Component Localization in Face Alignment

Yanyun Qu [1], Tianzhu Fang [2], Yanyun Cheng [3], Han Liu [4]
Xiamen University
Computer Science Department
Xiamen, China

***Abstract*—Face alignment is a significant problem in the processing of face image, and Active Shape Model (ASM) is a popular technology for this problem. However, the initiation of the alignment strongly affects the performance of ASM. If the initiation of alignment is bad, the iteration of ASM optimization will be stuck in a local minima, and the alignment will fail. In this paper, we propose a novel approach to improve ASM by building the classifiers of the face components. We design the SVM classifiers for eyes, mouth and nose, and we use Speeded Up Robust Features(SURF) and Local Binary Pattern(LBP) feature to describe the components which are discriminative for the components than Haar-like features. The face components are firstly located by the classifiers and they indicate the initiation of the alignment. Our approach can make the iterations of ASM optimization converge fast and with the less errors. We evaluate our approach on the frontal views of upright faces of IMM dataset. The experimental results have shown that our approach outperforms the original ASM in terms of efficiency and accuracy.**

Keywords-face alignment; ASM; component localization; LBP; SURF

I. INTRODUCTION

Face alignment has been widely used in computer vision, such as object detection, tracking, alignment, and etc. Face alignment aims to deform a face model to match it with the features of the image of a face by optimizing an appropriate cost function. It is essentially an image registration problem. Face alignment is a challenging problem due to the face variation on pose, illumination and expression, as well as the occlusion. There are many works related to image alignment. Kass et al [1] proposed Active Contour Models (ACM) in 1987. Cootes and Tayloy [2] proposed Active Shape Model (ASM) in 1994, which is one of the early and popular approaches that attempt to fit the model on data, and that is a generative model based on statistical computation.

In order to fit the shape to a face image robustly, many methods of face alignment combined the discriminative model with ASM [3,4,5,6,7]. These methods include three key factors: the description of the appearance shape, the design of the optimize function, and the search mechanism. In this paper, we pay attention to the search mechanism, especially to the initiation of the landmarks, because of that ASM is sensitive to the initial alignment, if the initial alignment is bad, the iteration of ASM optimization will be stuck in a local minima, and ASM will fail. In order to obtain the good initial alignment, we combine the detections of face components with ASM, and locate the initial landmarks according to the component locations.

How to describe the face components is a critical factor for the localization of face components. Haar-like features [8] are often used to represent the face appearance, and Viola et al. [9] have demonstrated that Adaboost classifier based on the Haar-like features are successful in face detection. However, we find that Haar-like features are not powerful in representing the face components, so we use Speeded Up Robust Features (SURF) and Local Binary Pattern (LBP) feature to describe the face components, which are more distinctive than Haar-like features. One advantage of our method is that it locates the face components appropriately; the other advantage is that it makes the initiation of alignment good for ASM.

Later we use extensive experiments to show that this framework improves the robustness, accuracy and efficiency, compared with the original ASM, especially for the new data.

The rest of this paper is organized as follows. Section 2 briefly introduces the ASM. In Section 3, the description of the component detector and its experimental evaluation are provided. Section 4 describes our approach. The experiment results in Section 5 shows the advantages of our approach. In section 6, we draw the conclusions.

II. ACTIVE SHAPE MODEL

Given a face image $E=\{(x,y)\in R^2\}$, the aim of face alignment is to find N landmark points to characterize it, which can be expressed as $p_i=(x_i,y_i), i=1,2,...,N$. In the face dataset we used, each face image is manually labeled 58 landmarks, which are distributed in the face contour, the eyebrows, the eyes, the nose and the lip. We define a feature vector for the i-th face image as $f_i=(x_{i1},y_{i1},x_{i2},y_{i2},\cdots,x_{iN},y_{iN})^T$. Hence, the M training images form a matrix $F=[f_1,f_2,\cdots,f_M]$. Next, making use of principal component analysis (PCA), we can get a parametric model about the face shape, which is expressed as

$$S(b)=T(\bar{f}+\Phi b) \tag{1}$$

The research work was supported by the National Basic Research Program of China under Grant No. 2007CB311005, the Fundamental Research Funds for the Central Universities, No.2010121067, and National Defense Basic Scientific Research program of China under Grant B1420110155.

where $b=[b_1,b_2,\cdots,b_t]^T$ is the shape parameter controlling the shape change; $\overline{f}$ is the average shape of all the training face images,

$$\overline{f}=\frac{1}{m}\sum_{i=1}^{m} f_i \tag{2}$$

$\Phi=(\phi_1|\phi_2|...|\phi_t)$ is the projection matrix obtained by PCA which consists of the first t eigenvectors that are corresponding to the first t eigenvalues of the covariance matrix of F. ASM synthesizes new shape by adjusting the parameter b, for the human face shape, b satisfies the constraints $-3\sqrt{\lambda_k}\le b_k\le 3\sqrt{\lambda_k}$, where λ_k is the k-th largest eigenvalue of the covariance matrix of F. From (1) we have,

$$b=\Phi^{-1}(T^{-1}(S(b))-\overline{f}) \tag{3}$$

In order to find the best matching point for each landmark point, a Point Distribution Model is used to capture the shape variants in ASM, and each landmark has a point distribution model. We choose l points with an equal interval in the direction of profile at each landmark point and compute their first-order derivatives as g_1, g_2, …, g_l. The quality of fitting a gradient profile g_i at the location i of a query image to the j-th model is computed as the Mahalanobis distance,

$$f(g_i)=(g_i-\overline{g}_j)^T g_{Cov}^{-1}(g_i-\overline{g}_j) \tag{4}$$

where $\overline{g}_j$ is the mean of the profile in the model of the j-th landmark and g_{cov} is the covariance matrix along the profile in the model of the j-th landmark.

Given a query face image, the algorithm of ASM is shown as follows, more details see [2]:

- Step 1. Initialize with the mean shape $\overline{S}$.
- Step 2. Start the coarsest resolution level.
- Step 3. For each landmark, compute the Mahalanobis distance for each point at the profile, and then move landmark to the position where the Mahalanobis distance is the minimum.
- Step 4. Fit the shape model to the displaced landmarks via (3) and (1).
- Step 5. Iterate steps 3 and 4 until the process converges.
- Step 6. If the current resolution is similar to the previous, the iteration is stopped, otherwise, go to Step 3.

III. COMPOMENT DETECTOR

Major prior works of face alignment is using Haar-like features to describe the face appearance in their discriminative models. However, the Haar-likes feature is not suitable for the component description. In our experiment, the detection rate of Haar-like features is as high as the error rate. Besides, the Haar-like features with the Adaboost classifier do not localize the component accurately. As shown in Fig. 1, Adaboost classifier based on Haar-like features does not localize the nose in a tight bounding box, but we need to precisely calibrate the target location just like in Fig. 1(c). Therefore, the Haar-like features do not meet our requirements. So we have to consider more discriminative descriptors just like SURF features and LBP features to describe the face components.

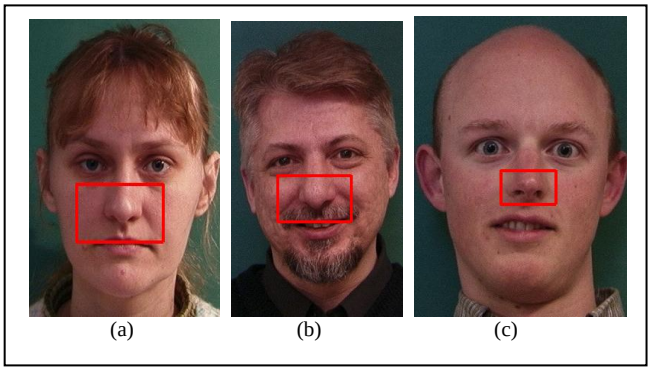


Figure 1. Some results detected by Haar-like features. (a)(b) the result of Haar-like features detection. (c) the ground truth label.

A. SURF

SURF is a local feature descriptor. There are two parts in SURF description: one is to detect the interesting points, and the other is to describe the interesting points by SURF. We detail it as follows.

Firstly, we randomly sample points in the component patch, and then construct a circular region around the sampled points. We compute the dominant orientation for the sampled points and describe the sampled patches by the invariant local feature. The orientation is computed using Haar wavelet, and responses in both x and y directions. The dominant orientation is estimated and included in the interest point information.

Secondly, SURF descriptors are constructed by extracting square regions around the interest points, which are oriented in the directions assigned in the previous step. The windows are split up in 4×4 sub-regions in order to retain some spatial information. In each sub-region, Haar wavelets are extracted at regularly spaced sample points. The wavelet responses in horizontal and vertical directions (d_x and d_y) are summed up over each sub-region. Furthermore, the absolute values $|d_x|$ and $|d_y|$ are summed in order to obtain information about the polarity of the image intensity changes. Hence, the underlying intensity pattern of each sub-region is described by a vector $\mathbf{V}=\left[\Sigma d_x,\Sigma d_y,\Sigma|d_x|,\Sigma|d_y|\right]$. Therefore, we obtain the SURF feature which is a vector of 64 dimensions. Furthermore,

we can obtain the more discriminative features if we compute the sum of d_x under the condition $d_y \geq 0$ and $d_y < 0$ and do the same operation for $|d_x|$ and do the similar operation for d_y and $|d_y|$. This results in a descriptor vector for all 4×4 sub-regions of 128 dimensions. Fig. 2 shows a simple process for SURF. See more details in [10].

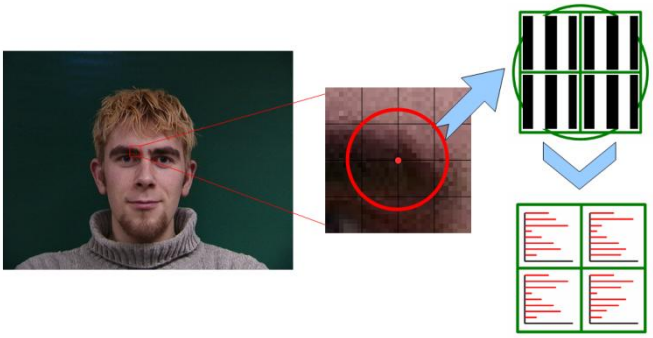


Figure 2. A simple process for SURF.

Finally, we cluster the feature vectors through k-means algorithm, thus the clustering center could be regard as a word. We represent the vectors of these component patches as histograms by the histograms of words, and use SVM in [11] for training. In this way, we get a SVM classifier based on the SURF features.

B. Local Binary Pattern

The Local Binary Pattern (LBP) operator was introduced by Ojala et al [12]. The operator labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value and considering the results as a binary number. The 256-bin histogram of the labels computed over a region can be used as a texture descriptor. Each bin can be regarded as a micro-texture. Later the operator was extended to use neighborhood of different sizes using circular neighborhoods [13].

The $LBP_{P,R}$ operator produces 2^P different output values, corresponding to the different binary patterns that can be formed by the P pixels in the neighbor set. It has been shown that certain bins contain more information than others [13]. Ojala et al. called these fundamental patterns uniform patterns. Fig. 3 shows an example for LBP.

After representing the face components by LBP features, we also carry on SVM [11] to train them, thus obtain a classifier.

C. Evaluation of Component Recognition

We compared SURF and LBP with Haar-like features on the performance of the component recognition.

We use the Adaboost classifier to detect face components, which are described by Haar-like feature. As comparison, we apply the SVM classifier with Gaussian kernel to detect eyes and noses which are described by LBP feature, besides, we also use the SVM classifier to detect eyes and mouths which are described by SURF.

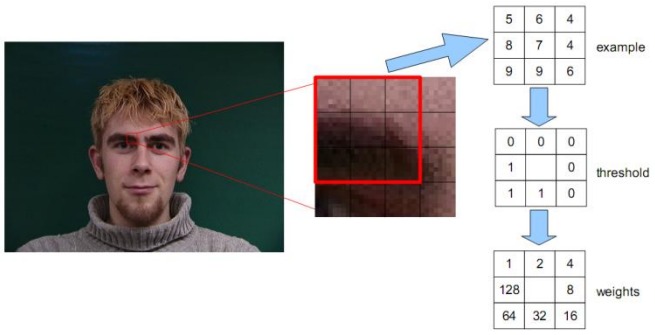


Figure 3. An example for LBP.

As shown in Table I, SURF is not sensitive to the noses features (possibly due to noses were at little difference with the surrounding texture and the edge features were relatively few), and LBP is not sensitive to the lip features (mainly due to the beard shading and the teeth impact), they still have the satisfying results: for nose, the error rate by LBP classifier is much smaller than the one by Haar-like classifier, and for mouth, the error rate by SURF classifier is much smaller than the one by Haar-like classifier, and for eyes the error rates by both the two mentioned classifier are much smaller than Haar-like classifier. Generally speaking, SURF and LBP are more distinctive than Haar-like features in describing components. And we do not show the detection rate of mouths by LBP classifier and that of noses by SURF classifier, due to their low detection rate.

TABLE I. THE DETECTION RATES OF SOME CLASSIFIERS

Component		Eyes	Nose	Mouth
Haar-like	Error Rate	24.17%	100%	100%
	Miss Rate	2.08%	15.83%	0%
LBP	Error Rate	1.6%	2.96%	
	Miss Rate	43.75%	6.67%	
SURF	Error Rate	8.33%		12.5%
	Miss Rate	4.58%		13.75%

IV. ALGORITHM OF FACE ALIGNMENT

According to the description in Section II, the original ASM can be described as

$$S_0 = \bar{S};$$

$$S_{t+1} = \{ g' \mid f(g') = min\, f(g_i),\ g_i \in \{profiles\ in\ points\ g\},\ g \in S_t \}$$

where $f(g_i) = (g_i - \bar{g}) g_{Cov}^{-1} (g_i - \bar{g})$ and S_t denotes the shape state of the t-th time.

Face alignment is sensitive to the initial shape S_0. And all points in S_t are searched by adjusting S_0. If the initial landmarks are put in the suitable localization, the shape would aims at the initial points localization through component recognition. The flow chart of our approach is shown in Fig. 4.

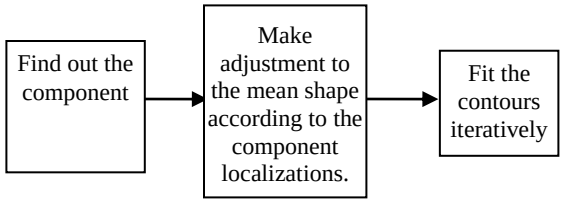


Figure 4. The flow chart of our approach.

A. *Consructing the training dataset*

As we observe, the important components of a face are the eyes, the mouth and the nose. So we train the three component detectors.

We crop the component patches from the IMM dataset as the positive samples and cropped other patches as negative samples. In the eyes training set, there are 37 positive samples and 178 negative samples; in the nose training set, there are 37 positive samples and 124 negative samples; and in the mouth training set, there are 37 positive samples and 88 negative samples. Some examples are given in Fig. 5 and Fig. 6.

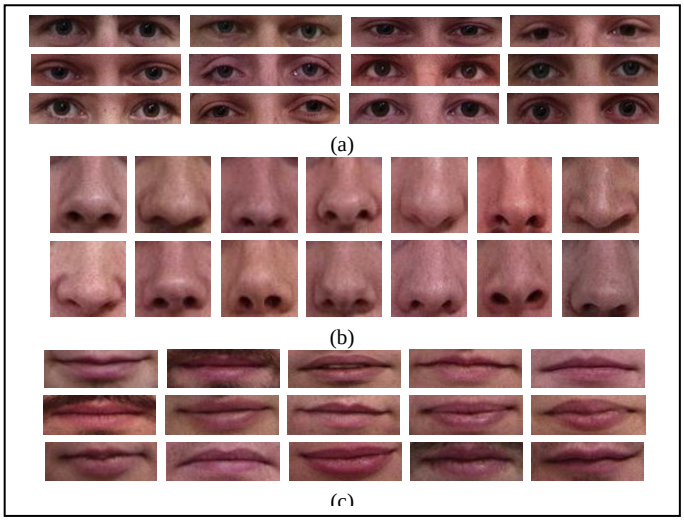


Figure 5. Some positive examples used in component detection. (a) the samples for eyes, (b) the samples for noses, (c) the samples for mouths.

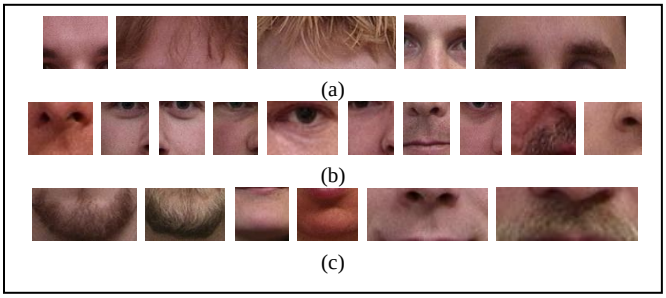


Figure 6. Some negative examples used in component detection. (a) the negative samples for the eyes training, (b) the negative samples for the nose training, (c) the negative samples for the mouth training.

B. *ASM Algorithm*

1) *Training:*

- Step 1: Select m face images which have been annotated manually using 58 landmarks around the eyebrows, the eyes, the nose, the mouth and the jaw, and then build the shape model:

$$S(b) = T(\bar{f} + \Phi b)$$

- Step 2: For each landmark, generate sample *l* points along the profile of the landmark, and build the gray-level appearance model.
- Step 3: Train the three classifiers for the eyes, the nose and the mouth.

2) *Fitting:*

- Step 1: Detect the three components mentioned above and put the initial landmarks in suitable points, and adjust the mean shape.
- Step 2: Make the adjusted mean shape as iterative starting shape.
- Step 3: For each landmark, computer the Mahalanobis distance and find the position which has minimum Mahalanobis distance.
- Step 4:. Fit the shape model to displaced landmarks via (3) and (1).
- Step 5: Iterate steps 3 and 4 until the solution is convergence.
- Step 6: If the current resolution is similar to the previous, the iteration terminate; otherwise, goto Step 3.

V. EXPERIMENTAL RESULTS

We implement our method on IMM dataset with PIII1.8GHZ processor and 256MB memory. We used 40 upright faces in the dataset for training and the others 80 upright faces for testing.

In addition, we use two criteria to evaluate our approach, the average frequency of convergence (AFC) given by the number of trials where the alignment converges divided by the total number of trials and the mean square error (RMSE). And the AFC is defined as

$$c = \sum_{i=1}^{n} c_i \Big/ n$$

where c_i denotes the convergence frequencies of the i-th image.

The RMSE is defined as

$$Err = \sqrt{\frac{\sum_{i=1}^{n}((x'_i - x_i)^2 + (y'_i - y_i)^2)}{n-1}}$$

where $\{(x'_i, y'_i)\}$ denotes the shape computed by our approach and $\{(x_i, y_i)\}$ denotes the ground truth shape.

In this paper, we have 58 landmark points for each face image, that is, n = 58. We considered two factors which influence our approach performance: the number of the face components and the descriptor. We compared the five methods: (1) the original ASM, (2) the eye detector with LBP feature combined with ASM, (3) the detectors for the eyes and the nose based on LBP feature combined with ASM, (4) the eye detector with SURF combined with ASM, (5) the detectors for the eyes and the nose with SURF combined with ASM. The result is shown in Table II. It demonstrates that the component detection can improve the AFC and RMSE of ASM, and the more components are combined with ASM, the better accuracy the ASM is. Furthermore, the components detector with SURF is superior to those with LBP in the fitting accuracy.

Fig. 7 and Fig. 8 show some results in or outside the training set. The results demonstrate that our approach is stable and can improve the accuracy of shape fitting in ASM.

However, we also have some failed cases, just as shown in Fig. 9. We analyze the failure cases and find that the faces have large rotation in the failure cases, while our approach is only suitable to the upright faces.

TABLE II. THE GENERAL SITUATION OF THE EXPERIMENT

METHOD	AFC	RMSE
ASM	1150.158	44.83577
Eyes + LBP	842.5444	9.600739
Eyes + Nose + LBP	935.5402	9.30989
Eyes + SURF	1044.188	11.4465
Eyes + Nose + SURF	1068.383	9.115111

VI. CONCLUSIONS

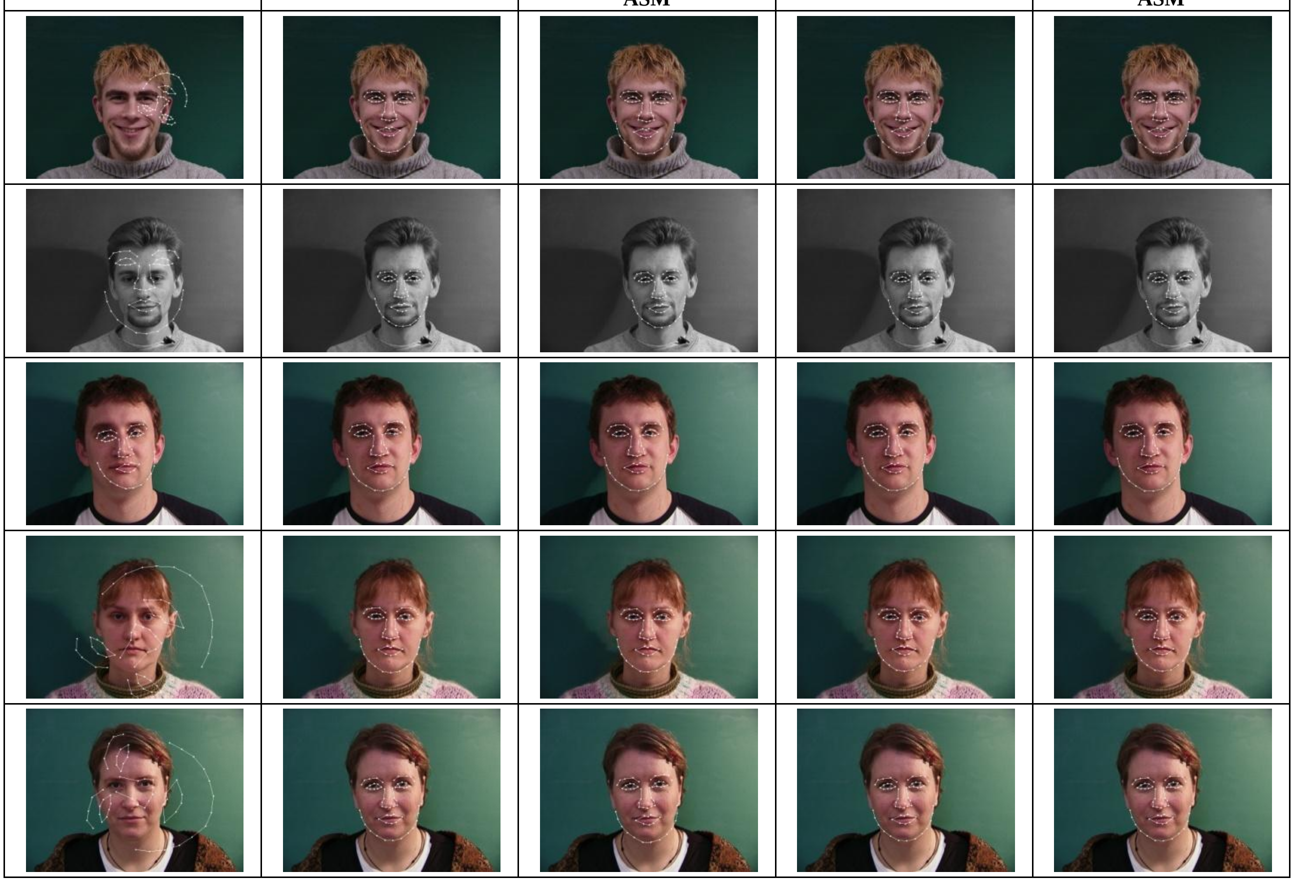
In this paper, we have proposed an improved approach, which combines the component learning with ASM for face alignment. The experiment results on IMM dataset have shown that component localization speeds up the convergence of ASM iterations, and the results in matching the face shape is more accurate than the original ASM. In the future, we will apply the symmetrical information in the face alignment, which may reduce the time complexity of ASM and improve the results fitting the template on the face image.

Figure 9. Some failure examples.

ASM	Describe Eyes Features Using LBP in ASM	Describe Eyes and Nose Features Using LBP in ASM	Describe Eyes Features Using SURF in ASM	Describe Eyes and Mouth Features Using SURF in ASM

Figure 7. Some examples on the training set based on component localization combined with ASM vs original ASM.

ASM	Describe Eyes Features Using LBP in ASM	Describe Eyes and Nose Features Using LBP in ASM	Describe Eyes Features Using SURF in ASM	Describe Eyes and Mouth Features Using SURF in ASM



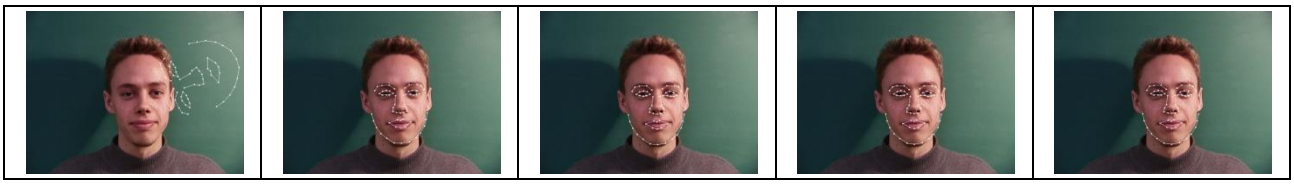


Figure 8. Some examples in the testing set based on component localization combined with ASM vs original ASM.

REFERENCES

[1] M. Kass, A. Witkin and D. Terzopoulos. "Active Contour Models," The 1st ICCV, London, UK, 1987.

[2] T.F. Cootes, C.J. Taylor,D.H. Cooper, and J. Graham. "Active Shape Models – Their Training and Application," Computer Vision and Image Understanding, 1995, 61(1): 38-59.

[3] S.C. Yan, C.Liu, S.Z.Li. H.J.Zhang, H. Shum, Q.S.Cheng. "Face Alignment Using Texture-Constrained Active Shape Models," Imageand Vision Computing, 2003, 21(1): 69-75.

[4] B.V. Ginneken, A.F. Frangi etal. "A Non-linear Gray-level Appearance Model Improves Active Shape Model Segmentation," IEEE Workshop on Math Models in Biomedi Hawaii, USA, 2001.

[5] C Du, Q Wu, J Yang, Z Wu. "SVM based ASM for facial landmarks location," IEEE 8th International Conference on Computer and Information Technology, 2008.

[6] Y. Li and W. Ito. "Shape parameter optimization for adaboosted active shape model," In Proc. 10th ICCV, 2005.

[7] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component based discriminative search. In ECCV, 2008.

[8] Papageorgiou, Oren and Poggio, "A general framework for object detection", International Conference on Computer Vision, 1998.

[9] Viola and Jones, "Rapid object detection using boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.

[10] H. Bay, T. Tuytelaars, and L. Van Gool. "Surf: Speeded up robust features," The 9th European Conference on Computer Vision, 2006.

[11] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM – A Library for Support Vector Machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[12] T. Ojala, M Pietikinen, and D. Harwood. "A comparative study of texture measures with classification based on featured distribution," Pattern Recognition, 1996, 29(1):51–59.

[13] T. Ojala, M. Pietikinen, and T. Menp. "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7):971–987.

AUTHORS PROFILE

Yanyun Qu, received her B.Sc.and M.Sc. degrees in Computational Mathematics from Xiamen University and Fudan University, China, in 1995 and 1998, respectively, and received her Ph.D. degrees in Automatic Control from Xi'an Jiaotong University, China, in 2006. She joined the faculty of Department of Computer Science in Xiamen University since 1998. She was appointed as a lecturer from 2000 to 2007 and was appointed as an associate professor since 2007. She is a member of the IEEE. Her current research interests include pattern recognition, computer vision, image/video processing, machine learning, etc.

Tianzhu Fang, born in 1987, is currently a graduate at Xiamen University. He received his B.Sc. degree in Software Engineering from Fuzhou University in 2009. His research interests largely lie in the areas of pattern recognition, machine learning, computer vision and related areas.

Yanyun Cheng, born in 1985, is currently a graduate at Xiamen University. She was awarded a B.Sc. in Computer Science and Technology Department, Xiamen University in 2008. Her research interests largely lie in the areas of pattern recognition, machine learning, computer vision and related areas.

Han Liu, born in 1985, is currently a graduate at Xiamen University. He was awarded a B.Sc. in Mathematical Science, Xiamen University in 2008. His research interests largely lie in the areas of pattern recognition, machine learning, computer vision and related areas.