

# Face Alignment Using Boosting and Evolutionary Search

Hua Zhang<sup>1</sup>, Duanduan Liu<sup>2</sup>, Mannes Poel<sup>3</sup>, and Anton Nijholt<sup>3</sup>

<sup>1</sup> College of Software Engineering, Southeast University, Nanjing 210096, China  
reynzhang@sina.com

<sup>2</sup> Lab of Science and Technology, Southeast University, Nanjing 210096, China  
liuduanduan@seu.edu.cn

<sup>3</sup> Human Media Interaction, University of Twente, P.O. Box 217  
7500 AE Enschede, The Netherlands  
{anijholt, mpoel}@cs.utwente.nl

**Abstract.** In this paper, we present a face alignment approach using granular features, boosting, and an evolutionary search algorithm. Active Appearance Models (AAM) integrate a shape-texture-combined morphable face model into an efficient fitting strategy, then Boosting Appearance Models (BAM) consider the face alignment problem as a process of maximizing the response from a boosting classifier. Enlightened by AAM and BAM, we present a framework which implements improved boosting classifiers based on more discriminative features and exhaustive search strategies. In this paper, we utilize granular features to replace the conventional rectangular Haar-like features, to improve discriminability, computational efficiency, and a larger search space. At the same time, we adopt the evolutionary search process to solve the deficiency of searching in the large feature space. Finally, we test our approach on a series of challenging data sets, to show the accuracy and efficiency on versatile face images.

**Keywords:** face alignment, boosting appearance models, granular features, evolutionary search.

## 1 Introduction

Face alignment is usually regarded as minimizing the distance between a template and a given face image. Among the various technologies of face alignment, Active Shape Models (ASM) [1] and Active Appearance Models (AAM) [2] have gradually taken the stage center. ASM utilized the local texture information in search of a better template, and AAM constructed appearance models according to shape parameters and global texture constraints. After ASM and AAM, Zhou *et al.* [3] introduced Bayesian Tangent Shape Model (BTSM) with an EM-based method to implement the MAP estimation, Liang *et al.* [4] utilized a Constrained Markov Network for accurate face alignment, and Boosting Appearance Models (BAM) [5] presented a discriminative method with boosting algorithm and rectangular Haar-like features, which resulted in outstanding accuracy and robustness. Enlightened by BAM, the speed of face alignment can be improved by more discriminative features and boosting classifiers bring in the benefit of

computational efficiency. Huang *et al.* [6] introduced granular features to form a larger feature space. At the same time, evolutionary search process [7] made great improvements on exploring the better granular features in a large feature space. In this paper, we improve BAM by introducing granular features and evolutionary search. Firstly, we generate a large number of positive samples by wrapping the image from the average shape, and then we harvest negative samples by randomly perturbing parameters from the current shape. Secondly, we generate a series of granular features from the feature space. After the evolutionary search, we can find a set of granular features to construct a strong classifier. Finally, the face alignment process is regarded as finding the warped image, which has a higher response than the final threshold of a strong classifier.

This paper is organized in the following way: Section 2 introduces Boosting Appearance Models, Section 3 expatiates the process of exploring the better weak classifier of boosting algorithm, and the fitting process of alignment is presented in Section 4. Finally, Section 5 compares our method with other methods by experiments.

## 2 Boosting Appearance Models

Active Appearance Models (AAM) [2] are composed of a shape model, a texture model, and a fitting method. Boosting Appearance Models (BAM) [5] propose a more discriminative method via rectangular Haar-like features and boosting. Inspired by AAM and BAM, we propose a framework based on granular features, a Bayesian stump weak learner, and evolutionary search for features.

### 2.1 Shape and Texture Models in Active Appearance Models [2]

Inspired by Active Appearance Models (AAM) [2], the morphable face model is generated from a set of facial images. From a giving face database, we manually label a series of 2D annotations  $\{x_i, y_i\}, i = 1, 2, \dots, n$ , which include important facial components such as eyes, nose, and mouth. For each face image, we constitute a shape  $s = [x_1, y_1, \dots, x_n, y_n]^T$  from these annotations. After applying Principle Component Analysis (PCA) [2], a morphable shape model is constructed as

$$s = \bar{s} + U_s P, \quad (1)$$

where  $\bar{s}$  is the mean shape,  $P = [p_1, \dots, p_n]$  are the first  $n$  principal component vectors, and  $U_s$  is the coefficients of  $s$  with respect to these first  $n$  principal components. In virtue of shape, texture information of the images is warped into the mean shape  $\bar{s}$  via piecewise affine transformation  $T(x, y; U_s)$ . If we want to warp an image  $\mathbf{I}$ , a set of pints  $I_j \in \mathbf{I}, j = 1, \dots, n$  in the coordinates  $\{x_i, y_i | i \in 1, \dots, n\}$  are mapped to new positions  $\{x'_i, y'_i\}$  by defined warping function

$$T(x, y; P) = [1, x, y] \mathbf{A}(P), \quad (2)$$

where  $\mathbf{A}(P)$  is a transformation matrix between average shape  $\bar{s}$  and current shape  $s$  [2]. When shape parameters  $P$  are given, the  $\mathbf{A}(P)$  matrix needs to be computed for

each triangle. It is a method to normalize all warp images as the same size. Then the eigen-texture information is presented by

$$t = \bar{t} + U_t Q. \quad (3)$$

Finally the PCA based shape and texture model are combined to form the appearance model.

Conventionally, the fitting process of AAM is in search of the minimum between current warped texture and the model texture. Hence, the matching process is

$$\delta(P) = \|t_s - t_m\|^2, \quad (4)$$

where  $P$  is the shape parameters of the shape model,  $t_s = I(T(x, y; P))$  ( $I$  means cropping the texture from the transformed image  $T(x, y; P)$ ) is the warped texture of the current shape, and  $t_m$  is the current model texture given by Equation 3. By gradient ascent methods, this minimization can be solved.

## 2.2 Appearance Modeling

Similar to AAM and BAM, our appearance model is derived from the warped image  $I(T(x, y; P))$ . If we consider face alignment as a two-category classification problem, the shape instance  $S(P)$  is the manually landmark of a face image  $I$ , then  $P$  becomes the positive shape parameters. At the same time, we perturb  $P$  to generate the negative shape parameters. If we can define a function  $h(\bullet)$ , which outputs positive score when the given sample is positive, or outputs negative score when the given sample is negative, then we can collect a set of  $h(\bullet)$  to add the responses from  $h(\bullet)$ . When the added response is over a given threshold, the current parameters are just the landmark parameters. Adaboost is a simple and robust method to learn an accurate classifier from a set of weak classifiers [8]. After a feature  $\theta_i = \theta(I(T(x_i, y_i; P_i)))$  is extracted from the wrapped image as a weak classifier, we can construct a combined strong classifier by these features. Therefore, we define a combination of many weak classifiers and local features as the appearance model

$$H(I(T(x, y; P)), \Theta) = \sum_{m=1}^M h_m(I(T(x, y; P)), \theta_i), \theta_i \in \Theta, \quad (5)$$

where  $h(I(T(x, y; P)), \theta_i)$  is a function on using feature  $\theta_i$  to operate image patch  $I(T(x, y; P))$ . Namely,  $H(\bullet)$  and  $h_m(\bullet)$  are strong and weak classifiers respectively.

## 2.3 Real Adaboost Learning for Strong Classifier

Boosting [8] algorithm is a method of integration of various "weak" classifiers into a powerful "board". In this paper, we choose Real AdaBoost [9] algorithm, which returns the response of weak classifiers as real numbers (Table 1). Given a set of faces with annotated landmarks, we generate training data for boosting learning. From each shape, we warp image  $I(W(x, y; P))$  as the positive samples. Then we randomly perturb  $P$  to

get the negative samples. Each sample is normalized to the same size to construct the training set (Figure 2(a)). The final strong classifier contains a series of weak classifiers, which preserves a granular feature and a threshold. After the responses of the weak classifier are accumulated, we can get the response trace shown in Figure 2(b).

**Table 1.** Real Adaboost for learning strong classifier

---

<ul style="list-style-type: none"> <li>• Input and initialize  Training data <math>\{x_i; i = 1, \dots, K\}</math>, and their labels <math>\{y_i; i = 1, \dots, K\}</math>.  Initialize weights <math>\omega_i = \frac{1}{K}, i = 1, \dots, K</math>.</li> <li>• For <math>m = 1, \dots, M</math>, do <ol style="list-style-type: none"> <li>(1) Fit the class probability estimate <math>h_m(x) = \arg \min \sum h(x) = \sum_{i=1}^k \omega_i (y_i - h(x_i))^2</math>.</li> <li>(2) Choose this weak classifier <math>h_m^* = \frac{1}{2} \log \frac{h_m(x)}{1-h_m(x)} \in R</math>.</li> <li>(3) Update the weight <math>\omega_i = \frac{\omega_i \exp[-y_i h_m(x_i)]}{Z_t}</math>, where <math>Z_t</math> is a normalization factor.</li> </ol> </li> <li>• Output  The strong classifier <math>sign[H(x)] = sign[\sum_m h_m(x)]</math>.</li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

### 3 Learning Sparse Granular Features for Weak Classifier

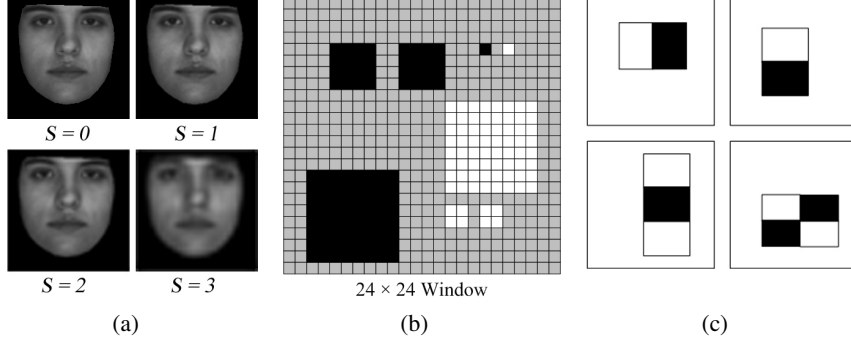
Since face alignment is time-constrained, BAM constructs the weak classifiers based on rectangular Haar-like features [10], which lead to great success because of integral image. However, Haar-like features encounter defects in irregular patterns. In order to overcome this difficulty, Huang *et al.* [6] presented a granular space to generate a series of granular features, which adopts a heuristic search algorithm to search for discriminative sparse features. In the process of search for better features, Treptow and Zell [11], Abramson *et al.* [7] utilized an evolutionary method to find better features. We combine these ideas, and introduce an evolutionary search algorithm to pursue discriminative granular features.

#### 3.1 Granular Features

A granular space is established by a pyramid of bitmaps  $\{I_0, I_1, I_2, I_3\}$ , and each layer of the pyramid is denoted from a smooth filtering in a way of averaging  $2^s \times 2^s$  patches of the input image (Figure 1(a)). In space, a sparse feature is represented by a linear combination of several granules, as

$$\theta = \sum_i \alpha_i I(p(x_i, y_i, s_i)), \alpha \in \{-1, +1\}, s_i \in \{0, 1, 2, 3\}, \quad (6)$$

where  $I(\bullet)$  indicates the pixel data of a granule. Through three parameters: x-offset  $x_i$ , y-offset  $y_i$ , and scale  $s_i$ , a granule  $p(x_i, y_i, s_i)$  means a square at the coordinate  $(x_i, y_i)$  with the size of  $2^{s_i} \times 2^{s_i}$ . From a  $24 \times 24$  reference window, we can totally extract  $\sum_{s=0,1,2,3} (24 - 2^s + 1)^2 = 1835$  different granules. Compared to conventional rectangular Haar-like features [10], sparse granular features are more scalable and robust [6].



**Fig. 1.** (a) Pyramid of granular space. (b) Example of granular sparse features which black or white color indicates the coefficient  $\alpha_i$  in Section 3.1. (c) Examples of rectangular Haar-like features [10] for initialization in Section 3.3.

### 3.2 Bayesian Stump Look Up Table Weak Classifier

When we consider the problem of two-category classification, the probability of Bayesian error is defined as

$$P(\text{error}|x) = \begin{cases} P(\omega_1|x) & \text{if we decide } \omega_2 \\ P(\omega_2|x) & \text{if we decide } \omega_1, \end{cases} \quad (7)$$

and the expected error is  $P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}, x) dx = \int_{-\infty}^{\infty} P(\text{error}|x) p(x) dx$ . If we follow the Bayes' decision rule, we decide  $\omega_1$  if  $P(\omega_1|x) > P(\omega_2|x)$  and otherwise decide  $\omega_2$ , then Equation 7 becomes  $P(\text{error}|x) = \min[P(\omega_1|x), P(\omega_2|x)]$ , and the total Bayesian error is

$$B_{\text{error}} = P(\text{error}) = \int_{-\infty}^{\infty} \min[P(\omega_1|x), P(\omega_2|x)] dx. \quad (8)$$

Xiao *et al.* [12] proposed a method called Bayesian Stump to find  $P(\omega_c, x)$ ,  $c \in \{1, 2\}$  by using histogram to estimate the probability distribution. We divide all features' output value  $\{\mu(\theta_i)\}$  into  $k$  sections  $\delta_k = (r_{k-1}, r_k]$ , and the histogram of  $P(\omega_c, x)$  is

$$P(k, \omega_c) = \int_{\mu(\theta_i) \in \delta_k} P(\mu(\theta_i), \omega_c) d\mu(\theta_i), c \in \{1, 2\}. \quad (9)$$

Following the method in [12], we can easily build a  $k$ -bins Bayesian Stump. Moreover, we can extend it to a Look Up Table(LUT) weak classifier for RealBoost algorithms by using log-likelihood output to replace the binary output in every interval. In summary, we can define the weak classifier as

$$h(x, \theta) = \frac{1}{2} \ln \sum_{k=1}^K \left\{ \frac{W_{-1}^k + \varepsilon}{W_{+1}^k + \varepsilon} \right\} B_k(\theta(x)), B_k(u) = \begin{cases} 1, & u \in \delta_k \\ 0, & \text{otherwise} \end{cases}. \quad (10)$$

$W_{+1}^j$  and  $W_{-1}^j$  are the total weights of positive and negative samples falling into the  $j$ th bin,  $\theta(x) = \theta(I(T(x, y; p)))$  represent the feature under the current wrapped face patch, and  $\varepsilon$  is a small constant to avoid that denominator is zero.

### 3.3 Evolutionary Search for Sparse Feature Selection

Although sparse granular features bring abundance to construct a versatile classifier, the gigantic number of possible features consume enormous computational resources. To address this issue, an evolutionary search process is introduced to efficiently constitute a compact granular feature set. Howard *et al.* [13] implement Genetic Programming (GP) to detect ships in satellite images. Treptow and Zell [11] combine an evolutionary algorithm with the Adaboost framework to detect human faces. Abramson *et al.* [7] use a hybrid method of Hill Climbing and Evolutionary Search to detect cars. In our method, firstly we generate a large number of traditional Haar-like features in the granular space (Figure 1(c)). Through calculating the *Fitness* as Function 11, we choose  $l$  features with the highest score to construct the initial feature set  $\Theta_i$ . After many rounds of evolutionary search loop, we can harvest a large set  $\Theta_i$  with diversified granular features. The best feature is drawn out as the current weak classifier from the set.

*Fitness* evaluation of a sparse granular feature reflects the discriminability of the feature and dominates the search process. In order to improve the performance of LUT weak classifier (Section 3.2), we should find the feature with lower Bayesian error. Meanwhile, since the feature with less granulae give rise to less computational cost and simpler structural complexities, we prefer finding the features with few granule and low Bayesian error. The discriminability of a sparse granular feature is defined as  $D(\theta_i) = 1 - B_{error}(\theta_i)$ , where  $B_{error}(\theta_i)$  is the upper bound of Bayesian error. The sparse feature is more discriminative when  $D(\theta_i)$  gets higher. By considering the complexity of features, we present the *Fitness* function as

$$Fitness(\theta_i) = D(\theta_i) - \beta^c, \quad (11)$$

where  $c$  is the granule number of the sparse feature,  $\beta$  is the empirical parameter for the penalty for more granules. Generally speaking, we can preserve hundreds of granular features in each loop.

## 4 Face Fitting with Boosting Classifier

According to (10), the final classifier can be written as

$$H(\Theta, x, y; p) = \sum_{m=1}^M \frac{1}{2} \ln \sum_{k=1}^K \left\{ \frac{W_{-1}^k + \varepsilon}{W_{+1}^k + \varepsilon} \right\} B_k(\theta_m(I(W(x, y; P))), \theta_m \in \Theta. \quad (12)$$

And the derivative by  $P$  is

$$\frac{dH}{dP} = \frac{1}{2} \sum_{m=1}^M \frac{1}{M} \sum_{k=1}^K \left\{ \frac{W_{-1}^k + \varepsilon}{W_{+1}^k + \varepsilon} \right\} \nabla B_k \nabla I \nabla \theta_m \frac{\partial W}{\partial P}, \quad (13)$$

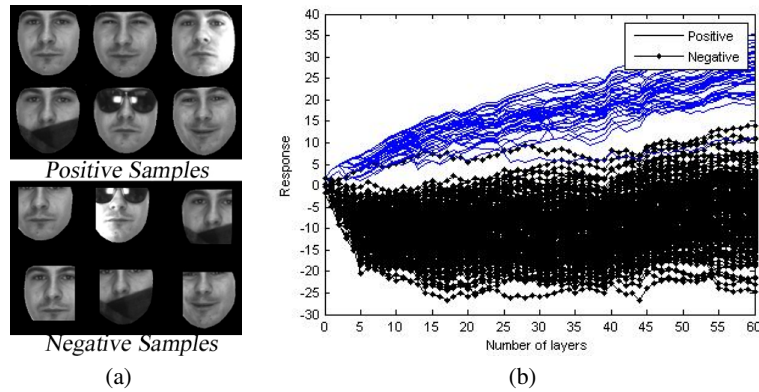
$$M = \sum_{k=1}^K \left\{ \frac{W_{-1}^k + \varepsilon}{W_{+1}^k + \varepsilon} \right\} B_k(\theta_m(I(W(x, y; P))). \quad (14)$$

**Table 2.** Weak classifier learning based on evolutionary search for sparse granular feature

---

- Input  
Training set  $\{x_i\}$ , corresponding weight set  $\{\omega_i\}$ , and Haar-like features set  $\Theta = \{\theta_1, \dots, \theta_n\}$  in the granular space.
- Initialize  
Choose better features with higher *Fitness*, add them to initial feature set  $\Theta_i = \{\theta_i | \forall \theta_i \in \Theta, \forall \theta_j \in \{\Theta - \Theta_i\}, Fitness(\theta_i) \geq Fitness(\theta_j)\}$ , constrict the set as  $\|\Theta_i\| = l$ .
- Evolutionary search loop
  - (1) To every feature  $\theta_i$  in the set  $\Theta_i$ , we implement variance in the four ways.
    - ◊ **Add.** If this granular feature contains less than eight granula, we add a new granule. All possible granulae in the granular spaces are separately added into this feature to generate new features.
    - ◊ **Delete.** Delete a granule in the current granular features.
    - ◊ **Move.** To each granule, we randomly move the coordinate between -5 and 5 pixels.
    - ◊ **Resize.** To each granule, we randomly adjust the scale  $s$  to change its size.
  - (2) After these variations, we can harvest varied granular features  $\theta_i$  to form a set  $\Theta_g$ . Then we randomly choose  $m$  features from  $\Theta_g$ , and combine them with initial feature set  $\Theta_i = \Theta_i \cup \{\theta_i | \theta_i \in \Theta_g\}$ ,  $\|\{\theta_i\}\| = m$ .
- Weak classifier learning
  - (1) To every feature  $\theta_i \in \Theta_i$ , we construct a weak classifier  $h(x, \theta_i)$ .
  - (2) Find the weak classifier  $h(x, \theta) = \arg \max_{h(x, \theta)} (Fitness(\theta_i))$ , which has the highest *Fitness*.
- Output  
The weak classifier  $h(x, \theta)$  and corresponding granular feature  $\theta$ .

---

**Fig. 2.** (a) Some positive samples and negative samples. (b) Response trace of different samples.

Face alignment factually is a process to find the best parameters  $P$  to get the best shape. After given a face image  $I$ , we firstly compute the warped  $I(W(x, y; P))$  image via a piecewise affine transformation. The face alignment algorithm is presented in Table 3.

**Table 3.** Face alignment algorithm

---

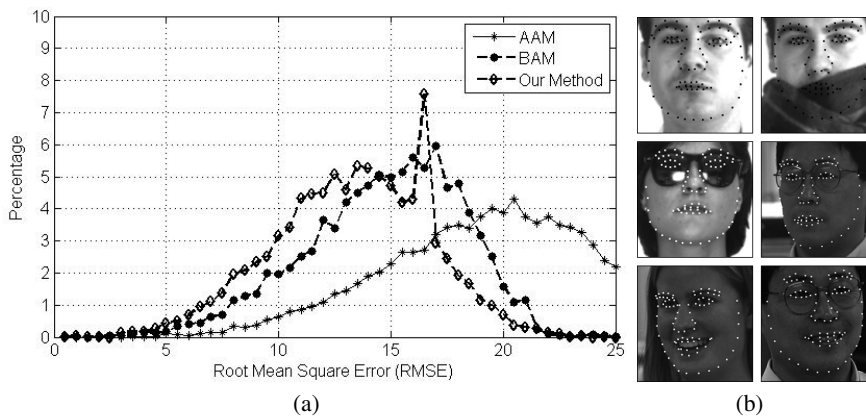
• Input
Input image $I$ , initial shape parameters $P$ , boosted strong classifier $H$ , initial response $t$ and final rejection threshold $T$ .
• While $t < T$
(1) Warp $I$ with piecewise affine transformation to generate $I(W(x, y; P))$ .
(2) Compute the current response $t$ with each classifier in 10.
(3) Compute the current $\nabla P$ by Equation 13.
(4) Update $P = P + \nabla P$ .
• Output
The shape parameters $P$ .

---

## 5 Experiments

In our experiments, we have collected about 2148 images from several databases, including the AR database [14], FERET database [15], PIE database [16]. We randomly select 1208 images for training and reserve the rest for testing. For each image in the training set, we manually label 87 points on facial components such as eyes, eyebrow, nose, mouth, *etc.* In order to train boosting classifiers, we generate 1208 positive samples. To every positive sample, we perturb parameters to generate 10 negative samples. After boosting training, we can get a classifier with eighty weak classifiers. In the process of model calibration, we choose images from the AR database. In AR database, the same person is shown in different images under various conditions. We choose 13 different conditions from the same person to calibrate the classifiers.

To test our method, we have implemented benchmark tests among Active Appearance Models (AAM), Boosting Appearance Models (BAM), and our method (Figure 3(a)). In Figure 3(a), Root Mean Square Error (RMSE) indicates the distribution of



**Fig. 3.** (a) The RMSE results among AAM, BAM, and our method on test set. (b) Face alignment results by our method.



between test results and ground-truth label. Figure 3(b) shows some face alignment results by our method.

## 6 Conclusion

In this paper, we have introduced a novel framework of face alignment, which brings in granular features, an evolutionary search process, and boosting learning process to find better weak classifiers. Since granular features produce a lot of diversified and discriminative rectangles, the boosting process has better discriminative capabilities with less weak classifiers. At the same time, we implement an evolutionary search in order to deal with deficiency of gigantic feature space. Evolutionary search not only generates versatile granular features, but also guarantees the robustness of classifiers. With granular features and evolutionary search, we can construct a novel fitting process for real time face alignment. In the future, there are more improvements that can be implemented on our approach. Firstly, other features can be added into the training system to achieve better discriminative capabilities. Secondly, calibration methodology can be used to tune the final strong classifier. Therefore, new explorations still wait for further consideration.

## Acknowledgement

This work is partially supported by the European IST Programme Project FP6-033812 (AMIDA). Many thanks to Lynn Packwood for the proof reading and many important suggestions.

## References

1. Cootes, T.F., Cooper, D.H., Taylor, C.J., Graham, J.: Trainable method of parametric shape description. *Image and Vision Computing* 10, 289–294 (1992)
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 681–685 (2001)
3. Zhou, Y., Gu, L., Zhang, H.J.: Bayesian tangent shape model: Estimating shape and pose parameters via Bayesian reference. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 109–116 (2003)
4. Lin, L., Fang, W., Ying-Qing, X., Xiaoou, T., Heung-Yeung, S.: Accurate face alignment using shape constrained Markov network. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1313–1319 (2006)
5. Liu, X.: Generic face alignment using boosted appearance model. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1–8 (2007)
6. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 671–686 (2007)
7. Abramson, Y., Moutarde, F., Steux, B., Stanculescu, B.: Combining adaboost with a hill-climbing evolutionary feature-search for efficient training of performant visual object detectors. In: *Proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence* (2006)

8. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
9. Fridman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. *The Annals of Statistics* 28, 337–374 (2000)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
11. Treptow, A., Zell, A.: Combining adaboost learning and evolutionary search to select features for real-time object detection. In: Proceedings of the Congress on Evolutionary Computation, vol. 2, pp. 2107–2113 (2004)
12. Xiao, R., Zhu, H., Sun, H., Tang, X.: Dynamic cascade for face detection. In: Proceedings of IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007)
13. Howard, D., Roberts, S.C., Brankin, R.: Evolution of ship detectors for satellite sar imagery. In: Langdon, W.B., Fogarty, T.C., Nordin, P., Poli, R. (eds.) EuroGP 1999. LNCS, vol. 1598, pp. 135–148. Springer, Heidelberg (1999)
14. Martinez, A.R., Benavente, R.: The AR face database. CVC Technical Report, vol. 24 (1998)
15. Philips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1090–1104 (2000)
16. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) database. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1615–1618 (2003)