

# Face Alignment via Component-based Discriminative Search

Lin Liang    Rong Xiao    Fang Wen    Jian Sun

Microsoft Research Asia  
Beijing, China  
{lliang,rxiao,fangwen,jiansun}@microsoft.com

**Abstract.** In this paper, we propose a component-based discriminative approach for face alignment without requiring initialization<sup>1</sup>. Unlike many approaches which locally optimize in a small range, our approach searches the face shape in a large range at the component level by a *discriminative search* algorithm. Specifically, a set of *direction classifiers* guide the search of the configurations of facial components among multiple detected modes of facial components. The direction classifiers are learned using a large number of aligned local patches and misaligned local patches from the training data. The discriminative search is extremely effective and able to find very good alignment results only in a few (2~3) search iterations. As the new approach gives excellent alignment results on the commonly used datasets (e.g., AR [18], FERET [21]) created under-controlled conditions, we evaluate our approach on a more challenging dataset containing over 1,700 well-labeled facial images with a large range of variations in pose, lighting, expression, and background. The experimental results show the superiority of our approach on both accuracy and efficiency.

## 1 Introduction

Fitting a shape or template to a given image has been studied many years in computer vision [27, 2]. The goal of face alignment is to match a 2D face shape or appearance with a facial image. Applications of face alignment range from face recognition [26], face tracking, facial expression analysis, 3D face modeling, to face cartoon animation [3]. Today, with the explosively increasing of face photos on the web and the desktop, new applications such as face search and annotation [10] have raised new requirements for the face alignment: fully automatic, efficient, and robust to facial images in under-controlled conditions.

Active Shape Model (ASM) [6] and Active Appearance Model (AAM) [4] are two most representative face alignment models. In ASM [6], a Point Distribution model captures the shape variants and gradient distributions of a set of landmark points describe the local appearance. The shape parameters are iteratively updated by locally finding the best nearby match for each landmark point. In AAM [4], the appearance is modeled globally by PCA on the mean shape coordinates (also called “shape-normalized frame”). The shape parameters are locally searched using a linear regression function on

---

<sup>1</sup> We assume that a very rough face location is given by a face detector.

the texture residual. Many variants of ASM or AAM have been proposed, for example, Bayesian Tangent Shape Model (BTSM) [28], Mixture or non-linear shape model [7, 22], View-based AAM [8], Direct Appearance Model (DAM) [12], Hierarchical DDM-CMC [16], Constrained Markov Network [15], Inverse Compositional updating [19], 3D face alignment [11], Boosted Appearance Model (BAM) [17], and Boosted regression [9].

However, the models described above require good initialization because most of them are based on gradient descent or local search from a rough initial shape. The objective functions of these models might have a large number of local minimums in the high dimensional solution space. Even using the coarse-to-fine strategy, these approaches often get stuck into a local minimum if the initial shape is far from the correct solution, especially on the facial images in real applications.

A natural solution for the initialization problem is sampling - starting from multiple initial shapes. For instance, hierarchical CONDENSATION [24] uses 50 samples in each iteration and three-level pyramid to make the result less dependent on the initialization. The sampling method reduces the sensitivity to the initialization but is computational expensive. Moreover, it is unclear that how many samples are sufficient to cover the correct solution.

In this paper, we propose a component-based discriminative approach to address the efficiency and robustness problems. Our approach searches the face shape at the facial component level. We first detect a number of candidate *modes* for facial components (e.g., eyes, nose, mouth, and profiles) and construct an initial shape. Then, a *discriminative search* algorithm searches a new position for each facial component. The searching direction is determined by learned *direction classifiers*, and the searching step is adaptively set according to the candidate modes. For each facial component, nine direction classifiers (eight directions + one of “no move”) are trained on aligned and misaligned local patches from the training data in the mean shape coordinates. Using our discriminative search among candidate modes, better configurations of facial components can be effectively and rapidly discovered in very large searching range. The shape is further refined using the components as constraints. Our experiments demonstrate that we are able to visit very good fitted shape only within a few (usually 2-3) iterations.

Our approach is unique in several aspects: we optimize the shape at the component level among candidate modes so that we have the ability to jump out the local minimums; we learn how to search facial components from the training data. The direction classifier is trained on a large number of positive examples and negative examples; it is fully automatic without requiring initialization; it is a very efficient deterministic algorithm which significantly outperforms the sampling based approach on the accuracy, and is much faster on the speed.

The discriminative approach has been applied to in the face alignment [17, 9] and visual tracking [1, 20]. Our proposal differs from these works in that: 1) we train the classifier to predict the searching direction. It is more robust than directly predicting the exact position; 2) we design an searching algorithm which can effectively utilize the multi-modes, noisy, and incomplete observations (component detection results).

## 2 Problem Formulation

**Notation.** Given an image  $I$ , the task of face alignment is to locate a number of  $N$  facial feature points  $\mathbf{S}_f = \{s_i^f\}_{i=1}^N$  on the 2D image. In this work, we also define  $C$  facial component points  $\mathbf{S}_c = \{s_i^c\}_{i=1}^C$ . Figure 2 shows 11 facial components (eyes, brows, nose, mouth, upper/lower lips, left/right/lower profiles) we used.

Applying PCA analysis on a set of training shapes in the tangent shape space [28], the face shape  $\mathbf{S} = [\mathbf{S}_f, \mathbf{S}_c]$  can be expressed as:

$$\mathbf{S}(\mathbf{p}) = T(\bar{\mathbf{x}} + \mathbf{Q}\mathbf{b}; \mathbf{t}), \quad (1)$$

where  $\bar{\mathbf{x}}$  is the mean shape vector,  $\mathbf{Q}$  is a shape basis matrix, and  $\mathbf{b}$  is a vector of tangent shape parameters, and  $T(\cdot; \mathbf{t})$  is a global similarity transformation with parameters  $\mathbf{t}$ . The shape parameters are denoted as  $\mathbf{p} = [\mathbf{t}, \mathbf{b}]$ .

To find the shape parameters  $\mathbf{p}$  for a given image  $I$ , an iterative searching algorithm is usually used in ASM [6] and its variants: find an observed shape  $\mathbf{y}$  by locally finding the best nearby match for each point; optimize the shape parameters  $\mathbf{p}$  based on new matched points. In the second step, we optimize the following energy function:

$$E(\mathbf{p}|\mathbf{y}) = \Delta_y^T \Sigma_y^{-1} \Delta_y + \lambda \cdot \mathbf{b}^T \Sigma_b^{-1} \mathbf{b}, \quad (2)$$

where  $\Delta_y = \mathbf{S}(\mathbf{p}) - \mathbf{y}$  is the difference between estimated shape  $\mathbf{S}(\mathbf{p})$  and the observed shape  $\mathbf{y}$ ,  $\Sigma_y^{-1}$  is the image noise covariance matrix,  $\Sigma_b$  is the covariance matrix of the shape, and  $\lambda$  is a regularization factor. The shape parameters  $\mathbf{p}$  can be optimized using EM algorithm [28].

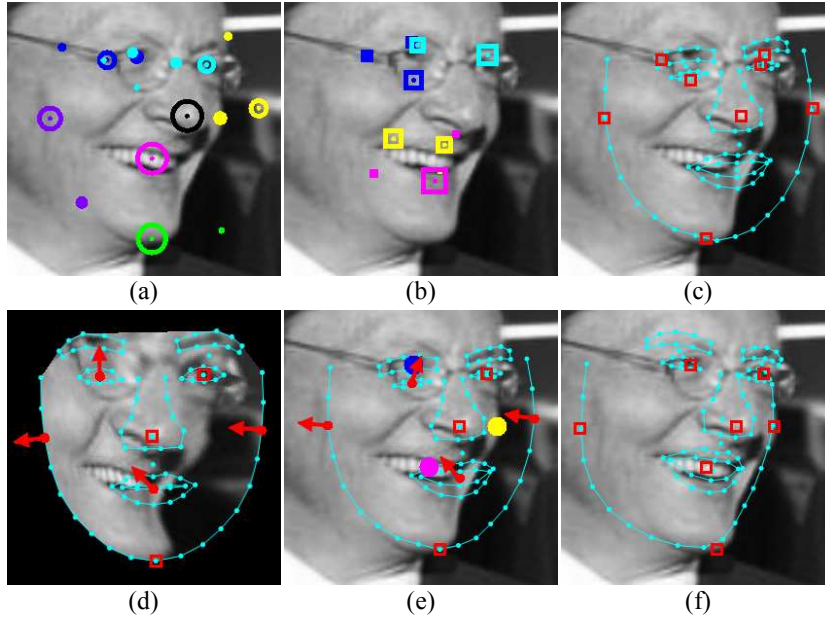
As we mentioned before, the above model is sensitive to the initialization and may not always to converge the correction solution since the search is only performed in a small range in each iteration. A natural solution is to provide prior estimates of the position to constrain some of the shape points. In [5], Cootes and Taylor show that extra user-specified constraints can greatly improve the AAM model. In this work, we also follow this general idea: we detect facial components and use multiple detected modes as constrains.

**Component-constrained optimization.** Suppose we have detected positions  $\widehat{\mathbf{S}}_c = \{\widehat{s}_i^c\}_{i=1}^C$  of facial components and corresponding variances  $\{\sigma_i\}_{i=1}^C$ , we can enforce the component constraints to the energy (2):

$$E'(\mathbf{p}|\mathbf{y}) = E(\mathbf{p}|\mathbf{y}) + \lambda_c \Delta_c^T \Sigma_c^{-1} \Delta_c, \quad (3)$$

where  $\Delta_c = \mathbf{S}_c(\mathbf{p}) - \widehat{\mathbf{S}}_c$  is the difference of estimated component positions  $\mathbf{S}_c(\mathbf{p})$  with the constrained positions  $\widehat{\mathbf{S}}_c$ , covariance matrix  $\Sigma_c = \text{diag}(\sigma_1^2, \dots, \sigma_C^2)$  models the uncertainty of detected positions, and  $\lambda_c$  is a weighting parameter.

However, the component detectors are usually error-prone. Because the local patch alone is not discriminative enough, multiple modes might be detected or no mode can be detected for one component. Even worse, the detected modes are rather noisy and contain localization errors. Thus, the challenge is how to effectively and efficiently use these multi-modes, noisy, and incomplete observations.



**Fig. 1.** An example. (a-b) detected modes of facial components. We use the color and shape to distinguish the modes from different components. The size of each mode represents its weight. (c) fitted initial shape  $S_0$  using the best mode (marked as small red square) with maximal weight of each component. (d) warped appearance to the mean shape coordinate. In this coordinate, we apply direction classifier to determine the searching direction of each component. The red arrows indicate the decided directions and the red rectangles represent “no move”. (e) solid circles are searched new modes. (f) the fitted shape after one step searching. **We encourage the reader to see this figure in the electronic version.**

A simple solution is to sample each component’s modes based on their confidence. But, the combinations of all modes of whole components could be large, which result in an inefficient algorithm. More essentially, the sampling approach is difficult to handle the case that no correct mode is detected.

To address this problem, in the next section, we present a component-based discriminative searching algorithm which can efficiently find very good configurations of components only in a few iterations.

### 3 Component-based Discriminative Search

#### 3.1 Algorithm overview

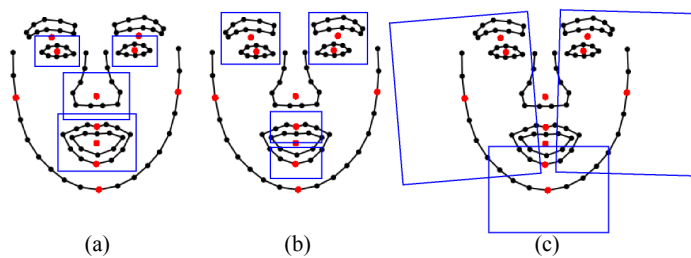
Before we describe the details, we first give an overview of our algorithm. At first, we detect multiple possible positions, *modes*, of each facial component by a component detector (section 3.2). Using the best mode (with largest wight) of each component as constraint, we search an initial shape  $S_0$  by minimizing the energy (3). Then, starting from the initial shape, we search the new mode for each component by a discriminative search algorithm (section 3.4). The searching direction is determined by boosted

- 
- Detect the face
  - Detect the components' modes in the face region
  - Constrain each component at the mode with maximal weight, search an initial shape  $\mathbf{S}_0$  using Eqn. (3).
  - for  $k = 1$  to  $K$ 
    - (1) For each component, perform discriminative search
      - Determine the searching direction by direction classifiers
      - Move the component to an appropriate position along the direction
    - (2) Search with new component constraints, get the shape  $\mathbf{S}_k$  by Eqn. (3)
    - (3) Evaluate the shape score  $f(\mathbf{S}_k)$
  - The output shape  $\mathbf{S}^* = \max_{\mathbf{S}_k} f(\mathbf{S}_k)$
- 

**Table 1.** The flowchart of our algorithm.

direction classifiers (section 3.3). Using the new searched modes as constraints, we re-optimize the face shape. We repeat this step for  $K$  iterations. Finally, we calculate the score  $f(\mathbf{S}_k)$  of all visited shapes  $\{\mathbf{S}_0, \dots, \mathbf{S}_K\}$  using the Boosted Appearance Model (BAM) [17], which is a discriminative approach for evaluating the goodness of a fitted shape. We pick the best one as the final output. The algorithm flowchart is shown in Table 1.

Figure 1 shows a simple example for the illustration purpose. We draw the component detection results on Figure 1 (a) (eyes, nose, and mouth) and Figure 1 (b) (brows, upper/lower lips). Figure 1 (c) is the fitted initial shape and Figure 1 (d) is warped image from the initial shape to the mean shape coordinates. In this image, we perform the first step of the discriminative search - applying direction classifiers to decide the searching direction of each component. Then, in Figure 1 (e), we inverse-warp the searching directions back to the original image coordinates and perform the second step of the discriminative search - moving the components to appropriate positions along the searching directions. Note that some components decide not to move. Figure 1 (f) shows the re-optimized shape after one step searching.



**Fig. 2.** Component definition and image patches for the detection. Red points are the positions of components. Blue rectangles are extracted image patch for the training of the component detection: (a) eyes, nose, and mouth. (b) brows, upper/lower lips. (c) three profiles.

### 3.2 Component detection

For each component, we train a component detector using Harr wavelet features by Adaboost [25]. The patch definitions of components are shown in Figure 2. For three profiles, we experimentally find that the large patches are better than the small patches. We collect 4000 positive examples for each component. The negative examples are generated by “bootstrap” in each stage of the cascade training. In the detection phase, we scan all patches at multiple scales with the face region. For the efficiency consideration, we only run the detectors on a regular grid with 3 pixels interval.

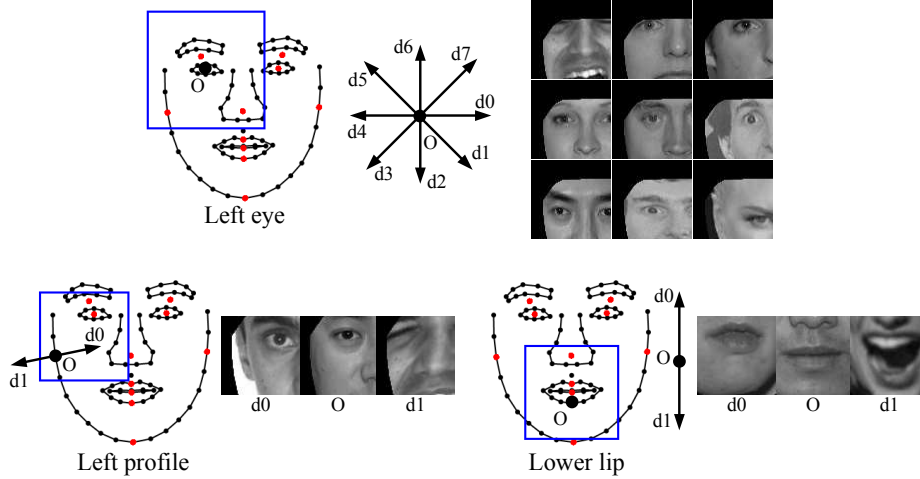
The confidence output of the detector is defined as:

$$c = 1/(1 + \phi_h \exp(-r)) \quad (4)$$

where  $r$  is the output score of the last strong classifier (at  $h$ -th layer) it passed, and  $\phi_h$  is the ratio of negative examples to positive examples for the  $h$ -th layer (obtained during training process) [14]. The final confidence value  $c(x, y)$  at the position  $(x, y)$  is the sum of the confidences by all detectors at multiple scales. Thus, we obtain a set of points with confidence values.

To reduce the number of candidate point, we fit a Mixture of Gaussian (MoG) on these points. We first find local maximums to initial the MoG fitting algorithm. Note that we use the confidence value as the weight of each point. We also merge very closed clusters after fitting. Finally, we use MoG’s components as our detected *modes*:  $\{w(\mathbf{m}_i), \mathbf{m}_i\}$ , where  $w(\mathbf{m}_i)$  and  $\mathbf{m}_i$  are weight and center of one cluster in MoG.

In the component-constraint optimization Equation (3), a covariance matrix  $\Sigma_c = \text{diag}(\sigma_i^2)$  models the uncertainty of detected mode. In our implementation, we define the variance as  $\sigma_i = \frac{1}{w(\mathbf{m}_i) + \epsilon}$ , where  $\epsilon$  is set to 0.01;



**Fig. 3.** Direction classifiers. Top: nine kinds of patches for left eye.  $O$  is “no move” and  $d_0, d_1, \dots, d_8$  are for eight directions. Bottom-left: three kinds of patches for left profile. Bottom-right: three kinds of patches for lower lip.

### 3.3 Learning direction classifiers

The goal of direction classifier is: given a patch in the facial image, we want to know whether the patch is well-aligned with a specific facial component or not; if not, what the relative direction between the patch and the facial component is. To achieve this goal, we take a discriminative approach and learn from the data.

We generate training examples from 4,000 well-labeled facial images in four ways: 1) 4,000 well-aligned “true” shapes ; 2) 60,000 randomly perturbed shapes from the true shapes by translation, rotation and scaling; 3) 30,000 randomly perturbed shapes then locally optimized using equation (2); 4) 30,000 randomly perturbed shapes then locally optimized using randomly sampled the modes as constraints. We warp all shapes and images to the mean shape coordinates. The mean image is normalized to  $263 \times 263$ .

For each component, we extract total 120,000,  $40 \times 40$  image patches. The size of the local patch is designed to include other components that may be helpful as the context. We divide those patches into three classes based on the distance to the “true” position: well-aligned ( $< 5$  pixels distance), mis-aligned ( $> 20$  pixels distance), and the others. We discard the last class since they are ambiguous. We further divide the mis-aligned patches into eight sets based on eight discrete directions with respect to the “true” position. Finally, we have nine sets of patches with known relative direction w.r.t the true position, as shown in the top row of Figure 3.

Using these training patches, we train nine classifiers for different directions (the direction of the well-aligned set is “no move”). For an individual direction, we use the corresponding patches as positive samples and all the other patches as negative samples. The classifier is trained by Adaboost on Gabor feature (8 directions and 5 scales filter responses). The Gabor feature is a  $40 \times 40 \times 8 \times 5$  vector. We use the stump decision as the weak learner on each dimension. The number of weak learner is around 200-300 on average.

For left and right profiles, it is reasonable to only consider three directions: left, right, and no move. For brows and upper/lower lips, we treat them as the child components of eyes and mouth respectively. Therefore, we only consider up, down, and no move since we will move them after the movements of eyes and mouth in the discriminative search. The example patches for these components are shown in the bottom row of Figure 3.

Equipped with the direction classifiers, we will introduce a discriminative search algorithm to effectively search new positions of facial components in the next subsection.

### 3.4 Discriminative search

Given the current shape, we first warp the image to the mean shape coordinates. For each component, we calculate the raw scores  $[s(O), s(d_0), \dots, s(d_8)]$  of nine direction classifiers and select the optimal direction with maximal score. Then the optimal direction is transformed back to the image coordinate. We denote the optimal direction and position of the component under the image coordinate as  $\mathbf{d}^* = [d_x^*, d_y^*]$  and  $\mathbf{s}$ , the associated score as  $s(\mathbf{d}^*)$ . We also define a set of modes  $\mathcal{M}$  which are within the searching range of  $\mathbf{d}^*$ , viz

$$\mathcal{M} = \{\mathbf{m}_k | \langle \mathbf{m}_k - \mathbf{s}, \mathbf{d}^* \rangle \geq 0.7\}. \quad (5)$$

---

Input: component's position  $\mathbf{s}$  and weight  $w$   
Output: component's new position  $\mathbf{s}'$  and weight  $w'$

- Find the optimal direction  $\mathbf{d}^*$  with maximal classifier score  $s(\mathbf{d}^*)$
- if the direction is "no move"
  - $\mathbf{s}' = \mathbf{s}$
  - if  $s(\mathbf{d}^*) > 0$ 
    - $w' = w,$
  - else
    - $w' = 0$
- else (other directions)
  - construct the mode set  $\mathcal{M} = \{\mathbf{m}_k | \langle \mathbf{m}_k - \mathbf{s}, \mathbf{d}^* \rangle \geq 0.7\}$
  - if  $\mathcal{M} \neq \emptyset$ 
    - select best mode  $\mathbf{m}_k^* = \arg \max_{\mathbf{m}_k} \{\langle \mathbf{m}_k - \mathbf{s}, \mathbf{d}^* \rangle \cdot w(\mathbf{m}_k)\}$
    - $\mathbf{s}' = \mathbf{m}_k^*$
    - $w' = w(\mathbf{m}_k^*)$
  - else
    - $\mathbf{s}' = \mathbf{s} + 0.1 \cdot \text{FaceWidth} \cdot \mathbf{d}^*$
    - $w' = 1$

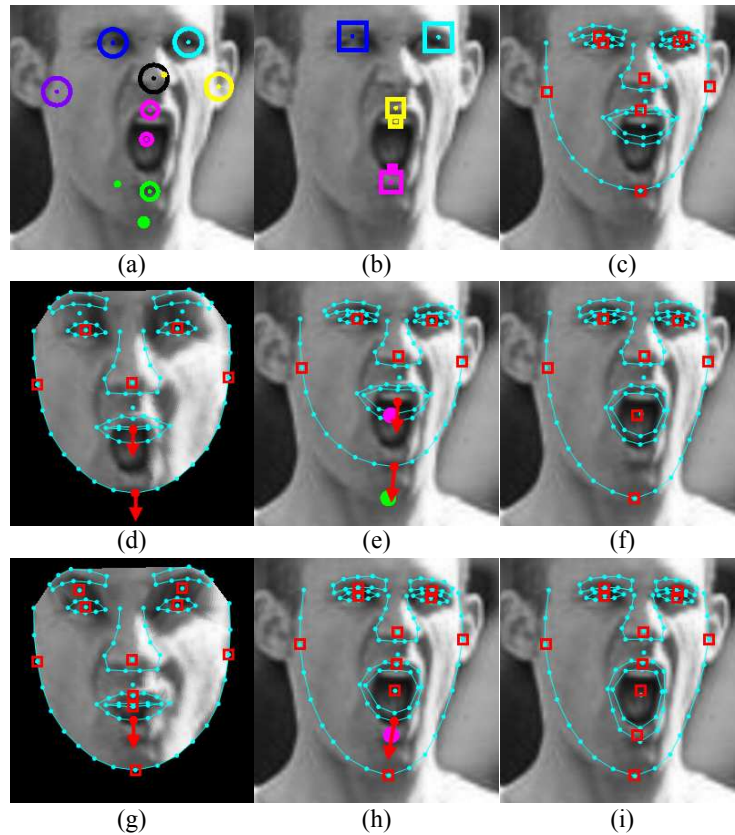
---

**Table 2.** The pseudo code of discriminative search for an individual component.

Based on these information, we search each component using strategies as follow:

1.  $s(\mathbf{d}^*) > 0$  and  $\mathcal{M} \neq \emptyset$   
The score of the direction is high and  $\mathcal{M}$  is not empty. We move the component to a mode in  $\mathcal{M}$ . We select the mode  $\mathbf{m}_k^*$  in  $\mathcal{M}$  by considering both direction and mode's weight  $w(\mathbf{m}_k)$ :
$$\mathbf{m}_k^* = \arg \max_{\mathbf{m}_k} \{\langle \mathbf{m}_k - \mathbf{s}, \mathbf{d}^* \rangle \cdot w(\mathbf{m}_k)\}. \quad (6)$$
2.  $s(\mathbf{d}^*) > 0$  and  $\mathcal{M} = \emptyset$   
The score of the direction is high but  $\mathcal{M}$  is empty. In this case, we move the component a constant step along the direction  $\mathbf{d}^*$ . The step size is set as  $0.1 \times \text{face width}$ . The new position constrains the shape searching. This strategy is designed to handle the situation that no correct mode is detected.
3.  $s(\mathbf{d}^*) \leq 0$   
The score of the direction is low, which means that the direction classifier is not sure about the proposed searching direction. In this case, we do not move the component and do not use component's position as the constraint.

Notice that for the direction "no move", we only apply the last strategy. The pseudo code of discriminative search for a single component is shown in Table 2. Due to the hierarchical nature of the facial components, we use a two level searching scheme. We first search eyes, nose, mouth and profiles at the top level. For brows and upper/lower lips, we only search them when their parent components (brows are children of eyes, and lips are children of mouth) are judged as "no move".



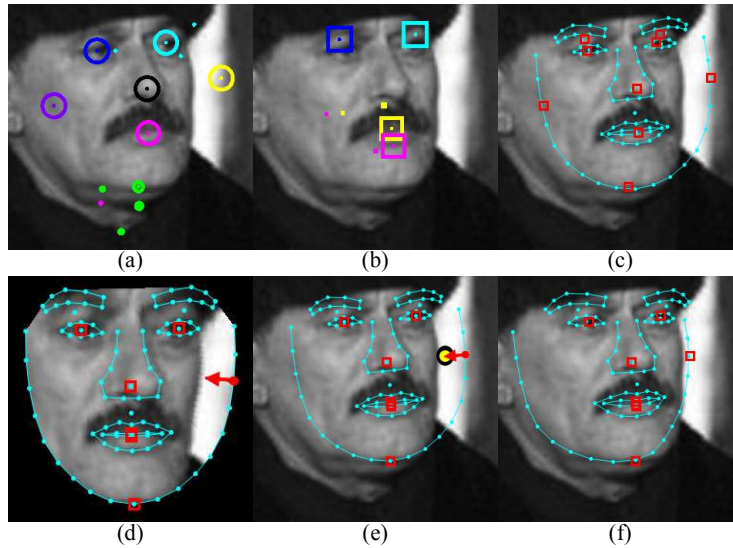
**Fig. 4.** Hierarchical search. (a-c) detected modes and initial shape. (d-f) top level discriminative search (eyes, nose, mouth, and profiles). (g-i) bottom level discriminative searching (brows and lips). Upper and lower lips are more accurately located.

Figure 4 shows an example. In the first iteration, mouth and lower profile are moved toward "down" (Figure 4 (d-e)). The fitted shape is shown in Figure 4 (f). In this step, we do not use upper and lower lips as constraints. In the next iteration, lower lip is moved toward "down" (Figure 4 (d-e)). The final fitted shape is obtained using upper and lower lips as constraints (Figure 4 (i)).

The discriminative search also has the capability to handle the case that no correct mode is detected for partial facial components. Figure 4 is an example.

## 4 Experiments

**Datasets.** To evaluate our algorithm, we collect 6,471 images from multiple databases, including AR database [18], FERET database [21], PIE database [23], and Labeled Faces in the Wild (LFW) [13]. We randomly select 4,002 images for training and 2,469 images for testing. To analysis the performance on different data sets, we separate the



**Fig. 5.** Handling incorrect mode. (a-b) are detected modes of components. No correct mode of right profile is detected. (c) initial shape. (d) warped image in the mean shape coordinates. The direction classifier decides to move the right profile to “left” by a constant step. (e) direction in the image coordinates. (f) fitted shape after one step search.

Component	Accuracy	Component	Accuracy
left eye	98.4%	right eye	98.6%
left brow	97.8%	right brow	97.4%
nose	98.9%	mouth	97.1%
upper lip	97.6%	lower lip	97.0%
left profile	98.2%	right profile	98.0%

**Table 3.** The accuracy of direction classifiers.

testing image into two sets A and B. The set A contains 753 images that are from AR, FERET, and PIE. These images are collected under controlled conditions. The set B contains 1,716 images that are all from LFW which are collected from the web and show much larger variations in pose, expression, lighting, focus, and background. See [13] for the detailed description of this database.

**Performance of direction classifiers.** First, we evaluate the performance of our direction classifiers. We generate 14,897 aligned and 48,663 misaligned shapes on all 2,469 testing images and extract the image patches of all facial components. The directions classifiers are applied on these patches to estimate the searching directions. Table 3 shows the accuracy (percentage of correctly predicted directions w.r.t the “true” directions) of the estimated searching detection of each component. Even using the simple maximal score rule, the average accuracy rate is very high - above 97%. It demonstrates the ability of the learned direction classifiers on the prediction of directions.

**Comparison.** We compare our approach with Bayesian Tangent Shape Model (BTSM) [28], constrained BTSM (CBTSM) (using the most confident mode of each component as constraint), and sampling-based approach (SMP). The SMP samples the mode of

	RMSE < 5 pixels	RMSE < 7.5 pixels	RMSE < 10 pixels
Our approach	94.0%	98.9%	99.3%
SMP (50 samples)	91.3%	97.8%	99.1%
SMP (20 samples)	91.2%	97.6%	99.1%
SMP (5 samples)	89.6%	96.3%	98.9%
CBTSM	88.7%	95.8%	98.5%
BTSM	89.0%	95.6%	98.1%

**Table 4.** Comparison on the set A. Each column is the percentage of images with RMSE less than a given threshold.

	RMSE < 5 pixels	RMSE < 7.5 pixels	RMSE < 10 pixels
Our approach	74.7%	93.5%	97.8%
SMP (50 samples)	63.3%	86.8%	95.1%
SMP (20 samples)	63.1%	86.6%	94.9%
SMP (5 samples)	58.2%	83.0%	92.7%
CBTSM	57.2%	82.8%	92.1%
BTSM	51.4%	72.5%	85.2%

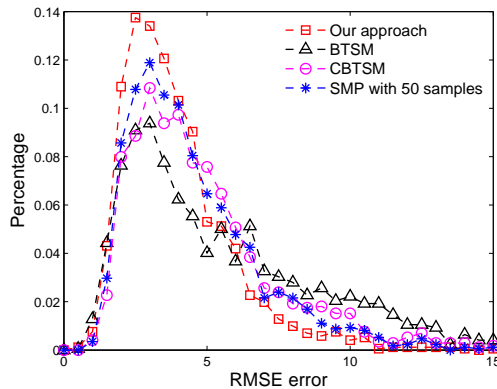
**Table 5.** Comparison on the set B.

each component based on its weight and uses the sampled modes as the constraints. We select the best fitted sample at the output using the same shape score function [17]. For all the algorithms except for SMP, the initial shape is set as a mean shape transformed by the optimal similarity transformation using the most confident modes. All algorithms uses four level pyramid for the coarse-to-fine search. In our approach, we performs 3 iterations in the discriminative search.

**Set A.** Table 4 shows the percentage of images with RMSE (Root Mean Squared Error w.r.t the manually labeled “true” shape) less than given thresholds. All images are resized to 200-300 pixels. As we can see, with the help of component detectors, all algorithms perform very well on this data set and our result is slightly better. It indicates the commonly used AR, FERET and PIE databases created in controlled conditions are largely saturated.

**Set B.** Table 5 are the percentages of images with RMSE less than given thresholds and Figure 6 shows corresponding RMSE histograms. The RMSE histograms give more accurate distribution of images in given RMSE intervals. As we can see, our result is significantly better on this more challenging data set. Due to the large variation of this data set, the performance of component detectors degrade. Consequently, for many cases, the initial shape might be far from the correct shape. The performances of BTSM and CBTSM drop a lot because the most confident modes may not be correct. Using the sampling strategy, SMP is better. But the the improvement is marginal when we increase the sample number from 20 to 50. A possible reason is that the correct position is not within the detected modes. In our approach, the discriminative search algorithm can move a component along the estimated searching direction even if the component detector has failed. Thus, our algorithm can still find a better shape. As shown in Table 5, for the threshold of 7.5 pixels, our result is 93.5% and the best next one is 86.8%.

**Iteration number.** We also test the effect of iteration number in our algorithm on the set B. Table 6 shows the percentage of correctly aligned samples (within the threshold



**Fig. 6.** The RMSE histograms of our approach, BTSM, constrained BTSM and the sampling method on the set B.

Iteration 0	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
82.8%	92.9%	93.4%	93.5%	93.5%	93.5%

**Table 6.** The effect of iteration number in our algorithm. The second row is the percentage of correctly aligned samples at each iteration.

of 7.5 pixels) vis the number of iterations. It indicates that our algorithm can correctly align most of the images within 3 iterations. Note that the computation cost of the discriminative search is neglectable since the direction classifier is extremely fast ( $<1\text{ms}$ ). Therefore, our whole approach is about 3 times slower than BTSM and CBTSM but 7 times faster than SMP with 20 samples.

Figure 7 shows example alignment results by our approach on the images with non-frontal view, large expression, poor lighting, and complex background.

## 5 Conclusion

In this paper, we have presented a discriminative approach for face alignment. By learning from the data, a new discriminative search algorithm can very effectively find a good configuration of face shape at the component level. The algorithm is very efficient and able to handle the initialization problem.

The performance of our approach is greatly benefit from the discriminative learning: detecting components' modes, searching components, and evaluating the shape. Therefore, it can be improved using a larger training data with high variability to make the face alignment more robust in the uncontrolled, real applications. To obtain more accurate results, we are going to train the direction classifiers at the finer level and apply them after the optimization at the component level.



**Fig. 7.** Aligned examples by our approach. The images in the top row are from the set A. The other images are from the set B which is a more challenging dataset.

## References

1. S. Avidan. Support vector tracking. *IEEE Trans. on PAMI*, 26(8):1064–1072, 2004.
2. A. Blake and M. Isard. *Active Contours*. Springer-Verlag, London, 1998.
3. Hong Chen, Ying-Qing Xu, Heung-Yeung Shum, Song-Chun Zhu, and Nan-Ning Zheng. Example-based facial sketch generation with non-parametric sampling. volume 2, pages 433–438, 2001.
4. T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998.
5. T. F. Cootes and C. J. Taylor. Constrained active appearance models. In *ICCV*, volume I, pages 748–754, 2001.

6. T. F. Cootes, C. J. Taylor, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.
7. T. F. Cootes and C.J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–574, 1999.
8. T. F. Cootes, K. Walker, and C. J. Taylor. View-based active appearance models. *Automatic Face and Gesture Recognition*, pages 227–232, 2000.
9. D. Cristinacce and T. F. Cootes. Boosted regression active shape models. volume 2, pages 880–889, 2007.
10. Jingyu Cui, Fang Wen, Rong Xiao, and Xiaoou Tang. Easyalbum: An interactive photo annotation system based on face clustering and re-ranking. In *SIGCHI*, 2007.
11. Lie Gu and T. Kanade. 3d alignment of face in a single image. volume 1, pages 1305–1312, 2006.
12. Xinwen Hou, S.Z. Li, Hongjiang Zhang, and Qiansheng Cheng. Direct appearance models. volume I, pages 828–833, 2001.
13. G. B. Huang, M. Ramesh, T. Berg, and E. L. Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Technical Report*, 25(12):07–49, October 2007.
14. Yuan Li, Haizhou Ai, T. Yamashita, Shihong Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. *CVPR*, pages 1–8, 2007.
15. Lin Liang, Fang Wen, YingQing Xu, Xiaoou Tang, and Heung-Yeung Shum. Accurate face alignment using shape constrained markov network. In *CVPR*, 2006.
16. C. Liu, H-Y. Shum, and C. Zhang. Hierarchical shape modeling for automatic face localization. In *ECCV*, pages 687–703, 2002.
17. Xiaoming Liu. Generic face alignment using boosted appearance model. In *CVPR*, 2007.
18. A.M. Martinez and R. Benavente. The ar face database. *CVC Technical Report*, (24), 1998.
19. I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.
20. A. Blake O. Williams and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Trans. on PAMI*, 27(8):1292C1304, 2005.
21. P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. on PAMI*, 22(10):1090–1104, 2000.
22. S. Romdhani, S. Cong, and A. Psarrou. A multi-view non-linear active shape model using kernel pca. In *BMVC*, 1999.
23. T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. on PAMI*, 25(12):1615–1618, 2003.
24. Jilin Tu, Zhenqiu Zhang, Zhihong Zeng, and Thomas Huang. Face localization via hierarchical condensation with fisher boosting feature selection. In *CVPR*, volume 2, pages 719–724, 2004.
25. Paul Viola and Michael J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.
26. L. Wiskott, J.M.Fellous, N.Krüger, and C.von der Malsburg. Face recognition by elastic bunch graph matching. In *ICIP*, 1997.
27. A. L. Yuille, D.S. Cohen, and P. W. Hallinan. Feature extraction from faces using deformable templates. In *CVPR*, pages 104–109.
28. Y. Zhou, L. Gu, and H-J. Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In *CVPR*, volume 1, pages 109–116, 2003.