Statistical Learning of Evaluation Function for ASM/AAM Image Alignment

Xiangsheng Huang[1] Stan Z. Li[2] Yangsheng Wang[1]

[1] Institute of Automation,Chinese Academy of Sciences,Beijing, China 100080
[2] Microsoft Research Asia,Beijing, China 100080

Abstract. Alignment between the input and target objects has great impact on the performance of image analysis and recognition system, such as those for medical image and face recognition. Active Shape Models (ASM)[1] and Active Appearance Models (AAM) [2, 3] provide an important framework for this task. However, an effective method for the evaluation of ASM/AAM alignment results has been lacking. Without an alignment quality evaluation mechanism, a bad alignment cannot be identified and this can drop system performance.
In this paper, we propose a statistical learning approach for constructing an evaluation function for face alignment. A *nonlinear* classification function is learned from a set of positive (good alignment) and negative (bad alignment) training examples to effectively distinguish between qualified and un-qualified alignment results. The AdaBoost learning algorithm is used, where weak classifiers are constructed based on edge features and combined into a strong classifier. Several strong classifiers is learned in stages using bootstrap samples during the training, and are then used in cascade in the test. Experimental results demonstrate that the classification function learned using the proposed approach provides semantically more meaningful scoring than the reconstruction error used in AAM for classification between qualified and un-qualified face alignment.

1 Introduction

Many image analysis and recognition application require alignment between an object in the input image and a target object. Alignment can have a great impact on the system performance. For examples, in appearance based face recognition, the alignment provide a more sensible foundation for template matching based recognition; the use of bad alignment can drop system performance significantly.

Active Shape Models (ASM) [1] and Active Appearance Models(AAM) [2, 3] have been used as alignment algorithms in medical image analysis and face recognition [4]. However, an effective method for the evaluation of ASM/AAM alignment results has been lacking: There has been no convergence criterion for ASM. As such, the ASM search can give a bad result without giving the user a warning. In the AAM, the PCA (Principal Component Analysis) reconstruction error is used as a distance measure for the evaluation of alignment quality (and for guiding the search as well). However, the reconstruction error may not be

a good discriminant for the evaluation of alignment quality because a non-face can look like a face when projected onto the PCA face subspace.

In this paper, we propose a statistical learning approach for constructing an effective evaluation function for face alignment. A *nonlinear* classification function is learned from a training set of positive and negative training examples to effectively distinguish between qualified and un-qualified alignment results. The positive subset consists of qualified face alignment examples and the negative subset consists of obviously un-qualified and near-but-not-qualified examples.

We use AdaBoost algorithm [5, 6] for the learning. A set of candidate weak classifiers are created based on edge features extracted using Sobel-like operators. We choose to use edge features because crucial cues for alignment quality are around edges. Experimentally, we also found that the Sobel features produced significant better results than other features such as Haar wavelets. The AdaBoost learning selects or learns a sequence of best features and the corresponding weak classifiers and combines them into a strong classifier.

In the training stage several strong classifiers is learned in stages using bootstrap training samples, and in the test they are cascaded to form a stronger classifier, following an idea in boosting based face detection [7]. Such a divide-conquer strategy makes the training easier and the good-bad classification more effective. The evaluation function thus learned gives a quantitative confidence and the good-bad classification is achieved by comparing the confidence with a learned optimal threshold.

There are two important distinctions between an evaluation function thus learned and the linear evaluation function of reconstruction error used in AAM. First, the evaluation is learned in such a way to distinguish between good and bad alignment. Secondly, the scoring is nonlinear, which provides a semantically more meaningful classification between good and bad alignment. Experimental results demonstrate that the classification function learned using the proposed approach provides semantically meaningful scoring for classification between qualified and un-qualified face alignment.

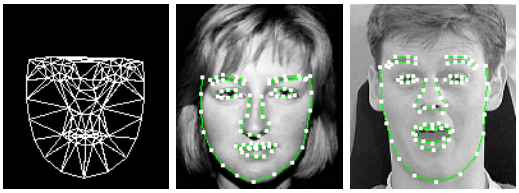
The rest of the paper is organized as follows: Section 2 briefly describes the ASM method and the problem of alignment quality evaluation. AdaBoost based learning is presented in Section 3. Section 4 provides the construction of candidate weak classifiers. Section 5 proposes the learning of weak classifiers. Section 6 provides experimental results. Section 7 draws a conclusion.

2 ASM/AAM and Solution Quality Evaluation

Let us briefly describe the ASM and AAM methods before a discussion about the issue of alignment evaluation. The standard ASM consists of two statistical models: (1) global shape model, which is derived from the landmarks in the object contour; (2) local appearance models, which is derived from the profiles perpendicular to the object contour around each landmark. ASM uses local models to find the candidate shape and the global model to constrain the searched shape.

AAM makes use of the PCA techniques to model both shape variation and texture variation, and the correlations between the shape subspace and texture subspace to model the face. In searching for a solution, it assumes linear relationships between appearance variation and texture variation, and between texture variation and position variation; and learns the two linear regression models from training data. The minimizations in high dimensional space is reduced in two models facilitate. This strategy is also developed in the active blob model by Sclaroff and Isidoro [8].

While the training data for ASM consists of shape only, and that for AAM consists of both shape and texture. Denote a *shape* $S_0 = ((x_1, y_1), \ldots, (x_K, y_K)) \in \mathbb{R}^{2K}$ by a sequence of K points in the 2D image plane, and a *texture* T_0 using the patch of pixel intensities enclosed by S_0. Let $\overline{S}$ be the mean shape of all the training shapes, as illustrated in Fig. 1(a). Fig. 1(b) and (c) show two examples of shapes overlayed on the faces. In AAM, all the shapes are aligned or warping to the tangent space of the mean shape $\overline{S}$. After that, the texture T_0 is warped correspondingly to $T \in \mathbb{R}^L$, where L is the number of pixels in the mean shape $\overline{S}$. The warping may be done by pixel value interpolation, *e.g.* using a triangulation or thin plate spline method.



(a) (b) (c)

Fig. 1. (a) The mesh of the mean shape. (b) & (c): Two face instances labelled with 83 landmarks.

There has been no convergence criterion for ASM search nor quality evaluation. In ASM search, the mean shape is placed near the center of the detected image and a coarse to fine search performed. Large movements are made in the first few iterations, getting the position roughly. As the search progressing, more subtle adjustments are made. The result can gives a good match to the target image or it can fail (see Figure. 2). The failure can happen even if the starting position is near the target. When the variations of expression and illumination are large, ASM search can diverge in order to match the local image pattern.

In AAM search, the PCA reconstruction error is used to guide the search and used as the convergence and evaluation criterion. Such an error function is defined as the distance between the image patch (aimed to contain the face region only) after warping to the mean shape and the projection of the patch onto the PCA subspace of face texture. However, the reconstruction error may not be a good measure for the evaluation of alignment quality because a non-face can look like a face when projected onto the PCA face subspace. Cootes pointed

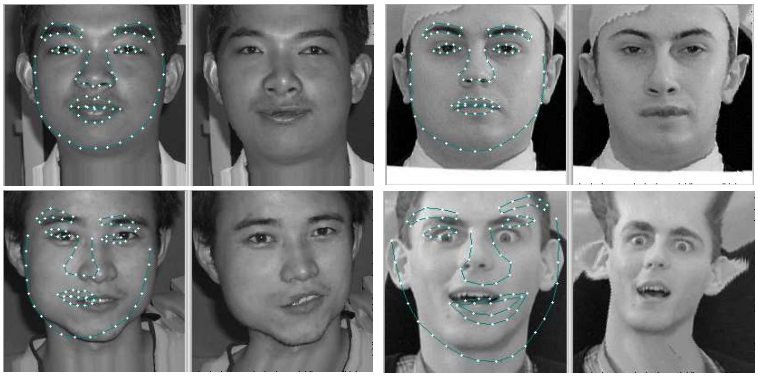


Fig. 2. Four face instances of qualified (top) and un-qualified (bottom) examples with their warped images

out that, of 2700 testing examples, 519 failed to converge to a satisfactory result (the mean point position error is greater than 7.5 pixels per point) [4].

In the following we present a learning based approach for learning evaluation function for ASM/AAM based alignment.

3 AdaBoost Based Learning

Our objective is to learn an evaluation function from a training set of qualified and un-qualified alignment examples. From now on, we use the terms positive and negative examples for classes of data. These examples are the face image after warping to mean shape, as shown in Fig. 2. Face alignment quality evaluation can be posed as a two class classification problem: given an alignment result x (*i.e.* warped face), the evaluation function $H(x) = +1$ if x is positive example, or -1 otherwise. we want to learn such an $H(x)$ that can provide a score in $[-1, +1]$ with a threshold around 0 for the binary classification.

For two class problems, a set of N labelled training examples is given as $(x_1, y_1), \ldots, (x_N, y_N)$, where $y_i \in \{+1, -1\}$ is the class label associated with example $x_i \in \mathbb{R}^n$. A stronger classifier is a linear combination of M weak classifiers

$$H_M(x) = \sum_{m=1}^{M} h_m(x) \tag{1}$$

In the real version of AdaBoost [5, 6], the weak classifiers can take a real value, $h_m(x) \in \mathbb{R}$, and have absorbed the coefficients needed in the discrete version ($h_m(x) \in -1, +1$ in the latter case). The class label for x is obtained as $H(x) = \text{sign}[H_M(x)]$ while the magnitude $|H_M(x)|$ indicates the confidence. Every training example is associated with a weight. During the learning process, the weights are updated dynamically in such a way that more emphasis is placed on hard examples which are erroneously classified previously. It is noted in recent

studies [9–11] that the artificial operation of explicit re-weighting is unnecessary and can be incorporated into a functional optimization procedure of boosting.

---

0. (Input)
   (1) Training examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$,
   where $N = a + b$; of which a examples have $y_i = +1$
   and b examples have $y_i = -1$;
   (2) The maximum number $M_{\max}$ of weak classifiers to be combined;
1. (Initialization)
   $w_i^{(0)} = \frac{1}{2a}$ for those examples with $y_i = +1$ or
   $w_i^{(0)} = \frac{1}{2b}$ for those examples with $y_i = -1$.
   $M = 0$;
2. (Forward Inclusion)
   while $M < M_{\max}$
   (1) $M \leftarrow M + 1$;
   (2) Choose h_M according to Eq.4;
   (3) Update $w_i^{(M)} \leftarrow \exp[-y_i H_M(x_i)]$, and normalize to $\sum_i w_i^{(M)} = 1$;
3. (Output)
   $H(x) = \text{sign}[\sum_{m=1}^{M} h_m(x)]$.

Fig. 3. RealBoost Algorithm.

An error occurs when $H(x) \neq y$, or $yH_M(x) < 0$. The "margin" of an example (x, y) achieved by $h(x) \in \mathbb{R}$ on the training set examples is defined as $yh(x)$. This can be considered as a measure of the confidence of the h's prediction. The upper bound on classification error achieved by H_M can be derived as the following exponential loss function [12]

$$J(H_M) = \sum_i \mathrm{e}^{-y_i H_M(x_i)} = \sum_i \mathrm{e}^{-y_i \sum_{m=1}^{M} h_m(x)} \tag{2}$$

AdaBoost construct $h_m(x)$ by stagewise minimization of Eq.(2). Given the current $H_{M-1}(x) = \sum_{m=1}^{M-1} h_m(x)$, the best $h_M(x)$ for the new strong classifier $H_M(x) = H_{M-1}(x) + h_M(x)$ is the one which leads to the minimum cost

$$h_M = \arg\min_{h^\dagger} J(H_{M-1}(x) + h^\dagger(x)) \tag{3}$$

The minimizer is [5, 6]

$$h_M(x) = \frac{1}{2} \log \frac{P(y = +1|x, w^{(M-1)})}{P(y = -1|x, w^{(M-1)})} \tag{4}$$

where $w^{(M-1)}(x, y) = \exp(-yF_{M-1}(x))$ is the weight for the labeled example (x, y) and

$$P(y = +1|x, w^{(M-1)}) = \frac{E\left(w(x, y) \cdot 1_{[y=+1]}|x\right)}{E\left(w(x, y) \mid x\right)} \tag{5}$$

where $E(\cdot)$ stands for the mathematical expectation and $1_{[C]}$ is one if C is true or zero otherwise. $P(y=-1|x,w^{(M-1)})$ is defined similarly.

The AdaBoost algorithm based on the descriptions from [5, 6] is shown in Fig. 3. There, the re-weight formula in step 2.(3) is equivalent to the multiplicative rule in the original form of AdaBoost [13, 5]. In Section 3.2, we will present a statistical model for stagewise approximation of $P(y=+1|x,w^{(M-1)})$.

4 Construction of Candidate Weak Classifiers

The optimal weak classifier at stage M is derived as Eq.(4). Using $P(y|x,w) = p(x|y,w)P(y)$, it can be expressed as

$$h_M(x) = L_M(x) - T \tag{6}$$

where

$$L_M(x) = \frac{1}{2}\log\frac{p(x|y=+1,w)}{p(x|y=-1,w)} \tag{7}$$

$$T = \frac{1}{2}\log\frac{P(y=+1)}{P(y=-1)} \tag{8}$$

The log likelihood ratio (LLR) $L_M(x)$ is learned from the training examples of the two classes. The threshold T is determined by the log ratio of prior probabilities. In practice, T can be adjusted to balance between the detection and false alarm rates (*i.e.* to choose a point on the ROC curve).

Learning optimal weak classifiers requires modelling the LLR of Eq.(7). Estimating the likelihood for high dimensional data x is a non-trivial task. In this work, we make use of the stagewise characteristics of boosting, and derive the likelihood $p(x|y,w^{(M-1)})$ based on an over-complete scalar feature set $\mathcal{Z} = \{z'_1,\ldots,z'_K\}$. More specifically, we approximate $p(x|y,w^{(M-1)})$ by $p(z_1,\ldots,z_{M-1},z'|y,w^{(M-1)})$ where z_m $(m=1,\ldots,M-1)$ are the features that have already been selected from $\mathcal{Z}$ by the previous stages, and z' is the feature to be selected. The following describes the candidate feature set $\mathcal{Z}$, and presents a method for constructing weak classifiers based on these features.

Because the shape is about boundaries between regions, it makes sense to use edge information (magnitude or orientation or both) extracted from a grey-scale image. In this work, we use the simple Sobel filter for extracting the edge information. Two filters are used: K_w for horizontal edges and K_h for vertical edges, as follows:

$$K_w(w,h) = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad and \quad K_h(w,h) = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{9}$$

The convolution of the image with the two filter masks gives two edge strength values.

$$G_w(w,h) = K_w * I(w,h) \tag{10}$$

$$G_h(w,h) = K_h * I(w,h) \tag{11}$$

The edge magnitude and direction are obtained as:

$$S(w,h) = \sqrt{G_w^2(w,h) + G_h^2(w,h)} \tag{12}$$

$$\phi(w,h) = \arctan(\frac{G_h(w,h)}{G_w(w,h)}) \tag{13}$$

The edge information based on Sobel operator is sensitive to noise. To solve this problem, we use sub-block of image to convolve with Sobel filter (see Fig. 4), which is similar to Haar-like feature calculation.

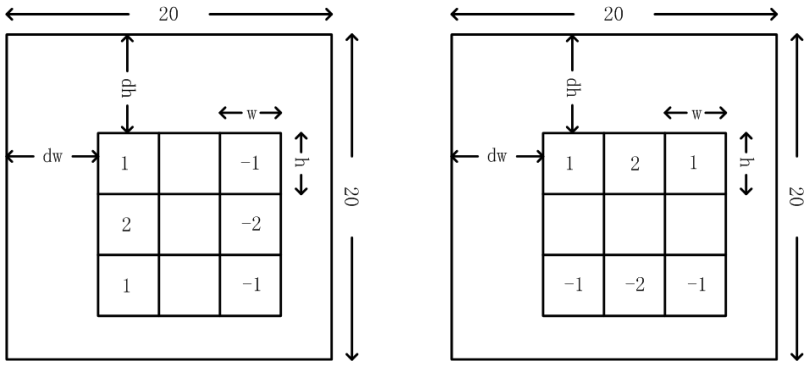


Fig. 4. The two types of simple Sobel-like filters defined on sub-windows. The rectangles are of size $w \times h$ and are at distances of (dw, dh) apart. Each feature takes a value calculated by the weighted $(\pm 1, \pm 2)$ sum of the pixels in the rectangles.

5 Statistical Learning of Weak Classifiers

A scalar feature $z'_k : x \rightarrow \mathbf{R}$ is a transform from the n-dimensional (400-D if a face example x is of size 20x20) data space to the real line. These block differences are an extension to the Sobel filters. For each face example of size 20x20, there are hundreds of thousands of different z'_k for admissible w, h, dw, dh values, so $\mathcal{Z}$ is an over-complete feature set for the intrinsically low-dimensional face pattern x. In this work, an optimal weak classifier (6) is associated with a single scalar feature; to find the best new weak classifier is to choose the best corresponding feature.

We can define the following component LLR's for the target $L_M(x)$:

$$\tilde{L}_m(x) = \frac{1}{2} \log \frac{p(z_m|y=+1, w^{(m-1)})}{p(z_m|y=-1, w^{(m-1)})} \tag{14}$$

for the selected features, z_m's ($m = 1, \ldots, M-1$), and

$$L_k^{(M)}(x) = \frac{1}{2} \log \frac{p(z'_k(x)|y=+1, w^{(M-1)})}{p(z'_k(x)|y=-1, w^{(M-1)})} \tag{15}$$

for features to be selected, $z'_k \in \mathcal{Z}$. Then, after some mathematical derivation, we can approximate the target LLR function as

$$L_M(x) = \frac{1}{2} \log \frac{p(x|y=+1, w^{(M-1)})}{p(x|y=-1, w^{(M-1)})} \approx \sum_{m=1}^{M-1} \tilde{L}_m(x) + L_k^{(M)}(x) \tag{16}$$

Let

$$\Delta L_M(x) = L_M(x) - \sum_{m=1}^{M-1} \tilde{L}_m(x) \tag{17}$$

The best feature is the one whose corresponding $L_k^{(M)}(x)$ best fits $\Delta L_M(x)$. It can be found as the solution to the following minimization problem

$$k^* = \arg\min_{k,\beta} \sum_{i=1}^{N} \left[\Delta L_M(x_i) - \beta L_k^{(M)}(x_i) \right]^2 \tag{18}$$

This can be done in two steps as follows: First, find k^* for which

$$(L_k^{(M)}(x_1), L_k^{(M)}(x_2), \ldots, L_k^{(M)}(x_N)) \tag{19}$$

is most parallel to

$$(\Delta L_M(x_1), \Delta L_M(x_2), \ldots, \Delta L_M(x_N)) \tag{20}$$

This amounts to finding k for which $L_k^{(M)}$ is most correlated with ΔL_M over the data distribution, and set $z_M = z'_{k^*}$. Then, we compute

$$\beta^* = \frac{\sum_{i=1}^{N} \Delta L_M(x_i) L_{k^*}(x_i)}{\sum_{i=1}^{N} [L_{k^*}(x_i)]^2} \tag{21}$$

After that, we obtain

$$\tilde{L}_M(x) = \beta^* L_{k^*}(x) \tag{22}$$

The strong classifier is then given as

$$H_M(x) = \sum_{m=1}^{M} \left(\tilde{L}_m(x) - T \right) = \sum_{m=1}^{M} \tilde{L}_m(x) - MT \tag{23}$$

The evaluation function $H_M(x)$ thus learned gives a quantitative confidence and the good-bad classification is achieved by comparing the confidence with the threshold value of 0 (zero).

There are two important distinctions between an evaluation functions thus learned and the linear evaluation function of reconstruction error used in AAM. First, the evaluation is learned in such a way to distinguish between good and bad alignment. Secondly, the scoring is nonlinear, which provides a semantically more meaningful classification between good and bad alignment.

6 Experimental Results

The positive and negative training examples are generated as follows: All the shapes are aligned or warping to the tangent space of the mean shape $\overline{S}$. After that, the texture T_0 is warped correspondingly to $T \in \mathbb{R}^L$, where L is the number of pixels in the mean shape $\overline{S}$.

In our work, 2536 positive examples and 3000 negative examples are used to train a strong classifier. The 2536 positive examples are derived from 1268 original positive examples plus the mirror images. The negative examples are generated by random rotating, scaling, shifting positive examples' shape points. A strong classifier is trained to reject 92% negative examples, while correctly accepting 100% of positive examples.

A cascade of classifiers is trained to obtain a computational effective model, makes training easier with divide-conquer strategy. When training a new stage, negative examples are bootstrapped based on the classifier trained in the previous stages. The details of training a cascade of 5 stages is summarized Table 1. As the result of training, we achieved 100% correct acceptance and correct rejection rates on the training set.

Table 1. Training results (WC: weak classifier)

stage	number of pos	number of neg	number of WC	False Alarm
1	2536	3000	22	0.076
2	2536	3000	237	0.069
3	2536	888	294	0.263
4	2536	235	263	0.409
5	2536	96	208	0.0

We compare the proposed Adaboost learning based method with the PCA texture reconstruction error based evaluation method, using the same data sets (but PCA does not need negative examples in the training). The dimensionality of the PCA subspace is chosen to retain 99% of the total variance of the data. The best threshold of reconstruction error is selected to minimize the classification error. Fig. 5 shows the ROC curve for the reconstruction error based alignment evaluation method for the training set. Note that this method cannot achieve 100% correct rates.

During the test, a total of 1528 aligned examples (800 qualified images and 728 un-qualified images), which are not seen during the training, are used. We evaluate each face images and give a score in terms of (a) the confidence value $H_M(x)$ for the learning based method and (b) the confidence value threshold $-$ dist_{PCA} for the PCA based method. The qualified and un-qualified alignment decision is judged by comparing the score with the normalized threshold of 0. Some examples of qualified (the top part) and un-qualified (the bottom part) face alignment results are shown Fig. 6, with the corresponding scores (the first line of the numbers is for the proposed method, and the second line for the

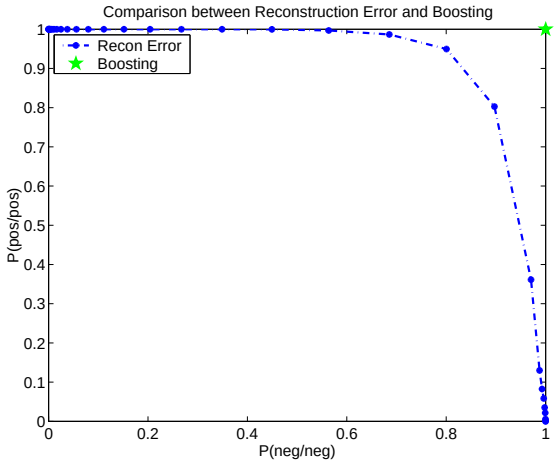


Fig. 5. Correct rate curve for the reconstruction error based alignment evaluation for the training set.

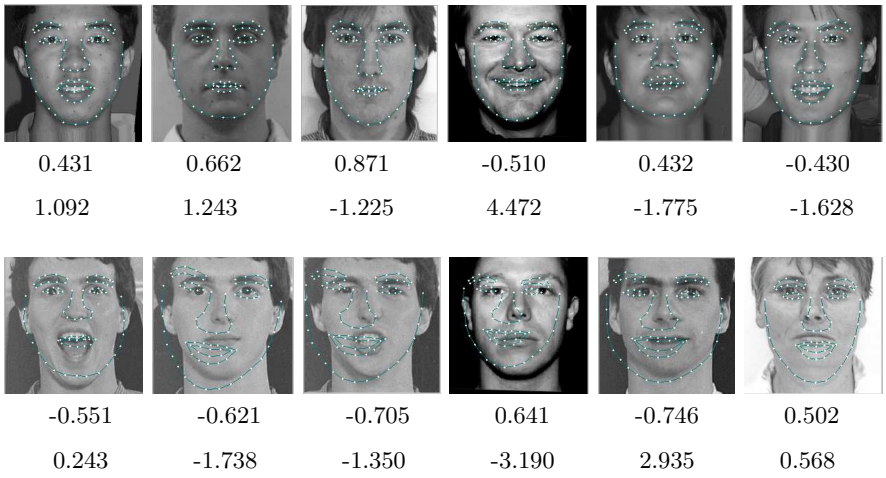


Fig. 6. Alignment quality evaluation results: qualified (top part) and un-qualified (bottom part) alignment examples

PCA based method). This qualitatively demonstrates better sensibility of the proposed method for alignment evaluation.

Fig. 7 quantitatively compares the two methods in terms of their ROC curves (first plot) and error curves (the second plot), where the axis label $P(pos/neg)$ means the false positive rate and so on. From the error curves, we can see that the equal error rate of the proposed method is about 40%, while that of reconstruction error based method is 48%. The proposed approach provides a more

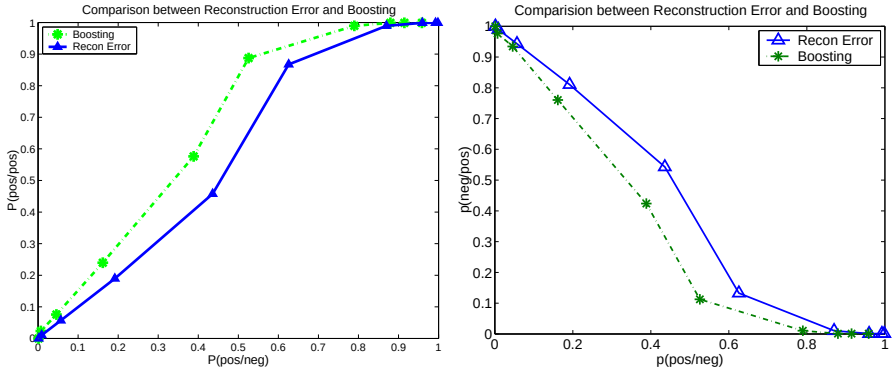


Fig. 7. Comparision between reconstruction error method and boost method

effective method to distinguish between qualified and un-qualified face alignment than the reconstruction error used in AAM.

Lastly, we would like to make a comment on the choice of image features for construction weak classifiers: Experimentally, we also found that the Sobel features produced significant better results than other features such as Haar wavelets. This is not elaborated here.

7 Conclusion and Future Work

In this paper, we proposed a statistical learning approach for constructing an effective evaluation function for face alignment. A set of candidate weak classifiers are created based on edge features extracted using Sobel-like operators. Experimental results demonstrate that the classification function learned using the proposed approach provides semantically more meaningful scoring than the reconstruction error used in AAM for classification between qualified and un-qualified face alignment. While the number of negative examples (un-qualified alignment) is huge, so far we used only about 40,000+, and 2536 positive examples. This training set is still smaller; and so when we easily achieved 100% of training accuracy, the test performance is significantly lower. We expect a better trained nonlinear quality evaluation function when a larger training data which covers larger variation is used.

References

1. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: "Active shape models: Their training and application". CVGIP: Image Understanding **61** (1995) 38–59
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: ECCV98. Volume 2. (1998) 484–498
3. Edwards, G.J., Cootes, T.F., Taylor, C.J.: "Face recognition using active appearance models". In: Proceedings of the European Conference on Computer Vision. Volume 2. (1998) 581–695

4. Cootes, T.F., , Taylor, C.J.: Statistical models of appearance for computer vision. Technical report, `www.isbe.man.ac.uk/~bim/refs.html` (2001)
5. Schapire, R.E., Singer, Y.: "Improved boosting algorithms using confidence-rated predictions". In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory. (1998) 80–91
6. Friedman, J., Hastie, T., Tibshirani, R.: "Additive logistic regression: a statistical view of boosting". The Annals of Statistics **28** (2000) 337–374
7. Viola, P., Jones, M.: "Robust real time object detection". In: IEEE ICCV Workshop on Statistical and Computational Theories of Vision, Vancouver, Canada (2001)
8. Sclaroff, S., Isidoro, J.: "Active blobs". In: Proceedings of IEEE International Conference on Computer Vision, Bombay, India (1998)
9. Friedman, J.: "Greedy function approximation: A gradient boosting machine". The Annals of Statistics **29** (2001)
10. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Functional gradient techniques for combining hypotheses. In Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press, Cambridge, MA (1999) 221–247
11. Zemel, R., Pitassi, T.: "A gradient-based boosting algorithm for regression problems". In: Advances in Neural Information Processing Systems. Volume 13., Cambridge, MA, MIT Press (2001)
12. Schapire, R., Freund, Y., Bartlett, P., Lee, W.S.: "Boosting the margin: A new explanation for the effectiveness of voting methods". The Annals of Statistics **26** (1998) 1651–1686
13. Freund, Y., Schapire, R.: "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences **55** (1997) 119–139