

Robust Real-Time Face Alignment Based on ASM with Boosting Regression for Displacement Prediction

Yanchao SU¹, Haizhou AI¹, Shihong LAO²

¹ *Computer Science and Technology Department, Tsinghua University, China*

² *Sensing and Control Technology Laboratory, Omron Corporation, Japan*
ahz@mail.tsinghua.edu.cn

Abstract

Face alignment is a critical problem in many face related applications such as facial expression analysis, face recognition, etc. This paper presents a novel real time face alignment algorithm based on Active Shape Model (ASM). In our algorithm, local textures of each label point is used to predict the displacement of each label point by applying boost non-linear regressions on the local search stage of the ASM framework. Experiments on different datasets show that our algorithm is much faster in speed and more robust to the initialization than previous ASM method.

1. Introduction

Face alignment (FA) in images and videos is usually an essential preprocessing step of many face related computer vision tasks such as 3D face modeling, pose estimation, expression analysis etc. The state-of-the-art FA algorithms are mostly based on two classic methods: Active Shape Model (ASM) [1] and Active Appearance Model (AAM) [2].

ASM and AAM share the same Point Distribution Model which represents the face shape as feature points and constrains it with a PCA model, but they are different in the ways of deploying texture model. ASM characterizes the local texture around each label point with a local texture descriptor and use it to search in a neighborhood for the feature point, while AAM has a global appearance model which utilizes the holistic texture of face to conduct the fitting of the shape. AAM is robust against shape variation (expression, pose), but is sensitive to the illumination and noisy background texture due to its global texture model. And the model trained on a specific dataset is hard to

generalize to a different dataset. ASM performs more accurately on shape localization, and is relatively more robust to illumination and bad initialization, but is more sensitive to shape variation.

Recently, ASM with local texture classifiers [4][5] are proposed and show their robustness and accuracy of alignment. In their works, discriminative models like boosted classifier based on Haar-like feature rather than generative models are used to characterize the local textures and they significantly improve the accuracy of local search of the label points. But at the same time, more computation is taken by the boosted local texture classifier. And since all ASM based methods need to search in a local neighborhood for the feature point, the computation cost is in direct proportion to the search range, and therefore a proper search range should be set as a tradeoff between speed and robustness.

A latest advance in AAM is the work of Saragih et.al.[6], in which a boost non-linear regression function based on Haar-like features is used to predict the shape parameter updates. An advantage of the AAM based method is that it calculates parameter update directly from the texture instead of search in a neighborhood, so it has lower computational cost and does not have the limitation caused by the search range. Inspired by their work, we catch up an idea that similar techniques can be introduced into ASM method to have better performance in both speed and robustness since in this way the advantages of both ASM and AAM methods are combined together. Therefore, in this paper the idea of boosting non-linear regression function is applied to the local texture around each label point to predict the displacements of that label point in the local search stage of ASM framework, and then the ASM shape model is used to reconstruct a smooth shape as the initial shape of the next iteration.

Experiments show that this new method is robust to initialization and much faster than the ASM with local texture classifier.

The rest of this paper is organized as follow: in section 2 we describe the algorithm based on ASM framework with non-linear local texture regression function, section 3 shows the experiments and conclusion is given in section 4.

2. ASM with non-linear regression

As in ASM, the shape of face is represented as a set of N feature points (x_i, y_i) concatenated as a shape vector $s=(x_1, y_1, x_2, y_2, \dots, x_N, y_N)$. A PCA model of the shape vector is built on a set of manually labeled training samples in order to reduce the dimension of the model:

$$s = U \cdot p + \varepsilon \quad (1)$$

Ignoring the Gaussian noise ε , the shape of face can be represented using the shape parameter p .

In the algorithm, while initial shape is specified, boost non-linear regression functions are applied to the local textures at the label points to predict the displacements of the label points and find the new shape. Then the shape model is used to reconstruct the predicted shape so as to eliminate the outliers and guarantee a smooth shape.

2.1. Non-linear regressions on local textures

In classic ASM framework, local descriptors or local classifiers at label points are used to search in a neighborhood around the initial locations of label points to find the best displacements of the points by calculating the fitness of each pixel in the search range. In our algorithm, the non-linear regression functions on local textures are used to predict the displacements of the label points.

Given a manually labeled dataset, we could randomly sample around each label point to generate the samples $\{(\Delta x_i, \Delta y_i, I_i)\}$ as the training set for boosting regression learning. The non-linear regression function of each label point i is formed as the weighted sum of a set of weak regression functions:

$$(\Delta x_i, \Delta y_i)^T = F_i(I_i) = \sum_k \omega_i^k f_i^k(I_i) \quad (2)$$

The boosting regression learning starts from an empty set, one weak regression function is added in each iteration:

$$F_i^{k+1}(I_i) = F_i^k(I_i) + \omega_i^{k+1} f_i^{k+1}(I_i) \quad (3)$$

where the pair $(\omega_i^{k+1}, f_i^{k+1})$ is chosen by minimizing the total regression error on all training samples:

$$\begin{aligned} & (\omega_i^{k+1}, f_i^{k+1}) \\ & = \arg \min \sum \left\| (\Delta x_i^k, \Delta y_i^k)^T - \omega_i^{k+1} f_i^{k+1}(I) \right\| \quad (4) \end{aligned}$$

The set of weak regression functions will be described later in section 2.2, and the weight ω_i^{k+1} can be calculated analytically.

The training procedure is described in Algorithm 1.

Algorithm 1 Training of A Regression Function a Label Point

Input: a set of N_s training samples $\{(\Delta x_i, \Delta y_i, I)\}$

a set of N_f features

Start:

$F_i = 0$

for $k = 0$ to N

$E^* = 0, f_i^{k*} = 0, \omega_i^{k*} = 0$

for $j = 1$ to N_f

learn weak function $f_i^{k,j}$ base on the j th

feature

calculate the weight $\omega_i^{k,j}$

calculate the regression error E

if $E < E^*$

$f_i^{k*} = f_i^{k,j}, \omega_i^{k*} = \omega_i^{k,j}$

end if

end for

$F_i = F_i + \omega_i^{k*} f_i^{k*}$

for $j = 1$ to N_s

$(\Delta x_i^j, \Delta y_i^j) = (\Delta x_i^j, \Delta y_i^j) - \omega_i^{k*} f_i^{k*}$

end for

end for

2.2. Weak regression functions

The regression problem of predicting displacements from the texture can be solved in various ways such as linear regression, CCA [7] etc. But in fact, the variations of textures are not linear to the displacements, so we have to adopt some non-linear regression methods to solve this problem.

The weak regression function should have two characteristics: low computational cost and sufficient regression power. Among all kinds of image features, Haar-like feature has achieved great success in face detection [8] and face alignment [5]. So we use Haar-like feature as the basis of weak regression functions to map the local texture to a scalar feature value $h(I)$. And the displacements are calculated from the feature value $h(I)$ by a weak regression function $L(\cdot)$:

$$(\Delta x_i, \Delta y_i)^T = f_i^k(I) = L(h(I)) \quad (5)$$

A popular choice of L is the look up table (LUT) which can approximate any kind of the non-linear functions at low cost.

The structure of the weak function is shown in figure 1:

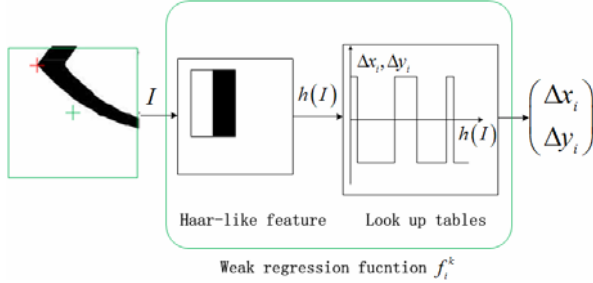


Figure 1: Structure of weak regression functions

Given a local patch, we can compute the value of a Haar-like feature, and then from a look up table (LUT) to find the displacement Δx_i or Δy_i .

For simplicity, we use binary LUT as the function L , which means that each weak regression function only predict the direction of the displacement Δx_i or Δy_i , and the magnitude is determined by the weight of this weak function.

In the training procedure of the LUT, first the histogram of feature values of all the training samples is calculated:

$$H_x(v) = \sum_{h(I)=v} \Delta x, \quad H_y(v) = \sum_{h(I)=v} \Delta y \quad (6)$$

$H_x(v), H_y(v)$ characterizes the expectation of displacements of those training samples with the same feature value v .

Then the value in each bin of the LUT is determined by the sign of the histogram:

$$L(v) = \begin{pmatrix} \text{sign}(H_x(v)) \\ \text{sign}(H_y(v)) \end{pmatrix} \quad (7)$$

2.3. ASM fitting procedure

Given a face photo, the face can be detected using a face detector [10], and then an initial shape can be estimated from the bounding box of the face. Followed by the fitting algorithm goes as follow:

- (1). Step 1, label point update: for each label point, with the regression function described in section 2.1, we can predict the displacement of each label point from the local texture around the initial location and then use the displacement to update

the location of the label point.

- (2). Step 2, parameter estimation: use the shape model to estimate the pose parameter and the shape parameter, then reconstruct the shape.
- (3). Step 1 and step 2 are iterated until the shape converged.

3. Experiments

Experiments on two different datasets are reported here to compare our algorithm with previous ASM method that with local texture classifier [5].

3.1. Datasets

The experiments are performed on two different datasets. The two datasets all have face images with manually labeled feature points as ground truth. The first one have 1836 images and the second one has 600 images which is selected from the ba, be and bf categories of the FERET [9] dataset.

In the experiments, we use 1434 images from the first dataset as the training set, and the rest 402 images are used as testing set 1. The second dataset is testing set 2.

3.2. Training of models

In our experiment, we use 88 label points to represent the face shape. The PCA shape model is built as the original ASM [1].

In the training of non-linear local texture regression functions, for each label point, we first sample randomly around the label points for the texture patches, then train a non-linear regression function for this label point with the algorithm described in section 2. The size of texture patches (local neighborhood) is set to 24×24 . For each label point, we train a non-linear regression function with 100 weak functions.

3.3. Results

To compare our algorithm with previous method, we run the alignment algorithm and the previous method automatically on testing set 1 and 2, and the alignment performance is measured by the point-to-point errors between the alignment result and the ground truth as shown in figure 2(a)(b). And figure 2(c) shows the results on testing set 1 when the initial shape is perturbed randomly within ± 10 pixels. In average our algorithm takes about 35ms to align a face while previous method needs about 203ms. We can see that

our algorithm achieves comparable accuracy and robustness with a speed that is about 6 times faster.

Figure 3 shows some examples of alignment results. Notice that our algorithm works well even if the initialization is bad and the face is with exaggerated expressions.

4. Conclusion

In this paper, we proposed an ASM based face alignment algorithm which use non-linear local texture regression function to calculate the displacements of each label point. Comparing to previous ASM based method with local texture classifiers, our algorithm is robust to bad initialization and is significantly much faster that will be very promising for real-time applications.

5. Acknowledgement

This work is supported by National Science Foundation of China under grant No.60673107, and it is also supported by a grant from Omron Corporation.

References

- [1] A. Hill, T. F. Cootes, and C. J. Taylor. Active shape models and the shape approximation problem. In 6th British Machine Vision Conference, pages 157-166. Sept. 1995.
- [2] T. F. Cootes. Statistical models of appearance for computer vision. <http://www.isbe.man.ac.uk/~bim/refs.html>, Sept. 2001.
- [3] F. Zuo, Peter H.N. de With. Fast facial feature extraction using a deformable shape model with Haar-wavelet based local texture attributes. ICIP 2004.
- [4] S. Yan, M. Li, et.al. Ranking prior likelihood distributions for Bayesian shape localization framework. ICCV 2003.
- [5] L. Zhang, H. Ai, et.al. Robust Face Alignment Based on Local Texture Classifiers. ICIP 2005.
- [6] J. Saragih, R. Goecke. A Nonlinear Discriminative Approach to AAM Fitting. ICCV 2007
- [7] Georg Langs, Philipp Peloschek, et.al. Active Feature Model. ICPR 2006
- [8] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. CVPR 2001
- [9] P.J. Phillips, et.al. The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing J (1998)
- [10] C. Huang, H. Ai, et.al. Boosting Nested Cascade Detector for Multi-View Face Detection. ICPR 2004.

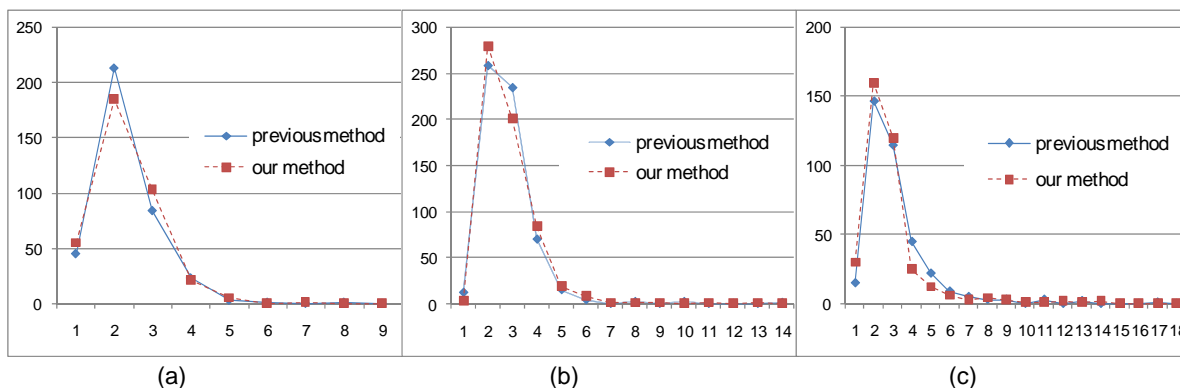


Figure 2. Point-to-point error distributions of the alignment result

- (a) On testing set 1; (b) On testing set 2; (c) Result on testing set 1 with the randomly disturbance within ± 10 pixels on the initial shape.

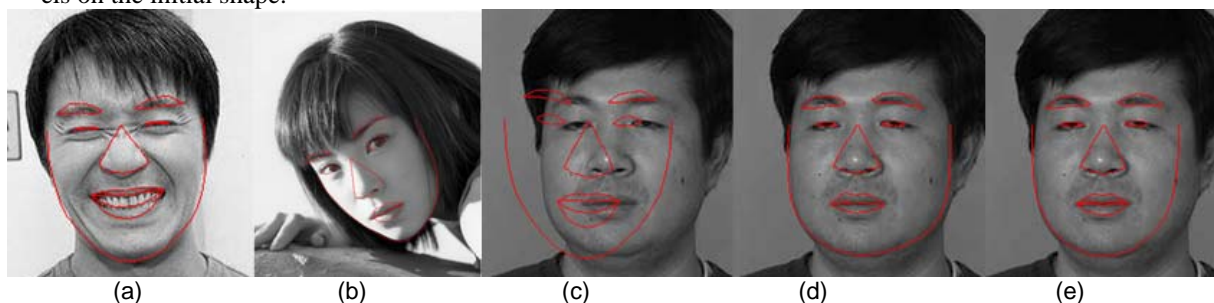


Figure 3. Some alignment results on images from the Internet and the FERET database

- (a)(b) Results on faces with exaggerated expression and rotation
 (d) Result achieved with bad shape initialization (c), while (e) is by the previous method.