Discriminative Face Alignment

Xiaoming Liu, *Member, IEEE*

Abstract—This paper proposes a discriminative framework for efficiently aligning images. Although conventional Active Appearance Models (AAM)-based approaches have achieved some success, they suffer from the generalization problem, i.e., how to align any image with a generic model. We treat the iterative image alignment problem as a process of maximizing the score of a trained two-class classifier that is able to distinguish correct alignment (positive class) from incorrect alignment (negative class). During the modeling stage, given a set of images with ground truth landmarks, we train a conventional Point Distribution Model (PDM) and a boosting-based classifier, which acts as an appearance model. When tested on an image with the initial landmark locations, the proposed algorithm iteratively updates the shape parameters of the PDM via the gradient ascent method such that the classification score of the warped image is maximized. We use the term *Boosted Appearance Models (BAM)* to refer the learned shape and appearance models, as well as our specific alignment method. The proposed framework is applied to the face alignment problem. Using extensive experimentation, we show that, compared to the AAM-based approach, this framework greatly improves the robustness, accuracy and efficiency of face alignment by a large margin, especially for unseen data.

Index Terms—Face, alignment, Boosting, Active Appearance Models, AAM, Boosted Appearance Models, BAM, image alignment, gradient descent, landmark, generative vs. discriminative model.

1 Introduction

Image alignment is the process of moving and deforming a *template* to minimize the *distance* between the template and an image. Since Lucas and Kanade's seminar work [1], image alignment has found many applications in computer vision such as face fitting [2], image coding [3], tracking [4], [5], image mosaicing [6], medical image interpretation [7], industrial inspection [8], etc. With the introduction of Active Shape Models (ASM) [9] and Active Appearance Models (AAM) [2], [10], generative model-based face alignment/fitting has become more popular in the vision community.

Essentially, there are three elements to image alignment, namely *template representation*, *distance metric*, and *optimization method*. The template can be represented using a simple image patch, or the more sophisticated ASM or AAM. The Mean Squared Error (MSE) between the warped image and the template is one of the most widely used distance metrics. For optimization, gradient descent methods are commonly used to iteratively update the shape parameters, including Gauss-Newton, Newton, Levenberg-Marquardt, etc. The Inverse Compositional (IC) and Simultaneously Inverse Compositional (SIC) methods proposed by Baker and Matthews [11] are excellent examples of recent advances in image alignment. Their novel formulation of warp update during the optimization results in an efficient algorithm for fitting AAM to facial images. However, as indicated by [12], [13], the alignment performance degrades quickly when the AAM are trained on a large dataset and fit to images that were not seen during the AAM training. We assert that this generalization issue is caused by the eigenspace-based appearance modeling and the use of MSE as the distance metric.

To remedy the generalization problem, this paper proposes a novel discriminative framework for image alignment. As shown in Fig. 1(a), for the *template representation*, we train a boosting-based classifier that learns the decision boundary between two classes, given a face dataset with ground truth landmarks. The positive class includes images warped with ground truth landmarks; the negative one includes images warped with perturbed landmarks. The set of trained weak classifiers, based on Haar-like rectangular features [14], [15], acts as an appearance model. We then use the score from the trained strong classifier as the *distance metric*, which is a continuous value proportional to the accuracy of alignment, to align an image by maximizing its classification score. Similar to the term AAM and ASM, we use *Boosted Appearance Models (BAM)* to refer to the learned shape and appearance models, as well as our specific alignment method. As shown in Fig. 1(b), the image warped using the initial shape parameters $\mathbf{p}^{(0)}$ will likely have a negative score. The shape parameters are iteratively updated via gradient ascent such that the classification score keeps increasing. The proposed framework is applied to the face alignment problem. With extensive experimentation, we show that, compared to the AAM-based approach, this framework greatly improves the robustness, accuracy and efficiency of face alignment by a large margin, especially for unseen data.

The proposed image alignment framework has three main contributions.

1 In terms of *template representation*, we propose a novel discriminative method of appearance modeling via boosting. Unlike the conventional generative model-based AAM that only model the Gaussian distribution of warped images under correct alignment, the BAM learn the discriminative properties between warped images under both correct and incorrect alignment. Also, the appearance model of BAM is a much more compact representation compared to that of AAM, since only

Manuscript received October 9, 2007; revised July 21, 2008; accepted August 28 2008.
Xiaoming Liu is with the GE Global Research.

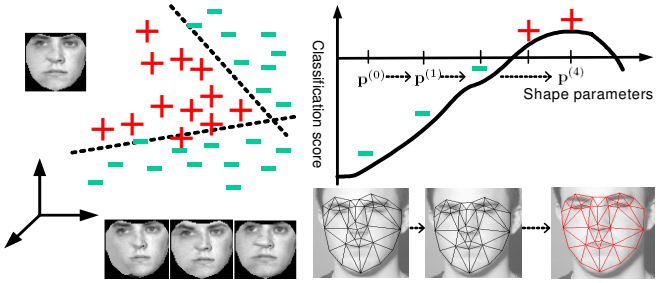


Fig. 1. (a) Model training: learn a two-class classifier that distinguishes correct alignment (positive class) from incorrect alignment (negative class) based on warped images; (b) Face alignment: given initial shape parameters, iteratively update the parameter via gradient ascent such that the warped image achieves the maximal score from the trained classifier.

the weak classifier parameters are stored as the *model*. Furthermore, the local rectangular features used in the BAM makes it robust to partial occlusion. Finally, we also incorporate an approach to model and enforce the shape constraint such that an improved BAM can be learned.

2 In terms of *distance metric*, we propose a novel alignment algorithm through maximizing the classification score. Compared to minimizing the MSE in AAM-based approaches, our method benefits from the fact that the boosting method is known to be capable of learning from a large dataset and generalizing well to unseen data. The final classification score after convergence also provides a natural way to describe the *quality* of the image alignment.

3 In terms of *applications*, we greatly improve the performance of generic face alignment. The AAM-based approach performs well for person-specific or small population-based face alignment. Our proposal improves it toward the ultimate goal that a face alignment algorithm should be very generic, *i.e.*, be able to fit to faces from any unknown subject with any pose, expression or lighting in real time.

The paper is organized as follows: After a brief description of the related work in Section 2, this paper presents the model learning and fitting methods of the conventional AAM in Section 3. The training of BAM is given in Section 4, and the BAM based fitting algorithm is presented in Section 5. Section 6 makes a detail comparison between BAM and AAM, as well as BAM and ASM. Section 7 describes our extensive experimental results. The paper concludes in Section 8.

2 Related Work

Image alignment is a fundamental problem in computer vision. Since early 90s, ASM [9] and AAM [2], [10] have become one of the most popular model-based image alignment methods because of their elegant mathematical formulation and efficient computation. For the template representation, AAM's basic idea is to use two eigenspaces to model the object shape and shape-free appearance respectively. For the distance metric, the MSE between the appearance instance synthesized from the appearance eigenspace and the warped appearance from the image observation is minimized by iteratively updating the shape and/or appearance parameters. ASM and AAM have been applied extensively in many computer vision tasks, such as facial image processing [16]–[18], medical image analysis [19], image coding [3], industrial inspection [8], object appearance modeling [20], etc. Cootes and Taylor [21] have an extensive survey on this topic.

Due to the needs of many practical applications such as face recognition, expression analysis and pose estimation, extensive research has been conducted in face alignment. Zhou *et al.* [22] propose a Bayesian inference solution and an EM based method is used to implement the MAP estimation. This work was further extended to multi-view face alignment via a Bayesian mixture model [23]. Liang *et al.* [24] employ a shape constrained Markov network searching for accurate face alignment. Even though variety of approaches are proposed, the majority of prior work in face alignment is based on ASM, AAM or their variations [16]–[18], [25]–[31]. Due to the rich literature on applying AAM/ASM for face alignment, detailed survey on this topic is beyond the scope of this paper. Some representative work are listed here. Yan *et al.* [16] introduce the texture-constrained active shape models, which effectively incorporate not only the shape prior and local appearance around each landmark, but also the global texture constraint over the shape. Dedeoglu *et al.* [18] integrate the AAM-based fitting with an image formulation model such that the fitting on low resolution images is greatly improved. AAM have also been adapted [26] and fused [27], which benefit the model fitting. Donner *et al.* [28] improve the fitting speed using the canonical correlation analysis that models the dependency between texture residuals and model parameters during search. Under the ASM framework, Cristinacce and Cootes incorporate a template model for each landmark either generatively [17] or discriminatively [25]. In all the previous work, AAM and their variations employ a generative appearance modeling approach.

It is well known that AAM-based face alignment has difficulty with generalization [12], [13]. That is, the alignment tends to diverge on images that are not included as the training data for learning the model, especially when the model is trained on a large dataset. We assert this is mostly due to the fact that the generative appearance model only learns the appearance variation retained in the training data. When more training data is used to model larger appearance variations, the representational power of the eigenspace is very limited even under the cost of a much higher-dimensional appearance subspace, which in turn results in a harder optimization problem. Because estimating higher-dimensional appearance parameters implies more chance to be fallen into local minimum. Also, using the MSE as the distance metric essentially employs an "interpretation through synthesis" approach, further limiting the generalization capability by the representational power of the appearance model. Researchers have noticed this problem and proposed methods to handle it. Jiao *et al.* [32] suggest

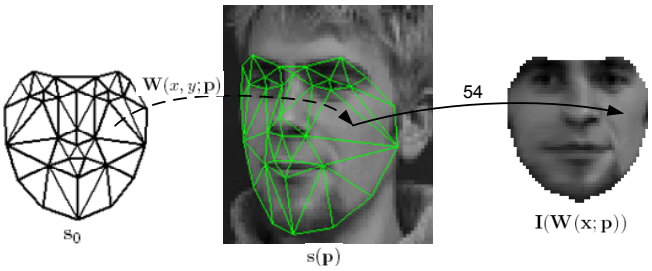


Fig. 2. Image warping from the image observation to the mean shape. Given a pixel coordinate (x, y) in the mean shape $\mathbf{s}_0$, $\mathbf{W}(x, y; \mathbf{p})$ indicates the corresponding pixel in the image observation, whose intensity value (54) is obtained via bilinear interpolation and treated as one element of the N-dimensional vector $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$.

using Gabor wavelet features to represent the local appearance information. Hu *et al.* [33] utilize a wavelet network representation to replace the eigenspace-based appearance model, and demonstrate improved alignment with respect to illumination changes and occlusions.

The basic idea of our proposal is optimization via *maximizing a classification score*. Similar ideas have been explored in object tracking research [34]–[36]. Avidan [34] estimates the 2D translation parameters by maximizing the Support Vector Machine (SVM) classification score. Limitations of this method include dealing with partial occlusions and the large number of support vectors which might be needed for tracking, burdening both computation and storage. Williams *et al.* [35] build a displacement expert, which takes an image as input and returns the displacement, by using Relevance Vector Machine (RVM). Since RVM is basically a probabilistic SVM, it still suffers from the problem of requiring a large set of support vectors. The recent work by Hidaka *et al.* [36] performs face tracking (2D translation only) via maximizing the score from a Viola and Jones face detector [14], where a face versus non-face classifier is trained. Our proposal differs from these works in that we are dealing with a much larger shape space than object tracking, where often only 2D translation is estimated.

3 AAM and Model Fitting

In this section, we will first introduce the training procedure of the conventional AAM, then describe the AAM-based fitting algorithms.

3.1 Shape and Appearance Modeling

The shape model and appearance model part of AAM are trained with a representative set of facial images. The shape model, which is conventionally called Point Distribution Model (PDM) [9], is learned in the following procedure. Given a face database, each facial image is manually labeled with a set of 2D landmarks, $[x_i, y_i]$ $i = 1, 2, ..., v$. The collection of landmarks of one image is treated as one observation from the random process defined by the shape model, $\mathbf{s} = [x_1, y_1, x_2, y_2, ..., x_v, y_v]^T$. Eigen-analysis is applied to the observation set and the resultant model represents a shape as,

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

where $\mathbf{s}_0$ is the mean shape, $\mathbf{s}_i$ is the i^{th} shape basis, and $\mathbf{p} = [p_1, p_2, ..., p_n]^T$ are the shape parameters. By design, the first four shape bases represent global translation and rotation. Together with other bases, a warping function from the model coordinate system to the coordinates in the image observation is defined as $\mathbf{W}(x, y; \mathbf{p})$, where (x, y) is a pixel coordinate within the face region $R(\mathbf{s}_0)$ defined by the mean shape $\mathbf{s}_0$. Fig. 2 shows one example of the warping process.

We define the warping function with a piecewise affine warp:

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

where $\mathbf{a}(\mathbf{p}) = [\mathbf{a}_1(\mathbf{p})\ \mathbf{a}_2(\mathbf{p})]$ is a 3 by 2 affine transformation matrix that is unique to each triangle pair between $\mathbf{s}_0$ and $\mathbf{s}(\mathbf{p})$. Given shape parameters $\mathbf{p}$, the $\mathbf{a}(\mathbf{p})$ matrix needs to be computed for each triangle. However, since the knowledge of which triangle each pixel (x, y) belongs to can be pre-computed, the warp can be efficiently performed via a table lookup, inner product as in (2), and bilinear interpolation of the image observation $\mathbf{I}$. We denote the resultant warped image as a N-dimensional vector $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$, where $\mathbf{x}$ is the set of all pixel coordinates within $R(\mathbf{s}_0)$.

Given the shape model, each facial image is warped into the mean shape via the above warping function. These shape-normalized appearances from all training images are fed into an eigen-analysis and the resulting model represents an appearance as,

$$\mathbf{A}(\mathbf{x}; \lambda) = \mathbf{A}_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \mathbf{A}_i(\mathbf{x}), \tag{3}$$

where $\mathbf{A}_0$ is the mean appearance, $\mathbf{A}_i$ is the i^{th} appearance basis, and $\lambda = [\lambda_1, \lambda_2, ..., \lambda_m]^T$ are the appearance parameters. Fig. 3 shows the AAM trained using 534 images of 200 subjects from the ND1 face database [37].

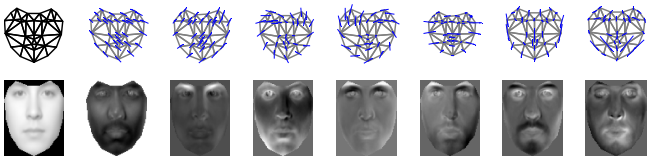


Fig. 3. The mean and first 7 basis vectors of the shape model (top) and the appearance model (bottom) trained from the ND1 database. The shape basis vectors are shown as arrows at the corresponding mean shape landmark locations.

3.2 AAM-based Fitting

AAM can synthesize facial images with arbitrary shape and appearance within a population. Thus, the AAM can be used to *explain* a facial image by estimating the optimal shape and appearance parameters such that the synthesized image is as

similar to the image observation as possible. This leads to the cost function used for model fitting [10],

$$J(\mathbf{P},\lambda)=\frac{1}{N}\sum_{\mathbf{x}\in R(\mathbf{s}_0)}||\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}))-\mathbf{A}(\mathbf{x};\lambda)||^2, \quad (4)$$

which is the MSE between the warped observation $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}))$ and the synthesized appearance instance $\mathbf{A}(\mathbf{x};\lambda)$, and N is the total number of pixels in $R(\mathbf{s}_0)$.

Traditionally this minimization is solved by gradient decent methods. Baker and Matthews [11] proposed the IC and SIC methods that greatly improve the fitting performance. Their basic idea is that the role of appearance templates and the input image is switched when computing $\Delta\mathbf{P}$. Thus a large portion of time-consuming steps in parameter estimation can be pre-computed and remain constant during the fitting iteration.

4 Boosted Appearance Models

The Boosted Appearance Models are composed of a shape model, a appearance model, and a specific model fitting method. The shape model of BAM is the same as the PDM of the conventional AAM, as we introduced in Section 3.1. In this section, we present the training method of the appearance model of BAM.

4.1 Appearance Modeling in BAM

Similar to AAM, our appearance model is defined on the warped image $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}))$. That is, we want to define a function $F(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p})$ as our appearance model, which takes the warped image $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}))$ and shape parameters $\mathbf{P}$ as input, and outputs a score. In our case we use how a shape instance represents the shape of the underlying facial features to determine the appearance model. Specifically, if the shape instance $\mathbf{s}(\mathbf{p})$ is the ground truth shape of the face image $\mathbf{I}$, where we denote $\mathbf{p}$ to be a *positive shape*, $F(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p})$ returns a positive score. Otherwise, $\mathbf{p}$ denotes a *negative shape* and $F(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p})$ is negative. In other words, $F(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p})$ indicates whether or not $\mathbf{p}$ represents the true shape parameters for the underlying face image.

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 [38], have been commonly used in face recognition. On the other hand, local features such as Haar [14], [15], HOG [39], and SIFT [40] are popular for representing objects with large variations. Since occlusion-robust face alignment is desired for many applications, we adopt a local feature representation. In particular, we use a linear combination of several local features to define the appearance model:

$$F(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p})=\sum_{m=1}^{M}f_m(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p}) \quad (5)$$

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

Given this formulation of the appearance model, machine learning tools such as boosting become a natural choice to learn such a model. Boosting refers to a simple yet effective method of learning an accurate prediction function by combining a set of weak classifiers [41] using summation. It has shown greater performance than many machine learning paradigms, when applied to challenging tasks [42]. Note that $f_m(\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}));\mathbf{p})$ in (5) can be viewed as a weak classifier operating on $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{P}))$. For the simplification of the notation, we will denote the weak classifier and the strong classifier as $f_m(\mathbf{p})$ and $F(\mathbf{p})$ respectively. To realize a boosting-based learning framework, we need to specify three key elements: training samples, weak classifier design, and learning procedure. These are described in the following sections.

4.1.1 Training samples

As described before, since our appearance model, i.e., the set of weak classifiers $f_m(\mathbf{p})$, is defined on the warped images, a training sample for our boosting-based appearance learning is a N-dimensional warped image $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}))$.

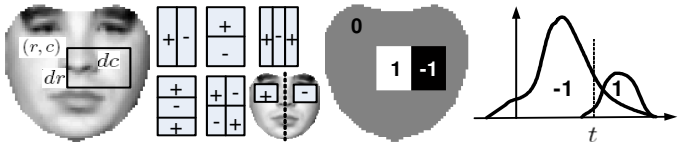
Given a face database with manually labeled landmarks $\mathbf{s}$, the ground truth shape parameters $\mathbf{p}$ for each face image $\mathbf{I}$ are computed based on (1). Then, the set of warped images $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}))$ are treated as *positive training samples* ($y_i=1$) for the boosting. For each image, a number of negative shapes $\mathbf{p}'$ are synthesized by random perturbation. Equation (6) describes our perturbation, where ν is a n-dimensional vector with each element uniformly distributed within $[-1,1]$, μ is the vectorized eigenvalues of all shape bases in the PDM, and σ is a constant scale controls the range of perturbation.

$$\mathbf{p}'=\mathbf{p}+\sigma\nu\cdot\mu \quad (6)$$

Together with the original face image, one perturbed negative shape can provide one *negative training samples* $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}'))$ ($y_i=-1$). Hence, this is an unbalanced learning problem since a number of negative shapes and also negative training samples can be generated with one positive training sample.

4.1.2 Weak classifier design

We now define the weak classifier. Given that real-time face alignment is desired, we construct the weak classifier based on the Haar-like rectangular features [14], [15], whose fast evaluation is enabled by the integral image [14]. As shown in Fig. 4(a), the rectangular feature can be parameterized by (r,c,dr,dc,b), where (r,c) is the top-left corner, (dr,dc) is the height and width, and b is the feature type. Fig. 4(b) shows the six feature types used in our algorithm. The first five feature types are the conventional ones used in the Viola-Jones face detection [14]. The sixth feature type, where two detached rectangles occupy the mirror-position of two sides of the face, is proposed based on the fact that the warped face is approximately symmetric in the horizontal direction. The hypothesis space $\mathcal{F}$, where (r,c,dr,dc,b) resides, is obtained via an exhaustive construction within the mean shape. For example, there are more than $300,000$ such rectangular features for a mean shape with size of 30×30.



(a) (b) (c) (d)

Fig. 4. (a) The parametrization of a weak classifier; (b) The six feature types; (c) The notional template $\mathbf{A}$, whose inner product with the warped image is mathematically equivalent to computing a rectangular feature; (d) Let the rectangular features of the positive samples have larger mean than that of the negative samples, by multiplying a sign $g = \{1, -1\}$, and then estimate the threshold t that has the minimal weighted LS error via binary search.

In summary, we use the weak classifier defined as follows:

$$f_m(\mathbf{p}) = \frac{2}{\pi} atan(g_m \mathbf{A}_m^T \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p})) - t_m), \qquad (7)$$

where $\mathbf{A}_m$ is a template, g_m is ± 1 and t_m is a threshold. Given a rectangular feature (r, c, dr, dc, b), we can generate a corresponding template $\mathbf{A}$, as shown in Fig. 4(c). The inner product between the template and the warped image is equivalent to computing the rectangular feature using the integral image.

The $atan()$ function makes this weak classifier different from the commonly used stump classifier in the AdaBoost algorithm, since the classifier response $f_m(\mathbf{p})$ is continuous within -1 and 1. As we will show in Section 5, this difference is very critical for the face alignment application because the continuous classifier response is utilized to update the shape parameters.

It is possible that the weak classifier can be defined differently as the $atan()$ function in (7). The basic guidance of designing weak classifiers is to consider both the discriminability, where a classifier is optimal with the least number of weak classifiers, and the derivability, where the response of the strong classifier is favorable for optimization. For example, the stump classifier has high discriminability. However, its derivability property (infinity at zero) prevents it being used in our approach. Other potential weak classifier definition could be the sigmoid function. We choose $atan()$ mostly due to the fact that its derivative is in a relative simple form.

4.1.3 Learning procedure

We employ the boosting framework (Algorithm 1) to train a classifier that is able to distinguish correct alignment from incorrect alignment. Different variants of boosting have been proposed in the literature [43]. We use the GentleBoost algorithm [42] based on two considerations. First, unlike the commonly used AdaBoost algorithm [41], the weak classifier in the GentleBoost algorithm is a soft classifier with continuous output. This property allows the output of the strong classifier to be smoother and favorable as an alignment metric. In contrast, the hard weak classifiers in the AdaBoost algorithm lead to a piecewise constant strong classifier, which is difficult to optimize. Second, as shown in [44], for object detection tasks, the GentleBoost algorithm outperforms other boosting methods in that it is more robust to noisy data and more resistant to outliers.

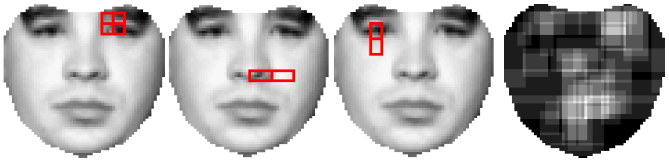
Algorithm 1: The GentleBoost algorithm.

Input: Training data $\{x_i; i = 1, 2, ..., K\}$ and their corresponding class labels $\{y_i; i = 1, 2, ..., K\}$.
Output: A strong classifier $F(x)$.
1. Initialize weights $w_i = 1/K$, and $F(x) = 0$.
2. **for** $m = 1, 2, ..., M$ **do**
   (a) Fit the regression function $f_m(x)$ by weighted least-squares (LS) of y_i to x_i with weights w_i:
   $$f_m(x) = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \epsilon(f) = \sum_{i=1}^{K} w_i (y_i - f(x_i))^2. \qquad (8)$$
   (b) Update $F(x) = F(x) + f_m(x)$.
   (c) Update the weights by $w_i = w_i e^{-y_i f_m(x_i)}$ and normalize the weights such that $\sum_{i=1}^{K} w_i = 1$.
end
3. Output the classifier $F(x) = \sum_{m=1}^{M} f_m(x)$.

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 (Step 2(a)) and 2) update weights of training samples so that the later learning stages focus on the challenging samples (Step 2(c)).

Given a set of facial images with manual label, positive and negative training samples are generated according to Section 4.1.1. Once the rectangular features for a set of training samples are computed, $g_m = -1$ if the mean of the features of positive samples is less than that of the negative samples, otherwise $g_m = 1$. In other words, for each weak classifier, g_m is set such that overall the positive samples have larger feature response than the negative ones. This is an important property of the weak classifier since it ensures that the maximization of the learned strong classifier will lead to the correct alignment. The threshold, t_m, is obtained through binary search along the span of the rectangular features, such that the weighted LS error is minimal. The aforementioned weak classifier computation is conducted for each feature in the hypothesis space, and the weak classifier with minimal error $\epsilon(f)$ is selected. Such an exhaustive search procedure is a crucial step in the GentleBoost algorithm, Step 2(a) in Algorithm 1, which is normally fairly slow due to the huge hypothesis space.

The results of the boosting are a number of weak classifiers, each with 7 parameters $\mathbf{c}_m = (r, c, dr, dc, b, g, t)$. We consider the set of weak classifiers $\{\mathbf{c}_m; m = 1, 2, ..., M\}$ as the appearance model of the Boosted Appearance Models (BAM). Fig. 5 shows the top 3 rectangular features, and the spatial distribution of the top 50 features trained from a face dataset with 400 images. Note that the learned rectangular features are well aligned with the boundary of facial features, such as eyes and nose. Also most rectangular features are located on facial features. This observation consists well with our intuition that it is the appearance information around the facial



(a) (b)

Fig. 5. (a) The top 3 features selected by the GentleBoost algorithm. The rectangles are well aligned with the boundary of facial features, such as eyes and nose; (b) The brightness indicates the density of the top 50 rectangular features. Most classifier features are located on facial features.

features determining the accuracy (or correctness) of the face alignment.

4.2 Appearance Modeling with Shape Constraint

It is well known that the Point Distribution Model assumes a multi-dimensional Gaussian distribution for the shape instances. Hence, any shape instance within a hyperellipsoid, whose radius is proportional to the corresponding eigenvalue, is assumed to be an *allowable* shape by the PDM. However, because of the non-Guassian nature of the sample distribution, this assumption might not be true. That is, it is possible to find instances within the hyperellipsoid to be *unallowable* shapes, i.e., ones that are unlikely to be the valid shapes of any human face. It is understood that an ideal shape model should be both *complete*, which means it models all valid variations, and *concise*, which means it does not model anything that is not valid. Many researchers have noticed this problem and proposed approaches to solve it [45], [46].

This modeling issue in the shape model also causes problems for our appearance modeling. BAM require a large set of negative shapes to be synthesized by perturbing the ground truth shape parameters, as shown in (6). With the random perturbation, it is very likely that part of perturbed negative shapes will be unallowable shape instances, whose corresponding negative training samples (the warped image $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}'))$) will be fed to the boosting-based learning. On one hand, in practical applications the unallowable shape instance will not be used to initialize the model fitting. Hence learning to distinguish these negative training samples is unnecessary. On the other hand, due to the high-dimensional shape space, there are practically infinite negative training samples for the BAM learning, where those from unallowable shapes will undoubtedly make the learning problem to be more complex.

To alleviate this problem, we adopt the shape parameter optimization approach proposed by Li and Ito [45]. Although the original idea was proposed for ASM, we are certain that it can be useful for providing shape constraint for BAM learning as well. This approach has two components. One is the learning of shape constraints, which can be conducted after the PDM is trained. The other is the enforcement of the shape constraints, which is used in generating the negative shapes and in each iteration of the model fitting.

The basic idea of the learning component is to quantize each element of the shape parameters and approximate its distribution as a table. Since the shape parameters are in a n-dimensional space, the complete description of the parameter distribution will require a n-dimensional table, which will be an extremely sparse and statistical unmeaningful table given the limited training shape instances. Thus, for each element of the shape parameters, we are only interested in the joint distribution between itself and two most *correlated* elements, which are obtained by computing the correlation coefficient between two elements and selecting the top two from the $n-1$ coefficients. The joint distribution for each element is described by a 3-dimensional table, where the table size equals to the level of the quantization (7 in our experiment), and the entry of the table counts the number of shape parameters whose three quantized elements match with the table index. In our experiments, the learning component results in a $n \times 2$ matrix for the most correlated elements, and a $7 \times 7 \times 7$ table for each of the n elements. To enforce of the shape constraints, it is required that all elements of given shape parameters should *hit* a non-zero entry in their corresponding n tables. If not, the element of the shape parameters needs to be *drag* until it hits a non-zero entry, while in the same time causing the least amount of shape deformation. Hence, the drag always starts with the element with the smaller eigenvalue and iteratively modifies the quantized element toward the center of the hyperellipsoid.

We can apply this shape constraint for improving the BAM learning. That is, after each negative shape is perturbed as (6), we enforce the shape constraint such that the negative shape is a valid shape instance. The remaining learning procedure is the same as before. Thus the shape constraint acts as a filter to ensure that BAM focus the learning on the valid training samples. We call the resulting BAM as Shape Constrained BAM (SC-BAM), whose performance will be presented in Section 7.2.

5 Face Alignment

5.1 Problem Definition

Given the trained BAM, we formally define the problem we are trying to solve: *Find the shape parameters* $\mathbf{p}$ *to maximize the score of the strong classifier*

$$\max_{\mathbf{p}} \sum_{m=1}^{M} f_m(\mathbf{p}). \tag{9}$$

In the context of face alignment, solving this problem means that given the initial shape parameters $\mathbf{p}^{(0)}$, we look for the new shape parameters that lead to the warped image with the maximal score from the strong classifier.

Because image 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.

5.2 Algorithm Derivation

Plugging (7) into (9), the function to be maximized is

$$F(\mathbf{p}) = \sum_{m=1}^{M} \frac{2}{\pi} atan(g_m \mathbf{A}_m^T \mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) - t_m). \tag{10}$$

Algorithm 2: The image alignment algorithm of BAM.

Input: Input image $\mathbf{I}$, initial shape parameters $\mathbf{p}$, pre-computed Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$, and BAM with the shape model $\{\mathbf{s}_i; i = 0, 1, ..., n\}$ and the appearance model $\{\mathbf{c}_m; m = 1, 2, ..., M\}$.

Output: Shape parameters $\mathbf{p}$.

0. Compute the 2D gradient of the image $\mathbf{I}$.

repeat

1. Warp $\mathbf{I}$ with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$.
2. Compute the feature for each weak classifier: $e_m = g_m \mathbf{A}_m^T \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p})) - t_m; m = 1, 2, ..., M$.
3. Bilinearly interpolate the gradient of image $\mathbf{I}$ at $\mathbf{W}(\mathbf{x}; \mathbf{p})$.
4. Compute the steepest descent image $SD = \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$.
5. Compute the integral images for each column of SD and obtain the rectangular features for each weak classifier: $\mathbf{b}_m = g_m SD^T \mathbf{A}_m; m = 1, 2, ..., M$.
6. Compute $\triangle \mathbf{p}$ using $\triangle \mathbf{p} = \alpha \frac{2}{\pi} \sum_{m=1}^{M} \frac{\mathbf{b}_m}{1+e_m^2}$.
7. Update $\mathbf{p} = \mathbf{p} + \triangle \mathbf{p}$.

until $|| \sum_{i=1}^{n} \triangle \mathbf{p}_i \mathbf{s}_i || \leq \tau$.

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

$$\frac{dF}{d\mathbf{p}} = \frac{2}{\pi} \sum_{m=1}^{M} \frac{g_m [\nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T \mathbf{A}_m}{1 + [g_m \mathbf{A}_m^T \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p})) - t_m]^2}, \tag{11}$$

where $\nabla \mathbf{I}$ is the *gradient* of the image $\mathbf{I}$ evaluated at $\mathbf{W}(\mathbf{x}; \mathbf{p})$, and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the *Jacobian* of the warp evaluated at $\mathbf{p}$.

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

$$\mathbf{p} = \mathbf{p} + \alpha \frac{dF}{d\mathbf{p}}, \tag{12}$$

where α is the step size, until the change of the facial landmark locations is less than a certain threshold τ.

We now discuss how to compute $\frac{dF}{d\mathbf{p}}$ efficiently. Based on (2) and the chain rule,

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = [\frac{\partial \mathbf{W}}{\partial \mathbf{a}_1} \frac{\partial \mathbf{a}_1}{\partial \mathbf{p}} \quad \frac{\partial \mathbf{W}}{\partial \mathbf{a}_2} \frac{\partial \mathbf{a}_2}{\partial \mathbf{p}}], \tag{13}$$

where $\frac{\partial \mathbf{W}}{\partial \mathbf{a}_1}$ and $\frac{\partial \mathbf{W}}{\partial \mathbf{a}_2}$ are both N by 3 matrices and N is the number of pixels in the warped images. 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 $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ does not depend on $\mathbf{p}$. In other words, it can be pre-computed and does not need updating in each alignment iteration. Note that we have this computational gain only because we use the piecewise affine warp, which is linear on $\mathbf{p}$. In theory, $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ needs to be re-evaluated if $\mathbf{p}$ are updated, when for example the warp is polynomial on $\mathbf{p}$.

The alignment algorithm is summarized in Algorithm 2. The first step is to compute the warped coordinates of all pixels in the mean shape space, *i.e.*,$\mathbf{W}(\mathbf{x}; \mathbf{p})$, and bilinearly interpolate $\mathbf{I}$ to compute $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$. The computation cost for both operations is $O(N)$. The second step is to compute the integral image of $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$, whose computation cost is $O(N)$, and obtain the feature response for each weak classifier via the integral image, $g_m \mathbf{A}_m^T \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p})) - t_m$, whose computation cost is constant for each classifier. Hence, the total coast for this step is $O(N+M)$. The third step interpolates the gradient of $\mathbf{I}$ at the known warped coordinates $\mathbf{W}(\mathbf{x}; \mathbf{p})$. Similar to the first step, its computation cost is also $O(N)$.

The fourth step is to multiply $\nabla \mathbf{I}$ and the pre-computed $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$. The result $SD = \nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is called the steepest descent image, which is a N by n matrix where n is the number of shape bases. The computation cost for this step is $O(nN)$. Similar to $\mathbf{A}_m^T \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$, in the fifth step, we do not need to perform the actual matrix multiplication between SD and $\mathbf{A}_m$. Instead, we first compute the integral images of each column in SD, whose computation cost is $O(nN)$, and then calculate the rectangular features of $\mathbf{A}_m$ by a fast table lookup. The sixth step is to compute the gradient $\triangle \mathbf{p}$ by combining the results from each weak classifier. The last step is to update the shape parameters $\mathbf{p}$.

Basically $\mathbf{b}_m$ in Step 5 can be considered as the gradient direction derived from each weak classifier. However, its contribution to the final gradient $\frac{dF}{d\mathbf{p}}$ is determined by $\frac{1}{1+e_m^2}$. The weak classifiers with low $|e_m|$ are less certain in their own classification decision. These weak classifiers contribute more to the final *travel direction*. Obviously this observation conforms well with intuition. Also, this observation opens the possibility to speed up the fitting algorithm. That is, we can rank the magnitudes of all feature responses $|e_m|$ after the second step, and *only* compute (the fifth step) and integrate (the sixth step) $\mathbf{b}_m$ for the weak classifiers whose $|e_m|$ are smaller than a threshold. This approximation will speed up the computation for both the fifth and sixth step.

Also, if the SC-BAM is trained, the model fitting should make use of the shape constraint as well. We achieve this by enforcing the constraint after Step 7 of Algorithm 2 in each iteration. Basically this will ensure that the intermediate estimations of the shape parameters during the fitting process are all valid shape instances.

We summarize the computation cost for each step during one iteration in Table 1. Note that because of using integral images, the most computationally intensive step, Step 5, can be computed in a relatively efficient way.

TABLE 1
The computation cost of the alignment algorithm at one iteration. n is the number of shape bases, N is the number of pixels within the mean shape, and M is the number of weak classifiers.

Step 1	**Step 2**	**Step 3**	**Step 4**
$O(N)$	$O(N+M)$	$O(N)$	$O(nN)$
Step 5	**Step 6**	**Step 7**	**Total**
$O(n(N+M))$	$O(nM)$	$O(n)$	$O(n(N+M))$

6 Comparison and Discussion

Having introduced the BAM, we now make a comparison between the generative model-based AAM and the discriminative model-based BAM. Table 2 summarizes the major aspects we

TABLE 2
Comparison between AAM and BAM.

	AAM	**BAM**
Shape model	PDM	PDM
Appearance model	eigenspace of global intensity variations	a set of local rectangular features
Labels for learning appearance	only ground truth labels	both ground truth labels and perturbed labels
Estimated variables	both shape and appearance parameters	only shape parameters
Fitting quality	L_2 norm (4)	classification score (10)
Storage	Nm	$7M$
Learning	fast and require less data	slow and require more data

would like to compare. Finally we also briefly compare BAM with ASM.

First, in terms of model representation, both AAM and BAM use exactly the same shape model, which is the PDM trained from the manual labels. However, these two models utilize totally different appearance models. AAM use a generative eigenspace representation that models the global intensity variation of the shape-normalized facial appearance. While BAM optimally learn the classification boundary between the correctly warped images and incorrectly warped images in a well-known boosting framework. Thus, BAM take better advantage of the manually labeled facial image compared to AAM, because the BAM learn from not only the appearance of correct alignment, which is essentially what AAM do, but also the appearance of incorrect alignment.

Second, because of the local rectangular features, the BAM are inherently more likely to be robust to partial occlusion compared to AAM, which models the global appearance variations. As shown in Fig. 13, even when a large portion of facial appearance is occluded, the local features on the un-occluded area still manage to align most of the images.

Third, AAM employ two generative models, whose parameters are the unknown in the objective function (4). Thus, during the AAM-based fitting, both the shape parameters and the appearance parameters need to be estimated. In contrast, BAM employ a generative shape model and a discriminative appearance model. Only the shape parameters are estimated during the fitting, while the appearance model is used in a discriminatively fashion to guild how the shape parameters should be updated. In this regard, BAM have much less parameters to be estimated, which implies a more reliable optimization and less chance to fall into local minimum, especially considering the fact that in AAM the dimension of the appearance subspace is generally higher than that of the shape subspace.

Fourth, often in computer vision, knowing when the algorithm fails is as important as how the algorithm performs. For example, given an arbitrary to-be-fitted facial image, in additional to output the facial landmarks' position, a fitting algorithm should also produce a *fitting quality* measurement, which indicates how well the fitting was performed on this particular image. If the fitting quality is smaller than a pre-defined threshold, fitting failure can be reported. For BAM, the final classification score after convergence can be directly used as the fitting quality because it was originally trained to reflect the correctness of the alignment. For instance, if the fitting quality is negative, it is very likely that the fitting has fallen into local maximum. However, for AAM-based fitting, the final measure from the objective function (4) might not directly reflect the quality of the fitting. Especially in the context of the generic face alignment, the limited representation power of the appearance model might contribute a larger amount of residual to the final measure, compared to the residual due to the incorrect shape parameters.

Fifth, from the storage point of view, the BAM have a much more storage-efficient way of modeling the appearance information. We do not store the training data. The knowledge of the training data is absorbed in the selected local rectangular features. Hence, the BAM only require a 7 by M matrix to be saved as the *appearance model*. In contrast, AAM need a N by m matrix where m is the number of appearance bases, $N \gg M$, and $m > 7$. The storage-efficient property of the BAM enables the potential of performing model-based face alignment from mobile devices such as cell phones.

Finally, in terms of the complexity of model learning, AAM have the advantage in that eigenspace computation is very fast and does not require a large amount of training samples. While the training of BAM is a lot slower due to the huge hypothesis space in the boosting-based learning process. Also, because of boosting, BAM normally require a relative large number of training samples. However, the price paid for the learning process does get the reward that BAM greatly improve the robustness, accuracy, and efficiency of generic face alignment, as we will present in the next section.

It might be interesting to compare the conventional ASM with BAM as well. In terms of model learning, both ASM and BAM treat the ensemble of local appearance representations as the appearance model. For ASM, only the local appearance information around each landmark is learned, which might not be the most effective way of modeling. For example, as shown in the left plot of Fig. 5, the top local feature learned in BAM does not center at any pre-defined landmark location. BAM learn an optimal set of local features without being constrained by the landmark positions under the boosting framework. In terms of model fitting, ASM update each landmark position based on its own appearance representation. While BAM use *all* local features to update the shape parameters, which modifies all landmarks' positions simultaneously. This might imply a computational advantage of BAM over ASM during the fitting, especially when the shape model includes a large

(a) (b) (c) (d)

Fig. 6. Examples of the face dataset: (a) ND1 database, (b) FERET database, (c) IMM database, and (d) BIOID database.

TABLE 3
Summary of the dataset.

	ND1	FERET	IMM	BIOID
Images	534	200	234	230
Subjects	200	200	40	23
Variation	Frontal	Pose	Pose, expr.	Background, lighting
Set 1	200	200		
Set 2	334			
Set 3			234	
Set 4				230

number of landmarks. For example, in AAM-based facial expression analysis [47], 74 facial landmarks are used such that various facial action units can be detected.

7 Experiments

7.1 Face Dataset and Experimental Procedure

To evaluate our algorithm, we collect a set of 1198 images from multiple public available databases, including the ND1 database [37], FERET database [48], IMM database [49], and BIOID database [50]. Fig. 6 shows sample images from these four databases. We partition all images into four distinct datasets. Table 3 lists the properties of each database and partition. Set 1 includes 400 images (one image per subject) from two databases and is used as the training set for the AAM and BAM. Set 2 includes 334 images from the *same* subjects but different images as the ND1 database in Set 1. Set 3 includes 234 images from 40 subjects in the IMM database that were never used in the training. Set 4 includes randomly selected 230 images of 23 subjects from the BIOID database. This partition ensures that we have two levels of generalization to be tested, *i.e.*, Set 2 is tested as the unseen data of seen subjects; Set 3 and 4 are tested as the unseen data of unseen subjects. Set 4 is a particular challenging dataset since it is captured under cluttered background and various real-world illumination environment. There are 33 manually labeled landmarks for each one of 1198 images. To speed up the training process, we down-sample the image set such that the facial width is roughly 40 pixels across the set.

Given a dataset with ground truth landmarks, the experimental procedure consists of running the alignment algorithm on each image with a number of initial landmarks and statistically evaluating the alignment results. To generate the initial landmarks for an image, we randomly perturb its ground truth shape parameters using the same way as the BAM learning (6). Note that, as described Section 3.1, since the first 4 shape bases represent global translation and rotation, perturbation of shape parameters is equivalent to displacement in the horizontal and vertical directions, in-plane rotation, as well as local landmark re-position.

We declare that the alignment converges if the resultant Root Mean Square Error (RMSE) between the aligned landmarks and the ground truth is less than 1.0 pixel after the algorithm terminates. Two metrics are used to evaluate the alignment results for the converged trials. One is the *Average Frequency of Convergence* (AFC), which is the number of trials where the alignment converges divided by the total number of trials. The other is the Histogram of the resultant RMSE (HRMSE) of the converged trials, which measures how close the aligned landmarks are with respect to the ground truth. These two metrics measure the robustness and accuracy of alignment respectively.

We compare our algorithm with the Simultaneous Inverse Compositional (SIC) algorithm, which has been shown to perform best among the family of AAM-based methods [2]. We ensure both algorithms are tested under the same conditions. For example, both algorithms are initialized with the *same* set of randomly perturbed landmarks. Both algorithms have the same termination condition. That is, if the number of iterations is larger than 55 or the RMSE of estimated landmarks in consecutive iterations is less than 0.025 pixels. Also, HRMSE is only computed on the trials where both algorithms converge.

7.2 Experimental Results

We train the BAM on Set 1. There are 400 positive and 4000 negative samples, where each image synthesizes 10 negative samples, used in the boosting-based learning. The resultant BAM have 33 shape bases and 150 weak classifiers. When the number of weak classifiers is 50, the strong classifiers can generate less than 0.1% false alarm rate at 0% missed detection rate on the training set. In order to study how the number of weak classifiers affects the fitting performance, we let the training algorithm run until it produces 150 weak classifiers. In contrast, the AAM use the same PDM model as ours and an appearance model with 24 appearance bases. The number of the appearance bases is chosen such that 99% of the energy is retained in the appearance model for the training set.

To test the generalization capability of the trained BAM, we perform the classification using the BAM with 50 weak

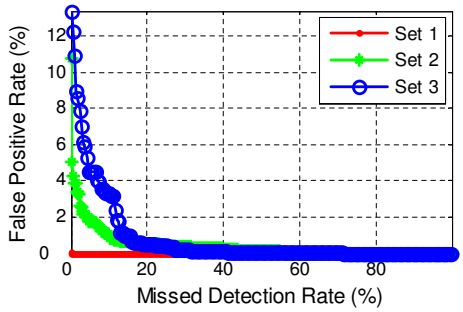
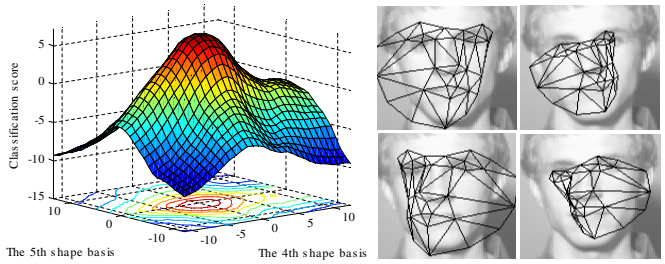


Fig. 7. Classification performance on three datasets.



(a) (b)

Fig. 8. (a) The classification score surface while perturbing the shape parameters in the neighborhood of the ground truth along the 4^{th} and 5^{th} shape basis. The convex surface favors the gradient ascent method; (b) The four perturbed facial landmarks when the perturbation is at the four corners of the surface on the left.

classifiers on three datasets. For Set 2, we obtain 334 positive samples by warping images using the ground truth landmarks and 3340 negative samples by perturbing each ground truth landmarks 10 times, using the same methodology as for Set 1. Similarly, 234 positive and 2340 negative samples are generated from Set 3. By setting different thresholds for the classification score $F(\mathbf{p})$, performance curves are shown in Fig. 7. Although it is expected that the performances on Set 2 and 3 are worse than that of Set 1, the BAM still achieve reasonable classification capability on the unseen data, which enables the potential of using the BAM in the alignment optimization.

Fig. 8(a) shows that for a given image, a convex surface of classification scores can be observed while perturbing the shape parameters along two shape bases and setting the shape parameters at other bases to the ground truth. It is obvious that the gradient ascent algorithm can perform well on this type of surface. The range of the perturbation (σ) equals 1.6 times the eigenvalue of these two bases. When the perturbation is at the maximal amount for two bases, the corresponding four perturbed landmarks are plotted at Fig. 8(b). In the following experiments, when the σ equals 1.6, the actual initial landmarks could be even further away from the ground truth compared to these four examples because all bases are allowed to be perturbed. To see the properties of score surfaces, more surfaces are plotted as images in Fig. 9, where the intensity corresponds to the classification score. Each sub-image is generated in the same way as Fig. 8(a). Each column is for one facial image and each row is for the perturbation along

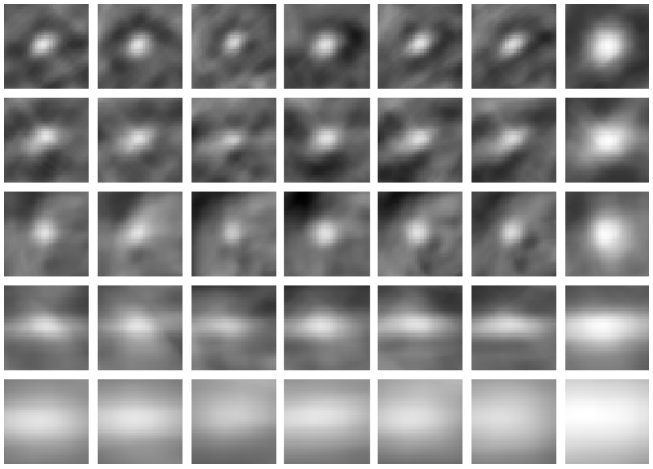
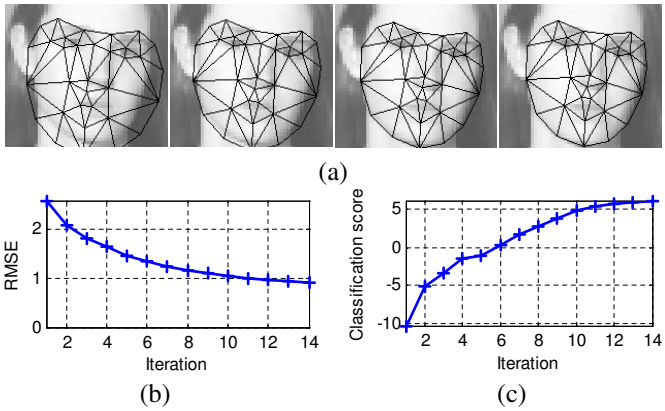


Fig. 9. The classification score surface of 7 facial images (one by each column) while perturbing the shape parameters along pairs of shape bases (from top to bottom 1^{st} & 2^{nd}, 2^{nd} & 3^{rd}, 3^{rd} & 4^{th}, 4^{th} & 5^{th}, 5^{th} & 6^{th} shape basis respectively).



(a) (b) (c)

Fig. 10. An example of boosting-based face alignment: (a) Estimated landmarks at iteration 1, 5, 10, and 14; (b) Decreasing RMSE during the iterative alignment process; (c) Increasing classification scores during the iterative alignment process.

two neighboring shape bases. For most cases, we see the intensity changes from high to low when the pixel deviates from the center, *i.e.*, the alignment gets less accurate. This nice monotonic surface is the key to a successful face alignment algorithm.

Fig. 10 illustrates an example of iterative boosting-based face alignment. Given the initial landmarks, as shown in the first image of Fig. 10(a), the alignment iteratively updates the facial landmarks, which has decreasing RMSE with respect to the ground truth and increasing classification score for the warped image. Note that computing the score is just for illustration purposes and is not a necessary step during the alignment iteration. However, the score after the alignment convergence, which is quickly computed via $\frac{2}{\pi}\sum_{m=1}^{M} atan(e_m)$, can serve as the measurement of the fitting quality.

The first experiment is to test the face alignment algorithms on Set 1. The results are shown in the first row of Fig. 11.

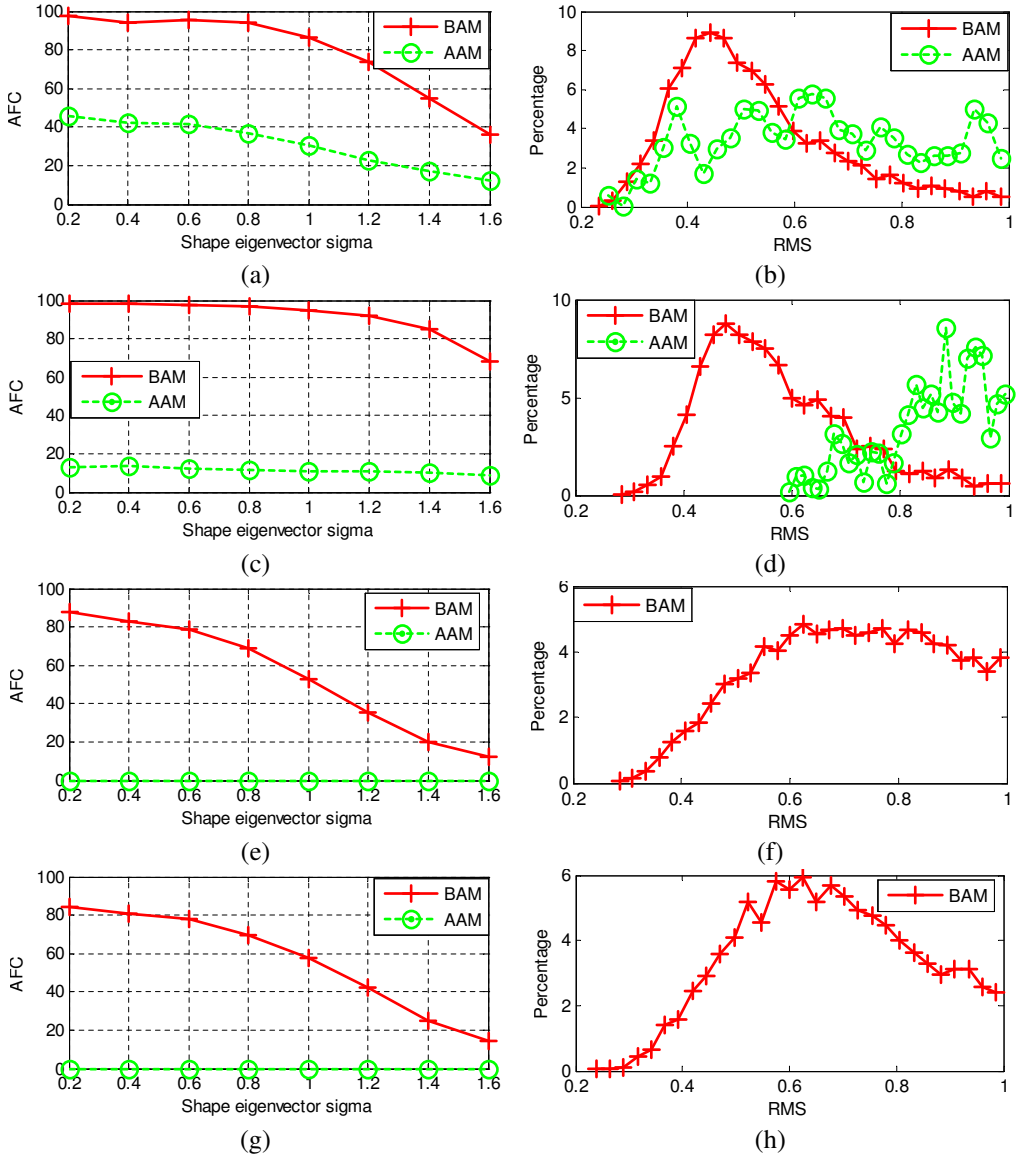


Fig. 11. Alignment results of two algorithms on Set 1, 2, 3, and 4. From top to bottom, each row is the result for one set. Left column is the AFC; right column is the histogram of the resultant RMSE for the trials where both algorithms converge. Only the HRMSE of the BAM is plotted at (f) and (h) since the SIC has no convergence on these two sets.

The horizontal axis determines the amount of the perturbation of the initial landmarks. Given one σ value, we randomly generate 2000 trials, where each one of 400 images has 5 random initializations. Each sample in Fig. 11(a) is averaged based on these 2000 trials. For the trials where both algorithms converge, we plot the histogram of their respective converged RMSE in Fig. 11(b). The same experiments are performed for Set 2, 3, and 4 with 2004, 2106, and 2070 trials respectively, using the same BAM as that of Set 1. The results are shown in the second, third, and fourth row of Fig. 11. The step size α is manually set to be the same constant for all experiments.

We make a number of observations from these experiments. First, BAM-based alignment performs substantially better than the AAM-based alignment using SIC, in terms of both alignment robustness (AFC) and accuracy (HRMSE). Second, although both algorithms have worse performance when fitting to unseen images, the BAM have a much lower performance drop compared to the SIC. Especially on Set 3 and 4, which are the most difficult cases, the AAM-based alignment always diverges, while the BAM perform reasonably well. In terms of AFC, the BAM perform slightly better on the unseen data of seen subjects (Set 2) than Set 1. This is mainly attribute to the fact that Set 2 includes only frontal view faces while Set 1 has substantial pose variations.

For all experiments in Fig. 11, the number of weak classifiers in the BAM is 100. An interesting question is the relationship between the number of weak classifiers and the fitting performance. Using the same experimental setup in Fig. 11, we perform the alignment experiment using the BAM with different numbers of weak classifiers. This experiment is conducted on Set 2 and the results are plotted in Fig. 12. It can be observed that 100 weak classifiers provide the best

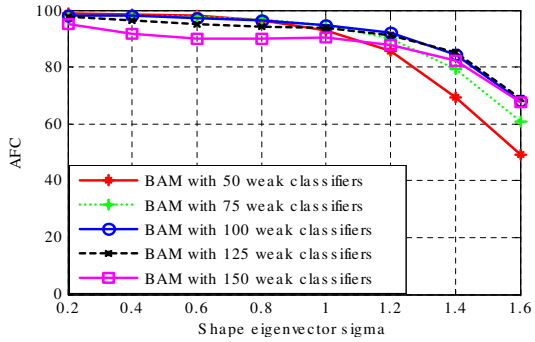
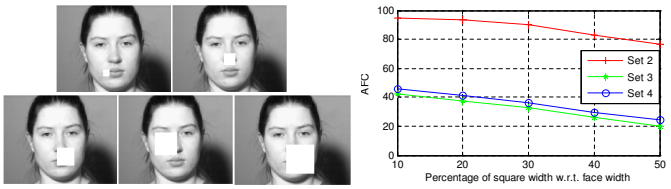


Fig. 12. Alignment results on Set 2 using the BAM with different numbers of weak classifiers.



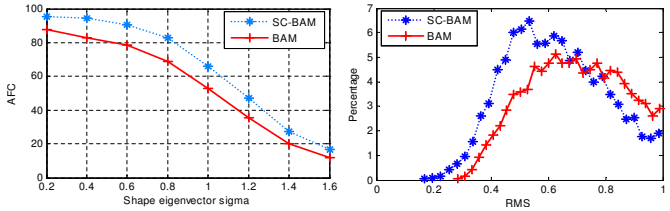
(a) (b)

Fig. 13. Alignment results on the occluded version of Set 2, 3 and 4: (a) Five different levels of occlusions; (b) The average frequency of convergence under five levels of occlusions.

overall performance. In other words, more weak classifiers do not necessarily result in better alignment performance. This is because that the most critical property for BAM to achieve optimal alignment performance is their continuous monotonic score surface. Even though more weak classifiers would improve the classification performance of BAM, the classification margin between positive and negative samples might be too large, which implies a score surface unfavorable to alignment.

One strength of rectangular features in the BAM is that they are localized features. Thus inherently they are likely to be robust to partial occlusion. We perform the second experiment to illustrate this. We generate a white square whose size is a certain percentage of the facial width and randomly place it on the tested face area. We perturb the initial landmarks in the usual way by fixing the σ of the shape bases to be 1.0. As shown in Fig. 13, five levels of occlusion are tested on Set 2, 3 and 4. Similar trend of performance degradation are observed in all three datasets when increasing the amount of occlusion. This shows that the boosting-based alignment can tolerate a certain level of occlusion because of the nature of features used in the appearance modeling.

The next experiment is to study how the shape constraint helps the model fitting. Similar to aforementioned BAM learning, we use the Set 1 as the training dataset and generate the same number of positive and negative training samples. The only difference comparing to BAM learning is that all negative shapes are shape constraint enforced. Once the SC-BAM is learned, the Set 3 is used for testing since it seems to be the most difficult test set as shown in Fig. 11. Both BAM-based and SC-BAM-based fitting are tested, while the latter has one



(a) (b)

Fig. 14. Performance comparison between SC-BAM and BAM on Set 3: (a) Average frequency of convergence; (b) Histogram of the RMSE.

TABLE 4
The computation cost of the alignment test on Set 2.

σ	0.2	0.6	1.0	1.4	1.6
BAM-iterations	7.1	7.3	7.6	9.1	9.0
SIC-iterations	54.4	54.8	54.6	54.4	55.4
BAM-time (sec.)	0.10	0.11	0.11	0.14	0.14
SIC-time (sec.)	0.62	0.59	0.61	0.62	0.60

additional step to enforce shape constraint compared to the former. Fig. 14 illustrate the experimental results. Note that the BAM results in Fig. 14 are the same as those in Fig. 11(e,f). It can be observed that for both the fitting robustness and accuracy, SC-BAM-based fitting achieves substantial better performance compared to BAM-based fitting. In particular, Fig. 14 (b) alleviates the potential concern that enforcing shape constraint might limit the solution space, hence reducing the fitting accuracy. We attribute the fitting performance improvement to enhanced appearance modeling induced by the shape constraint.

Table 4 lists the computation cost of the alignment test on Set 2, without occlusion. The number of iterations and times in fitting one image are averaged for the trials where both algorithms converge. It is clear that with different amount of perturbation, the BAM perform consistently faster than the SIC algorithm and converges in fewer iterations. The cost is based on a Matlab™ implementation of both algorithms running on a conventional 2.13 GHz Pentium™4 laptop. Our algorithm can run 8 frames per second (FPS) even with a Matlab™ implementation. It is anticipated that our algorithm will run faster than real-time (30 FPS) with a C++ implementation.

8 Conclusions

This paper proposes a novel discriminative framework for the image alignment problem. For the *template representation*, given a face dataset with ground truth landmarks, we train a boosting-based classifier that is able to learn the decision boundary between two classes: the warped images from ground truth landmarks and those from perturbed landmarks. The set of trained weak classifiers based on Haar-like rectangular features is considered as an appearance model. For the *distance metric*, we use the score from the strong classifier and treat the image alignment as the process of maximizing the classification score. On the generic face alignment problem, the proposed framework greatly improves the robustness,

accuracy, and efficiency of alignment. We use the term *Boosted Appearance Models (BAM)* to refer the learned shape and appearance models, as well as our specific alignment method.

There are several future directions to extend this framework. First, since this paper opens the door of applying discriminative learning in image alignment, many prior art in pattern recognition, such as other boosting variations or pattern classifiers, can be utilized to replace the GentleBoost algorithm for learning a better appearance model. For example, incremental boosting can be used for adding warped images that are hard to classify into the training data, so as to improve the classification capability of the BAM. Second, more sophisticated optimization methods can be used to maximize the classification score. Finally, as a generic image alignment framework, our proposal does not make use of the domain knowledge of the human faces, except the symmetric rectangular feature type. Hence, this framework can be applied to other image alignment problems, such as medical applications.

ACKNOWLEDGEMENTS

The author would like to thank anonymous reviewers for the constructive comments. This work was supported by awards #2005-IJ-CX-K060, #2006-IJ-CX-K045, and #2007-DE-BX-K191 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.

REFERENCES

[1] 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, Vancouver, Canada*, 1981, pp. 674–679.

[2] I. Matthews and S. Baker, "Active appearance models revisited," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[3] S. Baker, I. Matthews, and J. Schneider, "Automatic construction of active appearance models as an image coding problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1380–1384, 2004.

[4] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[5] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[6] H.-Y. Shum and R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," *Int. J. Computer Vision*, vol. 36, no. 2, pp. 101–130, 2000.

[7] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *IEEE Trans. Medical Imaging*, vol. 21, no. 11, pp. 1374–1383, 2002.

[8] B. Rolfe, M. Cardew-Hall, S. Abdallah, and G. West, "Geometric shape errors in forging: developing a metric and an inverse model," *Proceedings of The Institution of Mechanical Engineers Part B- Journal of Engineering Manufature*, vol. 215, no. 9, pp. 1229–1240, 2001.

[9] T. Cootes, D. Cooper, C. Tylor, and J. Graham, "A trainable method of parametric shape description," in *Proc. 2nd British Machine Vision Conference, Glasgow, UK*, September 1991, pp. 54–61.

[10] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[11] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[12] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 11, pp. 1080–1093, 2005.

[13] X. Liu, P. Tu, and F. Wheeler, "Face model fitting on low resolution images," in *Proc. 17th British Machine Vision Conference, Edinburgh, UK*, vol. 3, 2006, pp. 1079–1088.

[14] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[15] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. 6th Int. Conf. on Computer Vision, Bombay, India*, 1998, pp. 555–562.

[16] S. Yan, C. Liu, S. Z. Li, H. Zhang, H.-Y. Shum, and Q. Cheng, "Face alignment using texture-constrained active shape models," *Image and Vision Computing*, vol. 21, no. 1, pp. 69–75, 2003.

[17] D. Cristinacce and T. Cootes, "Facial feature detection and tracking with automatic template selection," in *Proc. 7th Int. Conf. on Automatic Face and Gesture Recognition, Southampton, UK*, 2006, pp. 429–434.

[18] G. Dedeoglu, T. Kanade, and S. Baker, "The asymmetry of image registration and its application to face tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 807–823, 2007.

[19] R. Beichel, H. Bischof, F. Leberl, and M. Sonka, "Robust active appearance models and their application to medical image analysis," *IEEE Trans. Medical Imaging*, vol. 24, no. 9, pp. 1151–1169, 2005.

[20] E. Jones and S. Soatto, "Layered active appearance models," in *Proc. 10th Int. Conf. on Computer Vision, Beijing, China*, vol. 2, 2005, pp. 1097–1102.

[21] T. Cootes and C. Taylor, "Statistical models of appearance for computer vision," Imaging Science and Biomedical Engineering, University of Mancheste, Tech. Rep., March 2004.

[22] Y. Zhou, L. Gu, and H. Zhang, "Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference," in *Proc. IEEE Computer Vision and Pattern Recognition, Madison, Wisconsin*, vol. 1, 2003, pp. 109–116.

[23] Y. Zhou, W. Zhang, X. Tang, and H. Shum, "A bayesian mixture model for multi-view face alignment," in *Proc. IEEE Computer Vision and Pattern Recognition, San Diego, California*, vol. 2, 2005, pp. 741–746.

[24] L. Liang, F. Wen, Y. Xu, X. Tang, and H. Shum, "Accurate face alignment using shape constrained markov network," in *Proc. IEEE Computer Vision and Pattern Recognition, New York, NY*, vol. 1, 2006, pp. 1313–1319.

[25] D. Cristinacce and T. Cootes, "Boosted regression active shape models," in *Proc. 18th British Machine Vision Conference, University of Warwick, UK*, vol. 2, 2007, pp. 880–889.

[26] A. Batur and M. Hayes, III, "Adaptive active appearance models," *IEEE Trans. Image Processing*, vol. 14, no. 11, pp. 1707–1721, 2005.

[27] C. Butakoff and A. Frangi, "A framework for weighted fusion of multiple statistical models of shape and appearance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1847–1857, 2006.

[28] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, "Fast active appearance model search using canonical correlation analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1690–1694, 2006.

[29] A. Kanaujia and D. Metaxas, "Large scale learning of active shape models," in *Proc. 2007 International Conference on Image Processing, San Antonio, Texas*, vol. 1, 2007, pp. 265–268.

[30] J. Tu, Z. Zhang, Z. Zeng, and T. Huang, "Face localization via hierarchical condensation with fisher boosting feature selection," in *Proc. IEEE Computer Vision and Pattern Recognition, Washington, DC*, vol. 2, 2004, pp. 719–724.

[31] F. D. la Torre Frade, A. C. Romea, J. Cohn, and T. Kanade, "Filtered component analysis to increase robustness to local minima in appearance models," in *Proc. IEEE Computer Vision and Pattern Recognition, Minneapolis, Minnesota*, 2007.

[32] F. Jiao, S. Li, H.-Y. Shum, and D. Schuurmans, "Face alignment using statistical models and wavelet features," in *Proc. IEEE Computer Vision and Pattern Recognition, Madison, Wisconsin*, vol. 1, 2003, pp. 321–327.

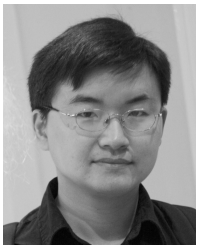
[33] C. Hu, R. Feris, and M. Turk, "Active wavelet networks for face alignment," in *Proc. 14th British Machine Vision Conference, Norwich, UK*, 2003.

[34] S. Avidan, "Support vector tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[35] O. Williams, A. Blake, and R. Cipolla, "Sparse Bayesian learning for efficient visual tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1292–1304, 2005.

[36] A. Hidaka, K. Nishida, and T. Kurita, "Face tracking by maximizing classification score of face detector based on rectangle features," in *Proc.*

IEEE International Conference on Computer Vision Systems, New York, NY, 2006, p. 48.
[37] K. Chang, K. Bowyer, and P. Flynn, "Face recognition using 2D and 3D facial data," in *Proc. ACM Workshop on Multimodal User Authentication*, December 2003, pp. 25–32.
[38] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
[39] N. Dalal and W. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Computer Vision and Pattern Recognition, San Diego, California*, vol. 1, 2005, pp. 886–893.
[40] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[41] 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*, vol. 55, no. 1, pp. 119–139, 1997.
[42] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 38, no. 2, pp. 337–374, 2000.
[43] R. Meir and G. Raetsch., *An introduction to boosting and leveraging*, ser. S. Mendelson and A. Smola, Editors, Advanced Lectures on Machine Learning, LNAI 2600. Springer, 2003.
[44] 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, Madgeburg, Germany*, 2003, pp. 297–304.
[45] Y. Li and W. Ito, "Shape parameter optimization for adaboosted active shape model," in *Proc. 10th Int. Conf. on Computer Vision, Beijing, China*, vol. 1, 2005, pp. 251–258.
[46] L. Gu, E. Xing, and T. Kanade, "Learning GMRF structures for spatial priors," in *Proc. IEEE Computer Vision and Pattern Recognition, Minneapolis, Minnesota*, 2007.
[47] S. Lucey, A. B. Ashraf, and J. Cohn, "Investigating spontaneous facial action recognition through AAM representations of the face," in *Face Recognition Book*, K. Kurihara, Ed. Mammendorf, Germany: Pro Literatur Verlag, April 2007.
[48] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
[49] M. B. Stegmann, B. K. Ersboll, and R. Larsen., "FAME - A flexible appearance modeling environment," *IEEE Trans. Medical Imaging*, vol. 22, no. 10, pp. 1319–1331, 2003.
[50] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *3rd International Conference on Audio- and Video-Based Biometric Person Authentication, Halmstad, Sweden*, 2001, pp. 90–95.



Xiaoming Liu received the BE degree from Beijing Information Technology Institute, China and the ME degree from Zhejiang University, China, in 1997 and 2000 respectively, both in Computer Science, and the PhD degree in Electrical and Computer Engineering from Carnegie Mellon University in 2004. His main research areas are human face recognition, biometrics, human computer interface, object tracking/recognition, online learning, computer vision and pattern recognition. Since 2004, he is a senior research scientist at the Visualization and Computer Vision Laboratory of GE Global Research, and is currently leading a number of facial image processing projects. He is the PI for NIJ "Site-Adaptive Face Recognition at a Distance"(2007-DE-BX-K191) program. He authored more than 40 technical publications, and has over 10 patents pending. He is a member of the IEEE and the Sigma Xi.