Face Alignment via an ensemble of random ferns

CONFERENCE PAPER · FEBRUARY 2016

READS
51

3 AUTHORS, INCLUDING:

Xiang Xu
University of Houston
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Ioannis A Kakadiaris
University of Houston
298 PUBLICATIONS **3,906** CITATIONS
SEE PROFILE

Face Alignment via an Ensemble of Random Ferns

Xiang Xu, Shishir K. Shah, Ioannis A. Kakadiaris
Computational Biomedicine Lab
Department of Computer Science, University of Houston, Houston, TX, USA
xxu21@uh.edu, sshah@central.uh.edu, ioannisk@uh.edu

Abstract

This paper proposes a simple but efficient shape regression method for face alignment using an ensemble of random ferns. First, a classification method is used to obtain several mean shapes for initialization. Second, an ensemble of local random ferns is learned based on the correlation between the projected regression targets and local pixel-difference matrix for each landmark. Third, the ensemble of random ferns is used to generate local binary features. Finally, the global projection matrix is learned based on concatenated binary features using ridge regression. The results demonstrate that the proposed method is efficient and accurate when compared with the state-of-the-art face alignment methods and achieve the best performance on LFPW and Helen datasets.

1. Introduction

The problem of face alignment, or face landmark detection, has been studied for a long time. Face alignment aims to localize the facial feature points (*e.g*. eye corners, nose tip, and mouth corners) on a facial image. It serves an important role in many face-related applications because it can provide the significant facial feature point locations for further face processing. For example, in a face recognition system [8, 11], alignment can provide positions to extract the local facial patches for feature comparison. Also, it benefits the performance of face tracking algorithms. Furthermore, many expression analysis systems are based on face alignment that provides the face shape.

Regression-based methods for face alignment learn a regression function that directly maps the image appearance or features to the target output or shape residuals. Cao *et al*. [5] applied the two-level regression using random ferns (ESR) to solve the face alignment method explicitly. The shape-indexed features and correlation-based feature selection were proposed in their work. Xiong *et al*. [18] investigated linear regression with strong hand-crafted features called Supervised Descent Method (SDM). SDM aimed to minimize the non-linear least squares function by learning a sequence of descent directions. At the same time, robust discriminative response map fitting (DRMF) [1] was proposed within the Constrained Local Model (CLM) framework. Deep learning has recently been applied in this field also. A four level convolutional network was designed in a coarse-to-fine manner by Zhou *et al*. [21]. In their system, each network level was trained to refine a subset of facial landmarks generated locally by previous network levels. Ren *et al*. [14] demonstrated that a set of local binary features is very discriminative for each facial landmark. Specifically, local binary features are used jointly to learn a ridge regression for the final output. The incremental face alignment method (IFA) [2] allows all the regression functions in a cascade to be updated independently in parallel. Lee *et al*. [10] proposed a face alignment method that uses cascaded Gaussian process regression trees, where the kernel measures the similarity between two inputs that derive to the same leaves. The project-out cascade regression (POCR) is proposed by Tzimiropoulos [16], which applied the regression to learn a sequence of averaged Jacobian and Hessian matrices from the data. The approach of Zhu *et al*. [22] starts with a coarse shape searching from a large shape pool and employs a probabilistic solution to constrain subsequent searches in finer and finer shapes. The main differences between the various cascaded regression-based methods can be summarized as differences in: the initialization method, features used in the algorithm, selected regressors, and the framework of regression.

Ferns were introduced by Ozuysal *et al*. [13] and were demonstrated to be efficient and accurate by Dollar *et al*. [6], Efraty *et al*. [7] and Cao *et al*. [5]. Dollar *et al*. [6] designed a sequence of random fern regressors that predicted the object parameters while Cao *et al*. [5] proposed a two-level cascade fern regression. Specifically, it used 500 cascaded ferns as primitive regressors in each iteration. However, this method is not efficient enough because each fern regressor only decreases the alignment error by a small step. In this work, which differs from the previous methods, we explore how to use probabilities to initialize the shape and

how to generate features for global alignment by concatenating the output of a set of ferns. Following the coarse-to-fine principle, we design a face component classification for initialization and cascade regression to learn the shape increments progressively. The shape-indexed features are extracted locally and adapted progressively to learn the ensemble of ferns, which are simple and computationally efficient. In learning each fern, the features and thresholds are learned based on the correlations between the randomly projected regression targets and local pixel-difference matrix. Then, the intensities are extracted from the training images according to an ensemble of random ferns and are compared with the corresponding thresholds to derive the local binary features. The local binary features obtained from the surroundings of each landmark are concatenated to form a global binary features matrix. The global linear regression matrix is learned from these global binary features by minimizing the squared loss function with L_2 regularization at the last step. Our main contributions are: (i) applied a probabilistic model to select the initial shape for face alignment; (ii) extended the face alignment approach using the ensemble of random ferns to learn local features; (iii) implemented the learning and testing method using parallel programming, so the performance of the method is not only accurate but also more efficient.

The remainder of the paper is organized as follows: Section 2 reviews briefly the cascade shape regression and Section 3 describes the details of the learning and testing of our proposed method. The implementation details and experiments are reported in Section 4 and Section 5, respectively.

2. Background

In this section, the classical cascade shape regression algorithm in CLM framework is reviewed and the symbols are defined to make the paper self-contained.

Assume that there are L facial feature points; the face shape $\mathbf{S} = [x_1, y_1; x_2, y_2; \ldots; x_L, y_L]$ is a $L \times 2$ matrix. Given a set $\{\mathbf{I}_i, \mathbf{S}_i, \boldsymbol{b}_i\}_{i=1}^N$, where N is the total of training images $\mathbf{I}$, and $\mathbf{S}$ and $\boldsymbol{b}$ are corresponding ground-truth shape and face bounding box output by a face detector, respectively, the T regressors $\{\mathbb{R}^t\}, t = 1, \ldots, T$ are organized in a cascade in a coarse-to-fine manner. Each regressor $\mathbb{R}$ is learned to minimize the differences between the shape increments $\{\Delta\mathbf{S}_i = \mathbf{S}_i - \hat{\mathbf{S}}_i\}_{i=1}^N$ and local image features defined as $\mathbb{H}(\mathbf{I}_i, \hat{\mathbf{S}}_i)$ based on the current estimated shape $\hat{\mathbf{S}}$:

$$\mathbb{R}_* = \arg\min_{\mathbb{R}} \sum_{i=1}^{N} \|\Delta\mathbf{S}_i - \mathbb{R}(\mathbf{I}_i, \hat{\mathbf{S}}_i)\|_2, \qquad (1)$$

where we denote $\mathbb{R}(\mathbb{H}(\mathbf{I}_i, \hat{\mathbf{S}}_i))$ by $\mathbb{R}(\mathbf{I}_i, \hat{\mathbf{S}}_i)$ for simplicity. Therefore, from an initial shape $\hat{\mathbf{S}}^0$, the shape is updated progressively by adding the estimated shape residual to the current shape. That is, in t step, the regressor $\mathbb{R}^t$ learns the ground-truth shape residual $\Delta\mathbf{S}$ and produces the estimated shape residual $\Delta\hat{\mathbf{S}}^t = \mathbb{R}^t(\mathbf{I}, \hat{\mathbf{S}}^{t-1})$. The shape is updated as follows:

$$\hat{\mathbf{S}}^t = \hat{\mathbf{S}}^{t-1} + \mathbb{R}^t(\mathbf{I}, \hat{\mathbf{S}}^{t-1}), \qquad (2)$$

which would be used as a base shape for the next iteration.

3. Method

In this section, we present a local component-based method for the initialization of cascade shape regression. We propose an ensemble of random ferns to learn local features and use these features for further regression.

3.1. Local component-based initialization

First, we define the facial patch as a square region around the facial components (eyebrows, eyes, nose, mouth). Since occlusions and variations in the square are common phenomena for the eyebrows and mouth components, we opted for the components that correspond to the two eyes and the nose tip. Given the set of training data, to train several local detectors for each part, the image is scaled to $[150, 150]$. This size was selected experimentally. Positive and negative facial component patch sets are constructed with the assumption that the facial component in the training dataset and testing dataset obey the same distribution. Unlike the work by Belhumeur *et al*. [3], we construct the base shape by clustering K ground truth shapes. As illustrated in Fig. 1, the positive samples are extracted according to a Gaussian distribution centered at the component center. Moreover, the negative image patches are sampled using a uniform distribution but the negative patches are kept at the

(a) (b)

Figure 1: (a) Illustration of the facial components. The circles mark the centers of the patches. During training, we extract the HOG features with 4×4 block size and 9 bins for the three facial components. (b) Illustration of the sampling of patches. The positive patches of nose tip (red circles) are sampled around the ground truth position following a Gaussian distribution while negative samples (colorful circles) follow the uniform distribution.

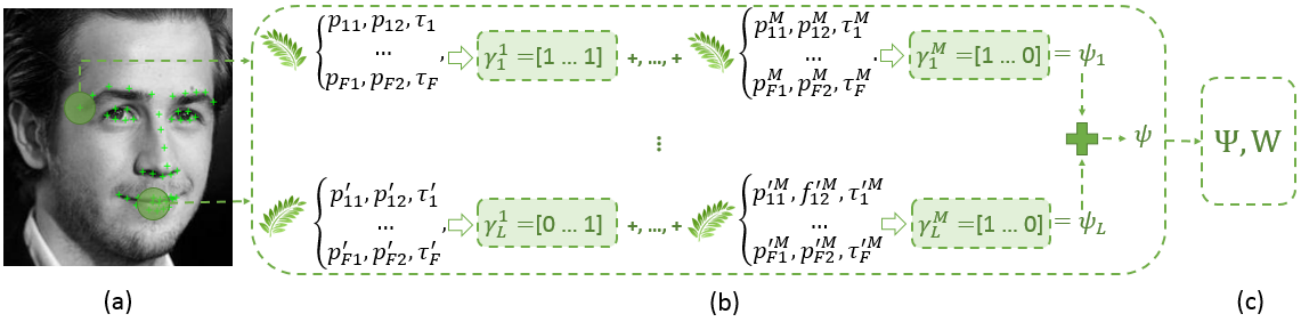


Figure 2: Overview of the ensemble of ferns: (a) randomly sample the pixels around each landmark; (b) construct an ensemble of random ferns based on a correlation-based method and use these learned random ferns to generate the local binary features; (c) generate the global features matrix and apply ridge regression to learn the global projection matrix.

distance of 20% of the inter-pupil distance from the component center. For each patch, HOG features are extracted and assigned a label to the component. The matrix of patch locations is denoted by $\mathbf{X}$ and $D = \{\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3\}$ denotes the response $\mathbf{D}_i$ of i detector. The objective in the initialization is to maximize the MAP probability of $\mathbf{X}$ given the responses.

We assume that the parts are independent of each other, therefore, the objective function is as follows:

$$\mathbf{X}_* = \arg\max_{\mathbf{X}} p(\mathbf{X}|D) = \prod_{i=1}^{3} p(\mathbf{X}_i|\mathbf{D}_i). \tag{3}$$

After obtaining trained classifiers, the classifiers are applied to each face and obtain the response maps. The locations of the components with the largest response scores are accepted, and a shape from K shapes is selected as the initial shape. Algorithm 1 describes the process of our initialization.

Algorithm 1 Initialization based on local information

Input: Training data $\{\mathbf{I}_i, \mathbf{S}_i\}_{i=1}^{N}$
Output: Initial Shape $\{\hat{\mathbf{S}}_i\}_{i=1}^{N}$
Procedure:

1: Construct the shape sets from the training data
2: Apply K-Means to the shape set to obtain K mean shapes
3: Extract the positive and negative samples from the training data
4: Train the component classifiers using SVM
5: Apply the component classifiers to each face image
6: Find the peaks in response maps and build component shape
7: Search the most similar shape in K shapes as an initialization

3.2. Ensemble of Random Ferns

The overview of our regression approach is illustrated in Fig. 2. Our Ensemble of Random Ferns (ERF) comprises $M \times L$ ferns $\{\mathbb{F}\}_{m=1,l=1}^{M,L}$. For each landmark, there are M ferns as local learners. Each fern is composed of F features and thresholds. To construct each fern, we uniformly sample P pixels around each landmark. Then, the correlation-based selection method is adopted to choose F pairs of pixels out of P^2 pixel-difference features with the aim of reducing the correlation between features but remaining discriminative. Finally, ridge regression is applied to learn the projection matrix based on the concatenated binary features learned from the ensemble of ferns.

Correlation-based selection: In the training phase, the training set is divided into M subsets randomly with replacement. Then, the intensities on the P pixels are extracted for each image in the same subset.

To form a good fern (*e.g.* a fern in which features are highly correlated to the shape increment while there is a low correlation between any feature pairs), F features are selected based on the correlation value. First, we divide all of the shape regression targets into L landmark regression target $\{\Delta\mathbf{S}\}_{i=1,l=1}^{N,L}$. Then, a random direction $\boldsymbol{v}$ is dot multiplied with each regression target to produce a scalar. These scalars are concatenated to the vector $\boldsymbol{u}$. Assuming that $\boldsymbol{\rho}_m$ and $\boldsymbol{\rho}_n$ are any pair of pixels from the set of the P pixels, the correlation between the projected regression targets $\boldsymbol{u}$ and pixel-difference features $\boldsymbol{\rho}_m - \boldsymbol{\rho}_n$ is computed as:

$$\begin{aligned} \mathbb{C}(\boldsymbol{u}, \boldsymbol{\rho}_m - \boldsymbol{\rho}_n) &= \frac{\mathbb{V}(\boldsymbol{u}, \boldsymbol{\rho}_m) - \mathbb{V}(\boldsymbol{u}, \boldsymbol{\rho}_n)}{\sqrt{\sigma(\boldsymbol{u})\sigma(\boldsymbol{\rho}_m - \boldsymbol{\rho}_n)}} \\ \sigma(\boldsymbol{\rho}_m - \boldsymbol{\rho}_n) &= \mathbb{V}(\boldsymbol{\rho}_m, \boldsymbol{\rho}_m) + \mathbb{V}(\boldsymbol{\rho}_n, \boldsymbol{\rho}_n) \\ &\quad -2\mathbb{V}(\boldsymbol{\rho}_m, \boldsymbol{\rho}_n) \end{aligned} \tag{4}$$

where $\mathbb{C}$ denotes the correlation and $\mathbb{V}$ denotes the covariance.

Two pixels from a sample with the highest correlation are selected. Then, a random threshold is generated according to the selected pixel-difference. We repeat this procedure

Algorithm 2 ERF Training procedure

Input: Training data $\{\mathbf{I}_i, \mathbf{S}_i\}_{i=1}^N$
Output: Ensemble of random ferns $\{\mathbb{F}^t\}_{t=1}^T$; projection matrix $\{\mathbf{W}^t\}_{t=1}^T$
Procedure:

Initialize $\{\hat{\mathbf{S}}_i^0\}_{i=1}^N$
Randomly divide the training data to M subsets
for $t = 1$ to T **do**
    Compute regression residuals $\{\Delta\mathbf{S}_i\}_{i=1}^N$
    for $m = 1$ to M **do**
        randomly sample P pixels around each landmark
        select F pairs of pixels and thresholds based on correlation-based method
    end for
    Extract local binary features $\{\boldsymbol{\gamma}_l^m\}_{l=1,m=1}^{L,M}$
    Concatenate the features to $\boldsymbol{\Psi}^t$
    Learn $\mathbf{W}^t$ based on (6)
    Predict shape increment by $\Delta\hat{\mathbf{S}}^t = \boldsymbol{\Psi}^t * \mathbf{W}^t$
    Update current shapes by $\hat{\mathbf{S}}^t = \hat{\mathbf{S}}^{t-1} + \Delta\hat{\mathbf{S}}^t$
end for

F times and construct a fern learner for a landmark. Based on a coarse-to-fine strategy, the features (pixels) should be sampled within a large region in the first several iterations and within a small region in the later iterations. Therefore, in sampling the data, the feature selection region is changed dynamically. Similarly, the local landmarks are used to index these sampled pixels.

Learning local features: To learn the local function, the fern learner uses the pixel-difference features. Assuming that two pixel locations $\{\boldsymbol{\rho}_j^1, \boldsymbol{\rho}_j^2\}_{j=1}^F$ have been selected ($\boldsymbol{\rho}_j^1, \boldsymbol{\rho}_j^2$ represent the j^{th} pair of pixel locations $\boldsymbol{\rho}^1$, $\boldsymbol{\rho}^2$), the value of each binary feature $\{f_j\}_{j=1}^F$ depends on the intensities of these two pixels:

$$f_j = \begin{cases} 1 & \text{if } \mathbf{I}(\boldsymbol{\rho}_j^1) - \mathbf{I}(\boldsymbol{\rho}_j^2) \geqslant \tau_j \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where $\mathbf{I}(\boldsymbol{\rho}_j)$ represents the image intensity on $\boldsymbol{\rho}_j$, and τ_j is the corresponding threshold. Therefore, each fern can generate F binary features, defined as $\boldsymbol{\gamma} = [f_1, f_2, \ldots, f_F]$.

After the ensemble of random ferns is learned, the entire training data go through the ensemble of ferns: for each image, the intensity on the relative local coordinates is saved in the fern and compared the difference with the corresponding threshold, thereby generating the local binary features by Eq. (5). Then, the binary features from each fern are concatenated to form the global binary shape features $\boldsymbol{\psi}^t = [\boldsymbol{\gamma}_1^1, \boldsymbol{\gamma}_1^2, \ldots, \boldsymbol{\gamma}_1^M, \ldots, \boldsymbol{\gamma}_L^M]$. From the training set, a $N \times (M \times L \times F)$ matrix $\boldsymbol{\Psi} = [\boldsymbol{\psi}_i;], i = 1, \ldots, N$ can be obtained. For convenience, we omit the iteration symbol t on each $\boldsymbol{\gamma}$.

Algorithm 3 ERF Testing procedure

Input: Image $\mathbf{I}$, an ERF model $\{\mathbb{F}^t\}_{t=1}^T$ and $\{\mathbf{W}^t\}_{t=1}^T$
Output: Prediction $\{\hat{\mathbf{S}}_i\}_{i=1}^N$
Procedure:

Initialize $\{\hat{\mathbf{S}}_i^0\}_{i=1}^N$
for $t = 1$ to T **do**
    Extract local binary features $\{\boldsymbol{\gamma}_l^m\}_{l=1,m=1}^{L,M}$
    Concatenate the features to $\boldsymbol{\Psi}^t$
    Predict shape increment by $\Delta\hat{\mathbf{S}}^t = \boldsymbol{\Psi}^t * \mathbf{W}^t$
    Update current shapes by $\hat{\mathbf{S}}^t = \hat{\mathbf{S}}^{t-1} + \Delta\hat{\mathbf{S}}^t$
end for

3.3. Global shape regression

Minimizing Eq. (1) is a well-known least squares problem, which can be solved using linear regression. The explicit linear regression model is subject to over-fitting because the dimensionality is high. Therefore, we impose L_2 regulation to avoid over-fitting, which is expressed as:

$$\mathbf{W}_*^t = \arg\min_{\mathbf{W}^t} \sum_{i=1}^N \|\Delta\mathbf{S}_i^t - \boldsymbol{\Psi}^t(\mathbf{I}_i, \hat{\mathbf{S}}_i^{t-1})\mathbf{W}^t\|_2^2 + \lambda\|\mathbf{W}^t\|_2^2 \tag{6}$$

where λ controls the regularization strength.

In the testing, the ensemble of ferns is applied to the test image to extract the local features and to form the global binary shape feature $\boldsymbol{\Psi}^t$ in each iteration. Then, $\{\boldsymbol{\Psi}^t * \mathbf{W}^t\}$ is used to predict the shape increment. The pseudo-code of the training and testing procedures is described in Alg. 2 and Alg. 3, respectively.

4. Implementation Details

Initialization: To train the model in a reasonable time, we augment the data by mirroring the data and sampling the five initial shapes using our local component-based initialization method. In the testing phase, we propose two different initializations. The first method, "ERF-mean" simply applies the mean shape on each image according to the face bounding box. The second method "ERF-init" uses our component detector to sample the five best shapes from the shapes obtained by the k-means algorithm. For the result, we obtain the median shape as the predicted shape.

Parameter setting: We use a randomly selected validation set and apply cross-validation to select the parameters on the LFPW dataset [3]. The number of the ensemble of random ferns is set to 10, and the number of iterations is set to 10. The algorithm samples the local pixels for each landmark within $[0.5, 0.4, 0.3, 0.2, 0.15, 0.1, 0.08]$ of face size in each iteration, respectively. The feature number F saved in the fern is 5. In our validation experiment, the number

of sample pixels P does not affect the performance significantly, so we sample 30 pixels around each landmark.

Running time performance: We implemented the algorithm in MATLAB. We trained the model using the LFPW training set. Our model is implemented in parallel, so it is very efficient with $811 \times 2 \times 2 = 3,244$ training samples (2 initial shapes per image as an example). It takes about 15 *min* for training $3,244$ samples and 6.7 *s* for testing 224 images, measured on our Intel Core i5 2.6 GHz MacBook.

Measurement: The shape prediction error is measured using Mean Square Root Error (MSRE) [5]:

$$\text{MSRE} = \frac{1}{N * L} \sum_{i=1}^{N} \frac{\|\mathbf{S}_i - \hat{\mathbf{S}}_i\|_2}{d_i} \tag{7}$$

where d_i is inter-pupil distance. All results are the averages of running the algorithms five times.

5. Experiments

In this section, we present the comparison with state-of-the-art algorithms on four datasets.

As proposed by Liu *et al*. [12], a good face alignment approach should meet these conditions: (1) be automatic and robust to the unconstrained environment; (2) efficient; and (3) be have a sufficient number of facial landmarks. In our experiments, we used 51 inner-face landmarks and all 68 face landmarks.

LFPW: LFPW [3] is a dataset that contains 1,400 images, and which originally contained 1,100 training images and 300 testing images. All images are collected from the Internet, but the authors provide only URLs, some of which are out of date. We only obtained the 811 training images and 224 testing images from the website [15].

Helen: Helen [9] comprises 2,330 high-resolution face images: 2,000 images for training and 330 images for testing. All of these images are collected from Flikr with an original 194 landmarks. This dataset provides more detailed information for accurate face alignment than the other datasets.

300-W: 300-W contains LFPW, Helen, AFW, and IBUG [15], as well as some controlled datasets XM2VTS, FRGC, and MultiPIE. AFW [23] was built by collecting the images from Flickr. The total number of images in the AFW is 205. The images contain large variations in face pose, appearance, and background. In the original dataset, each face is labeled with a face bounding box, six facial points and three viewpoints (pitch, yaw, and roll). The IBUG dataset and AFW do not contain separate training and testing sets. All datasets in 300-W are re-annotated to 68 points and corrected by an expert.

All datasets are challenging because of large variations in pose, illumination and expressions. For convenience and fair comparison, we use the image datasets with annotation files and face bounding box files provided by the ibug group [15]. For experiments in LFPW and Helen, we use the training and testing datasets provided. However, as the training set for the 300-W dataset we use the combined images from training subsets for LFPW and Helen and the entire AFW. For testing, we consider the testing subsets from LFPW and Helen as the common subset, the IBUG as a challenging subset, and the union of these three datasets as the full testing subset.

5.1. Comparison with state-of-the-art methods

We adopted the MSRE described in Eq. (7) as the evaluation metric. First, we present the comparison of the initialization using local-based shapes versus mean shapes. Second, we present the comparison with the state-of-art methods.

We compare our initialization method with mean shapes initialization using the LFPW validation set. We followed by applying the ERF but with different initializations. After obtaining the locations of eyes and nose, we select the best five shapes obtained by applying the k-means algorithm. Assuming that the time cost of the latter method is 0 *ms*, the comparison is summarized in Table 1. Note that ERF-mean uses the mean shape as an initialization while ERF-init uses the initial shapes obtained from Sec. 3.1. The second and third columns list the MSRE of initialization and final result, respectively. The error is reduced by about 38%. We observe that the initialization method can greatly reduce the error, but it is time-consuming when compared to the computation cost of regression (will discuss in Sec. 5.2).

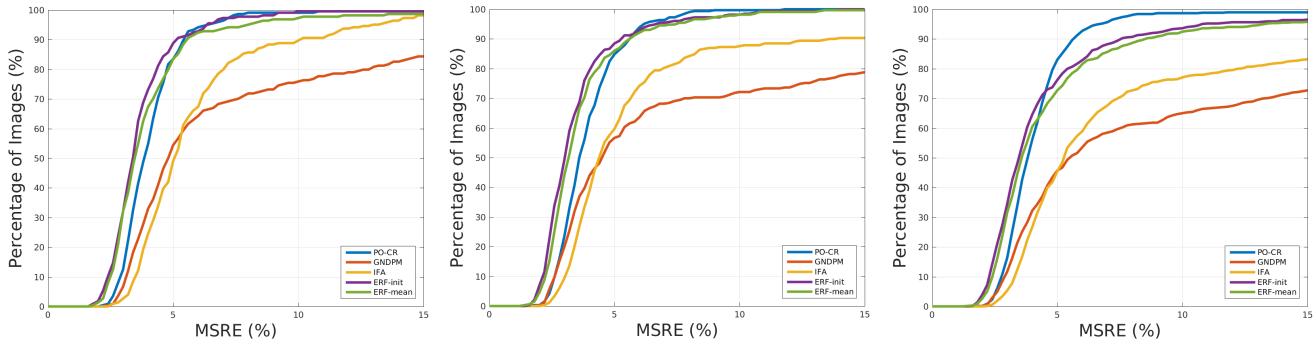
Method	MSRE (%)		Time (ms)
	Initialization	Result	
ERF-mean	21.41	5.38	11
ERF-init	13.34	5.10	210 + 78

Table 1: Comparison of the time of different initializations. We measure the average time (ms) of detecting 68 landmarks on a MacBook Pro with 2.6 GHz Intel Core i5 by conducting the validation experiment on the LFPW dataset.

For a full evaluation, we report the results of a protocol that uses 51 inner landmarks and 68 full landmarks for alignment, along with our re-implemented ESR [5] algorithm. The results are summarized in Table 2. Most of the results are obtained from the literature directly. In addition, we compare the state-of-the-art with publically available code. Specifically, we conducted experiments with the following methods: (i) Explicit Shape Regression [5] algorithm implemented by our team, (ii) Gauss-Newton deformable part models (GNDPM) [17], (iii) optimized part mixtures (OPM) model [20], (iv) incremental

Method	LFPW		Helen		300-W		
	49 pts	66 pts	49 pts	66 pts	Common	Challenge	Full set
ESR [5]	4.10	-	4.04	-	-	-	-
TSPM [23] $*$	7.78	8.29	7.43	8.16	8.22	18.33	10.20
RCPR [4] $*$	5.48	6.56	4.64	5.93	6.18	17.26	8.35
OPM [20]	9.75	12.16	-	-	-	-	-
DRMF [1] $*$	4.40	5.80	4.60	5.80	-	-	-
SDM [18] $*$	4.47	5.67	4.25	5.50	5.57	15.40	7.50
LBF [14] $*$	-	-	-	-	4.95	11.98	6.32
LBF-fast [14] $*$	-	-	-	-	5.38	15.50	7.37
GNDPM [17] $*$	4.43	5.92	4.06	5.69	5.78	-	-
IFA [2]	6.12	-	5.86	-	-	-	-
POCR [16]	4.08	-	3.90	-	-	-	-
CFSS [22] $*$	3.78	4.87	3.47	**4.63**	**4.73**	**9.98**	**5.76**
ERF-mean	4.05	4.80	3.63	5.22	5.05	17.14	7.43
ERF-init	**3.70**	**4.61**	**3.46**	4.98	4.83	15.05	6.84

Table 2: Comparison of different methods. Columns 2-8 list the MSRE (%) for each method normalized by the distance between the pupils. The results of methods with $*$ are obtained directly from the corresponding paper. Otherwise, the results are obtained by testing the publically available code with the model the author provided.



(a) (b) (c)

Figure 3: Comparison of cumulative error distribution curves of PO-CR, GNDPM, IFA and ERF with 2 different initializations when tested on LFPW (a), Helen (b), and 300-W datasets (c). For each algorithm, we use the inner-face 49 landmarks.

face alignment (IFA) [2] and (v) project-out cascade regression (POCR) [16].

As shown in Table 2, it could be observed that our method achieved the best performance on LFPW and Helen datasets compared with the other algorithms. Compared with ESR, we can observe that our method achieved competitive performance, and even dropped the error from $4.10\ \%$ to $3.70\ \%$ on the LFPW dataset with smaller training and testing time. In training, ESR needs to augment the data 20 times, which imposes a computational burden. However, our method only augments the data five times. Moreover, ESR learned 500 cascade ferns on each iteration, while ERF only learned 15 independent ferns. Therefore, the training time is greatly decreased. Moreover, our method exhibits excellent results on high-resolution images by decreasing the MSRE on Helen significantly (14.35%) compared with ESR. Our algorithm obtains similar results as CFSS [22], but CFSS needs to search similar shapes in a large shape database and uses hand-designed features. Our method selects the shape from our pre-detected key points and uses the pixel-differences as features, which are much more straightforward. In addition, to the overall evaluation of the performance of our algorithm, we compute the cumulative errors of the results obtained from ERF, GNDPM, IFA, and POCR. Figure 3 depicts the CED curves for these methods on LFPW, Helen, and the 300-W full set. Again, the proposed ERF can localize the facial points accurately. Figure 4 illustrates the results from ERF-init tested on the selected samples.

Furthermore, we compare the computation of our MATLAB implementation with algorithms the code and the pretrained model of which are provided online. We compared

Figure 4: Selected examples from LFPW (T) and Helen (B). Note that ERF is robust to large variations in facial expressions, illumination, image quality, and glasses.

the algorithms on Windows because the most of the codes can run on Windows only. Note that our computational time may be different than the time reported in the literature and Table 1 because of different computer configurations. Note that ERF-mean has the smallest computational cost to generate landmarks. Table 3 summarizes the comparison results on the Helen dataset. The computation cost of ERF is small because it is simple and uses pixel-differences as features.

5.2. Validation and Discussion

To explore the influence of the number of ferns on performance, we randomly divided the Helen training set into a training subset and a validation subset. The experiments were performed on the validation set randomly selected from the Helen using $M \in \{5, 10, 15, 20, 50\}$. The result is shown in Table 4. The number of random ferns was set to 15. The time goes up a little with an incremental increase in the number of random ferns.

Method	pts	Time (s)
DRFM [1]	66	9.00
GNDPM [17]	49	0.50
IFA [2]	49	0.20
POCR [16]	49	0.90
ERF-mean	51	0.12
ERF-init	51	1.90

Table 3: Computation cost of selected algorithms when running on Windows with Intel Core 1.86 GHz CPU. We tested the public code on the Helen test set. Note that POCR is implemented in pure C++ and IFA has some functions in C.

Based on the experiments, we observed that the component-based initialization method can improve the performance, but it is time-consuming compared with the regression procedure. However, our code is implemented in MATLAB. In addition, the classifier can be replaced by other faster classifiers. Also, recently, Yang *et al*. [19] proposed a fast deep learning algorithm which aims to predict the head pose. It is easy to extend our initialization method using these solutions to achieve the much lower computational cost. We conclude with some observations: (i) during testing, multiple proper initializations can improve the performance; (ii) multiple shapes may have outliers, which need to be eliminated; and (iii) there is room for a cascaded regression method to improve performance.

Number of ferns	5	10	15	20	50
MSRE (%)	3.79	3.51	3.50	3.50	3.62

Table 4: The impact on performance when different number of ferns is being used.

6. Conclusion

An initialization method for face alignment and a learning procedure using an ensemble of random ferns to learn local features have been proposed in this paper. The ERF is constructed in a cascade manner, where each cascaded ferns contains several simple but powerful ferns, which are used to produce the local binary features. We demonstrated the feasibility of random ferns as local learners. Our initialization method also benefits the cascaded regression. By using the proposed methods, a better performance was achieved compared with the state-of-the-art methods. The perfor-

mance of the algorithm has been demonstrated competitive in accuracy as well as speed on the LFPW, Helen, and 300-W datasets.

7. Acknowledgment

This research was funded in part by the US Army Research Lab (W911NF-13-1-0127) and the UH Hugh Roy and Lillie Cranz Cullen Endowment Fund. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of the sponsors.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, Portland, Oregon, June 23-28 2013.

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1866, Columbus, OH, June 2014.

[3] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proc.* 24^{th} *IEEE Conference on Computer Vision and Pattern Recognition*, pages 545–552, Springs, CO, Jun. 20-25 2011.

[4] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Proc. IEEE International Conference on Computer Vision*, pages 1–8, Sydney, Australia, Dec. 3–6 2013.

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2887–2894, Providence, RI, Jun 2012.

[6] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, San Francisco, CA, Jun. 13-18 2010.

[7] B. Efraty, C. Huang, S. Shah, and I. Kakadiaris. Facial landmark detection in uncontrolled conditions. In *Proc. IEEE International Joint Conference on Biometrics*, Washington, DC, Oct. 11–13 2011.

[8] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):640–649, 2007.

[9] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *Proc.* 12^{th} *European Conference on Computer Vision*, pages 679–692, Firenze, Italy, October 7-13 2012.

[10] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4204–4212, Boston, Massachusetts, June 7 - 12 2015.

[11] S. Z. Li and A. K. Jain. *Handbook of face recognition*. Springer, 2nd edition, 2011.

[12] L. Liu, J. Hu, S. Zhang, and W. Deng. Extended supervised descent method for robust face alignment. In *Proc. ACCV Workshop on Feature and Similarity Learning for Computer Vision*, pages 71–84, Singapore, Nov. 2014.

[13] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, Mar. 2010.

[14] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, Columbus, OH, June 2014.

[15] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: the first facial landmark localization challenge. In *Proc. IEEE International Conference on Computer Vision, 300 Faces in-the-Wild Challenge (300-W)*, pages 397–403, Sydney, Australia, Dec. 2013.

[16] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3659–3667, Boston, Massachusetts, June 7 - 12 2015.

[17] G. Tzimiropoulos and M. Pantic. Gauss-Newton deformable part models for face alignment in-the-wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, Columbus, OH, June 24-27 2014.

[18] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 532–539, Portland, Oregon, June 25-27 2013.

[19] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson. Face alignment assisted by head pose estimation. In *Proc. 26th British Machine Vision Conference*, Swansea, UK, September 7-10 2015.

[20] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *Proc. IEEE International Conference on Computer Vision*, pages 1944 – 1951, Sydney, Australia, December 3-6 2013.

[21] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proc. IEEE International Conference on Computer Vision Workshops*, pages 386 – 391, Sydney, Australia, December 2-8 2013.

[22] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse to fine shape searching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4998–5006, Boston, MA, June 7 - 12 2015.

[23] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886, Providence, RI, June 16–21 2012.