Face Alignment in Full Pose Range: A 3D Total Solution

Xiangyu Zhu, Xiaoming Liu, *Member, IEEE,* Zhen Lei, *Senior Member, IEEE,* and Stan Z. Li, *Fellow, IEEE*

Abstract— Face alignment, which fits a face model to an image and extracts the semantic meanings of facial pixels, has been an important topic in the computer vision community. However, most algorithms are designed for faces in small to medium poses (yaw angle is smaller than $45°$), which lack the ability to align faces in large poses up to $90°$. The challenges are three-fold. Firstly, the commonly used landmark face model assumes that all the landmarks are visible and is therefore not suitable for large poses. Secondly, the face appearance varies more drastically across large poses, from the frontal view to the profile view. Thirdly, labelling landmarks in large poses is extremely challenging since the invisible landmarks have to be guessed. In this paper, we propose to tackle these three challenges in an new alignment framework termed 3D Dense Face Alignment (3DDFA), in which a dense 3D Morphable Model (3DMM) is fitted to the image via Cascaded Convolutional Neural Networks. We also utilize 3D information to synthesize face images in profile views to provide abundant samples for training. Experiments on the challenging AFLW database show that the proposed approach achieves significant improvements over the state-of-the-art methods.

Index Terms—Face Alignment, 3D Morphable Model, Convolutional Neural Network, Cascaded Regression

1 Introduction

Face alignment is the process of moving and deforming a face model to an image, so as to extract the semantic meanings of facial pixels. It is an essential preprocessing step for many face analysis tasks, e.g. recognition [1], animation [2], tracking [3], attributes classification [4] and image restoration [5]. Traditionally, face alignment is approached as a landmark detection problem that aims to locate a sparse set of facial fiducial points, some of which include "eye corner", "nose tip" and "chin center". In the past two decades, a number of effective frameworks have been proposed such as ASM [6], AAM [7] and CLM [8]. Recently, with the introduction of Cascaded Regression [9], [10], [11] and Convolutional Neural Networks [12], [13], face alignment has observed significant improvements in accuracy. However, most of the existing methods are designed for medium poses, under the assumptions that the yaw angle is smaller than $45°$ and all the landmarks are visible. When the range of yaw angle is extended up to $90°$, significant challenges emerge. These challenges can be differentiated in three main ways:

Modelling: Landmark shape model [6] implicitly assumes that each landmark can be robustly detected by its distinctive visual patterns. However, when faces deviate from the frontal view, some landmarks become invisible due to self-occlusion [14]. In medium poses, this problem can be addressed by changing the semantic positions of face contour landmarks to the silhouette, which is termed landmark marching [15]. However, in large poses where half of face is occluded, some landmarks are inevitably invisible and show no detectable appearance. In turn, landmarks can lose their semantic meanings, which may cause the shape model to fail.

Fig. 1. Fitting results of 3DDFA (the blue/red points indicate visible/invisible landmarks). For each pair of the four results, on the left is the rendering of the fitted 3D face with the mean texture, which is made transparent to demonstrate the fitting accuracy. On the right is the landmarks overlayed on the fitted 3D face model.

Fitting: Another challenge in full-pose face alignment is derived from the dramatic appearance variations from front to profile. Cascaded Linear Regression [11] and traditional nonlinear models [16], [10] are not flexible enough to cover these complex variations in a unified way. Another framework demonstrates more flexibility by adopting different landmark and fitting models for differing view categories [14], [17], [18]. Unfortunately, since the nature of this framework must test every view, computational cost is likely to significantly increase. More recently, Convolutional Neural Network (CNN) based methods have demonstrated improved performance over traditional methods in many applications. For effective large-pose face alignment, CNN should be combined with the Cascaded Regression framework. However, most existing methods adopt a single network to complete fitting [13], which limits its performance.

Training Data: Labelled data is the basis for any supervised learning based algorithms. However, manual labelling of land-

- *X. Zhu, Z. Lei and S. Li are with Center for Biometrics and Security Research & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun Donglu, Beijing 100190, China. Email: {xiangyu.zhu,zlei,szli}@nlpr.ia.ac.cn.*
- *X. Liu is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA. Email: liuxm@msu.edu.*

marks on large-pose faces is very tedious since the occluded landmarks have to be "guessed" which is impossible for most of people. As a result, almost all the public face alignment databases such as AFW [18], LFPW [19], HELEN [20] and IBUG [21] are collected in medium poses. Few large-pose databases such as AFLW [22] only contain visible landmarks, which could be ambiguous in invisible landmarks, makes it hard to train a unified face alignment model.

In this paper, we aim to solve the problem of face alignment in full pose range, where the yaw angle is allowed to vary between $\pm 90°$. We believe that face alignment is not barely a 2D problem since self-occlusion and large appearance variations are caused by the face rotation in the 3D space, which can be conveniently addressed by incorporating 3D information. More specifically, we improve the face model from 2D sparse landmarks to a dense 3D Morphable Model (3DMM) [23] and consider face alignment as a 3DMM fitting task. The optimization concept therein will change accordingly from landmark positions to pose (scale, rotation and translation) and morphing (shape and expression) parameters. We call this novel face alignment framework 3D Dense Face Alignment (**3DDFA**). To realize 3DDFA, we propose to combine two achievements in recent years, namely, Cascaded Regression and the Convolutional Neural Network (CNN). This combination requires the introduction of a new **input feature** which fulfills the "cascade manner" and "convolution manner" simultaneously (see Sec. 3.2) and a new **cost function** which can model the priority of 3DMM parameters (see Sec. 3.4). Besides to provide enough data for training, we find that given a face image and its corresponding 3D model, it is possible to rotate the image out of plane with high fidelity. This rotation enables the synthesis of a large number of training samples in large poses.

In general, we propose a novel face alignment framework to address the three challenges of modelling, fitting and training data in large poses. The main contributions of the paper are summarized as follows:

1) To address the self-occlusion challenge, we assert that in large poses, fitting a 3DMM is more suitable than detecting 2D landmarks. The visibility estimated from 3DMM enables us to only fit the vertices with detected image patterns. The landmarks, if needed, can be sampled from the fitted 3D face afterwards. See the samples in Fig. 1.
2) To handle appearance variations across large poses, we propose a novel Cascaded Convolutional Neural Network as the regressor, in which two specially designed input features called Projected Normalized Coordinate Code (PNCC) and Pose Adaptive Feature (PAF) are introduced to connect CNNs in a cascade manner. Besides, a novel cost function named Optimized Weighted Parameter Distance Cost (OWPDC) is proposed to formulate the priority of 3DMM parameters during training.
3) To enable the training of the 3DDFA, we construct a face database consisting of pairs of 2D face images and 3D face models. We further elucidate a face profiling method to synthesize $60k+$ training samples across large poses. The synthesized samples well simulate the face appearances in large poses and boost the performance of both previous and the proposed face alignment approaches.

This paper is an extension of our previous work [24] the following four aspects: 1) Traditional 3DMM uses Euler angles to represent the 3D rotation, which shows ambiguity when the yaw angle reaches $90°$. In this paper, quaternions are used instead as the rotation formulation to eliminate the ambiguity. 2) A new input feature called Pose Adaptive Feature (PAF) is utilized to remedy the drawbacks of PNCC to further boost the performance. 3) We improve the cost function in [24] through the OWPDC which not only formulates the importance but also the priority of 3DMM parameters during training. 4) Additional experiments are conducted to better analyze the motivation behind the design of the input features and the cost function.

2 Related Works

Face alignment can be summarized as **fitting** a **face model** to an image. As such, there are two basic problems involved with this task: how to model the face shape and how to estimate the model parameters. In this section, we motivate our approach by discussing related works with respect to these two problems.

2.1 Face Model

Traditionally, face shape is represented by a sparse set of 2D facial fiducial points. Cootes et al. [6], [7] show that shape variations can be modeled with subspace analysis such as Principal Components Analysis (PCA). Although, this **2D-subspace model** can only cope with shape variations from a narrow range of face poses, since the non-linear out-of-plane rotation cannot be well represented with the linear subspace. To deal with the pose variations, some modifications like Kernel PCA [25] and Bayesian Mixture Model [14] are proposed to introduce non-linearity into the subspace models. Recently, Cao et al. [10] propose to abandon any explicit shape constraints and directly use landmark coordinates as the shape model, which called 2D Non-Parametric Model (**2D-NPM**). 2D-NPM considerably improves the flexibility of the shape model at the cost of losing any shape priors and increasing the difficulty of model fitting. Besides 2D shape model, Blanz et al. [26], [23] propose the 3D Morphable Model (**3DMM**) which applies PCA on a set of 3D face scans. By incorporating 3D information, 3DMM disentangles the non-linear out-of-plane transformation from the PCA subspace. The remaining shape and expression variations have shown high linearity [23], [2], which can be well modeled with PCA. Compared with 2D models, 3DMM separates rigid (pose) and non-rigid (shape and expression) transformations, enabling it to cover diverse shape variations and keep shape prior at the same time. Additionally, points visibility can be easily estimated by 3DMM [24], which can provide important clues to handle self-occlusion in profile views.

2.2 Model Fitting

Most fitting methods can be divided into two categories: the template fitting based [7], [27] and regression based [28], [9], [11], [29]. The template fitting methods always maintain a face appearance model to fit images. For example, Active Appearance Model (AAM) [7] and Analysis-by-Synthesis 3DMM Fitting [23] simulate the process of face image generation and achieve alignment by minimizing the difference between the model appearance and the input image. Active Shape Model (ASM) [6] and Constrained Local Model (CLM) [8], [30] build a template model for each landmark and use a PCA shape model to constrain the fitting results. TSPM [18] and CDM [17] employ part based model and DPM-like [31] method to align faces. Generally, the

performance of template fitting methods depends on whether the image patterns reside within the variations described by the face appearance model. Therefore, it shows limited robustness in unconstrained environment where appearance variations are too wide and complicated.

Regression based methods estimate model parameters by regressing image features. For example, Hou et al. [32] and Saragih et al. [33] perform regression between texture residuals and parameter updates to fit AAM. Valstar et al. [34] locate landmark positions by mapping the landmark related local patches with support vector regression. Recently, Cascaded Regression [9] has been proposed and becomes most popular in face alignment community [10], [11], [35], [36], which can be summarized in Eqn. 1:

$$\mathbf{p}^{k+1} = \mathbf{p}^k + Reg^k(Fea(\mathbf{I}, \mathbf{p}^k)). \quad (1)$$

where the shape parameter $\mathbf{p}^k$ at the kth iteration is updated by conducting regression Reg^k on the shape indexed feature Fea, which should depend on both the image $\mathbf{I}$ and the current parameter $\mathbf{p}^k$. The regression Reg^k shows an important "feedback" property that its input feature $Fea(\mathbf{I}, \mathbf{p})$ can be updated by its output since after each iteration $\mathbf{p}$ is updated. With this property an array of weak regressors can be cascaded to reduce the alignment error progressively.

Besides Cascaded Regression, another breakthrough is the introduction of Convolutional Neural Network (CNN), which formulates face alignment as a regression from raw pixels to landmarks positions. For example, Sun et al. [12] propose to use the CNN to locate landmarks in two stages, first the full set of landmarks are located with a global CNN and then each landmark is refined with a sub-network on its local patch. With one CNN for each landmark, the complexity of the method highly depends on the number of landmarks. Zhang et al. [13] combine face alignment with attribute analysis through multi-task CNN to boost the performance of both tasks. Wu et al. [37] cluster face appearances with mid-level CNN features and deal with each cluster with an independent regressor. Jourabloo et al. [38] arrange the local landmark patches into a large 2D map as the CNN input to regress model parameters. Trigeorgis et al. [29] convolve the landmark local patch as the shape index feature and conduct linear regression to locate landmarks.

2.3 Large Pose Face Alignment

Despite the great achievements in face alignment, most of the state-of-the-art methods lack the flexibility in large-pose scenarios, since they need to build the challenging relationship between the landmark displacement and landmark related image features, where the latter may be self-occluded. In 2D methods, a common solution is the multi-view framework which uses different landmark configurations for different views. It has been applied in AAM [39], DAM [40] and DPM [18], [17] to align faces with different shape models, among which the one having the highest possibility is chosen as the final result. However, since every view has to be tested, the computational cost is always high. Another method is explicitly estimating the visibility of landmarks and shrink the contribution of occluded features [14], [41], [42]. Nevertheless, occlusion estimation is itself a challenging task and handling varying dimensional feature is still an ill-posed problem.

Different from 2D methods, 3D face alignment [43] aims to fit a 3DMM [23] to a 2D image. By incorporating 3D information, 3DMM can inherently provide the visibility of each model point without any additional estimation, making it possible to deal with the self-occluded points. The original 3DMM fitting method [23] fits the 3D model by minimizing the pixel-wise difference between image and rendered face model. Since only the visible model vertices are fitted, it is the first method to cover arbitrary poses [23], [44], but it suffers from the one-minute-per-image computational cost. Recently, regression based 3DMM fitting, which estimates the model parameters by regressing the features at projected 3D landmarks [17], [45], [46], [47], [38], [48], [49], has looked to improve the efficiency. Although these methods face two major challenges. First the projected 3D landmarks may be self-occluded and lose their image patterns, making the features no longer pose invariant. Second, parameters of 3DMM have different priorities during fitting, despite that existing regression based methods treat them equally [10]. As a result, directly minimizing the parameter error may be sub-optimal, because smaller parameter errors are not necessarily equivalent to smaller alignment errors. This problem will be further discussed in Sec. 3.4. A relevant but distinct task is 3D face reconstruction [50], [15], [51], [52], which recovers a 3D face from given 2D landmarks. Interestingly, 2D/3D face alignment results can be mutually transformed, where 3D to 2D is made by sampling landmark vertices and 2D to 3D is made by 3D face reconstruction.

In this work, we propose a framework to combine three major achievements—3DMM, Cascaded Regression and CNN—to solve the large-pose face alignment problem.

3 3D Dense Face Alignment (3DDFA)

In this section, we introduce how to combine Cascaded Regression and CNNs to realize 3DDFA. By applying a CNN as the regressor in Eqn. 1, Cascaded CNN can be formulated as:

$$\mathbf{p}^{k+1} = \mathbf{p}^k + Net^k(Fea(\mathbf{I}, \mathbf{p}^k)). \quad (2)$$

There are four components in this framework: the regression objective $\mathbf{p}$ (Sec. 3.1), the image features Fea (Sec. 3.2), the CNN structure Net (Sec. 3.3) and the cost function to train the framework (Sec. 3.4).

3.1 3D Morphable Model

Blanz et al. [23] propose the 3D Morphable Model (3DMM) to describe the 3D face space with PCA:

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}, \quad (3)$$

where $\mathbf{S}$ is a 3D face, $\overline{\mathbf{S}}$ is the mean shape, $\mathbf{A}_{id}$ is the principle axes trained on the 3D face scans with neutral expression and $\boldsymbol{\alpha}_{id}$ is the shape parameter, $\mathbf{A}_{exp}$ is the principle axes trained on the offsets between expression scans and neutral scans and $\boldsymbol{\alpha}_{exp}$ is the expression parameter. In this work, the $\mathbf{A}_{id}$ and $\mathbf{A}_{exp}$ come from BFM [53] and FaceWarehouse [54] respectively. After the 3D face is constructed, it can be projected onto the image plane with scale orthographic projection:

$$V(\mathbf{p}) = f * \mathbf{Pr} * \mathbf{R} * (\overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}) + \mathbf{t}_{2d}, \quad (4)$$

where $V(\mathbf{p})$ is the model construction and projection function, leading to the 2D positions of model vertices, f is the scale factor, $\mathbf{Pr}$ is the orthographic projection matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $\mathbf{R}$ is the rotation matrix and $\mathbf{t}_{2d}$ is the translation vector. The collection of all the model parameters is $\mathbf{p} = [f, \mathbf{R}, \mathbf{t}_{2d}, \boldsymbol{\alpha}_{id}, \boldsymbol{\alpha}_{exp}]^{\mathrm{T}}$.

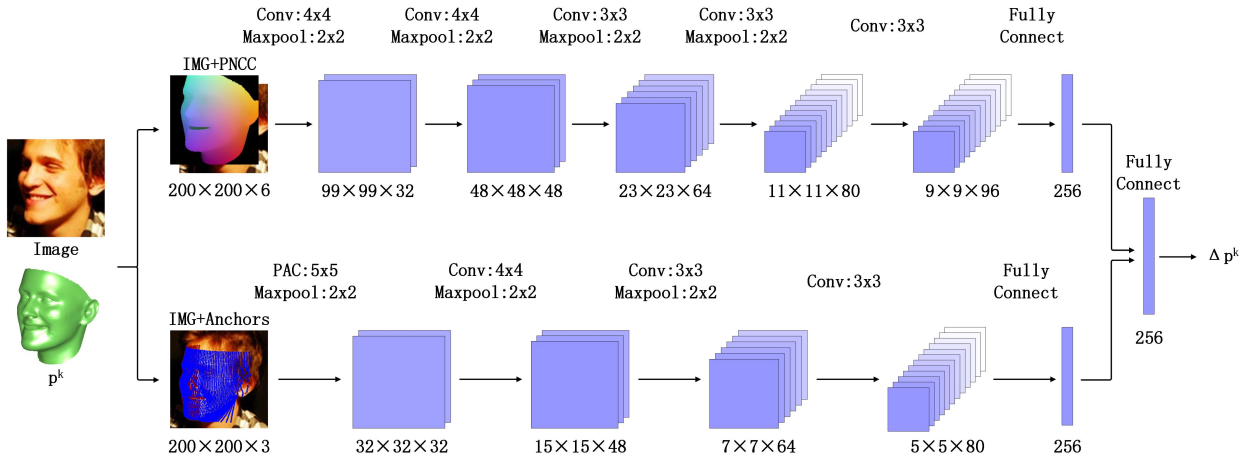


Fig. 2. An overview of the two-stream network in 3DDFA. With an intermediate parameter $\mathbf{p}^k$, in the first stream we construct a novel Projected Normalized Coordinate Code (PNCC), which is stacked with the input image and sent to the CNN. In the second stream, we get some feature anchors with consistent semantics and conduct Pose Adaptive Convolution (PAC) on them. The outputs of the two streams are merged with an additional fully connected layer to predict the parameter update $\Delta\mathbf{p}^k$.

3.1.1 Rotation Formulation

Face rotation is traditionally formulated with the Euler angles [55] including $pitch$, yaw and $roll$. However, when faces are close to the profile view, there is ambiguity in Euler angles termed gimbal lock [56], see Fig. 3 as a example.

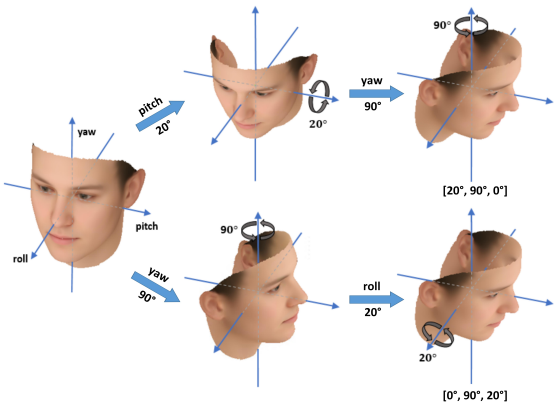


Fig. 3. An example of gimbal lock. We assume the rotation sequence is from $pitch$ to yaw to $roll$. In the first row, the face is firstly rotated $20°$ around the $pitch$ axis and then $90°$ around the yaw axis, whose Euler angles are $[20°, 90°, 0°]$. In the second row, the face is firstly rotated $90°$ around the yaw axis and then $20°$ around the $roll$ axis, whose Euler angles are $[0°, 90°, 20°]$. However the two different Euler angles correspond to the same rotation matrix, generating the profile view of a nodding face.

The ambiguity in Euler angles will confuse the regressor and affect the fitting performance. Therefore we adopt a four dimensional unit quaternion [56] $[q_0, q_1, q_2, q_3]$ instead of the Euler angles to formulate the rotation. The corresponding rotation matrix is:

$$\mathbf{R} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_0q_1+q_2q_3) \\ 2(q_0q_2+q_1q_3) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

In our implementation, we merge the scale parameter f into $[q_0, q_1, q_2, q_3]$ through dividing the quaternion by $\sqrt{f}$ and do not constrain the quaternion to be unit. As a result, the fitting objective will be $\mathbf{p} = [q_0, q_1, q_2, q_3, \mathbf{t}_{2d}, \boldsymbol{\alpha}_{id}, \boldsymbol{\alpha}_{exp}]^{\mathrm{T}}$.

3.2 Feature Design

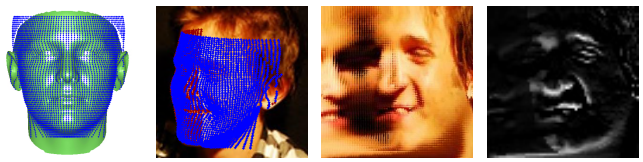
As the conjunction point of Cascaded Regression and CNN, the input feature should fulfill the requirements from both frameworks, which can be summarized as the following three aspects: Firstly, the **convolvable property** requires that the convolution operation on the input feature should make sense. As the CNN input, the feature should be a smooth 2D map reflecting the accuracy of current fitting. Secondly, to enable the cascade manner, the **feedback property** requires the input feature to depend on the CNN output [9]. Finally, to guarantee the cascade to converge at the ground truth parameter, the **convergence property** requires the input feature to be discriminative when the fitting is complete.

Besides the three requirements, we find that the input features of face alignment can be divided into two categories. The first category is the image-view feature, where the original image is directly sent to the regressor. For example, [12], [13], [37] use the input image as the CNN input and [57], [58] stack the image with a landmark response map as the input. These kind of features does not lose any information provided by the image but require the regressor to cover any face appearances. The second category is the model-view feature, where image pixels are rearranged based on the model condition. For example, AAM [7] warps the face image to the mean shape and SDM [11] extract SIFT features at landmark locations. This kind of features aligns the face appearance with current fitting, which simplifies the alignment task progressively during optimization. However, they do not cover the pixels beyond the face model, leading to a bad description of context. As such, fitting with model-view features is easily trapped in local minima [36]. In this paper, we propose a model-view feature called Pose Adaptive Feature (PAF) and a image-view feature called Projected Normalized Coordinate Code (PNCC). We further demonstrate that optimal results can be achieved by combining both features.

3.2.1 Pose Adaptive Convolution

Traditional convolutional layers convolve along a 2D map from pixel to pixel, while we intend to convolve at some semantically consistent locations on the face, called Pose Adaptive Convolution (PAC). Considering human face can be roughly approximated with

a cylinder [59], we compute the cylindrical coordinate of each vertex and sample 64×64 feature anchors with constant azimuth and height intervals, see Fig. 4(a).



(a) (b) (c) (d)

Fig. 4. Pose Adaptive Convolution (PAC): (a) The 64×64 feature anchors on the 3D face model. (b) The projected feature anchors $V(\mathbf{p})_{anchor}$ (the blue/red ones indicate visible/invisible anchors). (c) The feature patch map concatenated by the patches cropped at $V(\mathbf{p})_{anchor}$. (d) Conducting convolution, whose stride and filter size are the same with the patch size, on the feature patch map and shrinking the responses at invisible points, leading to the Pose Adaptive Feature (PAF).

Given a current model parameter $\mathbf{p}$, we first project 3DMM and sample the feature anchors on the image plane, getting $64 \times 64 \times 2$ projected feature anchors $V(\mathbf{p})_{anchor}$ (Fig. 4(b)). Second we crop $d \times d$ (5 in our implementation) patch at each feature anchor and concatenate the patches into a $(64*d) \times (64*d)$ patch map according to their cylindrical coordinates (Fig. 4(c)). Finally we conduct $d \times d$ convolutions at the stride of d on the patch map, generating 64×64 response maps (Fig. 4(d)). The convolutional filters are learned with a common convolutional layer, jointly with other CNN layers as described in Sec. 3.3.

Note that this process is equivalent to directly conducting $d \times d$ convolutions on the projected feature anchors $V(\mathbf{p})_{anchor}$, which implicitly localize and frontalize the face, making the convolution pose invariant. In order to shrink the features at the occluded region, we consider the vertices whose normal points to minus z as self-occluded and divide the responses at occluded region by two, generating the Pose Adaptive Feature (PAF). We do not eliminate occluded features as [45] since this information is still valuable prior to perfect fitting.

3.2.2 *Projected Normalized Coordinate Code*

The proposed image-view feature depends on a new type of vertex index, which is introduced as follows: we normalize the 3D mean face to $0-1$ in x, y, z axis as Eqn. 5:

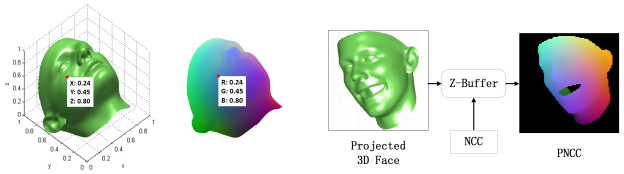
$$\text{NCC}_d = \frac{\overline{\mathbf{S}}_d - \min(\overline{\mathbf{S}}_d)}{\max(\overline{\mathbf{S}}_d) - \min(\overline{\mathbf{S}}_d)} \quad (d = x, y, z), \tag{5}$$

where the $\overline{\mathbf{S}}$ is the mean shape of 3DMM. After normalization, the 3D coordinate of each vertex **uniquely** distributes between $[0, 0, 0]$ and $[1, 1, 1]$, so it can be considered as a vertex index, which we call Normalized Coordinate Code (NCC) (Fig. 5(a)). Since NCC has three channels as RGB, we can also show NCC as the face texture. It can be seen as different from the traditional vertex index (from 1 to the number of vertices), NCC is smooth along the face surface.

In the fitting process, with a model parameter $\mathbf{p}$, we adopt Z-Buffer to render the projected 3D face colored by NCC (Fig. 5(b)) as in Eqn. 6:

$$\begin{aligned} \text{PNCC} &= \textit{Z-Buffer}(V_{3d}(\mathbf{p}), \text{NCC}), \\ V_{3d}(\mathbf{p}) &= \mathbf{R} * (\overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}) + [\mathbf{t}_{2d}, 0]^{\mathrm{T}}, \end{aligned} \tag{6}$$

where $\textit{Z-Buffer}(\boldsymbol{\nu}, \boldsymbol{\tau})$ renders the 3D mesh $\boldsymbol{\nu}$ colored by $\boldsymbol{\tau}$ and $V_{3d}(\mathbf{p})$ is the projected 3D face. We call the rendered image Projected Normalized Coordinate Code (PNCC). Afterwards, PNCC is stacked with the input image and sent to the CNN.



(a) NCC (b) PNCC

Fig. 5. The Normalized Coordinate Code (NCC) and the Projected Normalized Coordinate Code (PNCC). (a) The normalized mean face, which is also demonstrated with NCC as its texture (NCC_x = R, NCC_y = G, NCC_z = B). (b) The generation of PNCC, the projected 3D face is rendered by Z-Buffer with NCC as its colormap.

Comparing PAF and PNCC, we can see that PAF is a model-view feature since it implicitly warps the image with feature anchors and PNCC is an image-view feature it sends the original image into a CNN. Regarding the three properties, they fulfill the feedback property since they both depend on $\mathbf{p}$ which is updated by the output of the CNN. As for the convolvable property, PAC is the convolution on the continuous locations indicated by the feature anchors and its result PAF is a smooth 2D map. PNCC is also smooth in 2D and the convolution indicates the linear combination of NCCs on a local patch. As for the convergence property, when the CNN detects that in PAF the face is aligned to front and in PNCC each NCC superposes its corresponding image pattern, the cascade will converge.

3.3 Network Structure

Unlike existing CNN methods [12], [57] that apply different network structures for different fitting stages, 3DDFA employs a unified network structure across the cascade. In general, at iteration k ($k = 0, 1, ..., K$), given an initial parameter $\mathbf{p}^k$, we construct PNCC and PAF with $\mathbf{p}^k$ and train a two-stream CNN $\textit{Net}^k$ to conduct fitting. The output features from two streams are merged to predict the parameter update $\Delta\mathbf{p}^k$:

$$\Delta\mathbf{p}^k = \textit{Net}^k(\text{PAF}(\mathbf{p}^k, \mathbf{I}), \text{PNCC}(\mathbf{p}^k, \mathbf{I})). \tag{7}$$

Afterwards, a better intermediate parameter $\mathbf{p}^{k+1} = \mathbf{p}^k + \Delta\mathbf{p}^k$ becomes the input of the next network $\textit{Net}^{k+1}$ which has the same structure but different weights with $\textit{Net}^k$. Fig. 2 shows the network structure. In the PNCC stream, the input is the $200 \times 200 \times 3$ color image stacked by the $200 \times 200 \times 3$ PNCC. The network contains five convolutional layers, four pooling layers and one fully connected layer. In the PAF stream, the input is the $200 \times 200 \times 3$ color image and 64×64 feature anchors. The image is processed with the pose adaptive convolution, followed by three pooling layers, three convolutional layers and one fully connected layer. The outputs of the two streams are merged with an additional fully connected layer to predict the 234-dimensional parameter update including 6-dimensional pose parameters $[q_0, q_1, q_2, q_3, t_{2dx}, t_{2dy}]$, 199-dimensional shape parameters $\boldsymbol{\alpha}_{id}$ and 29-dimensional expression parameters $\boldsymbol{\alpha}_{exp}$.

3.4 Cost Function

Different from the landmark shape model, the parameters in 3DMM contribute to the fitting accuracy with very different impacts, giving parameters different priorities. As a result,

regression-based methods suffer from the inequivalence between parameter error and alignment error [10]. In this section, we will discuss this problem with two baseline cost functions and propose our own ways to model the parameter priority.

3.4.1 Parameter Distance Cost (PDC)

Take the first iteration as an example. The purpose of the CNN is to predict the parameter update $\Delta\mathbf{p}$ so as to move the initial parameter $\mathbf{p}^0$ closer to the ground truth $\mathbf{p}^g$. Intuitively, we can minimize the distance between the ground truth and the current parameter with the Parameter Distance Cost (PDC):

$$E_{pdc} = \|\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0)\|^2. \tag{8}$$

PDC has been traditionally used in regression based model fitting [32], [33], [60]. However, different dimension in $\mathbf{p}$ has different influences on the resultant 3D face. For example, with the same deviation, the yaw angle will bring a larger alignment error than a shape parameter, while PDC optimizes them equally, leading to sub-optimal results.

3.4.2 Vertex Distance Cost (VDC)

Since 3DDFA aims to morph the 3DMM to the ground truth 3D face, we can optimize $\Delta\mathbf{p}$ by minimizing the vertex distances between the current and the ground truth 3D face:

$$E_{vdc} = \|V(\mathbf{p}^0 + \Delta\mathbf{p}) - V(\mathbf{p}^g)\|^2, \tag{9}$$

where $V(\cdot)$ is the face construction and projection as Eqn. 4. We call this cost Vertex Distance Cost (VDC). Compared with PDC, VDC better models the fitting error by explicitly considering parameter semantics. However, VDC is not convex itself, the optimization is not guaranteed to converge to the ground truth parameter $\mathbf{p}^g$. Furthermore, we observe that VDC exhibits pathological curvature [61] since the directions of pose parameters always exhibit much higher curvatures than the PCA coefficients. As a result, optimizing VDC with gradient descent converges very slowly due to the "zig-zagging" problem. Second-order optimizations are preferred to handle the pathological curvature but they are expensive and hard to be implemented on GPU.

3.4.3 Weighted Parameter Distance Cost (WPDC)

In our previous work [24], we propose a cost function named Weighted Parameter Distance Cost (WPDC). The motivation is explicitly weighting parameter error by its importance:

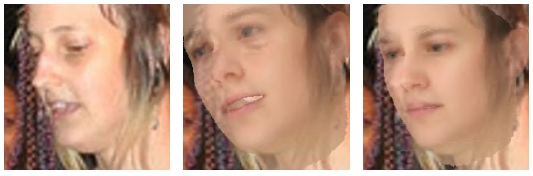
$$E_{wpdc} = (\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0))^{\mathrm{T}}\mathrm{diag}(\mathbf{w})(\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0)) \tag{10}$$

where $\mathbf{w}$ is the parameter importance vector, which is defined as follows:

$$\begin{aligned} \mathbf{w} &= (w_1, w_2, ..., w_i, ..., w_p), \\ w_i &= \|V(\mathbf{p}^{de,i}) - V(\mathbf{p}^g)\|/Z, \\ \mathbf{p}^{de,i} &= (\mathbf{p}^g_1, ..., \mathbf{p}^g_{i-1}, (\mathbf{p}^0 + \Delta\mathbf{p})_i, \mathbf{p}^g_{i+1}, ..., \mathbf{p}^g_p), \end{aligned} \tag{11}$$

where p is the number of parameter, $\mathbf{p}^{de,i}$ is the i-degraded parameter whose ith element comes from the predicted parameter $(\mathbf{p}^0 + \Delta\mathbf{p})$ and the others come from the ground truth parameter $\mathbf{p}^g$, Z is a regular term which is the maximum of $\mathbf{w}$. $\|V(\mathbf{p}^{de}(i)) - V(\mathbf{p}^g)\|$ models the alignment error brought by miss-predicting the ith model parameter, which is indicative of its importance. In the training process, the CNN firstly concentrates on the parameters with larger $\|V(\mathbf{p}^{de}(i)) - V(\mathbf{p}^g)\|$ such as rotation and translation. As $\mathbf{p}^{de}(i)$ is closer to $\mathbf{p}^g$, the weights of these parameters begin to shrink and the CNN will optimize less important parameters while simultaneously keeping the high-priority parameters sufficiently good. Compared with VDC, WPDC makes sure the parameter is optimized toward $\mathbf{p}^g$ and it remedies the pathological curvature issue at the same time.

However, the weight in WPDC only models the "importance" but not the "priority". In fact, parameters become important sequentially. Take Fig. 6 as an example, when WPDC evaluates a face image with open mouth and large pose, it will assign both expression and rotation high weights. We can observe that attempting to estimate expression makes little sense before the pose is accurate enough, see Fig. 6(b). One step further, if we force the CNN to only concentrate on pose parameters, we obtain a better fitting result, see Fig. 6(c). Consequently for this sample, even though pose and expression are both important, pose has higher priority than expression, but WPDC misses that.



(a) Image (b) Error = 10.72 (c) Error = 4.39

Fig. 6. (a) An open-mouth face in near-profile view. (b) The fitting result of WPDC in the first iteration. (c) The fitting result when the CNN is restricted to only regress the 6-dimensional pose parameters. Errors are measured by Normalized Mean Error.

3.4.4 Optimized Weighted Parameter Distance Cost (OWPDC)

We can observe that "priority" is a between-parameter relationship which can only be modeled by treating all the parameters as a whole rather than evaluating them separately as WPDC. In this paper, we propose to find the best weights through optimization:

$$\begin{aligned} E_{owpdc} &= (\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0))^{\mathrm{T}}\mathrm{diag}(\mathbf{w}^*)(\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0)), \\ \mathbf{w}^* &= \arg\min_{\mathbf{w}} \left\| V\left(\mathbf{p}^c + \mathrm{diag}(\mathbf{w}) * (\mathbf{p}^g - \mathbf{p}^c)\right) - V\left(\mathbf{p}^g\right) \right\|^2 \\ &\quad + \lambda \left\| \mathrm{diag}(\mathbf{w}) * (\mathbf{p}^g - \mathbf{p}^c) \right\|^2, \\ &\quad s.t. \quad \mathbf{0} \preceq \mathbf{w} \preceq \mathbf{1}, \end{aligned} \tag{12}$$

where $\mathbf{w}$ is the weights vector, $\Delta\mathbf{p}$ is the CNN output, $\mathbf{p}^c = \mathbf{p}^0 + \Delta\mathbf{p}$ is the current predicted parameter, $\mathbf{0}$ and $\mathbf{1}$ are the zeros and ones vectors respectively and $\preceq$ is the element-wise less than. In Eqn. 12, by adding a weighted parameter update $\mathrm{diag}(\mathbf{w})(\mathbf{p}^g - \mathbf{p}^c)$ to the current parameter $\mathbf{p}^c$, we hope the new face is closer to the ground truth face with limited updating. Note that $\|\mathrm{diag}(\mathbf{w}) * (\mathbf{p}^g - \mathbf{p}^c)\|^2$ is the square sum of the gradient of OWPDC, which models how much CNN weights need to be tuned to predict each parameter. We use this penalty term to choose the parameters which are most beneficial to the fitting and are easiest to learn. The range of $\mathbf{w}$ is constrained to be $[0, 1]$ to make sure the parameter is optimized to $\mathbf{p}^g$. Obviously, when the λ is set to 0, there will be a trivial solution that $\mathbf{w} = \mathbf{1}$ and OWPDC will deteriorate to PDC.

In the training process, directly optimizing Eqn. 12 for each sample is computationally intensive. We expand $V(\mathbf{p}^c +$

$\mathrm{diag}(\mathbf{w})(\mathbf{p}^g - \mathbf{p}^c))$ at $\mathbf{p}^g$ with the Taylor formula and let $\Delta\mathbf{p}^c = \mathbf{p}^g - \mathbf{p}^c$, Eqn. 12 will be:

$$\left\|V'(\mathbf{p}^g) * \mathrm{diag}(\mathbf{w} - \mathbf{1}) * \Delta\mathbf{p}^c\right\|^2 + \lambda \left\|\mathrm{diag}(\mathbf{w}) * \Delta\mathbf{p}^c\right\|^2, \quad (13)$$

where $V'(\mathbf{p}^g)$ is the Jacobian. Expanding Eqn. 13 and removing the constant terms, we get:

$$\begin{aligned} &\mathbf{w}^{\mathrm{T}}\Big(\mathrm{diag}(\Delta\mathbf{p}^c)V'(\mathbf{p}^g)^{\mathrm{T}}V'(\mathbf{p}^g)\mathrm{diag}(\Delta\mathbf{p}^c)\Big)\mathbf{w} \\ -2 * &\mathbf{1}^{\mathrm{T}}\Big(\mathrm{diag}(\Delta\mathbf{p}^c)V'(\mathbf{p}^g)^{\mathrm{T}}V'(\mathbf{p}^g)\mathrm{diag}(\Delta\mathbf{p}^c)\Big)\mathbf{w} \\ &+\lambda * \mathbf{w}^{\mathrm{T}}\mathrm{diag}(\Delta\mathbf{p}^c . * \Delta\mathbf{p}^c)\mathbf{w}, \end{aligned} \quad (14)$$

where $.*$ is the element-wise multiplication. Let $\mathbf{H} = V'(\mathbf{p}^g)\mathrm{diag}(\Delta\mathbf{p}^c)$ which is a $2n \times p$ matrix where n is the number of vertices and p is the number of parameters, the optimization will be:

$$\begin{aligned} \arg\min_{\mathbf{w}} &\mathbf{w}^{\mathrm{T}} * (\mathbf{H}^{\mathrm{T}} * \mathbf{H} + \lambda * \mathrm{diag}(\Delta\mathbf{p}^c . * \Delta\mathbf{p}^c)) * \mathbf{w} \\ &+2 * \mathbf{1}^{\mathrm{T}} * \mathbf{H}^{\mathrm{T}} * \mathbf{H} * \mathbf{w}, \\ s.t. \quad &\mathbf{0} \preceq \mathbf{w} \preceq \mathbf{1}, \end{aligned} \quad (15)$$

which is a standard quadratic programming problem with the unique solution. The most consuming component in Eqn. 15 is the computation of $V'(\mathbf{p}^g)$. Fortunately, $\mathbf{p}^g$ is constant during training and $V'(\mathbf{p}^g)$ can be pre-computed offline. As a result, the computation of $\mathbf{w}^*$ can be reduced to a p-dimensional quadratic programming which can be efficiently solved. The only parameter in OWPDC is the λ. It directly determines which parameter is valid during training. We set $\lambda = 0.17 * \|V(\mathbf{p}^c) - V(\mathbf{p}^g)\|^2$ in our implementation.

4 Face Profiling

All the regression based methods rely on training data, especially for CNNs which have thousands of parameters to learn. Therefore, massive labelled faces in large poses are crucial for 3DDFA. However, few of the released face alignment databases contain large-pose samples [18], [19], [20], [21] since labelling standardized landmarks on them is very challenging. In this work, we demonstrate that profile faces can be well synthesized from existing training samples with the help of 3D information. Inspired by the recent achievements in face frontalization [15], [62] which generates the frontal view of faces, we propose to invert this process to synthesize the profile view of faces from medium-pose samples, which is called face profiling. Different from the face synthesizing in recognition [63], face profiling is not required to keep the identity information but to make the synthesizing results realistic. However, current synthesizing methods do not keep the external face region [64], [63], which contains important context information for face alignment. In this section, we elucidate a novel face synthesizing method to generate the profile views of face image with out-of-plane rotation, providing abundant realistic training samples for 3DDFA.

4.1 3D Image Meshing

The depth estimation of a face image can be conducted on the face region and the external region respectively, with different requirements of accuracy. On the face region, we fit a 3DMM through the Multi-Features Framework (MFF) [44] (see Fig. 7(b)). With the ground truth landmarks as a solid constraint throughout the fitting process, MFF can always get accurate results. Few difficult samples can be easily adjusted manually. On the external region, we follow the 3D meshing method proposed by Zhu et al. [15] to mark some anchors beyond the face region and simulate their depth, see Fig. 7(c). Afterwards the whole image can be tuned into a 3D object through triangulation (see Fig. 7(c)7(d)).

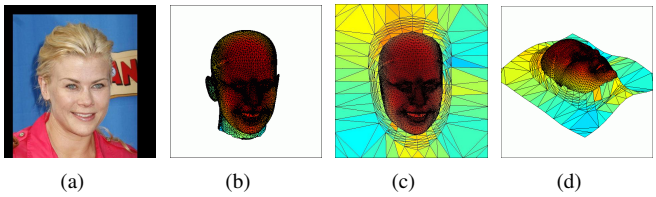


Fig. 7. 3D Image Meshing. (a) The input image. (b) The fitted 3D face through MFF. (c) The depth image from 3D meshing. (d) A different view of the depth image.

4.2 3D Image Rotation

The simulated depth information enables the 2D image to rotate out of plane to generate the appearances in larger poses. However, as shown in Fig. 8(b), the 3D rotation squeezes the external face region and loses the background. As a result, we need to further adjust the anchors to keep the background relatively unchanged and preserve the smoothness simultaneously. Inspired by our previous work [15], we propose to adjust background anchors by solving an equation list about their relative positions.

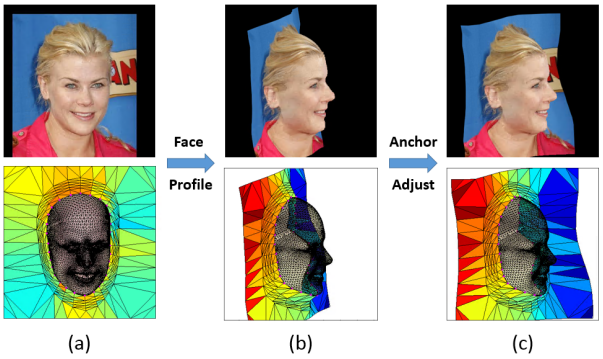


Fig. 8. The face profiling and anchor adjustment process. (a) The source image. (b) The profiled face with out of plane rotation. It can be seen that the face locates on the hollow since the background is squeezed. (c) The synthesized image after anchor adjustment.

In the source image as shown in Fig. 8(a), the triangulated anchors build up a graph where the anchors are the vertices and the mesh lines are the edges. In the graph, each edge represents an anchor-to-anchor relationship:

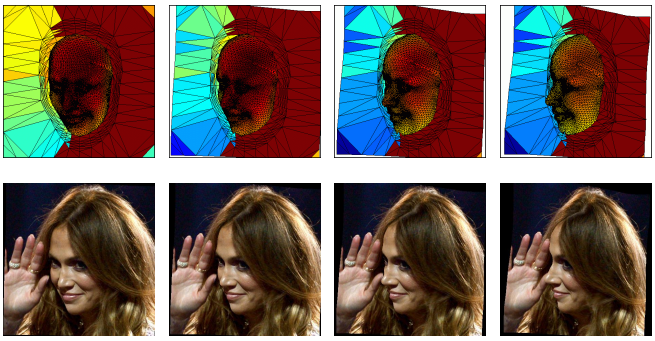
$$x_{a\_src} - x_{b\_src} = \Delta x_{src}, \qquad y_{a\_src} - y_{b\_src} = \Delta y_{src}, \quad (16)$$

where (x_{a_src}, y_{a_src}) and (x_{b_src}, y_{b_src}) are two connecting anchors, Δx_{src} and Δy_{src} are the spatial offsets in x, y axes, which should be preserved in synthesizing. After profiling, we keep the face contour anchors (the magenta points in Fig. 8(b)) consistent and predicting other anchors with the unchanged anchor offsets:

$$x_{a\_adj} - x_{b\_adj} = \Delta x_{src}, \quad y_{a\_adj} - y_{b\_adj} = \Delta y_{src}, \quad (17)$$

Specifically, if a is a face contour anchor, we set (x_{a_adj}, y_{a_adj}) to the positions after profiling (x_{a_pro}, y_{a_pro}), otherwise (x_{a_adj}, y_{a_adj}) are two unknowns need to be solved. By collecting Eqn. 17 for each graph edge, we form an equation list whose least square solution is the adjusted anchors (as seen in Fig. 8(c)).

In this work, we enlarge the yaw angle of image at the step of $5°$ until $90°$, see Fig. 9. Different from face frontalization, with larger rotation angles the self-occluded region can only be expanded. As a result, we avoid the troubling invisible region filling which may produce large artifacts [15]. Through face profiling, we not only obtain face samples in large poses but also augment the dataset to a large scale.



(a) (b) (c) (d)

Fig. 9. 2D and 3D view of face profiling. (a) The original yaw angle yaw_0. (b) $yaw_0 + 20°$. (c) $yaw_0 + 30°$. (d) $yaw_0 + 40°$.

5 Implementation

Training Strategy: With a huge number of parameters, the CNN tends to overfit the training set and the deeper cascade might learn nothing with overfitted samples. Therefore we regenerate $\mathbf{p}^k$ at each iteration using a nearest neighbor strategy. By observing that the fitting error highly depends on the ground truth face posture (FP), we perturb a training sample based on a set of similar-FP validation samples. In this paper, we define the face posture as the rotated 3D face without scaling and translation:

$$\text{FP} = \mathbf{R}^g * (\overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id}^g + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}^g), \tag{18}$$

where $\mathbf{R}^g$ is constructed from the normalized ground truth quaternion, $\boldsymbol{\alpha}_{id}^g$ and $\boldsymbol{\alpha}_{exp}^g$ are the ground truth shape and expression parameters respectively. Before training, we select two folds of samples as the validation set and for each training sample we construct a validation subset $\{v_1, ..., v_m\}$ whose members share similar FP with the training sample. At iteration k, we regenerate the initial parameter by:

$$\mathbf{p}^k = \mathbf{p}^g - (\mathbf{p}_{v_i}^g - \mathbf{p}_{v_i}^k), \tag{19}$$

where $\mathbf{p}^k$ and $\mathbf{p}^g$ are the initial and ground truth parameter of a training sample, $\mathbf{p}_{v_i}^k$ and $\mathbf{p}_{v_i}^g$ come from a validation sample v_i which is randomly chosen from the corresponding validation subset. Note that v_i is never used in training.

Initialization: Besides the face profiling, we also augment the training data (10 times) by randomly in-plane rotating images (up to 30 degrees) and perturbing bounding boxes. Specifically, the bounding boxes are randomly perturbed by a multivariate normal distribution whose mean vector and covariance matrix are obtained by the difference between ground truth bounding boxes and automated detected face rectangles using FTF [65]. This augmentation is quite effective in improving the robustness of the model. During testing, to get $\mathbf{p}^0$ we first set $\boldsymbol{\alpha}_{id}$, $\boldsymbol{\alpha}_{exp}$ to zero and the quaternion to $[1, 0, 0, 0]$, getting a frontal 3D mean face. Then we calculate $\mathbf{t}_{2d}$ by moving the mean point of the 3D face to the center of the bounding box. Finally, we scale the 3D face, which is equivalent to scaling the quaternion, to make the bounding box enclose the whole face region.

Running Time: During testing, 3DDFA takes 21.3ms for each iteration, among which PAF and PNCC take 11.6ms and 6.8ms respectively on 3.40GHZ CPU and CNN forward propagation takes 2.9ms on GTX TITAN X GPU. In our implementation, 3DDFA has three iterations and takes 63.9ms (15.65fps) for each sample. Note that the efficiency is mainly limited by the input features, which can be further improved by GPU implementation.

6 Experiments

6.1 Datasets

Three databases are used in our experiments, i.e. 300W-LP, AFLW [22] and a specifically constructed AFLW2000-3D.

300W-LP: 300W [66] standardises multiple face alignment databases with 68 landmarks, including AFW [18], LFPW [67], HELEN [68], IBUG [66] and XM2VTS [69]. With 300W, we adopt the proposed face profiling to generate $61,225$ samples across large poses ($1,786$ from IBUG, $5,207$ from AFW, $16,556$ from LFPW and $37,676$ from HELEN, XM2VTS is not used), which is further flipped to $122,450$ samples. We call the synthesized database as 300W Across Large Poses (300W-LP).

AFLW: AFLW [22] contains $21,080$ in-the-wild faces with large pose variations (yaw from $-90°$ to $90°$). Each image is annotated up to 21 visible landmarks. The database is very suitable for evaluating face alignment performance in large poses.

AFLW2000-3D: Evaluating 3D face alignment in the wild is difficult due to the lack of pairs of 2D image and 3D scan. Considering the recent achievements in 3D face reconstruction which can construct a 3D face from 2D landmarks [50], [15], we assume that a 3D model can be accurately fitted if sufficient 2D landmarks are provided. Therefore the evaluation can be degraded to 2D landmark evaluation which also makes it possible to compare 3DDFA with other 2D face alignment methods. While AFLW is not suitable for this task since only visible landmarks may lead to serious ambiguity in 3D shape, as reflected by the fake good alignment phenomenon in Fig. 10. In this work, we construct a database called AFLW2000-3D for 3D face alignment evaluation, which contains the ground truth 3D faces and the corresponding 68 landmarks of the first 2,000 AFLW samples. More details about the construction of AFLW2000-3D are given in supplemental material.

In all the following experiments, we follow [36] and regard the 300W-LP samples synthesized from the training part of LFPW, HELEN and the whole AFW as the training set ($101,144$ images in total). The testing are conducted on three databases: the 300W testing part for general face alignment, the AFLW for large-pose face alignment and the AFLW2000-3D for 3D face alignment. The alignment accuracy is evaluated by the Normalized Mean Error (NME).

6.2 Performance with Different Input Features

As described in Sec. 3.2, the input features of face alignment methods can be divided into two categories, the image-view

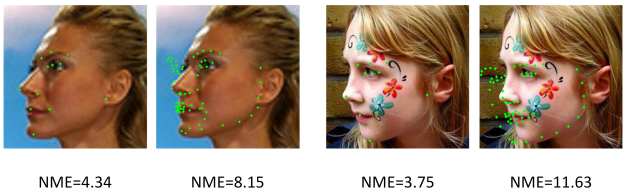


Fig. 10. Fake good alignment in AFLW. For each sample, the first shows the visible 21 landmarks and the second shows all the 68 landmarks. The Normalized Mean Error (NME) reflects their accuracy. It can be seen that only evaluating visible landmarks cannot well reflect the accuracy of 3D fitting.

feature and the model-view feature, which correspond to PNCC and PAF in this paper. To test their effectiveness respectively and evaluate their complementarity, we divide the network in Fig. 2 into PNCC stream and PAF stream by removing the last fully connected layer and regress the 256-dimensional output of each stream to the parameter update respectively. The combined two-stream network is also reported to demonstrate the improvements. As shown in Fig. 11, PNCC performs better than PAF when used

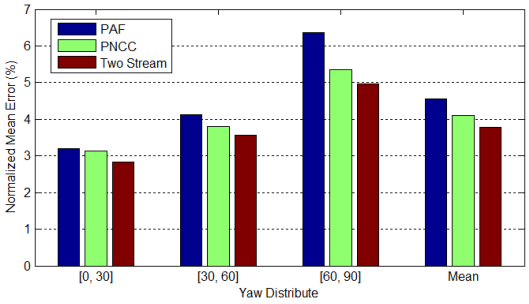


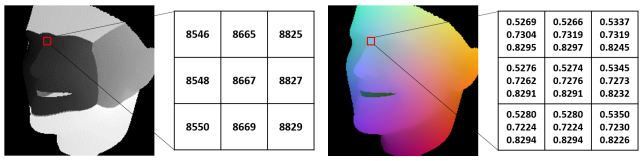
Fig. 11. The Normalized Mean Error (%) with different input features, evaluated on AFLW2000-3D with different yaw intervals.

individually and the improvement is enlarged as the pose becomes larger. Besides, PNCC and PAF achieve better performance when combined, which may infer a complementary relationship. This complementary relationship might be because PNCC covers the whole image and contains rich context information, enabling it to fit large scale facial components like the face contour. While PAF is more adept at fitting facial features due to the implicit frontalizion, which can well assist PNCC.

6.3 Analysis of Feature Properties

In Sec. 3.2, we introduce three requirements of the input feature: feedback, convolvable and convergence. Among them, the benefits from convolvable and convergence may not be obvious and are further evaluated here. Corresponding to PNCC and PAF, we propose two alternative input features which miss these two properties respectively.

Convovable Property: As the alternative to PNCC, we propose the Projected Index (PIndex) which renders the projected 3D face with the 1-channel vertex index (from 1 to 53,490 in BFM [53]) rather than the 3-channel NCC, see Fig. 12. Note that even though PIndex provides the semantic meaning of each pixel, it is not smooth and the convolution of vertex indexes on a local patch is hard to be interpreted by the CNN. As a result, PIndex violates the convolvable requirement. Using the PNCC stream as the network, we adopt PNCC and PIndex as the input feature respectively. As



(a) PIndex (b) PNCC

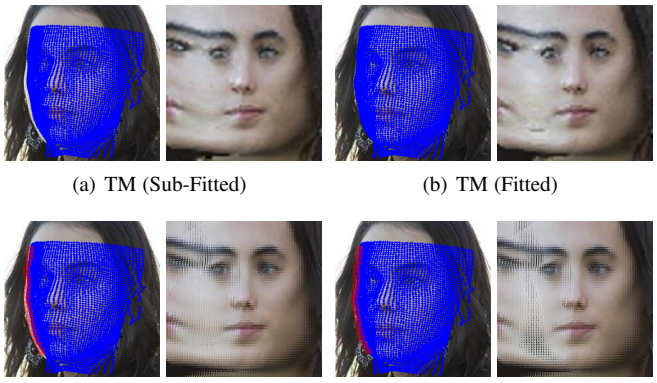
Fig. 12. The convolvable property of PNCC and PIndex: (a) A local patch of PIndex. The values can only be smooth in the indexing direction (vertical in this figure). (b) A local patch of PNCC. Values are smooth in 2D along each channel.

shown in Table 1, by violating the convolvable requirement, the performance drops since the learning task becomes more difficult.

TABLE 1
The NME(%) of PAF, PNCC and their corresponding alternative features, evaluated on AFLW2000-3D with different yaw interval.

Feature	[0, 30]	[30, 60]	[60, 90]	Mean
PIndex	3.33	3.95	5.60	4.29
PNCC	3.14	3.81	5.35	4.10
TM	3.38	4.48	6.76	4.87
PAF	3.20	4.12	6.36	4.56

Convergence Property: As the alternative to PAF, we propose the Texture Mapping (TM) [50] which rearranges the pixels on the projected feature anchors to a 64×64 image, see Fig. 13. Compared with PAF, the main drawback of TM is the weak



(a) TM (Sub-Fitted) (b) TM (Fitted)

(c) PAF (Sub-Fitted) (d) PAF (Fitted)

Fig. 13. The convergence property of TM and PAF. The first row: the mapped textures from a sub-fitted (a) and a fitted (b) sample. They show very similar appearances. The second row: the feature patch maps of PAF from a sub-fitted (c) and a fitted (d) sample. The convolution on the face contour vertices (the red grid) cover the pixels beyond the face region, enable PAF to exhibit discriminative appearance when the face contour is fitted.

description beyond the model region. As shown in Fig. 13(a) and Fig. 13(b), TM cannot discriminate whether the projected 3D model occludes the face in the image completely [70]. As a result, whether the fitting is complete is not discriminative for TM, which means the convergence requirement is not fulfilled. On the contrary, PAF can better describe the context information with the convolution on the face contour vertices. As shown in Fig. 13(c) and Fig. 13(d), PAF shows different appearances before and after the face contour is fitted. Table 1 shows the results of PAF and

TM which use the PAF stream as the network. We can see that PAF outperforms TM by over 6% which verifies the effectiveness of the convergence property.

6.4 Analysis of Cost Function

Performance with Different Cost: We demonstrate the errors along the cascade with different cost functions including PDC, VDC, WPDC and OWPDC. Fig. 14 demonstrates the testing error at each iteration. All the networks are trained until convergence. It is shown that PDC cannot well model the fitting error and

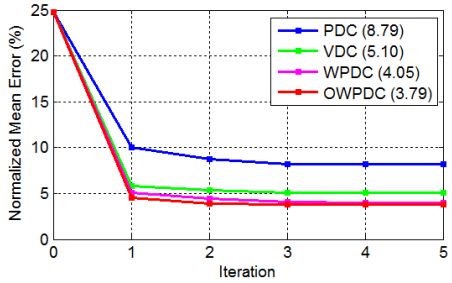


Fig. 14. The testing errors with different cost functions, evaluated on AFLW2000-3D. The value in the bracket indicates the NME after the third iteration.

converges to an unsatisfied result. VDC is better than PDC, but the pathological curvature problem makes it only concentrate on a small set of parameters and limits its performance. WPDC models the importance of each parameter and achieves a better result. Finally OWPDC further models the parameter priority, leading to faster convergence and the best performance.

Weights of OWPDC: Since the weights of OWPDC reflect the priority of parameters, how the priority changes along the training process is also an interesting point to investigate. In this experiment, for each mini-batch during training, we record the mean weights of the mini-batch and plot the mini-batch weight in Fig. 15. It can be seen that at beginning, the pose parameters (rotation and translation) show much higher priority than morphing parameters (shape and expression). As the training proceeds with error reducing, the pose weights begin to decrease and the CNN deals out its concentration to morphing parameters.

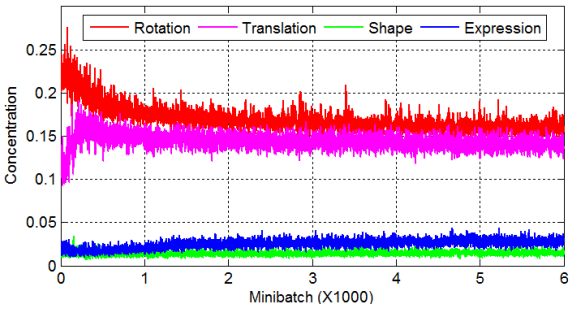


Fig. 15. The mean weights of each mini-batch along the training process in the first iteration. The weights are normalized by $\mathbf{w}/\sum\mathbf{w}$ for better representation. The curves indicate the max value among the quaternion (rotation curve), x and y translation (translation curve), PCA shape (shape curve) and expression parameters (expression curve).

6.5 Error Reduction in Cascade

To analyze the overfitting problem in Cascaded Regression and evaluate the effectiveness of initialization regeneration, we divide 300W-LP into $97,967$ samples for training and $24,483$ samples for testing, without identity overlapping. Fig. 16 shows the training and testing errors at each iteration, without and with initialization regeneration. As observed, in traditional Cascaded Regression the

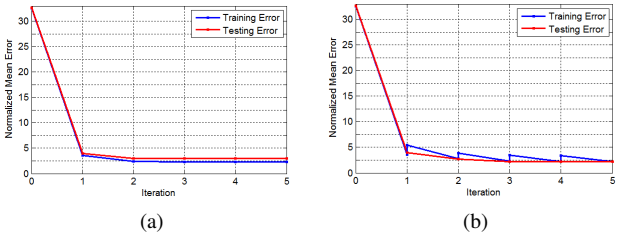


Fig. 16. The training and testing errors without (a) and with (b) initialization regeneration.

training and testing errors converge fast after two iterations. While with initialization regeneration, the training error is updated at the beginning of each iteration and the testing error continues descending. Considering both effectiveness and efficiency we choose three iterations in 3DDFA.

6.6 Comparison Experiments

In this paper, we evaluate the performance of 3DDFA on three different tasks: the large-pose face alignment on AFLW, the 3D face alignment on AFLW2000-3D and the medium-pose face alignment on 300W.

6.6.1 Large Pose Face Alignment on AFLW

Protocol: In this experiment, we regard the whole AFLW as the testing set and divide it into three subsets according to their absolute yaw angles: $[0°, 30°]$, $[30°, 60°]$, and $[60°, 90°]$ with $11,596$, $5,457$ and $4,027$ samples respectively. The alignment accuracy is evaluated by the Normalized Mean Error (NME), which is the average of landmarks error normalised by face size [45]. The face size is defined as the $\sqrt{width * height}$ of the bounding box (the rectangle hull of all the 68 landmarks). Besides, we report the standard deviation of NMEs across testing subsets to measure the pose robustness. During training, we use the projected 3D landmarks as the ground truth to train 2D methods. For convenient comparison, the ground truth bounding boxes are used for initialization.

Methods: Since little experiment has been conducted on the whole AFLW, we choose some baselines with released training codes,

Fig. 17. Results of SDM, DCN and our approach on AFLW.

TABLE 2
The NME(%) of face alignment results on AFLW and AFLW2000-3D with the first and the second best results highlighted. The brackets show the training sets.

	AFLW Dataset (21 pts)					AFLW2000-3D Dataset (68 pts)				
Method	$[0, 30]$	$[30, 60]$	$[60, 90]$	Mean	Std	$[0, 30]$	$[30, 60]$	$[60, 90]$	Mean	Std
LBF(300W)	7.17	17.54	28.45	17.72	10.64	6.17	16.48	25.90	16.19	9.87
LBF(300W-LP)	8.43	9.54	13.06	10.34	2.42	8.15	9.49	12.91	10.19	2.45
ESR(300W)	5.58	10.62	20.02	12.07	7.33	4.38	10.47	20.31	11.72	8.04
ESR(300W-LP)	5.66	7.12	11.94	8.24	3.29	4.60	6.70	12.67	7.99	4.19
CFSS(300W)	4.68	9.78	23.07	12.51	9.49	**3.44**	10.90	24.72	13.02	10.08
CFSS(300W-LP)	5.42	6.73	11.48	7.88	3.19	4.77	6.71	11.79	7.76	3.63
RCPR(300W)	5.40	9.80	20.61	11.94	7.83	4.16	9.88	22.58	12.21	9.43
RCPR(300W-LP)	5.43	6.58	11.53	7.85	3.24	4.26	5.96	13.18	7.80	4.74
MDM(300W)	5.14	10.95	24.11	13.40	9.72	4.64	10.35	24.21	13.07	10.07
MDM(300W-LP)	5.57	5.99	9.96	7.17	2.43	4.85	5.92	8.47	6.41	1.86
SDM(300W)	**4.67**	6.78	16.13	9.19	6.10	3.56	7.08	17.48	9.37	7.23
SDM(300W-LP)	4.75	5.55	9.34	6.55	2.45	3.67	4.94	9.76	6.12	3.21
TSPM(300W-LP)	5.91	6.52	7.68	6.70	0.90	-	-	-	-	-
RMFA	5.67	7.77	11.29	8.24	2.84	4.96	8.44	13.92	9.11	4.52
DCN(300W-LP)	4.99	5.47	8.10	6.19	1.68	3.93	4.67	**7.71**	5.44	**2.00**
3DDFA(Pre) [24]	5.00	**5.06**	**6.74**	**5.60**	**0.99**	3.78	**4.54**	7.93	**5.42**	2.21
Proposed	**4.11**	**4.38**	**5.16**	**4.55**	**0.54**	**2.84**	**3.57**	**4.96**	**3.79**	**1.08**

including RCPR [42], ESR [10], LBF [35], CFSS [36], SDM [71], MDM [29], RMFA [72] and TSPM [18]. Among them RCPR is an occlusion-robust method with the potential to deal with self-occlusion and we train it with landmark visibility computed by 3D information [62]. ESR, SDM, LBF and CFSS are popular Cascaded Regression based methods, among which SDM [71] is the winner of ICCV2013 300W face alignment challenge. MDM is a deep learning base method which adopts CNNs to extract image features. TSPM and RMFA adopt the multi-view framework which can deal with large poses. Besides the state-of-the-art methods, we introduce a Deep Convolutional Network (DCN) as a CNN based baseline. DCN directly regresses raw image pixels to the landmark positions with a CNN. The CNN has five convolutional layers, four pooling layers and two fully connected layers (the same as the PNCC stream) to estimate 68 landmarks from a $200 \times 200 \times 3$ input image. Besides, we also compare with our previous work [24] but we do not adopt the SDM based landmark refinement here.

Table 2 shows the comparison results and Fig. 18 shows the corresponding CED curves. Each 2D method is trained on 300W and 300W-LP respectively to demonstrate the boost from face profiling. For DCN, 3DDFA and TSPM which depend on large scales of data or large-pose data, we only evaluate the models trained on 300W-LP. Given that RMFA only releases the testing code, we just evaluate it with the provided model. Besides, in large poses TSPM model only detects 10 of the 21 landmarks, we only evaluate the error of the 10 points for TSPM.

Results: Firstly, the results indicate that all the methods benefit substantially from face profiling when dealing with large poses. The improvements in $[60°, 90°]$ exceed 40% for all the methods. This is especially impressive since the alignment models are trained on the synthesized data and tested on real samples, which well demonstrates the fidelity of face profiling. Secondly, in near frontal view, most of methods show very similar performance as shown in Fig 18(a). As the yaw angle increases in Fig 18(b) and Fig 18(c), most of 2D methods begin to degrade but 3DDFA could still maintain its performance. Finally, 3DDFA reaches the state of the art above all the 2D methods especially beyond medium poses. The minimum standard deviation also demonstrates its robustness to pose variations.

In Fig. 17, we demonstrate some alignment results of 3DDFA and representative 2D methods. Besides, Fig. 20 show some typical failure cases.

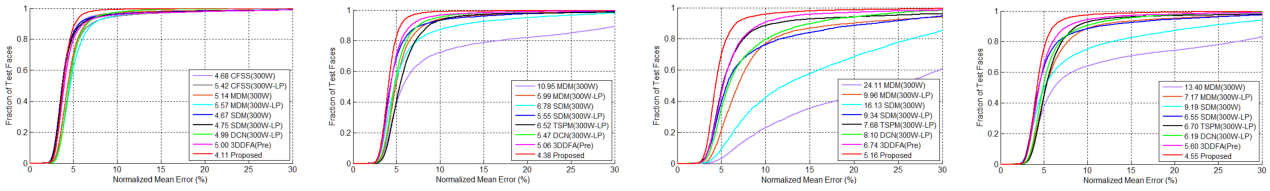
6.6.2 3D Face Alignment in AFLW2000-3D

As described in Section 6.1, 3D face alignment evaluation can be degraded to full-landmarks evaluation considering both visible and invisible ones. Using AFLW2000-3D as the testing set, this experiment follows the same protocol as AFLW, except all the 68 landmarks are used for evaluation. There are $1,306$ samples in $[0°, 30°]$, 462 samples in $[30°, 60°]$ and 232 samples in $[60°, 90°]$. The results are demonstrated in Table 2 and the CED curves are ploted in Fig. 19. We do not report the performance of TSPM models since they do not detect invisible landmarks.

Compared with the results in AFLW, we can see that the standard deviation is dramatically increased, meaning that it is more difficult to keep pose robustness when considering all the landmarks. Besides, the improvement of 3DDFA over the best 2D method DCN is increased from 26.49% in AFLW to 30.33% in AFLW2000-3D, which demonstrates the superiority of 3DDFA in 3D face alignment.

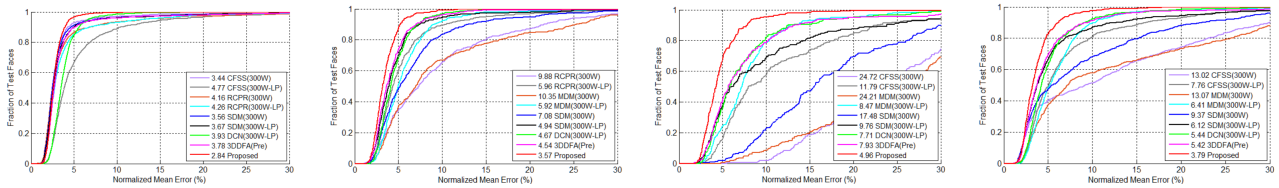
6.6.3 Medium Pose Face Alignment

As a face alignment approach to deal with full pose range, 3DDFA also shows competitive performance on the medium-pose 300W database, using the common protocol in [36]. The alignment accuracy is evaluated by the standard landmark mean error normalized by the inter-pupil distance (NME). For 3DDFA, we sample the 68 landmarks from the fitted 3D face and refine them with SDM to reduce the labelling bias. Table 3 shows that even in medium poses 3DDFA performs competitively, especially on the challenging set.



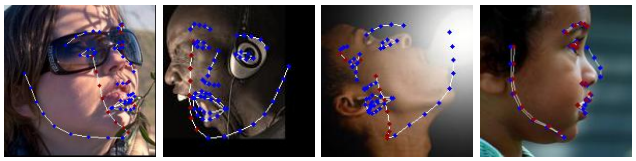
(a) $0°$ to $30°$ (b) $30°$ to $60°$ (c) $60°$ to $90°$ (d) Mean

Fig. 18. Comparisons of cumulative errors distribution (CED) curves on AFLW with yaw distributing at: (a) $[0°, 30°]$, (b) $[30°, 60°]$ and (c) $[60°, 90°]$. We further plot a mean CED curve (d) with a subset of 12,081 samples whose absolute yaw angles within each yaw iterval are 1/3 each. Only the top 6 methods are shown.



(a) $0°$ to $30°$ (b) $30°$ to $60°$ (c) $60°$ to $90°$ (d) Mean

Fig. 19. Comparisons of cumulative errors distribution (CED) curves on AFLW2000-3D with yaw distributing at: (a) $[0°, 30°]$, (b) $[30°, 60°]$ and (c) $[60°, 90°]$. We further plot a mean CED curve (d) with a subset of 696 samples whose absolute yaw angles within each yaw iterval are 1/3 each. Only the top 6 methods are shown.



(a) (b) (c) (d)

Fig. 20. Typical failure reasons of 3DDFA, including (a) complicated shadow and occlusion, (b) extreme pose and expression, (c) extreme illumination and (d) limited shape variations of 3DMM on nose.

TABLE 3
The NME(%) of face alignment results on 300W, with the first and the second best results highlighted.

Method	Common	Challenging	Full
TSPM [18]	8.22	18.33	10.20
ESR [10]	5.28	17.00	7.58
RCPR [42]	6.18	17.26	8.35
SDM [11]	5.57	15.40	7.50
LBF [35]	4.95	11.98	6.32
CFSS [36]	**4.73**	9.98	5.76
TCDCN [73]	**4.80**	**8.60**	**5.54**
3DDFA(Pre)	5.53	9.56	6.31
Proposed	5.09	**8.07**	**5.63**

6.6.4 Robustness to Initialization

The alignment performance can be greatly affected by the bounding boxes used for initialization. In this experiment, we initialize alignment methods with detected bounding boxes by FTF face detector [65] rather than the ground truth bounding boxes. We drop the bad boxes whose IOU with ground truth bounding boxes are less than 0.6 and generate the bounding boxes of undetected faces by random perturbation used in training. Table 4 shows the comparison results with the best two competitors DCN and SDM. Firstly, it can be seen that our method still outperforms others when initialized with face detectors. Besides, by comparing the performance drop brought by replacing bounding boxes, our method demonstrates best robustness to initialization.

TABLE 4
Alignment performance (NME) initialized by detected bounding boxes. The value in the brackets are the NME difference between results initialized by the detected and the ground truth bounding boxes

	AFLW			AFLW2000-3D		
	SDM	DCN	Ours	SDM	DCN	Ours
$[0, 30]$	5.09 (0.34)	5.31 (0.32)	4.24 (0.13)	4.11 (0.44)	4.34 (0.41)	3.00 (0.16)
$[30, 60]$	6.02 (0.47)	5.95 (0.48)	4.59 (0.21)	6.19 (1.25)	5.42 (0.75)	3.89 (0.32)
$[60, 90]$	10.13 (0.79)	8.13 (0.03)	5.32 (0.16)	12.03 (2.27)	8.72 (1.01)	5.55 (0.59)
Mean	7.08 (0.53)	6.47 (0.28)	4.72 (0.17)	7.44 (1.32)	6.16 (0.74)	4.15 (0.36)

7 Conclusions

Most of face alignment methods tend to fail in profile view since the self-occluded landmarks cannot be detected. Instead of the traditional landmark detection framework, this paper fits a dense 3D Morphable Model to achieve pose-free face alignment. By proposing two input features of PNCC and PAF, we cascade a couple of CNNs as a strong regressor to estimate model parameters. A novel OWPDC cost function is also proposed to consider the priority of parameters. To provide abundant samples for training, we propose a face profiling method to synthesize face appearances in profile views. Experiments show the state-of-the-art performance on AFLW, AFLW2000-3D and 300W.

8 Acknowledgments

This work was supported by the National Key Research and Development Plan (Grant No.2016YFC0801002), the Chinese National

Natural Science Foundation Projects #61473291, #61572501, #61502491, #61572536 and AuthenMetric R&D Funds. Zhen Lei is the corresponding author.

REFERENCES

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1701–1708. 1

[2] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D shape regression for real-time facial animation." *ACM Trans. Graph.*, vol. 32, no. 4, p. 41, 2013. 1, 2

[3] X. Xiong and F. De la Torre, "Global supervised descent method," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2664–2673. 1

[4] V. Bettadapura, "Face expression recognition and analysis: The state of the art," *Computer Science*, 2012. 1

[5] C.-Y. Yang, S. Liu, and M.-H. Yang, "Structured face hallucination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1099–1106. 1

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995. 1, 2

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 681–685, 2001. 1, 2, 4

[8] D. Cristinacce and T. F. Cootes, "Feature detection and tracking with constrained local models." in *BMVC*, vol. 17, 2006, pp. 929–938. 1, 2

[9] P. Dollár, P. Welinder, and P. Perona, "Cascaded pose regression," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1078–1085. 1, 2, 3, 4

[10] 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, pp. 2887–2894. 1, 2, 3, 6, 11, 12

[11] 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, pp. 532–539. 1, 2, 3, 4, 12

[12] 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, pp. 3476–3483. 1, 3, 4, 5

[13] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 94–108. 1, 3, 4

[14] Y. Zhou, W. Zhang, X. Tang, and H. Shum, "A Bayesian mixture model for multi-view face alignment," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 741–746. 1, 2, 3

[15] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 787–796. 1, 3, 7, 8

[16] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 1–16. 1

[17] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1944–1951. 1, 2, 3

[18] 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, pp. 2879–2886. 1, 2, 3, 7, 8, 11, 12

[19] S. Jaiswal, T. R. Almaev, and M. F. Valstar, "Guided unsupervised learning of mode specific models for facial point detection in the wild," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 370–377. 2, 7

[20] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 679–692. 2, 7

[21] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE, 2013, pp. 896–903. 2, 7

[22] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2144–2151. 2, 8

[23] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1063–1074, 2003. 2, 3

[24] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3D solution," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016. 2, 6, 11

[25] S. Romdhani, S. Gong, A. Psarrou *et al.*, "A multi-view nonlinear active shape model using kernel pca," in *BMVC*, vol. 10, 1999, pp. 483–492. 2

[26] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. 2

[27] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008. 2

[28] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "A comparative evaluation of active appearance model algorithms." in *BMVC*, vol. 98, 1998, pp. 680–689. 2

[29] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *Proceedings of IEEE International Conference on Computer Vision & Pattern Recognition (CVPR16), Las Vegas, NV, USA*, 2016. 2, 3, 11

[30] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011. 2

[31] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010. 2

[32] X. Hou, S. Z. Li, H. Zhang, and Q. Cheng, "Direct appearance models," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–828. 3, 6

[33] J. Saragih and R. Goecke, "A nonlinear discriminative approach to aam fitting," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8. 3, 6

[34] 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, pp. 2729–2736. 3

[35] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 FPS via regressing local binary features," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1685–1692. 3, 11, 12

[36] S. Zhu, C. Li, C. C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4998–5006. 3, 4, 8, 11, 12

[37] Y. Wu, T. Hassner, K. G. Kim, G. Medioni, and P. Natarajan, "Facial landmark detection with tweaked convolutional neural networks," *Computer Science*, 2015. 3, 4

[38] A. Jourabloo and X. Liu, "Large-pose face alignment via CNN-based dense 3D model fitting," in *Proceedings of IEEE International Conference on Computer Vision & Pattern Recognition (CVPR16), Las Vegas, NV, USA*, 2016. 3

[39] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor, "View-based active appearance models," *Image and vision computing*, vol. 20, no. 9, pp. 657–664, 2002. 3

[40] S. Z. Li, H. Zhang, Q. Cheng *et al.*, "Multi-view face alignment using direct appearance models," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 324–329. 3

[41] R. Gross, I. Matthews, and S. Baker, "Active appearance models with occlusion," *Image and Vision Computing*, vol. 24, no. 6, pp. 593–604, 2006. 3

[42] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1513–1520. 3, 11, 12

[43] L. Gu and T. Kanade, "3D alignment of face in a single image," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 1305–1312. 3

[44] S. Romdhani and T. Vetter, "Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," in *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, vol. 2. IEEE, 2005, pp. 986–993. 3, 7

[45] A. Jourabloo and X. Liu, "Pose-invariant 3D face alignment," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015. 3, 5, 10

[46] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 43, 2014. 3

[47] L. A. Jeni, J. F. Cohn, and T. Kanade, "Dense 3D face alignment from 2D videos in real-time," in *Automatic Face & Gesture Recognition, 2015. FG'15. 11th IEEE International Conference on*. IEEE, 2015. 3

[48] A. Jourabloo and X. Liu, "Pose-invariant face alignment via cnn-based dense 3d model fitting," *International Journal of Computer Vision*, pp. 1–17, 2017. 3

[49] A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Pose-invariant face alignment with a single cnn," in *In Proceeding of International Conference on Computer Vision*, Venice, Italy, October 2017. 3

[50] O. Aldrian and W. A. Smith, "Inverse rendering of faces with a 3D morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 5, pp. 1080–1093, 2013. 3, 8, 9

[51] T. Hassner, "Viewing real-world faces in 3D," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3607–3614. 3

[52] J. Roth, Y. Tong, and X. Liu, "Adaptive 3D face reconstruction from unconstrained photo collections," in *Proceedings of IEEE International Conference on Computer Vision & Pattern Recognition (CVPR16), Las Vegas, NV, USA*, 2016. 3

[53] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. IEEE, 2009, pp. 296–301. 3, 9

[54] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: a 3D facial expression database for visual computing," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 3, pp. 413–425, 2014. 3

[55] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 607–626, 2009. 4

[56] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005. 4

[57] Z. Liang, S. Ding, and L. Lin, "Unconstrained facial landmark localization with backbone-branches fully-convolutional networks," *Computer Science*, 2015. 4, 5

[58] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016. 4

[59] L. Spreeuwers, "Fast and accurate 3D face recognition," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 389–414, 2011. 5

[60] X. Zhu, J. Yan, D. Yi, Z. Lei, and S. Z. Li, "Discriminative 3D morphable model fitting," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. IEEE, 2015, pp. 1–8. 6

[61] J. Martens, "Deep learning via Hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 735–742. 6

[62] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 7, 11

[63] U. Prabhu, J. Heo, and M. Savvides, "Unconstrained pose-invariant face recognition using 3D generic elastic models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 10, pp. 1952–1961, 2011. 7

[64] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, "Pose-aware face recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4838–4846. 7

[65] P. Hu and D. Ramanan, "Finding tiny faces," *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017. 8, 12

[66] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 397–403. 8

[67] P. N. Belhumeur, D. W. Jacobs, D. 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, pp. 545–552. 8

[68] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 386–391. 8

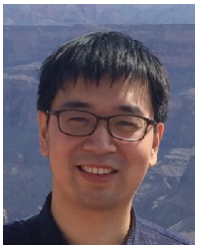
[69] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Second international conference on audio and video-based biometric person authentication*, vol. 964. Citeseer, 1999, pp. 965–966. 8

[70] M. Piotraschke and V. Blanz, "Automated 3D face reconstruction from multiple images using quality measures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3418–3427. 9

[71] J. Yan, Z. Lei, D. Yi, and S. Z. Li, "Learn to combine multiple hypotheses for accurate face alignment," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 392–396. 11

[72] F. Chen, F. Liu, and Q. Zhao, "Robust multi-view face alignment based on cascaded 2d/3d face shape regression," in *Chinese Conference on Biometric Recognition*, 2016, pp. 40–49. 11

[73] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 918–930, 2016. 12



Xiangyu Zhu received the BS degree in Sichuan University (SCU) in 2012, and the PhD degree from Institute of Automation, Chinese Academy of Sciences, in 2017, where he is currently an assistant professor. His research interests include pattern recognition and computer vision, in particular, image processing, 3D face model, face alignment and face recognition.

Xiaoming Liu is an Assistant Professor at the Department of Computer Science and Engineering of Michigan State University. He received the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2004. Before joining MSU in Fall 2012, he was a research scientist at General Electric Global Research. His main research areas are human face recognition, biometrics, human computer interface, object tracking/recognition, online learning, computer vision, and pattern recognition.

Zhen Lei received the BS degree in automation from the University of Science and Technology of China, in 2005, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences,in 2010, where he is currently an associate professor. His research interests are in computer vision, pattern recognition, image processing, and face recognition in particular.

Stan Z. Li received his B.Eng from Hunan University, China, M.Eng from National University of Defense Technology, China, and PhD degree from Surrey University, UK. He is currently a professor and the director of Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences (CASIA). He worked at Microsoft Research Asia as a researcher from 2000 to 2004. Prior to that, he was an associate professor at Nanyang Technological University, Singapore. He was elevated to IEEE Fellow for his contributions to the fields of face recognition, pattern recognition and computer vision. His research interest includes pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance.