TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 13/13 pp690–700
DOI: 10.26599/TST.2020.9010001
Volume 25, Number 5, October 2020

Real-Time Facial Pose Estimation and Tracking by Coarse-to-Fine Iterative Optimization

Xiaolong Yang, Xiaohong Jia*, Mengke Yuan, and Dong-Ming Yan

Abstract: We present a novel and efficient method for real-time multiple facial poses estimation and tracking in a single frame or video. First, we combine two standard convolutional neural network models for face detection and mean shape learning to generate initial estimations of alignment and pose. Then, we design a bi-objective optimization strategy to iteratively refine the obtained estimations. This strategy achieves faster speed and more accurate outputs. Finally, we further apply algebraic filtering processing, including Gaussian filter for background removal and extended Kalman filter for target prediction, to maintain real-time tracking superiority. Only general RGB photos or videos are required, which are captured by a commodity monocular camera without any priori or label. We demonstrate the advantages of our approach by comparing it with the most recent work in terms of performance and accuracy.

Key words: facial pose recognition; facial pose estimation; real-time tracking

1 Introduction

Research on the human faces has been of great interests in computer vision and visual media processing for several decades. Most existing works focus on the sources of 2D intensity or colored images, which are always influenced by variations of poses, expressions, illuminations, and subordinates. It is still a challenging problem to develop a robust automatic three-dimensional face recognition system[1].

Recent developments[2–5] start to explore how 3D information can be recovered for facial modeling and analysis. 3D information, such as pose, depth, and the complete model, provides a promising way to understand the characteristics of the human face in the 3D domain and has the potential to improve the performance of current 2D-based approaches in both scientific and commercial fields. For example, Apple's animoji, which imitates animals by tracking users' head pose and expression in real time, has been widely used in public. Similarly, a popular face swapping app called Zao has emerged in China. This app enables users to pretend to star in blockbuster movies by uploading pictures and then creating simulated video clips by using machine learning. The demand of virtual makeup or hairdressing test, monitoring, and recognition of non-frontal faces also show the potential market of high-level information recover.

However, most of existing approaches have their own weakness. Some of them require more sophisticated acquisition equipment and redundant source input. Furthermore, some trade-off has to be made between efficiency and accuracy generally. Therefore, in this paper, we present a novel and efficient method for fast and accurate facial pose estimation in a single frame and extend the proposed method to real-time multiple tracking in video sequences. First, we aim to improve

• Xiaolong Yang and Xiaohong Jia are with the Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing 100190, China and also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: yangxiaolong17@mails.ucas.ac.cn; xhjia@amss.ac.cn.
• Mengke Yuan and Dong-Ming Yan are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: mkyuan@ia.ac.cn; yandongming@gmail.com.
* To whom correspondence should be addressed.

Manuscript received: 2019-12-31; accepted: 2020-01-02

the estimation accuracy as much as possible while maintaining high efficiency. Hence, we combine two simple Convolutional Neural Network (CNN) models of face detection and mean shape learning, which make an initial estimation of the input image efficiently. Furthermore, we design a bi-objective optimization strategy to iteratively refine the obtained estimation. The carefully-designed optimization function allows us to obtain fast and reliable results. Finally, to take full advantage of the abundance of information when videos or multiple images are used as input, we apply a series of algebraic filtering operations, such as Gaussian filter for background removal and extended Kalman filter for target prediction, to maintain real-time tracking performance. The main contributions of our approach are as follows:

- We propose a fast and reliable real-time facial pose estimation and tracking system that combines a coarse initialization with face detection and mean shape learning, an accuracy refinement step with bi-objective optimization strategy and efficient tracking by algebraic filtering processes.
- We require no traditional constraints on the input, which means that besides facial close-up, more general RGB data without any priori or labels, such as photo taken casually in common daily life and video clip from a movie, are also processable.

2 Related Work

We briefly review the most recent approaches for traditional face tracking, facial information recovery, pose estimation, and target tracking that are closely related to our work.

Traditional face tracking involves some facial pose estimation issues, but it does not solve the problem completely. Meyer et al.[6] used a commodity depth camera and registered a deformable face model with the measured depth information. Moreover, their particle swarm optimization and the iterative closest point algorithm can deal with the point cloud model. This method can accurately estimate facial poses, yet it is not suitable for monocular RGB camera and more generally requirement. Face2Face[7] is a popular real-time face capture and reenactment application. This work tracked facial expressions in both the source and target video by using non-rigid model-based bundling and a dense photometric consistency measure. They focused on lighting estimation and expression transfer (especially mouth synthesis) rather than facial pose estimation. Cao et al.[8] used a dynamic rigidity prior learned from realistic datasets to perform real-time monocular face tracking. They were able to solve challenging exaggerated facial expressions with stable head poses. However, they did not calculate the head pose, and only made the facial region less obtrusive by applying minor adjustments to the head. Saragih et al.[9] updated equations such as mean-shift over the landmarks but with regularization imposed through a global prior over their joint motion. Their method performs well in the task of generic face fitting but cannot handle the pose estimation problem successfully.

Facial information recovery aims to use high-level information to perform estimations and has gained great success in recent years. Yang et al.[2] proposed a method for head estimation from a single image without landmark or depth information. On the basis of regression and feature aggregation, they learned a fine-grained structure mapping, and their results outperformed many state-of-the-art methods in terms of efficiency. Chaudhuri et al.[3] addressed the problem of facial motion retargeting. By using CNN from 2D face images to learn a 3D Morphable Model (3DMM), they generated ground-truth 3DMM parameters, and performed joint face detection and motion retargeting for images with multiple faces. Their results exhibit high face detection accuracy and are robust in estimating extreme challenging poses. Gecer et al.[4] applied the capability of Generative Adversarial Networks (GANs) and deep CNNs, and used nonlinear optimization to reconstruct the facial texture and shape from single images. They achieved photorealistic and identity-preserving 3D face reconstructions for the first time. Wu et al.[5] addressed the problem of recovering the 3D geometry of a human face from images in multiple views. Compared with traditional 3DMM methods, they avoided the drawback in the single-view setting, and the synthetic projections from one view to another can better align with the observed image. However, the above approaches have their own weakness, i.e., the general trade-off between efficiency and accuracy. Multiview data were required in Refs. [2, 4] as input, such as multiview images or cameras, which cannot be used for direct video tracking. The results in Refs. [3, 5] had high accuracy, but the authors were unable to perform real-time processing due to the complexity of networks and the computation on multiple objects, which increased the computational burden.

Pose estimation has also received extensive attention recently. Kocabas et al.[10] proposed a self-supervised learning method and provided guidance for us to estimate poses for 3D humans by using multiple view geometry. Ge et al.[11] estimated the full 3D hand shape and pose from a single RGB image. This work is also helpful in the facial problem given the use of landmark and skeleton information of a hand model. Wang et al.[12] performed 6D object pose estimation from an RGB-D image and achieved near real-time inference. Peng et al.[13] addressed the problem of 6D object pose estimation from a single RGB image while facing challenging conditions such as occlusion or truncation. They also built a voting network for key point localization to avoid uncertainty when solving poses. Kumar and Chellappa[14] addressed the problem of disentangling 3D pose by designing a single dendritic CNN with heatmap regression and reduced the error of face alignment. Cao et al.[15] achieved inherent mapping between frontal and profile faces, whereas many existing works may only focus on frontal faces, thus enhancing the face recognition performance. Abbas et al.[16] proposed a method for facial morphological descriptors based on 3D geodesic path curvatures. However, this method was still applied only to more salient features, such as gender analysis. In contrast to all of the above mentioned works, our goal is to recover the facial pose information only by using general RGB sources as input and to achieve better performance.

Target tracking is an increasingly sophisticated technology. Xiang et al.[17] led the way for robust multitarget tracking with Markov decision processes. The accuracy of this method is not satisfactory enough but it lays a solid foundation for future work. Crivellaro et al.[18] learned to use extended Kalman filtering based on the interframe relation of video input. Although the proposed method solved the problems of tracking efficiency and target loss, the accuracy and abundance of results were limited to simple objects such as boxes due to a simple model design. Girdhar et al.[19] proposed a lightweight yet highly effective approach for estimating and tracking key points of human body in complex multiperson videos. This approach combined human skeleton tracking and video understanding, and achieved great performance. Song et al.[20] designed a three-stage real-time detector that responds well to small objects, such as traffic signs, especially those that are far from the camera, which is therefore challenging to traditional object detection methods. Our method benefits from the above advantages. With the use of sequential algebraic filtering processing, multitarget problem is transformed into a parallel single-target problem, and areas of concern are tracked with robust precision, thereby guaranteeing the efficiency of our method.

3 Our Approach

In this section, we detail the main steps of our approach, including fast initial pose estimation, accurate estimation of refined pose, and real-time tracking. Two sampled example of our algorithm are shown in Fig. 1.

3.1 Initial pose estimation

Many existing works pursue face recognition accuracy and the abundance of recovered 3D information, thereby leading to significant inefficiency when complicated network structures are used. On the basis of Ref. [21] (as shown in the left and middle parts of Fig. 2), we train a CNN called Landmark-CNN with a standard multiclass architecture. This CNN is trained with 68 landmark images from the 300-W dataset[22], and its output consists of the coordinates of the landmarks. This simple CNN model can help us perform face alignment quickly. Inspired by Ref. [23] (as shown in the middle and right portions of Fig. 2), we train another network called Pose-CNN, whose architecture is the same as that of Landmark-CNN, with a set of images that have different poses from the OFD, CASIA 3D Face (v1.0)[24] and VGG datasets[25]. The output consists of the Euler angle, which identifies the initial pose.

We use Keras with TensorFlow backend to implement those two CNNs. We apply random cropping and random

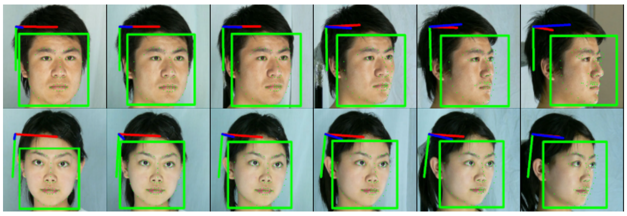


Fig. 1 Sampled outputs of our framework. Our algorithm takes single RGB frame as input, and produces the estimated pose parameters. The colored lines (green, blue, and red) indicate the direction of Euler angles (roll, pitch, and yaw) of the head. When multiframe images are tracked in realtime, it generates a hot spot of interest (green box) rather than using the entire image to improve the performance. Our algorithm also produces landmark prediction (green points) of tracked faces.

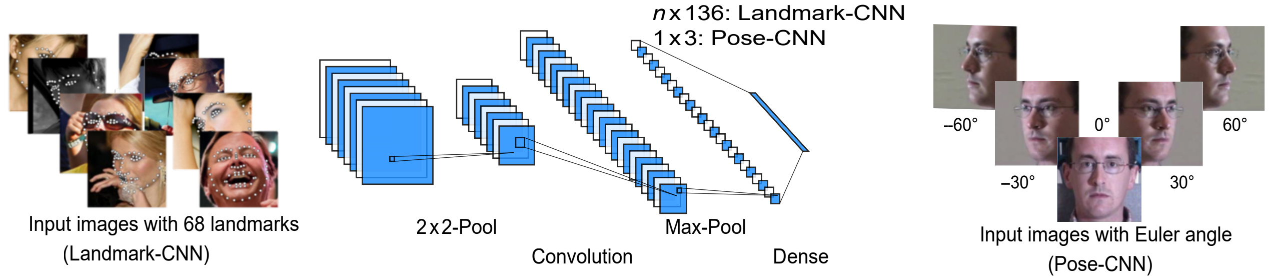


Fig. 2 Architecture of our Landmark-CNN and Pose-CNN. Landmark-CNN uses images with 68 landmarks for training and generates an $n\times136$ vector. Pose-CNN uses images with Euler angle for training and generates a 1×3 angle result.

scaling $(0.6-1.2)$ for data augmentation in training stage. We use 150 epochs to train the network with the Adam optimizer with an initial learning rate of 0.001. The learning rate is reduced by a factor of 0.1 every 10 epochs. When an image comes to Landmark-CNN, it generates a 1D vector $\mathcal{P}$ with $n\times136$ length, where n means the number of faces and 136 denotes the 2D coordinates of each of the 68 landmarks' locations. Then, each group k of 2D coordinates of the 68 landmarks, denoted by $P^{\text{CNN}}_{i,k\ (1\times2)} = (\mathcal{P}[136(k-1)+2i-1], \mathcal{P}[136(k-1)+2i])$ $(i=1,2,\dots,68$ and $k=1,2,\dots,n)$, enters the Pose-CNN and estimates the Euler angle $\theta^{\text{CNN}}_{k(1\times3)} = (\theta_{\text{roll}}, \theta_{\text{pitch}}, \theta_{\text{yaw}})$ for face alignment.

These two intermediate results, $P^{\text{CNN}}_{i,k}$ and θ^{CNN}_{k}, are not accurate enough, which will be further refined in later process. However, our model has a lightweight network structure, which promises better efficiency while maintaining almost the same recognition performance. Moreover, our approach, which is also based on deep learning, does not need to pay extensive effort on training or parameter redeployment.

3.2 Accurate pose refinement

In this step, we use a bi-objective optimization strategy to refine the results obtained in previous step (Section 3.1). Figure 3 shows how the landmarks on mean face shape (frontal face) change in the refinement process.

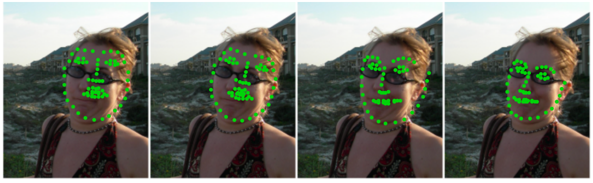


Fig. 3 Iterative refinement process. The input image to be processed (left), where the landmarks of mean face shape are drawn in green. As the optimization process proceeds (middle two images), landmark points transform to the final estimation location (right).

In the following discussion, we use a single face as an example with P^{CNN}_{i} and θ^{CNN}, and ignore the subscript k. We observe that pose estimation can be converted into landmark estimation, which can be formulated as an error approximation problem:

$$P_i^{Es}(z^*,\theta) = P_i(z^*)\cdot Tr(\theta) \tag{1}$$

where $P^{Es}_{i\ (1\times3)}$ is the coordinate of the i-th $(i=1,2,\dots,68)$ landmark and contains variables (z^*,θ). $P_{i\ (1\times3)} = (x,y,z^*)$ is the coordinate of the i-th landmark on mean face shape model with depth value z^*. $Tr(\cdot)\subset\mathbb{R}^{3\times3}$ is a rotation matrix generated by Euler angle $\theta=(\theta_r,\theta_p,\theta_y)$.

First, we define two error functions averaged on eye error (E_{aver}) and cumulative of 68-landmark location error (E_{cumu}):

$$E_{\text{aver}} = \frac{1}{12}\sum_{j=1}^{12}\max\{H(P^{Es}_{\text{eye},j},P^{\text{CNN}}_{\text{eye},j}),H(P^{\text{CNN}}_{\text{eye},j},P^{Es}_{\text{eye},j})\} \tag{2}$$

$$E_{\text{cumu}} = \sum_{i=1}^{68}\parallel P_i^{Es}-P_i^{\text{CNN}}\parallel_2 \tag{3}$$

where $P^{(\cdot)}_{\text{eye},j}\subset P^{(\cdot)}$ is the j-th landmark around the eye region $(j=1,2,\dots,12)$. E_{aver} is used to calculate the internal relations of the relevant eye-landmarks on the basis of Hausdorff distance function:

$$H(a,b) = \max_{a'\in set(a)}\ \min_{b'\in set(b)}\parallel a'-b'\parallel_2 \tag{4}$$

The reason for choosing points around eyes as the loss function is that the evaluation of Landmark-CNN loss is based on all the points, which may cause the optimization step to diverge or the convergence to oscillate. E_{aver} measures the location error of all landmarks, which is a metric of pose error.

With the unsolved (z^*,θ) taken as the optimization variable, the total error is a weighted sum of the above and alternatively optimizes the following problem:

$$\theta = \arg\min_{(z^*,\theta)}(E_{\text{aver}}+\lambda\cdot E_{\text{cumu}}) \tag{5}$$

where $\lambda = 0.18$ in our experiment is a trade-off parameter. In the iterative optimization process, we fix (z^*, θ) alternately and obtain the optimal solution only related to another variable. The initial value of θ is obtained from the initial pose estimation (Section 3.1), and when convergence is achieved or the total error reaches a predefined threshold, our refinement will be terminated. Finally, we obtain new estimation coordinates $P_i^{Es} (i = 1, 2, \ldots, 68)$ and the corresponding pose Euler angle θ.

Recall that we mentioned in Section 3.1, if more than one face are in the image, in another words, many groups of landmarks are present, then each of them can be calculated and processed separately in parallel. This parallel optimization further improves the algorithm's efficiency.

3.3 Real-time tracking

To extend our work to video sequence for real-time tracking, we have to improve the initial estimation efficiency instead of processing each frame separately. As discussed in Ref. [18], when targets are tracked across a video sequence, if a pose is estimated for a given frame, then it can be treated as an initial value of the pose prior for the next frame. Therefore, extended Kalman filter can be used for this purpose.

In our experiment, if the background changes significantly, then we will skip this data preprocessing step. However, the camera is stationary and a large amount of redundant background information is available, such as surveillance cameras, selfies, and movie shots. To reduce unnecessary computation, we clean the background by using Gaussian filter, i.e., we focus only on the regions where the motion change occurs during real-time tracking and crop the frame to reduce the range of objects.

Gaussian filter is a commonly used linear smoothing filter that is suitable for eliminating Gaussian noise and widely used in noise reduction of image processing. Inspired by this approach, we use Gaussian filter to judge background information. Generally speaking, this process obtains the weighted average of the whole image, and the value of each pixel is obtained by its weighted average and that of other pixel values in the neighborhood. In accordance with the differences in the value of each pixel between multiple images, we can focus on the areas that are dynamically changing. A 2D Gaussian function is defined as follows:

$$g_{x,y} = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6}$$

where (x, y) is the coordinate and σ is the standard deviation. In practical application, we need to discretize the Gaussian function. In case that the size of the window template is $(2k + 1) \times (2k + 1)$, the value of each pixel (i, j) is calculated as follows:

$$G_{i,j} = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}} \tag{7}$$

Extended Kalman filter is a recursive estimator, which means that only the estimated state from the previous time step and the current measurement are required to compute the estimation for the current state. In contrast to batch estimation techniques, no history of observation and/or estimation is required. Kalman filter is based on linear dynamical systems discretized in the time domain and a Markov chain built on linear operators perturbed by errors that may include Gaussian noise.

In our work, we consider the Gaussian value of each pixel $G_{i,j}$ in the hot spot after being calculated in previous steps as a state $\mathcal{X} = \{G_{i,j}\}$. At each discrete time increment (video frame), a linear operator is applied to the state to generate the new state, with some noise mixed in, and optionally some information from the controls on the system if they are known. Then, another linear operator mixed with more noise generates the observed outputs from the hidden state:

$$\mathcal{X}_k = \Gamma_k \mathcal{X}_{k-1} + \mathcal{M}_k \mathcal{V}_k + \delta_k \tag{8}$$

where $\mathcal{X}_k$ is the state of object at time k; Γ_k is the state transition model which is applied to the previous state $\mathcal{X}_{k-1}$; $\mathcal{M}_k$ is the control input model, which is applied to the control vector $\mathcal{V}_k$, and sometimes $\mathcal{M}_k = 0$ if no external constraint exists; and δ_k is the process noise, which is assumed to be drawn from a zero-mean multivariate normal distribution $\mathcal{N}(0, \mathcal{E}_k)$ with the covariance of the process noise $\mathcal{E}_k$. The extended Kalman filter can be written as a single equation; however, it is most often conceptualized as two distinct phases, namely, predict and update. Through the update of these two stages, we can always obtain the prediction of the change of the Gaussian value to lock the target region. More detailed information can be found in Ref. [18].

In addition to Gaussian filter and extended Kalman filter, we also use other algebraic filters including median filter, expansion operation, and backprojection to avoid broken or lost tracking targets. Figure 4 shows an example frame in our tracking pipeline and all results are produced in real-time.

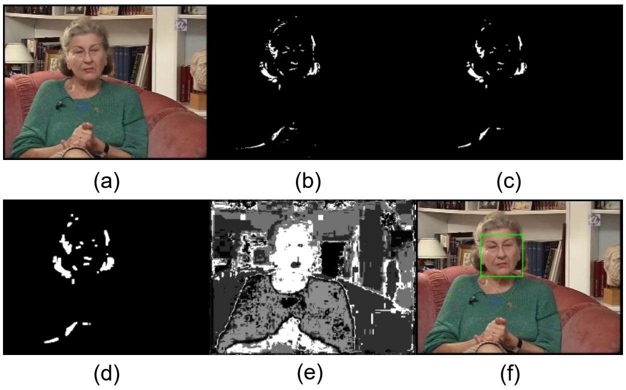


Fig. 4 Tracking pipeline. (a) Original frame, (b) Gaussian filter for background removal, (c) median filter for outlier removal, (d) expansion operation for pixel connection, (e) backprojection for object connection, and (f) tracking results. The focal area is cropped by a green box.

4 Experimental Result

In this section, we provide implementation details, datasets for training and testing, evaluation criterion, and comparison with several representative competing methods. All the results are conducted in Linux OS with GTX 1080Ti. We used Keras with TensorFlow 1.12 and CUDA 9.0 to implement the networks involved in all methods.

The well-known 300-W datasets for 2D face landmarks are adopted in our experiments, i.e., AFW, HELEN, IBUG, and LFPW[22]. These datasets contribute significantly to fast face alignment and initial estimation of landmarks. To infer more pose information, the pose variations training databases are also required. We use OFD, CASIA 3D Face (v1.0)[24] and VGG datasets[25] to improve our experimental design and obtain the best pose training results. In addition to RGB frames, the dataset also provides the 3D depth image. Figure 5 shows several examples sampled from these three datasets.

4.1 Evaluation criterion

We compare our approach with two state-of-the-art methods, i.e., Hopenet[26] and FSA[2], which only take RGB images or videos as input to generate recognition results in realtime. Hopenet proposes an elegant and robust way to determine pose by training a multiloss CNN and predict intrinsic Euler angles directly from image intensities rather than landmark-to-pose methods. The FSA approach can be seen as an extension of Hopenet. On the basis of the fine-grained structure of Hopenet, regression and feature aggregation are proposed to form a complementary ensemble.

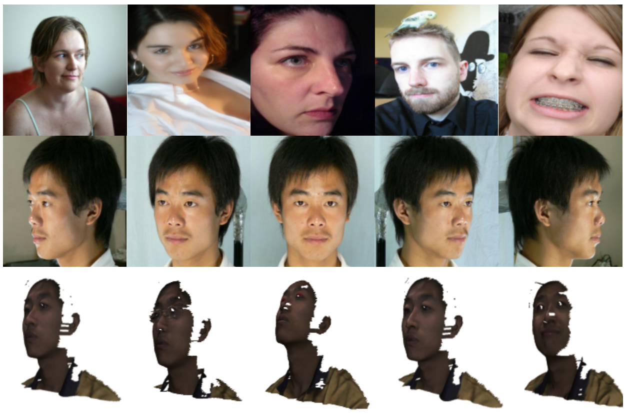


Fig. 5 Examples sampled from datasets. The first row is from the 300-W dataset and has an arbitrary camera pose with 68 landmarks. The middle row is from the OFD dataset, which contains a standard deflection angle. The last row is from the CASIA 3D Face dataset and includes a corresponding 3D model for different poses.

To demonstrate the advantages of our approach, we perform additional comparisons with earlier approaches. ERT[27] predicts landmark detection, accomplishes face alignment, and becomes a standard face library Dlib. FAN[28] is also a robust landmark detection method and merges block features multiple times on layers. KEPLER[29] is close to our work. It estimates facial landmarks and poses at the same time with a modified GoogleNet architecture. 3DDFA[30] matches a 3D model with an RGB image by CNNs and allows alignment of landmarks even with occlusion.

To measure the accuracy of pose estimation, we choose the differences of Euler angle (roll, pitch, and yaw) with ground truth as a criterion. To facilitate calculation, we propose a new relevant error measure, i.e., the Average Relative Error Margin (AREM), which more likely reflects the degree of confidence when the true value range is known:

$$\text{AREM} = \frac{1}{N}\sum_{N}\frac{\sum\limits_{x}^{\{\text{roll, pitch, yaw}\}}\left|\theta_x^{Est}-\theta_x^{GT}\right|}{90^\circ\times 3}\times 100\%,$$

where N is the number of samples; x indicates the direction as one of roll, pitch, and yaw; the superscripts *Est* and *GT* represent the estimation and the ground truth, respectively.

Unlike the compared landmark-free methods, our method can generate landmarks, which is an indirect but quantitative measure of the superiority of our results.

4.2 Comparison

We first show the visual comparisons with Hopenet and FSA, and then present qualitative comparisons of all competing methods. To demonstrate the robustness of our algorithm in the real-world environment, which may have poorly textured backgrounds or drastic light changes, we divide the test samples into lab-standard labeled data and challenging real data. Then, we conduct different experiments and give the corresponding result analysis respectively.

4.2.1 Lab-standard labeled situation

Figure 6 shows the pose estimation results of lab-standard labeled input samples. We select two groups (girl and boy) of photos taken at fixed angle intervals: ±90°, ±75°, ±60°, ±45°, ±30°, and 0°.

Time performance. Table 1 provides the time performance of all test examples shown in Fig. 6. The time cost involves initialization cost (for learning/invoking the model) and tracking cost. Our method has more lightweight network structures and better efficiency while maintaining almost the same recognition performance. Our approach does not spend much cost in neural networks, which improves the performance drastically.

Accuracy analysis. Because of the use of clean data, it is difficult to tell the difference between different methods shown in Fig. 6 with naked human eyes. Even so, as shown in Fig. 7, with the number of samples increasing, our approach still has a slight advantage in AREM. We also use another challenging dataset for comparison. We show that the other two methods perform unsatisfactorily when using naturally captured data as test samples, and our approach has much better accuracy.

4.2.2 Challenging natural data

We further verify the robustness of our approach using more challenging natural data. As shown in Fig. 8, we select some photos that are taken at an arbitrary angle or involve multiple targets. Besides, a movie sequence that is not a close-up of a human face is also selected to verify the applicability of algorithm.

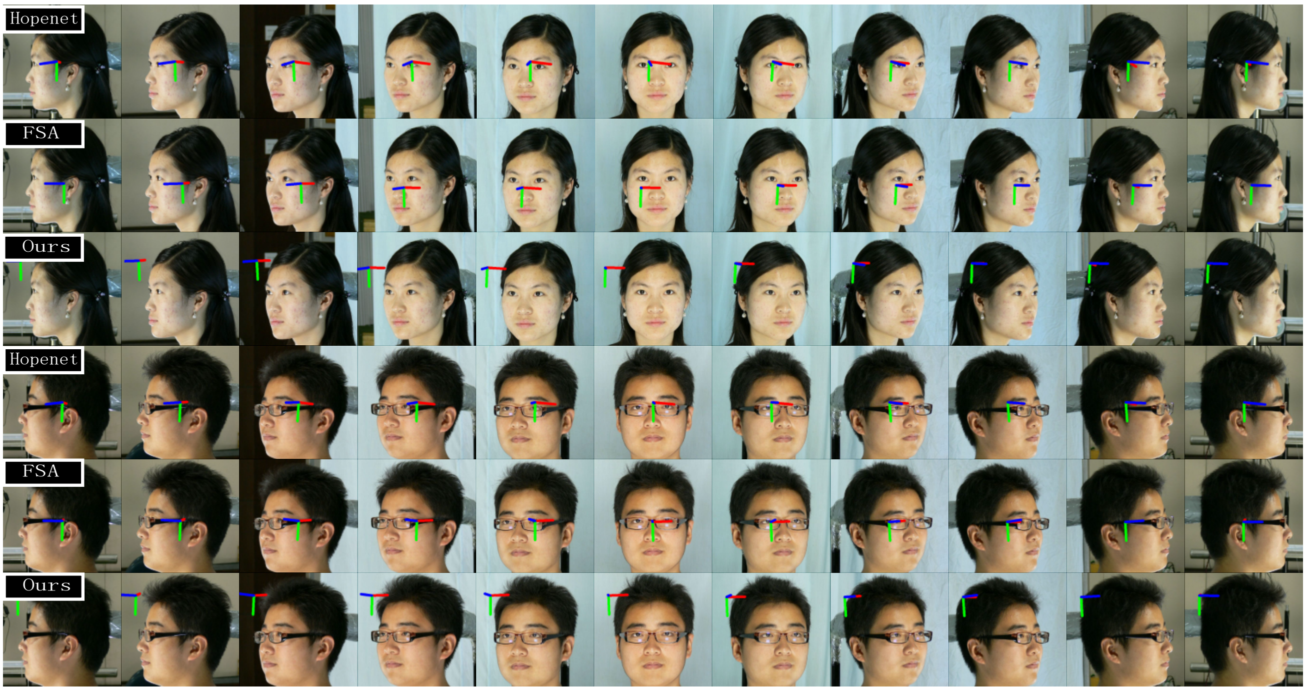


Fig. 6 Comparison in lab-standard labeled data input. Top to bottom: results of Hopenet, FSA, and the proposed method. The data used in this experiment are collected in the laboratory environment and precisely labeled at certain intervals. The head deflection angle is (from left to right): −90°, −75°, −60°, −45°, −30°, 0°, 30°, 45°, 60°, 75°, and 90° (a total of 11 poses).

Table 1 Time performance of lab-standard data.

(s)

Method	Tracking cost											Intialization cost	Total cost
	Pose 01: −90°	Pose 02: −75°	Pose 03: −60°	Pose 04: −45°	Pose 05: −30°	Pose 06: 0°	Pose 07: 30°	Pose 08: 45°	Pose 09: 60°	Pose 10: 75°	Pose 11: 90°		
Hopenet	1.68	1.20	1.30	0.96	0.61	1.00	0.15	0.61	0.32	0.48	0.44	12.53	21.28
FSA	1.62	0.65	0.73	0.94	0.17	0.28	0.08	0.08	0.03	0.05	0.07	7.31	12.01
Ours	0.11	0.07	0.03	0.02	0.03	0.01	0.02	0.01	0.03	0.02	0.01	2.95	**3.31**

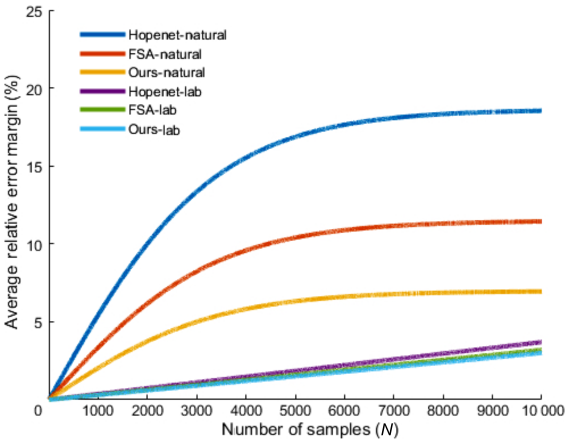


Fig. 7 AREM of two situations: Lab-standard labeled data and challenging natural data as input. The former database is so clean that the results are indistinguishable. The experiment on challenging datasets shows a significant difference.

All methods perform well on real-time single facial pose tracking, while our pose recognition is with higher accuracy. Moreover, our algorithm can still maintain real-time efficiency for the case of multitarget objects, while Hopenet needs more time and fails to track the girl's face in the first multitarget task. FSA's speed is faster than that of Hopenet but slower than ours, and it also fails to track the boy's face in the first multitarget task. Finally, our algorithm can also be used to locate landmarks as a by-product, which is significantly important in face alignment and expression capture. In the following analysis, we demonstrate through quantitative evaluation that our algorithm has better performance.

Time performance. Table 2 lists the time performance

Table 2 Time performance of the challenging dataset. (s)

Data type	Method	Intialization cost	Recognition/tracking cost				
Single frame	Hopenet	35.63	6.25	3.56	6.85	9.56	14.35
	FSA	12.47	0.77	0.32	0.41	1.21	3.83
	Ours	2.33	0.23	0.19	0.12	0.37	0.28
Video sequence	Hopenet	18.49	3.48	1.54	1.77	0.83	0.45
	FSA	6.88	1.96	0.57	0.33	0.15	0.08
	Ours	0.64	0.12	0.07	0.03	0.03	0.01

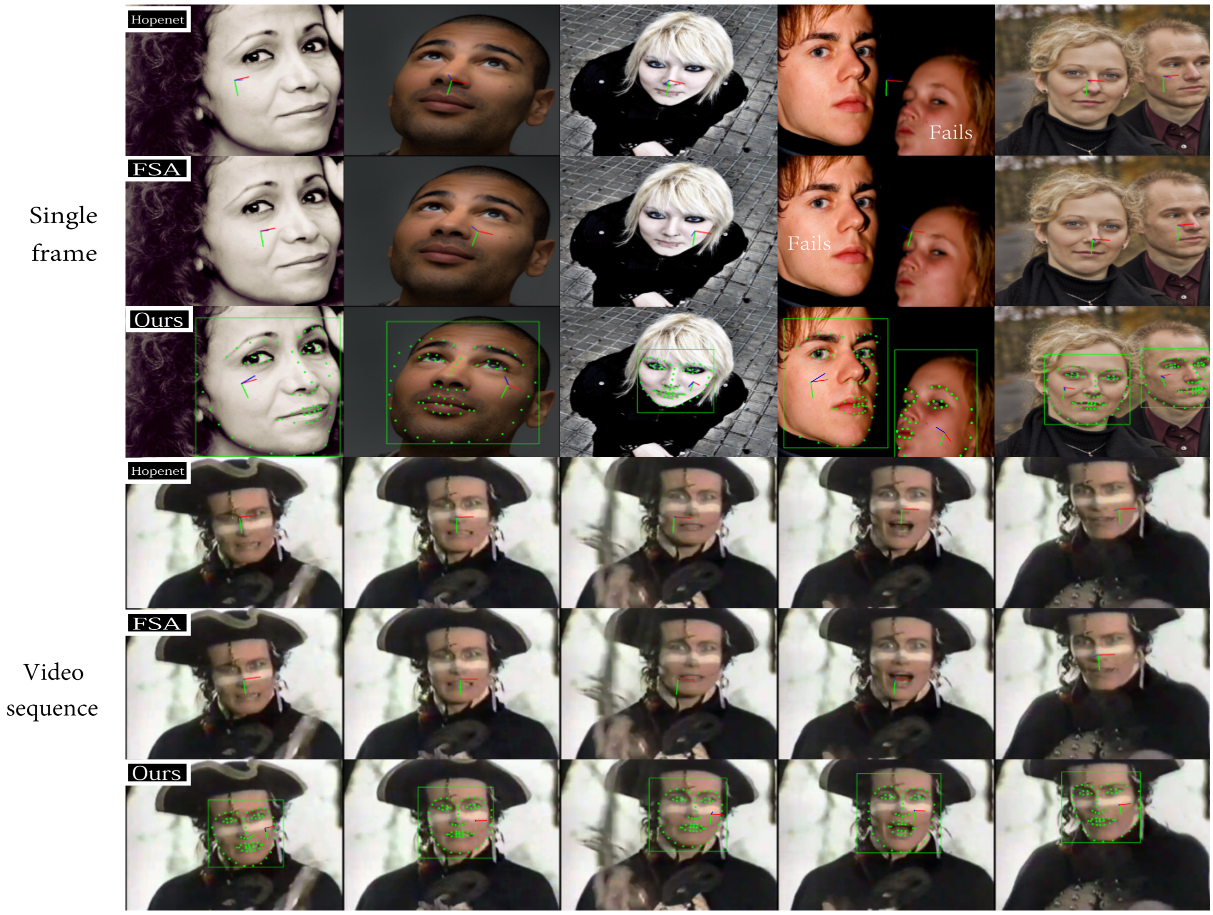


Fig. 8 Comparison in challenging naturally shot data input. We take the photos (top three rows: [single-target] S1–S3; [multitarget] M1 and M2) and video frames (bottom three rows: V1–V5) as input. Hopenet fails in the first multitarget task M1 with the girl's face and FSA also fails in the same task with the boy's face.

of all test examples shown in Fig. 8. The time cost involves intialization cost (for learning/invoking the model), recognition cost (for a single frame) or tracking cost (for video sequences). It shows that our method has a simpler network structure and better recognition efficiency, especially when tracking multiple targets.

Accuracy analysis. The statistic of accuracy performance is shown in Table 3. Besides pose error, landmark location can also validate the advantages of our results from another perspective. We predict the location by estimating the Euler angle of the pose; thus, our pose estimation results are highly reliable if landmarks are similar to the ground truth. We observe that the outcomes of our method outperform the others in terms of pose error. In the right column of Table 3, two types of additional error statistics for our method are provided, which were mentioned in Section 3.2. It also validates that our algorithm has reliable performance of landmark location in face alignment and expression capture.

Additional comparisons. To validate the superiority and robustness of our method, we conduct additional experiments on the AFLW2000 dataset with more methods. Table 4 shows the detailed experimental results, and Fig. 9 explains how AREM changes with the increase in test samples of each method.

Table 4 Additional compasions with representative methods on the AFLW2000 dataset. The training is performed on the 300-W dataset.

Method	Roll (°)	Pitch (°)	Yaw (°)	AREM (%)	Time (s)
ERT(68 points)[27]	10.61	13.18	23.53	17.48	144.3
FAN(12 points)[28]	8.35	6.71	12.40	10.17	274.7
KEPLER (GoogleNet)[29]	8.92	11.27	5.86	9.65	177.6
3DDFA (standard model)[30]	8.40	8.25	5.53	8.21	269.4
Hopenet (best $\alpha = 2$)[26]	5.47	6.44	6.56	6.84	135.9
FSA (best caps-fusion)[2]	4.50	6.64	4.08	5.63	71.7
Ours	**3.11**	**4.12**	**3.54**	**3.99**	**54.3**

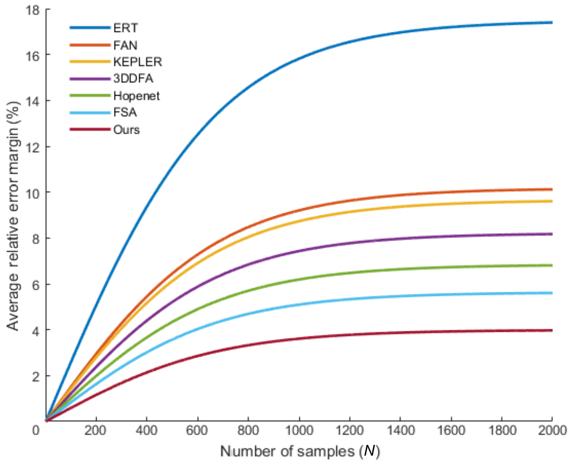


Fig. 9 AREM for different methods on the AFLW2000 dataset to determine the robustness of the methods to low-resolution images.

5 Conclusion and Future Work

We present a novel and efficient method for real-time multiple facial pose estimation and tracking in a single frame or video. First, we combine two simple standard CNN models of face detection and mean shape learning, and conduct an initial estimation of the input image. Then, we design a bi-objective optimization strategy to iteratively refine the obtained estimation. Finally, we introduce a series of algebraic filtering operations including Gaussian filter for background removal and

Table 3 Accuracy performance of the challenging dataset.

Input		Hopenet	FSA	Ours	Additional statistics	
					E_{aver}	E_{cumu}
Single frame	S1	(−10.4, 15.2, −10.5)	(−29.2, 2.5, −11.2)	(−18.0, 12.8, −10.5)	1.24	4.06
	S2	(15.5, 30.5, 12.6)	(21.0, 18.8, 20.2)	(21.6, 31.8, 5.7)	2.58	3.41
	S3	(5.8, −32.1, 1.9)	(12.0, −18.6, 6.9)	(3.1, −17.7, −2.8)	0.63	2.72
	M1	(−0.7, 1.5, −7.6)	**Failed**	(1.1, 10.4, −13.6)	2.96	5.44
		Failed	(41.3, 23.2, 20.1)	(26.7, 25.7, 61.2)	0.83	4.08
	M2	(−1.2, −3.3, 2.7)	(−9.2, −6.3, 5.2)	(−3.8, −9.3, 2.9)	0.91	1.37
		(−10.0, −2.5, −6.7)	(−1.8, −6.1, −2.6)	(−9.2, 1.8, −4.2)	0.95	2.79
Video sequence	V1	(−3.3, −10.3, −4.1)	(−2.3, −4.3, −8.1)	(−5.7, −6.2, −5.3)	0.55	4.08
	V2	(0.2, 1.7, 0.1)	(3.0, 1.4, 0.3)	(1.2, 1.2, 0.5)	0.58	2.56
	V3	(0.8, −0.2, 2.2)	(−6.6, −0.7, 4.1)	(1.7, −0.9, 5.2)	0.54	2.53
	V4	(0.3, −6.6, 1.2)	(3.1, −11.9, 2.8)	(2.1, −6.3, 2.4)	0.58	3.47
	V5	(−5.1, −15.8, 33.6)	(1.6, −8.8, 26.1)	(−8.8, −12.7, 24.9)	0.56	2.03
AREM (%)		18.63	11.49	**6.98**	–	

Note: S, M, and V are arranged in the order shown in Fig. 8; all $\Delta\theta(\text{roll}, \text{pitch}, \text{yaw}) = \theta^{Est} - \theta^{GT}$ are angle errors in degree (°) and numerical errors are in pixel-level.

extended Kalman filter for target prediction, to maintain real-time tracking performance. Our results are superior to those of state-of-the-art methods in terms of efficiency and accuracy.

In future work, we would like to focus on the improvement of the tracking algorithm to enhance its accuracy. Face detection or alignment under challenging environment will also be considered.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61872354, 61772523, 61620106003, and 61802406), the National Key R&D Program of China (No. 2019YFB2204104), the Beijing Natural Science Foundation (Nos. L182059 and Z190004), the Intelligent Science and Technology Advanced Subject Project of University of Chinese Academy of Sciences (No. 115200S001), and the Alibaba Group through Alibaba Innovative Research Program.

References

[1] Y. B. Hu, X. Wu, B. Yu, R. He, and Z. Sun, Pose-guided photorealistic face rotation, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8398–8406.

[2] T. Y. Yang, Y. T. Chen, Y. Y. Lin, and Y. Y. Chuang, FSA-Net: Learning fine-grained structure aggregation for head pose estimation from a single image, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 1087–1096.

[3] B. Chaudhuri, N. Vesdapunt, and B. Y. Wang, Joint face detection and facial motion retargeting for multiple faces, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 9719–9728.

[4] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, GANFIT: Generative adversarial network fitting for high fidelity 3D face reconstruction, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 1155–1164.

[5] F. Z. Wu, L. C. Bao, Y. J. Chen, Y. G. Ling, Y. B. Song, S. N. Li, K. N. Ngan, and W. Liu, MVF-Net: Multi-view 3D face morphable model regression, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 959–968.

[6] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz, Robust model-based 3D head pose estimation, in *Proc. IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015, pp. 3649–3657.

[7] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, Face2Face: Real-time face capture and reenactment of RGB videos, in *IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2387–2395.

[8] C. Cao, M. L. Chai, O. Woodford, and L. J. Luo, Stabilized real-time face tracking via a learned dynamic rigidity prior, *ACM Trans. Graph.*, vol. 37, no. 6, p. 233, 2018.

[9] J. M. Saragih, S. Lucey, and J. F. Cohn, Deformable model fitting by regularized landmark mean-shift, *Int. J. Comput. Vis.*, vol. 91, no. 2, pp. 200–215, 2011.

[10] M. Kocabas, S. Karagoz, and E. Akbas, Self-supervised learning of 3D human pose using multi-view geometry, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 1077–1086.

[11] L. H. Ge, Z. Ren, Y. C. Li, Z. H. Xue, Y. Y. Wang, J. F. Cai, and J. S. Yuan, 3D hand shape and pose estimation from a single RGB image, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 10833–10842.

[12] C. Wang, D. F. Xu, Y. K. Zhu, R. Martín-Martín, C. W. Lu, F. F. Li, and S. Savarese, DenseFusion: 6D object pose estimation by iterative dense fusion, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 3343–3352.

[13] S. D. Peng, Y. Liu, Q. X. Huang, X. W. Zhou, and H. J. Bao, PVNet: Pixel-wise voting network for 6DoF pose estimation, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 4561–4570.

[14] A. Kumar and R. Chellappa, Disentangling 3D pose in a dendritic CNN for unconstrained 2D face alignment, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 430–439.

[15] K. D. Cao, Y. Rong, C. Li, X. O. Tang, and C. C. Loy, Pose-robust face recognition via deep residual equivariant mapping, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 5187–5196.

[16] H. Abbas, Y. Hicks, D. Marshall, A. I. Zhurov, and S. Richmond, A 3D morphometric perspective for facial gender analysis and classification using geodesic path curvature features, *Comput. Vis. Media*, vol. 4, no. 1, pp. 17–32, 2018.

[17] Y. Xiang, A. Alahi, and S. Savarese, Learning to track: Online multi-object tracking by decision making, in *Proc. IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015, pp. 4705–4713.

[18] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, Robust 3D object tracking from monocular images using stable parts, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1465–1479, 2018.

[19] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, Detect-and-track: Efficient pose estimation in videos, in *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 350–359.

[20] Y. Z. Song, R. C. Fan, S. Huang, Z. Zhu, and R. F. Tong, A three-stage real-time detector for traffic signs in large panoramas, *Comput. Vis. Media*, doi: 10.1007/s41095-019-0152-1.

[21] S. Z. Zhu, C. Li, C. C. Loy, and X. O. Tang, Face alignment by coarse-to-fine shape searching, in *IEEE Conf. Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 4998–5006.

[22] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou,

and M. Pantic, 300 faces in-the-wild challenge: Database and results, *Image Vis. Comput.*, vol. 47, pp. 3–18, 2016.

[23] C. W. Luo, J. Y. Zhang, J. Yu, C. W. Chen, and S. J. Wang, Real-time head pose estimation and face modeling from a depth image, *IEEE Trans. Multimed.*, vol. 21, no. 10, pp. 2473–2481, 2019.

[24] CASIA-3D Face V1. Institute of Automation, Chinese Academy of Sciences (CASIA), http://biometrics.idealtest.org, 2019.

[25] O. M. Parkhi, A. Vedaldi, and A. Zisserman, Deep face recognition, presented at the British Machine Vision Conference, Swansea, UK, 2015.

[26] N. Ruiz, E. Chong, and J. M. Rehg, Fine-grained head pose estimation without keypoints, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops*, Salt Lake City, UT, USA, 2018, pp. 2074–2083.

[27] V. Kazemi and J. Sullivan, One millisecond face alignment with an ensemble of regression trees, in *IEEE Conf. Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 1867–1874.

[28] A. Bulat and G. Tzimiropoulos, How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks), in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 1021–1030.

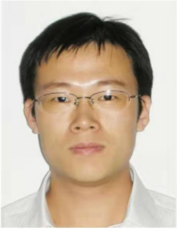
[29] A. Kumar, A. Alavi, and R. Chellappa, KEPLER: Keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors, in *Proc. 12^{th} IEEE Int. Conf. Automatic Face & Gesture Recognition*, Washington, DC, USA, 2017, pp. 258–265.

[30] X. Y. Zhu, Z. Lei, X. M. Liu, H. L. Shi, and S. Z. Li, Face alignment across large poses: A 3D solution, in *IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 146–155.

Xiaolong Yang received the BS degree in information and computing science from Northwestern Polytechnical University in 2017 and is currently pursuing the MS and PhD degrees at Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences. His research interest is on computer graphics and computer vision.

Mengke Yuan received the BS degree in applied mathematics, the MS degree in computational mathematics from Zhengzhou University in 2012 and 2015, respectively, and the PhD degree in computer sciences from Chinese Academy of Sciences in 2019. He is currently a post-doc at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). His research interest lies in image processing, computer vision, and computer graphics.

Xiaohong Jia received the PhD and bachelor's degrees from the University of Science and Technology of China in 2009 and 2004, respectively. She is an associate professor at Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS). Her research interests include computer graphics, computer aided geometric design, and computational algebraic geometry.



Dong-Ming Yan received the PhD degree from Hong Kong University in 2010, MEng and BEng degrees from Tsinghua University in 2005 and 2002, respectively. He is a professor at the National Laboratory of Pattern Recognition of the Institute of Automation, Chinese Academy of Sciences (CAS). His research interests include computer graphics, computer vision, geometric processing, and pattern recognition.