Facial Landmark Localization based on Hierarchical Pose Regression with Cascaded Random Ferns

Zhanpeng Zhang[1, 2] Wei Zhang[1] Jianzhuang Liu[1, 2, 3] Xiaoou Tang[1, 2]
[1]Shenzhen Key Lab of Computer Vision and Pattern Recognition,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2]Department of Information Engineering, The Chinese University of Hong Kong, China
[3]Media Lab, Huawei Technologies Co. Ltd., China
{zp.zhang, zhangwei}@siat.ac.cn, liu.jianzhuang@huawei.com, xtang@ie.cuhk.edu.hk

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 regression in the 3D and 2D facial pose spaces simultaneously, we propose a hierarchical pose regression approach, estimating the head rotation, facial components and landmarks hierarchically. The regression process works in a unified cascaded fern framework. We present generalized gradient boosted ferns (GBFs) for the regression framework, which give better performance than traditional ferns. The framework also achieves real time performance. We verify our method on the latest benchmark datasets. The results show that it outperforms state-of-the-art methods in both accuracy and speed.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications

Keywords

facial landmark localization; pose regression; random fern

1. INTRODUCTION

Facial landmark localization is a key step in many human-computer interaction systems. Many recent algorithms locate facial landmarks by optimization or regression. Optimization based approaches, such as AAM [3], usually solve this problem by fitting a parametric model to the input face. Specifically, AAM contains the statistical information of the shape and texture of the faces in the training set. The algorithm finds the model parameters which minimize the difference between the input face and synthesized model, based on gradient descent optimization. But these approaches have difficulty with generalization and may not work well on unseen instances. Meanwhile, the optimization process may fall into local optimal. Differently, regression based approaches aim to learn the mapping between the image appearance and the targets [4, 2]. Due to the high computational efficiency, random fern regression has been used in this area recently. In [5], a cascaded fern approach is proposed for 2D pose regression. Also, [2] employs cascaded ferns for face alignment. In both cases, each fern needs pose-indexed features which satisfy a weak invariant assumption [5]. These features are ideal for 2D pose estimation with 2D data in an image. However, when dealing with 3D pose, we cannot define such features. So the performance of these approaches may drop when dealing with faces with large rotations. Meanwhile, computing the pose-indexed feature in every cascaded fern adds additional cost.

To overcome these problems, we propose a unified cascaded fern regression framework to locate the facial landmarks. Usually, a fern is fed with pixel-comparison-based features. In this paper, we generalize the use of ferns by employing patch-comparison-based features, to increase the discriminative power of our regression framework. Besides, the algorithms works in a hierarchical way. Fig. 1 shows the overview of our approach. There are three levels in the hierarchical pose regression: the head rotation, face components, and facial landmarks. For simplicity, we often refer the parameters of these three levels together as facial pose. In each level, we estimate the pose using generalized Gradient Boosted Ferns (GBFs). 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. The motivation for our hierarchical structure is that joint regression in the entire pose space is too difficult. We can reduce the image appearance variations in each level gradually. Meanwhile, reducing the regression dimension also makes the learning process easier.

In the experiments, we evaluate the performance of our approach with the latest challenging datasets [4, 11], which contain outdoor faces with various pose, expression, and illumination. The experiments show that our approach generates high quality results with real time performance.

2. BOOSTED REGRESSION WITH COMPARISON-BASED FEATURES

Comparison-based features are ideal for real time applications as they can be computed very fast. Traditional random ferns work with pixel-comparison-based features, which can work well within limited target space. However, for images

MM'13, October 21–25, 2013, Barcelona, Spain.

ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502148.

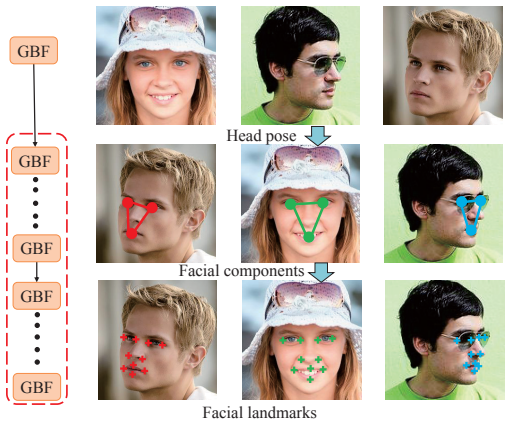

Figure 1: Overview of our hierarchical pose regression approach, which is based on a unified framework with sequential groups of generalized gradient boosted ferns (GBFs). The conditional view-based GBFs are enclosed by the red rectangle on the left.

with large appearance variations, these features are too weak and lower the convergence rate. So we extend the use of ferns and use patch-comparison-based features. These generalized ferns work with comparison-based features of different scales in different levels of our hierarchical regression.

Given input data $\{x_i \in \mathbb{R}^F\}_1^N$ in an F-dimensional feature space with the regression target $\{y_i \in \mathbb{R}^S\}_1^N$, a fern takes an 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 q_i, assigning every input vector to one of the 2^M bins. The M-dimensional input features and thresholds are selected randomly in the training process. The output of a bin is the mean of the predictions y of the training samples that fall into the bin.

We introduce random fern to the gradient boosting framework [8]. Our goal is to train a function $F(x)$ that maps a feature vector x to target y, while minimizing the expected value of the loss function $\Psi(y, F(x))$. $F(x)$ is the sum of weak regression functions, $F(x) = \sum_{t=1}^{T} \alpha f(q^t; \theta^t)$, where α is a learning rate. $f(q^t; \theta^t)$ is the regression function of a fern, with q^t and θ^t as the selected features and thresholds.

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

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

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

We use the least-squares for loss function $\Psi(y, F(x))$ and then $\tilde{y}_i = y_i - F_{t-1}(x_i)$. So we have

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

The pseudocode of the training process is as Algorithm 1.

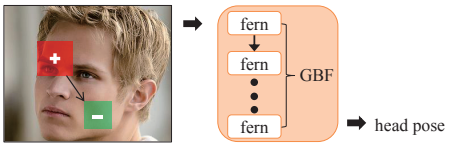

Figure 2: Patch-comparison-based features used in the GBF regression to estimate the head pose.

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**

3. HIERARCHICAL POSE REGRESSION

3.1 Head pose level

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 ω. In the training process of GBF, we randomly generate a pool of simple patch-comparison-based features:

$$v(\gamma, I) = \frac{1}{|Q_1|} \sum_{p \in Q_1} I(p) - \frac{1}{|Q_2|} \sum_{p \in Q_2} I(p), \quad (3)$$

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 with integral images. After the GBF training process, we store γ, the threshold and the predictions of the bins for every fern. In testing, as there are just some comparison and look-up operations for a fern, the head pose regression can run extremely fast. The process 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 $d(\omega', \Phi_i)$ between ω' and Φ_i, so $d(\omega', \Phi_i) = \frac{1}{2\pi\sigma^2} \exp(-\|\omega' - \omega_i\|^2 / 2\sigma^2)$, where ω_i is the centroid of Φ_i, and σ is the bandwidth parameter ($\sigma = 5$ in implementation). To estimate the 2D facial pose u, we have

$$u = \sum_{\Omega_i} u(i) P(\omega' | \Omega_i) = \sum_{\Omega_i} u(i) \frac{d(\omega', \Phi_i)}{\Sigma_{\Omega_i} d(\omega', \Phi_i)}, \quad (4)$$

where $u(i)$ is the 2D pose estimated by the GBFs in the Φ_i view space. The estimation algorithm is presented in Sections 3.2 and 3.3.

3.2 Facial component level

The 2D facial pose is estimated in view-based model. Here we further sperate the pose into the component level and landmark level (i.e., $u = \{u_c, u_l\}$), to reduce the regression difficulty. Then we solve this problem with our hierarchical

(a) (b) (c)

Figure 3: (a) 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.

approach. The regression process firstly works on a component level, estimating the locations of facial parts (e.g., eyes, nose, mouth).

We use cascaded GBFs $(G^1, G^2, ..., G^K)$ in this level. Given the input image I and initial pose u_c^0, each GBF estimates the pose increment Δu_c and update the pose, as shown in Fig. 3(a). For each GBF, the features are related to the image I and the current pose (called *pose-indexed features* [5]). So we have

$$u_c^k = u_c^{k-1} + G^k(I, u_c^{k-1}),\ k = 1, 2, ...K. \quad (5)$$

Given an object, the value of pose-indexed features should only depends on the difference between the input pose and the ground truth pose. Here, for the pose-indexed feature, we simply use the intensity difference of two pixels, which are indexed by pose, not the image coordinates. Such feature is extremely easy to compute. We can define an associated homography matrix for each facial component and express the pixel in the homogeneous coordinates, as illustrated in Fig. 3(b). 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 to train the cascaded GBFs sequentially. Firstly, we take the mean pose of the training samples as the initial pose. In every stage, we randomly generate a pool of pose-indexed features, and then train a GBF as Algorithm 1, which minimizes the current residual. Afterwards, current poses are updated by this GBF. The training process stops if it reaches the maximum stage or the residual is unable to reduce.

3.3 Facial landmark level

Cascaded GBFs are also used in this stage. A hierarchical configuration for the facial components and landmarks is defined, as illustrated in Fig. 3(c). 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 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. 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 the mean displacement vectors in the training samples, and the facial component locations estimated by the upper level as constraints. 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 Equation (4).

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

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

4. EXPERIMENTS AND EVALUATIONS

4.1 Head pose regression

We use the Biwi Kinect Head Pose Database [7] to verify the head pose regression. The head rotation range in the database is between $\pm 75°$ for yaw, $\pm 60°$ for pitch, and $\pm 50°$ for roll. We roughly crop the faces in RGB images and then rescale them to 150×150 pixels for our experiment.

The main parameters of a GBF are the numbers T of stages, the dimension F of features generated for training, the fern depth M, learning rate α and the R feature subsets from which we select the best in each stage. Here we set $T = 5000, F = 10000, M = 5, \alpha = 0.05$ and $R = 20$.

Estimation accuracy. We perform a 4-fold cross validation on the dataset and compare the GBF regression with the support vector regression (SVR), which is a popular technique for head pose regression [1]. In the experiment, SVR is fed with the same features as GBF. The results in Table 1 show that the GBF regression outperforms SVR.

Running time performance. The GBF regression runs on an Intel Pentium 3.2GHz CPU with C++ implementation. It is extremely fast, taking only 0.55ms for an image.

4.2 2D facial pose regression

The dataset published most recently in [4] is used to verify the proposed algorithm. It contains 13,233 faces taken from the LFW database [9]. Different from earlier datasets like [10], this dataset contains outdoor faces with large variations in pose and illumination. The faces in this dataset are annotated with the locations of 10 facial landmarks (like the ones in Fig. 5a). 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 crop the faces with a face detection algorithm and then rescale the images to 150×150 pixels. The faces in this dataset are split into 5 subsets manually, based on the yaw angle of the head. We use this information for head pose regression by labeling them with real world angles $\omega \in \{-60, -30, 0, 30, 60\}$.

For the head pose regression, we use the same parameters as Section 4.1. As for the 2D facial pose regression, we use 20 GBFs in both level 2 and level 3. In each GBF, we set $T = 500, F = 256, M = 5$, and $R = 5$. Five view-based-GBF models are trained according to the yaw angles.

We perform a 10-fold cross validation on the dataset. Similar to most previous works, the localization error is normalized by inter-ocular distance. Table 2 presents the results of our method and two state-of-the-art ones [6, 4], showing that our method outperforms both methods at most of the landmarks. Besides, our method is much faster than the two competitors. The method in [6] cannot achieve real time performance and the method in [4] is reported to consume about 100ms for the accuracy listed in Table 2, while ours takes only about 30ms on a similar desktop computer.

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

Method	[6]	[4]	Ours
left eye left	16.21	6.82	5.78
left eye right	10.70	5.65	5.32
right eye left	9.37	5.67	5.40
right eye right	11.16	7.36	5.75
mouth left	10.76	7.38	7.13
mouth right	15.14	7.80	7.30
nose strip left	10.85	5.92	6.69
nose strip right	12.08	7.05	6.71
upper outer lip	-	6.40	6.69
lower outer lip	-	9.53	8.52

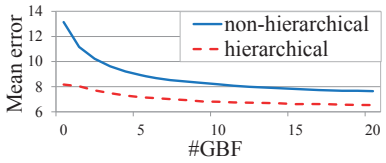

Figure 4: Convergence of the localization errors ($\times 10^{-2}$) by hierarchical and non-hierarchical approaches.

To further demonstrate the effectiveness of the proposed hierarchical approach, we compare it with non-hierarchical pose regression, which skips the first 2 levels and estimate the landmarks directly. The mean errors of the two approaches in the facial landmark level are shown in Fig. 4. For the hierarchical approach, the initial error is much less and we obtain better results. Fig. 4 also demonstrates the convergence of the algorithm. The mean errors decrease gradually, converge with around 20 GBFs and does not overfit. It shows that we do not need to carefully tune the number of GBF for the algorithm.

Fig. 5 presents some results of our algorithm on the test images and AFLW dataset [11]. It shows that the algorithm can deal with appearance variations caused by head rotations, facial expressions and partial occlusions.

5. CONCLUSIONS

In this paper, we split the facial pose into three different levels and propose a real time hierarchical pose regression approach for facial landmark localization. The experiments show that the algorithm runs faster and obtains higher accuracy than the state-of-the-art algorithms. Besides, we propose a unified cascaded fern regression framework based on generalized gradient boosted ferns. The discriminative power, computation efficiency and convergence are demonstrated in the tests.

6. ACKNOWLEDGEMENTS

This work was supported by grants from Natural Science Foundation of China (61201443, 61070148), Science, Industry, Trade, and Information Technology Commission of Shenzhen Municipality, China (JC201005270378A), and Guangdong Innovative Research Team Program (No. 201001 D0104648280).

7. REFERENCES

[1] C. BenAbdelkader. Robust head pose estimation using supervised manifold learning. In *ECCV*, 2010.

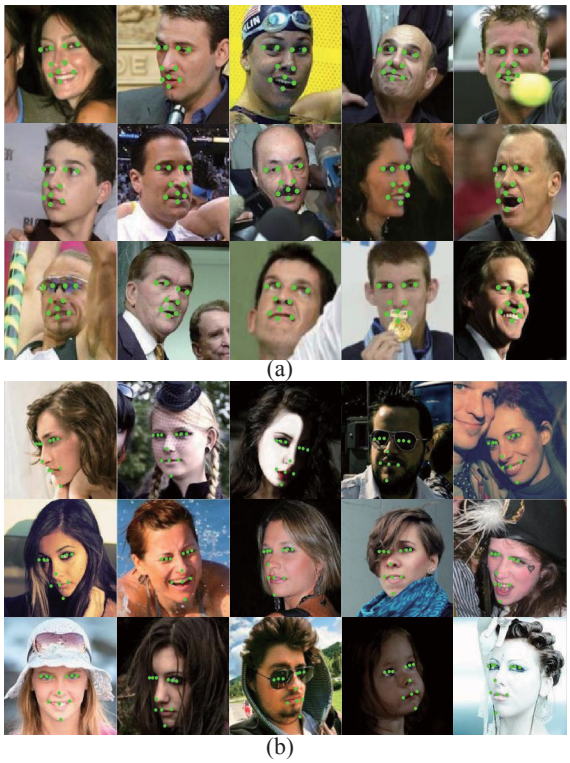

Figure 5: Some results on the LFW(a) and AFLW(b) dataset.

[2] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE TPAMI*, 23(6):681 –685, 2001.

[4] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012.

[5] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010.

[6] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy-automatic naming of characters in tv video. In *BMVC*, 2006.

[7] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Gool. Random forests for real time 3d face analysis. *IJCV*, 101:437–458, 2013.

[8] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[9] G. B. Huang, M. Ramesh, T. Berg, and 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.

[10] O. Jesorsky, K. J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, 2001.

[11] M. Kostinger, P. Wohlhart, P. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV Workshops*, 2011.