Face Alignment via Boosted Ranking Model *

Hao Wu
Center for Automation Research
University of Maryland, College Park, MD 20742
wh2003@cfar.umd.edu

Xiaoming Liu Gianfranco Doretto
Visualization and Computer Vision Lab
GE Global Research, Niskayuna, NY 12309
{liux,doretto}@research.ge.com

Abstract

Face alignment seeks to deform a face model to match it with the features of the image of a face by optimizing an appropriate cost function. We propose a new face model that is aligned by maximizing a score function, which we learn from training data, and that we impose to be concave. We show that this problem can be reduced to learning a classifier that is able to say whether or not by switching from one alignment to a new one, the model is approaching the correct fitting. This relates to the ranking problem where a number of instances need to be ordered. For training the model, we propose to extend GentleBoost [23] to rank-learning. Extensive experimentation shows the superiority of this approach to other learning paradigms, and demonstrates that this model exceeds the alignment performance of the state-of-the-art.

1. Introduction

Face alignment/fitting is essentially an image registration problem, where a face model needs to be deformed to match the image of a face, so that the natural facial features are aligned with the model. The dramatic variations of facial appearance due to shape, pose, illumination, expression, occlusions, and image resolution make this a challenging problem. Due to its importance in a wide range of applications, there is a sizable literature on face alignment. The Active Shape Model (ASM) [6] is one of the early approaches that attempts to fit the data with a model that can deform in ways consistent with a training set. The Active Appearance Model (AAM) [2, 5] is a popular extension of the ASM. During a training phase, the AAM learns from labeled data the statistical *generative models* for the *shape* of a face (represented by landmark positions, see Figure 1(a)), and for the *appearance* of a face (represented by pixel intensities in the shape-normalized domain). During the fitting phase, the AAM is aligned in such a way that the data can be best explained (or reproduced) by the model in the least mean square error sense. It is known that these models perform well if trained to work with a limited number of known subjects. On the other hand, the alignment performance degrades quickly if either the AAM is trained on a large dataset [21], or it is fitted to unseen subjects, or both [15].

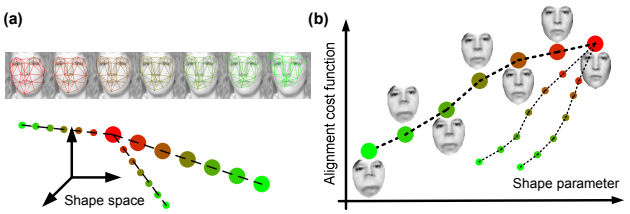

Figure 1. **Image Alignment via Ranking Function Learning.** (a) Images of a face with superimposed *shape* (landmarks) of a *face model* as it deforms away from the correct alignment (from red to green). (b) We propose to learn an *alignment score function* with *concave* properties while the shape undergoes the above deformation.

In order to tackle this generalization problem, the recently proposed Boosted Appearance Model (BAM) [20] uses a shape representation similar to the AAM, whereas the appearance is given by a set of discriminative features, trained to form a boosted classifier, able to distinguish between *correct* and *incorrect* alignment. Fitting the BAM amounts to updating the landmark positions according to gradient ascent on the corresponding classifier score function. Although it has been shown that the BAM improves the generalization capabilities of the AAM, we point out that since the score function has been learned to distinguish between correct and incorrect alignment, there is no guarantee that moving along its gradient will always improve the alignment, to a great detriment of the generalization capabilities.

In order to address this limitation we propose the *Boosted Ranking Model* (BRM), which has similar representations for shape and appearance, but arises from a very different formulation of the alignment problem. In particular, *we propose to learn from data an alignment score func-*

*Part of this work was done while the first author was visiting GE Global Research as an intern. This work was supported by the National Institute of Justice, US Department of Justice, under the award #2006-IJ-CX-K045. 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.

tion that is concave *within the neighborhood of the correct alignment*. In this way we assure that by updating the alignment of the BRM via gradient ascent, we will always deform the model towards the correct fitting (see Figure 1(b)).

First, we will show that the original problem can be approximated by the problem of learning a classifier that is able to say whether or not by switching from one alignment to another one, the BRM moves closer to the correct solution. Then, we train a boosted classifier, which when given a pair of images warped from different landmarks, informs which of the two corresponds to a better alignment. This naturally leads to the creation of a positive training set and a negative training set with the same cardinality, making the learning problem balanced. The particular structure of the resulting classifier allows to map the original problem to a *ranking problem* [12], because it implies the learning of a function, i.e. the alignment score function, which can be interpreted as a *ranking function* [12] (able to order instances corresponding to different degrees of alignment of the BRM), and which is meant to be concave. We propose to learn the alignment score function by extending the use of GentleBoost [23] for ranking, and show experimentally that it converges faster, and performs better alignment ranking than other approaches, such as RankBoost [12]. Finally, we show that the BRM learns an alignment function that is concave, and that has better generalization capabilities than the BAM, especially in terms of robustness in achieving convergence, but also in terms of accuracy, and computational speed.

2. Prior Art

The majority of the prior work in face alignment is based on ASM, AAM or their variations [7–10, 17, 26, 30]. In particular, in [7] the ASM incorporates a generative template model for each landmark, whereas in [8] the same model is discriminative. Most of the AAM body of work is based on the generative model of [2], which greatly improves the efficiency of the AAM-based face alignment. Some AAM variations include discriminative fitting methods [10, 26]. Other representative works include [18, 34].

In the problem of ranking, the goal is to learn an ordering or ranking over objects. The wide variety of applications in which ranking is required includes, information retrieval, collaborative filtering, computational biology, econometrics, and social sciences. As relevant references, we mention [16] that proposes an SVM-based ranking method to improve search engines, and [12] that proposes RankBoost for collaborative filtering.

In the Computer Vision community, [1, 14] have utilized ranking algorithms for shape and image retrieval. In [32] Constrained RankBoost is proposed to model the likelihood of local features associated to the landmarks of a face model. On the other hand, our approach allows boosting to optimally chose the position of the local features. In [33] RankBoost learning is used to provide a relative similarity measure between a given shape and a reference shape. This is used to rank a fixed number of predefined warpings of an image, and then combine the first few top ranked to perform shape detection. In contrast, we define and train a model for the shape variability, which we use to optimize the learned alignment score function.

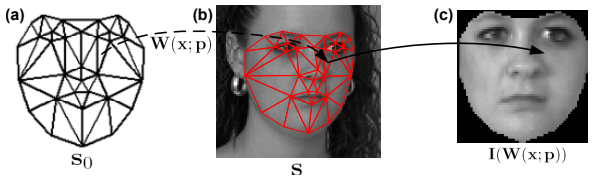

Figure 2. **Shape Model and Warping Function.** (a) Representation of the mean shape. (b) The face image with a superimposed shape. (c) The face image warped to the mean shape domain.

3. Face Model

Unlike AAM's, where a face model is represented by the combination of two generative models, one for the shape and one for the appearance of a face, we use a generative model for the shape, whereas we represent the appearance with a set of discriminative features, which will be automatically selected for the purpose of solving the face alignment problem (described in Section 4). In this section we will introduce the representation of these two model components.

3.1. Shape Model

The shape of a face is represented by a set of l 2D-landmarks, defined by their image coordinates $\{\mathbf{x}_i = (x_i, y_i)\}_{i=1,\cdots,l}$, which we stack with a predefined order to form the *shape* vector $\mathbf{s} \doteq [x_1, y_1, x_2, y_2, ..., x_l, y_l]^T$. We represent the statistical variability of $\mathbf{s}$ with an affine variety, which means that

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

where $\mathbf{s}_0$ is the *mean shape*, $\mathbf{s}_i$ is the i-th *shape basis*, and $\mathbf{p} = [p_1, p_2, ..., p_n]^T$ is the *shape parameter*. The mean shape and the shape basis can be learned from a labeled training set of face images via Principal Component Analysis (PCA). Analogous models for shape representation have been used before [5, 11, 20, 22].

The mean shape $\mathbf{s}_0$ (Figure 2(a)), and the shape $\mathbf{s}$ (Figure 2(b)), define 2-dimensional domains which can be related by a piecewise affine warping function $\mathbf{W}(\mathbf{x}; \mathbf{p})$ that maps points from the mean shape domain into the face image domain[1]. With this warping function, a face image $\mathbf{I}(\mathbf{x})$ can be warped to the mean shape domain, obtaining $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ (Figure 2(c)), where a shape-normalized face appearance model can be computed .

[1] See [22] for a description on how to design and parameterize $\mathbf{W}$.

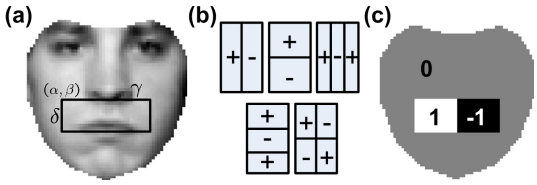

Figure 3. **Appearance Features.** (a) Warped face image with feature parametrization. (b) Representation of the five feature types used by the appearance model. (c) Notional template $\mathbf{A}$.

3.2. Appearance Model

The appearance model is simply a collection of m features $\{\varphi_i\}_{i=1,\cdots,m}$, computed over the shape-normalized face image $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}))$. As features we choose the popular rectangular Haar-like features [24, 29], mainly because of their computational efficiency, which exploits the integral image representation [29], and because of their success in face-related applications [20, 29].

A rectangular feature can be parameterized as follows

$$\varphi \doteq \mathbf{A}^T \mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \;, \tag{2}$$

which is intended as the inner product between the vectorized version of an image template $\mathbf{A}$ (Figure 3(c)), and the vectorized version of the warped face image (Figure 3(a)). The inner product between the template and the warped image is equivalent to computing the rectangular feature using the integral image. The image template $\mathbf{A}$ can in turn be parameterized by $(\alpha, \beta, \gamma, \delta, \tau)$, as shown in Figure 3(a), where (α, β) is the top-left corner, γ and δ are the width and height, and τ is the feature type. Figure 3(b) shows the feature types used in our model.

4. Alignment Learning Problem

In this section we formulate the problem of learning an alignment score function that will then be used in Section 7 to perform the fitting of the face model. More precisely, for a given image, let us suppose that $\mathbf{p}$ is the shape parameter that represents the current alignment of the shape model (1), with the face in the image: *We are interested in learning from data a score function F, such that, when maximized with respect to $\mathbf{p}$, it will return the shape parameter corresponding to the correct alignment.* Mathematically, if $\mathbf{p}_0$ is the shape parameter representing the correct alignment, F has to be such that

$$\mathbf{p}_0 = \arg\max_{\mathbf{p}} F(\mathbf{p}) \;. \tag{3}$$

Our program is to optimize F via *gradient ascent*. Therefore, to avoid local maxima, we would like F to be *concave* on $B(\mathbf{p}_0)$, a convex neighborhood around $\mathbf{p}_0$. This means that for all $\mathbf{p}_1, \mathbf{p}_2 \in B(\mathbf{p}_0)$, assuming that F is differentiable, the following should hold [3]:

$$F(\mathbf{p}_2) > F(\mathbf{p}_1) \Longrightarrow \nabla F(\mathbf{p}_1)^T (\mathbf{p}_2 - \mathbf{p}_1) > 0 \;. \tag{4}$$

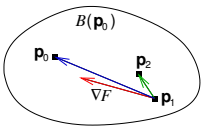

Figure 4. **Convex Neighborhood.** Representation of the convex neighborhood $B(\mathbf{p}_0)$, around the maximum $\mathbf{p}_0$ of the alignment score function F. If $\|\mathbf{p}_2 - \mathbf{p}_1\| \ll \|\mathbf{p}_0 - \mathbf{p}_1\|$ then $\nabla F(\mathbf{p}_1)$ is almost proportional to $\mathbf{p}_0 - \mathbf{p}_1$, as well as to $\mathbf{p}_0 - \mathbf{p}_2$.

4.1. Alignment as a Classification Problem

In this section we propose to reduce the problem of learning the function F to the problem of learning a strong classifier. We start by performing an approximation of the condition (4) that is valid when $\mathbf{p}_2$ corresponds to a small perturbation of $\mathbf{p}_1$, which means[2] that $\rho \doteq \frac{\|\mathbf{p}_2 - \mathbf{p}_1\|}{\|\mathbf{p}_0 - \mathbf{p}_1\|} \ll 1$. We do so because by solving (3) via gradient ascent, we compute small updates of the shape parameter $\mathbf{p}$. Under this assumption it is reasonable to assume that $\nabla F(\mathbf{p}_1)$ is almost proportional to $\mathbf{p}_0 - \mathbf{p}_1$, as well as to $\mathbf{p}_0 - \mathbf{p}_2$ (see Figure 4). Therefore, condition (4) can be approximated as

$$F(\mathbf{p}_2) > F(\mathbf{p}_1) \Longrightarrow \|\mathbf{p}_2 - \mathbf{p}_0\| < \|\mathbf{p}_1 - \mathbf{p}_0\| \;. \tag{5}$$

It is obvious that (5) states that if $F(\mathbf{p}_2) > F(\mathbf{p}_1)$, then by moving from $\mathbf{p}_1$ to $\mathbf{p}_2$ the alignment improves in the Euclidean sense.

Equation (5) suggests that the function F could be used to solve a classification problem. More precisely, if we define a classifier $H(\mathbf{p}_1, \mathbf{p}_2) \doteq \text{sign}[F(\mathbf{p}_2) - F(\mathbf{p}_1)]$, then

$$H(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} +1 & \Longrightarrow \|\mathbf{p}_2 - \mathbf{p}_0\| < \|\mathbf{p}_1 - \mathbf{p}_0\| \;, \\ -1 & \Longrightarrow \|\mathbf{p}_2 - \mathbf{p}_0\| \geq \|\mathbf{p}_1 - \mathbf{p}_0\| \;, \end{cases} \tag{6}$$

and H informs whether or not (i.e. ± 1) switching from $\mathbf{p}_1$ to $\mathbf{p}_2$ constitutes an alignment improvement.

It becomes natural at this stage to view the problem of learning F as the problem of learning the classifier H, which can be seen as the strong classifier output by a boosting procedure. More precisely, we can assume H to be the sign of the additive model

$$\sum_{i=1}^{m} h_i(\mathbf{p}_1, \mathbf{p}_2) \;, \tag{7}$$

where each h_i is a week classifier. Note that the structure of H suggests a structure for the $\{h_i\}$, and the generic weak classifier will be given by

$$h_i(\mathbf{p}_1, \mathbf{p}_2) = f_i(\mathbf{p}_2) - f_i(\mathbf{p}_1) \;, \tag{8}$$

where the $\{f_i\}$ are such that F, in the neighborhood $B(\mathbf{p}_c)$, will be given by the following additive model

$$F(\mathbf{p}) \doteq \sum_{i=1}^{m} f_i(\mathbf{p}) \;. \tag{9}$$

[2] The symbol $\|\cdot\|$ denotes the Euclidean norm.

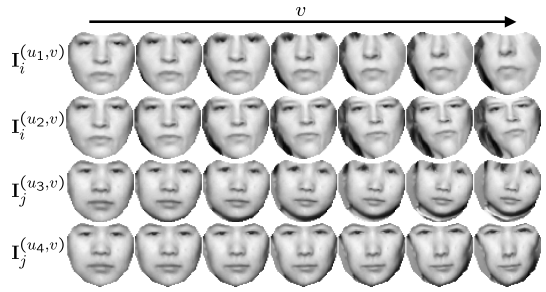

Figure 5. **Training Samples.** Top two rows and bottom two rows are training samples generated from the same face image ($\mathbf{I}_i$ and $\mathbf{I}_j$ respectively). Samples from each row have been generated from the original shape-normalized face image on the left, and with shape-perturbation parameters u_1, u_2, u_3, u_4. From left to right the parameter v increases, and the shape parameter is varying according to Equation (10).

4.2. Positive and Negative Training Sets

In this section we will describe how to build the class of positive and negative samples used to train the classifier H. We start by considering a face dataset made of N face images. Each image $\mathbf{I}_i$ has been manually labeled with landmarks $\mathbf{s}_i$, and the corresponding ground truth shape parameter $\mathbf{p}_i$ can be computed according to Equation (1). From this dataset we produce a number of shape-normalized images. More precisely, we uniformly draw a set of U shape-perturbation parameters $\{\Delta\mathbf{p}_u \,|\, \|\Delta\mathbf{p}_u\| = \rho\}$, with which we compute a set of shape parameters

$$\{\mathbf{p}_i + v\Delta\mathbf{p}_u\}_{i=1,\cdots,N;\, u=1,\cdots,U;\, v=0,\cdots,V} \tag{10}$$

that we use to generate the training samples $\{\mathbf{I}_i^{(u,v)}\}$, such that $\mathbf{I}_i^{(u,v)} \doteq \mathbf{I}_i(\mathbf{W}(\mathbf{x}; \mathbf{p}_i + v\Delta\mathbf{p}_u))$ (see Figure 5). Then, a positive sample is defined as the ordered pair $x_{+iuv} = (\mathbf{I}_i^{(u,v+1)}, \mathbf{I}_i^{(u,v)})$, and will be labeled with $y_{+iuv} = +1$. Similarly, a negative sample is defined as the ordered pair $x_{-iuv} = (\mathbf{I}_i^{(u,v)}, \mathbf{I}_i^{(u,v+1)})$, and will be labeled with $y_{-iuv} = -1$. Therefore, the training sets of positive and negative samples, $\mathfrak{P}$ and $\mathfrak{N}$ respectively, are given by

$$\begin{aligned} \mathfrak{P} &\doteq \{x_{+iuv}\}_{i=1,\cdots,N;\, u=1,\cdots,U;\, v=0,\cdots,V-1} \ , \\ \mathfrak{N} &\doteq \{x_{-iuv}\}_{i=1,\cdots,N;\, u=1,\cdots,U;\, v=0,\cdots,V-1} \ . \end{aligned} \tag{11}$$

The reader may notice that the sets $\mathfrak{P}$ and $\mathfrak{N}$ have the same cardinality, which means that, in this approach, achieving the right balance between the representations of the null and the alternate hypothesis is not an issue!

4.3. Training the Weak Classifiers

In this section we define the weak classifiers that we intend to use, and describe how to learn them. In order to define h_i, Equation (8) shows that we only need to define f_i, where $f_i(\mathbf{p}_1)$ operates on a pool of features computed on $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}_1))$, and $f_i(\mathbf{p}_2)$ operates on the same pool of features, but computed on $\mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p}_2))$. In particular, we assume that the pool is made only by one feature, φ_i. Since we require F to be a differentiable function, it follows from (9) that f_i has to be differentiable. Finally, we assume that f_i is going to perform a comparison of the feature φ_i against a threshold t_i. By taking into account all of the above, we define f_i as

$$f_i(\mathbf{p}) \doteq \frac{1}{\pi} \arctan(g_i \varphi_i(\mathbf{p}) - t_i) \ , \tag{13}$$

where $g_i = \pm 1$, and the normalizing constant ensures that h_i stays within the range of $[-1, 1]$.

In order to learn the weak classifiers we use a boosting procedure called GentleBoost [23]. Compared to AdaBoost [13], it is numerically more robust, has been shown experimentally that has better convergence properties, and performs well on several face-related applications [19, 20, 28,31]. In this specific case, it allows for sequentially fitting additive models of the form of (7), where the week classifiers are smooth sigmoid functions ranging from -1 to $+1$.

Algorithm 1, given above, describes the GentleBoost procedure for learning the alignment score function (9). Note that step 4 is computationally the most intensive, as the entire feature hypothesis space is exhaustively searched. Also note that, since $h(x_{+iuv}) = -h(x_{-iuv})$, the score function in Equation (12) could be simplified to $\sum_{iuv} w_{+iuv} \left(1 - h(x_{+iuv})\right)^2$.

Ultimately, learning the score function F amounts to learning the set of features $\{\varphi_i\}$, the thresholds $\{t_i\}$, and the feature signs $\{g_i\}$. Suppose that the mean of the feature φ_i computed over the positive samples is greater than the mean computed over the negative samples, then we set $g_i = +1$, otherwise we set $g_i = -1$. The final set of triples $\{(\varphi_i, g_i, t_i)\}_{i=1,\cdots,m}$, together with the shape model $\{\mathbf{s}_i\}_{i=0,\cdots,n}$ is called a *Boosted Ranking Model* (BRM). Figure 6 shows the top 15 features selected by the learning algorithm, as well as the spatial density map of the top 50 features. The reader may notice that most of the features are aligned with the boundaries of the natural facial features.

Algorithm 1: Alignment Score Function Learning

Data: Positive and negative samples $\mathfrak{P}$, and $\mathfrak{N}$ from Equation (11), with labels $\{y_{siuv}\}_{s=\pm,i=1,\cdots,N;\, u=1,\cdots,U;\, v=0,\cdots,V-1}$
Result: The alignment score function F

1 Initialize the weights $w_{siuv} = \frac{1}{2NUV}$
2 Initialize the score function $F = 0$
3 **foreach** $j = 1, \cdots, m$ **do**
4 Fit f_j in the weighted least squares sense, such that

$$f_j = \underset{f}{\operatorname{argmin}} \sum_{siuv} w_{siuv} \left(y_{siuv} - h(x_{siuv})\right)^2 \tag{12}$$

where $h(x_{siuv}) = f(\mathbf{I}_i^{(u,v+\frac{1-s}{2})}) - f(\mathbf{I}_i^{(u,v+\frac{1+s}{2})})$
5 $F \longleftarrow F + f_j$
6 $w_{siuv} \longleftarrow w_{siuv} e^{-y_{siuv} h_j(x_{siuv})}$
7 Normalize the weights such that $\sum_{siuv} w_{siuv} = 1$
8 **return** $F = \sum_{j=1}^{m} f_j$

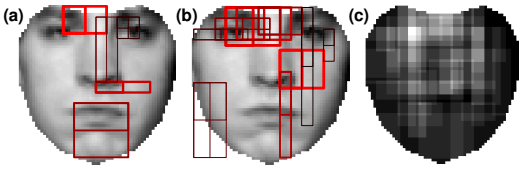

Figure 6. **Selected Appearance Features.** (a) Representation of the top 5 Haar features selected by Algorithm 1. (b) Representation of the top 6-15 Haar features. (c) Spatial density map of the top 50 Haar features. Most features are well aligned with the boundaries of the natural facial features.

5. Comparison with Other Models

The BRM belongs to the same class of models as the BAM model [20]. Therefore, compared to AAM [5, 22], it enjoys the same benefits, such as robustness to partial occlusions, improved alignment speed, and ability to incorporate knowledge about both good and bad alignment, while being substantially more parsimonious.

In comparison with BAM, our model is the outcome of a very different problem formulation. More precisely, the BAM is produced by learning a strong classifier that is able to distinguish between correct and incorrect alignment, and the results in [20] empirically show that face alignment can be achieved via gradient ascent on the corresponding classifier score function. However, there is no guarantee that the gradient will be aimed at improving the alignment (because the strong classifier can distinguish only between right or wrong fittings). On the other hand, we consider this fundamental issue at the outset, and propose to solve the alignment problem by looking for a score function that is concave, hence optimizable via gradient ascent. This leads to learning a strong classifier that is able to say whether by switching from one alignment to another one we are actually making an improvement, as opposed to saying whether or not the alignment is correct. Another advantage (as opposed to the BAM), is also the fact that positive and negative training sets naturally have the same cardinality, which makes the training problem balanced. These advantages lead to a superior alignment performance of the BRM over the BAM, as we will show in Section 8.

6. Relation with Ranking

Given a set of *instances* that we call *instance space*, the *ranking problem* is to design a *ranking* function that is able to tell whether one instance should be ranked higher than another one, and therefore it can produce a linear ordering of the instances. RankBoost [12] is an algorithm that, from information about the relative ranking of individual pair of instances, learns a ranking function by combining a number of *weak ranking* functions selected in a greedy fashion. Roughly speaking, the algorithm is a direct extension of AdaBoost [13] in that the ranking function is the byproduct of learning a classifier (the so called *feedback function*), which says whether a pair of instances appear to be ranked in ascending or descending order. Therefore, this process minimizes the weighted number of incorrectly ranked pairs, as opposed to AdaBoost that minimizes the weighted number of misclassifications.

In Section 4 we have shown that starting from concave optimization arguments, we can reduce the initial learning problem to learning a classifier that says whether moving from a shape parameter to another one corresponds to an alignment improvement. This is analogous to the ranking problem. In fact, H and $\{h_i\}$ are essentially *strong* and *weak* feedback functions, whereas $\{f_i\}$ and $\{\varphi_i\}$ are the so-called *weak rankings* and *ranking features*. F instead is the ranking function. For the reasons highlighted in Section 4, we learn the weak feedback functions using GentleBoost, and this has naturally lead to an extension of this algorithm to solving the ranking problem, in the same way as RankBoost is an extension of AdaBoost.

The proposed learning algorithm has two other distinctive aspects. The traditional ranking problem labels every possible ordered pair of training samples. On the other hand, we label only pairs differing by one shape-perturbation parameter step. We do so because the alignment will be computed via gradient ascent by making small shape parameter updates, hopefully in the direction of $\mathbf{p}_0 - \mathbf{p}_1$. Therefore, there is no need to learn the feedback function when $\mathbf{p}_1$ and $\mathbf{p}_2$ are very far apart. If we were doing so, the BRM learning would become more difficult, because it would need a much larger training set, and because the BRM would be forced to learn something unnecessary. Finally, we highlight the fact that we want to learn a smooth score function, inherently defined on a continuous domain, which we discretize to make the problem tractable. This is in contrast with traditional ranking problems, which are defined on a discrete instance space.

We have experimented with RankBoost while keeping the same ranking features and weak rankings, and found out that our proposed extension of GentleBoost shows better convergence properties, and performs better in the classification test of the synthesized pairs (see Section 8).

7. Face Alignment

In order to align a BRM with the face in a given image $\mathbf{I}$, we assume that the model is currently aligned with a shape parameter $\mathbf{p}^{(i)}$ (at the i-th iteration). As explained in Section 4, in order to achieve the optimal alignment one may perform a simple gradient ascent on the score function F, and therefore update the shape parameter as follows

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} + \nu \frac{\partial F}{\partial \mathbf{p}} \,, \tag{14}$$

where ν is a suitable constant. Figure 7 shows a few face images with the face model corresponding to the initial shape parameter and the shape parameter at convergence.

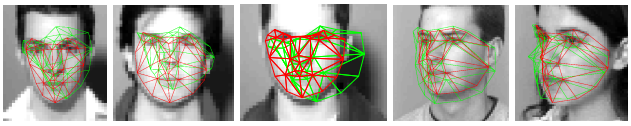

Figure 7. **Alignment Examples.** Face images with superimposed initial face model (green), and aligned face model (red).

From (2), (9), and (13) one can see that the derivative of F with respect to $\mathbf{p}$ is

$$\frac{\partial F}{\partial \mathbf{p}} = \frac{1}{\pi}\sum_{i=1}^{m} \frac{g_i \left(\nabla \mathbf{I} \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right)^T \mathbf{A}_i}{1 + \left(g_i \mathbf{A}_i^T \mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) - t_i\right)^2}, \qquad (15)$$

where $\nabla \mathbf{I}$ is the gradient of the image 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 interested reader is referred to [20] for a deeper discussion on the alignment procedure, and the computational complexity, and efficient implementation of $\partial F/\partial \mathbf{p}$.

8. Experiments

Face dataset. We start by describing the dataset used for training and testing our proposed approach. It is composed of a total of 964 images coming from the aggregation of three publicly available datasets: ND1 [4] (534 images of 200 subjects appearing in frontal view), FERET [25] (200 images of 200 subjects appearing in different pose), and BioID [27] (230 images of 23 subjects appearing under different background and lighting conditions). Figure 8 shows some typical face images from the datasets. Each image has 33 manually labeled landmarks. To speed up the training process, we down-sample each image so that the face width is roughly 40 pixels. We divide the 964 images in three parts, namely Set 1, Set 2, and Set 3. Set 1 contains the 200 images from FERET, and 200 images from ND1 (one image per subject). Set 2 contains the remaining 334 images from ND1. Set 3 is the BioID dataset. Set 1 is used as training dataset. All the three sets are used in the alignment tests. In particular, Set 2 allows for testing the performance over unseen data of seen subjects (because different images of them have been used for training), whereas Set 3 allows for testing the performance over unseen data of unseen subjects (never used for training). Note that Set 3 is particularly challenging because the subjects are captured under different cluttered background, and illumination.

Training. Throughout the section we compare three models: the proposed BRM, BAM, and an adaptation of RankBoost [12] that uses the same weak rankings, and training pairs of the BRM. We do not compare our model against AAM-based methods [5, 22], as it has been shown in [20] that the BAM outperforms them. We train the three models with Set 1, which originates the training samples $\{\mathbf{I}_i^{(u,v)}\}$, where $i = 1, \cdots, 400$, $u = 1, \cdots, 10$ and $v = 0, \cdots, 6$, corresponding to 24000 positive (and also negative) training pairs. In contrast, the BAM uses 400 positive and 4000 negative samples, since each image generates 10 negative samples. The resulting appearance models are such that the BRM and RankBoost have 50 weak rankings, whereas the BAM has 50 weak classifiers. The shape model has 33 shape bases and it is the same for all the models.

Figure 8. **Face Dataset Samples.** ND1 database [4] (left), FERET database [25] (center), and BioID database [27] (right).

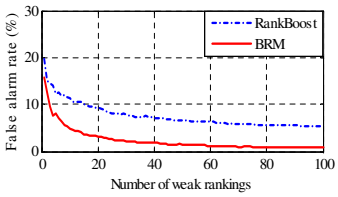

Figure 9. **Feedback Function Performance.** False alarm rate of the strong feedback function when the miss-detection rate on the training set is set to 0%.

Convergence properties. Figure 9 plots the false alarm rate (FAR) of the strong feedback functions of both BRM and RankBoost, as function of the number of weak rankings, when the miss-detection rate on the training set is set to 0%. This shows that the BRM converges faster than RankBoost. In particular, for 50 weak rankings the FAR's of BRM and RankBoost are 1.44%, and 6.58%, respectively.

Score function concavity. Figure 10(a) plots the learned score (ranking) function F for 3 images, perturbed along 10 different shape-perturbation parameters $\{\Delta \mathbf{p}_u\}$. Figure 10(b) plots the score function for 100 images of Set 1, each of which is perturbed along one shape-perturbation parameter. Both cases highlight the concavity properties of F.

Another way to show the score function is by using grayscale values, as in Figure 11, where each column represents F computed for one image, and each image has been produced by varying the intensity of only two shape bases. The range of perturbation is ± 1.6 times the eigenvalue of the corresponding bases. For both seen data in Figure 11(a), and unseen data in Figure 11(b), F shows concave properties, as required by construction, with the brightest pixel in the center, and intensity fading towards the borders.

Ranking performance. Using the same methodology for building the training sets of pairs, we build two testing sets of pairs, one from Set 1, and one from Set 3, and test the ranking performance of the BRM, BAM, and RankBoost. The correct ranking rates are reported in Figure 12(a), which shows the superiority of the BRM versus the BAM, especially for Set 3, highlighting the stronger generalization capabilities of the BRM to unseen data. Also, BRM performs slightly better than RankBoost on both sets, and

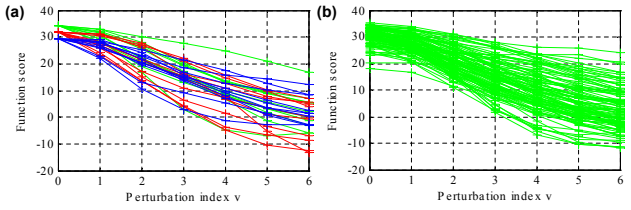

Figure 10. **Alignment Score Function Profile.** (a) Score functions of 3 images, corresponding to 10 shape-perturbation parameters. (b) Score functions of 100 training images, each of which corresponding to one shape-perturbation parameter.

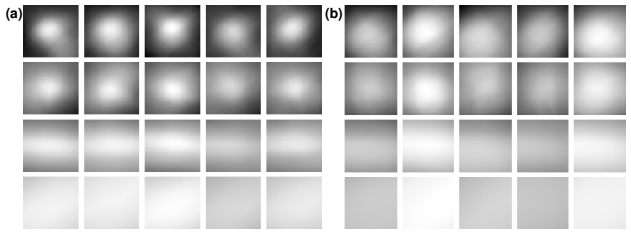

Figure 11. **Alignment Score Function Surface.** Score function F of 5 images randomly selected from Set 1 (a), and 5 images from Set 2 (b), one per column. Each image is produced by varying the shape parameter corresponding to two shape bases at a time. From the top to the bottom rows we vary: (p_1, p_2), (p_3, p_4), (p_5, p_6), and (p_7, p_8). F is concave in both seen and unseen data, and this ensures high frequency of convergence of the alignment.

therefore it is expected to achieve better alignment performance as well. Figure 12(b) shows that BRM outperforms the BAM also in a much harder scenario, where testing pairs are built from Set 3, but with half, and one quarter of the perturbation used to produce Figure 12(a). The reader may notice the slight ranking performance drop of both methods as the perturbation becomes smaller, because it makes the ranking task more difficult.

Alignment performance. In order to evaluate the alignment quality of a modeling framework, we randomly perturb the ground truth landmarks of a face image, and use them as initial condition to align the model. The procedure is repeated multiple times on each image of the testing set in order to perform a statistical evaluation of the result. The initial position of the landmarks is generated by perturbing the components $\{p_i\}$ of the shape parameter with independent Gaussian noise with variances multiple of the eigenvalues of the corresponding shape bases. An alignment is claimed as converged if the Root Mean Square Error (RMSE) between the aligned landmarks and the ground truth is less than one pixel. Finally, we assess the alignment robustness and accuracy by computing: (a) the Average Frequency of Convergence (AFC), given by the number of trials where the alignment converges divided by the total number of trials; and (b) the histogram of the RMSE (HRMSE) of the converged trials, which measures how close the aligned landmarks are to the ground truth.

We test BRM and BAM under the same conditions. For example, both algorithms are initialized with the same set

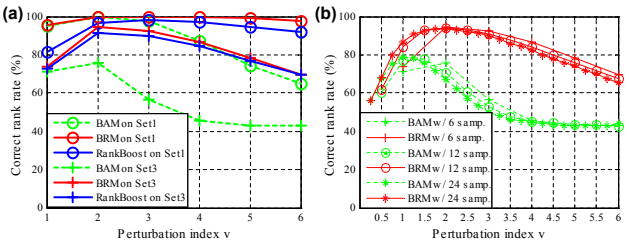

Figure 12. **Ranking Performance.** (a) Correct ranking rates of the BRM, BAM, and RankBoost on test pairs from Set 1 and Set 3. (b) Correct ranking rates of the BRM and BAM on test pairs sampled from Set 3, but with half (12 samples) and one quarter (24 samples) of the perturbation used in (a).

of randomly perturbed landmarks. Both algorithms have the same constant ν in Equation (14), and also the same termination condition. That is, if the number of iterations is larger than 55 or the RMSE between consecutive iterations is less than 0.025 pixels. Figures 13(a)(c)(e) plot the AFC of the BRM and BAM against the amount of the initial landmarks perturbation, computed over Set 1, Set 2, and Set 3, respectively. In particular, for each perturbation value, each image of each set is randomly perturbed 5, 6, or 9 times depending on whether it belongs to Set 1, Set 2, or Set 3, respectively.

The AFC plots in Figure 13 show that BRM-based alignment is substantially more robust than BAM-based alignment for both seen and unseen data. In contrast, the accuracy improvement of the BRM over the BAM, demonstrated by HRMSE, is not as large as the AFC melioration. For example, on Set 3 the average ($\pm$ the standard deviation) BRM-RMSE is 0.5745 ± 0.1725, whereas the average BAM-RMSE is 0.6533 ± 0.1594. This means that, when approaching convergence, BAM and BRM have comparable ability to rank pairs. This aspect is confirmed also by the left-most plot of Figure 12(b).

Speed. When computing Figure 13(c) on a low-end PC, we recorded the time and number of iterations taken by our Matlab^{TM} implementation of the BAM, and of the BRM, to converge. When both algorithms converge, the BAM takes an average of 8.06 iterations, and 0.122 seconds, whereas the BRM takes an average of 7.4 iterations, and 0.112 seconds. We attribute this improvement to the superior property of the ranking function of the BRM, compared to the classifier function of the BAM.

9. Conclusions

We have introduced the Boosted Ranking Model (BRM), a new discriminative face model suitable to perform face alignment. The BRM is associated to a score function learned from data, which is meant to be concave to ensure that fitting can be achieved via gradient ascent. Learning a BRM corresponds to training a boosted classifier with a particular structure, that makes it equivalent to learning a boosted ranking function. This is done by extending Gen-

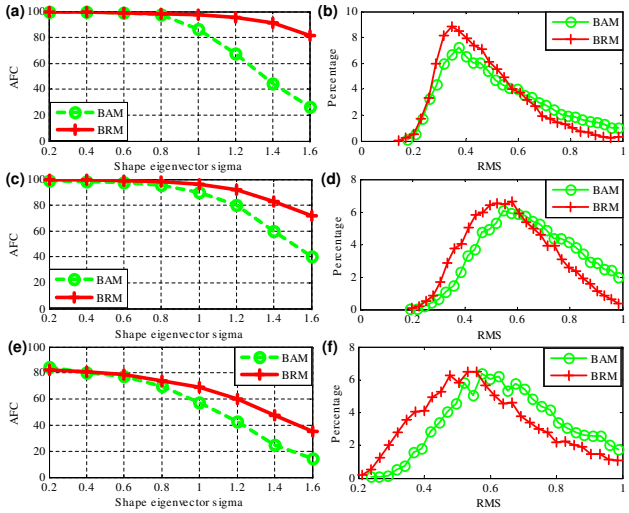

Figure 13. **Alignment Performance.** From top to bottom, AFC and HRMSE of both BAM and BRM computed on Set 1, Set 2, and Set 3, respectively. The HRMSE is computed on the trials where both algorithms converge.

tleBoost to rank-learning, which we found to work better than other methods. The BRM outperforms the BAM for both seen and unseen subjects, especially in terms of alignment robustness (due to the concave properties of the score function), while slightly improving the accuracy and computational speed. This is a parsimonious model (especially if compared with the AAM), with enhanced generalization properties, that holds the promise of fitting multiple face models to new subjects in real-time.

Our approach is not bounded to work with faces, and it could be extended to work with other objects of interest. Moreover, the idea of building a concave function through rank-learning could be applied to other vision problems, such as discriminative object tracking, which could greatly benefit from a smooth and concave tracking score function.

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. BoostMap: A method for efficient approximate similarity rankings. In *CVPR*, volume 2, pages 268–275, 2004.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 years on: a unifying framework. *IJCV*, 56(3):221–255, 2004.

[3] S. Boyg and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[4] K. Chang, K. Bowyer, and P. Flynn. Face recognition using 2D and 3D facial data. In *Proceedings of ACM Workshop on Multimodal User Authentication*, pages 25–32, 2003.

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

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *BMVC*, pages 9–18, 1992.

[7] D. Cristinacce and T. F. Cootes. Facial feature detection and tracking with automatic template selection. In *FGR*, pages 429–434, 2006.

[8] D. Cristinacce and T. F. Cootes. Boosted regression active shape models. In *BMVC*, volume 2, pages 880–889, 2007.

[9] G. Dedeoglu, T. Kanade, and S. Baker. The asymmetry of image registration and its application to face tracking. *IEEE TPAMI*, 29(5):807–823, 2007.

[10] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast Active Appearance Model search using Canonical Correlation Analysis. *IEEE TPAMI*, 28(10):1690–1694, 2006.

[11] G. Doretto and S. Soatto. Dynamic shape and appearance models. *IEEE TPAMI*, 28(12):2006–2019, 2006.

[12] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[13] 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*, 55:119–139, 1997.

[14] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.

[15] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[16] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of SIGKDD*, pages 133–142, 2002.

[17] A. Kanaujia and D. N. Metaxas. Large scale learning of Active Shape Models. In *ICIP*, volume 1, pages 265–268, 2007.

[18] L. Liang, F. Wen, Y. Q. Xu, X. Tang, and H. Y. Shum. Accurate face alignment using shape constrained Markov network. In *CVPR*, volume 1, pages 1313–1319, 2006.

[19] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM*, pages 297–304, 2003.

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

[21] X. Liu, P. Tu, and F. Wheeler. Face model fitting on low resolution images. In *BMVC*, volume 3, pages 1079–1088, 2006.

[22] I. Matthews and S. Baker. Active Appearance Models revisited. *IJCV*, 60(2):135–164, 2004.

[23] R. Meir and G. Rätsch. An introduction to boosting and leveraging. In *Advanced Lectures on Machine Learning: Machine Learning Summer School 2002*, pages 118–183. Springer, 2004.

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

[25] P. J. Phillips, M. Hyeonjoon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE TPAMI*, 22(10):1090–1104, 2000.

[26] J. Saragih and R. Goecke. A nonlinear discriminative approach to AAM fitting. In *ICCV*, 2007.

[27] M. B. Stegmann, B. K. Ersboll, and R. Larsen. FAME-a flexible appearance modeling environment. *IEEE Trans. on Med. Imag.*, 22(10):1319–1331, 2003.

[28] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE TPAMI*, 29(5):854–869, 2007.

[29] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004.

[30] C. Vogler, Z. Li, A. Kanaujia, S. Goldenstein, and D. Metaxas. The best of both worlds: Combining 3D deformable models with Active Shape Models. In *ICCV*, 2007.

[31] L. Wolf and I. Martin. Robust boosting for learning from few examples. In *CVPR*, volume 1, pages 359–364, 2005.

[32] S. Yan, M. Li, H. Zhang, and Q. Cheng. Ranking prior likelihood distributions for Bayesian shape localization framework. In *ICCV*, volume 1, pages 51–58, 2003.

[33] Y. Zheng, X. S. Zhou, B. Georgescu, S. K. Zhou, and D. Comaniciu. Example based non-rigid shape detection. In *ECCV*, pages 423–436, 2006.

[34] Y. Zhou, L. Gu, and H. J. Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via Bayesian inference. In *CVPR*, volume 1, pages 109–116, 2003.