Locating Facial Landmarks Using Probabilistic Random Forest

Changwei Luo, Zengfu Wang, Shaobiao Wang, Juyong Zhang, and Jun Yu

***Abstract*—Random forest is a useful tool for face alignment/tracking. The method of regressing local binary features learned from random forest has achieved state-of-the-art performance both in fitting accuracy and speed. Despite the great success of this method, it has certain weaknesses: the number of available local binary features is rather limited and is not optimal for face alignment; the binary features inevitably lead to serious jitter when tracking a video sequence. To address these problems, we propose learning probability features from probabilistic random forest (PRF). The proposed PRF is the same as standard random forest except that it models the probability of a sample belonging to the nodes of a tree. By using the probability features, our method significantly outperforms the state-of-the-art in terms of accuracy. It also achieves about 60 fps for locating a few facial landmarks. In addition, our method shows excellent stability in face tracking.**

***Index Terms*—Face alignment, local binary features, probabilistic random forest, probability features.**

I. Introduction

LOCATING semantic facial landmarks is highly required in many applications, such as face recognition and facial animation [1]. Due to large variations of head pose, facial expressions, illumination and occlusions, automatic and accurate detection of facial landmarks is still a challenging task.

Active appearance model (AAM) [2], [3] solves the task of face alignment by modeling the holistic appearance of a face. The appearance model of AAM has quite weak generalization capability as the texture space is too large to be modeled using a limited number of training images. Compared with AAM, constrained local model (CLM) [4], [5] only models the local appearance around the landmarks. This leads to better generalization capability and robustness. CLM learns a set of local detectors independently. The local detectors are used to generate a response map for each landmark. A parametric shape model is then fitted to these response maps.

In recent years, shape regression based methods have show impressive results [6]–[8]. Asthana *et al.* [9] proposes a discriminative regression based approach in the framework of CLM. It learns robust functions from response maps to the shape parameters updates. This approach relies on a parametric shape model and minimizes model parameter error in training. The parametric shape model is usually learned from training shapes using principal component analysis (PCA). In [7], the shape constrain is adaptively enforced in the process of regression without using a parametric shape model. Compared with a PCA based shape model, non-parametric shape model can express face shapes in more details.

A critical issue for shape regression is what kind of image features should be used. In [10], SIFT features are used. Yan *et al.* [11] have compared several local feature descriptors and find that HOG (histogram of oriented gradient) shows best performance. Although the hand-crafted features work well, they are general purpose features and not optimal for face alignment. The method of [7] jointly learns the image features and regression functions in a fern based framework. In [12], cascaded convolution neural networks are used to learn the features and the regression functions.

Random forests (or regression trees) are popular for shape regression [13], [14]. Ren *et al.* [15] propose a two-step approach based on random forests. The method has achieved state-of-the-art performance both in accuracy and speed. Despite the great success of this method, it has several drawbacks. Firstly, for a local binary feature vector, the number of non-zero elements is equivalent to the number of trees in the forest. Therefore, the number of available local binary features is rather limited compared with the huge space of face texture. This results in many-to-one mappings from face texture to local binary features, i.e., different texture would correspond to the same shape increments. This may be not true in many cases. Secondly, the method suffers from serious jitter problem when tracking a video sequence. This is reasonable due to the binary features. Although smoothing the tracking results would alleviate this problem to a certain extent, the tracking accuracy will deteriorate, and there are still notable jitters in the tracking.

To address the above problems, we propose learning probability features from probabilistic random forest (PRF). The

Manuscript received July 18, 2015; revised September 06, 2015; accepted September 08, 2015. Date of publication September 22, 2015; date of current version September 25, 2015. This work was supported by the National Natural Science Foundation of China under Grants 61472393 and 61303150, and by the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University under Grant A1501. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jie Liang.

C. Luo, Z. Wang, and J. Yu are with Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: luocw@mail.ustc.edu.cn; zfwang@ustc.edu.cn; harryjun@ustc.edu.cn).

Z. Wang is also with the Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China.

S. Wang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: shaobiao@mail.ustc.edu.cn).

J. Zhang is with the School of Mathematical Sciences, University of Science and Technology of China, Hefei 230027, China (e-mail: juyong@ustc.edu.cn).

(*Corresponding author: J. Yu.*)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2480758

proposed PRF is the same as standard random forest except that it models the probability of a sample belonging to the tree nodes. By using the probability features, our method achieves state-of-the-art performance. It also achieves about 60 frames per second for locating a few facial landmarks. In addition, the jitter problem is solved essentially.

II. Review of Shape Regression

Regressing the face shape is very challenging in the presence of large image appearance variations. Cascaded regression has proven to be effective for improving the accuracy of shape regression. Cascaded regression combines T regressors in an additive manner. Let R^t denotes the tth regressor, $t = 1, 2, \ldots, T$. The face shape is denoted with vector S, which contains the x, y-coordinates of K facial landmarks. Given an image I and an initial shape estimation S^0, the regressor R^t computes a shape increment ΔS^t. This increment is then used to update the previous shape estimation S^{t-1},

$$S^t = S^{t-1} + \Delta S^t \tag{1}$$

S^t is the current shape estimation. ΔS^t is predicted as follows,

$$\Delta S^t = W^t \Phi^t(I, S^{t-1}) \tag{2}$$

where W^t is a regression matrix. $\Phi^t(I, S^{t-1})$ is a feature mapping function, and is computed from I and indexed to S^{t-1}.

In [15], $\Phi^t(I, S^{t-1})$ and W^t are learned using a two-step learning framework. Firstly, it learns local binary features (LBF) for each landmark independently. Secondly, it performs global linear regression for all landmarks. In the following, we explain these two steps in more details. The index t is dropped for notational brevity.

Local binary features. To lean Φ for each landmark, Φ is decomposed into a set of local mapping functions [15],

$$\Phi = [\varphi_1; \varphi_2; \ldots; \varphi_K] \tag{3}$$

where φ_k is the mapping function for the kth landmark. φ_k is learned by using standard regression forest. The regression target is a 2D vector $\Delta \hat{S}^{(k)}$, which is the ground truth shape increment for the kth landmark. The split nodes of a tree are trained using pixel difference feature. After training, each leaf node stores a 2D vector that is the average of all training samples in the leaf node.

During testing, a sample traverses a tree until it reaches one leaf node. If the number of leaf nodes is m, then φ_k is a m-dimensional binary vector. For each element of φ_k, its value is set to 1 if the sample reaches the corresponding leaf node and 0 otherwise. Thus, φ_k is a rather sparse binary vector, and is called local binary feature.

Global regression. After local learning of the binary features using random forest, global linear regression is performed on the ground truth shape increments and the local binary features [15], and the regression matrix W is obtained.

III. Probability Feature Regression

In this section, we first describe the proposed probabilistic random forest. Then we show how probability features are obtained and used in shape regression.

A. Probabilistic Random Forest

The proposed probabilistic random forest is built in the same way as the standard random forest. However, for each split node of the trees in probabilistic random forest, we need to model the probability of a sample belonging to the left and right child nodes. We call these trees probabilistic trees.

Following the method of [15], we independently train a random forest for each landmark. After training samples stored in the split node have been assigned to left and right child nodes. The probability of a sample belonging to the right child node is modeled by fitting a logistic regression function,

$$p(l = 1|g) = \frac{1}{1 + \exp(-\alpha(g - g_{split}))} \tag{4}$$

where g is the pixel difference feature [7]. g_{split} is the splitting threshold. $l \in \{1, -1\}$, $l = 1$ means the training sample belonging to the right child node. α is a constant parameter. The probability of the sample belonging the left child node $p(l = -1|g)$ is computed as follows,

$$p(l = -1|g) = 1 - p(l = 1|g) \tag{5}$$

Determining the value of α. The probability $p(l = 1|g)$ can be simply set to 1.0 for training samples in the right child node, and 0 for samples in the left child node. Then, the parameter α can be determined by fitting equation (4) to the training samples. The problem is that both the number of split nodes and the number of training samples in the split nodes are very large. It would take much time to train the logistic regressors for all split nodes. Moreover, for pixel difference feature with a value close to the splitting threshold, even a small perturbation would lead to extremely larger probability changes.

To over these problems, we compute the mean pixel difference feature $\bar{g}_{right}$ for samples in the right child node, and set the corresponding probability $p(l = 1|\bar{g}_{right})$ to a constant $\beta (0.5 < \beta \leq 1)$. Similarly, the mean pixel difference feature for left child node is denoted as $\bar{g}_{left}$, and $p(l = 1|\bar{g}_{left})$ is set to $1 - \beta$. Given β, we solve for α by minimizing the following objective function:

$$E(a) = (p(l = 1|\bar{g}_{right}) - \beta)^2 + (p(l = 1|\bar{g}_{left}) - 1 + \beta)^2 \tag{6}$$

β is determined experimentally.

B. Probability Features

During testing, a sample is sent to all trees of the PRF. For each tree, we calculate the probability of the sample reaching **each** leaf node. The probability for one leaf node is denoted as p^{leaf}, and is computed as follows,

$$p^{leaf} = \prod_{q \in Q} p(l_q|g) \tag{7}$$

where Q is the set of split nodes on the path where the sample goes from tree root to a leaf node. For split node q, $l_q = 1$ if the path goes from node q to its right child node, and $l_q = -1$ otherwise. We call p^{leaf} the probability feature of a leaf node. The output of a leaf node is written as

$$\omega \cdot p^{leaf} \tag{8}$$

where ω is the average of all training samples stored in the leaf node. The output of a tree is the summation of the outputs of all leaf nodes. Compared with standard regression trees, the predicted output of our probabilistic trees is continuous and depends on all leaf nodes. This makes the prediction more accurate and reliable.

For the probabilistic random forest corresponding to kth landmark, the learned probability feature vector is denoted as

$$P_k^{leaf} = [p_{k,1}^{leaf}; p_{k,2}^{leaf}; \ldots; p_{k,m}^{leaf}] \tag{9}$$

where m is the number of leaf nodes in the random forest. $p_{k,j}^{leaf}$ is the probability for the jth leaf node, and is computed using equation (7). We call P_k^{leaf} the probability features of the kth landmark. By concatenating the probability features of all landmarks, we obtain the probability features of a shape,

$$\mathrm{P}^{leaf} = [P_1^{leaf}; P_2^{leaf}; \ldots; P_K^{leaf}] \tag{10}$$

Since the outputs $\omega \cdot p^{leaf}$ are learned independently for each landmark, they are noisy and lack enough reliability. To enforce global shape constrain, we perform global linear regression on the ground truth shape increments and the probability features. The regression matrix W is obtained by minimizing the following objective function,

$$E(W) = \sum_{i=1}^{n} \|\Delta\hat{S}_i - W\mathrm{P}^{leaf}\|^2 + \lambda\|W\|^2 \tag{11}$$

where n is the number of training samples, $\Delta\hat{S}_i$ is the ground truth shape increment for the ith training sample. The parameter λ controls strength of the regularization term.

IV. Experiments

To evaluate the performance of our method, we first conduct face alignment experiments on two standard databases: LFPW (labeled face parts in the wild) [16] and 300-W [17]. Then, we apply our method for facial feature tracking.

A. Face Alignment

LFPW. LFPW database [16] consists of the URLs to 1100 training and 300 test images that can be downloaded from internet. We were able to download only 813 training images and 224 test images because some of the URLs are no longer valid. These images were manually annotated with 68 points to generate the ground-truths [17].

In our experiments, the initial shape is chosen as the mean shape of the training data, translated and scaled according to the output rectangle of a face detector. The fitting accuracy is measured by the normalized landmark error [15], which is calculated as the average distance between the detected landmarks and the ground truth landmarks, normalized by the inter-pupil distance. The error is represented as the percentage of inter-pupil distance, we drop the notation % for brevity.

For the proposed method, the most important parameter is β. We evaluate the impact of different values of β on fitting accuracy. The experiments are performed with the following parameter setting. The number of regressors in the cascade is $T = 5$. The number of trees in each regressor is $N = 300$, and the depth of the trees is $D = 4$.

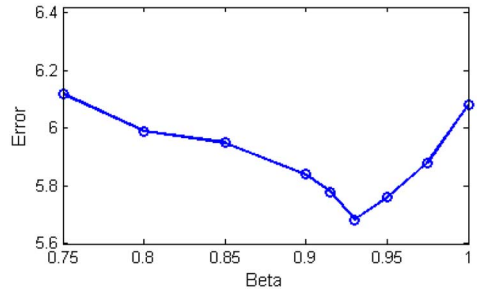

Fig. 1. The average error for different values of β.

TABLE I
The Average Errors of the Two Methods Under Different Parameter Settings

	$N = 300, T = 5$	$N = 600, T = 7$
LBF [15]	6.36	5.79
Proposed	**5.68**	**5.35**

Fig. 2. Selected results from LBF (top row) and the proposed method (bottom row) on the LFPW database.

Fig. 1 shows the experimental results. It is shown that the fitting accuracy is not very sensitive to is β. The smallest average error is achieved when β is set to about 0.93. This value is used in the rest of our experiments.

To evaluate the effectiveness of the probabilistic features, we compared our method with LBF [15]. Two parameter settings are used: (1) $N = 300, T = 5, D = 4$, (2) $N = 600, T = 7, D = 4$. Table I shows the errors for the two methods. We can see that the probability features significantly improve the results of LBF. In the first case, the error reduction with respect to LBF is over 10%. It is also worth mentioning that our method with 300 trees is still superior to LBF with 600 trees.

Fig. 2 shows a few results from LBF and the proposed method. We can see that our method performs better than LBF. Specifically, our method accurately localizes the outer landmarks while LBF fails. Our method requires running all the split tests of a tree, this procedure is about 4 times slower than evaluating a standard tree. In addition, the probability features are not able to benefit from efficiency of sparse computations, leading to lower efficiency. Our current implementation (300 trees) runs at about 65 fps in a desktop with an i7 CPU.

300-W. The 300-W database [17] is created from existing databases, including AFW [18], LFPW and Helen [19]. It also includes a new database called IBUG. The IBUG database is extremely challenge as the images of the database exhibit large

TABLE II
COMPARISON WITH [15]. IN ALL CASES, ONLY 300 TREES ARE USED. THE NUMBER WITHIN THE BRACKET DENOTES THE DEPTH OF THE TREES

Method	[15]	LBF	Proposed (4)	Proposed (5)
Error	7.37	7.58	**6.92**	**6.84**

variations such as illuminations, expressions and occlusions. All images in 300-W are labeled with 68 landmarks.

Following the methodology of [15], we split the 300-W database into two parts for our training and testing. The training set consists of AFW, the training set of LFPW and the training set of Helen. The testing set consists of the testing set of LFPW, the testing set of Helen, and IBUG. We conduct two experiments on 300-W database.

In the first experiments, the parameters are set as follows. $N = 300$ and $T = 5$. The depth of the trees is set to $D = 4$ and $D = 5$, respectively. We also compared our method with LBF. The parameter settings of LBF are the same as the proposed method except that the depth of the trees is set to $D = 5$.

Table II shows the experimental results for LBF, both the results reported by [15] and the results for our replication of [15] are shown. We can see that our replication achieves similar accuracy as reported by [15]. It is also shown that LBF with $D = 5$ is still inferior to our method with $D = 4$. For our method, the dimension of the probability features is doubled when D is increased from 4 to 5. However, the error reduction is marginal. This is probably due to over-fitting or the lack of sufficient training samples. Therefore, $D = 4$ is a good compromise between accuracy and efficiency.

In the second experiments, our testing set is further divided into two subsets: the common subset (consisting of the testing sets of LFPW and Helen) and the challenging subset (consisting of IBUG). We report the experimental results on the full testing set and the two subsets. In addition to [15], we also compared our method with the following methods:

(1) DRMF (discriminative response map fitting) [9];
(2) ESR (explicit shape regression) [7];
(3) SDM (supervised descent method) [10];
(4) ERT (ensemble of regression trees) [20];

The comparison results are show in Table III. We can see that our method shows the best performance. Note that 1200 trees with $D = \mathbf{7}$ are used in LBF [15]. Our method only uses 600 trees with $D = \mathbf{4}$. Nevertheless our method achieves accuracy superior to LBF. We believe that this is because the probability features are more accurate than the local binary features. The space of probability features is large and continuous, while the space of the binary features is small and discrete. From a probabilistic point of view, the local binary features are a simplified version of the probability features: the leaf node which a sample falls into outputs probability 1 while the other leaf nodes output probability 0.

B. Facial Feature Tracking in a Sequence

Evaluating the performance of fitting algorithms using videos is meaningful as many applications require accurate and stable face tracking. We manually labeled 6 videos with 68 landmarks. The number of frames of the videos ranges from 120 to 1500. The tracking model is trained on the 300-W database. To track the facial landmarks, the initial shape for first image frame is obtained based on the mean shape and the output rectangle of a face detector. For the following frames, the initial shape is obtained by transforming the mean shape to the tracking results of the previous frame.

TABLE III
COMPARISON OF AVERAGE ERROR WITH STATE-OF-THE-ART METHODS. THE RESULTS FOR THESE METHODS ARE OBTAINED DIRECTLY FROM THE LITERATURES OR EVALUATED BASED ON THE RELEASED CODES

Method	Common subset	Challenging subset	Full set	fps
DRMF [9]	6.65	19.79	9.22	1
ESR [7]	5.28	17.00	7.58	120
SDM [10]	5.60	15.40	7.52	70
ERT [20]	-	-	6.40	-
LBF [15]	4.95	11.98	6.32	320
Proposed	4.90	11.96	**6.28**	30

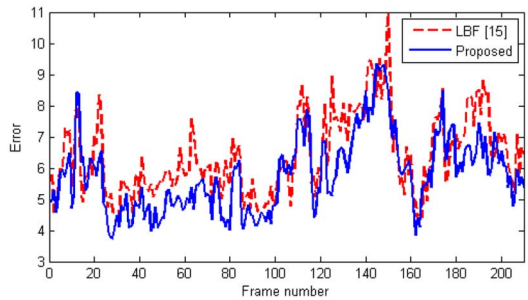

Fig. 3. The landmark error as a function of frame number.

Fig. 3 shows the landmark error as a function of frame number for an annotated video. We can see that our method gives better results than LBF [15]. Please refer to the accompanying video to better appreciate the performance of our method. From the tracking results of LBF, we can observe serious sudden jumps of tracked landmarks. In contrast, our method shows excellent stability throughout the sequence.

V. CONCLUSION

We propose probabilistic random forest or probabilistic trees for locating facial landmarks. The probabilistic trees distinguish from standard trees in that they model the probability of a sample reaching each leaf node. By regressing the probability features, our method achieves fitting accuracy superior to state-of-the-art methods. Since the output of a probabilistic tree is continuous and depends on all leaf nodes of the tree, it expands the expression capability of a standard tree. We believe that the probabilistic trees will be a useful tool for solving many regression problems.

ACKNOWLEDGMENT

The authors would like to thank Shaoqing Ren for his help in the experiments.

REFERENCES

[1] C. Luo, C. Jiang, J. Yu, and Z. Wang, "Expressive facial animation from videos," in *IEEE Int. Conf. Image Processing*, 2014, pp. 4617–4621.

[2] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 23, pp. 681–685, 2001.

[3] Y. Chen, F. Yu, and C. Ai, "Sequential active appearance model based on online instance learning," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 567–570, 2013.

[4] D. Cristinacce and T. Cootes, "Feature detection and tracking with constrained local models," in *Brit. Machine Vision Conf.*, 2006, pp. 929–938.

[5] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *IEEE Int. Conf. Computer Vision Workshops*, 2013, pp. 354–361.

[6] P. Dollar, P. Welinder, and P. Perona, "Cascaded pose regression," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2010, pp. 1078–1085.

[7] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2012, pp. 2887–2894.

[8] Z. Feng, P. Huber, J. Kittler, W. Christmas, and X. Wu, "Random cascaded-regression copse for robust facial landmark detection," *IEEE Signal Process. Lett.*, vol. 22, no. 1, pp. 76–80, 2015.

[9] A. Asthana, S. Zafeiriou, and S. C. M. Pantic, "Robust discriminative response map fitting with constrained local models," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2013, pp. 3444–3451.

[10] X. Xiong and F. D. la Torre, "Supervised descent method and its applications to face alignment," in *IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2013, pp. 532–539.

[11] J. Yan, Z. Lei, D. Yi, and S. Z. Li, "Learn to combine multiple hypotheses for accurate face alignment," in *IEEE Int. Conf. Computer Vision Workshops on 300-W Challenge, ICCVW*, 2013, pp. 392–396.

[12] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2013, pp. 3476–3483.

[13] T. Cootes, M. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *Brit. Machine Vision Conf.*, 2012.

[14] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2012, pp. 2578–2585.

[15] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2014, pp. 1685–1692.

[16] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2011.

[17] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *IEEE Int. Conf. Computer Vision Workshops*, 2013.

[18] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Computer Soc. Conf. Computer Vision and Pattern recognition*, 2012, pp. 2879–2886.

[19] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *12th Eur. Conf. Computer Vision (ECCV)*, 2012.

[20] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.