Pattern Recognition 48 (2015) 1277–1288

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Hierarchical facial landmark localization via cascaded random binary patterns

Zhanpeng Zhang [a,b], Wei Zhang [c,*], Huijun Ding [d], Jianzhuang Liu [a,c], Xiaoou Tang [a,b]

[a] Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
[b] Shenzhen Key Laboratory for Computer Vision and Pattern Recognition, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, P.R. China
[c] Media Technology Lab, Huawei Technologies Co. Ltd., P.R. China
[d] Institute of Biomedical Engineering, Medical College, Shenzhen University, P.R. China

ARTICLE INFO

Article history:
Received 13 September 2013
Received in revised form
22 July 2014
Accepted 8 September 2014
Available online 18 September 2014

Keywords:
Facial landmark localization
Random binary pattern
Hierarchical regression
Gradient boosting decision tree

ABSTRACT

The main challenge of facial landmark localization in real-world application is that the large changes of head pose and facial expressions cause substantial image appearance variations. To avoid high dimensional facial shape regression, we propose a hierarchical pose regression approach, estimating the head rotation, face components, and facial landmarks hierarchically. The regression process works in a unified cascaded fern framework with binary patterns. We present generalized gradient boosted ferns (GBFs) for the regression framework, which give better performance than ferns. The framework also achieves real time performance. We verify our method on the latest benchmark datasets and show that it achieves the state-of-the-art performance.

1. Introduction

Automatic facial landmark detection/localization is a long-standing problem in computer vision. It plays a key role in face recognition systems and many other face analysis applications. In [1], it has been shown that the performance of face recognition can be remarkably elevated when facial landmark locations can be utilized. In the application of facial attribute analysis [2,3], precise facial landmark locations need to be found for feature extraction. In [4], the facial landmarks are used as the input to drive the animation of a 3D avatar. For the above reasons, the problem of facial landmark localization has been extensively studied during the past decades, and great improvements have been achieved on the standard benchmarks, such as BioID [5], LFPW [6], AFLW [7] and 300-W [8]. However, the large variations of face appearance caused by illumination, expression, and out-of-plane rotation make the robust and accurate localization in real-world applications still a challenging task.

Recently, explicit regression based methods have achieved the state-of-the-art performance for accurate and robust face alignment. The basic framework of these methods is to treat the landmark localization as a regression task: Let S be a parametric face shape. For a given input image I with an initial shape estimation S^0, S is progressively refined by cascaded regressors ϕ at stage t:

$$S^t = S^{t-1} \circ \phi^t(f^t(I, S^{t-1})), \tag{1}$$

where f represents a feature extraction function, such as SIFT [9], HOG [10], and binary feature [11–14].

Compared with the generative model based methods, such as ASM [15] and AAM [16], this framework has the following advantages: (a) since it incorporates facial appearance in a reasonable coarse-to-fine manner, the regression strategy avoids large computation caused by local window search or model fitting; (b) global facial context is incorporated into the regression at the beginning; during the cascaded regression stages, the facial context is refined from coarse to fine so that it is constrained to a local region for precise landmark localization; (c) it is capable of handling a large amount of training data, which improves the generalization power when used in real world scenarios.

However, since the above approaches utilize global regressors for shape regression, they might suffer from the high dimensional regression problem when a large number of landmark points are required: Firstly, the high dimensional regression training cost might be unaffordable if we need to learn the features from large training data; Secondly, it can easily cause overfitting and hurt generalization ability during testing. In addition, it might not be the optimal strategy to use a global regression during the whole landmark localization process, because the face shape is refined in local regions during the latter stages of the regression. For example, it does not make sense that the local features in the

* Corresponding author. Tel.: +86 755 86392199; fax: +86 755 86392073.
E-mail addresses: zz013@ie.cuhk.edu.hk (Z. Zhang), zhangwei@siat.ac.cn (W. Zhang), hjding@szu.edu.cn (H. Ding), liu.jianzhuang@huawei.com (J. Liu), xtang@ie.cuhk.edu.hk (X. Tang).

http://dx.doi.org/10.1016/j.patcog.2014.09.007
0031-3203/

components of the eyes will influence the position of the mouth. In [11,6], a non-parametric shape prior is utilized to handle the high dimensional regression and it achieves the state-of-the-art performance.

In this paper, we propose a new regression framework to locate facial landmarks for real world applications. To handle the high dimensional face shape regression problem, we estimate facial landmarks in a hierarchical way, where the high dimensional shape is decoupled into a set of low dimensional parameters, which includes head rotation, facial component location and the whole facial landmark position. In the remaining parts of the paper, the head rotation and the locations of facial components and landmarks together are referred to as facial pose. Fig. 1 shows the overview of the framework. There are three levels in the hierarchical pose regression: head rotation, face components, and facial landmarks. In each level, we estimate the pose using generalized Gradient Boosted Ferns (GBFs). The motivation for our hierarchical structure is that the image appearance variations can be reduced in each level gradually. Besides, reducing the regression dimension also makes the learning process easier. Specifically, with the head rotation estimated in the top level, we obtain the conditional probability over the whole view space. Then we estimate the rest pose parameters with the view-based GBFs in level 2 and level 3. Also, in level 2, we estimate the locations of a few facial components, further constraining the regression space for level 3. The recent work [17] is especially related to our approach in its hierarchical strategy for shape regression. The high dimensional face shape input is decoupled into a set of facial components and the pose estimation is also performed in the final refinement stage. The deep convolutional neural network (CNN) [18] is used for the cascaded regression. Different from [17,18], our approach does not need the heavy computation used by CNN. Also, it works in a unified framework and does not need to crop the facial component patches in the cascaded stages for regression, which also saves substantial computation.

In the experimental section, we will show that using simple binary features with tree-based regression approaches can efficiently handle the high dimensional shape input. The proposed method is evaluated on the latest challenging datasets of [19,1,8] and achieves the state-of-the-art performance.

2. Related work

Early work on facial landmark detection is often treated as a component of face detection. Burl et al. [20] develop a bottom up approach for face detection where it needs to first detect candidate facial landmarks over the whole image. Gabor filters [21] have been applied to large-scale facial parts such as eyes, nose, and mouth. Without the global shape constraints, false alarm is the main challenge for these component based detection approaches, even for well-trained detectors.

To better handle larger pose variation, constraints can be built on the relative locations between facial components. It can be expressed as predicted locations of one facial component given another location [21]. In [7], the DPM [22] style detector is used for multi-view facial landmark detection and pose estimation simultaneously. Alternatively, the constraints can also be built on the joint distribution of all facial components. When such constraints are modeled as a multivariate normal distribution, it results in the well-known Active Shape Model (ASM) [15,23] and Active Appearance Model (AAM) [16,24,25]. ASM is extended in [26,27] by using a Gaussian Mixture Model for shape distribution whereas [28] utilizes a mixture of Gaussian trees to describe the relation between landmark positions and the face bounding box. Non-parametric shape constraint derived directly from training samples is used in [6].

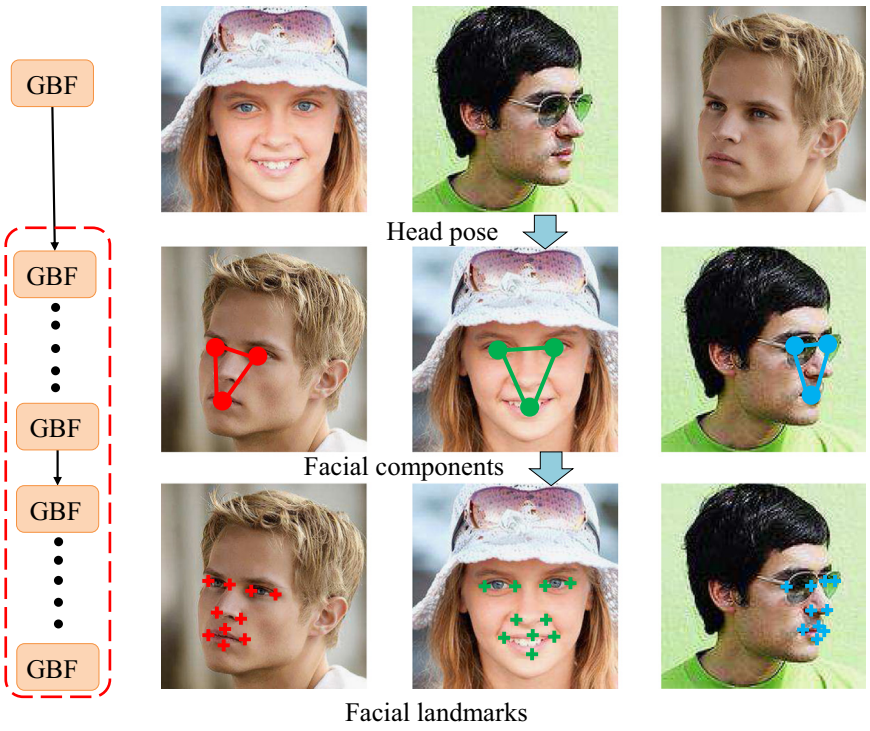

Fig. 1. Overview of our hierarchical pose regression approach, which is based on a unified framework with sequential groups of generalized gradient boosted ferns (GBFs). The conditional view-based GBFs are enclosed by the red rectangle on the left. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Explicit shape regression has emerged as the leading approach for accurate face alignment in the past several years. As mentioned in the introduction section, it can incorporate the holistic facial context in a coarse-to-fine manner and avoid expensive local searching. These approaches can be divided into two categories based on the used features, handcrafted or learned. The process in [9] uses the SIFT feature whereas the HOG feature is used in [10]. Both of them utilize simple linear regression. For the learning based methods, [18] formulates the regression into the framework of a convolution neural network (CNN) and uses image patches as the input directly.

3. Boosted regression with comparison-based features

Comparison-based features have been applied to many computer vision problems. These features are ideal for real time applications as they can be computed very fast. Besides, the algorithms have great discriminative power by aggregating these comparison-based features. Generally, traditional random ferns work with pixel-based features, which describe the pixel value difference and can work well when the regression space is relatively small. However, for images with substantial appearance variations, pixel-based features are too weak and lower the convergence rate of an algorithm. Instead, we extend the use of ferns and employ patch-based features to fit our approach. These generalized ferns work with patch and pixel comparison features in different levels in our hierarchical regression.

Given input data $\{x_i \in \mathbb{R}^F\}_1^N$ in an F-dimensional feature space and an S-dimensional regression target, a fern takes the input feature vector $q_i \in \mathbb{R}^M$ ($M < F, q_i$ is a subset of x_i) and outputs prediction $y_i \in \mathbb{R}^S$. It contains a threshold for each dimension of x_i. The M-dimensional input features and thresholds are selected randomly in the training process. In testing, each dimension of x_i is compared with the corresponding threshold to create a binary signature of length M. Consequently, every input vector can be assigned to one of the 2^M bins. The output of a bin is the mean of the predictions y of the training samples that fall into the bin. Random ferns can also be treated as a lower-parametric version of random forests [29]. It is reported in [30] that by aggregating random ferns, we can obtain comparable discriminative power as random forests in the classification problem.

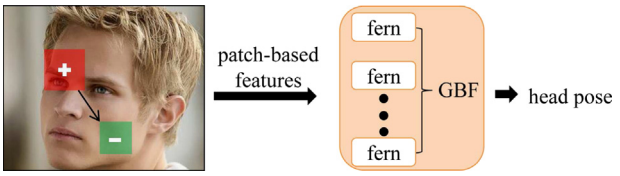

Fig. 2. Patch-based features used in the GBF regression to estimate the head pose.

We introduce random fern to the gradient boosting framework [31]. The boosting method fits our problem since it provides an efficient way to select the features for the random ferns. Specifically, in the training process of GBF, our goal is to find a function $F(x)$ that maps an input feature vector x to a target value y, while minimizing the expected value of the loss function $\Psi(y, F(x))$. $F(x)$ is in a form of a sum of weak regression functions, $F(x) = \sum_{t=1}^{T} \alpha f(q^t; \theta^t)$, where α is a learning rate ($\alpha = 0.05$ in our experiments), and $f(q^t; \theta^t)$ is the regression function of a fern, with q^t and θ^t being the corresponding feature and threshold respectively. For simplicity, the outputs of the bins in a fern are not identified explicitly in the equation, as they are determined directly by the training samples together with q and θ.

A greedy stage-wise approach is employed in the learning process. At each stage t, we find a weak regressor $f(q^t; \theta^t)$ that maximally decreases the loss function:

$$\{q^t, \theta^t\} = \arg\min_{q,\theta} \sum_{i=1}^{N} \Psi(y_i, F_{t-1}(x_i) + f(q_i; \theta)). \tag{2}$$

A steepest descent step is then applied for the minimization problem of (1). However, it is infeasible to apply gradient descent on q and θ as a fern represents a piecewise-constant function. Instead, at each stage t, we compute the "pseudo-residuals" by

$$\tilde{y}_i = -\left[\frac{\partial \Psi(y_i - F(x_i))}{\partial F(x_i)}\right]_{F(x) = F_{t-1}(x)}. \tag{3}$$

In our implementation, we use the least-squares for the loss function $\Psi(y, F(x))$ and then $\tilde{y}_i = y_i - F_{t-1}(x_i)$. The problem is thus transferred to

$$\{q^t, \theta^t\} = \arg\min_{q,\theta} \sum_{i=1}^{N} \|\tilde{y}_i - f(q_i; \theta)\|^2. \tag{4}$$

Given q and θ, a fern's output can naturally solve the minimization problem of (3), as a fern's output is the mean of $\tilde{y}_i$ of the samples that fall into the bin. That means we should just choose the suitable q and θ in training. The pseudocode of our gradient boosted fern regression is described in Algorithm 1.

Algorithm 1. Gradient boosted fern regression.

1: Given the training samples $\{x_i \in \mathbb{R}^F\}_1^N$ with target values $\{y_i \in \mathbb{R}^S\}_1^N$.
2: $F_0(x) = \text{mean}\{y_i\}_1^N$.
3: **for** $t = 1$ to T **do**
4: Randomly select a set of M-dimensional features $\{q^r\}_{r=1}^R$ from the F-dimensional input features, and a set of corresponding thresholds $\{\theta^r\}_{r=1}^R$.
5: $\{q^t, \theta^t\} = \arg\min_{q^r, \theta^r} \sum_{i=1}^{N} \|\tilde{y}_i - f(q_i^r; \theta^r)\|^2$, where $\tilde{y}_i = y_i - F_{t-1}(x_i)$.
6: $F_t(x) = F_{t-1}(x) + \alpha f(q^t; \theta^t)$
7: **end for**

Fig. 3. (a) Facial component level in the hierarchical regression. The red points are the positions. Green circles roughly indicate our sampling radius for the features. (b) Cascaded GBF regression. (b) Red pixel pairs indexed by the homogeneous coordinates (white crosses) of current estimated components. (c) A hierarchical configuration for the facial components and landmarks. A landmark (the cross) is described by a displacement vector (the arrow) from it to its parent component. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

4. Hierarchical pose regression

The GBFs described in Section 3 are the basic components of our regression framework. In the hierarchical pose regression process, several GBFs are connected sequentially. In this section, we describe how these GBFs work together for the regression of the head pose, and the localization of the facial components and landmarks. In general, the head pose estimated in the first level is used to drive the view-based model in the following levels. In the facial component level, the algorithm estimates the locations of the salient facial parts, which will serve as the initialization for the facial landmark level, where all landmarks are included.

4.1. Head pose level

For head pose regression, we estimate the 3D head rotation with a GBF. Each training sample contains a face roughly localized by a face detector and annotated with head rotation values $\omega = \{\text{yaw}, \text{pitch}, \text{roll}\}$. We use the gray-scale version of the image and apply a global illumination normalization as a preprocessing step to reduce the effect of varying illumination conditions. In the learning process, we randomly generate a pool of simple patch-based features for GBF regression:

$$v(\gamma, I) = \frac{1}{|Q_1|_{p \in Q_1}} \sum I(p) - \frac{1}{|Q_2|_{p \in Q_2}} \sum I(p), \tag{5}$$

where $\gamma = \{Q_1, Q_2\}$ with Q_1 and Q_2 being the squares within the image I. This feature can be efficiently computed using integral images. It can be treated as a generalized form of Haar-like features, allowing higher degree of freedom. After the feature selection in the GBF training process, we store γ, the threshold and the predictions of the bins for every fern. Because there are just some comparison and look-up operations for a fern, in testing, we compute all the selected features for the image and then it can go through every stage in the GBF extremely fast. The GBF regression for the head pose is illustrated in Fig. 2.

With the estimated head rotation ω', we can compute the conditional probabilities over the 3D view space and estimate the 2D facial pose with conditional view-based GBFs. Here we discretize the space of ω into disjoint sets $\{\Phi_i\}$. The Gaussian kernel is employed to estimate the distance between ω' and Φ_i: $d(\omega', \Phi_i) = 1/2\pi\sigma^2 \exp(-\|\omega' - \omega_i\|^2/2\sigma^2)$, where ω_i is the centroid of Φ_i, and σ is the bandwidth parameter. To estimate the 2D facial pose, we have

$$u = \sum_{\Phi_i} u(i) P(\omega'|\Phi_i) = \sum_{\Phi_i} u(i) \frac{d(\omega', \Phi_i)}{\sum_{\Omega_i} d(\omega', \Phi_i)}, \tag{6}$$

where $u(i)$ is the 2D pose estimated by the GBFs in the Φ_i view space and it can be obtained as described in Sections 4.2 and 4.3.

Table 1
Mean and standard deviation of the errors for the 3D head rotation estimation.

Method	GBF	SVR
Pitch error	$8.60^\circ \pm 8.56^\circ$	$14.96^\circ \pm 11.34^\circ$
Yaw error	$6.77^\circ \pm 6.69^\circ$	$10.05^\circ \pm 7.99^\circ$
Roll error	$4.75^\circ \pm 5.68^\circ$	$6.89^\circ \pm 6.87^\circ$

Fig. 4. Example images in the dataset for the head pose regression experiment.

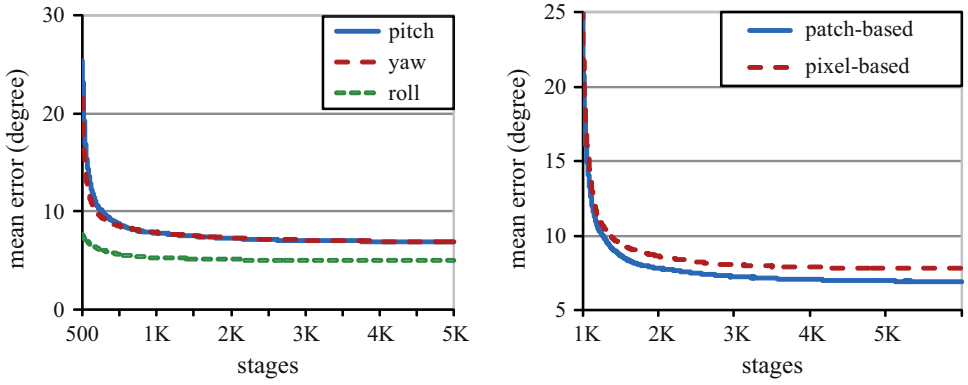

Fig. 5. Left: Mean head rotation errors in different stages in the GBF regression. Right: Mean pitch angle errors of patch-based features and pixel-based features.

4.2. Facial component level

The 2D facial pose is estimated in the view-based model. However, as the pose space of facial landmarks is large, the regression is still difficult or needs a good initialization. We sperate the pose into the component level and landmark level (i.e., $u=\{s_c, s_l\}$). Then we solve this problem with our hierarchical approach. The regression process firstly works on a component level, estimating the locations of some salient facial parts (e.g., eyes, nose, mouth), as illustrated in Fig. 3(a).

Cascaded GBFs $(G^1, G^2, \ldots, G^K)$ are included in this regression level. Given the input image I and initial pose s_c^0, each GBF estimates the pose increment Δs_c and update the pose, as shown in Fig. 3(b). Specially, for each GBF, the features are related to the image I and the pose updated by the previous GBF (called pose-indexed features [12]). So we have

$$s_c^k = s_c^{k-1} + G^k(I, s_c^{k-1}), \quad k=1,2,\ldots K. \tag{7}$$

The underlying assumption of the pose-indexed features is that, given an object, the feature value only depends on the difference between the input pose and the ground truth pose. These features are ideal for computing the 2D pose of objects in images. For a pose-indexed feature, we simply use the intensity difference of two pixels in the image. Such features are extremely easy to compute and have shown impressive performance in many other computer vision problems [32,33]. Specially, the pixel is indexed by pose, not the image coordinates. We define an associated homography matrix for each facial component and express the pixel in the homogeneous coordinates, as illustrated in Fig. 3(c). We use a hierarchical structure to manage the components. The rotation of the homography matrix is defined by the displacement between the child and parent components.

We take a greedy approach in the training of cascaded GBFs, training each GBF sequentially and minimizing the residual in each stage k. The method is described as follows:

1. Given the training images within a same view space and their ground truth facial component poses, take the mean pose as the initial pose.
2. Randomly generate a pool of pose-indexed features.
3. Train a GBF as Algorithm 1. The input is the pool of generated features and the target is the pose residual.
4. Update current pose with the pose increment predicted by the trained GBF.
5. Repeat Steps 2, 3 and 4 K times or until the residual is unable to reduce.

Fig. 6. Example images in the LFW face database [19] for facial landmark localization. The left image shows the annotated landmarks.

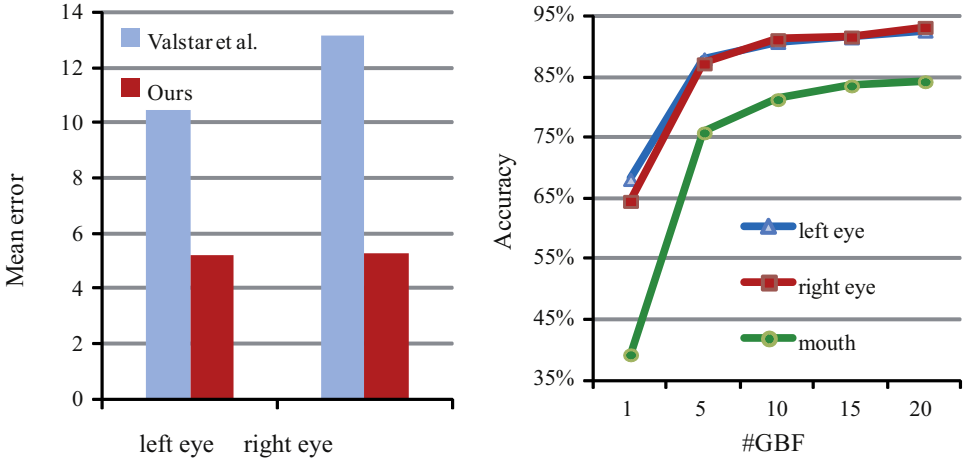

Fig. 7. Accuracy in the facial component level. Left: Mean error ($\times 10^{-2}$ of inter-ocular distance) of the eye locations between our method and Valstar et al.'s [13]. Right: Accuracy with different numbers of GBFs.

Table 2
Mean errors ($\times 10^{-1}$) of the landmark localization by three methods.

Method	Everingham et al. [28]	Dantone et al. [13]	Ours
1. Left eye left corner	16.21	6.82	5.78
2. Left eye right corner	10.70	5.65	5.32
3. Right eye left corner	9.37	5.67	5.40
4. Right eye right	11.16	7.36	5.75
5. Mouth left	10.76	7.38	7.13
6. Mouth right	15.14	7.80	7.30
7. Nose strip left	10.85	5.92	6.69
8. Nose strip right	12.08	7.05	6.71
9. Upper outer lip	–	6.40	6.69
10. Lower outer lip	–	9.53	8.52

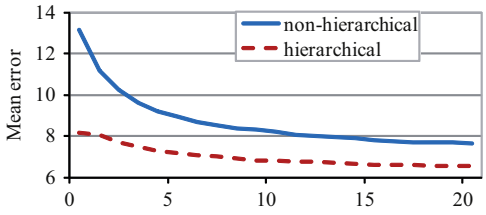

Fig. 8. Mean errors ($\times 10^{-2}$ of inter-ocular distance) of the landmark localization by the hierarchical and non-hierarchical approaches.

4.3. Facial landmark level

We use the estimated facial component locations as the initialization for the regression of the facial landmark locations (i.e., s_l). Cascaded GBFs are also used in this stage. A hierarchical configuration for the facial components and landmarks is defined, as illustrated in Fig. 3(d). We assign a parent component to each landmark based on the spatial distribution. A landmark is described by a displacement vector from it to its parent component, so we need to estimate the displacement via the cascaded GBF regression. The motivation for using the displacement vector is that the variation of the relative positions is much smaller and the shape constraint is encoded implicitly in this case. Besides, we also employ the facial components' locations as the regression target, meaning that we can update the locations of the landmarks and components jointly. This method improves the accuracy due to the high correlation between the components and landmarks.

The training process for the cascaded GBFs in this level is similar to that of the upper one. The only difference is that the pose-indexed features are sampled within a smaller area (proportional to the distance between neighboring landmarks). This is to reduce the effect of nonrigid deformation and to capture features in a more detailed level.

For the initial pose in testing, we use the facial component locations estimated by the upper level, and the mean displacement

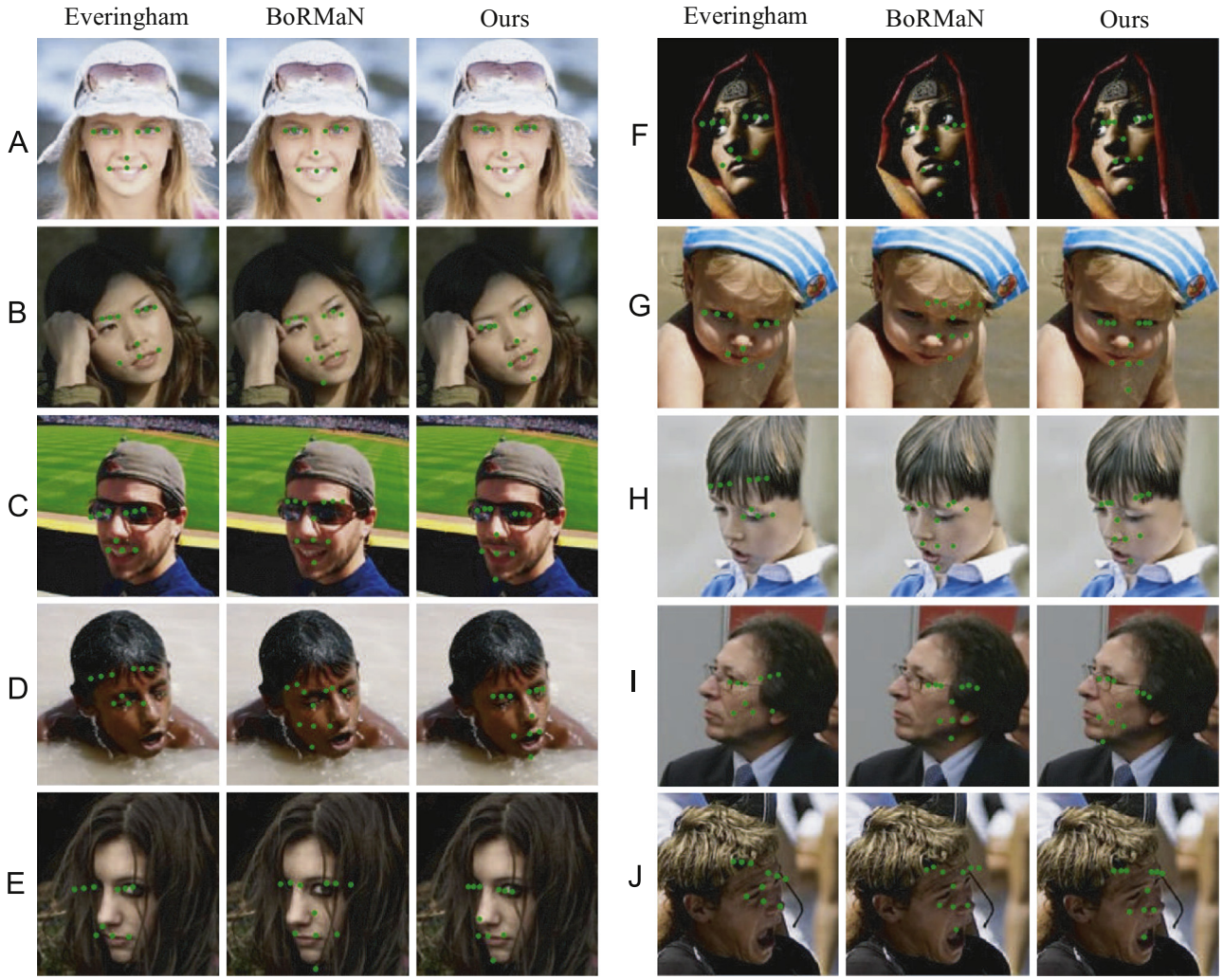

Fig. 9. Qualitative comparison among the method of Everingham et al. [28], BoRMaN facial point detector [13] and our algorithm. We randomly select faces with different degrees of error caused by our algorithm on the AFLW database [1].

vectors in the training samples. The test instances go through the cascaded GBFs and we obtain the 2D facial pose $u(i)$ in a view space Φ_i. Then the final locations for the landmarks are computed by Eq. (5).

5. Experiments and evaluations

5.1. Head pose regression

We first verify the head pose regression. Because we use only one GBF in the head pose level, we can also evaluate the performance of GBF regression directly. We use the Biwi Kinect Head Pose Database created in [34] in the experiment. This database is built with a Microsoft Kinect sensor. It contains 24 sequences of 20 different people recorded while sitting in front of the sensor. They are asked to rotate their heads to span all possible orientations. An off-line template-based head tracker is used to label the 3D rotation angles, which range between $\pm 75^\circ$ for yaw, $\pm 60^\circ$ for pitch, and $\pm 50^\circ$ for roll. In our experiment, we use the RGB images in this database. The database contains 15,000 frames. We randomly select 3800 frames to perform our experiment. Before the training and testing, we crop the face in an image based on the labeled head center. We apply some random displacement and scale transformation on the face. Then the images are rescaled to 150×150 pixels. Fig. 4 shows some sample images in the training and testing process.

Average accuracy (%) on AFLW

Method / landmark	Luxand	Zhu's	Ours
#1	94.45	90.44	96.02
#2	93.93	92.54	95.03
#3	90.82	74.02	90.01
#4	92.40	86.18	93.66
#5	93.02	92.54	96.62
#6	94.50	93.11	96.44
#7	91.71	89.58	94.00
#8	91.07	80.32	63.90
#9	89.87	79.01	93.29
#10	89.16	83.70	91.06
#11	75.55	67.30	81.80

Mean Error ($\times 10^{-2}$) on AFLW

Method / landmark	Luxand	Zhu's	Ours
#1	8.60	10.48	7.49
#2	8.70	10.64	7.92
#3	10.58	16.13	10.61
#4	10.03	13.24	8.84
#5	9.67	10.69	7.52
#6	8.58	11.08	7.53
#7	10.74	12.14	9.35
#8	11.38	15.55	18.57
#9	10.78	15.01	9.27
#10	11.51	13.63	10.33
#11	17.44	18.87	14.60

Variations of the error ($\times 10^{-2}$) on AFLW

Method / landmark	Luxand	Zhu's	Ours
#1	12.65	14.33	10.23
#2	12.80	14.50	11.62
#3	14.42	13.25	9.39
#4	15.81	15.04	12.37
#5	10.26	13.73	8.64
#6	10.37	13.54	9.69
#7	15.68	15.85	13.63
#8	11.07	16.25	13.00
#9	16.41	14.05	9.63
#10	16.52	12.51	10.88
#11	21.03	17.51	17.10

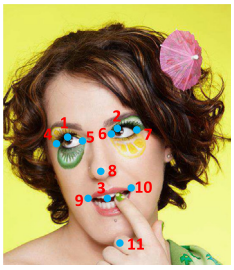

Fig. 10. Quantitative comparison among the Luxand commercial face SDK [41], Zhu and Ramanan's method [7] and the proposed algorithm on the AFLW database [1]. The accuracy is defined by an error threshold of 0.2 $\times$ inter-ocular distance. The right bottom face image is labeled with the index of the facial landmarks.

The main parameters of a GBF are the number T of stages, the dimension F of features generated for training, the fern depth M, and the R feature subsets from which we select the best in each stage. Here we set $T = 5000, F = 10,000, M = 5$, and $R = 20$ for our experiment.

Convergence analysis: Firstly, we analyze the effect of the number of stage T. We randomly select 2000 images for training and the other 1800 images for testing. From Fig. 5, we can see that the GBF regression converges gradually and does not overfit, showing that we do not need to carefully tune the learning rate and stage parameter for the algorithm. In contrast, the need to tune these two parameters is usually a problem for boosting algorithms. Besides, we can see that the convergence rate is fast. It is also evaluated with the patch-based and pixel-based features. As discussed in Section 3, the pixel-based features are weaker and depress the convergence rate, which is also verified by Fig. 5.

Estimation accuracy: To evaluate the accuracy of the GBF regression, we perform a 4-fold cross validation experiment on the dataset. We also compare the GBF regression with the support vector regression (SVR), which is a popular regression technique and has been applied to head pose estimation [35,36]. In the experiment, SVR is also fed with the same patch-based features as GBF. The parameters for SVR is set by the adaptive approach proposed in [37]. The results are shown in Table 1, indicating that the GBF regression outperforms SVR. We can see that the GBF regression fits for very high dimensional data. Besides, the results also demonstrate that by aggregating the ferns, we can obtain substantial discriminative power.

Running time performance: We measure the running time performance of the GBF regression on an Intel Pentium 3.2 GHz

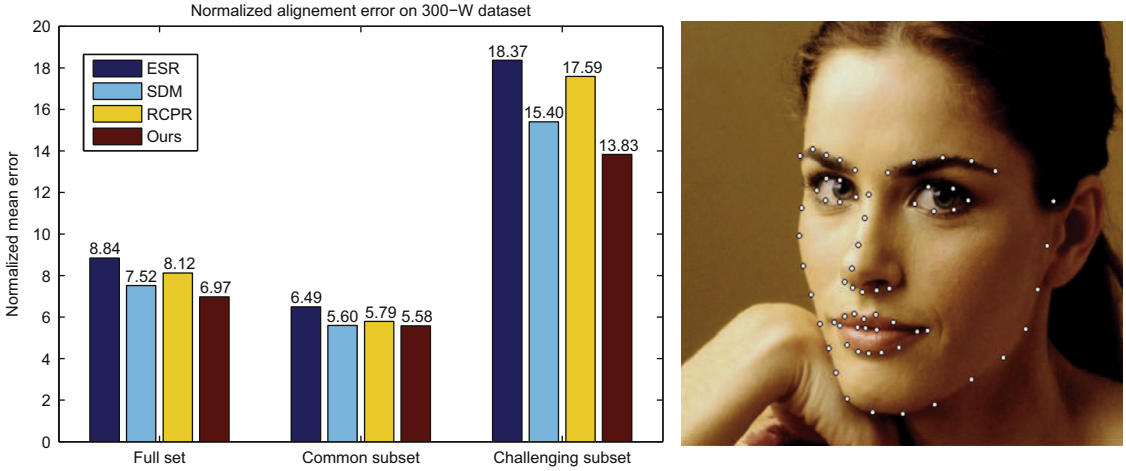

Fig. 11. Left: Quantitative comparison among ESR [11], SDM [9], RCPR [43] and the proposed algorithm on the 300-W database [8]. To further analyze the performance, we divide the testing set into two subsets as [8]. The common subset includes the testing sets of LFPW and Helen, while the challenging subset contains the IBUG faces. Right: The 68 annotated landmarks on 300-W.

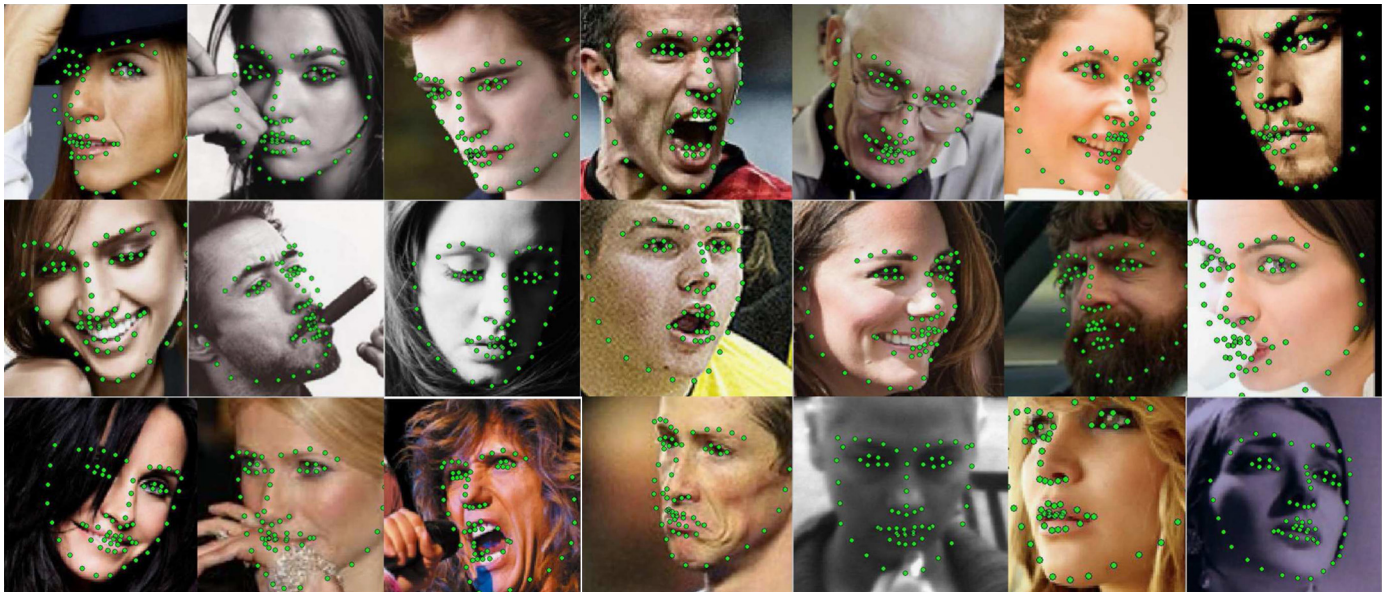

Fig. 12. Typical landmark localization results of our algorithm on the 300-W database [8].

CPU with C++ implementation. It takes only 0.55 ms for an image in the test dataset. This extremely fast performance attributes to the ferns and comparison-based features.

5.2. Facial landmark localization on the LFW database [19]

There are several existing databases used for the evaluation of landmark localization [5,38,39]. However, these databases are either limited to frontal views or acquired under controlled conditions. So they cannot exhibit enough variations of face appearances and imaging conditions, which are crucial for practical applications. In recent years, much more databases with real world images have been created [6,13,1]. These databases contain outdoor faces with large variations in pose, lighting, expression and make-up.

Firstly, we use the dataset published most recently in [19] to verify the proposed algorithm. It contains 13,233 faces taken from the LFW database [19]. The faces are annotated with the locations of 10 facial landmarks manually. Fig. 6 shows the annotated landmarks on one face and some sample images in this database. As our algorithm also needs the locations of eyes and mouth for the facial component level. The eyes' locations are set as the mean of the eyes' corners, and the mouth location as the mean of the mouth's corners. We conduct the experiment based on the result of the face detection algorithm. The detected face bounding box is enlarged by 40% and the face image is rescaled to 150×150 pixels. The faces in this dataset are split into 5 subsets based on the yaw angle of the head manually. We use this information for head pose regression by labeling them with real world angles $\omega \in \{-60, 30, 0, 30, 60\}$.

We use the same parameters as those in Section 5.1 for head pose regression. As for the 2D facial pose regression, different parameters are set in the experiments because we use different features. Specifically, we use 20 GBFs in both the facial component and landmark levels. For the parameters in training each GBF, we set $T = 500, F = 256, M = 5$, and $R = 20$. Five view-based-GBF models are trained according to the classification of yaw angles.

We perform 10-fold cross validation experiment. Similar to most previous works, the localization error is normalized by inter-ocular distance to make it invariant to face size. The accuracy is defined by a strict error threshold (0.1 inter-ocular distance). Fig. 7 shows our results on the facial component level. The mean error is compared between our method and Valstar et al.'s [40]. It shows that our accuracy is more than twice higher. The convergence of the sequential GBFs is also given in Fig. 7. Table 2 presents the comparison between our method and two state-of-the-art ones [28,13], on the facial landmark level, showing that our method outperforms both methods at most of the landmarks. Also our method is much faster. The method in [28] cannot achieve realtime performance and the method in [13] is reported to consume about 100 ms for the accuracy listed in Table 2. The computation cost of our algorithm is much less. With our current implementation, it takes only about 30 ms.

To further demonstrate the effectiveness of the proposed hierarchical approach. We compare it with non-hierarchical pose regression in the same cascaded fern framework using the same dataset. The non-hierarchical algorithm skips the head pose and facial component levels and estimates the landmarks directly. The training samples consist of the faces with head pose in the five different views. The mean errors of the two approaches on the facial landmark level are shown in Fig. 8. We can see that both the mean errors converge with 20 GBFs. In the hierarchical approach, the initial error is much less and the converging results is also better.

Fig. 14 shows some results of our algorithm on the test images. We see that it can deal with variations caused by head rotations (the first row) and facial expressions (the second and third rows). Due to the encoded shape constraint, in some cases with occlusions (the fourth and fifth rows), we can also obtain good results.

5.3. Facial landmark localization on the AFLW database [1]

AFLW database contains annotated face images gathered from Flickr.[1] In the experiment on this database, we use 11 landmarks for training and testing, as shown in Fig. 14. Specifically, we select the faces labeled with all the 11 landmarks, from which we randomly choose 4000 faces to train our model. In this database, each face is labeled with a yaw angle value and we use it to train the head pose regression model. As for the 2D pose regression, we

[1] An image hosting website (www.flickr.com).

use K-means to divide the faces into 3 clusters according to the yaw angle, and then train 3 view-based models. In this section, we conduct qualitative and quantitative analysis. The training parameters are the same as those in the previous experiment in Section 5.2.

For qualitative analysis, we test the trained model on the remaining 5857 faces in AFLW. We sort the facial landmark localization results on AFLW by the drift errors from the ground truth, and randomly select 10 faces with different degrees of error which are presented in Fig. 9. From faces A to J in Fig. 9, the errors of our algorithm increase gradually. The results are also compared with those by the BoRMaN facial point detector [13] and the method of Everingham et al. [28]. Since these two algorithms do not estimate the centers of the eyes or mouth, the eye locations are set as the mean of the eyes' corners, and the mouth location as the mean of the mouth's corners. We can see that for the frontal face (face A), all the three methods performs well. For the faces with some rotations (faces B, D, E, H), ours performs better. In a few cases, our algorithm may fail if the head pose estimation has large errors (faces I, J). The assignment of the wrong view-based model cannot well capture the face appearance. Also, in cases where the pose is far from frontal in the training set (like the face G), the algorithm may cause some errors. This is because the characteristics of these samples may be omitted, due to the average in the fern's leaves. Better choice of the features and split function can reduce this effect. However, our algorithm achieves better overall performance.

For quantitative analysis, we compare our method with Luxand [41], which is a high-quality commercial face SDK, and the algorithm in [7], which also achieves the state-of-the-art performance. Here we define the localization accuracy by an error threshold of 0.2. Fig. 10 shows the accuracy, mean and standard deviation of the errors. We can see that in all of the facial landmarks except the nose tip, our mean error is the smallest. The variation of our error is also smaller than the other two, meaning that the result is more stable. The performance of our method drops on the landmark of nose tip. It is mainly because we use simple pixel-comparison feature and it cannot work well in these textureless areas.

5.4. Facial landmark localization on the 300-W database [8]

The 300-W database [8] is a collection of faces from LFPW [6], AFW [7], Helen [42] and XM2VTS [38]. It also contains faces from a new database called IBUG. In total, this 300-W database has 3837 faces. Each face is annotated with 68 landmarks (as shown in Fig. 11).

To annotate the yaw angle of the head, we take a scheme similar to [1]. In particular, we fit a 3D face model to the annotated landmarks. Then the head pose parameters are adjusted to minimize the distance between the annotations and the projected points. We also use K-means to divide the faces into three clusters according to the yaw angle, and then train three view-based models. For the parameters, we use 20 GBFs in the facial component

Fig. 13. Typical landmark localization results of our algorithm on the LFW database [19].

Fig. 14. Typical landmark localization results of our algorithm on the AFLW database [1].

level and 40 GBFs in the landmark level. Specially, as the annotation in this dataset contains the contour of a face, we extend our hierarchical framework by adding a third level regression of cascaded GBFs. In this level, the GBFs target at the landmarks on the contour. There are also 40 GBFs in this level. For the parameters in training each GBF, we set $T=300$, $F=256$, $M=5$, and $R=5$.

The training set contains 3148 faces, including AFW, the training set of LFPW, and the training set of Helen. The testing set has 689 faces from IBUG, the testing set of LFPW, and the testing set of Helen. Our main competitors are the shape regression based methods, including explicit shape regression (ESR) [11], supervised descent method (SDM) [9] and robust cascaded pose regression (RCPR) [43]. We use the publicly available code [43] for ESR and RCPR, while we implement SDM and our implementation achieves comparable accuracy to that which was reported by the original authors. To conduct a fair comparison, we follow the same evaluation protocol as in [6,11], where the inter-pupil distance is used to normalize the landmark error. Fig. 11 shows the normalized mean errors of the proposed method with the three baseline methods. Figs. 12–14 show some results of our method on three databases.

6. Conclusions

We have presented a real time hierarchical pose regression for facial landmark localization in this paper. Different from many existing algorithms, the facial pose is estimated in a hierarchical configuration with three levels: the head pose, facial component, and facial landmark. We believe that the hierarchical pose regression can also be applied to other image-based pose regression problems. We have also proposed a generalized gradient boosted fern (GBF) regression, and the hierarchical pose regression is conducted in a unified cascaded fern framework. The discriminative power and computation efficiency are demonstrated in the experiments. Tested on the latest datasets, our experiments show that our algorithm not only runs faster but also obtains better accuracy than the state-of-the-art algorithms. Besides, due to the randomized process, the GBF can avoid the overfitting problem. In the future work, we intend to further explore this regression technique and apply it to other feature point localization problems.

Conflict of interest

None declared.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 61201443 and 61201440; in part by the Science, Industry, Trade, Information Technology Commission of Shenzhen Municipality, China, under Grant JC201005270378A; and in part by the Guangdong Natural Science Foundation under Grant S2012010010295.

References

[1] M. Koestinger, P. Wohlhart, P.M. Roth, H. Bischof, Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 2144–2151.

[2] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: IEEE International Conference on Computer Vision (ICCV), 2009, pp. 365–372.

[3] T. Berg, P.N. Belhumeur, Poof: part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 955–962.

[4] C. Cao, Y. Weng, S. Lin, K. Zhou, 3D shape regression for real-time facial animation, ACM Trans. Gr. 32 (2013) 41:1–41:10.

[5] O. Jesorsky, K.J. Kirchberg, R. Frischholz, Robust face detection using the Hausdorff distance, in: International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), 2001, pp. 90–95.

[6] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 545–552.

[7] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2879–2886.

[8] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, 300 faces in-the-wild challenge: the first facial landmark localization challenge, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2013, pp. 397–403.

[9] X. Xiong, F.D. la Torre, Supervised descent method and its applications to face alignment, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 532–539.

[10] J. Yan, Z. Lei, D. Yi, S.Z. Li, Learn to combine multiple hypotheses for accurate face alignment, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2013, pp. 392–396.

[11] X. Cao, Y. Wei, F. Wen, J. Sun, Face alignment by explicit shape regression, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2887–2894.

[12] P. Dollar, P. Welinder, P. Perona, Cascaded pose regression, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1078–1085.

[13] M. Dantone, J. Gall, G. Fanelli, L. V. Gool, Real-time facial feature detection using conditional regression forests, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2578–2585.

[14] B. Efraty, C. Huang, S. Shah, I. A. Kakadiaris, Facial landmark detection in uncontrolled conditions, in: International Joint Conference on Biometrics (IJCB), 2011, pp. 1–8.

[15] T.F. Cootes, C.J. Taylor, D. Cooper, J. Graham, Active shape models: their training and application, in: Computer Vision and Image Understanding (CVIU), 1995.

[16] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, IEEE Trans. Pattern Anal. Mach. Intell. (2001) 681–685.

[17] E. Zhou, H. Fan, Z. Cao, Y. Jiang, Q. Yin, Extensive facial landmark localization with coarse-to-fine convolutional network cascade, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2013, pp. 386–391.

[18] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 3476–3483.

[19] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[20] M.C. Burl, T.K. Leung, P. Perona, Face localization via shape statistics, in: IEEE Conference on Automatic Face and Gesture Recognition Workshops (FG Workshops), 1995.

[21] D. Cristinacce, T. Cootes, I. Scott, A multi-stage approach to facial feature detection, in: Proceedings of the British Machine Vision Conference (BMVC), 2004, pp. 231–240.

[22] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1627–1645.

[23] T.F. Cootes, M.C. Ionita, C. Lindner, P. Sauer, Robust and accurate shape model fitting using random forest regression voting, in: European Conference on Computer Vision (ECCV), 2012, pp. 278–291.

[24] X. Liu, Generic face alignment using boosted appearance model, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.

[25] J. Saragih, R. Goecke, A nonlinear discriminative approach to AAM fitting, in: International Conference on Computer Vision (ICCV), 2007, pp. 1–8.

[26] J.M. Saragih, S. Lucey, J. Cohn, Face alignment through subspace constrained mean-shifts, in: International Conference on Computer Vision (ICCV), 2009, pp. 1034–1041.

[27] V. Rapp, T. Senechal, K. Bailly, L. Prevost, Multiple kernel learning SVM and statistical validation for facial landmark detection, in: IEEE Conference on Automatic Face and Gesture Recognition Workshops (FG Workshops), 2011, pp. 265–271.

[28] M. Everingham, J. Sivic, A. Zisserman, Hello! my name is… buffy—automatic naming of characters in TV video, in: Proceedings of the British Machine Vision Conference (BMVC), 2006, pp. 889–908.

[29] A. Criminisi, J. Shotton, E. Konukoglu, Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning, Foundations and Trends in Computer Graphics and Vision 7 (2-3) (2011) 81–227. http://dx.doi.org/10.1561/0600000035.

[30] M. Ozuysal, P. Fua, V. Lepetit, Fast keypoint recognition in ten lines of code, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1–8.

[31] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[32] J. Gall, V. Lempitsky, Class-specific Hough forests for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1022–1029.

[33] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1297–1304.

[34] G. Fanelli, M. Dantone, A. Fossati, J. Gall, L.V. Gool, Random forests for real time 3D face analysis, Int. J. Comput. Vis. 101 (2013) 437–458.

[35] Y. Li, S. Gong, H.L. Jamie Sherrah, Support vector machine based multi-view face detection and recognition, Image Vis. Comput. 22 (2004) 413–427.

[36] C. BenAbdelkader, Robust head pose estimation using supervised manifold learning, in: European Conference on Computer Vision (ECCV), 2010, pp. 518–531.

[37] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Netw. 17 (2004) 113–126.

[38] K. Messer, J. Matas, J. Kittler, J. Lttin, G. Maitre, XM2VTSDB: the extended M2VTS database, in: International Conference on Audio and Video-based Biometric Person Authentication, pp. 72–77.

[39] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The feret evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1090–1104.

[40] M. Valstar, B. Martinez, X. Binefa, M. Pantic, Facial point detection using boosted regression and graph models, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2729–2736.

[41] Luxand face SDK, ⟨http://www.luxand.com⟩, 2013.

[42] V. Le, J. Brandt, Z. Lin, L. Bourdev, T. S. Huang, Interactive facial feature localization, in: European Conference on Computer Vision (ECCV), 2012, pp. 679–692.

[43] X. P. Burgos-Artizzu, P. Perona, P. Dollr, Robust face landmark estimation under occlusion, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1513–1520.

Zhanpeng Zhang received the B.E. and M.E. degree in Computer Engineering from Sun Yat-sen University, P.R. China, in 2010 and 2013 respectively. Currently, he is a candidate for the Ph.D. degree for Information Engineering in the Chinese University of Hong Kong, anticipating completion in 2016. His research interests include image processing and pattern recognition.

Wei Zhang received the B.S degree in Computer Engineering from Nankai University, China, in 2002, the M.E. degree in Computer Engineering from Tsinghua University, P.R. China, in 2005, and the Ph.D. degree from The Chinese University of Hong Kong, P.R. China, in 2010. He is now a research assistant in the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. His research interests include computer vision and pattern recognition.

Huijun Ding received the B.E degree in Electronic Engineering and Information Science from The University of Science and Technology of China, in 2006, the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singagpore, in 2011. She is now a lecturer of Shenzhen University, P.R. China, in 2013. Her current research interests include speech enhancement, objective measure and image processing applied in bio-medical engineering.

Jianzhuang Liu received the Ph.D. degree in computer vision from The Chinese University of Hong Kong, Hong Kong, in 1997. From 1998 to 2000, he was a research fellow with Nanyang Technological University, Singapore. From 2000 to 2012, he was a postdoctoral fellow, then an assistant professor, and then an adjunct associate professor with The Chinese University of Hong Kong. He joined Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, as a professor, in 2011. He is currently a chief

scientist with Huawei Technologies Co. Ltd., Shenzhen, China. He has published more than 100 papers, most of which are in prestigious journals and conferences in computer science. His research interests include computer vision, image processing, machine learning, multimedia, and graphics.

Xiaoou Tang received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor in the Department of Information Engineering, the Chinese University of Hong Kong. He worked as the group manager of the Visual Computing Group at the Microsoft Research Asia from 2005 to 2008. He received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. Dr. Tang is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and International Journal of Computer Vision (IJCV). He is a Fellow of IEEE. His research interests include computer vision, pattern recognition, and video processing.