Towards Fast, Accurate and Stable 3D Dense Face Alignment

Jianzhu Guo[1,2⋆][0000−0002−8493−3689], Xiangyu Zhu[1,2⋆][0000−0002−4636−9677], Yang Yang[1,2][0000−0003−0559−5464], Fan Yang[3][0000−0003−4348−3148], Zhen Lei[1,2†][0000−0002−0791−189X], and Stan Z. Li[4][0000−0002−2961−8096]

[1] CBSR&NLPR, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] College of Software, Beihang University
[4] School of Engineering, Westlake University
{jianzhu.guo,xiangyu.zhu,yang.yang,zlei,szli}@nlpr.ia.ac.cn, fanyang@buaa.edu.cn

Abstract. Existing methods of 3D dense face alignment mainly concentrate on accuracy, thus limiting the scope of their practical applications. In this paper, we propose a novel regression framework which makes a balance among speed, accuracy and stability. Firstly, on the basis of a lightweight backbone, we propose a meta-joint optimization strategy to dynamically regress a small set of 3DMM parameters, which greatly enhances speed and accuracy simultaneously. To further improve the stability on videos, we present a virtual synthesis method to transform one still image to a short-video which incorporates in-plane and out-of-plane face moving. On the premise of high accuracy and stability, our model runs at over 50fps on a single CPU core and outperforms other state-of-the-art heavy models simultaneously. Experiments on several challenging datasets validate the efficiency of our method. The code and models will be available at https://github.com/cleardusk/3DDFA_V2.

Keywords: 3D Dense Face Alignment · 3D Face Reconstruction

1 Introduction

3D dense face alignment is essential for many face related tasks, e.g., recognition [42,6,25,23,12], animation [9], avatar retargeting [8], tracking [47], attribute classification [3,21,20], image restoration [48,11,10], anti-spoofing [46,51,38,50,24]. Recent studies are mainly divided into two categories: 3D Morphable Model (3DMM) parameters regression [27,55,32,34,45,56,22] and dense vertices regression [26,17]. Dense vertices regression methods directly regress the coordinates of all the 3D points (usually more than 20,000) through a fully convolutional network [26,17], achieving the state-of-the-art performance. However, the resolution of reconstructed faces relies on the size of the feature map and these

⋆ Equal contribution.
† Corresponding author.

methods rely on heavy networks like hourglass [36] or its variants, which are slow and memory-consuming in inference. The natural way of speeding it up is to prune channels. We try to prune 77.5% channels on the state-of-the-art PRNet [17] to achieve real-time speed on CPU, but find the error greatly increases 44.8% (3.62% vs. 5.24%). Besides, an obvious disadvantage is the presence of checkerboard artifacts due to the deconvolution operators, which is present in the supplementary material. Another strategy is to regress a small set of 3DMM

Fig. 1: A few results from our *MobileNet (M+R+S)* model, which runs at over 50fps on a single CPU core or over 130fps on multiple CPU cores.

parameters (usually less than 200). Compared with dense vertices, 3DMM parameters have low dimensionality and low redundancy, which are appropriate to regress by a lightweight network. However, different 3DMM parameters influence the reconstructed 3D face [55] differently, making the regression challenging since we have to dynamically re-weight each parameter according to their importance during training. Cascaded structures [55,34,56] are always adopted to progressively update the parameters but the computation cost is increased linearly with the number of cascaded stages.

In this paper, we aim to accelerate the speed to CPU real time and achieve the state-of-the-art performance simultaneously. To this end, we choose to regress 3DMM parameters with a fast backbone, e.g. MobileNet. To handle the optimization problem of the parameters regression framework, we exploit two different loss terms WPDC and VDC [55] (see Sec. 2.2) and propose our meta-joint optimization to combine the advantages of them. The meta-joint optimization looks ahead by k-steps with WPDC and VDC on the meta-train batches, then dynamically selects the better one according to the error on the meta-test batch. By doing so, the whole optimization converges faster and achieves better performance than the vanilla-joint optimization. Besides, a landmark-regression regularization is introduced to further alleviate the optimization problem to achieve higher accuracy. In addition to single image, 3D face applications on videos are becoming more and more popular [9,8,29,28], where reconstructing stable results across consecutive frames is important, but it is often ignored by recent methods [55,26,17,56]. Video-based training [37,33,16,41] is always adopted to improve the stability in 2D face alignment. However, no video databases are

publicly available for 3D dense face alignment. To address it, we propose a 3D aided short-video-synthesis method, which simulates both in-plane and out-of-plane face moving to transform one still image to a short video, so that our network can adjust results of consecutive frames. Experiments show our short-video-synthesis method significantly improves the stability on videos.

In general, our proposed methods are (i) *fast*: It takes about 7.2ms with an single image as input (almost 24x faster than PRNet) and runs at over 50fps (19.2ms) on a single CPU core or over 130fps (7.2ms) on multiple CPU cores (i5-8259U processor), (ii) *accurate*: By dynamically optimizing 3DMM parameters through a novel meta-optimization strategy combining the fast WPDC and VDC, we surpass the state-of-the-art results [55,26,17,56] under a strict computation burden in inference, and (iii) *stable*: In a mini-batch, one still image is transformed slightly and smoothly into a short synthetic video, involving both in-plane and out-of-plane rotations, which provides temporal information of adjacent frames for training. Extensive experimental results on four datasets show that the overall performance of our method is the best.

2 Methodology

This section details our proposed approach. We first discuss 3D Morphable Model (3DMM) [5]. Then, we introduce the proposed methods of the meta-joint optimization, landmark-regression regularization and 3D aided short-video-synthesis. The overall pipeline is illustrated in Fig. 2 and the algorithm is described in Algorithm 1.

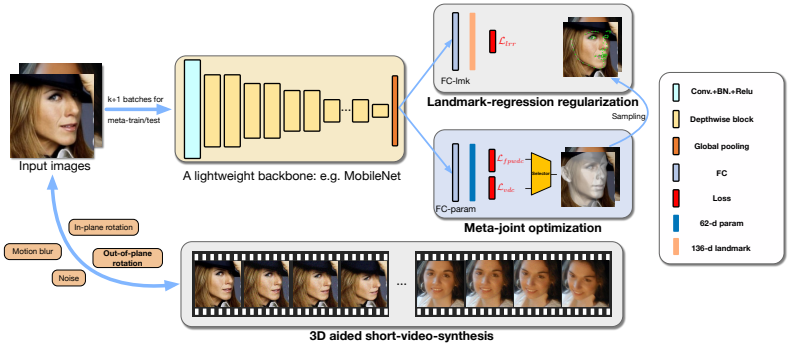

Fig. 2: Overview of our method. Our architecture consists of four parts: the lightweight backbone like MobileNet for predicting 3DMM parameters, the meta-joint optimization of fWPDC and VDC, the landmark-regression regularization and the short-video-synthesis for training. The landmark-regression branch is discarded in inference, thus not increasing any computation burden.

Algorithm 1: The overall algorithm of our proposed methods.

Input: Training data $\mathcal{X} = \{(x^l, p^l)\}_{l=1}^{M}$.
Init: Model parameters θ initialized randomly, the learning rate α, look-ahead step k, length of 3D aided short-video-synthesis n, and batch-size of B.

for *i in max_iterations* **do**
  Randomly sampling k batches $\{\mathcal{X}_{mtr}^l\}_{l=1}^{k}$ for meta-train and one disjoint batch $\mathcal{X}_{mte}$ for meta-test, each batch contains B pairs: $\{(x^l, p^l)\}_{l=1}^{B}$.
  `// short-video-synthesis`
  for *each* $x \in \mathcal{X}_{mtr}$ *or* $\mathcal{X}_{mte}$ **do**
    Synthesize a short-video with n adjacent frames: $\{(x_0, p_0|x_0)|x_0 = x\} \cup \{(x'_j, p'_j|x'_j)|x'_j = (M \circ P)(x_j), x_j = (T \circ F)(x_{j-1}), 1 \leq j \leq n-1\}$.
  end
  `// Meta-joint optimization with landmark-regression regularization`
  Let $\theta_i^f, \theta_i^v \leftarrow \theta_i$;
  for $j = 1 \cdots k$ **do**
    $\theta_{i+j}^f \leftarrow \alpha \nabla_{\theta_{i+j-1}^f} \left(\mathcal{L}_{fwpdc}(\theta_{i+j-1}^f, \mathcal{X}_{mtr}^j) + \frac{|l_{fwpdc}|}{|l_{lrr}|} \cdot \mathcal{L}_{lrr}(\theta_{i+j-1}^f, \mathcal{X}_{mtr}^j) \right)$;
    $\theta_{i+j}^v \leftarrow \alpha \nabla_{\theta_{i+j-1}^v} \left(\mathcal{L}_{vdc}(\theta_{i+j-1}^v, \mathcal{X}_{mtr}^j) + \frac{|l_{vdc}|}{|l_{lrr}|} \cdot \mathcal{L}_{lrr}(\theta_{i+j-1}^v, \mathcal{X}_{mtr}^j) \right)$;
  end
  Select $\theta_{i+1} \leftarrow \arg\min_{\theta_{i+k}} \left(\mathcal{L}_{vdc}(\theta_{i+k}^f, \mathcal{X}_{mte}), \mathcal{L}_{vdc}(\theta_{i+k}^v, \mathcal{X}_{mte}) \right)$;
end

2.1 Preliminary of 3DMM

The original 3DMM can be described as:

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}, \tag{1}$$

where $\mathbf{S}$ is the 3D face mesh, $\overline{\mathbf{S}}$ is the mean 3D shape, $\boldsymbol{\alpha}_{id}$ is the shape parameter corresponding to the 3D shape base $\mathbf{A}_{id}$, $\mathbf{A}_{exp}$ is the expression base and $\boldsymbol{\alpha}_{exp}$ is the expression parameter. After the 3D face is reconstructed, it can be projected onto the image plane with the scale orthographic projection:

$$V_{2d}(\mathbf{p}) = f * \mathbf{Pr} * \mathbf{R} * \left(\overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}\right) + \mathbf{t}_{2d}, \tag{2}$$

where $V_{2d}(\mathbf{p})$ is the projection function generating the 2D positions of model vertices, f is the scale factor, $\mathbf{Pr}$ is the orthographic projection matrix, $\mathbf{R}$ is the rotation matrix constructed by Euler angles including pitch, yaw, roll and $\mathbf{t}_{2d}$ is the translation vector. The complete parameters of 3DMM are $\mathbf{p} = [f, \text{pitch}, \text{yaw}, \text{roll}, \mathbf{t}_{2d}, \boldsymbol{\alpha}_{id}, \boldsymbol{\alpha}_{exp}]$.

However, the three Euler angles will cause the gimbal lock [31] when faces are close to the profile view. This ambiguity will confuse the regressor to degrade the performance, so we choose to regress the similarity transformation matrix instead of $[f, \text{pitch}, \text{yaw}, \text{roll}, \mathbf{t}_{2d}]$ to reduce the regression difficulty: $\mathbf{T} = f\,[\mathbf{R}; \mathbf{t}_{3d}]$, where $\mathbf{T} \in \mathbb{R}^{3\times4}$ is constructed by a scale factor f, a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}_{3d} = \begin{bmatrix} \mathbf{t}_{2d} \\ 0 \end{bmatrix}$. Therefore, the scale orthographic projection in Eqn. 2 can be simplified as:

$$V_{2d}(\mathbf{p}) = \mathbf{Pr} * \mathbf{T} * \begin{bmatrix} \overline{\mathbf{S}} + \mathbf{A}\boldsymbol{\alpha} \\ \mathbf{1} \end{bmatrix}, \tag{3}$$

where $\mathbf{A} = [\mathbf{A}_{id}, \mathbf{A}_{exp}]$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_{id}, \boldsymbol{\alpha}_{exp}]$. Our regression objective is described as $\mathbf{p} = [\mathbf{T}, \boldsymbol{\alpha}]$.

The high-dimensional parameters $\boldsymbol{\alpha}_{shp} \in \mathbb{R}^{199}$, $\boldsymbol{\alpha}_{exp} \in \mathbb{R}^{29}$ are redundant, since 3DMM models the 3D face shape with PCA and the last parts of parameters have little effect on the face shape. We choose only the first 40 dimensions of $\boldsymbol{\alpha}_{shp}$ and the first 10 dimensions of $\boldsymbol{\alpha}_{exp}$ as our regression target, since the NME increase is acceptable and the reconstruction can be greatly accelerated. The NME error heatmap caused by different size of shape and expression dimensions is present in the supplementary material. Therefore, our complete regression target is simplified as $\mathbf{p} = [\mathbf{T}^{3\times4}, \boldsymbol{\alpha}^{50}]$, with 62 dimensions in total, where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{40}_{shp}, \boldsymbol{\alpha}^{10}_{exp}]$. To eliminate the negative impact of magnitude differences between $\mathbf{T}$ and $\boldsymbol{\alpha}$, Z-score normalizing is adopted: $\mathbf{p} = (\mathbf{p} - \boldsymbol{\mu}_p)/\boldsymbol{\sigma}_p$, where $\boldsymbol{\mu}_p \in \mathbb{R}^{62}$ is the mean of parameters and $\boldsymbol{\sigma}_p \in \mathbb{R}^{62}$ indicates the standard deviation of parameters.

2.2 Meta-joint Optimization

We first review the Vertex Distance Cost (VDC) and Weighted Parameter Distance Cost (WPDC) in [55], then derivate the meta-joint optimization to facilitate the parameters regression.

The VDC term $\mathcal{L}_{vdc}$ directly optimizes $\mathbf{p}$ by minimizing the vertex distances between the fitted 3D face and the ground truth:

$$\mathcal{L}_{vdc} = \left\| V_{3d}\left(\mathbf{p}\right) - V_{3d}\left(\mathbf{p}^g\right) \right\|^2, \tag{4}$$

where $\mathbf{p}^g$ is the ground truth parameter, $\mathbf{p}$ is the predicted parameter and $V_{3d}(\cdot)$ is the 3D face reconstruction formulated as:

$$V_{3d}(\mathbf{p}) = \mathbf{T} * \begin{bmatrix} \overline{\mathbf{S}} + \mathbf{A}\boldsymbol{\alpha} \\ \mathbf{1} \end{bmatrix}. \tag{5}$$

Different from VDC, the WPDC term [55] $\mathcal{L}_{wpdc}$ assigns different weights to each parameter:

$$\mathcal{L}_{wpdc} = \left\| \mathbf{w} \cdot (\mathbf{p} - \mathbf{p}^g) \right\|^2, \tag{6}$$

where $\mathbf{w}$ indicates the importance weight as follows:

$$\begin{aligned} \mathbf{w} &= (w_1, w_2, \ldots, w_i, \ldots, w_n), \\ w_i &= \left\| V_{3d}(\mathbf{p}^{de,i}) - V_{3d}(\mathbf{p}^g) \right\| / Z, \\ \mathbf{p}^{de,i} &= (\mathbf{p}^g_1, \mathbf{p}^g_2, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}^g_n), \end{aligned} \tag{7}$$

where n is the number of parameters ($n = 62$ in our regression framework), $\mathbf{p}^{de,i}$ is the i-degraded parameter whose i-th element is from the predicted $\mathbf{p}$, Z is the

maximum of $\mathbf{w}$ for regularization. The term $\left\|V_{3d}(\mathbf{p}^{de,i}) - V_{3d}(\mathbf{p}^{g})\right\|$ models the importance of i-th parameter.

Algorithm 2: fWPDC: Fast WPDC Algorithm.

Input : Shape and expression base: $\mathbf{A} = [\mathbf{A}_{id}, \mathbf{A}_{exp}] \in \mathbb{R}^{3N\times 50}$
Mean shape: $\overline{\mathbf{S}} \in \mathbb{R}^{3\times N}$
Predicted parameters: $\mathbf{p} = [\mathbf{T} \in \mathbb{R}^{3\times 4}, \boldsymbol{\alpha} \in \mathbb{R}^{50}]$
Ground truth parameters: $\mathbf{p}^{g} = [\mathbf{T}^{g} \in \mathbb{R}^{3\times 4}, \boldsymbol{\alpha}^{g} \in \mathbb{R}^{50}]$
Scale factor scalar: f
Output: WPDC item

Initialize the weights of the parameter $\mathbf{T}$ and $\boldsymbol{\alpha}$: $\mathbf{w}_T \in \mathbb{R}^{3\times 4}$, $\mathbf{w}_\alpha \in \mathbb{R}^{50}$;
`// Calculating the weight of transform matrix`
Reconstruct the vertices without projection: $\mathbf{S} = \overline{\mathbf{S}} + \mathbf{A}\boldsymbol{\alpha} \in \mathbb{R}^{3\times N}$;
for $i = 1, 2, 3$ **do**
  $\mathbf{w}_T(:, i) = \left(\mathbf{T}(:, i) - \mathbf{T}^{g}(:, i)\right) \cdot \|\mathbf{S}\,(i, :)\|$;
end
$\mathbf{w}_T(:, 4) = \left(\mathbf{T}(:, 4) - \mathbf{T}^{g}(:, 4)\right) \cdot \sqrt{N/3}$ and then flatten $\mathbf{w}_T$ to the vector form in row-major order;
`// Calculating the weight of shape and expression parameters`
for $i = 1 \ldots 50$ **do**
  $\mathbf{w}_\alpha(i) = f \cdot \left(\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}^{g}(i)\right) \cdot \|\mathbf{A}\,(:, i)\|$;
end
`// Calculating the fWPDC item`
Get the maximum value Z of the weights $(\mathbf{w}_T, \mathbf{w}_\alpha)$ and normalize them: $\mathbf{w}_T = \mathbf{w}_T/Z$, $\mathbf{w}_\alpha = \mathbf{w}_\alpha/Z$;
Calculate the WPDC item: $\mathcal{L}_{fwpdc} = \|\mathbf{w}_T \cdot (\mathbf{T} - \mathbf{T}^{g})\|^2 + \|\mathbf{w}_\alpha \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{g})\|^2$

fWPDC. The original calculation of $\mathbf{w}$ in WPDC is rather slow as the calculation of each w_i needs to reconstruct all the vertices once, which is a bottleneck for fast training. We find that the vertices can be only reconstructed once by decomposing the weight calculation into two parts: the similarity transformation matrix $\mathbf{T}$, and the combination of shape and expression parameters $\boldsymbol{\alpha}$. Therefore, we design a fast implementation of WPDC named fWPDC: (i) reconstructing the vertices without projection $\mathbf{S} = \overline{\mathbf{S}} + \mathbf{A}\boldsymbol{\alpha}$ and calculating $\mathbf{w}_T$ using the norm of row vectors; (ii) calculating $\mathbf{w}_\alpha$ using the norm of column vectors of $\mathbf{A}$ and the input scale f: $\mathbf{w}_\alpha(i) = f \cdot \left(\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}^{g}(i)\right) \cdot \|\mathbf{A}\,(:, i)\|$; (iii) Combining them to calculate the final cost. The detailed algorithm of fWPDC is described in Algorithm 2. fWPDC only reconstructs dense vertices once, not 62 times as WPDC, thus greatly reducing the computation cost. With 128 samples as a batch input, the original WPDC takes 41.7ms while fWPDC only takes 3.6ms. fWPDC is over 10x faster than the original WPDC while preserving the same outputs.

Exploitation of VDC and fWPDC. Through Eqn. 4 and Eqn. 6, we find: WPDC/fWPDC is suitable for parameters regression since each parameter is appropriately weighted, while VDC can directly reflect the goodness of the 3D face reconstructed from parameters. In Fig. 3, we investigate how these two losses converge as the training progresses. It is shown that the optimization is difficult for VDC since the vertex error is still over 15 when training converges. The work in [56] also demonstrates that optimizing VDC with gradient descent converges very slowly due to the "zig-zagging" problem. In contrast, the convergence of fWPDC is much faster than VDC and the error is about 7 when training converges. Surprisingly, if the fWPDC-trained model is fine-tuned by VDC, we can get a much lower error than fWPDC. Based on the above observa-

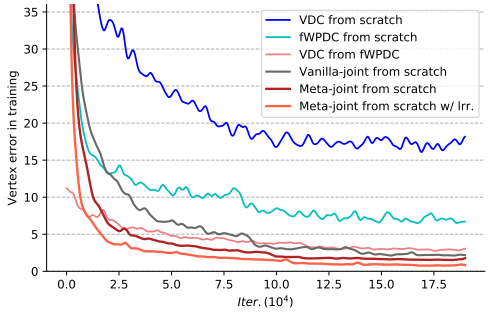

Fig. 3: The vertex error in training on 300W-LP supervised by different loss terms. VDC from scratch has the highest error, fWPDC is lower than VDC, and VDC from fWPDC is better than both. When combining VDC and fWPDC, the proposed meta-joint optimization converges faster and reaches lower error than vanilla-joint, and achieves even better convergence when incorporating the landmark-regression regularization.

tion, we conclude that: *training from scratch with VDC is hard to converge* and *the network is not fully trained by fWPDC in the late stage.*

Meta-joint optimization. Based on above discussions, it is natural to weight two terms to perform a vanilla-joint optimization: $\mathcal{L}_{vanilla\text{-}joint} = \beta\mathcal{L}_{fwpdc} + (1-\beta)\frac{|l_{fwpdc}|}{|l_{vdc}|}\cdot\mathcal{L}_{vdc}$, where $\beta \in [0,1]$ controls the importance between fWPDC and VDC. However, the vanilla-joint optimization relies on the manually set hyper-parameter β and does not achieves satisfactory results in Fig. 3. Inspired by Lookahead [53] and MAML [18], we propose a meta-joint optimization strategy to dynamically combine fWPDC and VDC. The overview of the meta-joint optimization is shown in Fig. 4. In the training process, the model looks ahead by k-steps with the cost fWPDC or VDC on k meta-train batches $\mathcal{X}_{mtr}$, then selects the better one between fWPDC and VDC according to the vertex error on the meta-test batch. Specifically, the whole meta-joint optimization consists of four steps: (i) sampling k batches of training samples $\mathcal{X}_{mtr}$ for meta-train and one batch $\mathcal{X}_{mte}$ for meta-test; (ii) meta-train: updating the current model parameters θ_i with fWPDC and VDC on $\mathcal{X}_{mtr}$ by k-steps, respectively, getting two parameter states θ^f_{i+k} and θ^v_{i+k}; (iii) meta-test: evaluating the vertex error θ^f_{i+k} and θ^v_{i+k} on $\mathcal{X}_{mte}$; (iv) selecting the parameters which have the lower error to update θ_i. The proposed meta-joint optimization can be directly embedded into the standard training regime. From Fig. 3, we can observe that the meta-joint optimization converges faster than vanilla-joint and has the lower error.

2.3 Landmark-regression Regularization

In 3D face reconstruction [15,14,44,43,19], the 2D sparse landmarks after projecting are usually used as an extra regularization to facilitate the parameters

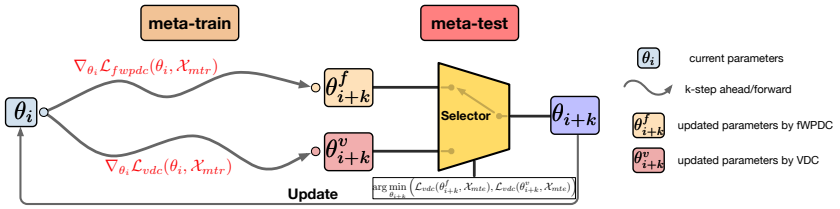

Fig. 4: Overview of the meta-joint optimization.

regression. In our regression framework, we find that treating 2D sparse landmarks as a auxiliary regression task benefits more. As shown in Fig. 2, we add an additional landmark-regression task on the global pooling layer, trained by L2 loss. The difference between the former landmark-regularization and the latter landmark-regression regularization is that the latter introduces extra parameters to regress the landmarks. In other words, the landmark-regression regularization is a task-level regularization. From the tomato curve in Fig. 3, we get a lower error by incorporating the landmark-regression regularization. The comparative results in Table 3 show our proposed landmark-regression regularization is better than landmark-regularization (3.59% vs. 3.71% on AFLW2000-3D). The landmark-regression regularization is formulated as: $\mathcal{L}_{lrr} = \frac{1}{N}\sum_{i=1}^{N}\|l_i - l_i^g\|_2^2$, where N is 136 here as we utilize 68 2D landmarks and flatten them into a 136-d vector.

2.4 3D Aided Short-video-synthesis

Video based 3D face applications have become more and more popular [9,8,29,28] recently. In these applications, 3D dense face alignment methods are required to run on videos and provide stable reconstruction results across adjacent frames. The stability means that the changing of the reconstructed 3D faces across adjacent frames should be consistent with the true face moving in a fine-grained level. However, most of existing methods [55,56,26,17] omit this requirenment and the predictions suffer from random jittering. In 2D face alignment, post-processing like temporal filtering is a common strategy to reduce the jittering, but it degrades the precision and causes the frame delay. Besides, since no public video databases for 3D dense face alignment are available, the video training strategies [16,37,41,33] cannot work here. A challenge arises: *can we improve the stability on videos with only still images available when training?*

To address this challenge, we propose a batch-level 3D aided short-video-synthesis strategy, which expands one still image to several adjacent frames, forming a short synthetic video in a mini-batch. The common patterns in a video can be modelled as: (i) Noise. We model noise as $P(x) = x + \mathcal{N}(0, \Sigma)$, where $\Sigma = \sigma^2 I$. (ii) Motion Blur. Motion blur can be formulated as $M(x) = K * x$, where K is the convolution kernel (the operator $*$ denotes a convolution). (iii) In-plane rotation. Given two adjacent frames x_t and x_{t+1}, the in-plane temporal

change from x_t to x_{t+1} can be described as a similarity transform $T(\cdot)$:

$$T(\cdot) = \Delta s \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & \Delta t_1 \\ \sin(\Delta\theta) & \cos(\Delta\theta) & \Delta t_2 \end{bmatrix}, \tag{8}$$

where Δs is the scale perturbation, $\Delta\theta$ is the rotation perturbation, Δt_1 and Δt_2 are translation perturbations. (iv) Since human faces share similar 3D structure, we are also able to synthesize the out-of-plane face moving. Face profiling [55] $F(\cdot)$, which is originally proposed to solving large-pose face alignment, is utilized to progressively increase the yaw angle $\Delta\phi$ and pitch angle $\Delta\gamma$ of the face. Specifically, we sample several still images in a mini-batch and for each still image x_0, we transform it slightly and smoothly to generate a synthetic video with n adjacent frames: $\{x'_j | x'_j = (M \circ P)(x_j), x_j = (T \circ F)(x_{j-1}), 1 \leq j \leq n-1\} \cup \{x_0\}$. In Fig. 5, we give an illustration of how these transformations are applied on an image to generate several adjacent frames.

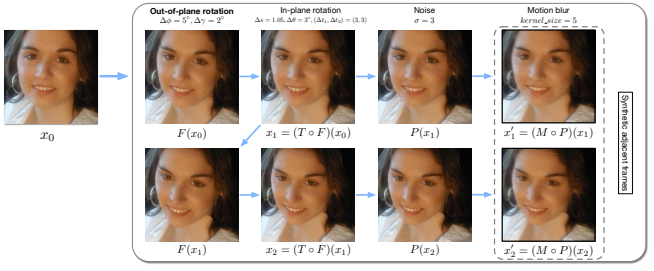

Fig. 5: An illustration of how two adjacent frames are synthesized in our 3D aided short-video-synthesis.

3 Experiments

In this section, we first introduce the datasets and protocols; then, we give comparison experiments on the accuracy and stability; thirdly, the complexity and running speed are evaluated; extensive discussions are finally made. The implementation details, generalization and scaling-up ability of our proposed method are in the supplementary material.

3.1 Datasets and Evaluation Protocols

Five datasets are used in our experiments: **300W-LP** [55] (300W Across Large Poses) is composed of the synthesized large-pose face images from 300W [39], including AFW [57], LFPW [2], HELEN [54], IBUG [39], and XM2VTS [35]. Specifically, the face profiling method [55] is adopted to generate 122,450 samples across large poses. **AFLW** [30] consists of 21,080 in-the-wild faces (following

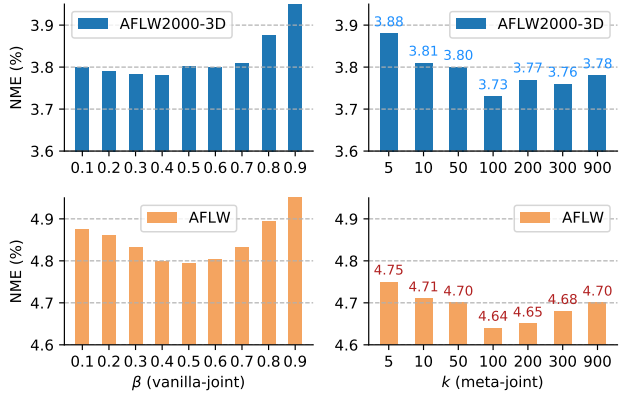

Fig. 6: Ablative results of the vanilla-joint optimization with different β and meta-joint optimization with different k. Lower NME (%) is better.

Table 1: The NME (%) of different methods on AFLW2000-3D and AFLW. The first and the second best results are highlighted. M, R, S denote the meta-joint optimization, landmark-regression regularization and short-video-synthesis, respectively.

Method	AFLW2000-3D (68 pts)				AFLW (21 pts)			
	[0, 30]	[30, 60]	[60, 90]	Mean	[0, 30]	[30, 60]	[60, 90]	Mean
ESR [13]	4.60	6.70	12.67	7.99	5.66	7.12	11.94	8.24
SDM [47]	3.67	4.94	9.67	6.12	4.75	5.55	9.34	6.55
3DDFA [55]	3.78	4.54	7.93	5.42	5.00	5.06	6.74	5.60
3DDFA+SDM [55]	3.43	4.24	7.17	4.94	4.75	4.83	6.38	5.32
Yu et al. [49]	3.62	6.06	9.56	6.41	-	-	-	-
DeFA [34]	-	-	-	4.50	-	-	-	-
3DSTN [4]	3.15	4.33	5.98	4.49	**3.55**	**3.92**	5.21	**4.23**
3D-FAN [7]	3.15	3.53	4.60	3.76	4.40	4.52	5.17	4.69
3DDFA-TPAMI [56]	2.84	3.57	4.96	3.79	4.11	4.38	5.16	4.55
PRNet [17]	**2.75**	3.51	4.61	3.62	4.19	4.69	5.45	4.77
MobileNet (M+R)	**2.75**	**3.49**	**4.53**	**3.59**	4.06	4.41	**5.02**	4.50
MobileNet (M+R+S)	**2.63**	**3.42**	**4.48**	**3.51**	**3.98**	**4.31**	**4.99**	**4.43**

[56,22]) with large poses (yaw from -90° to 90°). Each image is annotated up to 21 visible landmarks. **AFLW2000-3D** [55] is constructed by [55] for evaluating 3D face alignment performance, which contains the ground truth 3D faces and the corresponding 68 landmarks of the first 2,000 AFLW samples. **Florence** [1] is a 3D face dataset containing 53 subjects with its ground truth 3D mesh acquired from a structured-light scanning system. For evaluation, we generate renderings with different poses for each subject following VRN [26] and PRNet [17]. **Menpo-3D** [52] provides a benchmark for evaluating 3D facial landmark localization algorithms in the wild in arbitrary poses. Specifically, Menpo-3D provides 3D facial landmarks for 55 videos from 300-VW [40] competition.

Table 2: The NME (%) on Florence, AFLW2000-3D (Dense), NME (%) / Stability (%) on Menpo-3D, running complexity and time with different methods. Our method outputs 3D dense vertices with only 2.1ms (2ms for parameters prediction and 0.1ms for vertices reconstruction) in GPU or 7.2ms in CPU (6.2ms for parameters prediction and 1ms for vertices reconstruction). The first and second best results are highlighted.

Methods	Florence	AFLW2000-3D (Dense)	Menpo-3D	Params	MACs	Run Time (ms)
3DDFA [55]	6.38	6.56	-	-	-	75.7=23.2(GPU)+52.5(CPU)
VRN [26]	5.27	-	-	-	-	69.0(GPU)
DeFA [34]	-	6.04	-	-	1426M	35.4=11.8(GPU)+23.6(CPU)
PRNet [17]	3.76	4.41	1.90 / 0.54	13.4M	6190M	9.8(GPU) / 175.0(CPU)
MobileNet (M+R)	**3.59**	**4.20**	**1.86 / 0.52**	**3.27M**	**183M**	**2.1(GPU) / 7.2(CPU)**
MobileNet (M+R+S)	**3.56**	**4.18**	**1.71 / 0.48**			

Protocols. The protocol on AFLW follows [55] and Normalized Mean Error (NME) by bounding box size is reported. Two protocols on AFLW2000-3D are applied: the first one follows AFLW, and the other one follows [17] to evaluate the NME of 3D face reconstruction normalized by the bounding box size. For Florence, we follow [26,17] to evaluate the NME of 3D face reconstruction normalized by outer interocular distance. As for Menpo-3D, we evaluate the NME on still frames and the stability across adjacent frames. We calculate the stability following [41] by measuring the NME between the predicted offsets and the ground-truth offsets of adjacent frames. Specifically, at frame $t-1$ and t, the ground-truth landmark offset is $\Delta p = p_t - p_{t-1}$, the prediction offset is $\Delta q = q_t - q_{t-1}$, the error $\Delta p - \Delta q$ normalized by the bounding box size represents the stability. Since 300W-LP only has the indices of 68 landmarks, we use 68 landmarks of Menpo-3D for consistency.

3.2 Ablation Study

Table 3: The comparative and ablative results on AFLW2000-3D and AFLW. The mean NMEs (%) across small, medium and large poses on AFLW2000-3D and AFLW are reported. lmk. indicates landmark constraint on the parameter regression like [44] and lrr. is the proposed landmark-regression regularization.

	Method	AFLW2000-3D	AFLW
Baseline	VDC	5.23	6.37
	fWDPC	4.04	5.10
Joint-optimization Options	VDC from fWPDC	3.88	4.83
	Vanilla-joint	3.80	4.80
	Meta-joint	**3.73**	**4.64**
Utilization of 2D landmarks	VDC w/ lrr.	3.92	4.92
	fWPDC w/ lrr.	3.89	4.84
	Meta-joint w/ lmk.	3.71	4.80
	Meta-joint w/ lrr.	**3.59**	**4.50**

Table 4: Comparisons of NME (%) / Stability (%) on Menpo-3D. svs. indicates short-video-synthesis, rnd. indicates applying in-plane and out-of-plane rotations randomly in one mini-batch.

Method	Menpo-3D
fWPDC w/o svs.	1.96 / 0.54
fWPDC w/ svs.	**1.84 / 0.51**
Meta-joint+lrr. w/o svs.	1.86 / 0.52
Meta-joint+lrr. w/ rnd.	1.76 / 0.50
Meta-joint+lrr. w/ svs.	**1.71 / 0.48**

To evaluate the effectiveness of the meta-joint optimization and the landmark-regression regularization, we carry out comparative experiments including our two baselines: *VDC* and *fWPDC*, three joint options: (i) *VDC from fWPDC*: fine-tune the model with VDC loss from the pre-trained model by fWPDC; (ii) *Vanilla-joint*: weight VDC and fWPDC by the best scalar $\beta = 0.5$; (iii) *Meta-joint*: the proposed meta-joint optimization with best $k = 100$ and four options of how the 2D landmarks are utilized. From Table 3, Table 4, Fig. 3 and Fig. 6, we can draw the following conclusions:

Meta-joint optimization performs better. Comparing with two baselines *VDC* and *fWPDC*, all three joint optimization methods perform better. Among three joint optimization methods, the proposed meta-joint performs better than *VDC from fWPDC* and *vanilla-joint*: the mean NME drops from 4.04% to 3.73% on AFLW2000-3D and 5.10% to 4.64% on AFLW when compared with the baseline *fWPDC*. Furthermore, we conduct ablative experiments with different β for *vanilla-joint* and different look-ahead step k in Fig. 6. We can observe that $\beta = 0.5$ is the best setting for *vanilla-joint*, but *meta-joint* still outperforms it and $k = 100$ performs best on both AFLW2000-3D and AFLW. Overall, the proposed meta-joint optimization is effective in alleviate the training and promoting the performance.

Landmark-regression regularization benefits. Another contribution is the landmark-regression regularization, which can also be regarded as an auxiliary task to parameters regression. From Table 3, the improvements from *fWPDC* to *fWPDC w/ lrr.* on AFLW2000-3D and AFLW are 0.15% and 0.26%, and the improvements from *Meta-joint* to *Meta-joint w/ lrr.* on AFLW2000-3D and AFLW are 0.14% and 0.14%. We also compare the proposed landmark-regression regularization with prior methods [44,43] which directly impose landmark constraint on the parameter regression, the results show ours is significantly better: 3.59% vs. 3.71% on AFLW2000-3D. We further evaluate the performance of the landmark-regression branch on AFLW2000-3D and AFLW. The performances are 3.58% and 4.52% respectively, which are close to the parameter branch. It indicates that these two tasks are highly related. Overall, the landmark-regression regularization benefits the training and promotes the performance.

Short-video-synthesis improves stability. The last contribution is 3D aided short-video-synthesis, which is designed to enhance stability on videos by augmenting one still image to a short video in a mini-batch. The results in

Table 4 indicate that short-video-synthesis works for both the fWPDC and meta-joint optimization. With short-video-synthesis and landmark-regression regularization, the performance on still frames improves from 1.86% to 1.71% and the stability improves from 0.52% to 0.48%. We also evaluate the performance by randomly applying in-plane and out-of-plane rotations in each mini-batch and find it is worse than short-video-synthesis: 1.76% / 0.50% v.s. 1.71% / 0.48%. These results validate the effectiveness of the 3D aided short-video-synthesis.

3.3 Evaluations of Accuracy and Stability

Sparse Face Alignment. We use AFLW2000-3D and AFLW to evaluate sparse face alignment performance with small, medium and large yaw angles. The results in Table 1 indicate that our method performs better than PRNet (3.51% vs. 3.62%) in AFLW2000-3D and better than 3DDFA-TPAMI [56] in AFLW (4.43% vs. 4.55%). Note that these results are achieved with only 3.27M parameters (24% of PRNet) and it takes 6.2ms (3.5% of PRNet) in CPU. The sampling of 68 / 21 landmarks from 3DMM is extremely fast, only 0.01ms (CPU), which can be ignored.

Dense Face Alignment. Dense face alignment is evaluated on Florence and AFLW2000-3D. Our evaluation settings follow [17] to keep consistency. The results in Table 2 show that the proposed method significantly outperforms others. As for 3D dense vertices reconstruction, 45K vertices only takes 1ms in CPU (0.1ms in GPU) with our regression framework.

Video-based 3D Face Alignment. We use Menpo-3D to evaluate both the accuracy and stability. Table 4 has already shown the superiority of short-video-synthesis. We choose to compare our method with recent PRNet [17] in Table 2. The results indicate that our method significantly surpasses PRNet in both the accuracy and stability on videos of Menpo-3D with a much lower computation cost.

3.4 Evaluations of Speed

We compare parameter numbers, MACs (Multiply-Accumulates) measuring the number of fused Multiplication and Addition operations, and the running time of our method with others in Table 2. As for the running speed, 3DDFA [55] takes 23.2ms (GPU) for predicting parameters and 52.5ms (CPU) for PNCC construction, DeFA [17] needs 11.8ms (GPU) to predict 3DMM parameters and 23.6ms (CPU) for post-processing, VRN [26] detects 68 2D landmarks with 28.4ms (GPU) and regresses the 3D dense vertices with 40.6ms (GPU), PRNet [17] predicts the 3D dense vertices with 9.8ms (GPU) or 175ms (CPU). Compared with them, our method takes only 2ms (GPU) or 6.2ms (CPU) to predict 3DMM parameters and 0.1ms (GPU) or 1ms (CPU) to reconstruct 3D dense vertices.

Specifically, compared with the recent PRNet [17], the parameters of our model (3.27M) are less than one-quarter of PRNet (13.4M), and the MACs are less than 1/30 (183.5M vs. 6190M). We measure the overall running time on

GeForce GTX 1080 GPU and i5-8259U CPU with 4 cores. Note that our method takes only 7.2ms, which is almost 24x faster than PRNet (175ms). Besides, we benchmark our method on a single CPU core (using only one thread) and *the running speed of our method is about 19.2ms (over 50fps), including the reconstruction time*. The specific CPU configuration is i5-8259U CPU @ 2.30GHz on a 13-inch MacBook Pro.

3.5 Analysis of Meta-joint Optimization

We visualize the auto-selection result of fWPDC and VDC in the meta-joint optimization, as shown in Fig. 7. We can observe that both $k = 100$ and $k = 200$ show the same trend: fPWDC dominates in the early stage and VDC guides in the late stage. This trend is consistent with the previous observations and gives a clear description of why our proposed meta-joint optimization works.

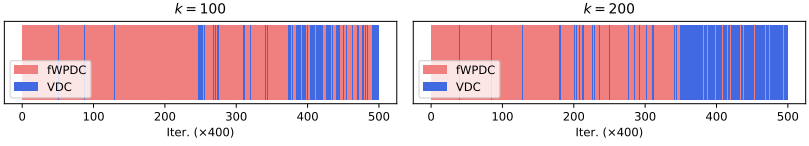

Fig. 7: Auto-selection result of the selector in the meta-joint optimization.

4 Conclusion

In this paper, we have successfully pursued the fast, accurate and stable 3D dense face alignment simultaneously. Towards this target, we make three main efforts: (i) proposing a fast WPDC named fWPDC and the meta-joint optimization to combine fWPDC and VDC to alleviate the problem of optimization; (ii) imposing an extra landmark-regression regularization to promote the performance to state-of-the-art; (iii) proposing the 3D aided short-video-synthesis method to improve the stability on videos. The experimental results demonstrate the effectiveness and efficiency of our proposed methods. Our promising results pave the way for real-time 3D dense face alignment in practical use and the proposed methods may improve the environment by reducing the amount of carbon dioxide released by the huge amounts of energy consumed by GPUs.

5 Acknowledgement

This work was supported in part by the National Key Research & Development Program (No. 2020YFC2003901), Chinese National Natural Science Foundation Projects #61872367, #61876178, #61806196, #61976229.

References

1. Bagdanov, A.D., Bimbo, A.D., Masi, I.: The florence 2d/3d hybrid face dataset. In: ACM workshop on Human gesture and behavior understanding (2011) 10
2. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. TPAMI (2013) 9
3. Bettadapura, V.: Face expression recognition and analysis: the state of the art. arXiv:1203.6722 (2012) 1
4. Bhagavatula, C., Zhu, C., Luu, K., Savvides, M.: Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. In: ICCV (2017) 10
5. Blanz, V., Vetter, T., et al.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH (1999) 3
6. Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., Dunaway, D.: A 3d morphable model learnt from 10,000 faces. In: CVPR (2016) 1
7. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In: ICCV (2017) 10
8. Cao, C., Chai, M., Woodford, O., Luo, L.: Stabilized real-time face tracking via a learned dynamic rigidity prior. In: SIGGRAPH Asia 2018 Technical Papers. ACM (2018) 1, 2, 8
9. Cao, C., Weng, Y., Lin, S., Zhou, K.: 3d shape regression for real-time facial animation. TOG (2013) 1, 2, 8
10. Cao, J., Hu, Y., Zhang, H., He, R., Sun, Z.: Learning a high fidelity pose invariant model for high-resolution face frontalization. In: Advances in neural information processing systems. pp. 2867–2877 (2018) 1
11. Cao, J., Hu, Y., Zhang, H., He, R., Sun, Z.: Towards high fidelity face frontalization in the wild. International Journal of Computer Vision pp. 1–20 (2019) 1
12. Cao, J., Huang, H., Li, Y., He, R., Sun, Z.: Informative sample mining network for multi-domain image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 1
13. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. IJCV (2014) 10
14. Chinaev, N., Chigorin, A., Laptev, I.: Mobileface: 3d face reconstruction with efficient cnn regression. In: ECCV (2018) 7
15. Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., Tong, X.: Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In: CVPR Workshop (2019) 7
16. Dong, X., Yu, S.I., Weng, X., Wei, S.E., Yang, Y., Sheikh, Y.: Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In: CVPR (2018) 2, 8
17. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: ECCV (2018) 1, 2, 3, 8, 10, 11, 13
18. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017) 7
19. Gecer, B., Ploumpis, S., Kotsia, I., Zafeiriou, S.: Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In: CVPR (2019) 7
20. Guo, J., Lei, Z., Wan, J., Avots, E., Hajarolasvadi, N., Knyazev, B., Kuharenko, A., Junior, J.C.S.J., Baró, X., Demirel, H., et al.: Dominant and complementary

emotion recognition from still images of faces. IEEE Access **6**, 26391–26403 (2018) 1
21. Guo, J., Zhou, S., Wu, J., Wan, J., Zhu, X., Lei, Z., Li, S.Z.: Multi-modality network with visual and geometrical information for micro emotion recognition. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). pp. 814–819. IEEE (2017) 1
22. Guo, J., Zhu, X., Lei, Z.: 3ddfa. https://github.com/cleardusk/3DDFA (2018) 1, 10
23. Guo, J., Zhu, X., Lei, Z., Li, S.Z.: Face synthesis for eyeglass-robust face recognition. In: Chinese Conference on Biometric Recognition. pp. 275–284. Springer (2018) 1
24. Guo, J., Zhu, X., Xiao, J., Lei, Z., Wan, G., Li, S.Z.: Improving face anti-spoofing by 3d virtual synthesis. In: 2019 International Conference on Biometrics (ICB). pp. 1–8. IEEE (2019) 1
25. Guo, J., Zhu, X., Zhao, C., Cao, D., Lei, Z., Li, S.Z.: Learning meta face recognition in unseen domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6163–6172 (2020) 1
26. Jackson, A., Bulat, A., Argyriou, V., Tzimiropoulos, G.: Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In: ICCV (2017) 1, 2, 3, 8, 10, 11, 13
27. Jourabloo, A., Liu, X.: Large-pose face alignment via cnn-based dense 3d model fitting. In: CVPR (2016) 1
28. Kim, H., Elgharib, M., Zollhöfer, M., Seidel, H.P., Beeler, T., Richardt, C., Theobalt, C.: Neural style-preserving visual dubbing. ACM Transactions on Graphics (TOG) (2019) 2, 8
29. Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Nießner, M., Pérez, P., Richardt, C., Zollhöfer, M., Theobalt, C.: Deep video portraits. ACM Transactions on Graphics (TOG) (2018) 2, 8
30. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: ICCV Workshop (2011) 9
31. Lepetit, V., Fua, P., et al.: Monocular model-based 3d tracking of rigid objects: A survey. Foundations and Trends® in Computer Graphics and Vision (2005) 4
32. Liu, F., Zeng, D., Zhao, Q., Liu, X.: Joint face alignment and 3d face reconstruction. In: ECCV (2016) 1
33. Liu, H., Lu, J., Feng, J., Zhou, J.: Two-stream transformer networks for video-based face alignment. TPAMI (2018) 2, 8
34. Liu, Y., Jourabloo, A., Ren, W., Liu, X.: Dense face alignment. In: ICCV (2017) 1, 2, 10, 11
35. Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: Xm2vtsdb: The extended m2vts database. In: Second international conference on audio and video-based biometric person authentication (1999) 9
36. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV (2016) 2
37. Peng, X., Feris, R.S., Wang, X., Metaxas, D.N.: A recurrent encoder-decoder network for sequential face alignment. In: ECCV (2016) 2, 8
38. Qin, Y., Zhao, C., Zhu, X., Wang, Z., Yu, Z., Fu, T., Zhou, F., Shi, J., Lei, Z.: Learning meta model for zero-and few-shot face anti-spoofing. The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI) (2020) 1
39. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: CVPRW (2013) 9

40. Shen, J., Zafeiriou, S., Chrysos, G.G., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The first facial landmark tracking in-the-wild challenge: Benchmark and results. In: ICCV Workshops (2015) 10
41. Tai, Y., Liang, Y., Liu, X., Duan, L., Li, J., Wang, C., Huang, F., Chen, Y.: Towards highly accurate and stable face alignment for high-resolution videos. arXiv:1811.00342 (2018) 2, 8, 11
42. Taigman, Y., Yang., M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: CVPR (2014) 1
43. Tewari, A., Bernard, F., Garrido, P., Bharaj, G., Elgharib, M., Seidel, H., Pérez, P., Zollhofer, M., Theobalt, C.: Fml: face model learning from videos. In: CVPR (2019) 7, 12
44. Tewari, A., Zollhofer, M., Kim., H., Garrido, P., Bernard, F., Perez, P., Theobalt, C.: Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: ICCV (2017) 7, 11, 12
45. Tuan, T., Hassner, T., Masi, I., Medioni, G.: Regressing robust and discriminative 3d morphable models with a very deep neural network. In: CVPR (2017) 1
46. Wang, Z., Yu, Z., Zhao, C., Zhu, X., Qin, Y., Zhou, Q., Zhou, F., Lei, Z.: Deep spatial gradient and temporal depth learning for face anti-spoofing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5042–5051 (2020) 1
47. Xiong, X., De, T.F.: Global supervised descent method. In: CVPR (2015) 1, 10
48. Yang, C.Y., Liu, S., Yang, M.H.: Structured face hallucination. In: CVPR (2013) 1
49. Yu, R., Saito, S., Li, H., Ceylan, D., Li, H.: Learning dense facial correspondences in unconstrained images. In: ICCV (2017) 10
50. Yu, Z., Li, X., Niu, X., Shi, J., Zhao, G.: Face anti-spoofing with human material perception. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 1
51. Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., Zhou, F., Zhao, G.: Searching central difference convolutional networks for face anti-spoofing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5295–5305 (2020) 1
52. Zafeiriou, S., Chrysos, G.G., Roussos, A., Ververas, E., Deng, J., Trigeorgis, G.: The 3d menpo facial landmark tracking challenge. In: ICCV (2017) 10
53. Zhang, M., Lucas, J., Ba, J., Hinton, G.E.: Lookahead optimizer: k steps forward, 1 step back. In: NeurIPS (2019) 7
54. Zhou, E., Fan, H., Cao, Z., Jiang, Y., Yin, Q.: Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In: CVPRW (2013) 9
55. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: A 3d solution. In: CVPR (2016) 1, 2, 3, 5, 8, 9, 10, 11, 13
56. Zhu, X., Liu, X., Lei, Z., Li, S.Z.: Face alignment in full pose range: A 3d total solution. TPAMI (2019) 1, 2, 3, 6, 8, 10, 13
57. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: CVPR (2012) 9