# Face Alignment at 3000 FPS via Regressing Local Binary Features

Shaoqing Ren Xudong Cao
University of Science and Technology of China
sqren@mail.ustc.edu.cn

Yichen Wei Jian Sun
Microsoft Research
{xudongca,yichenw,jiansun}@microsoft.com

## Abstract

*This paper presents a highly efficient, very accurate regression approach for face alignment. Our approach has two novel components: a set of local binary features, and a locality principle for learning those features. The locality principle guides us to learn a set of highly discriminative local binary features for each facial landmark independently. The obtained local binary features are used to jointly learn a linear regression for the final output. Our approach achieves the state-of-the-art results when tested on the current most challenging benchmarks. Furthermore, because extracting and regressing local binary features is computationally very cheap, our system is much faster than previous methods. It achieves over 3,000 fps on a desktop or 300 fps on a mobile phone for locating a few dozens of landmarks.*

## 1. Introduction

Discriminative shape regression has emerged as the leading approach for accurate and robust face alignment [5, 11, 12, 29, 4, 32, 3, 27]. This is primarily because these approaches have some distinct characteristics : 1) they are purely discriminative; 2) they are able to enforce shape constrain adaptively; 3) they are capable of effectively leveraging large bodies of training data.

The shape regression approach predicts facial shape $S$ in a cascaded manner [12, 5, 4, 32, 3]. Beginning with an initial shape $S^0$, $S$ is progressively refined by estimating a shape increment $\Delta$S stage-by-stage. In a generic form, a shape increment $\Delta S^t$ at stage $t$ is regressed as:

$$\Delta \mathrm{S}^t = W^t \Phi^t \left(I, S^{t-1}\right), \quad (1)$$

where $I$ is the input image, $S^{t-1}$ is the shape from the previous stage, $\Phi^t$ is a feature mapping function, and $W^t$ is a linear regression matrix. Note that $\Phi^t$ depends on both $I$ and $S^{t-1}$. The feature learned in this way is referred to as a "shape-indexed" feature [5, 3]. The regression goes to the next stage by adding $\Delta S^t$ to $S^{t-1}$.

The feature mapping function $\Phi^t$ is essential in shape regression. In previous works, it is either designed by hand [32] or by learning [5, 3]. The process in [32] simply uses SIFT features for feature mapping and trains $W^t$ by a linear regression. While this simple approach works well, the handcrafted general purpose features are not optimal for specific face alignment. In contrast, the processes in [5, 3] jointly learn both $\Phi^t$ and $W^t$ by a tree-based regression, on the whole face region in a data-driven manner.

In principle, the latter learning-based approach should be better because it learns task-specific features. However, as reported in existing literature, it is only on par with the approach using a hand-designed SIFT feature. We believe this is due to two issues caused by the overly high freedom of $\Phi^t$. The first is a practical issue. Using the entire face region as the training input results in an extremely large feature pool, which translates into unaffordable training costs if we want to learn the most discriminative feature combination. The second is a generalization issue, which is more crucial. The large feature pool has many noisy features. This can easily cause over fitting and hurt performance in testing.

In this work, we propose a better learning based approach. It regularizes learning with a "locality" principle. This principle is based on two insights: for locating a certain landmark at a stage, 1) the most discriminative texture information lies in a local region around the estimated landmark from the previous stage; 2) the *shape context* (locations of other landmarks) and *local texture* of this landmark provide sufficient information. These insights imply that we may first learn intrinsic features to encode the local texture for each landmark independently, then perform joint regression to incorporate the shape context.

We propose the following two types of regularization for learning $\Phi^t$:

- $\Phi^t$ is decomposed into a set of independent local feature mapping functions, i.e. $\Phi^t = [\phi_1^t, \phi_2^t, ..., \phi_L^t]$ ($L$

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.218

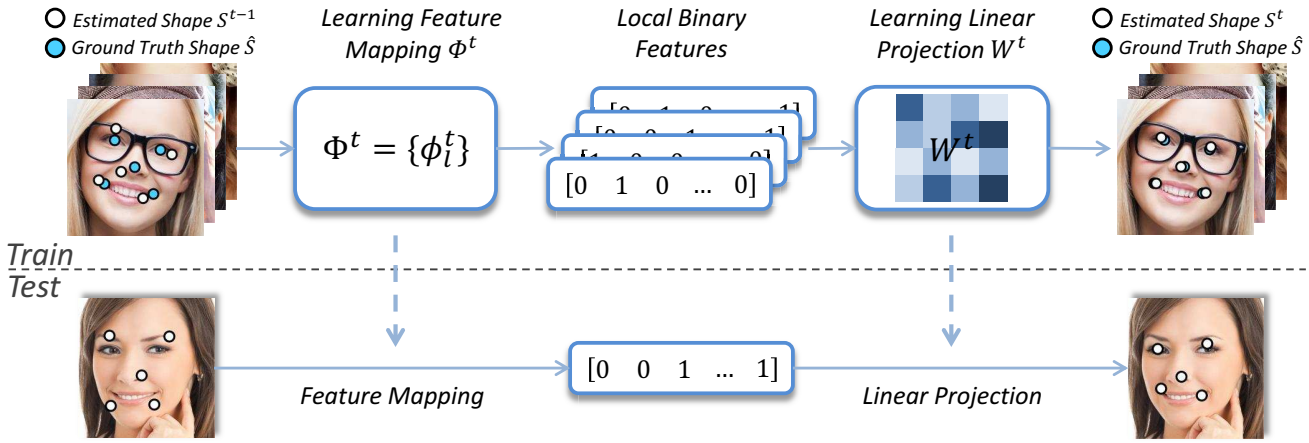


Figure 1. Overview of our approach. In the training phase, we begin by learning a feature mapping function $\Phi^t(I_i, S_i^{t-1})$ to generate local binary features. Given the features and target shape increments $\{\Delta\hat{S}_i^t = \hat{S}_i - S_i^{t-1}\}$, we learn a linear projection $W^t$ by linear regression. In the testing phase, the shape increment is directly predicted and applied to update the current estimated shape.

is the number of landmarks).

- Each $\phi_l^t$ is learned by independently regressing $l$th landmark, in the corresponding *local* region.

The proposed regularization can effectively screen out the majority of noisy or less discriminative features, reduce learning complexity, and lead to better generalization.

To learn each $\phi_l^t$, we use ensemble trees based regression to *induce* binary features. The binary features encode the intrinsic structure in a local region, for predicating the landmark position. After concatenating all *local binary features* to form the feature mapping $\Phi^t$, we discriminatively learn $W^t$ for global shape estimation. We find that our two-step learning process (local binary features and global linear regression) is much better than the one-step joint learning of $\Phi^t$ and $W^t$ by tree-based regression in [5, 3].

In addition to better accuracy, our approach is also much more efficient. Because the local binary features are tree based and highly sparse, the process of extracting and regressing such features is extremely rapid. We show that a fast version of our approach runs at 3,000+ frames per second (FPS) on a single-core desktop and achieves comparable results with state-of-the-art methods. Our normal version runs at 300+ FPS and significantly outperforms state-of-the-art equivalents in terms of accuracy on a variety of benchmarks. The high speed of our approach is crucial for scenarios and devices where computational power is limited and computational budget is a major concern. For example, our fast version still runs at 300 FPS on a modern mobile phone. To the best of our knowledge, this is the first approach that is several times faster than real-time face alignment approach on mobile phone. This opens up new opportunities for all online face applications.

## 2. Related Works

Active Appearance Models (AAM) [7] solves the face alignment problem by jointly modeling holistic appearance and shape. Many improvements over AAM have been proposed [19, 18, 14, 15, 25, 28]. Instead of modeling holistic appearance, "Constrained Local Model" [8, 9, 10, 1, 35, 29, 34, 26] learns a set of local experts (detectors [9, 31, 24, 1, 34] or regressors [10, 29, 11]) and constrains them using various shape models. These approaches are better for generalization and robustness.

Our work belongs to the shape regression approach [5, 11, 12, 29, 4, 32, 3] category. Xiong *et al*. [32] predict shape increment by applying linear regression on SIFT features. Both Cao *et al*. [5] and Burgos-Artizzu *et al*. [3] use boosted ferns (a kind of tree) to regress the shape increment. We note that the ensemble tree-based methods (either boosted trees or random forest) can also be viewed as a linear summation of regressors using binary features induced by the trees, yet, our feature learning method differs from previous tree based methods.

Ensemble trees can be used as a codebook for efficient encoding [22] or learning better descriptors [6, 33]. Ensemble trees have recently been exploited for direct feature

mapping to handle non-linear classification [30, 16]. In this work, we demonstrate the effectiveness of ensemble trees induced features in shape regression.

## 3. Regressing Local Binary Features

In Equation (1), both the linear regression matrix $W^t$ and the feature mapping function $\Phi^t$ are unknown. In our approach, we propose learning them in two consecutive steps. We first learn a local feature mapping function to generate local binary features for each landmark. We concatenate all local features to get $\Phi^t$. Then we learn $W^t$ by linear regression. This learning process is repeated stage-by-stage in a cascaded fashion. Figure 1 shows the overview of our approach.

### 3.1. Learning local binary features $\Phi^t$

The feature mapping function is composed of a set of local feature mapping functions i.e., $\Phi^t = [\phi_1^t, \phi_2^t, ..., \phi_L^t]$. We learn each of them independently. The regression target for learning $\phi_l^t$ is the ground truth shape increment $\Delta\hat{S}^t$:

$$\min_{w^t, \phi_l^t} \sum_{i=1} \| \pi_l \circ \Delta\hat{S}_i^t - w_l^t \phi_l^t \left( I_i, S_i^{t-1} \right) \|_2^2, \tag{2}$$

where $i$ iterates over all training samples, operator $\pi_l$ extracts two elements $(2l-1, 2l)$ from the vector $\Delta\hat{S}_i$, and $\pi_l \circ \Delta\hat{S}_i$ is the ground truth 2D-offset of $l$th landmark in $i$th training sample.

We use a standard regression random forest [2] to learn each local mapping function $\phi_l^t$. The split nodes in the trees are trained using the pixel-difference feature [5, 3]. To train each split node, we test 500 randomly sampled features and pick the feature that gives rise to maximum variance reduction. Testing more features results in only marginal improvement in our experiment. After training, each leaf node stores a 2D offset vector that is the average of all the training samples in the leaf.

We only sample pixel features in a local region around the landmark that is estimated. Using such a local region is critical to our approach. In the training, the optimal region size is estimated in each stage via cross validation. We will discuss more details in Section 3.3.

During testing, a sample traverses the trees until it reaches one leaf node for each tree. The output of the random forest is the summation of the outputs stored in these leaf nodes. Supposing the total number of leaf nodes is $D$, the output can be rewritten as:

$$w_l^t \phi_l^t \left( I_i, S_i^{t-1} \right), \tag{3}$$

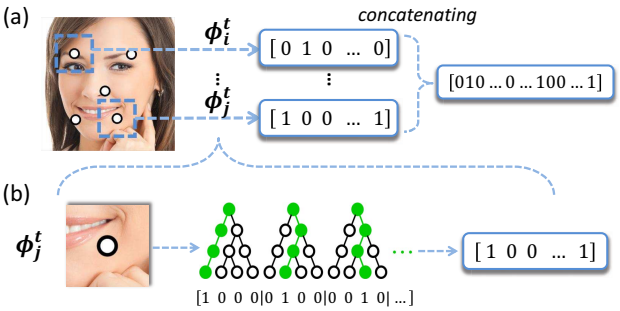


Figure 2. Local binary features. (a) The local feature mapping function $\phi_l^t$ encodes the corresponding local region into a binary feature; all local binary features are concatenated to form high-dimensional binary features. (b) We use random forest as the local mapping function. Each extracted binary feature indicates whether the input image contains some local patterns or not.

where $w_l^t$ is a 2-by-$D$ matrix in which each column is the 2D vector stored in the corresponding leaf node, and $\phi_l^t$ is a $D$-dimensional binary vector. For each dimension in $\phi_l^t$, its value is 1 if the test sample reaches the corresponding leaf node and 0 otherwise. Therefore, $\phi_l^t$ is a very sparse binary vector. The number of non-zero elements in $\phi_l^t$ is the same as the number of trees in the forest, which is much smaller than $D$. We call such $\phi_l^t$s "*local binary features*". Figure 2 illustrates the process of extracting local binary features.

### 3.2. Learning global linear regression $W^t$

After the local random forest learning, we obtain not only the binary features $\phi_l^t$, but also the local regression output $w_l^t$. We *discard* such learned local output $w_l^t$. Instead, we concatenate the binary features to a global feature mapping function $\Phi^t$ and learn a global linear projection $W^t$ by minimizing the following objective function:

$$\min_{W^t} \sum_{i=1}^{N} \| \Delta\hat{S}_i^t - W^t \Phi^t (I_i, S_i^{t-1}) \|_2^2 + \lambda ||W^t||_2^2, \tag{4}$$

where the first term is the regression target, the second term is a L2 regularization on $W^t$, and $\lambda$ controls the regularization strength. Regularization is necessary because the dimensionality of the features is very high. In our experiment, for 68 landmarks, the dimensionality of $\Phi^t$ could be 100K+. Without regularization, we observe substantial overfitting. Because the binary features are highly sparse, we use a dual coordinate descent method [13] to deal with such a large-scale sparse linear system. Since the objective function is quadratic with respect to $W^t$, we can always reach its global optimum.

We find that such global "relearning" or "transfer learning" significantly improves performance. We believe this is

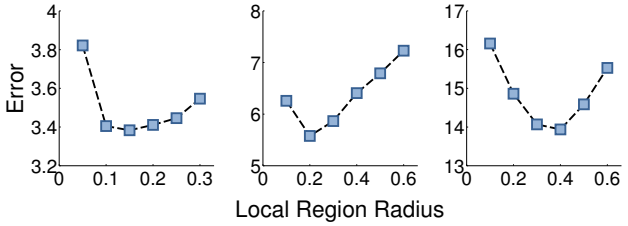


Figure 3. The horizontal axis stands for the local region radius. The vertical axis stands for the alignment error on a test set. From left to right, standard deviation of the distribution of $\Delta s$ are 0.05, 0.1, 0.2. Herein both local region radius and alignment error is normalized by the size of face rectangle.

for two reasons. On one hand, the locally learned output by random forest is noisy because the number of training samples in a leaf node may be insufficient. On the other hand, the global regression can effectively enforce a global shape constraint and reduce local errors caused by occlusion and ambiguous local appearance.

### 3.3. Locality principle

As we have described previously, we apply two important regularization methods in feature learning, as guided by a locality principle: 1) we learn a forest for each landmark independently; 2) we only consider the pixel features in the local region of a landmark. In this section, we explain why we made such choices.

**Why the local region?** We begin with the second choice. Suppose we want to predict the offset $\Delta s$ of a single landmark and we select features from a local region with radius $r$. Intuitively, the optimal radius $r$ should depend on the distribution of $\Delta s$. If $\Delta s$ of all training samples are scattered widely, we should use a large $r$; otherwise we use a small one.

To study the relationship between the distribution of $\Delta s$ and the optimal radius $r$, for a landmark we synthesize training and test sample regions whose $\Delta s$ follow a Gaussian distribution with different standard deviations. For each distribution, we experimentally determine the optimal region radius (in terms of test error) by training regression forests on various radii. We use the same forest parameters (tree depth and number of trees) as in our cascade training. We repeat this experiment for all landmarks and take the average of the optimal region radius.

Figure 3 shows the results of three distributions whose std. are 0.05, 0.1, and 0.2 (normalized distance by face rectangle size). The optimal radiuses are 0.12, 0.21 and 0.39. The results indicate that the optimal region radius is almost linearly to the standard deviation of $\Delta s$. Therefore, we can conclude that, given limited computation budget (the number of features tested in training forests), it is more effective to only consider candidate features in a local region instead of the global face image.

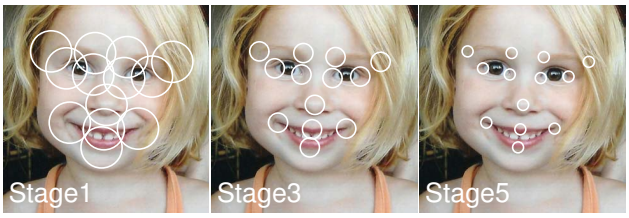


Figure 4. The best local region sizes at stage 1, 3, and 5.

In our cascade training, at each stage, we search for the best region radius (from 10 discrete values) by cross-validation on an hold-out validation set. Figure 4 shows the best region radiuses found at stage 1, 3, and 5. As expected, the radius gradually shrinks from early stage to later stage, because the variation of regressed face shapes decreases during the cascade.

**Why a single landmark regression?** It may appear that independent regression of each landmark is sub-optimal. For example, we could probably miss a good feature that can be shared by multiple landmarks. However, we argue that local regression has a few advantages over the global learning such as in [5].

First, the feature pool in local learning is less noisy. There may be more useful features in global learning. But the "signal-to-noise ratio" in global learning could be lower, which will make feature selection more difficult.

Second, using local learning does not mean that we do local prediction. In our approach, the linear regression in the second step exploits all learned local features to make a global prediction. Because the local learning of landmarks is independent, the resulting features are by nature more diverse and complementary to each other. Such features are more appropriate for global learning in the second step.

Last, the local learning is adaptive in different stages. In the early stage, the local region size is relatively large and a local region actually covers multiple landmarks. The features learned from one landmark can indeed help its neighboring landmarks. In the late stage, the region size is small and local regression fine-tunes each landmark. Local learning is actually more appropriate in the late stage.

Note that we do not claim that global learning is inferior to our local learning by nature. We believe that local learning delivers better performance mainly due to practical reasons. Given limited training capability (the amount of training data, affordable training time, available computing resources, and power of learning algorithm), the local ap-

proach can better resist noisy features in the global feature pool, which is extremely large and may cause over fitting. We hope our empirical findings in this work can encourage more similar investigations in the future.

## 4. Experiments

**Datasets** There are quite a few datasets for face alignment. We use three more recent and challenging ones. They present different variations in face shape, appearance, and number of landmarks.

*LFPW* (29 landmarks) [1] is collected from the web. As some URLs are no longer valid, we only use 717 of the 1,100 images for training and 249 of the 300 images for testing. Although each image is labeled with 35 landmarks, we use 29 of 35 landmarks in our experiments, following previous work [5].

*Helen* (194 landmarks) [17] contains 2,300 high resolution web images. We follow the same setting in [17]: 2000 images for training and 330 images for testing. The high resolution is beneficial for high accuracy alignment, but the large number of landmarks is challenging in terms of computation.

*300-W* (68 landmarks) is short for 300 Faces in-the-Wild [23]. It is created from existing datasets, including LFPW [1], AFW [35], Helen [17], XM2VTS [20], and a new dataset called IBUG. It is created as a challenge and only provides training data. We split their training data into two parts for our own training and testing. Our training set consists of AFW, the training sets of LFPW, and the training sets of Helen, with 3148 images in total. Our testing set consists of IBUG, the testing sets of LFPW, and the testing sets of Helen, with 689 images in total. We do not use images from XM2VTS as it is taken under a controlled environment and is too simple. We should point out that the IBUG subset is extremely challenging as its images have large variations in face poses, expressions and illuminations.

**Evaluation metric** Following the standard [1, 5], we use the inter-pupil distance normalized landmark error. For each dataset we report the error averaged over all landmarks and images. Note that the error is represented as a percentage of the pupil-distance, and we drop the notation $\%$ in the reported results for clarity.

In the following section, we first compare our approach against state-of-the-art methods, then validate the proposed approach via comparison with certain baseline methods.

### 4.1. Comparison with state-of-the-art methods

During our training, we use similar data augmentation as in [5] to enlarge the training data and improve generalization ability: each training image is translated to multiple training samples by randomly sampling the initial shape multiple times. Note that during testing we only use the mean shape as the initialization. We do not use multiple initializations and median based refinement as in [5].

Our approach has a few free parameters: the number of stages $T$, the number of trees in each stage $N$[1], and the tree depth $D$. To test different speed-accuracy trade-offs, we use two sets of settings: 1) more accurate: $T = 5, N = 1200, D = 7$; and 2) faster: $(T = 5, N = 300, D = 5)$. We call the two versions *LBF* (local binary features) and *LBF fast*.

Our main competitors are the shape regression based methods, including explicit shape regression (ESR) [5] and supervised descent method (SDM) [32]. We implement these two methods and our implementation achieves comparable accuracy to that which was reported by the original authors. For comparison with other methods, we used the original results in the literature. Table 1 reports the errors and speeds (frames per second or FPS) of all compared methods on three datasets. Note that we also divide the testing set of 300-W into two subsets: the common subset consists of the testing sets of Helen and LFPW, and the challenging IBUG subset. We report all results on the two subsets as well.

**Comparison of accuracy** Overall, the regression-based approaches are significantly better than ASM-based methods. Our proposed approach LBF wins by a large margin over all datasets. Our faster version is also comparable with the previous best. Specifically, our method achieves significant error reduction with respect to ESR and SDM of 30% and 22%, respectively, on the challenging IBUG subset. We believe this is due to the good generalization ability of our method. In Figure 7–9, some example images and comparison results from IBUG are shown. Note that the performance on LFPW is almost saturated, because the human performance is 3.28 as reported in [3].

**Comparison of speed** Our approach, ESR, and SDM are all implemented in C++ and tested on a single core i7-2600 CPU. The speed of other methods is quoted from the original papers. While ESR and SDM are already the fastest face alignment methods in the literature, our method has a even larger advantage in terms of speed. Our fast version is dozens of times faster and achieves thousands of FPS for a large number of landmarks. The high speed comes from the sparse binary features. As each testing sample has only

[1] We fix the total number of trees so few trees will be used for each landmark if there are more landmarks.

| LFPW (29 landmarks) | | |
|---|---|---|
| Method | Error | FPS |
| [1] | 3.99 | $\approx 1$ |
| ESR[5] | 3.47 | 220 |
| RCPR[3] | 3.50 | - |
| SDM[32] | 3.49 | 160 |
| EGM[34] | 3.98 | $< 1$ |
| **LBF** | **3.35** | 460 |
| **LBF fast** | **3.35** | **4200** |

| Helen (194 landmarks) | | |
|---|---|---|
| Method | Error | FPS |
| STASM[21] | 11.1 | - |
| CompASM[17] | 9.10 | - |
| ESR[5] | 5.70 | 70 |
| RCPR[3] | 6.50 | - |
| SDM[32] | 5.85 | 21 |
| **LBF** | **5.41** | 200 |
| **LBF fast** | 5.80 | **1500** |

| 300-W (68 landmarks) | | | | |
|---|---|---|---|---|
| Method | Fullset | Common Subset | Challenging Subset | FPS |
| ESR[5] | 7.58 | 5.28 | 17.00 | 120 |
| SDM[32] | 7.52 | 5.60 | 15.40 | 70 |
| **LBF** | **6.32** | **4.95** | **11.98** | 320 |
| **LBF fast** | 7.37 | 5.38 | 15.50 | **3100** |

Table 1. Error and runtime (in FPS) on LFPW, Helen and 300-W datasets, respectively. The errors of ESR and SDM are from our implementation. Note that ESR and SDM have reported error on LFPW in the original papers. Their accuracy is similar as ours (3.43 and 3.47, respectively)

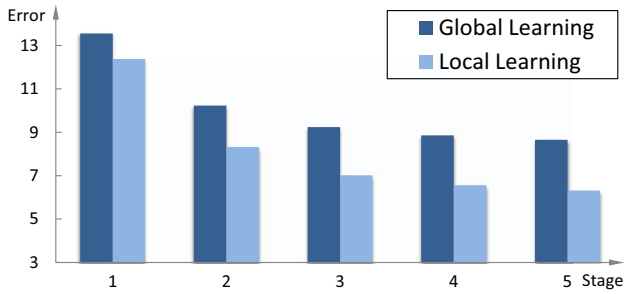


Figure 5. Comparison between local learning and global learning.

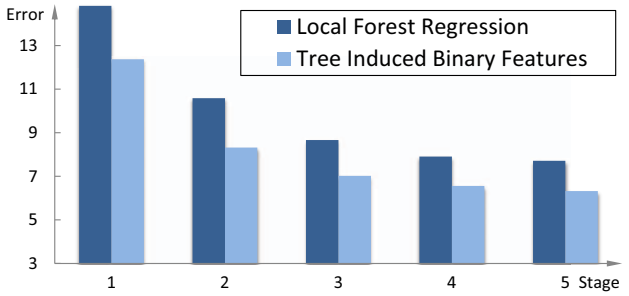


Figure 6. Comparison between tree induced binary features and local forest regression.

a small number of non-zero entries in its high dimensional features, the shape update is performed only a few times by efficient look up table and vector addition, instead of matrix multiplication in the global linear regression. The surprisingly high performance makes our approach especially attractive for applications with limited computational power. For example, our method runs in about 300 FPS on a mobile. This opens up new opportunities for online face applications on mobile phone.

### 4.2. Validation of proposed approach

We verify the effectiveness of the two key components of our approach, *local learning* and *binary features*, by comparing them with baseline methods that only differ in those aspects but remain exactly the same in all others. We use the 300-W dataset and LBF settings.

**Local learning vs. global learning.** In the baseline method, the difference is that, during the learning of local binary features, the pixels are indexed over the global shape, in the same way as [5], instead of only in a local region around the local landmark as in the proposed approach. Regression is performed on the entire shape instead of only the local landmark. All other parameters are the same to ensure the same training effort. We call this baseline *global learning*. Figure 5 shows that the proposed *local learning* is significantly better (25% error reduction) and verifies that it is capable of finding much better features.

**Tree induced binary features vs. local forest regression.** In the baseline method, we do not use the locally learned high dimensional binary features for global regression. Instead, we directly use the local random forest's regression output (a 2D offset vector) of each landmark as features to learn a global regression in the same way. Note that the learning process of the local trees is also exactly the same. Figure 6 shows that high dimensional binary features clearly outperform the simple raw output from local regression as features, because the former faithfully retains the full information of local learning.

## 5. Conclusion

In this work, we have presented a novel approach to learning local binary features for highly accurate and extremely fast face alignment. The shape regression framework regularized by locality principle is also promising for use in other relevant areas such as anatomic structure segmentation and human pose estimation. Furthermore, it is worth exploring the refitting strategy in other scenarios where regression trees are applied.

## References


[1] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011. 2, 5, 6

[2] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 3

[3] X. P. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. 2013. 1, 2, 3, 5, 6

[4] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *SIGGRAPH*, 32(4):41, 2013. 1, 2

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012. 1, 2, 3, 4, 5, 6

[6] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010. 2

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2001. 2

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 1995. 2

[9] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference (BMVC)*, 2006. 2

[10] D. Cristinacce and T. F. Cootes. Boosted regression active shape models. In *British Machine Vision Conference (BMVC)*, 2007. 2

[11] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012. 1, 2

[12] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010. 1, 2

[13] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 2008. 3

[14] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23:1080–1093, 2005. 2

[15] R. Gross, I. Matthews, and S. Baker. Active appearance models with occlusion. *Image and Vision Computing*, 2006. 2

[16] M. Kobetski and J. Sullivan. Discriminative tree-based feature mapping. In *British Machine Vision Conference (BMVC)*, 2013. 3

[17] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *12th European Conference on Computer Vision (ECCV)*. 2012. 5, 6

[18] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search. In *10th European Conference on Computer Vision (ECCV)*. 2008. 2

[19] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 2004. 2

[20] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *Second international conference on audio and video-based biometric person authentication*. Citeseer, 1999. 5

[21] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *10th European Conference on Computer Vision (ECCV)*. 2008. 6

[22] F. Moosmann, B. Triggs, F. Jurie, et al. Fast discriminative visual codebooks using randomized clustering forests. *Advances in Neural Information Processing Systems*, 2007. 2

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR Workshops*, 2013. 5

[24] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91:200–215, 2011. 2

[25] P. Sauer, T. F. Cootes, and C. J. Taylor. Accurate regression procedures for active appearance models. In *British Machine Vision Conference (BMVC)*, 2011. 2

[26] B. M. Smith and L. Zhang. Joint face alignment with non-parametric shape models. In *12th European Conference on Computer Vision (ECCV)*. 2012. 2

[27] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013. 1

[28] P. A. Tresadern, P. Sauer, and T. F. Cootes. Additive update predictors in active appearance models. In *British Machine Vision Conference (BMVC)*. Citeseer, 2010. 2

[29] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010. 1, 2

[30] C. Vens and F. Costa. Random forest based feature induction. In *11th IEEE International Conference on Data Mining (ICDM)*, 2011. 3

[31] Y. Wang, S. Lucey, and J. F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*. IEEE, 2008. 2

[32] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013. 1, 2, 5, 6

[33] W. Zhang, X. Wang, and X. Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011. 2

[34] F. Zhou, J. Brandt, and Z. Lin. Exemplar-based graph matching for robust facial landmark localization. In *IEEE 14th International Conference on Computer Vision (ICCV)*, 2013. 2, 6

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012. 2, 5

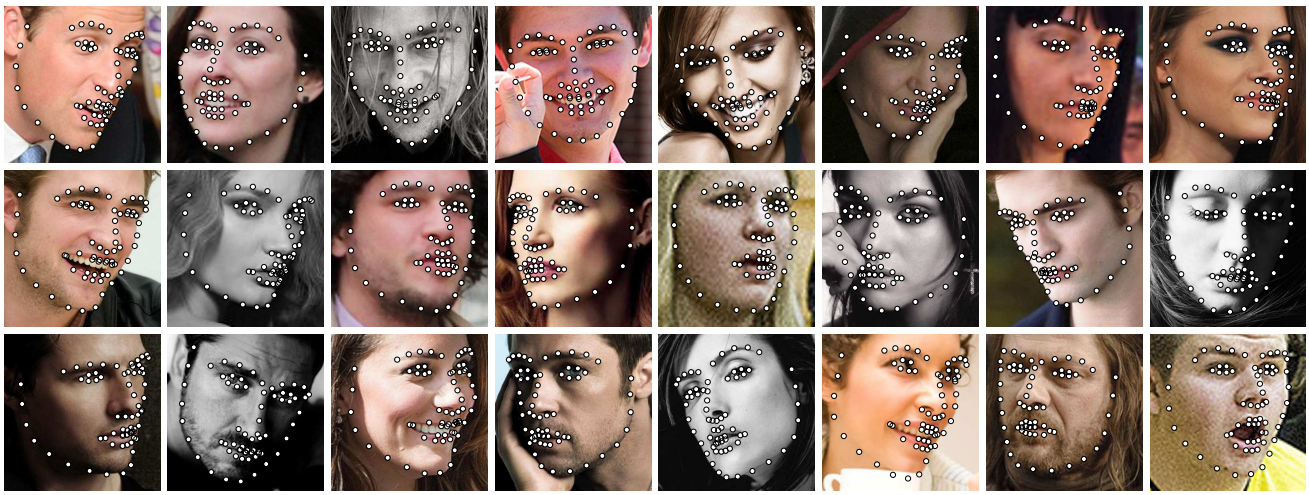

Figure 7. Example results from the Challenging Subset of the 300-W dataset.

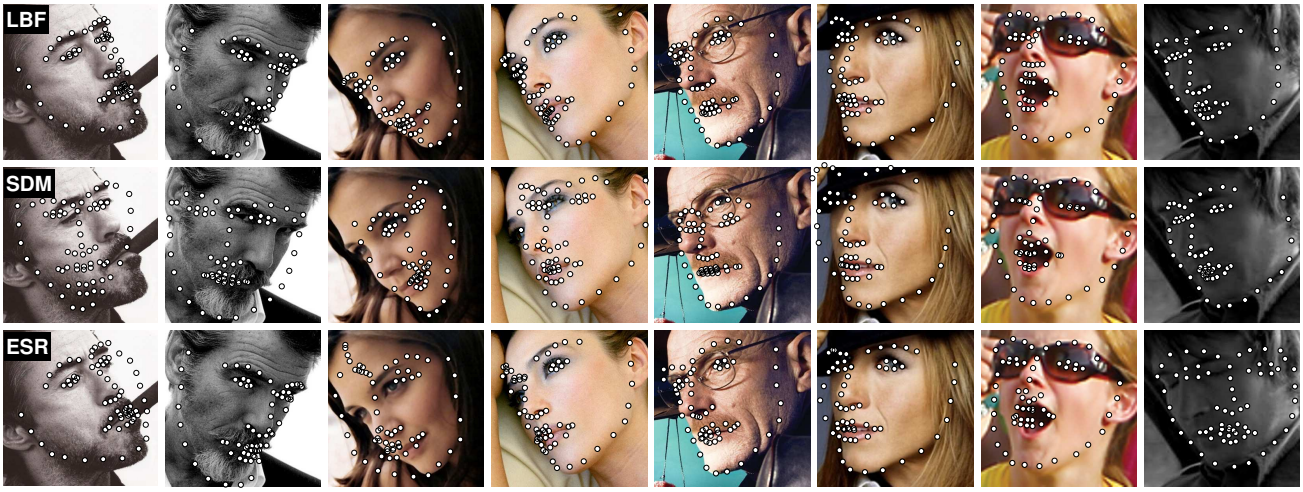


Figure 8. Example images from the Challenging Subset of 300-W dataset where our method outperforms ESR and SDM. These cases are extremely difficult due to the mixing of large head poses, extreme lighting, and partial occlusions.

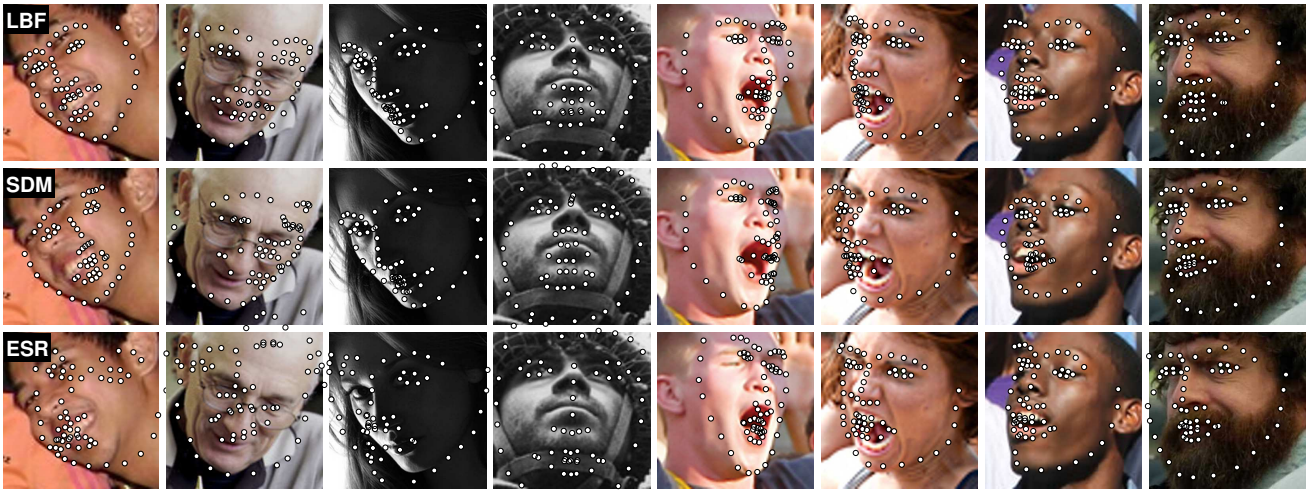


Figure 9. Some failure cases from the Challenging Subset of 300-W dataset.