Large-pose Face Alignment via CNN-based Dense 3D Model Fitting

Amin Jourabloo, Xiaoming Liu
Department of Computer Science and Engineering
Michigan State University, East Lansing MI 48824
{jourablo, liuxm}@msu.edu

Abstract

Large-pose face alignment is a very challenging problem in computer vision, which is used as a prerequisite for many important vision tasks, e.g, face recognition and 3D face reconstruction. Recently, there have been a few attempts to solve this problem, but still more research is needed to achieve highly accurate results. In this paper, we propose a face alignment method for large-pose face images, by combining the powerful cascaded CNN regressor method and 3DMM. We formulate the face alignment as a 3DMM fitting problem, where the camera projection matrix and 3D shape parameters are estimated by a cascade of CNN-based regressors. The dense 3D shape allows us to design pose-invariant appearance features for effective CNN learning. Extensive experiments are conducted on the challenging databases (AFLW and AFW), with comparison to the state of the art.

1. Introduction

Face alignment is the process of aligning a face image and detecting specific fiducial points, such as eye corners, nose tip, etc. Improving the face alignment accuracy is beneficial for many computer vision tasks related to facial analysis, because it is used as a prerequisite for these tasks, e.g., face recognition [28], 3D face reconstruction [20, 21] and face de-identification [10].

Given its importance, face alignment has been an active research topic since 1990s [29], with the well-known Active Shape Model [5] and Active Appearance Model (AAM) [15, 13]. Recently, face alignment works are very popular in top vision venues, as demonstrated by the progress in Constrained Local Model based approaches [5, 22], AAM-based approaches [15, 13, 14] and regression-based approaches [27, 4, 33]. Despite the fruitful prior work and continuous progress of face alignment (e.g., the latest impressive iBUG results [26]), face alignment for large-pose faces is still very challenging and there is only a few published work in this direction, as summarized in Table 1.

Figure 1. The proposed method estimates landmarks for large-pose faces by fitting a dense 3D shape. From left to right: initial landmarks, fitted 3D dense shape, estimated landmarks with visibility. The green/red/yellow dots in the right column show the visible/invisible/cheek landmarks, respectively.

Therefore, this is a clear research gap that needs to be addressed, which is exactly the focus of this work.

To tackle large-pose face alignment, our technical approach is driven by the inherent challenges associated with this problem. First of all, faces have different numbers of visible landmarks under pose variation, and the spatial distribution of the landmarks is highly pose dependent. This presents challenges for existing face alignment approaches since most are based on 2D shape models, which inherently have difficulty in modeling the 3D out-of-plane deformation. In contrast, given the fact that a face image is a projection of a 3D face, we propose to use a dense 3D Morphable Model (3DMM) and the projection matrix as the *representation* of a 2D face image. Therefore, face alignment amounts to estimating this representation, i.e., performing the 3DMM fitting to a face image with *arbitrary* poses.

Second, the typical analysis-by-synthesis-based optimization approach for 3DMM fitting is inefficient and also assumes the 2D landmarks are provided either manually or with a separate face alignment method, which conflicts with the goal of our work. This motivates us to employ the powerful cascaded regressor approach to learn the mapping between a 2D face image and its representation. Since the representation is composed of 3D parameters, the mapping

Table 1. The comparison of large-pose face alignment methods.

Method	Dense 3D model fitting	Visibility	Database	Pose range	Training face #	Testing face #	Landmarks #	Estimation errors
RCPR [1]	No	Yes	COFW	frontal w. occlu.	1,345	507	19	8.5
TSPM [37]	No	No	AFW	all poses	2,118	468	6	11.1
CDM [31]	No	No	AFW	all poses	1,300	468	6	9.1
TCDCN [34]	No	No	AFLW, AFW	$[-60°, 60°]$	10,000	3,000; ∼313	5	8.0; 8.2
PIFA [9]	No	Yes	AFLW, AFW	all poses	3,901	1,299; 468	21, 6	6.5; 8.6
Proposed method	Yes	Yes	AFLW, AFW	all poses	3,901	1,299; 468	34, 6	4.7; 7.4

Table 2. The comparison of most recent 3D face model fitting methods.

Method	Integrated 2D landmark	# of 2D landmarks	Testing database	Pose range	3D bases	Method
BMVC 2015 [19]	No	68	Basel	$[-30°, 30°]$	Basel bases	Adaptive contour fitting
FG 2015 [8]	No	77 to 1024	BU-4DFE; BP-4DS; videos	$[-60°, 60°]$	Bases from BU-4DFE & BP-4DS	Cascaded regressor; EM
FG 2015 [38]	Yes	-	FRGC	Frontal	Basel bases	Cascaded regressor
Proposed method	Yes	-	AFW; AFLW	All poses	Basel bases	3D cascaded regressor

is likely to be more complicated than the cascaded regressor in 2D face alignment [4]. As a result, we propose to use Convolutional Neural Networks (CNN) as the regressor in the cascaded framework, to learn the mapping. While prior work on CNN for face alignment estimate no more than 6 2D landmarks per image, our cascaded CNN can estimate a substantially larger number (34) of 2D and 3D landmarks. Further, using landmark marching [36], our algorithm can adaptively adjust the 3D landmarks during the fitting, so that the cheek landmarks can contribute to the fitting.

Third, conventional 2D face alignment approaches are often driven by the local feature patch around each estimated 2D landmark. Even at the ground truth landmark, such as the outer eye corner, it is hard to make sure that the local patches from faces at various poses cover the exactly the same part of facial skin *anatomically*, which poses additional challenge for the learning algorithm to associate a unified pattern with the ground truth landmark. Fortunately, in our work, we can use the dense 3D face model as an oracle to build enhanced feature correspondence across various poses and expressions. Therefore, we propose two novel pose-invariant local features, as the input layer for CNN learning. We also utilize person-specific surface normals to estimate the visibility of each landmark.

These algorithm designs collectively lead to the proposed large-pose face alignment algorithm. We conduct extensive experiments to demonstrate its capability in aligning faces across poses, in comparison with the state of the art.

We summarize the main contributions of this work as:

◇ Large-pose face alignment by fitting a dense 3DMM.

◇ The cascaded CNN-based 3D face model fitting algorithm that is applicable to all poses, with integrated landmark marching.

◇ Dense 3D face-enabled pose-invariant local features.

2. Prior Work

We review papers in three areas related to the proposed method: large-pose face alignment, face alignment via deep learning, and 3D face model fitting to a single image.

Large-pose face alignment The methods of [31, 37, 7] combines face detection, pose estimation and face alignment. By using a 3D shape model with optimized mixture of parts, [31] can be applied to faces with a large range of poses. In [30], a face alignment method based on cascade regressors is proposed to handle invisible landmarks. Each stage is composed of two regressors for estimating the probability of landmark visibility and the location of landmarks. This method is applied to profile view faces of FERET database [18]. As a 2D landmark-based approach, it cannot estimate 3D face poses. Occlusion-invariant face alignment, such as RCPR [1], may also be applied to handle large poses since non-frontal faces are one type of occlusions. [25] is a very recent work that performs 3D landmark estimation via regressors. However, it only tests on synthesized face images up to $\sim 50°$ yaw. The most relevant prior work is [9], which aligns faces of arbitrary poses with the assistant of a sparse 3D point distribution model. The model parameter and projection matrix are estimated by the cascade of linear or non-linear regressors. We extend [9] in a number of aspects, including *fitting a dense 3D morphable model, employing the powerful CNN as the regressor, using 3D-enabled features, and estimating cheek landmarks*. Table 1 compares the large-pose face alignment methods.

Face alignment via deep learning With the continuous success of deep learning in vision, researchers start to apply deep learning to face alignment. Sun et al. [24] proposed a three-stage face alignment algorithm with CNN. At the first stage, three CNNs are applied to different face parts to estimate positions of different landmarks, whose averages are regarded as the first stage results. At the next two stages, by using local patches with different sizes around each landmark, the landmark positions are refined. Similar face alignment algorithms based on multi-stage CNNs are further developed by Zhou et al. [35] and CFAN [32]. TCDCN [34] uses one-stage CNN to estimates positions

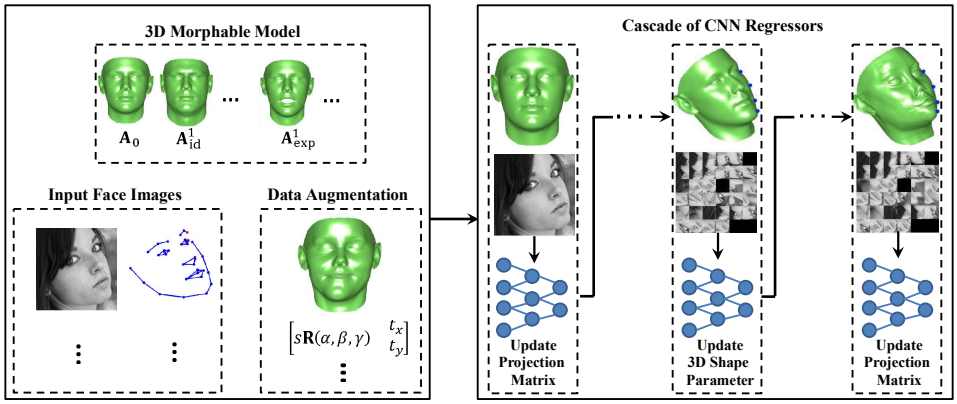

Figure 2. The overall process of the proposed method.

of five landmarks given a face image. The commonality among all these prior works is that they only estimate 2D landmark locations and the number of landmarks is limited to 6. In comparison, our proposed method *employs CNN to estimate 3D landmarks, as part of the 3D surface reconstruction*. As a result, the number of estimated landmarks is bounded by the number of 3D vertexes, although the evaluation is conducted for 34 landmarks.

3D face model fitting Table 2 shows the comparison of most recent 3D face model fitting methods to a single image. Almost all prior works assume that the 2D landmarks of the input face image is either manually labeled or estimated via a face alignment method. The authors in [19] aim to make sure that the location of 2D contour landmarks is consistent with 3D face shape. In [38], a 3D face model fitting method based on the similarity of frontal view face images is proposed. In contrast, our proposed method is the first approach to *integrate 2D landmark estimation as part of the 3D face model fitting for large poses*. Furthermore, all prior 3D face model fitting works process face images with up to 60° yaw while our method can handle all view angles.

3. Unconstrained 3D Face Alignment

The core of our proposed 3D face alignment method is the ability to fit a dense 3D Morphable Model to a 2D face image with arbitrary poses. The unknown parameters of fitting, the 3D shape parameters and the projection matrix parameters, are sequentially estimated through a cascade of CNN-based regressors. By employing the dense 3D shape model, we enjoy the benefits of being able to estimate the locations of cheek landmarks, to use person-specific 3D surface normals, and extract pose-invariant local feature representation. Figure 2 shows the overall process of the proposed method.

3.1. 3D Morphable Model

To represent a dense 3D shape of an individual's face, we use 3D Morphable Model (3DMM),

$$\mathbf{A} = \mathbf{A}_0 + \sum_{i=1}^{N_{id}} p_{id}^i \mathbf{A}_{id}^i + \sum_{i=1}^{N_{exp}} p_{exp}^i \mathbf{A}_{exp}^i, \tag{1}$$

where $\mathbf{A}$ is the 3D shape matrix, $\mathbf{A}_0$ is the mean shape, $\mathbf{A}_{id}^i$ is the ith identity basis, $\mathbf{A}_{exp}^i$ is the ith expression basis, p_{id}^i is the ith identity coefficient, and p_{exp}^i is the ith expression coefficient. The collection of both coefficients is denoted as the shape parameter of a 3D face, $\mathbf{p} = (\mathbf{p}_{id}^\intercal, \mathbf{p}_{exp}^\intercal)^\intercal$. We use the Basel 3D face model as the identity bases [16] and the face wearhouse as the expression bases [3]. The 3D shape $\mathbf{A}$, along with $\mathbf{A}_0$, $\mathbf{A}_{id}^i$, and $\mathbf{A}_{exp}^i$, is a $3 \times Q$ matrix which contains x, y and z coordinates of Q vertexes on the 3D face surface,

$$\mathbf{A} = \begin{pmatrix} x_1 & x_2 & \cdots & x_Q \\ y_1 & y_2 & \cdots & y_Q \\ z_1 & z_2 & \cdots & z_Q \end{pmatrix}. \tag{2}$$

Any 3D face model will be projected onto a 2D image where the face shape may be represented as a sparse set of N landmarks, on the facial fiducial points. We denote x and y coordinates of these 2D landmarks as a matrix $\mathbf{U}$,

$$\mathbf{U} = \begin{pmatrix} u_1 & u_2 & \cdots & u_N \\ v_1 & v_2 & \cdots & v_N \end{pmatrix}. \tag{3}$$

The relationship between the 3D shape $\mathbf{A}$ and $2D$ landmarks $\mathbf{U}$ can be described by using the weak perspective projection, i.e.,

$$\mathbf{U} = s\mathbf{R}\mathbf{A}(:, \mathbf{d}) + \mathbf{t}, \tag{4}$$

where s is a scale parameter, $\mathbf{R}$ is the first two rows of a 3×3 rotation matrix controlled by three rotation angles α, β, and γ, $\mathbf{t}$ is a translation parameter composed of t_x and t_y, $\mathbf{d}$ is a N-dim index vector indicating the indexes of semantically meaningful 3D vertexes that correspond to 2D landmarks. By collecting all parameters related to this projection, we form a projection vector $\mathbf{m} = (s, \alpha, \beta, \gamma, t_x, t_y)^\intercal$.

Algorithm 1: Landmark marching $g(\mathbf{A}, \mathbf{m})$.

Data: Estimated 3D face $\mathbf{A}$ and projection parameter $\mathbf{m}$
Result: Index vector $\mathbf{d}$
/* Rotate $\mathbf{A}$ by the estimated α, β */
$\hat{\mathbf{A}} = \mathbf{R}(\alpha, \beta, 0)\mathbf{A}$
if $0° < \beta < 70°$ **then**
  foreach $i = 1, \cdots, 4$ **do**
    $V_{\text{cheek}}(i) = \arg\max_{id}(\hat{\mathbf{A}}(1, \text{Path}_{\text{cheek}}(i)))$
if $-70° < \beta < 0°$ **then**
  foreach $i = 5, \cdots, 8$ **do**
    $V_{\text{cheek}}(i) = \arg\min_{id}(\hat{\mathbf{A}}(1, \text{Path}_{\text{cheek}}(i)))$
Update 8 elements of $\mathbf{d}$ with V_{cheek}.

At this point, we can represent any 2D face shape as the projection of a 3D face shape. In other words, the projection parameter $\mathbf{m}$ and shape parameter $\mathbf{p}$ can uniquely represent a 2D face shape. Therefore, the face alignment problem amounts to estimating $\mathbf{m}$ and $\mathbf{p}$, given a face image.

Cheek landmarks correspondence The projection relationship in Eqn. 4 is correct for frontal-view faces, given a constant index vector $\mathbf{d}$. However, as soon as a face turns to the side view, the original 3D landmarks on the cheek become invisible on the 2D image. Yet most 2D face alignment algorithms still detect 2D landmarks on the contour of the cheek, termed "cheek landmarks". Therefore, in order to still maintain the correspondences as Eqn. 4, it is best to estimate the 3D vertexes that match with these cheek landmarks. A few prior works have proposed various approaches to handle this [19, 36, 2]. We leverage the landmark marching method proposed in [36].

Specifically, we define a set of *paths* each storing the indexes of vertexes that are not only the most closest ones to the original 3D cheek landmarks, but also on the contour of the 3D face as it turns. Given a non-frontal 3D face $\mathbf{A}$, we rotate $\mathbf{A}$ by using the α and β angles (pitch and yaw angles), and search for a vertex in each defined path which has the maximum (minimum) x coordinate, i.e., the boundary vertex on the right (left) cheek. These searched vertexes will be the new 3D landmarks that correspond to the 2D cheek landmarks. We will then update relevant elements of $\mathbf{d}$ to make sure these vertexes are selected in the projection of Eqn. 4. This landmark marching process is summarized in Algorithm 1 as a function $\mathbf{d} \leftarrow g(\mathbf{A}, \mathbf{m})$. Note that when the face is almost of profile view ($|\beta| > 70°$), we do not apply landmark marching since the marched landmarks would overlap with the existing 2D landmarks on the middle of nose and mouth.

3.2. Data Augmentation

Given that the projection parameter $\mathbf{m}$ and shape parameter $\mathbf{p}$ are the representation of a face image, we should have a collection of face images with ground truth $\mathbf{m}$ and $\mathbf{p}$ so that the learning algorithm can be applied. However, for most existing face alignment databases, only 2D landmark locations and sometimes the visibilities of landmarks are manually labeled, with no associated 3D information such as $\mathbf{m}$ and $\mathbf{p}$. In order to make the learning possible, we propose a data augmentation process for 2D face images, with the goal of estimating its $\mathbf{m}$ and $\mathbf{p}$ representation.

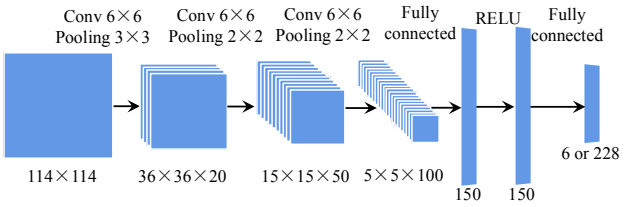

Figure 3. Architecture of CNN used in each stage of the proposed method.

Specifically, given the labeled visible $2D$ landmarks $\mathbf{U}$ and the landmark visibilities $\mathbf{V}$, we use the following objective function to estimate $\mathbf{m}$ and $\mathbf{p}$,

$$J(\mathbf{m}, \mathbf{p}) = ||(s\mathbf{R}\mathbf{A}(:, g(\mathbf{A}, \mathbf{m})) + \mathbf{t} - \mathbf{U}) \odot \mathbf{V}||_F^2, \quad (5)$$

which basically minimizes the difference between the projection of 3D landmarks and the 2D labeled landmarks. Note that although the landmark marching $g(:, :)$ can make cheek landmarks "visible" for non-profile views, the visibility $\mathbf{V}$ is useful to avoid invisible landmarks such as outer eye corners and half of the face at the profile view being part of the optimization.

To minimize this objective function, we alternate the minimization w.r.t. $\mathbf{m}$ and $\mathbf{p}$ at each iteration. We initialize the 3D shape parameter $\mathbf{p} = \mathbf{0}$ and estimate $\mathbf{m}$ first. At each iteration, the $g(\mathbf{A}, \mathbf{m})$ is a constant computed using the currently estimated $\mathbf{m}$ and $\mathbf{p}$.

3.3. Cascaded CNN Coupled-Regressor

Given a set of N_d training face images and their augmented (a.k.a. ground truth in this context) $\mathbf{m}$ and $\mathbf{p}$ representation, we are interested in learning a mapping function that is able to predict $\mathbf{m}$ and $\mathbf{p}$ from the appearance of a face image. Clearly this is a complicated non-linear mapping function. Given the success of CNN in vision tasks such as pose estimation [17], face detection [12], and face alignment [34], we decide to marry the CNN with the cascade regressor framework by learning a series of CNN-based regressors to alternate the estimation of $\mathbf{m}$ and $\mathbf{p}$. To the best of our knowledge, this is the first time CNN is used in 3D face alignment, with the estimation of over 10 landmarks.

In addition to the ground truth $\mathbf{m}$ and $\mathbf{p}$, we also assume each training image has the initial values of these two parameters, denoted as $\mathbf{m}^0$ and $\mathbf{p}^0$. Thus, at the stage k of the cascaded CNN, we can learn a CNN to estimate the desired

update of the projection parameter,

$$\Theta_m^k = \arg\min_{\Theta_m^k} \sum_{i=1}^{N_d} ||\Delta \mathbf{m}_i^k - \text{CNN}_m^k(\mathbf{I}_i, \mathbf{U}_i, \mathbf{v}_i^{k-1}; \Theta_m^k)||^2, \tag{6}$$

where the true projection update is the difference between the current projection parameter and the ground truth, i.e., $\Delta \mathbf{m}_i^k = \mathbf{m}_i - \mathbf{m}_i^{k-1}$, $\mathbf{U}_i$ is current estimated $2D$ landmarks, computed via Eqn. 4 based on $\mathbf{m}_i^{k-1}$ and $\mathbf{d}_i^{k-1}$, and $\mathbf{v}_i^{k-1}$ is estimated landmark visibility at stage $k-1$.

Similarly another CNN regressor can be learned to estimate the updates of the shape parameter,

$$\Theta_p^k = \arg\min_{\Theta_p^k} \sum_{i=1}^{N_d} ||\Delta \mathbf{p}_i^k - \text{CNN}_p^k(\mathbf{I}_i, \mathbf{U}_i, \mathbf{v}_i^k; \Theta_p^k)||^2. \tag{7}$$

Note that $\mathbf{U}_i$ will be re-computed via Eqn. 4, based on the updated $\mathbf{m}_i^k$ and $\mathbf{d}_i^k$ by CNN_m.

We use a six-stage cascaded CNN, including CNN_m^1, CNN_m^2, CNN_p^3, CNN_m^4, CNN_p^5, and CNN_m^6. At the first stage, the input layer of CNN_m^1 is the entire face region cropped by the initial bounding box, with the goal of roughly estimating the pose of the face. The input for the second to sixth stages is a 114×114 image that contains an array of 19×19 pose-invariant feature patches, extracted from the current estimated 2D landmarks $\mathbf{U}_i$. In our implementation, since we have $N = 34$ landmarks, the last two patches of 114×114 image are filled with zero. Similarly, for invisible 2D landmarks, their corresponding patches will be filled with zeros as well. These concatenated feature patches encode sufficient information about the local appearance around the current 2D landmarks, which drives the CNN to optimize the parameters Θ_m^k or Θ_p^k. This method can be extended to use a larger number of landmarks and hence a more accurate dense 3D model can be estimated.

Note that since landmark marching is used, the estimated 2D landmarks $\mathbf{U}_i$ include the projection of marched 3D landmarks, i.e., 2D cheek landmarks. As a result, the appearance features around these cheek landmarks are part of the input to CNN as well. This is in sharp contrast to [9] where no cheek landmarks participate the regressor learning. Effectively, these additional cheek landmarks serve as constraints to affect how the facial silhouettes at various poses should look like, which is basically the shape of the 3D face surface.

We used rectified linear unit (ReLU) [6] as the activation function which enables CNN to achieve the best performance without unsupervised pre-training. We use the same CNN architecture (Fig. 3) for all six stages.

3.4. Visibility and 2D Appearance Features

One notable advantage of employing a dense 3D shape model is that more advanced 2D features, which might be only possible because of the 3D model, can be extracted and contribute to the cascaded CNN learning. In this work, these 2D features refer to the 2D landmark visibility and the appearance patch around each 2D landmark.

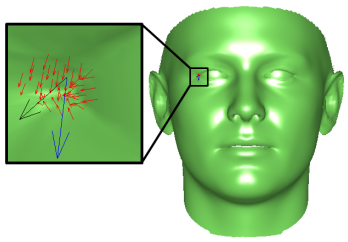

Figure 4. The person-specific 3D surface normal as the average of normals around a 3D landmark (black arrow). Notice the relatively noisy surface normal of the 3D "left eye corner" landmark (blue arrow).

In order to compute the visibility of each 2D landmark, we leverage the basic idea of examining whether the 3D surface normal of the corresponding 3D landmark is pointing to the camera or not, under the current camera projection matrix [9]. Instead of using the average 3D surface normal for all humans, we extend it by using person-specific 3D surface normal. Specifically, given the current estimated 3D shape $\mathbf{A}$, we compute the 3D surface normals for a set of sparse vertexes around the 3D landmark of interest, and the average of these 3D normals is denoted as $\vec{\mathbf{N}}$. Figure 4 shows the advantage of using the average 3D surface normal. Given $\vec{\mathbf{N}}$, we compute $\mathbf{v} = \vec{\mathbf{N}}^\intercal \cdot (\mathbf{R}_1 \times \mathbf{R}_2)$, where $\mathbf{R}_1$ and $\mathbf{R}_2$ are the first two rows of $\mathbf{R}$. If $\mathbf{v}$ is positive, the 2D landmark is considered as visible and its 2D appearance feature will be part of the input for CNN. Otherwise, it is invisible and the corresponding feature will be zero for CNN. Note that this method does not estimate occlusion due to other objects such as hairs.

In addition to visibility estimation, a 3D shape model can also contribute in generating advanced appearance features as the input layer for CNN. Specifically, we aim to extract a pose-invariant appearance patch around each estimated 2D landmark, and the array of these patches will form the input layer. We now describe two proposed approaches to extract an appearance feature, i.e., a 19×19 patch, for the nth 2D landmark.

Piecewise affine-warped feature (PAWF): Feature correspondence is always very important for any visual learning, as evident by the importance of eye-based rectification to face recognition [23]. Yet, due to the fact that a 2D face is a projection of 3D surface with an arbitrary view angle, it is hard to make sure that a local patch extracted from this 2D image corresponds to the patch from another 2D image, even both patches are centered at the ground truth locations of the same nth 2D landmark. Here, "correspond" means that the patches cover the exactly same local region of faces anatomically. However, with a dense 3D shape model in

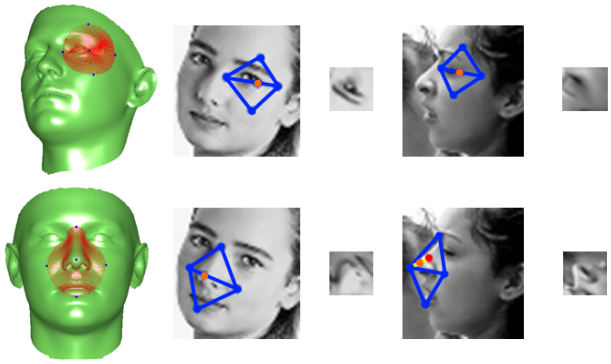

Figure 5. Examples of extracting PAWF feature. When one of the four neighborhood points (red point in the bottom-right) is invisible, it connects to the 2D landmark, extends the same distance further, and generate a new neighborhood point. This helps to include the background context around the nose.

hand, we may extract local patches across different subjects and poses with anatomical correspondence.

In the offline learning stage, we first search for T vertexes on the mean 3D shape $\mathbf{A}_0$ that are the most closest to the nth landmark. Second, we rotate the T vertexes such that the 3D surface normal of the nth landmark points toward the camera. Third, among the T vertexes we find four "neighborhood vertexes", which have the minimum and maximum x and y coordinates, and denote the four vertex IDs as a 4-dim vector $\mathbf{d}_p^{(n)}$.

During the CNN learning, for the nth landmark of ith image, we project the four neighborhood vertexes onto the ith image and obtain four neighborhood points, $\mathbf{U}_i^{(n)} = s\mathbf{R}\mathbf{A}(:, \mathbf{d}_p^{(n)}) + \mathbf{t}$, based on the current estimated projection parameter $\mathbf{m}$. Across all 2D face images, $\mathbf{U}_i^{(n)}$ correspond to the same face vertexes anatomically. Therefore, we warp the imagery content within these neighborhood points to a 19×19 patch by using the piecewise affine transformation.

This novel feature representation can be well extracted in most cases, except for cases such as the nose tip at the profile view, where one of the two scenarios could happen. One is the projection of the nth landmark is outside the region specified by the neighborhood points. The other is that one of neighborhood points is invisible. When these happen, we change the location of the invisible point by using its relative distance to the projected landmark location, as shown in Fig. 5.

Direct 3D projected feature (D3PF): Both D3PF and PAWF start with the T vertexes surrounding the nth 3D landmark. Instead of finding four neighborhood vertexes as in PAWF, D3PF put a 19×19 grid covering the T vertexes, and store the vertexes of the grid points in $\mathbf{d}_d^{(n)}$. Similar to PAWF, we can now project the set of 3D vertexes $\mathbf{A}(:, \mathbf{d}_d^{(n)})$ to the 2D image and extract a 19×19 patch via bilinear-interpolation, as shown in Fig. 6. We also estimate

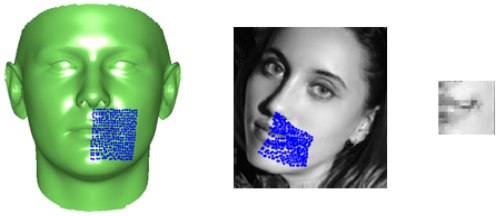

Figure 6. Example of extracting D3PF feature.

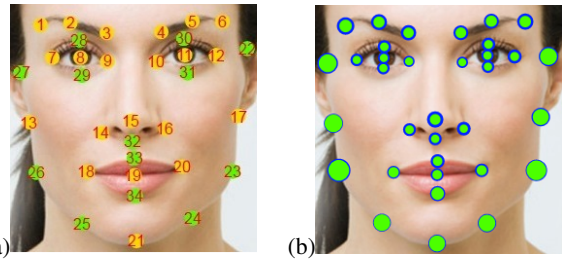

Figure 7. (a) AFLW original (yellow) and added landmarks (green), (b) Comparison of mean NME of each landmark for RCPR (blue) and proposed method (green). The radius of circles is determined by the mean NME multipled with the face box size.

the visibility of these 3D vertexes via their surface normals, and zero will be placed in the patch for invisible vertexes. For D3PF, every pixel in the patch will be corresponding to the same pixel in the patches of other images, while for PAWF, this is true only for the four neighborhood points.

4. Experimental Results

Databases Given that this work focus on large-pose face alignment, we choose two publicly available face datasets with labeled landmarks and a large range of poses.

AFLW database [11] is a large face dataset of 25K face images. Each image is manually labeled with up to 21 landmarks, with a visibility label for each landmark. In [9], a subset of AFLW is selected to have a balanced distribution of yaw angles, including $3,901$ images for training and $1,299$ images for testing. We use the same subset and manually label 13 additional landmarks for all $5,200$ images. The definition of original landmarks and added landmarks is shown in Fig. 7(a). Using ground truth landmarks of each image, we find the tightest bounding box, expand it by 10% of its size, and add 10% noise to the top-left corner, width and height of the bounding box. These randomly generated bounding boxes mimic the imprecise face detection window and will be used for both training and testing.

AFW dataset [37] contains 468 faces in 205 images. Each face image is manually labeled with up to 6 landmarks and has a visibility label for each landmark. For each face image a detected bounding box is provided. Given the small number of images, we only use this dataset for testing.

We use the $N_{id} = 199$ bases of Basel Face Model [16] for representing identity variation and the $N_{exp} = 29$ bases of face wearhouse [3] for representing expression variation. In total, there are 228 bases representing 3D face shapes

Table 3. NME (%) of the proposed method with different features.

PAWF + Cheek Landmarks	D3PF + Cheek Landmarks	PAWF	Extracted Patch
4.72	5.02	5.19	5.51

with $53,215$ vertexes.

Baseline selection We select the most recent large-pose face alignment methods for comparing with the proposed method, according to Table 1. We compare the proposed method with PIFA [9] and RCPR [1] on AFLW, and with PIFA [9], CDM [31] and TSPM [37] on AFW.

Parameter setting For the proposed method, the learning rate of CNN is constant at 0.0001 during training. We use ten epochs for training each CNN. For RCPR, we use the parameters reported in its paper, with 100 iterations and 15 boosted regressors. For PIFA, we use 200 iterations and 5 boosted regressors. For PAWF and D3PF, at the second stage T is $5,000$, and $3,000$ for the other stages. According to our empirical evaluation, six stages of CNN are sufficient for convergence of fitting process.

Evaluation metrics We use two conventional metrics for measuring the error of up to 34 landmarks. For AFLW dataset, we use the mean error of visible landmarks normalized by the bounding box size (NME) [9]. The eye-to-eye distance is not used in NME since it is not well defined in large poses such as profile. For AFW dataset, we use the Mean Average Pixel Error (MAPE) [31].

Feature extraction methods To show the advantages of the proposed features, Table 3 compares the accuracy of the proposed method on AFLW, with various feature presentation (i.e., the input layer for CNN^2 to CNN^6). The "Extracted Patch" refers to extracting a constant size (19×19) patch from a face image normalized using the bounding box, which serves as a baseline feature. For the feature "+Cheek Landmarks", additional up to four 19×19 patches of the contour landmarks, which are invisible for non-frontal faces, will be replaced with patches of the cheek landmarks, and used in the input layer of CNN learning. The PAWF feature can achieve a higher accuracy than the D3PF. By comparing Column 1 and 3 of Table 3, it shows that extracting features from cheek landmarks are very effective in acting as additional visual cues for the cascaded CNN regressors. The combination of using the cheek landmarks and extracting PAWE feature achieves the highest accuracy, which will be used in the remaining experiments.

CNN is known for requiring a large training set, while the AFLW training set is certainly small from CNN's perspective. However, our CNN-based regressor is still able to learn and align well on unseen images. We attribute this fact to the effective appearance features proposed in this work, i.e., we hypothesize that the good feature correspondence reduces CNN's demand for massive training data.

Experiments on AFLW dataset We compare the proposed method with the two most related methods for aligning faces with arbitrary poses. We use the source code of RCPR for performing training and testing. Similarly for PIFA, we use the source code to train on the AFLW train set with 13 more landmarks. The accuracy of the three methods are shown in Table 4. The proposed method can achieve better results than the two baselines. The error comparison for each landmark is shown in Fig. 7(b). As expected, the contour landmarks have higher errors and the proposed method has lower errors than RCPR across all of the landmarks.

Table 4. The NME (%) of three methods on AFLW.

Proposed method	PIFA	RCPR
4.72	8.04	6.26

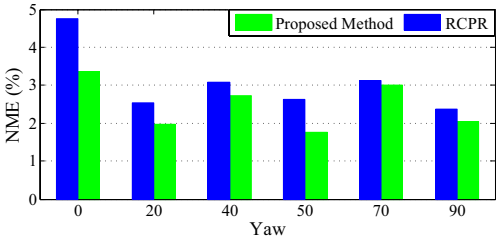

Figure 8. Comparison of NME for each pose.

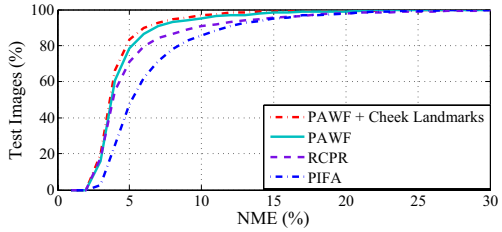

Figure 9. The comparison of CED for different methods.

Table 5. The MAPE of four methods on AFW.

Proposed method (PAWF)	Proposed method (D3PF)	PIFA	CDM	TSPM
7.43	7.83	8.61	9.13	11.09

By using the ground truth landmark locations of the test images, we divide the test set images to six subsets according to the estimated yaw angle of each image. Fig. 8 compares the proposed method with RCPR. The proposed method can achieve better results across different poses, and more importantly, is more robust or has less variation across poses. For the detailed comparison on the NME distribution, the cumulative errors distribution (CED) diagrams of various methods are shown in Fig. 9. The improvement seems to be over all NME values, and is especially larger around lower NMEs ($\leq 8\%$).

Expriments on AFW dataset The AFW dataset contains faces of all pose ranges with 6 landmarks. We report the MAPE for five methods in Table 5. For PIFA, CDM and TSPM, we use the reported errors in their papers. Again we see the consistent improvement of our proposed method (with two feature types) over the baseline methods.

Qualitative results Some examples of alignment results

Figure 10. The results of the proposed method on AFLW and AFW. The green/red/yellow dots show the visible/invisible/cheek landmarks, respectively. First row: initial landmarks for AFLW, Second: estimated 3D dense shapes, Third: estimated landmarks, Forth and Fifth: estimated landmarks for AFLW, Sixth: estimated landmarks for AFW.

Figure 11. The result of the proposed method across stages, with the extracted features (1st row) and alignment results (2nd row).

for the proposed method on AFLW and AFW datasets are shown in Fig. 10. The result of the proposed method at each stage is shown in Fig. 11. Note the changes of the landmark position and visibility (the top-right patch) over stages.

Time complexity The speeds of PAWF and D3PF methods are 0.6 and 0.26 FPS respectively, with unoptimized Matlab implementation. We believe this can be substantially improved with C coding and parallel feature extraction.

5. Conclusions

We proposed a method to fit a 3D dense shape to a face image with large poses by combining cascade CNN regressors and the 3D Morphable Model (3DMM). We proposed two types of pose invariant features for boosting the accuracy of face alignment. Also, we estimate the location of landmarks on the cheek, which also drives the 3D face model fitting. Finally, we achieve the state-of-the-art performance on two challenging face databases with larger poses.

References

[1] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, 2013.

[2] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG)*, 33(4):43, 2014.

[3] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: a 3D facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graphics*, 20(3):413–425, 2014.

[4] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *Int. J. Comput. Vision*, 107(2):177–190, 2014.

[5] T. Cootes, C. Taylor, and A. Lanitis. Active shape models: Evaluation of a multi-resolution method for improving image search. In *BMVC*, volume 1, pages 327–336, 1994.

[6] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proc. Artificial Intelligence and Statistics (AISTATS)*, pages 315–323, 2011.

[7] G.-S. Hsu, K.-H. Chang, and S.-C. Huang. Regressive tree structured model for facial landmark localization. In *ICCV*, pages 3855–3861, 2015.

[8] L. A. Jeni, J. F. Cohn, and T. Kanade. Dense 3D face alignment from 2d videos in real-time. In *FG*, 2015.

[9] A. Jourabloo and X. Liu. Pose-invariant 3D face alignment. In *ICCV*, 2015.

[10] A. Jourabloo, X. Yin, and X. Liu. Attribute preserved face de-identification. In *ICB*, 2015.

[11] 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 *ICCVW*, pages 2144–2151, 2011.

[12] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *CVPR*, pages 5325–5334, 2015.

[13] X. Liu. Discriminative face alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):1941–1954, 2009.

[14] X. Liu. Video-based face model fitting using adaptive active appearance model. *J. Image Vision Computing*, 28(7):1162–1172, 2010.

[15] I. Matthews and S. Baker. Active appearance models revisited. *Int. J. Comput. Vision*, 60(2):135–164, 2004.

[16] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *AVSS*, pages 296–301, 2009.

[17] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *ACCV*, pages 538–552, 2015.

[18] P. J. Phillips, H. Moon, S. Rizvi, P. J. Rauss, et al. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, 2000.

[19] C. Qu, E. Monari, T. Schuchert, and J. Beyerer. Adaptive contour fitting for pose-invariant 3D face shape reconstruction. In *BMVC*, 2015.

[20] J. Roth, Y. Tong, and X. Liu. Unconstrained 3D face reconstruction. In *CVPR*, 2015.

[21] J. Roth, Y. Tong, and X. Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *CVPR*, 2016.

[22] J. M. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *ICCV*, 2009.

[23] S. Shan, Y. Chang, W. Gao, B. Cao, and P. Yang. Curse of mis-alignment in face recognition: problem and a novel mis-alignment learning solution. In *FG*, pages 314–320, 2004.

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

[25] S. Tulyakov and N. Sebe. Regressing a 3D face shape from a single image. In *ICCV*, 2015.

[26] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *CVPR*, pages 3659–3667, 2015.

[27] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, pages 2729–2736, 2010.

[28] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma. Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(2):372–386, 2012.

[29] N. Wang, X. Gao, D. Tao, and X. Li. Facial feature point detection: A comprehensive survey. *arXiv preprint arXiv:1410.1037*, 2014.

[30] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *ICCV*, 2015.

[31] 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 *ICCV*, pages 1944–1951, 2013.

[32] 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.

[33] J. Zhang, S. K. Zhou, D. Comaniciu, and L. McMillan. Conditional density learning via regression with application to deformable shape segmentation. In *CVPR*, pages 1–8, 2008.

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

[35] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCVW*, pages 386–391, 2013.

[36] 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 *CVPR*, pages 787–796, 2015.

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

[38] X. Zhu, J. Yan, D. Yi, Z. Lei, and S. Z. Li. Discriminative 3D morphable model fitting. In *FG*, pages 1–8, 2015.