Face Alignment by Coarse-to-Fine Shape Searching

Shizhan Zhu[1,2] Cheng Li[2] Chen Change Loy[1,3] Xiaoou Tang[1,3]
[1]Department of Information Engineering, The Chinese University of Hong Kong
[2]SenseTime Group
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
zs014@ie.cuhk.edu.hk, chengli@sensetime.com, ccloy@ie.cuhk.edu.hk, xtang@ie.cuhk.edu.hk

Abstract

We present a novel face alignment framework based on coarse-to-fine shape searching. Unlike the conventional cascaded regression approaches that start with an initial shape and refine the shape in a cascaded manner, our approach begins with a coarse search over a shape space that contains diverse shapes, and employs the coarse solution to constrain subsequent finer search of shapes. The unique stage-by-stage progressive and adaptive search i) prevents the final solution from being trapped in local optima due to poor initialisation, a common problem encountered by cascaded regression approaches; and ii) improves the robustness in coping with large pose variations. The framework demonstrates real-time performance and state-of-the-art results on various benchmarks including the challenging 300-W dataset.

1. Introduction

Face alignment aims at locating facial key points automatically. It is essential to many facial analysis tasks, *e.g*. face verification and recognition [11], expression recognition [2], or facial attributes analysis [16]. Among the many different approaches for face alignment, cascaded pose regression [8, 10, 29, 37] has emerged as one of the most popular and state-of-the-art methods. The algorithm typically starts from an initial shape, *e.g*. mean shape of training samples, and refines the shape through sequentially trained regressors.

In this study, we re-consider the face alignment problem from a different view point by taking a coarse-to-fine shape searching approach (Fig. 1(a)). The algorithm begins with a coarse searching in a shape space that encompasses a large number of candidate shapes. The coarse searching stage identifies a sub-region within the shape space for further searching in subsequent finer stages and simultaneously discards unpromising shape space sub-regions. Subsequent finer stages progressively and adaptively shrink the plausible region and converge the space to a small region where the final shape is estimated. In practice, only three stages are required.

In comparison to the conventional cascaded regression approaches, the coarse-to-fine framework is attractive in two aspects:

1) *Initialisation independent:* A widely acknowledged shortcoming of cascaded regression approach is its dependence on initialisation [32]. In particular, if the initialised shape is far from the target shape, it is unlikely that the discrepancy will be completely rectified by subsequent iterations in the cascade. As a consequence, the final solution may be trapped in local optima (Fig. 1(c)). Existing methods often circumvent this problem by adopting some heuristic assumptions or strategies (see Sec. 2 for details), which mitigate the problem to certain extent, but do not fully resolve the issue. The proposed coarse-to-fine framework relaxes the need of shape initialisation. It starts its first stage by exploring the whole shape space, without locking itself on a specific single initialisation point. This frees the alignment process from being affected by poor initialisation, leading to more robust face alignment.

2) *Robust to large pose variation:* The early stages in the coarse-to-fine search is formulated to simultaneously accommodate and consider diverse pose variations, *e.g*. with different degrees of head pose, and face contours. The search then progressively focus the processing on dedicated shape sub-region to estimate the best shape. Experimental results show that this searching mechanism is more robust in coping with large pose variations in comparison to the cascaded regression approach.

Since searching through shape space is challenging w.r.t. speed issue, we propose a hybrid features setting to achieve practical speed. Owing to the unique error tolerance in the coarse-to-fine searching mechanism, our framework is capable of exploiting the advantages and characteristics of different features. For instance, we have the flexibility to employ less accurate but computationally efficient feature, *e.g*. BRIEF [9] at the early stages, and use more accurate but

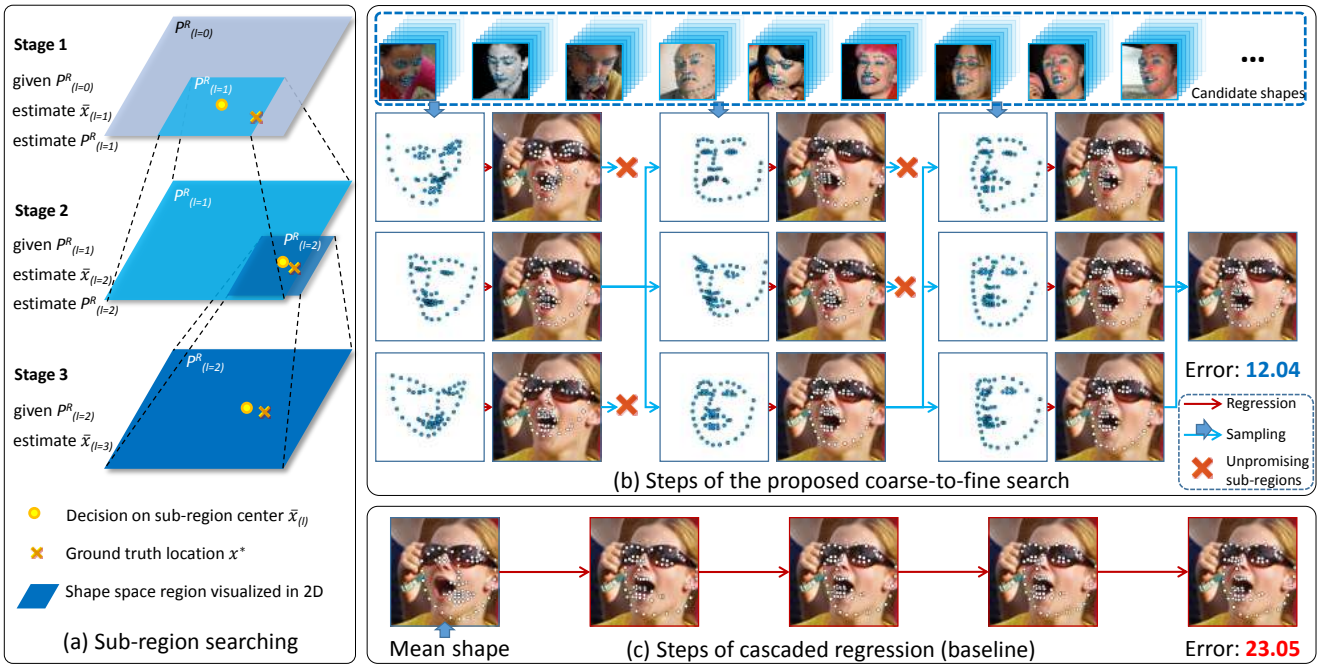


Figure 1. (a) A diagram that illustrates the coarse-to-fine shape searching method for estimating the target shape. (b) to (c) Comparison of the steps between proposed coarse-to-fine search and cascaded regression. Landmarks on nose and mouth are trapped in local optima in cascaded regression due to poor initialisation, and latter cascaded iterations seldom contribute much to rectifying the shape. The proposed method overcomes these problems through coarse-to-fine shape searching.

relatively slow feature, *e.g.* SIFT [23], at later stage. Such a setting allows the proposed framework to achieve improved computational efficiency, whilst it is still capable of maintaining high accuracy rate without using accurate features in all stages. Our MATLAB implementation achieves 25 fps real-time performance on a single core i5-4590. It is worth pointing out that impressive alignment speed (more than 1000 fps even for 194 landmarks) has been achieved by Ren *et al.* [29] and Kazemi *et al.* [20]. Though it is beyond the scope of this work to explore learning-based shape indexed feature, we believe the proposed shape searching framework could benefit from such high-speed feature.

Experimental results demonstrate that the coarse-to-fine shape searching framework is a compelling alternative to the popular cascaded regression approaches. Our method outperforms existing methods in various benchmark datasets including the challenging 300-W dataset [30]. Our code is available in project page `mmlab.ie.cuhk.edu.hk/projects/CFSS.html`.

2. Related work

A number of methods have been proposed for face alignment, including the classic active appearance model [12, 22, 24] and constrained local model [13, 35, 31, 15].

Face alignment by cascaded regression: There are a few successful methods that adopt the concept of cascaded pose regression [17]. Supervised descent method (SDM) [37] is proposed to solve nonlinear least squares optimisation problem. The non-linear SIFT [23] feature and linear regressors are applied. Feature learning based method, *e.g.* Cao *et al.* [10] and Burgos-Artizzu *et al.* [8], regress selected discriminative pixel-difference features with random ferns [27]. Ren *et al.* [29] learns the local binary features with random forest [6], achieving very fast performance.

All the aforementioned methods assume the initial shape is provided in some forms, typically a mean shape [37, 29]. Mean shape is used with the assumption that the test samples are distributed close to the mean pose of the training samples. This assumption does not always hold especially for faces with large pose variations. Cao *et al.* [10] propose to run the algorithm several times using different initialisations and take as final output the median of all predictions. Burgos-Artizzu *et al.* [8] improve the strategy by a smart restart method but it requires cross-validation to determine a threshold and the number of runs. In general, these strategies mitigate the problem to some extents, but still do not fully eliminate the dependence on shape initialisation. Zhang *et al.* [38] propose to obtain initialisation by predicting a rough estimation from global image patch, still followed by sequentially trained auto-encoder regression networks. Our method instead solves the initialisation problem via optimising shape sub-region. We will show in Sec. 4 that our proposed searching method is robust to large pose variation and outperforms previous methods.

Coarse-to-fine methods: The coarse-to-fine approach has been widely used to address various image processing and computer vision problems such as face detection [18], shape detection [1] and optical flow [7]. Some existing face alignment methods also adopt a coarse-to-fine approach but with a significantly different notion than our shape searching framework. Sun *et al*. [33] first have a coarse estimation of landmark locations and apply cascaded deep models to refine the position of landmarks of each facial part. Zhang *et al*. [38] define coarse-to-fine as applying cascaded of auto-encoder networks on images with increasing resolution.

3. Coarse-to-fine shape searching

Conventional cascaded regression methods refine a shape via sequentially regressing local appearance patterns indexed by the current estimated shape. In particular,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + r_k(\phi(I; \mathbf{x}_k)), \tag{1}$$

where the $2n$ dimensional shape vector $\mathbf{x}_k$ represents the current estimate of (x, y) coordinates of the n landmarks after the k^{th} iteration. The local appearance patterns indexed by the shape $\mathbf{x}$ on the face image I is denoted as $\phi(I; \mathbf{x})$, and r_k is the k^{th} learned regressor. For simplicity we always omit 'I' in Eq. 1.

The estimation by cascaded regression can be easily trapped in local optima given a poor shape initialisation since the method refines a shape by optimising a single shape vector $\mathbf{x}$ (Fig. 1(c)). In our approach, we overcome the problem through a coarse-to-fine shape searching within a shape space (Fig. 1(a) and (b)).

3.1. Overview of coarse-to-fine shape searching

Formally, we form a $2n$ dimensional shape space. We denote N candidate shapes in the space as $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N\}$ ($N \gg 2n$). The candidate shapes in $\mathcal{S}$ are obtained from training set pre-processed by Procrustes analysis [19]. $\mathcal{S}$ is fixed throughout the whole shape searching process.

Given a face image, face alignment is performed through $l = 1, \ldots, L$ stages of shape searching, as depicted in Fig. 1(a). In each l^{th} stage, we aim to find a finer shape sub-region, which is represented by $\left(\bar{\mathbf{x}}_{(l)}, P^R_{(l)}\right)$, where $\bar{\mathbf{x}}_{(l)}$ denotes the center of the estimated shape sub-region, and $P^R_{(l)}$ represents the probability distribution that defines the scope of estimated sub-region around the center. When the searching progresses through stages, *e.g*. from Stage 1 to 2, the algorithm adaptively determines the values of $\bar{\mathbf{x}}$ and P^R, leading to a finer shape sub-region for the next searching stage, with closer estimate to the target shape. The process continues until convergence and the center of the last finest sub-region is the final shape estimation.

In each stage, we first determine the sub-region center $\bar{\mathbf{x}}$ based on the given sub-region for this stage, and then estimate the finer sub-region used for further searching. A larger/coarser region is expected at earlier stages, whilst a smaller/finer region is expected at latter stages. In the first searching stage, the given 'sub-region' $P^R_{(l=0)}$ is set to be a uniform distribution over all candidate shapes, *i.e*. the searching region is over the full set of $\mathcal{S}$. In the subsequent stages, the given sub-region is the estimated $P^R_{(l-1)}$ from the preceding stage.

Algorithm 1 Training of coarse-to-fine shape searching

1: **procedure** TRAINING(Shapes $\mathcal{S}$, Training set $\{I^i; \mathbf{x}^{i*}\}_{i=1}^N$)
2: Set $P^R_{(0)}$ to be uniform distribution over $\mathcal{S}$
3: **for** $l = 1, 2, \ldots, L$ **do**
4: Sample candidate shapes $\mathbf{x}_0^{ij}$ according to $P^R_{(l-1)}$
5: Learn K_l regressors $\{r_k\}_{k=1}^{K_l}$ with $\{\mathbf{x}_0^{ij}, \mathbf{x}^{i*}\}_{i=1,j=1}^{N,\ N_l}$
6: Get regressed shapes $\mathbf{x}_{K_l}^{ij}$ based on the K_l regressors
7: Set initial weight to be equal: $\mathbf{w}^i(0) = \mathbf{e}/N_l$
8: Construct G^i and edge weight according to Eq. 4
9: **for** $t = 0, 1, \ldots, T-1$ **do**
10: Update $\mathbf{w}^i(t+1)$ according to Eq. 6
11: **end for**
12: Compute sub-region center $\bar{\mathbf{x}}^i_{(l)}$ via Eq. 3
13: **if** $l < L$ **then**
14: Learn distribution with $\{\bar{\mathbf{x}}^i_{(l)}, \mathbf{x}^{i*}\}_{i=1}^N$
15: Set probabilistic distribution $P^R_{(l)}$ via Eq. 7
16: **end if**
17: **end for**
18: **end procedure**

As an overview of the whole approach, we list the major training steps in Algorithm 1, and introduce the learning method in Sec. 3.2 and Sec. 3.3. Testing procedure of the approach is similar excluding the learning steps. More precisely, the learning steps involve learning the regressors in each stage (Eq. 2 and Step 5 in Algorithm 1) and parameters for estimating probabilistic distribution (Eq. 8 and 10, Step 14 in Algorithm 1).

3.2. Learn to estimate sub-region center $\bar{\mathbf{x}}$ given P^R

To learn to compute the sub-region center $\bar{\mathbf{x}}_{(l)}$ for the l^{th} searching stage, three specific steps are conducted:

Step-1: In contrast to cascaded regression that employs a single initial shape (typically the mean shape) for regression, we explore a larger area in the shape space guided by the probabilistic distribution $P^R_{(l-1)}$. In particular, for each training sample, we randomly draw N_l initial shapes from $\mathcal{S}$ based on $P^R_{(l-1)}$. We denote the N_l initial shapes of the i^{th} training sample as $\mathbf{x}_0^{ij}$, with $i = 1 \ldots N$ representing the index of training sample, and $j = 1 \ldots N_l$ denoting the index of the randomly drawn shapes.

Step-2: This step aims to regress each initial shape $\mathbf{x}_0^{ij}$ to a shape closer to the ground truth shape $\mathbf{x}^{i*}$. Specifically, we learn K_l regressors in a sequential manner with iteration

$k = 0, \ldots, K_l - 1$, *i.e.*

$$r_k = \underset{r}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{N_l} \|\mathbf{x}^{i*} - \mathbf{x}_k^{ij} - r(\phi(\mathbf{x}_k^{ij}))\|_2^2 + \Phi(r),$$
$$\mathbf{x}_{k+1}^{ij} = \mathbf{x}_k^{ij} + r_k(\phi(\mathbf{x}_k^{ij})) \qquad k = 0, \ldots, K_l - 1 \tag{2}$$

where $\Phi(r)$ denotes the ℓ_2 regularisation term for each parameter in model r. It is worth pointing out that K_l is smaller than the number of regression iterations typically needed in cascaded regression. This is because *i)* due to the error tolerance of coarse-to-fine searching, regressed shapes for early stages need not be accurate, and *ii)* for later stages initial candidate shapes $\mathbf{x}_0^{ij}$ tend to be similar to the target shape, thus fewer iterations are needed for convergence.

Step-3: After we learn the regressors and obtain the set of regressed shapes, $\left\{\mathbf{x}_{K_l}^{ij}\right\}_{j=1}^{N_l}$, we wish to learn a weight vector $\mathbf{w}^i = (w^{i1}, \ldots w^{iN_l})^\top$ to linearly combine all the regressed shapes for collectively estimating the sub-region center $\bar{\mathbf{x}}_{(l)}^i$ for i-th training sample

$$\bar{\mathbf{x}}_{(l)}^i = \sum\nolimits_{j=1}^{N_l} w^{ij}\, \mathbf{x}_{K_l}^{ij}. \tag{3}$$

A straightforward method to obtain $\bar{\mathbf{x}}_{(l)}^i$ is to average all the regressed shapes by fixing $w^{ij} = 1/N_l$. However, this simple method is found susceptible to small quantity of erroneous regressed shapes caused by local optima. In order to suppress their influence in computing the sub-region center, we adopt the dominant set approach [28] for estimating $\mathbf{w}^i$. Intuitively, a high weight is assigned to regressed shapes that form a cohesive cluster, whilst a low weight is given to outliers. This amounts to finding a maximal clique in an undirected graph. Note that this step is purely unsupervised.

More precisely, we construct an undirected graph, $G^i = (V^i, E^i)$, where the vertices are the regressed shapes $V^i = \left\{\mathbf{x}_{K_l}^{ij}\right\}_{j=1}^{N_l}$, and each edge in the edge set E^i is weighted by affinity defined as

$$a_{pq} = \operatorname{sim}(\mathbf{x}_{K_l}^{ip}, \mathbf{x}_{K_l}^{iq}) = \begin{cases} \exp(-\beta\|\mathbf{x}_{K_l}^{ip} - \mathbf{x}_{K_l}^{iq}\|_2^2), & p \neq q \\ 0, & p = q \end{cases}. \tag{4}$$

Representing all the elements a_{pq} in a matrix forms an affinity matrix, $\mathbf{A}$. Note that we set the diagonal elements of $\mathbf{A}$ to zero to avoid self-loops. Following [28], we find the weight vector $\mathbf{w}^i$ by optimising the following problem,

$$\max_{\mathbf{w}^i} \ \mathbf{w}^{i\top}\mathbf{A}\mathbf{w}^i \quad \text{s.t.} \ \ \mathbf{w}^i \in \Delta_{N_l}. \tag{5}$$

We denote the simplex as $\Delta_n = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} \geq 0, \mathbf{e}^\top\mathbf{x} = 1\}$, where $\mathbf{e} = (1, 1, ..., 1)^\top$. An efficient way to optimise Eq. 5 is by using continuous optimisation technique known as replicator dynamics [28, 36]

$$\mathbf{w}^i(t+1) = \frac{\mathbf{w}^i(t) \circ (\mathbf{A}\mathbf{w}^i(t))}{\mathbf{w}^i(t)^\top \mathbf{A}\mathbf{w}^i(t)}, \tag{6}$$

where $t = 0, 1, \ldots, T-1$, and symbol '$\circ$' denotes elementary multiplication. Intuitively, in each weighting iteration t, each vertex votes all its weight to other vertex, w.r.t. the affinity between the two vertices. After optimising Eq. 6 for T iterations, we obtain $\mathbf{w}^i(t = T)$ and plug the weight vector into Eq. 3 for estimating the sub-region center.

3.3. Learn to estimate probabilistic distribution P^R given $\bar{\mathbf{x}}$

We then learn to estimate the probabilistic distribution $P_{(l)}^R$ based on the estimated sub-region center $\bar{\mathbf{x}}_{(l)}$. We aim to determine the probabilistic distribution, $P_{(l)}^R(\mathbf{s}|\bar{\mathbf{x}}_{(l)}) = P(\mathbf{s} - \bar{\mathbf{x}}_{(l)}|\phi(\bar{\mathbf{x}}_{(l)}))$, where $\mathbf{s} \in \mathcal{S}$ and $\sum_{\mathbf{s}\in\mathcal{S}} P_{(l)}^R(\mathbf{s}|\bar{\mathbf{x}}_{(l)}) = 1$. For clarity, we drop the subscripts (l) from $\bar{\mathbf{x}}_{(l)}$ and $P_{(l)}^R$. We model the probabilistic distribution $P_{(l)}^R$ as

$$P(\mathbf{s} - \bar{\mathbf{x}}|\phi(\bar{\mathbf{x}})) = \frac{P(\mathbf{s} - \bar{\mathbf{x}})P(\phi(\bar{\mathbf{x}})|\mathbf{s} - \bar{\mathbf{x}})}{\sum_{y\in\mathcal{S}} P(\mathbf{y} - \bar{\mathbf{x}})P(\phi(\bar{\mathbf{x}})|\mathbf{y} - \bar{\mathbf{x}})}. \tag{7}$$

The denominator is a normalising factor. Thus, when estimating the posterior probability of each shape $\mathbf{s}$ in $\mathcal{S}$ we focus on the two factors $P(\mathbf{s} - \bar{\mathbf{x}})$, and $P(\phi(\bar{\mathbf{x}})|\mathbf{s} - \bar{\mathbf{x}})$.

The factor $P(\mathbf{s} - \bar{\mathbf{x}})$, referred as shape adjustment prior, is modelled as

$$P(\mathbf{s} - \bar{\mathbf{x}}) \propto \exp(-\frac{1}{2}(\mathbf{s} - \bar{\mathbf{x}})^\top \Sigma^{-1}(\mathbf{s} - \bar{\mathbf{x}})). \tag{8}$$

The covariance matrix is learned by $\{\bar{\mathbf{x}}^i, \mathbf{x}^{i*}\}_{i=1}^N$ pairs on training data, where $\mathbf{x}^*$ denotes the ground truth shape[1]. In practice, Σ is restricted to be diagonal and we decorrelate the shape residual by principle component analysis. This shape adjustment prior aims to approximately delineate the searching scope near $\bar{\mathbf{x}}$, and typically the distribution is more concentrated for later searching stages.

The other factor $P(\phi(\bar{\mathbf{x}})|\mathbf{s} - \bar{\mathbf{x}})$ is referred as feature similarity likelihood. Following [5], we divide this factor into different facial parts,

$$P(\phi(\bar{\mathbf{x}})|\mathbf{s} - \bar{\mathbf{x}}) = \prod_j P(\phi(\bar{\mathbf{x}}^{(j)})|\mathbf{s}^{(j)} - \bar{\mathbf{x}}^{(j)}), \tag{9}$$

where j represents the facial part index. The probabilistic independence comes from our conditioning on the given

[1] We assume $\mathbb{E}(\mathbf{x}^* - \bar{\mathbf{x}}) = \mathbf{0}$.

exemplar candidate shapes $\mathbf{s}$ and $\bar{\mathbf{x}}$, and throughout our approach, all intermediate estimated poses are strictly shapes. Again by applying Baye's rule, we can rewrite Eq. 9 into

$$P(\phi(\bar{\mathbf{x}})|\mathbf{s}-\bar{\mathbf{x}}) = \frac{\prod_j P(\phi(\bar{\mathbf{x}}^{(j)}))}{\prod_j P(\mathbf{s}^{(j)})} \prod_j P(\mathbf{s}^{(j)} - \bar{\mathbf{x}}^{(j)}|\phi(\bar{\mathbf{x}}^{(j)})) \propto \prod_j P(\mathbf{s}^{(j)} - \bar{\mathbf{x}}^{(j)}|\phi(\bar{\mathbf{x}}^{(j)})), \tag{10}$$

which could be learned via discriminative mapping for each facial part. This feature similarity likelihood aims to guide shapes moving towards more plausible shape region, by separately considering local appearance from each facial part.

By combining the two factors, we form the probabilistic estimate for the shape space and could sample candidate shapes for next stage. Such probabilistic sampling enables us to estimate current shape error and refine current estimate via local appearance, while at the same time the shape constraints are still strictly encoded.

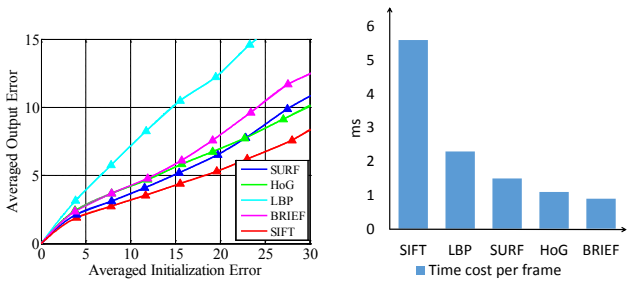
3.4. Shape searching with hybrid features

In the conventional cascaded regression framework, one often selects a particular features for regression, *e.g*. SIFT in [37]. The selection of features involves the tradeoff between alignment accuracy and speed. It can be observed from Fig. 2 that different features (*e.g*. HoG [14], SIFT [23], LBP [26], SURF [4], BRIEF [9]) exhibit different characteristics in accuracy and speed. It is clear that if one adheres to the SIFT feature throughout the whole regression procedure, the best performance in our method can be obtained. However, the run time efficiency is much lower than that of the BRIEF feature.

Our coarse-to-fine shape searching framework is capable of exploiting different types of features at different stages, taking advantages of their specific characteristics, *i.e*. speed and accuracy. Based on the feature characteristics observed in Fig. 2, we can operate the coarse-to-fine framework in two different feature settings through switching features in different searching stages:

- *CFSS* - The SIFT feature is used in all stages to obtain the best accuracy in our approach.
- *CFSS-Practical* - Since our framework only seeks for a coarse shape sub-region in the early stages, thus relatively weaker features with much faster speed (*e.g*. BRIEF) would be a better choice for early stage, and SIFT is only used in the last stage for refinement. In our 3-stage implementation, we use the BRIEF feature in the first two stages, and SIFT in the last stage.

In the experiments we will demonstrate that the CFSS-Practical performs competitively to the CFSS, despite using the less accurate BRIEF for the first two stages. The



(a) Regression curves for features. (b) Speed for features.

Figure 2. We evaluate each feature's accuracy and speed using a validation set extracted from the training set. (a) We simulate different initial conditions with different initialisation errors to evaluate the averaged output error of cascaded regression. We ensure that the result has converged for each initialisation condition. (b) Comparison of speed of various features measured under the same quantity of regression tasks.

CFSS enjoys such feature switching flexibility thanks to the error tolerance of the searching framework. In particular, CFSS allows for less accurate shape sub-region in the earlier searching stages, since subsequent stages can rapidly converge to the desired shape space location for target shape estimation.

3.5. Time complexity analysis

The most time consuming module is feature extraction, which directly influences the time complexity. We assume the complexity for feature extraction is $O(F)$. The complexity of CFSS is thus $O(F(L-1+\sum_{l=1}^{L} N_l K_l))$. By applying the hybrid feature setting, the complexity reduces to $O(FN_L K_L)$, since only the last searching stage utilises the more accurate feature, and the time spent on the fast feature contributes only a small fraction to the whole processing time. As is shown in Sec. 4.2, the efficiency of the searching approach is in the same order of magnitude compared with cascaded regression method, but with much more accurate prediction.

3.6. Implementation details

In practice, we use $L = 3$ searching stages in the CFSS. Increasing the number of stages only leads to marginal improvement. The number of regressors, K_l, and initial shapes N_l, are set without optimisation. In general, we found setting $K_l = 3$, and $N_l = 15$ works well for CFSS. Only marginal improvement is obtained with larger number of K_l and N_l. For CFSS-Practical, we gain further run time efficiency by reducing the regression iterations K_l, and decreasing N_l without sacrificing too much accuracy. We choose K_l in the range of 1 to 2, and N_l in the range of 5 to 10. We observe that the alignment accuracy is not sensitive to these parameters. We set $T = 10$ in Eq. 6. β (in

Eq. 4) is determined through cross-validation. Linear model is applied as the regressor in Eq. 2.

4. Experiments

Datasets Evaluations are performed on three widely used benchmark datasets. These datasets are challenging due to images with large head pose, occlusions, and illumination variations.

300-W dataset [30]: This dataset standardises various alignment databases, including AFW [41], LFPW [5], HELEN [21] and XM2VTS [25] with 68-point annotation. In addition, it contains a challenging 135-image IBUG set. For fair comparison, we follow the same dataset configuration as in [29]. Specifically, we regard all the training samples from LFPW, HELEN and the whole AFW as the training set (3148 images in total), and perform testing on three parts: the test samples from LFPW and HELEN as the common subset, the 135-image IBUG as the challenging subset, and the union of them as the full set (689 images in total).

HELEN dataset [21]: it contains 2000 training and 330 test images. We conduct evaluations on 194 points (provided by [21]) and 68 / 49 points (provided by [30]).

LFPW dataset [5]: it originally contains 1100 training and 300 test images. However due to some invalid URLs, we only employ the 811 training and 224 test images provided by [30]. We perform evaluations on the 68 and 49 points settings. For the conventional 29 points setting, our result is comparable to that reported in [29], which has almost reached human performance and become saturated.

Since the 300-W dataset [30] provides prescribed face bounding boxes for all the data mentioned above, we do not use proposed boxes from external face detectors and thus no faces are missed during testing.

Evaluation We evaluate the alignment accuracy for each sample using the standard landmarks mean error normalised by the inter-pupil distance. For simplicity we omit the '%' symbol. The overall accuracy is reported based either on the averaged errors or cumulative errors distribution (CED) curve to cater for different evaluation schemes in the literature.

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

We compare the CFSS and CFSS-Practical with other state-of-the-art methods. We trained our model only using the data from the specific training set without external sources. We do not compare with deep learning based methods [33, 39] since they mainly detect 5 facial landmarks and the deep models are pre-trained with enormous quantity of external data sources. The results are thus not comparable.

Averaged error comparison We summarise the comparative results in Table 1. It can be observed that both settings of our proposed method outperform all previous methods on these datasets. It is worth noting that we only apply hand-designed features in the searching framework. Even so, our method still outperforms existing feature-learning-based approaches [10, 8, 29], especially on the challenging subset of 300-W dataset (over **16%** of error reduction in comparison to the state-of-the-art method [29]). The results suggest the robustness of the searching framework over the conventional cascaded regression approaches. We believe further improvement can be gained by extending our framework to a feature-learning-based approach.

Cumulative error distribution comparison To compare the results with literatures reporting CED performance, we plot the CED curves for various methods in Fig. 3. Again, the proposed CFSS achieves the best performance, whilst CFSS-Practical is competitive to the best setting, thanks to the robustness and error tolerance in the early searching stages. We provide examples of alignment results of our method and [37, 29, 38] in Fig. 4.

4.2. Comparison with cascaded regression on different initialisations

To highlight the advantages of the proposed coarse-to-fine shape searching method over the conventional cascaded regression approach, we compare the proposed CFSS and CFSS-Practical with SDM [37]. It is a representative cascaded regression method, and it applies similar type of feature and regressor as applied in our framework. For fair comparison, we compare the cascaded regression method with the two feature settings applied, *i.e.* 1) SIFT throughout (*i.e.* the best setting of our method); 2) Hybrid features with SIFT for the last iteration and BRIEF for the others (*i.e.* the practical setting of our method). Moreover, the results of cascaded regression method are reported based on 4 different widely used initialisation methods. We ensure result is converged for cascaded regression. The full set of 300-W dataset is used for evaluation.

Results are shown in Table 2. It is observed that both the proposed CFSS and CFSS-Practical outperform the cascaded regression on all initialisation schemes, including the 'good initialisation', where the initial shape for each cascaded regression is pre-estimated using a cascaded deep model [33]. It is worth noting that the performance of CFSS-Practical is competitive to CFSS (5.99 vs. 5.76 averaged error after stage 3 searching). In contrast, the cascaded regression method performs poorly on hybrid features, suggesting that it cannot benefit from using different feature types.

Since we use the computationally cheap BRIEF features in the first two searching stages, the proposed CFSS-

LFPW Dataset			Helen Dataset				300-W Dataset (All 68 points)			
Method	68 -pts	49 -pts	Method	194 -pts	68 -pts	49 -pts	Method	Common Subset	Challenging Subset	Fullset
Zhu et. al [41]	8.29	7.78	Zhu et. al [41]	-	8.16	7.43	Zhu et. al [41]	8.22	18.33	10.20
DRMF [3]	6.57	-	DRMF [3]	-	6.70	-	DRMF [3]	6.65	19.79	9.22
			ESR [10]	5.70	-	-	ESR [10]	5.28	17.00	7.58
RCPR [8]	6.56	5.48	RCPR [8]	6.50	5.93	4.64	RCPR [8]	6.18	17.26	8.35
SDM [37]	5.67	4.47	SDM [37]	5.85	5.50	4.25	SDM [37]	5.57	15.40	7.50
							Smith et. al [32]	-	13.30	-
							Zhao et. al [40]	-	-	6.31
GN-DPM [34]	5.92	4.43	GN-DPM [34]	-	5.69	4.06	GN-DPM [34]	5.78	-	-
CFAN [38]	5.44	-	CFAN [38]	-	5.53	-	CFAN [38]	5.50	-	-
			ERT [20]	4.90	-	-	ERT [20]	-	-	6.40
			LBF [29]	5.41	-	-	LBF [29]	4.95	11.98	6.32
			LBF fast [29]	5.80	-	-	LBF fast [29]	5.38	15.50	7.37
CFSS	**4.87**	**3.78**	**CFSS**	**4.74**	**4.63**	**3.47**	**CFSS**	**4.73**	**9.98**	**5.76**
CFSS Practical	**4.90**	**3.80**	**CFSS Practical**	**4.84**	**4.72**	**3.50**	**CFSS Practical**	**4.79**	**10.92**	**5.99**

Table 1. Comparison of averaged errors with state-of-the-art methods. It is worth noting that our result on LFPW (29-pts) is comparable to [29], of which is almost saturated to human labelling as stated in [29]. For most methods, the results are obtained directly from the literatures or evaluated based on the released codes. For methods that jointly perform face detection and alignment, we only average their relative errors on true positive detected faces.

Features settings	Cascaded regression				Coarse-to-fine searching		
	Random initialisation	Random voting	Mean shape initialisation	Good initialisation	After Stage 1	After Stage 2	After Stage 3
(1) SIFT throughout	10.68	8.17	7.50	6.33	13.68	8.18	**5.76**
(2) Hybrid features	15.36	11.62	10.20	7.01	18.68	11.43	**5.99**

Table 2. Comparison of averaged error for cascaded regression and coarse-to-fine searching methods. Our CFSS and CFSS-Practical correspond to the feature settings (1) and (2), respectively. Errors after Stage 1 and 2 denote the mean error of candidate shapes sampled from the chosen sub-region, and Stage 3 denotes final estimate. 'Good initialisation' of cascaded regression is achieved by pre-estimating the initial shape using 5 landmark points obtained from a deep model [33]. It is worth pointing out that our method outperforms cascaded regression method even the latter adopts a 'good initialisation' scheme.

Practical achieves competitive speed performance with cascaded regression. Specifically, our MATLAB implementation achieves 40 ms per-frame on a single core i5-4590 CPU, compared to 28 ms for cascaded regression without random voting, and 124 ms with 5 random votes. Efficient learning-based feature has been proposed in [29] and [20]. We believe our framework could benefit from them by incorporating more efficient features in the future.

4.3. Further analyses

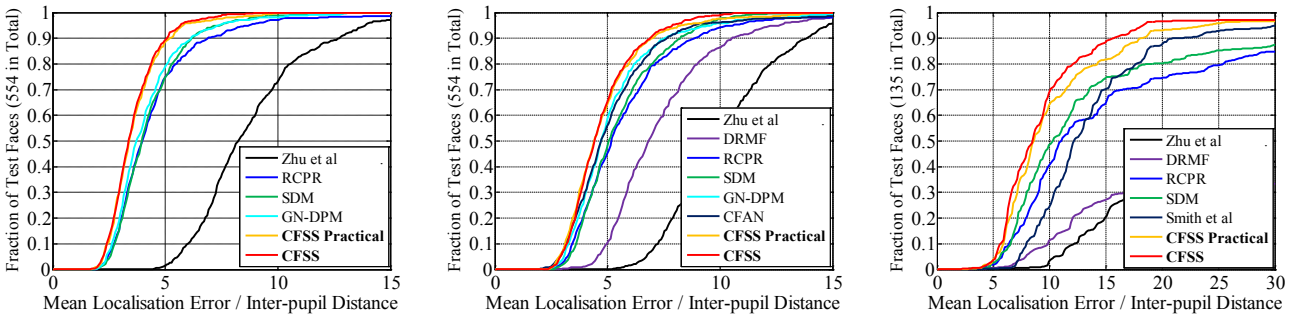
Probabilistic vs. random sub-region sampling Probabilistic sub-region sampling plays an important role in our framework in offering a probable scope of sub-region for selecting shapes from the shape space for later searching stages. If we simply select shapes randomly from a fixed range of neighbourhood of $\bar{\mathbf{x}}$, the averaged errors by CFSS-Practical would increase from 5.99 to 6.42.

Dominant set approach vs. mean weighting Dominant set approach is essential to filter out erroneous shape candidates, especially in the early stages. With dominant set approach, we achieve 5.99 averaged error on the 300-W full set by CFSS-Practical. However, if we replace the dominant set approach with mean weighting, the averaged error increases to 6.08. Errors are mainly observed in cases with large head pose.

5. Conclusion and future work

We have presented a novel face alignment method through coarse-to-fine shape searching. Superior error tolerance is achieved through probabilistic sub-region searching and dominant set approach for filtering out erroneous shape sub-regions. The framework is advantageous over the conventional cascaded approach in that i) it is initialisation independent and ii) it is robust to faces with large pose variation. We show that real-time performance can be achieved by using hybrid feature setting in the proposed method. We plan to incorporate learning-based feature in our framework in the future to further improve the accuracy and efficiency. [2]

[2]This work was done while Shizhan Zhu was an intern at Sensetime Group.



(a) CED for 49-pts common subset of 300-W. (b) CED for 68-pts common subset of 300-W. (c) CED for 68-pts challenging subset of 300-W.

Figure 3. Comparisons of cumulative errors distribution (CED) curves. The proposed method outperforms various state-of-the-art methods.

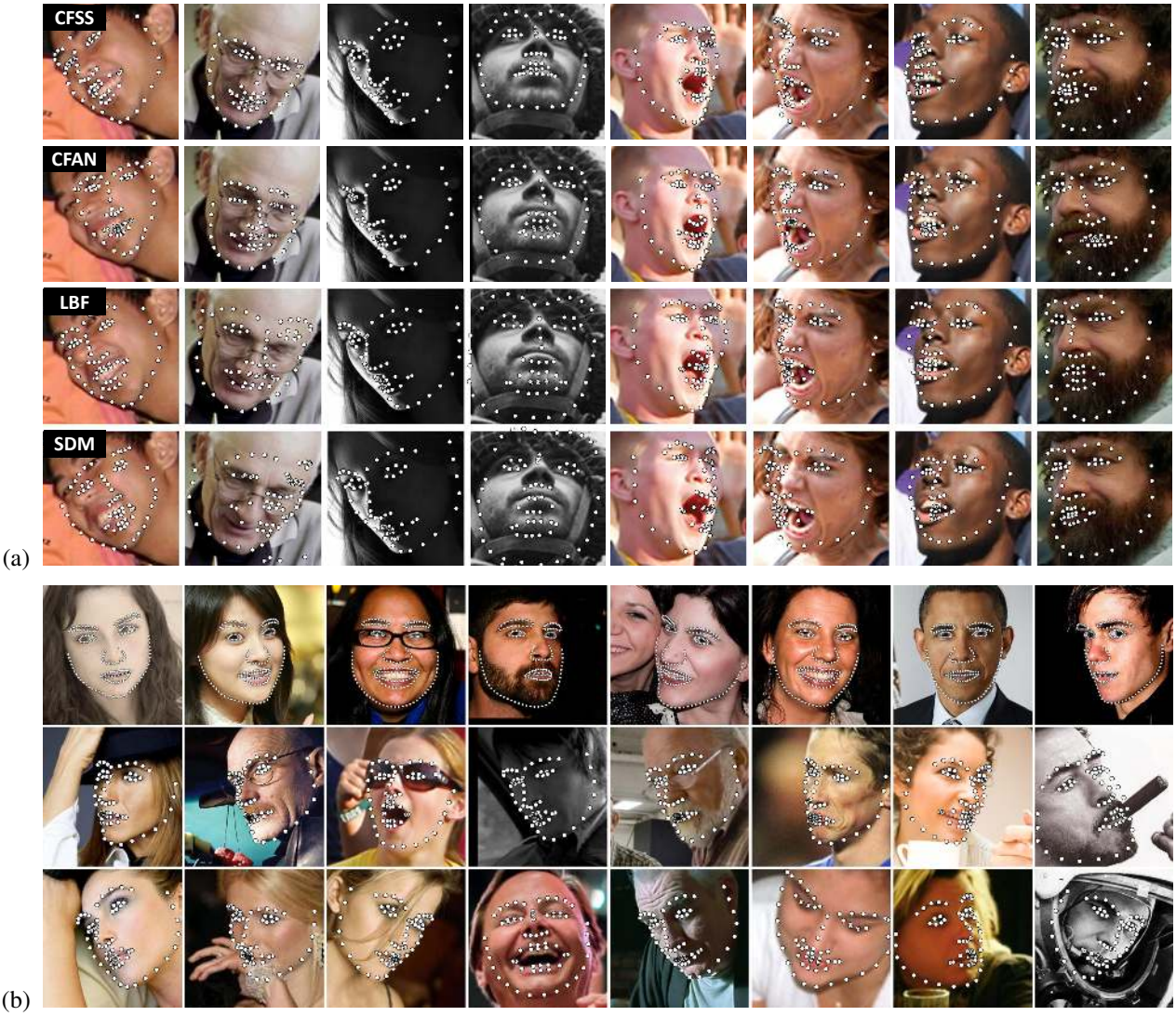


Figure 4. (a) Example images where the proposed CFSS outperforms CFAN [38], LBF [29], and SDM [37]. The images are challenging due to large head pose, severe occlusion, and extreme illumination. (b) More examples of CFSS: the images of the first row are from the Helen dataset and the last two rows are from the challenging subset of 300-W.

References

[1] Y. Amit, D. Geman, and X. Fan. A coarse-to-fine strategy for multiclass shape detection. *TPAMI*, 26(12):1606–1621, 2004. 3

[2] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, and P. E. Solomon. The painful face–pain expression recognition using active appearance models. *IVC*, 27(12):1788–1796, 2009. 1

[3] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, pages 3444–3451, 2013. 7

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, pages 404–417, 2006. 5

[5] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, pages 545–552, 2011. 4, 6

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 2

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004. 3

[8] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, pages 1513–1520, 2013. 1, 2, 6, 7

[9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *ECCV*, pages 778–792, 2010. 1, 5

[10] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *IJCV*, 107(2):177–190, 2014. 1, 2, 6, 7

[11] C. Chen, A. Dantcheva, and A. Ross. Automatic facial makeup detection with application in face recognition. In *ICB*, pages 1–8, 2013. 1

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *TPAMI*, 23(6):681–685, 2001. 2

[13] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, volume 2, page 6, 2006. 2

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005. 5

[15] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, pages 2578–2585. IEEE, 2012. 2

[16] A. Datta, R. Feris, and D. Vaquero. Hierarchical ranking of facial attributes. In *AFGR*, pages 36–42. IEEE, 2011. 1

[17] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, pages 1078–1085, 2010. 2

[18] F. Fleuret and D. Geman. Coarse-to-fine face detection. *IJCV*, 41(1-2):85–107, 2001. 3

[19] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 3

[20] V. Kazemi and S. Josephine. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014. 2, 7

[21] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, pages 679–692, 2012. 6

[22] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search. In *ECCV*, pages 72–85. Springer, 2008. 2

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2, 5

[24] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 2

[25] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *AVBPA*, volume 964, pages 965–966. Citeseer, 1999. 6

[26] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002. 5

[27] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *TPAMI*, 32(3):448–461, 2010. 2

[28] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *TPAMI*, 29(1):167–172, 2007. 4

[29] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, 2014. 1, 2, 6, 7, 8

[30] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCVW*, pages 397–403, 2013. 2, 6

[31] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011. 2

[32] B. M. Smith, J. Brandt, Z. Lin, and L. Zhang. Nonparametric context modeling of local appearance for pose-and expression-robust facial landmark localization. In *CVPR*, 2014. 1, 7

[33] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, pages 3476–3483, 2013. 3, 6, 7

[34] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *CVPR*, pages 1851–1858, 2014. 7

[35] Y. Wang, S. Lucey, and J. F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, pages 1–8. IEEE, 2008. 2

[36] J. W. Weibull. *Evolutionary game theory*. MIT press, 1997. 4

[37] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013. 1, 2, 5, 6, 7, 8

[38] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *ECCV*, pages 1–16. 2014. 2, 3, 6, 7, 8

[39] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, pages 94–108, 2014. 6

[40] X. Zhao, T.-K. Kim, and W. Luo. Unified face analysis by iterative multi-output random forests. In *CVPR*, pages 1765–1772, 2014. 7

[41] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012. 6, 7