evaluated at (x_m, y_m). It can be easily computed via discrete differentiation such as $\frac{\partial \bar{\mathbf{I}}_j}{\partial x}|_{(x_m, y_m)} = \frac{1}{2}[\bar{\mathbf{I}}_j(x_m + 1, y_m) - \bar{\mathbf{I}}_j(x_m - 1, y_m)]$. The derivative of the remaining $3b$ elements of $\mathbf{h}$ with respect to x_m can be computed in a similar fashion. Note that [15] uses the same approach to compute $\frac{\partial \mathbf{h}}{\partial x_m}$ for the purpose of online updating the boosted weak classifiers.

Algorithm 2: The BDM-based model fitting algorithm.

```
Input: BDM $\{\mathbf{s}_i, \mathbf{c}_m\}_{i=0,\cdots,n,\ m=1,\cdots,M}$, input image $\mathbf{I}$, initial
       shape parameter $\mathbf{p}$, and pre-computed Jacobian $[\frac{\partial x_m}{\partial \mathbf{p}}\ \frac{\partial y_m}{\partial \mathbf{p}}]$
Output: Shape parameter $\mathbf{p}$
Compute the $b$-bin integral histogram of image $\mathbf{I}$
repeat
    1. Compute the warped HOG locations on $\mathbf{I}$ by Equation 2
    2. Compute the HOG features: $e_m = \beta_m^T \mathbf{h} - t_m$
    3. Compute $\frac{\partial \mathbf{h}}{\partial x_m}$ and $\frac{\partial \mathbf{h}}{\partial y_m}$ by Equation 12
    4. Compute $\frac{\partial \mathbf{h}}{\partial \mathbf{p}}$ by Equation 10
    5. Compute $\triangle \mathbf{p}$ using $\triangle \mathbf{p} = \lambda \frac{2}{\pi} \sum_{m=1}^{M} \frac{\frac{\partial \mathbf{h}}{\partial \mathbf{p}}^T \beta_m}{1 + e_m^2}$
    6. Update $\mathbf{p} = \mathbf{p} + \triangle \mathbf{p}$
until $|| \sum_{i=1}^{n} \triangle \mathbf{p}_i \mathbf{s}_i || \leq \tau$.
```

Based on Equation 2 and the chain rule,

$$\begin{aligned} [\frac{\partial x_m}{\partial \mathbf{p}}\ \ \frac{\partial y_m}{\partial \mathbf{p}}] &= [\frac{\partial \mathbf{W}}{\partial \mathbf{a}_1} \frac{\partial \mathbf{a}_1}{\partial \mathbf{p}}\ \ \frac{\partial \mathbf{W}}{\partial \mathbf{a}_2} \frac{\partial \mathbf{a}_2}{\partial \mathbf{p}}] \\ &= \Big[[1\ x_m^0\ y_m^0] \frac{\partial \mathbf{a}_1}{\partial \mathbf{p}}\ \ [1\ x_m^0\ y_m^0] \frac{\partial \mathbf{a}_2}{\partial \mathbf{p}} \Big]. \end{aligned} \tag{13}$$

Since the affine parameter $\mathbf{a}$ is a linear function of $\mathbf{p}$, $\frac{\partial \mathbf{a}_1}{\partial \mathbf{p}}$ and $\frac{\partial \mathbf{a}_2}{\partial \mathbf{p}}$ are independent of $\mathbf{p}$. Thus $[\frac{\partial x_m}{\partial \mathbf{p}}\ \frac{\partial y_m}{\partial \mathbf{p}}]$ does not depend on $\mathbf{p}$. In other words, it can be pre-computed and does not need updating in each alignment iteration.

The derivative $\frac{dF}{d\mathbf{p}}$ indicates the direction to modify $\mathbf{p}$ such that the classification score increases. Thus, in the alignment iteration, the shape parameter $\mathbf{p}$ is updated via

$$\mathbf{p} = \mathbf{p} + \lambda \frac{dF}{d\mathbf{p}}, \tag{14}$$

where λ is the step size, until the change of the body landmark locations is less than a certain threshold τ. The complete model fitting algorithm is summarized in Algorithm 2.

We summarize the computation cost for each step during one iteration in Table 1. It is worth noting that the total cost per iteration, $O(b^2 nM)$, depends only linearly on the number of shape bases and weak classifiers. Furthermore, unlike the cost of BAM [14], $O(n(N + M))$, BDM does not depend on the image size N, which makes our model fitting faster and suitable for dealing with large images. This advantage directly benefits from the fact that we only deform/warp the feature locations, rather than the image itself. Note that the b^2 term in our cost is solely due to the particular HOG features used in this work. If we use the Haar feature in BDM-based fitting, as the BAM does

Table 1. The computation cost of each step in one alignment iteration. n is the number of shape bases, b is the number of bins in HOG, and M is the number of weak classifiers. The total cost is $O(b^2nM)$.

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
$O(M)$	$O(b^2M)$	$O(bM)$	$O(bnM)$	$O(b^2nM)$	$O(n)$

in [14], the total cost per iteration will be $O(nM)$, which is significantly smaller than $O(n(N+M))$.

3.3. Discussion and Comparison

In this section we compare the BDM to some of the conventional models that are applied to image alignment.

Active Shape Model (ASM) [2]: In terms of model learning, both ASM and BDM treat the ensemble of local features as the appearance model. For ASM, only the local appearance information centered at each landmark is learned, which limits the potential of using other object parts for the alignment. In contrast, BDM learns an optimal feature set without being constrained by the landmark positions. As shown in Figure 5(a), the top features learnt in BDM need not be centered at any pre-defined landmark location. In terms of model fitting, both models deform the PDM onto the image observation and maximize the feature responses. The difference is that ASM updates each landmark position *sequentially* based on its own appearance feature, whereas BDM uses *all* local features jointly to update the shape parameters and modifies all landmarks positions *simultaneously*. This results in a computational advantage of BDM over ASM during the fitting, especially when the shape model has a large number of landmarks.

Active Appearance Model (AAM) [3, 18]: Both AAM and BDM use the same shape model, but different appearance models. AAM uses a generative eigenspace representation that models the global intensity variation of the shape-normalized images. In contrast, BDM uses a set of local features discriminatively learnt from the images with positive and negative shapes. Since only the feature locations and parameters are saved, BDM is more storage-efficient than AAM. In terms of model fitting, AAM minimizes the MSE between the warped image and the appearance model instance by estimating *both* the shape and appearance parameters, while BDM maximizes the classification score by estimating the shape parameters *only*. Hence, BDM has considerably less parameters to be estimated, which implies a more reliable fitting optimization process. Also due to the use of local features, BDM is inherently more robust to partial occlusion compared to AAM, which models global appearance variations.

Boosted Appearance Model (BAM) [14]: Both BAM and BDM are discriminative image alignment methods where a set of local features are learned from positive and negative samples. The major difference is how the training samples are defined. BAM uses shape-normalized images warped from two shape classes, whereas BDM uses image observations directly with two shape classes, as training samples. Hence, BDM uses a different model fitting algorithm, giving it two advantages. First, it utilizes both interior and exterior appearance around object of interests, while BAM uses interior appearance only. For objects with large appearance variation such as the human body, edge information is shown to be more useful than the interior appearance for detection [4]. Second, it is more computationally efficient. BAM needs to perform image warping at every iteration of the fitting process, whereas BDM only deforms the feature locations. Since the number of features is typically much less than that of the pixels in the mean shape, BDM is more efficient.

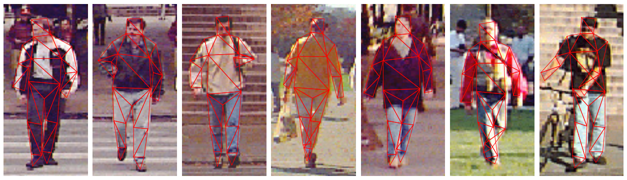

Figure 6. **Dataset examples with manual labels.** This dataset is challenging due to the low resolution (average body size is 110×46 pixels), cluttered background, and the presence of various accessories such as bags, bikes, etc. It has been used in pedestrian detection applications [21].

4. Experiments

4.1. Database and Evaluation Methodology

For testing we use a subset of 204 images from the MIT pedestrian set [21], because it closely resembles surveillance-type data (Figure 6). We manually label 29 landmarks for each image. We partition all images into two sets. Set 1 consists of 100 images and is used for training. The remaining 104 images forms Set 2. Both sets are used for testing.

Similar to the methodology used in face alignment [18], we perform the alignment on each image with different initialization and statistically evaluate the results. The initial landmarks are generated by randomly perturbing the manual landmarks by a Gaussian distribution whose variances are a multiple of the eigenvalues of shape bases. If the Root Mean Square Error (RMSE) between the aligned landmarks and the ground truth is less than 2.0 pixels, the alignment is deemed to have converged. To evaluate the robustness and accuracy of the alignment, we use the *Average Frequency of Convergence* (AFC), which is the percentage of the trials where the alignment converges, and the Histogram of the resultant RMSE (HRMSE) for the converged trials.

Finally, we compare the BDM with the method proposed in [5] where a decomposable triangulated graph is fitted to underlying image data in an energy minimization approach

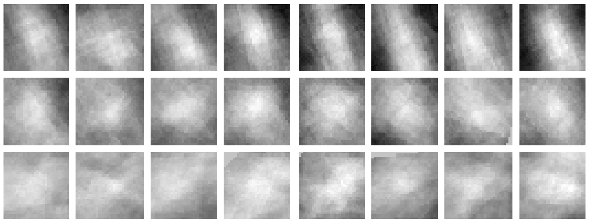

Figure 7. **Classification score surfaces.** For 8 body images (one per column), the score surface is generated while perturbing the shape parameter along pairs of shape bases (from top to bottom 1^{st} & 2^{nd}, 3^{rd} & 4^{th}, 5^{th} & 6^{th} shape basis respectively). The nice surface property helps the gradient-based fitting procedure.

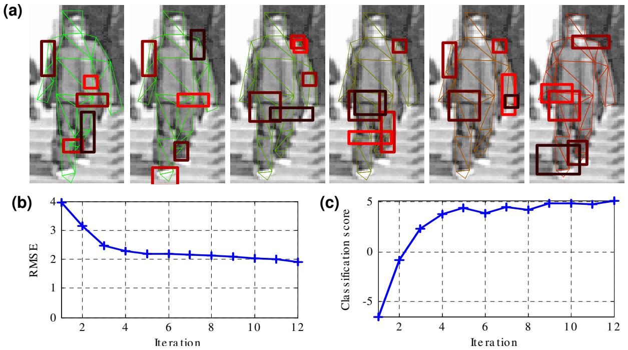

Figure 8. **An example of alignment process.** (a) Estimated landmarks at iteration 1, 2, 3, 4, 8 ,12 and the top 5 most contributing HOG features; (b) Decreasing RMSE during the alignment iteration; (c) Increasing classification score during the alignment.

using dynamic programming. The energy functional that is used consists of a data term that pulls the model towards salient edges, and a shape term that penalizes deformations from the underlying model using a piece-wise affine map. The candidate locations where vertices of the model could be placed using the DP algorithm are restricted to sampled Canny edge points to achieve reasonable run times. Both BRM and DP are tested using the same initializations.

4.2. Experimental Results

BDM is trained using images in Set 1. The set of 100 samples produces a PDM with 31 shape bases. For the appearance model, 100 positive and 2700 negative samples are used for boosting, since 27 negative shapes are synthesized for each image. The resulting strong classifier has 200 weak classifiers, as shown in Figure 5. We evaluate how the surface of classification score behaves under different perturbations of the ground truth shape parameter. In Figure 7, each column is the plot of the classification score of one image when its shape parameter is perturbed along a pair of shape bases while keeping the other bases fixed. The range of the perturbation equals 1.6 times the eigenvalues of two bases. Note that most surfaces are well-behaved and gradient ascent will find the optimum.

The intermediate steps in body alignment are shown in Figure 8. Starting with the initial landmarks shown in the left-most image of Figure 8(a), the alignment iteratively updates the landmarks, to reduce RMSE with respect to the ground truth and increase the classification score. The top 5 HOG features that contribute most to the current computation of $\triangle \mathbf{p}$ are shown. Note that in each iteration, different HOG features are chosen based on how the current shape deviates from the ground truth.

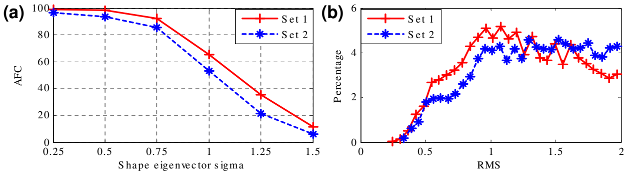

Figure 9. **Alignment results of BDM on Set 1 and 2.** (a) Average frequency of convergence with different amount of perturbation; (b) HRMSE for the trials where the alignment converges.

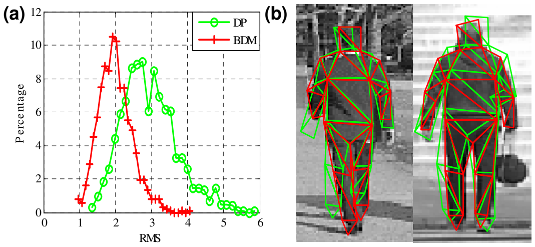

Figure 10. **Comparison between BDM and DP.** (a) HRMSE of all trials on Set 2 using BDM and DP when the perturbation sigma is 1; (b) Example fitting results of the DP algorithm.

We now perform large-scale experiments using BDM to study the robustness of the fitting using images in Set 1 and Set 2. The results are shown in Figure 9. The horizontal axis determines the amount of the perturbation of the initial landmarks. For a sigma value, we randomly generate 10 different initializations for each image. Hence there are 1000 trial for Set 1 and 1040 for Set 2. See Figure 9(a) for frequency of convergence. For the trials where the alignment converges, we plot the histogram of their respective converged RMSE in Figure 9(b). The step size λ is manually set to be the same constant for all experiments. Several observation can be made. First, as the perturbation gets larger, the alignment becomes more difficult. Second, there is only a marginal drop in performance when tested on Set 2, the unseen dataset. This shows the generalization capabilities of BDM. Some of the alignment results from both sets are shown in Figure 11. For these relatively low resolution images, the BDM algorithm does a good job in recovering the true body configuration, even in the presence of a bag or a bike. Finally, Figure 10 shows that on average BDM has lower RMSE than the DP method for the Set 2 experiment. Note that since the search-space of the DP algorithm is limited to sampled Canny edge points, the fitting accuracy is limited by the edge localization errors on these low-resolution images. The average running time of the DP algorithm is about 6 seconds in a C++ implementation.

In terms of computational efficiency, BDM-based fitting

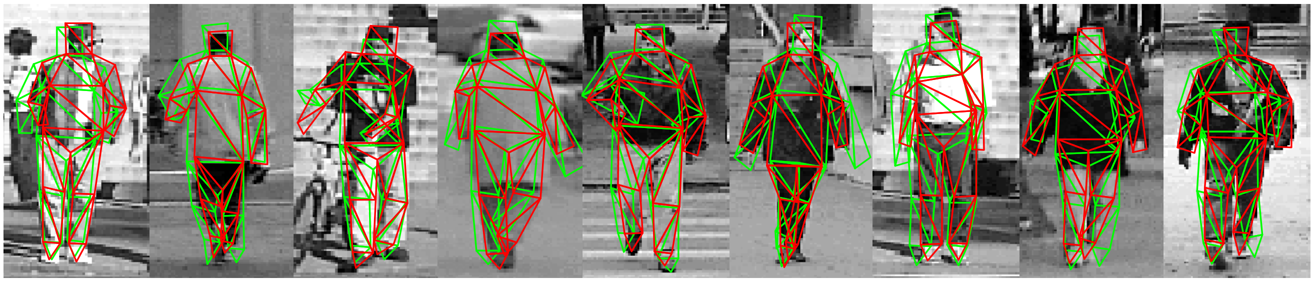

Figure 11. **Exemplar fitting result.** The initialization is shown in green lines. The fitted result is in red lines.

normally terminates in around 10 iterations. The average time for fitting one image is 0.20 seconds, which is based on a Matlab$^{\text{TM}}$ implementation running on a conventional 2.13 GHz Pentium$^{\text{TM}}$4 laptop. It is anticipated that our algorithm will run closer to real-time with a C++ implementation.

5. Conclusions

A novel image alignment framework is presented in this paper. Our appearance template is trained discriminatively using a set of local image features whose locations are indicated by two classes of shapes. During the fitting process, we deform the geometric distribution of local features by updating the shape parameter, such that the classification score of the warped features on the image observation is maximized. We apply this approach to the human body alignment problem in surveillance-type images, and obtain good results in comparison with prior work.

The successful application of our method to practical non-rigid object alignment problems requires two future steps. The first is a better choice of feature representation. In our framework, location sensitivity and rotation robustness are the two criteria of good features. In the case of human bodies, we can incorporate the orientation of body parts as the shifting factor of bins in HOG, which will improve its rotation robustness. The second is the modeling of the shape prior. Eigenspace is certainly a very primitive shape representation. More sophisticated shape modeling such as [8] will help both the learning and the fitting since only plausible shape instances are drawn from the model.

References

[1] S. Avidan. Support vector tracking. *IEEE TPAMI*, 26(8):1064–1072, 2004.

[2] T. Cootes, D. Cooper, C. Tylor, and J. Graham. A trainable method of parametric shape description. In *BMVC*, pages 54–61, 1991.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE TPAMI*, 23(6):681–685, 2001.

[4] N. Dalal and W. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[5] P. F. Felzenszwalb. Representation and detection of deformable shapes. *IEEE TPAMI*, 27(2):208–220, 2005.

[6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, 2000.

[8] L. Gu, E. P. Xing, and T. Kanade. Learning GMRF structures for spatial priors. In *CVPR*, 2007.

[9] G. Hua, M.-H. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. In *CVPR*, volume 2, pages 747–754, 2005.

[10] I. Laptev. Improvements of object detection using boosted histograms. In *BMVC*, volume 3, pages 949–958, 2006.

[11] M. W. Lee and I. Cohen. A model-based approach for estimating human 3D poses in static images. *IEEE TPAMI*, 28(6):905–916, 2006.

[12] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *CVPR*, volume 2, pages 53–60, 2004.

[13] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Proc. 25th Pattern Recognition Symposium*, pages 297–304, 2003.

[14] X. Liu. Generic face alignment using boosted appearance model. In *CVPR*, 2007.

[15] X. Liu and T. Yu. Gradient feature selection for online boosting. In *ICCV*, 2007.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] B. Lucas and T. Kanade. An iterative technique of image registration and its application to stereo. In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[18] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[19] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, volume 2, pages 326–333, 2004.

[20] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *ICCV*, pages 555–562, 1998.

[21] C. P. Papageorgiou and T. Poggio. Trainable pedestrian detection. In *ICIP*, volume 4, pages 35–39, 1999.

[22] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829–836, 2005.

[23] D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *CVPR*, volume 1, pages 206–213, 2006.

[24] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *ICCV*, volume 1, pages 824–831, 2005.

[25] L. Sigal and M. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, pages 2041–2048, 2006.

[26] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[27] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[28] J. Zhang, J. Luo, R. Collins, and Y. Liu. Body localization in still images using hierarchical models and hybrid search. In *CVPR*, volume 2, pages 1536–1543, 2006.