Machine Vision and Applications (2018) 29:375–391
https://doi.org/10.1007/s00138-017-0887-6

ORIGINAL PAPER

Annotated face model-based alignment: a robust landmark-free pose estimation approach for 3D model registration

Yuhang Wu[1] · **Shishir K. Shah**[1] · **Ioannis A. Kakadiaris**[1]

Received: 22 February 2017 / Revised: 24 July 2017 / Accepted: 2 September 2017 / Published online: 30 November 2017

Abstract Registering a 3D facial model onto a 2D image is important for constructing pixel-wise correspondences between different facial images. The registration is based on a 3×4 dimensional projection matrix, which is obtained from pose estimation. Conventional pose estimation approaches employ facial landmarks to determine the coefficients inside the projection matrix and are sensitive to missing or incorrect landmarks. In this paper, a landmark-free pose estimation method is presented. The method can be used to estimate the matrix when facial landmarks are not available. Experimental results show that the proposed method outperforms several landmark-free pose estimation methods and achieves competitive accuracy in terms of estimating pose parameters. The method is also demonstrated to be effective as part of a 3D-aided face recognition pipeline (UR2D), whose rank-1 identification rate is competitive to the methods that use landmarks to estimate head pose.

Keywords Pose estimation · Face alignment · Model registration · Face recognition

Abbreviations

GIS	Geometry image space
AFM	Annotated face model
T-AFM	Texture of annotated face model
RDD	Rotation determined decomposition
TBB	Target bounding box
SDM	Supervised descent method
GSDM	Global supervised descent method
RSSDM	Random subspace supervised descent method
2dSC	Two-dimensional sparse coding
G3D	Generic 3D model
PS3D	Personalized 3D model
E-AFMA	Ex-annotated face model-based alignment
AFMA	Annotated face model-based alignment

1 Introduction

Three-dimensional facial model registration is the process of aligning a 3D facial model onto a 2D facial image so that every 3D vertex in the model is projected to its corresponding location on the 2D image. After model registration, each 2D pixel on the facial image can be mapped back to its corresponding 3D position. After mapping all the 2D pixels back to the 3D model, a textured 3D model is obtained, which can be rendered for different applications such as face animation [8,36] and recognition [12,28,38].

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-BSH001. This grant is awarded to the Borders, Trade, and Immigration (BTI) Institute: A DHS Center of Excellence led by the University of Houston, and includes support for the project "Image and Video Person Identification in an Operational Environment: Phase I" awarded to the University of Houston. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

✉ Ioannis A. Kakadiaris
ikakadia@central.uh.edu

Yuhang Wu
ywu35@central.uh.edu

Shishir K. Shah
sshah@central.uh.edu

[1] Computational Biomedicine Lab, Department of Computer Science, University of Houston, 4849 Calhoun Road, Houston, TX 77004, USA

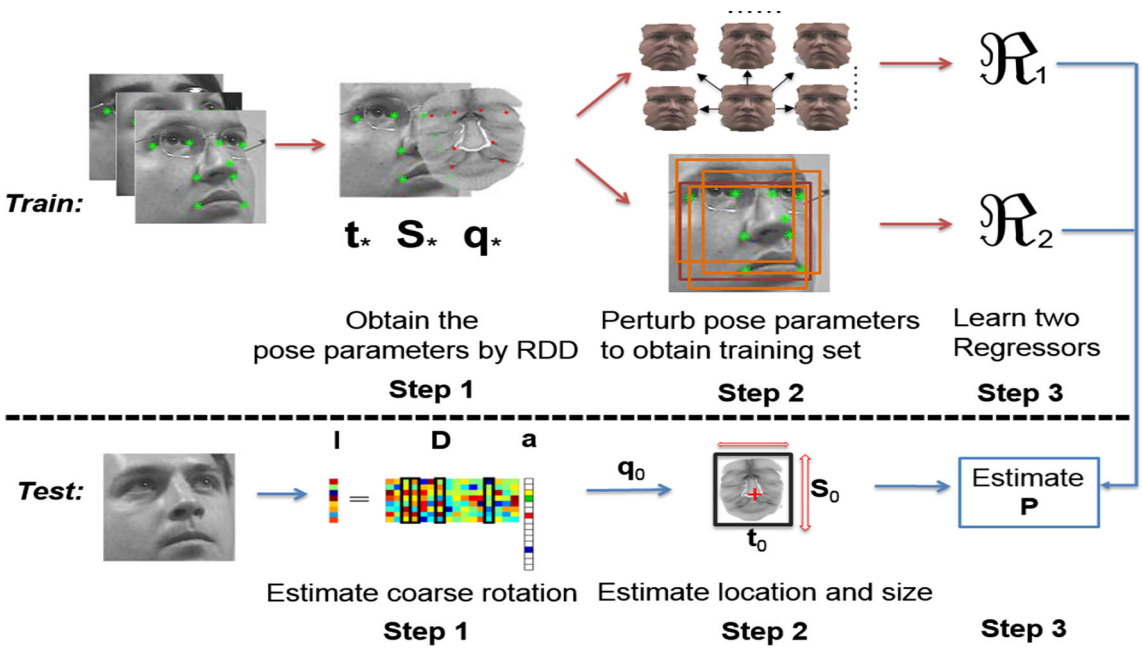

Fig. 1 Depicted is the overview of AFMA

Under a perspective projection model [18], 3D model registration relies on a 3 × 4 projection matrix to transform 3D vertices to 2D points. The coefficients inside the projection matrix are estimated by pose estimation. Conventional pose estimation approaches employ 2D and 3D facial landmarks to estimate the projection matrix by solving a least-squares minimization problem. However, automatically detected landmarks may have errors, especially in real-world applications. Landmark detection errors may lead to significant errors in pose estimation. In order to address this challenge, it is important to have a landmark-free pose estimation approach that is insensitive to landmark detection errors. Previous method [6,14,16,20,25,35] were able to estimate head orientation parameters. These methods exploit machine learning approaches to learn a mapping from a 2D image to the pose parameters. However, these approaches do not fully address existing challenges. First, they cannot estimate any poses that outside the quantized training set. Second, they cannot estimate the scaling and translation parameters of the 3D model, which are essential for model registration. Thus, a full projection matrix cannot be recovered by these methods.

In this paper, a new pose estimation method is proposed which is able to estimate a full projection matrix. The approach is derived from frontalization approaches in 3D-aided face recognition. This category of methods first registers a 3D model onto the 2D image, and then all the pixels on the face are mapped back to a Geometry Image Space (GIS) by rendering the facial texture into frontal pose [22,31]. In GIS, pixel-wise correspondences are constructed between different facial images so that face matching can be conducted in a space with much fewer variations caused by head pose changes. Based on the properties of GIS, it is observed that if a 3D model is perfectly registered to the 2D image, it will generate a frontalized facial texture without texture distortion. However, if the 3D model is not registered well on the 2D model, it will distort facial texture and generate distorted patterns. Motivated by this finding, a new pose estimation approach is proposed. It is able to estimate the projection matrix in model registration based on observing the distortion of facial texture on GIS. The overview of the proposed method is depicted in Fig. 1. A projection matrix is decomposed into 3D-rotation, 2D-scaling, and 2D-translation parameters based on a weak-perspective projection model. Multiple regressors are learned to iteratively estimate the parameters observing the feature derived from the GIS as well as the original image. In each regression stage, a personalized/generic 3D model is first aligned to the 2D image based on the current estimation of the projection matrix. Then, a frontalized facial texture is obtained and used to update the pose parameters. To produce a comprehensive evaluation, the proposed method is compared with both landmark-based and landmark-free pose estimation methods. A preliminary version of this work, ex-Annotated Face Model-based Alignment (E-AFMA), has appeared in Wu et al. [37]. An improved method, AFMA is presented in the current paper that solves the yaw regression problem, which was the main limitation of Wu et al. [37]. In addition, an improved solution for landmark-free pose estimation, which employs an additional regressor for adjusting the size and position of facial bounding box is presented. The paper also extends the work described in Wu et al. [37] by providing additional

experimental analysis and a more detailed description of the method.

The contribution of the work presented in this paper is a pose estimation approach that (i) does not require landmark detection; (2) can estimate poses that are not available in a quantized training database. The rest of the paper is organized as follows: The related work in pose estimation is presented in Sect. 2. AFMA is presented in Sect. 3. Experiments and analysis are presented in Sect. 4. The conclusions are drawn in Sect. 5.

2 Related work

Landmark-free pose estimation approaches can generally be divided into two categories: classification-based methods [14,25,35] and regression-based methods [7,16,44]. Classification-based methods are robust to small appearance noise but are constrained by the quantized space of training labels. They are not suitable for accurate 3D model registration. In contrast, regression methods [7,16,44] are able to estimate the rotation angles lying between the quantized grids. In this category of method, Zhen et al. [44] developed an accurate regression-based method through building a low-rank subspace between the target and input space with a supervised manifold regularization, which achieves state-of-the-art pose estimation accuracy in the Pointing 04 database [15]. Despite their advantages, most work in this area can only estimate the rotation parameters, regardless of the scaling and translation parameters encoded in the 3×4 dimensional projection matrix.

AFMA is built upon the work in facial landmark detection. Asthana et al. [3] proposed a method that decomposes facial shapes into 2D pose parameters and coefficients of shape bases, then estimate them by support vector regression (SVR) [13] chain. Xiong and De la Torre [39] proposed a cascaded approach where linear mappings are learned from a high-dimensional feature space to the landmark space to iteratively estimate facial landmark positions. Both methods employ the local patches extracted from landmarks as the input of regression. AFMA uses a different approach to [3,39], and it optimizes texture difference sampled from GIS instead of landmarks. Similar to Kemelmacher-Shlizerman and Seitz [24], a weak-perspective projection model is employed to decompose projection matrix into pose parameters. A sparse coding-based approach is employed in this paper, which is derived from Zhao et al. [43] and Yang et al. [42]. It initializes the pose parameters close to their target values at the beginning. From an application point of view, AFMA is similar to the method proposed by Jeni et al. [21], and their goal was automatic 3D model registration. However, AFMA does not rely on automatically detected landmarks for pose estimation.

AFMA employs a regression model [39], which has been demonstrated to be effective in solving nonlinear facial landmark regression problems. The model is modified by adding a regularization term on the step size to control the convergence speed. Instead of relying on the features extracted from the original image, AFMA mainly relies on the features extracted from the "texture of annotated facial model (T-AFM) [12,22,31]," where T-AFM comprises three channels of GIS. With the help of GIS, a correctly registered 3D model can be defined across all the pose variations.

3 Method

The intuition behind the landmark-free pose estimation problem is to decompose the general pose estimation problem into two subproblems according to their impacts on GIS, and solve them separately [37]. One problem is estimating pitch and yaw parameters; another problem is estimating the roll, scaling, and translation parameters. Since the pitch and yaw parameters determine the occlusion patterns of the face, they could be regarded as latent variables that generate the facial texture. The roll, scaling, and translation parameters control the 2D orientation, size, and position of the generated facial texture. To estimate their values, in the following subsections a correctly registered 3D model on T-AFM is firstly defined, and then a method is proposed to decompose the projection matrix into rotation, scaling, and translation parameters. In training, two regressors are learned to regress the parameters to target values based on the observations of T-AFM as well as the feature extracted from the facial ROI. One regressor $\Re_1$ is trained to estimate the pitch and yaw parameters, while another regressor $\Re_2$ is trained to estimate the roll, scaling, and translation parameters. In testing, given a facial ROI, the rotation of head pose is initialized through sparse coding, and then the two learned regressors are manipulated iteratively to register the 3D model to the 2D image. The final output of AFMA is a projection matrix $\mathbf{P}$, which can be used to align a 3D facial model to the 2D image.

3.1 Model formulation

Texture of annotated face model (T-AFM) Three-channel GIS, as shown in Fig. 2, is defined by Toderici et al. [23,31]. It is acquired in two stages. First, a 3D AFM (a smoothed facial model, proposed by Kakadiaris et al. [22]) is registered to a 2D image. Second, the facial texture is lifted from the original image to the T-AFM based on the vertices of the registered 3D model. Any texture lying on the same mesh of the aligned 3D model will be mapped to the same position on the T-AFM. With this geometric constraint, even though the faces may have different poses in the original images, they are frontalized in T-AFM. Notice that T-AFM can be regarded

Fig. 2 (T) Depicted are the original images, (B) depicted are the corresponding T-AFMs. The T-AFMs are cropped with a round mask to highlight the positions of eyes, nose, and mouth, which are pose-invariant

as a special case of GIS, which has different variations in recently proposed face frontalization methods [2,19,46].

To register a 3D model $\mathbb{M}$ to a 2D image, a projection matrix $\mathbf{P}$ is needed. Traditionally, it is acquired by solving a least-squares minimization problem with the help of 2D and 3D landmarks. Based on $\mathbf{P}$, the pixels of the original image are mapped into the T-AFM according to the 3D mesh in $\mathbb{M}$ through a texture lifting process g (please refer to [22] for more details). Let μ denote the index of an image, let $\boldsymbol{\Gamma}^{\mu}$ denote the T-AFM, the $\boldsymbol{\Gamma}^{\mu}$ could be obtained through the following operation:

$$\boldsymbol{\Gamma}^{\mu} = g(\mathbf{I}^{\mu}, \mathbf{P}^{\mu}, \mathbb{M}^{s}), \tag{1}$$

where $\mathbb{M}^{s}$ is a 3D AFM, s represents the index of the subject, and $\mathbf{I}^{\mu}$ is the face ROI for image μ. An accurate $\boldsymbol{\Gamma}^{\mu}$ can be obtained if both a personalized AFM $\mathbb{M}^{s}$ and a projection matrix $\mathbf{P}_{*}^{\mu}$ are available. The projection matrix $\mathbf{P}_{*}^{\mu}$ is usually estimated using manually annotated 2D and 3D landmarks. (All notations with a star in this paper are estimated from the manually annotated landmarks.) Unfortunately, in real applications, personalized 3D model $\mathbb{M}^{s}$ is hard to be accessible. An approximated $\boldsymbol{\Gamma}^{\mu}$ can be obtained by aligning a generic 3D AFM $\bar{\mathbb{M}}$ to the 2D image if the 2D landmarks are available. Nevertheless, in both circumstances, the accuracy of $\boldsymbol{\Gamma}^{\mu}$ relies on the estimation of $\mathbf{P}_{*}^{\mu}$.

Since it is assumed that 2D landmarks are not applicable in testing, $\mathbf{P}_{*}^{\mu}$ cannot be obtained in a conventional way. In this paper, a new method is proposed to leverage the current observation of $\boldsymbol{\Gamma}^{\mu}$ to update $\mathbf{P}^{\mu}$, and using the newly estimated $\mathbf{P}^{\mu}$ to update the $\boldsymbol{\Gamma}^{\mu}$. Due to the unique pose-invariant property of T-AFM, in ideal circumstances, given an $\mathbb{M}^{s}$, the distortion of T-AFM depends only on $\Delta\mathbf{P}^{\mu}$, the difference of $\mathbf{P}^{\mu}$ from the correct $\mathbf{P}_{*}^{\mu}$ rather than the current $\mathbf{P}$ itself. By learning a mapping from $\boldsymbol{\Gamma}^{\mu}$ to $\Delta\mathbf{P}^{\mu}$, the algorithm learns a *general* way to regress $\mathbf{P}$.

Rotation Determined Decomposition (RDD)[1] Instead of estimating $\mathbf{P}$ through least-squares minimization, it is decomposed into different functional units based on RDD. RDD extends the weak-perspective projection model employed in [3,4,24]. It contains two parameters that control the scale of the 3D model. Since the 3D model is rigid in this work, the non-rigid shape deformation term is left out.

Let $\mathbf{Y}$ denote a 2-by-n matrix representing the array of n 2D landmarks in an original image, $i \in \{1, \ldots, n\}$, let $\mathbf{X}$ denote a 3-by-n matrix representing the corresponding n 3D points in a personalized AFM. In the classical pose estimation framework, the 3×4 projection matrix $\mathbf{P}$ can be estimated based on Eq. 2:

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{1} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ \mathbf{1} \end{bmatrix}. \tag{2}$$

It can be further decomposed into the following matrices: an upper-triangular intrinsic matrix $\mathbf{K}$, an orthogonal 3×3 rotation matrix $\mathbf{R}$, and a 3×1 translation vector $\mathbf{e}$ as Eq. 3 represents:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{e}]. \tag{3}$$

However, since the $\mathbf{K}$ is unknown, the results of numerical decomposition of $\mathbf{P}$ (QR-decomposition) are not unique; if any column of $\mathbf{K}$ is modified, and so does the corresponding column in the $[\mathbf{R}|\mathbf{e}]$, the $\mathbf{P}$ remains the same. There exists a unique way to rotate an AFM to a pose where it shares the same look-up direction with the provided 2D image. This rotation matrix is denoted as the intrinsic rotation matrix denoted by $\mathbf{O}$. $\mathbf{O}_{*}$ can be determined using manually annotated landmarks through RDD.

The RDD can be formulated as the following equation:

$$\mathbf{Y} = \mathbf{CX} + \mathbf{t} = \mathbf{SOX} + \mathbf{t}. \tag{4}$$

The intuition of RDD is as follows: First, a 3D model is rotated by $\mathbf{O}$. Then, the model is projected into a 2D space using $\mathbf{S}$, which is a 2×3 matrix. Finally, the projected 3D model is translated to the position as it appears on the image based on a 2D translation vector $\mathbf{t}$. Compared with Eq. 2, this model is straightforward and similar to template matching. The term $\mathbf{SOX}$can be regarded as a deformable template, the two components t_x and t_y in $\mathbf{t}$ help determine the position of the template in a 2D image, and ($\mathbf{SO}$) controls the deformation of the template.

In RDD, a 2×3 intrinsic projection matrix $\mathbf{S}$ is used to project the rotated model from 3D space to 2D space, a different approach to use a 3×3 camera matrix $\mathbf{K}$ in the classical

[1] In the description of RDD, index μ is omitted for clarity.

model. Ideally, the values of $\mathbf{S}(1,1)$ and $\mathbf{S}(2,2)$, denoted as s_x and s_y, are nonzero. The pitch, yaw, and roll angle of the face in RDD are denoted by q_x, q_y, and q_z, respectively. A rotation vector is defined by: $\mathbf{q} = [q_x, q_y, q_z]$. The 3×3 matrix $\mathbf{O}$ can be decomposed into two basic matrices by Eq. 5:

$$\mathbf{O}(\mathbf{q}) = \mathbf{O}_x(q_x)\mathbf{O}_y(q_y)\mathbf{O}_z(q_z). \tag{5}$$

In the following, $\mathbf{O}(\mathbf{q})$ is used to denote the $\mathbf{O}$computed from Eq. 5 based on q_x, q_y, q_z. The general pose vector is denoted as $\mathbf{b} = [q_x, q_y, q_z, s_x, s_y, t_x, t_y]$, which is the target of AFMA. Vector $\mathbf{b}_* = [q_{x_*}, q_{y_*}, q_{z_*}, s_{x_*}, s_{y_*}, t_{x_*}, t_{y_*}]$ is used to represent the $\mathbf{b}$ estimated from manually annotated 2D and 3D landmarks.

In an annotated database, $\mathbf{Y}$ and $\mathbf{X}$ are available. When solving an inverse problem to estimate $\mathbf{O}(\mathbf{q}_*)$, the mean points of any two points in $\mathbf{Y}$ and $\mathbf{X}$ are computed to suppress the potential Gaussian noise in manual annotation [11]. The corresponding sets composed by mean points are denoted as $\bar{\mathbf{Y}}$ and $\bar{\mathbf{X}}$. Let $\bar{\mathbf{y}}$ denote a single 2D point in $\bar{\mathbf{Y}}$; let i and j denote the indices of the landmarks. Then, a 2D vector $\bar{\mathbf{y}}^i - \bar{\mathbf{y}}^j (i \neq j, i, j \in \{1, \ldots, n\})$ is computed for every pair of landmarks in $\bar{\mathbf{Y}}$ to generate $\mathbf{J}$. Correspondingly, vector $\mathbf{L}$ is constructed based on the same pair-wise operations on $\bar{\mathbf{X}}$. Hereby, Eq. 4 is simplified into Eq. 6:

$$\mathbf{J} = \mathbf{CL} = \mathbf{SO}(\mathbf{q})\mathbf{L}. \tag{6}$$

The term ($\mathbf{SO}$) is denoted by $\mathbf{C}$, which can be obtained using a standard least-squares minimization method based on $\mathbf{J}$ and $\mathbf{L}$. To disentangle the 2×3 intrinsic projection matrix $\mathbf{S}$and the orthogonal matrix $\mathbf{O}(\mathbf{q}_*)$from $\mathbf{C}$, following [24], a row is added to $\mathbf{C}$and its values is set to be the cross product between the first two rows. In this paper, an extended matrix is denoted by $\mathbf{C}'$. Then, the extended matrix $\mathbf{C}'$ is decomposed into $\mathbf{U}\boldsymbol{\Omega}\mathbf{V}^T$ through SVD, whereby the pose parameters can be obtained through the following equations:

$$\begin{aligned} \mathbf{O}(\mathbf{q}_*) &= \mathbf{U}\mathbf{V}^T \\ \mathbf{S}_* &= \mathbf{C}(\mathbf{O}(\mathbf{q}_*))^{-1} \\ \mathbf{t}_* &= \bar{\mathbf{Y}} - \mathbf{S}\mathbf{R}\bar{\mathbf{X}}. \end{aligned} \tag{7}$$

In Eq. 6, the decomposition of $\mathbf{C}'$ is independent of $\mathbf{t}$. Therefore, the $\mathbf{O}(\mathbf{q}_*)$ can be uniquely determined.

Conversely, when the actual 2D landmarks are not available, given $\mathbf{X}$ and an estimation of $\mathbf{b}$, Eq. 4 can be used to synthesize virtual 2D landmarks ($\hat{\mathbf{Y}}$). These virtual 2D landmarks can be used to estimate $\mathbf{P}$ based on Eq. 2. Hereby, the connection between RDD and the classical pose estimation approach is built up.

3.2 Training

In training, two regressors are learned. Regressor $\Re_1$ is trained to estimate the pitch and yaw. Regressor $\Re_2$ is trained to estimate the other five parameters in $\mathbf{b}$. To train these regressors, training samples need to be generated to simulate the misalignment in 3D registration. The training samples are augmented by perturbing the parameters inside $\mathbf{b}$. Let μ denote the image index ranging from 1 to Z (Z is the number of images in the training set) and ν denote the index of perturbation. The specific procedures for generating the training samples are shown in Alg. 1.

Data augmentation To train $\Re_1$, T-AFMs are generated with different offsets to simulate the inaccurate 3D registrations by perturbing both q_x and q_y several times and keep the roll, scaling, and translation parameters: $[q_{z_*}, s_{x_*}, s_{y_*}, t_{x_*}, t_{y_*}]$ unchanged. In this paper, both pitch and yaw are perturbed 15 times with offset: $\{-7° : 1° : 7°\}$. For each T-AFM, it is divided into overlapping patches and the image sharpness score FISH [33] is computed on each patch. FISH is employed in AFMA for its superior performance in describing the local distortion on GIS. All the scores are concatenated on T-AFM into a feature vector and use it as the input of $\Re_1$. In Fig. 3, the scores computed on different T-AFMs are visualized.

In the preliminary version of this work, $[s_x, s_y]$ and $[t_x, t_y]$, which were denoted as $\mathbf{S}_0^{\mu,\nu}$ and $\mathbf{t}_0^{\mu,\nu}$, were estimated based on the size and location of the face ROI, respectively. The q_z was forced to be zero. However, it was observed that the bounding box of face ROI can hardly be aligned to the face perfectly and the size may change case-by-case. Therefore, the pose parameters $[q_z, s_x, s_y, t_x, t_y]$ could not be fully determined by the input face ROI. In this work, a new bounding box is generated named the "target" bounding box (TBB), a regressor $\Re_2$ is learnt to update $[q_z, s_x, s_y, t_x, t_y]$ based on TBB. The previous method [37] is downgraded to initialize the TBB in testing. To generate training samples, keeping $[q_{x_*}, q_{y_*}]$ unchanged, random values are added on $[q_{z_*}, s_{x_*}, s_{y_*}, t_{x_*}, t_{y_*}]$ to generate 25 perturbed samples per image. To create the TBB for each perturbed sample, $\mathbf{X}^s$ is projected onto the 2D space using the perturbed pose parameters to synthesize virtual 2D landmarks. Assisted by these synthesized landmarks, a 2D bounding box (the inner rectangle in Fig. 4) is created. This bounding box is enlarged by a specific scale. The scale value is computed by using the

Algorithm 1 Data augmentation in an annotated database

Input: $\mathbf{Y}^\mu$ and $\mathbf{X}^s (\mu = 1 : Z)$
Output: $q_{x_0}^{\mu,\nu}$, $q_{y_0}^{\mu,\nu}$, $q_{z_0}^{\mu,\nu}$, $\mathbf{S}_0^{\mu,\nu}$, and $\mathbf{t}_0^{\mu,\nu}$
Step 1: Obtain the pose parameters in $\mathbf{b}_*$ by RDD
Step 2: Perturb $q_{x_*}^{\mu}$ and $q_{y_*}^{\mu}$ to obtain $q_{x_0}^{\mu,\nu}$ and $q_{y_0}^{\mu,\nu}$
Step 3: Perturb $q_{z_*}^{\mu}$, $\mathbf{S}_*^{\mu}$, and $\mathbf{t}_*^{\mu}$ to obtain $q_{z_0}^{\mu,\nu}$, $\mathbf{S}_0^{\mu,\nu}$, and $\mathbf{t}_0^{\mu,\nu}$

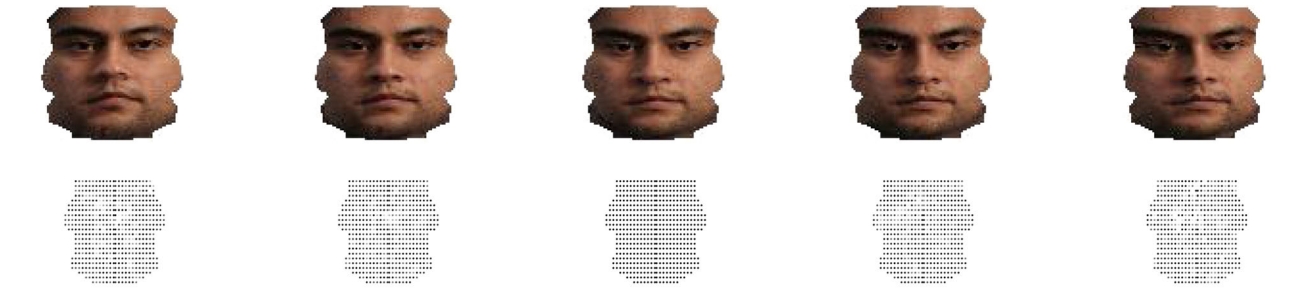

Fig. 3 The FISH score computed on the T-AFMs. The first row depicts the T-AFM under the corresponding pitch and yaw errors shown as follows: **a** Δpitch $= 12°$, Δyaw $= 0°$, **b** Δpitch $= 6°$, Δyaw $= 0°$, **c** Δpitch $= 0°$, Δyaw $= 0°$, **d** Δpitch $= 0°$, Δyaw $= 6°$, **e** Δpitch $= 0°$, Δyaw $= 12°$. The second row visualizes the difference between the perturbed T-AFM and the aligned T-AFM. Each dot depicts a FISH score extracted from one local patch of the image. The column **c** depicts a T-AFM generated by a well-registered 3D model without pitch or yaw errors. The grayscale of each dot depicts the absolute difference between two FISH scores extracted from the perturbed T-AFM and the well-registered T-AFM

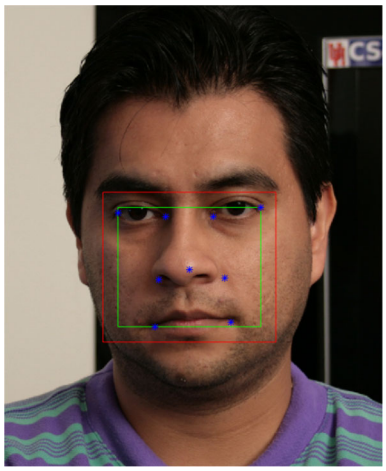

Fig. 4 Depicted is the face region for estimating pose parameters $[q_z, s_x, s_y, t_x, t_y]$. The blue dots depict the synthesized virtual landmarks, and the inner rectangle is determined by the outer three/four landmarks. The target bounding box (TBB), depicted by the red rectangle, is obtained through enlarging the inner bounding box by a product of an enlarged factor and the average value of s_x and s_y (color figure online)

average value of s_x and s_y multiplied by an enlarged factor: ξ_1. Finally, the TBB (outer rectangle in Fig. 4) is obtained. The histogram of oriented gradients (HOG) [9] feature is extracted from the whole TBB as the input of $\Re_2$.

Supervised descent method The engine of the regressor that employed in this paper is the supervised descent method (SDM) [39]. SDM has been widely used for landmark localization and is demonstrated to be effective in solving nonlinear regression problems. Specifically, let k denote the index of iteration. Descent Maps $\mathbf{Q}_k$ are learned by solving Eq. 8 based on multivariate linear regression. Pose vector $\mathbf{b}$ is updated with Eq. 9:

$$\arg\min_{\mathbf{Q}_k} \sum_{\mu} \sum_{\nu} ||\mathbf{b}_*^{\mu} - \mathbf{b}_k^{\mu,\nu} - \mathbf{Q}_k \Delta \boldsymbol{\Lambda}_k^{\mu,\nu}||^2; \tag{8}$$

$$\mathbf{b}_k^{\mu,\nu} = \mathbf{b}_{k-1}^{\mu,\nu} + \lambda \Delta \mathbf{b} = \mathbf{b}_{k-1}^{\mu,\nu} + \lambda \mathbf{Q}_{k-1} \Delta \boldsymbol{\Lambda}_{k-1}^{\mu,\nu}; \tag{9}$$

$$\Delta \boldsymbol{\Lambda}_k^{\mu,\nu} = \boldsymbol{\Lambda}_k^{\mu,\nu} - \bar{\boldsymbol{\Lambda}}_*. \tag{10}$$

where $\boldsymbol{\Lambda}_k^{\mu,\nu}$ stands for the features extracted from $\boldsymbol{\Gamma}_k^{\mu,\nu}$ or the TBB. $\bar{\boldsymbol{\Lambda}}_*$ represents a mean feature derived from all the aligned $\boldsymbol{\Gamma}_k^{\mu,\nu}$ or TBB using manually annotated landmarks in training. Since the original method is converged quadratically, a learning parameter λ is added to the original equation to control the speed of convergence.

Even though the SDM has been demonstrated to be effective in solving nonlinear regression problems, it has some limitations. In many applications, the energy function of SDM might have several local minima in a relatively small neighborhood. The standard SDM would average the local gradients in conflicting directions and lead to undesirable performance. In addition, the standard SDM learns a mapping from very high-dimensional input features to a low dimensional database, which easily overfits to the training samples. To address these challenges, Xiong and De la Torre [40] extended the SDM by dividing the search space of the original SDM into multiple regions and learning different Descent Maps from each. Through this subspace division process, the Global Supervised Descent Method [40] (GSDM) overcomes the conflicting gradient problems. In a more recent work, Yang et al. [41] considered the problems from a different aspect. Instead of learning the $\mathbf{Q}_k$ from the whole input feature vector, the method divides the feature vector into several sub-vectors randomly and learns different DM from each. The experimental results show that the method improved the performance in 2D landmark estimation. They named the method Random Subspace SDM (RSSDM).

Algorithm 2 $\Re_1$

Input: $\mathbf{I}^{\mu}$, $\mathbb{M}^{s}$, $\mathbf{Y}^{\mu}$, $\mathbf{X}^{s}$, $q_{x_0}^{\mu,\nu}$, $q_{y_0}^{\mu,\nu}$, $q_{z_*}^{\mu,\nu}$, $\mathbf{t}_{*}^{\mu,\nu}$, and $\mathbf{S}_{*}^{\mu,\nu}$
Output: $\mathbf{Q}^1$

while $\sum_{\mu}\sum_{\nu}||\mathbf{b}_k^{\mu,\nu}-\mathbf{b}_{k-1}^{\mu,\nu}|| \leq$ threshold **do**
  Synthesize the 2D landmarks $\hat{\mathbf{Y}}_k^{\mu,\nu}$ using Eq. 4
  Estimate the $\mathbf{P}_k^{\mu,\nu}$ using Eq. 2
  Obtain the T-AFM using $\boldsymbol{\Gamma}_k^{\mu,\nu}=g(\mathbf{I}^{\mu},\mathbf{P}_k^{\mu,\nu},\mathbb{M}^{s})$
  Compute the feature $\Delta\boldsymbol{\Lambda}_k^{\mu,\nu}$ (FISH) on $\boldsymbol{\Gamma}_k^{\mu,\nu}$
  Estimate the $\mathbf{Q}_k^1$ for $\Re_1$ using Eq. 8
  Update $\mathbf{b}_k^{\mu,\nu}$ using Eq. 9
  k=k+1
end while

Algorithm 3 Learn the parameters of $\Re_2$

Input: $\mathbf{I}^{\mu}$, $\mathbb{M}^{s}$, $\mathbf{Y}^{\mu}$, $\mathbf{X}^{s}$, $q_{x_*}^{\mu,\nu}$, $q_{y_*}^{\mu,\nu}$, $q_{z_0}^{\mu,\nu}$, $\mathbf{t}_{0}^{\mu,\nu}$, and $\mathbf{S}_{0}^{\mu,\nu}$
Output: $\mathbf{Q}^2$

Add some random noise to $q_{x_*}^{\mu,\nu}$, $q_{y_*}^{\mu,\nu}$, keep them unchanged in the following steps
while $\sum_{\mu}\sum_{\nu}||\mathbf{b}_k^{\mu,\nu}-\mathbf{b}_{k-1}^{\mu,\nu}|| \leq$ threshold **do**
  Synthesize the 2D landmarks $\hat{\mathbf{Y}}_k^{\mu,\nu}$ using Eq. 4
  Synthesize a TBB based on $\hat{\mathbf{Y}}_k^{\mu,\nu}$
  Compute the feature $\Delta\boldsymbol{\Lambda}_k^{\mu,\nu}$ (HOG) on the TBB.
  Estimate the $\mathbf{Q}_k^2$ for $\Re_2$ using Eq. 8 and subspace partitions
  Update $\mathbf{b}_k^{\mu,\nu}$ using Eq. 9
  k=k+1
end while

Learning the parameters of regressors To learn $\Re_1$, the feature is extracted from T-AFM ($\boldsymbol{\Gamma}_k^{\mu,\nu}$). According to Eq. 1, the projection matrix $\mathbf{P}_k^{\mu,\nu}$ is needed to compute $\boldsymbol{\Gamma}_k^{\mu,\nu}$. To compute the $\mathbf{P}_k^{\mu,\nu}$ from the $\mathbf{b}_k^{\mu,\nu}$, $\mathbf{O}(\mathbf{q}_k^{\mu,\nu})$ is estimated according to Eq. 5 based on $q_{x_0}^{\mu,\nu}$, $q_{y_0}^{\mu,\nu}$, and $q_{z_*}^{\mu,\nu}$. Then, the virtual 2D landmarks $\hat{\mathbf{Y}}_k^{\mu,\nu}$ are synthesized using Eq. 4 based on $\mathbf{O}(\mathbf{q}_k^{\mu,\nu})$, $\mathbf{t}_*^{\mu}$, and $\mathbf{S}_*^{\mu}$. Finally, $\mathbf{P}_k^{\mu,\nu}$ is estimated by solving a least-squares minimization problem using Eq. 2. With $\mathbf{P}_k^{\mu,\nu}$, $\boldsymbol{\Gamma}_k^{\mu,\nu}$ can be computed according to Eq. 1. The feature $\boldsymbol{\Lambda}_k^{\mu,\nu}$ is extracted from the $\boldsymbol{\Gamma}_k^{\mu,\nu}$ as follows: First, $\boldsymbol{\Gamma}_k^{\mu,\nu}$ is divided into two horizontal symmetric parts. Under the current pose, the part that is fully visible to the observer is nominated as the "visible component," which can be inferred by $\mathbf{b}_k^{\mu,\nu}$. The FISH scores on each part are computed and concatenated separately into two different feature vectors. The whitening-PCA is employed to reduce the dimension (98% energy preserved) of each feature vector. This feature vector is subtracted by a mean feature of $\boldsymbol{\Gamma}_*^{\mu,\nu}$ (derived from all the perfectly aligned T-AFMs in training) denoted as $\bar{\boldsymbol{\Lambda}}_*$ and finally forms the $\Delta\boldsymbol{\Lambda}_k^{\mu,\nu}$. Note that when regressing the pitch, the feature vector in the visible component is exploited, while regressing the yaw, the concatenated feature vector of both components is employed, which gives out the best performance. The algorithm for training $\Re_1$ is shown in Alg. 2.

Training $\Re_2$ is based on the same framework of training $\Re_1$; the algorithm is shown in Alg. 3. Different from $\boldsymbol{\Gamma}$, which is a frontalized space, the feature extracted from the TBB may be strongly affected by the pose variations. Therefore, multiple regressors are learned in each cascade stage based on the search space division method proposed in [40]. In this paper, the search space is separated into 16 areas. Four generated based on each of the signs of $q_{x_*}^{\mu,\nu}$ and $q_{y_*}^{\mu,\nu}$, and four generated via each of the top two principal components of the feature space. The readers are refereed to the original paper of GSDM [40] for more details about the subspace partition method. For consistency reasons, notation $\mathbf{Q}_k^2$ is employed to indicate the parameters of all GSDM regressors in stage k.

3.3 Testing

The algorithm of testing is depicted in Alg. 4.

Step 1: Given $\mathbf{I}$, an initial value of $\mathbf{q}$ is estimated, which is denoted by $\mathbf{q}_0$. Notice that recent research [14,44] indicates that using multiple adjacent soft labels to determine the pitch and yaw angle provides better results. In this work, an efficient sparse coding approach is proposed to obtain an initial pose.

The sparse coding (SC) approach has been widely used in face recognition [34]. However, few works extend it to solve a two-dimensional classification problem (e.g., pose estimation). Similar to Masi et al. [26], a dictionary $\mathbf{D} \in \mathbb{R}^{\theta\times wh}$ is designed, where each column is a θ-dimensional feature vector derived from the ROIs of a reference database. The columns are represented by $\mathbf{d}_s^{\sigma}$, where s is the element that belongs to the sth subject, and σ is the σth pose in a quantized pose space. This pose space is labeled by $w\cdot h$ poses; w denotes the number of yaw variations, and h denotes the number of pitch variations. The sparse coefficients $\mathbf{a}$ are estimated with Lasso [30] by solving Eq. 11. Since the dictionary lies on the 2D space, the method is named as **2dSC**:

$$\arg\min_{\mathbf{a}} |\mathbf{I}-\mathbf{a}\mathbf{D}|_2^2+\beta|\mathbf{a}|_1. \tag{11}$$

After solving Eq. 11, the sparse coefficients $\mathbf{a}$ can be obtained. Let a be a single element in $\mathbf{a}$; the elements across subjects $\sum_s a_s$ are summed into a 1D vector. This vector is arranged into a $w\cdot h$ 2D coefficient matrix $\mathbf{A}$. A filter $\mathbf{F}$ is applied on this 2D coefficient matrix, which is defined as follows: [0,1,0;1,1,1;0,1,0]. The maximum response position of $\mathbf{F}$ is obtained and annotated as $\mathbf{A}_{(\widetilde{w},\widetilde{h})}$. The $\widetilde{w}$ and $\widetilde{h}$ are the corresponding yaw and pitch angle in the quantized space.

Algorithm 4 Align a 3D model to a 2D image

Input: $\mathbf{I}$, $\mathbb{M}$, $\mathbf{X}$, $\mathbf{Q}^1$, $\mathbf{Q}^2$, $\mathbf{D}$
Output: $\mathbf{P}$
Step 1: Estimate $\mathbf{q}_0$ by 2dSC based on $\mathbf{I}$
Step 2: Estimate $\mathbf{S}_0$ and $\mathbf{t}_0$ using $\mathbf{q}_0$ and $\mathbf{I}$
Step 3: Apply the Alg. 5 to compute $\mathbf{P}$

Algorithm 5 Estimate **P** from the initialization

Input: **I**, $\mathbb{M}$, **X**, $\mathbf{Q}^1$, $\mathbf{Q}^2$, $\mathbf{q}_0$, $\mathbf{S}_0$, $\mathbf{t}_0$, and $\bar{\boldsymbol{\Lambda}}_*$
Output: Projection matrix **P**

for $k = 1{:}L$ **do**
  Synthesize $\hat{\mathbf{Y}}_k$ using Eq. 4 with $\mathbf{q}_{k-1}$, $\mathbf{S}_{k-1}$,$\mathbf{t}_{k-1}$
  Compute the feature $\Delta\boldsymbol{\Lambda}_{k-1}$ (HOG) on TBB
  Update q_{z_k}, $\mathbf{S}_{k-1}$, $\mathbf{t}_{k-1}$ by Eq. 9 using $\Re_2$
  Synthesize $\hat{\mathbf{Y}}_k$ using Eq. 4 with $\mathbf{q}_k$, $\mathbf{S}_k$,$\mathbf{t}_k$
  Compute $\mathbf{P}_k$ using Eq. 2
  Obtain the T-AFM using $\boldsymbol{\Gamma}_k = g(\mathbf{I}, \mathbf{P}_k, \mathbb{M})$
  Compute the feature $\Delta\boldsymbol{\Lambda}_k$ (FISH) on $\boldsymbol{\Gamma}_k$
  Update $q_{x_{k-1}}$, $q_{y_{k-1}}$ by Eq. 9 using $\Re_1$
end for
Synthesize $\hat{\mathbf{Y}}_k$ using Eq. 4 with $\mathbf{q}_k$, $\mathbf{S}_k$,$\mathbf{t}_k$
Compute the final $\mathbf{P}_k$ using Eq. 2

Along with $\mathbf{A}_{(\widetilde{w},\widetilde{h})}$, the four neighboring elements of $\mathbf{A}_{(\widetilde{w},\widetilde{h})}$ are included to determine the $\mathbf{q}_0$. A weighted sum of the five elements is computed using the five elements multiplied by the corresponding labels in the quantized space $w \cdot h$, and finally $\mathbf{q}_0$ is obtained by regressing this weighted sum to the nearest pose in the quantized space (notice that in initialization, q_{z_0} is set to be zero).

Step 2: After estimating $\mathbf{q}_0$, the values of $\mathbf{S}_0$ and $\mathbf{t}_0$ are estimated by the selected samples in the reference database whose $\mathbf{q}_*$ are lying in the same quantized space with $\mathbf{q}_0$. Using ϕ to index these samples in the reference database, the scaling matrix of a sample is denoted to be $\mathbf{S}^\phi$, and the translation vector of a sample is denoted to be $\mathbf{t}^\phi$. A normalized $\mathbf{S}^\phi$, denoted as $\hat{\mathbf{S}}^\phi$, is computed using the columns of $\mathbf{S}^\phi$ divided by the width and height of its bounding box, respectively. After computing $\hat{\mathbf{S}}^\phi$ for every selected sample in the reference database, a mean $\hat{\mathbf{S}}^\phi$ among all the selected samples is computed. Finally, $\mathbf{S}_0$ is obtained by using the mean $\hat{\mathbf{S}}^\phi$ multiplied by the width and height of **I**. To obtain $\mathbf{t}_0$, an average offset between each $\mathbf{t}^\phi$ and the reference image's bounding box's center is computed. This averaged offset is first normalized by the bounding box sizes of the selected samples and recovered based on the bounding box of the **I**. Then, this recovered offset is added to the center of **I**. The x and y coordinates of this updated center are employed to compute $\mathbf{t}_0$.

Step 3: The values of $\mathbf{q}_0$, $\mathbf{S}_0$, and $\mathbf{t}_0$ are updated based on Alg. 5 to compute **P**. The stages of testing are summarized in Alg. 4.

4 Experiments

Four experiments are conducted to demonstrate the effectiveness of the AFMA based on two challenging databases, their head pose distributions are depicted in Fig. 5. The first experiment is model selection, for which several settings of AFMA are explored to obtain the optimized module setting. In the second experiment, artificial perturbations are added to the initialized value of rotation, scaling, and translation, and the robustness of the AFMA is evaluated (Fig. 6). In the third experiment, AFMA is compared with the other pose estimation methods; both landmark-based and landmark-free approaches are employed for comparison. In the last experiment, AFMA is embedded into a 3D-aided face recognition system and the rank-k performance in closed-set face identification is proposed (Fig. 7).

4.1 Databases

Pointing 04 The original database [15] contains 15 subjects captured from 93 views. For each subject, there are two categories of data: one group of images wearing glasses and the other group of images not wearing glasses. In total, there are 2,790 images in this database. This database is employed as a reference database in testing. To implement the 2dSC, enough exemplars are essential to be selected to cover the possible variations of head pose to formulate an over determined dictionary. Therefore, a symmetric subset of the Pointing 04 database is selected, which contains 1,470 images. The images include seven yaw and pitch angles with the labels $\{-60°, -30°, -15°, 0°, 15°, 30°, 60°\}$. Each image is manually annotated with nine 2D landmarks (as shown in Fig. 1). The pose (pitch and yaw) of each image is re-estimated based on a generic 3D sparse model, which is obtained by annotating[2] the nine landmarks on a generic 3D model of the FRGC database [27]. The re-estimated pitch and yaw values are employed as the new labels for each group of images. The distribution of head poses in this database could be viewed as the black triangles in Fig. 5a. This re-annotated database is employed as a reference database to build the sparse dictionary in 2dSC.

UHDB31A[2] This annotated database is captured using nine high-resolution 3dMD [1] cameras with an average 800 $\times$ 600 facial ROI size. It contains a personalized 3D model for each subject. During enrollment, each subject is asked to look at three vertical positions to generate an additional three pitch variations. In sum, this database contains 10 subjects with 27 view-point variations, 270 images in total. This database is used as a training set for $\Re_1$ and $\Re_2$. The pose distribution in this database can be viewed in Fig. 5a.

UHDB11 This is a published database [32] that aims to validate the 3D-aided 2D face recognition algorithm. It includes images of 23 subjects captured from different illumination and pose changes. In addition, this database contains a personalized 3D model for each subject. There are 408 images without significant roll variations are selected in UHDB11 as

[2] A few imprecise landmark annotations were rectified in these model/database in this journal version.

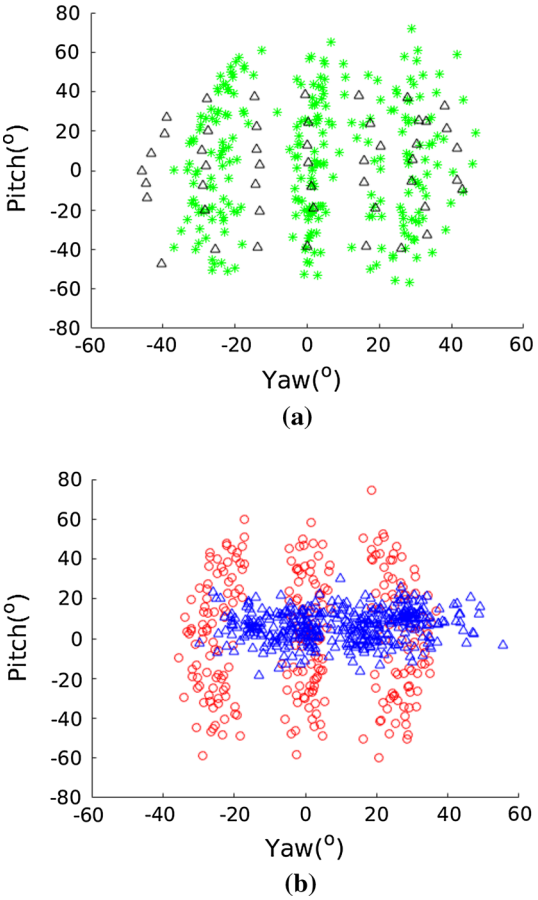

Fig. 5 Depicted is the pose distribution in training and testing sets. **a** The black triangles depict the poses employed in the Pointing 04 database. The green stars depict the poses employed to train the regressor in UHDB31-A. **b** The red circles depict the pose distribution in UHDB31B, and the blue triangles depict the pose distribution in UHDB11 (color figure online)

a testing database. The pose distribution of the this database is shown in Fig. 5b. This database is employed to evaluate the algorithm performance when the samples have large yaw variations.

UHDB31B[2] This annotated database is captured in the same environment as UHDB31A, but with ten different subjects. UHDB31B is used as one of the testing databases. This database also contains 270 images, and the pose distribution is shown as red circles in Fig. 5b. This database is used to evaluate the algorithm performance when samples have large pitch variations.

4.2 Parameter settings

The algorithm contains four regression stages. In training, a personalized AFM is used for texture lifting. In testing, since the personalized 3D model may not be available, as is typical for in-the-wild scenarios, accuracies are reported with and without a personalized 3D model. In feature extraction

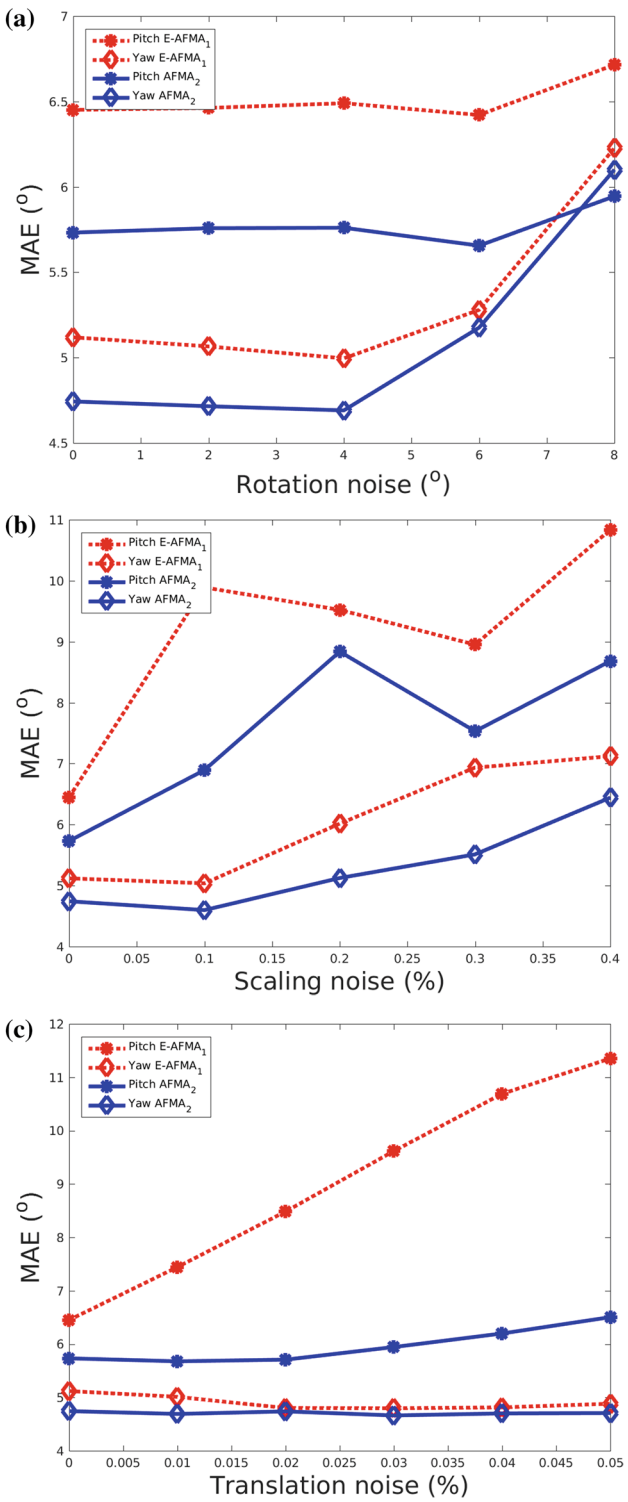

Fig. 6 Different levels of noise are added to the input of E-AFMA$_1$ and AFMA$_2$; the pitch and yaw error (MAE) are reported after the algorithm terminated

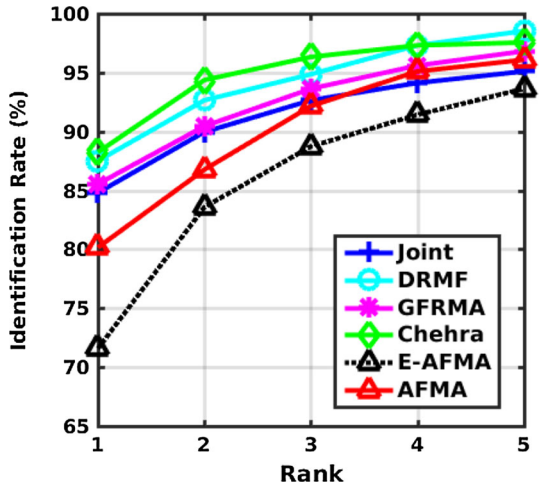

Fig. 7 Identification rates computed on UHDB11. AFMA versus the state-of-the-art landmark detectors employed for pose estimation

of 2dSC, ROI is resized into 100 × 100 pixels. Image illumination is normalized via Tan and Triggs algorithm [29]. The HOG feature is extracted with the default parameter settings [9]. In regression, the size of the sampling region for extracting FISH score is 8 × 8 and the stride is two pixels. The size of the T-AFM is 90 × 90. FISH feature is computed on an effective region with the sizes 53 × 38 (half face) and 53 × 74 (whole face) for pitch and yaw regression. Before extracting the FISH feature, the Tan and Triggs [29] illumination normalization algorithm was applied to minimize the effects of lighting. The parameter setting of the TBB is as follows: The enlarged parameter ξ_1 is set to be ten. The region inside the TBB is resized into $\xi_2 \times \xi_2$ pixels to extract HOG features. The value of ξ_2 equals 100 in the experiment.

The $\Re_1$ includes two sub-regressors, one for pitch regression, another for yaw regression. For convergence purposes, the learning parameters of the two sub-regressors λ_1 and λ_2 are set to be one and 0.2, respectively. In $\Re_2$, the five parameters are jointly estimated, and the learning rate in GSDM is set to be 0.1 for roll, 0.3 for scaling (pitch/yaw), and 0.5 for translation (pitch/yaw). In UHDB11, all the Difference of Gaussian (DoG) parameters in Tan and Triggs are set to 0.5. In UHDB31, due to the dark skin color of some of the subjects, the DoG parameter is set to be 1.0 for better convergence.

4.3 Model selection

The performance of AFMA is evaluated with different module settings on two databases: UHDB11 and UHDB31B. The performances of $\Re_1$ and $\Re_2$ with respect to a generic/personalized model are mainly investigated in this experiment. Specifically, the performance of RSSDM and SDM is evaluated in $\Re_1$ to figure out whether an additional random projection approach could help enhance the performance. To quantify the error of pose estimation, multiple metrics are necessary. For pitch, yaw, and roll, the absolute errors of angles are reported, which are measured in degrees. For scaling parameters (denoted as sc.x and sc.y in Table 1), the absolute errors which measured the percentage that the estimated values deviate from the real values are reported. For translation parameters (denoted as tr.x and tr.y in Table 1), the absolute errors between the estimated value and the true value are firstly computed, then they are normalized by dividing the size of the original bounding box, which yields a percentage error. The mean absolute errors (MAEs) of seven pose parameters are reported in Table 1, and the cumulative errors of six parameters are reported in Figs. 8 and 9. The abbreviations that employed in Table 1 are shown as follows:

2dSC: The 2D sparse coding approach employed for initialization.

PS3D: The T-AFM is generated through a personalized 3D model.

Table 1 The MAE of model selection computed on UHDB11 and UHDB31B

Methods	UHDB11							UHDB31B						
	pitch	yaw	roll	sc.x	sc.y	tr.x	tr.y	pitch	yaw	roll	sc.x	sc.y	tr.x	tr.y
2dSC	8.48	5.20	2.81	6.43	6.37	2.18	2.64	9.77	5.79	7.82	5.42	12.77	1.67	6.13
G3D + E-AFMA$_1$	6.61	5.09	2.81	6.47	6.10	2.19	2.62	8.65	5.64	7.82	5.09	11.20	1.67	6.13
G3D + E-AFMA$_2$	5.94	5.06	2.81	6.50	6.05	2.19	2.63	8.54	5.68	7.82	5.22	12.08	1.67	6.12
G3D + AFMA$_1$	6.50	4.96	2.14	6.39	5.49	2.24	1.86	8.50	5.54	5.57	5.31	**10.78**	1.47	**2.14**
G3D + AFMA$_2$	6.76	4.93	**2.12**	6.37	5.49	2.24	1.86	8.18	5.56	**5.56**	5.33	10.82	1.44	**2.14**
PS3D + E-AFMA$_1$	6.45	5.12	2.81	6.53	6.01	2.18	2.62	8.76	5.49	7.82	**5.08**	11.70	1.67	6.12
PS3D + E-AFMA$_2$	5.98	5.03	2.81	6.51	5.99	2.18	2.63	8.57	5.59	7.82	5.26	11.99	1.67	6.12
PS3D + AFMA$_1$	5.77	4.89	2.22	**6.15**	5.26	**1.83**	**1.57**	8.27	**5.46**	5.58	5.34	11.03	1.41	2.57
PS3D + AFMA$_2$	**5.73**	**4.74**	2.19	6.17	**5.20**	1.85	1.58	**8.10**	5.53	5.59	5.32	11.02	**1.40**	2.57

The bold values highlight the lowest MAE among all the methods

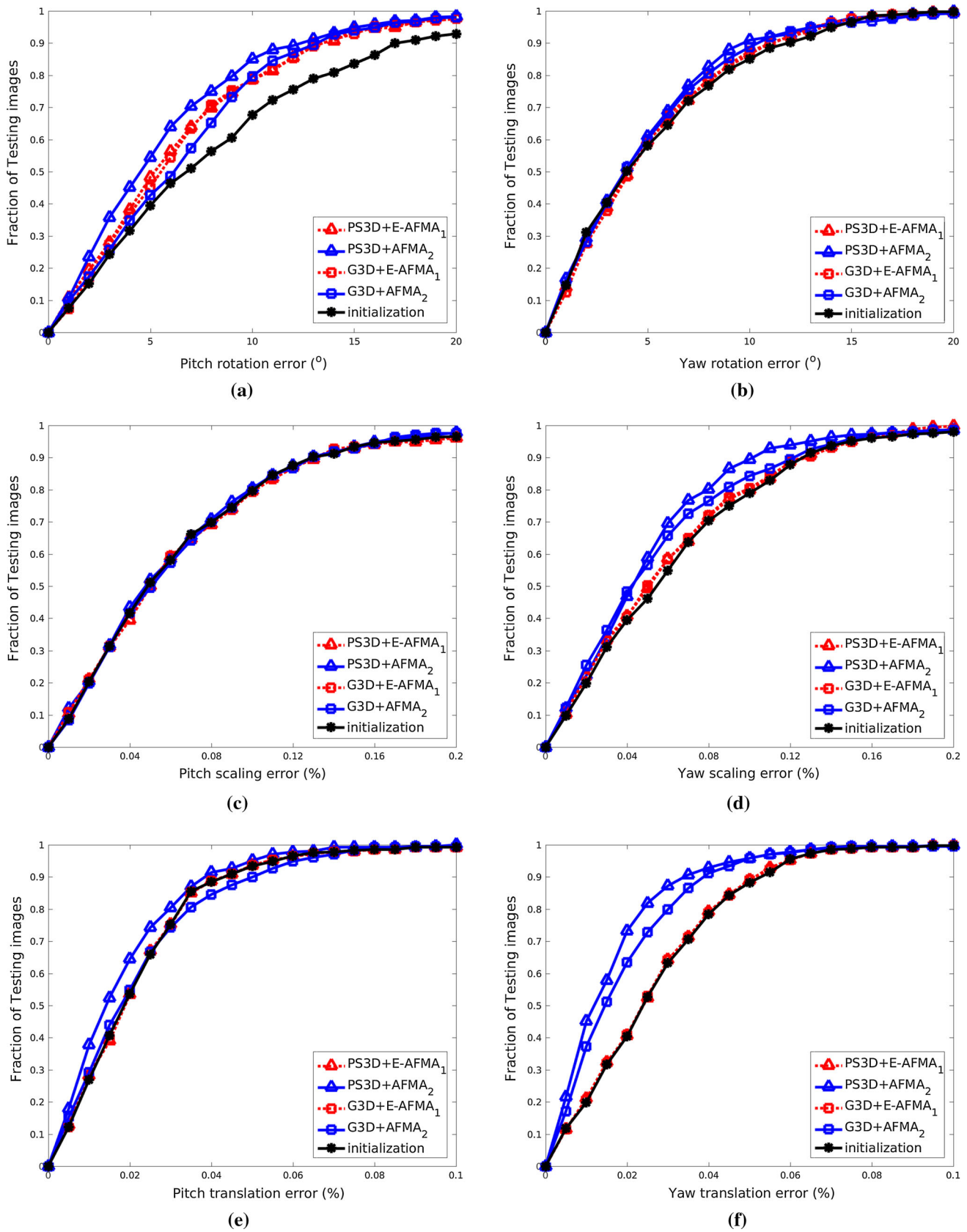

Fig. 8 The cumulative error distribution curves of the pose parameters computed on UHDB11 database

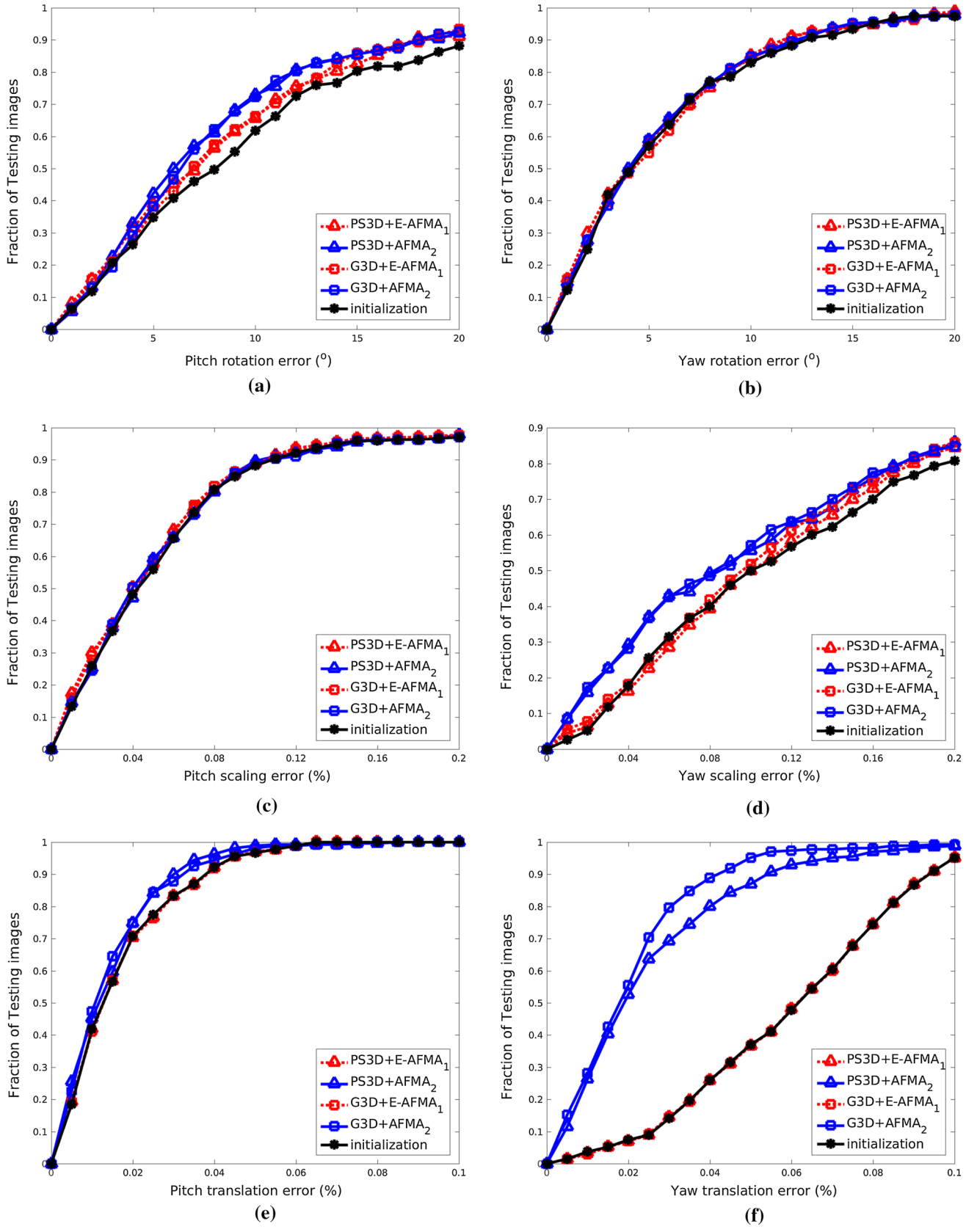

Fig. 9 The cumulative error distribution curves of the pose parameters computed on UHDB31B database

Table 2 This table depicts the pose estimation results of different approaches; the mean absolute error (MAE) is reported

Methods	UHDB11							UHDB31B						
	pitch	yaw	roll	sc.p	sc.y	tr.p	tr.y	pitch	yaw	roll	sc.p	sc.y	tr.p	tr.y
SDM [39]	15.76	8.30	4.12	8.70	38.77	2.16	3.24	10.38	6.27	5.18	16.07	25.80	1.89	**2.39**
GSDM [40]	8.19	9.70	3.54	10.74	42.53	2.09	2.63	8.64	6.51	5.78	18.84	31.38	**1.40**	3.07
RF [43]	9.76	12.06	2.76	40.95	54.49	1.92	2.61	11.62	6.97	3.19	35.36	49.06	1.65	5.00
KRF [17]	7.54	12.24	2.70	37.1	54.63	3.01	2.77	12.23	7.49	**3.10**	32.95	47.62	2.43	3.49
AFMA	**5.73**	**4.74**	**2.19**	**6.17**	**5.20**	**1.85**	**1.58**	**8.10**	**5.53**	5.59	**5.32**	**11.02**	**1.40**	2.57
Joint [45]	7.65	7.84	1.70	7.37	3.82	1.52	1.28	–	–	–	–	–	–	–
DRMF [3]	7.50	3.65	1.43	8.42	3.40	1.00	1.24	10.66	5.38	3.94	6.43	7.88	1.64	2.44
GFRMA [5]	7.24	3.67	1.29	8.83	3.62	1.00	1.28	10.21	4.03	3.88	6.76	7.90	1.72	2.64
IFA [4]	6.06	3.01	0.96	5.11	2.48	0.96	0.80	7.54	4.32	2.83	8.02	7.67	2.44	2.40

The bold values highlight the lowest MAE among all the landmark-free pose estimation approaches (SDM, GSDM, RF, and KRF)

G3D: The T-AFM is generated through a generic 3D model.

AFMA: Using both $\Re_1$, $\Re_2$ in estimating pose parameters.

AFMA$_1$: Using the SDM as the regressor in $\Re_1$.

AFMA$_2$: Using the RSSDM as the regressor in $\Re_1$.

E-AFMA: Using $\Re_1$ in estimating pose parameters. The scaling and translation parameters are updated based on the Step 2 mentioned in Sect. 3.3.

E-AFMA$_1$: Using SDM as the regressor in $\Re_1$. It is the method employed in [37].

E-AFMA$_2$: Using RSSDM as the regressor in $\Re_1$.

Table 1 depicts that compared to 2dSC, the error of pose parameters is reduced in all cases, which demonstrates the effectiveness of $\Re_1$. It is important to mention that in the preliminary version [37] of E-AFMA, the yaw error was not reduced. This is due to the overfitting. In this paper, the perturbation size is reduced from $\{-17 : 17\}$ to $\{-7 : 7\}$ and the learning rate is adjusted to control the convergence of the algorithm, and thereby the problem is solved.

In Table 1, AFMA outperforms E-AFMA in almost all cases, which demonstrates that $\Re_2$ is essential for boosting the performance of pose estimation. Among all the improvements, one of the biggest advantages of adopting $\Re_2$ is in estimating the translation parameters. This improvement can be visualized best in Figs. 8f and 9f. Comparing the results of E-AFMA and AFMA, it is observed that when the scaling and translation errors both go down, the pitch and yaw errors decrease at the same time. From this characteristic, it can be concluded that the location and size of the bounding box are critical for estimating the pitch and yaw.

Comparing the regression approaches, RSSDM outperforms SDM in general. However, some contradictory cases are also observed. One explanation is because the random subspace dividing approach in RSSDM is affected by the unknown background or the artifacts that appeared in T-AFM, which have not been learned in training.

As for the comparison of a personalized 3D model vs. a generic 3D model, it is difficult to conclude which one is better if only take into account the results of E-AFMA. In contrast, the personalized 3D model contributes more to AFMA. This phenomenon may caused by the TBB. Since the TBB is generated through projecting a sparse 3D model, a general 3D model may samples in-consistent amount of background regions between different subjects compared to the personalized 3D model.

4.4 Sensitivity analysis

In sensitivity analysis, the robustness of E-AFMA$_1$ and AFMA$_2$ are evaluated. The performance of both methods on the UH-DB11 database was investigated. A personalized 3D model was adopted to control the uncertain variables. The results are shown in Fig. 6.

For clarity reasons, only the pitch and yaw errors are reported in this experiment. It is believed that their errors are relevant to the errors of scaling and translation parameters. To investigate the model's robustness to the initialized values of pitch and yaw, random pitch and yaw noises are added to the input of E-AFMA/AFMA. The results are shown in Fig. 6a. The x-axis in Fig. 6a records the summation of pitch and yaw noises added to each subject. In investigating the scaling and translation parameters, to control the number of variables, the output of 2dSC is kept as the original initialized values unchanged and the size and position of the bounding box in the original image are adjusted w.r.t. its bounding box size. The percentage value of resizing/moving of the bounding box is shown as the x-axis in Fig. 6b, c.

Figure 6 indicates that both AFMA and E-AFMA are robust to rotation errors of four degrees of less, while AFMA is much more robust to scaling and translation errors caused by bounding box resizing and moving.

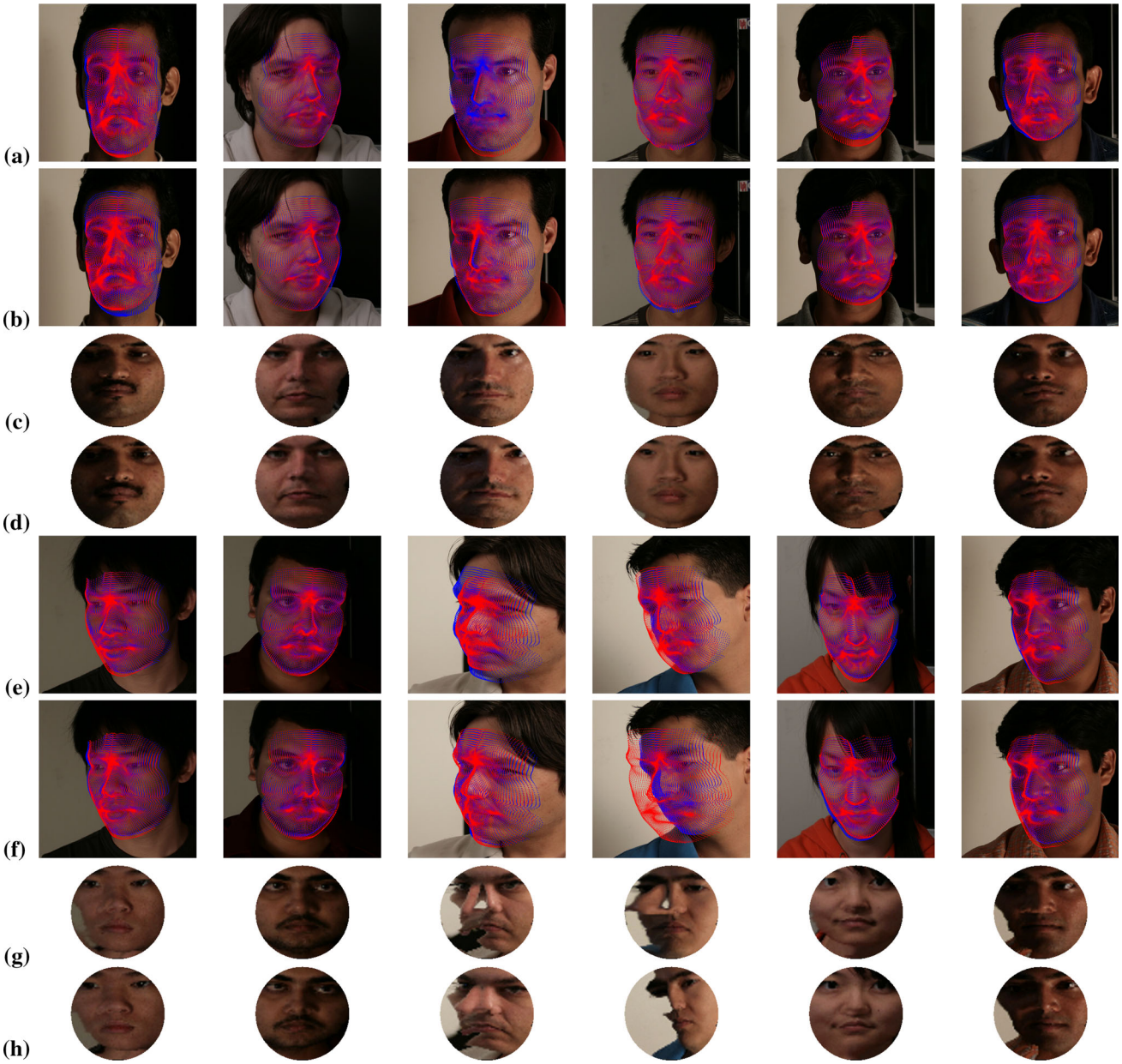

Fig. 10 Depicted are the model registration results (better viewed in color). The three-dimensional model is projected onto 2D images based on two pose estimation approaches. In the rows **a**, **b**, **e**, **f**, each dot depicts the projected 3D vertex on the 2D image. The blue dots in **a**, **b**, **e**, **f** depict the projected vertices of a personalized 3D model that are estimated by the manually annotated landmarks. The red dots in rows **a** and **e** depict the projected vertices that are estimated by the landmarks detected by IFA [4]. The red dots in rows **b** and **f** are estimated by AFMA. The T-AFMs in rows **c** and **g** are generated based on the projected vertices as depicted by the blue dots in **a**, **b**, **e**, **f**. The T-AFM in rows **d** and **h** are generated based on the projected vertices as depicted by the red dots in **b** and **f** (color figure online)

4.5 Performance evaluation

AFMA is compared with five regression-based approaches under a landmark-free pose estimation protocol. The performance of four state-of-the-art landmark-based face alignment methods is employed to report the pose estimation results under a common landmark-based pose estimation protocol. The MAE is reported in Table 2.

The landmark-free protocol is defined as follows. The candidate algorithm is able to exploit the whole face ROI as the input without any information obtained from a specific local region on the face. This protocol is widely adopted in landmark-free pose estimation [6,14,16,20,25,35], and

is also employed in object/scene recognition (e.g., the root-based deformable part models [10]). Since the candidate algorithm is required to return a value that is not constrained by the pose labels in the training database, the regression-based methods are selected for comparison. Under this protocol, the method proposed in [16,17] is employed. The HOG feature is leveraged and extracted from the TBB (defined in Sect. 3.2) as the input of the regressors. The target space in this experiment comprises all seven pose parameters. UHDB31A is employed as the training database for the regressors, and UHDB11 and UHDB31B are employed for testing. In experiments, besides the SDM/GSDM, the Random Forest (RF) and a recent extension [17] of Random Forest are employed for comparison. The number of trees is assigned to be 40, the depth of each tree is assigned to be five. The K value in KRF is assigned to be six. All algorithms are initialized from a frontal pose in both training and testing. From Table 2, it is observed that most of the approaches yield large errors in landmark-free pose estimation. Even though some of the algorithms attained relative better performance in estimating the pitch and yaw (e.g., GSDM), it failed to estimate the scaling parameters accurately. In conclusion, the selected regression-based approaches that rely on face ROI as inputs are not fine-grained enough for model registration, which requires high accuracy in estimating all the seven parameters.

For landmark-based pose estimation, four automatic facial landmark detectors [3–5,45] are employed to detect facial landmarks, RDD is employed to obtain the estimated pose parameters. The source codes of the landmark detector were downloaded from the authors' websites. To test the algorithms, UHDB11 and UHDB31B are down-sampled four times to fit the input size of the methods. Default face detectors provided by the landmark detectors are employed to generate the facial bounding boxes. If the default face detector failed to work, the algorithm is fed into manually cropped bounding boxes for initialization, which yields better results. Since the model [45] was not trained on a database containing large pitch variations, it failed to detect landmarks on UHDB31B. Hence, the landmarks generated by [45] was not used to estimate the pose on UHDB31B for fair comparison.

Table 2 depicts that the AFMA outperforms a majority of the landmark-based pose estimation methods in pitch estimation. For yaw, the AFMA shows competitive performance. It is observed that on the UHDB11 database, the translation errors estimated from the detected landmarks are significantly smaller than AFMA. This indicates the landmark-based approaches could localize the facial region very well based on the fiducial points. Among all the four algorithms, experimental results indicate that IFA [4] performs the best. The qualitative results of AFMA vs. IFA are compared in Fig. 10.

4.6 Performance in 3D-aided face recognition

In the last experiment, AFMA is embedded into a 3D-aided face recognition pipeline [12]. The rank-k identification rate on 408 images of the UHDB11 database is reported. Four automatic landmark detectors are leveraged in the pose estimation module for comparison. A personalized 3D model is assumed available for each subject.

The rank-k identification rate is reported in Fig. 7. Since in [12], the similarity of two subjects is compared based on the similarity of T-AFM, pose estimation plays a critical role in identification. Figure 7 shows that AFMA is close to the landmark-based pose estimation approaches' performance and better than the E-AFMA. Since AFMA outperforms [45] and comes close to the performance of [3–5] in both pitch/yaw estimation (Table 2), it can be inferred that the accuracy of estimating the translation parameter is critical in determining the quality of T-AFM and thereby affects identification performance.

5 Conclusion

In this paper, a landmark-free pose estimation approach was presented to directly estimate the projection matrix for model registration. This method leverages the textual information on annotated face model to estimate the pose parameters encoded in the projection matrix. The performance of AFMA was analyzed on two challenging databases. Comprehensive experiments were conducted to compare AFMA with other pose estimation approaches. It is demonstrated that AFMA, which is not using landmarks, is able to achieve 80% rank-1 face identification rate in a 3D-aided face recognition pipeline, within 5–9% of all methods that use landmarks.

References

1. 3dMD 3dMD: 3D imaging systems and software (2012). http://www.3dmd.com/
2. Abiantun, R., Prabhu, U., Savvides, M.: Sparse feature extraction for pose-tolerant face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **36**(10), 2061–2073 (2014)
3. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Robust discriminative response map fitting with constrained local models. In: Proceedings of IEEE conference on computer vision and pattern recognition, Portland, OR, pp. 3444–3451 (2013)
4. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: Proceedings of IEEE conference on computer vision and pattern recognition, Columbus, OH, pp. 1859–1866 (2014)
5. Asthana, A., Zafeiriou, S., Tzimiropoulos, G., Cheng, S., Pantic, M.: From pixels to response maps: discriminative image filtering for face alignment in the wild. IEEE Trans. Pattern Anal. Mach. Intell. **37**(6), 1312–1320 (2015)

6. Ba, S.O., Odobez, J.M.: Recognizing visual focus of attention from head pose in natural meetings. IEEE Int. Conf. Syst. Man Cybern. **39**(1), 16–33 (2009)
7. Balasubramanian, V., Ye, J., Panchanathan, S.: Biased manifold embedding: a framework for person-independent head pose estimation. In: Proceedings of IEEE conference on computer vision and pattern recognition, Minneapolis, MN, pp. 1–7 (2007)
8. Bouaziz, S., Wang, Y., Pauly, M.: Online modeling for realtime facial animation. ACM Trans. Graph. **32**(4), 40 (2013)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of IEEE conference on computer vision and pattern recognition, San Diego, CA, pp. 886–893 (2005)
10. Divvala, SK., Efros, AA., Hebert, M.: How important are deformable parts in the deformable parts model? In: Proceedings of European conference on computer vision (workshop), Florence, Italy, pp. 31–40 (2012)
11. Dou, P., Wu, Y., Shishir, S.K., Kakadiaris, I.A.: Benchmarking 3D pose estimation for face recognition. In: Proceedings of IEEE international conference on pattern recognition, Stockholm, Sweden, pp. 190–195 (2014)
12. Dou, P., Zhang, L., Wu, Y., Shah, S.K., Kakadiaris, I.A.: Pose-robust face signature for multi-view face recognition. In: Proceedings of IEEE international conference on biometrics: theory, applications and systems, Arlington, VA, pp. 1–8 (2015)
13. Drucker, H., Burges, C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: Advances in neural information processing systems, Denver, CO, pp. 155–161 (1997)
14. Geng, X., Xia, Y.: Head pose estimation based on multivariate label distribution. In: Proceedings of IEEE conference on computer vision and pattern recognition, Columbus, OH, pp. 1837–1842 (2014)
15. Gourier, N., Hall, D., Crowley, J.: Estimating face orientation from robust detection of salient facial structures. In: Proceedings of international workshop on visual observation of deictic gestures, Cambridge, UK, pp. 1–9 (2004)
16. Guo, G., Fu, Y., Dyer, CR., Huang, T.: Head pose estimation: classification or regression? In: Proceedings of international conference on pattern recognition, Tampa, FL, pp. 1–4 (2008)
17. Hara, K., Chellappa, R.: Growing regression forests by classification: applications to object pose estimation. In: Proceedings of European conference on computer vision, Zurich, Switzerland (2014)
18. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
19. Hsu, G., Peng, H.: Face recognition across poses using a single 3D reference model. In: Proceedings of IEEE conference on computer vision and pattern recognition, Portland, OR, pp. 869–874 (2013)
20. Huang, D., Storer, M., De la Torre, F., Bischof, H.: Supervised local subspace learning for continuous head pose estimation. In: Proceedings of IEEE conference on computer vision and pattern recognition, Colorado Springs, CO, pp. 2921–2928 (2011)
21. Jeni, L., Cohn, J., Kanade, T.: Dense 3D face alignment from 2D videos in real-time. In: Proceedings of IEEE international conference and workshops on automatic face and gesture recognition, Ljubljana, Slovenia, vol. 1, pp. 1–8 (2015)
22. Kakadiaris, I.A., Passalis, G., Toderici, G., Murtuza, M., Lu, Y., Karampatziakis, N., Theoharis, T.: Three-dimensional face recognition in the presence of facial expressions: an annotated deformable model approach. IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), 640–649 (2007)
23. Kakadiaris, I.A., Toderici, G., Evangelopoulos, G., Passalis, G., Zhao, X., Shah, S.K., Theoharis, T.: 3D–2D face recognition with pose and illumination normalization. Comput. Vis. Image Underst. **154**, 137–151 (2017)
24. Kemelmacher-Shlizerman, I., Seitz, S.: Face reconstruction in the wild. In: Proceedings of IEEE international conference on computer vision, Barcelona, Spain, pp. 1746–1753 (2011)
25. Ma, B., Li, A., Chai, X., Shan, S.: CovGa: a novel descriptor based on symmetry of regions for head pose estimation. Neurocomputing **143**, 97–108 (2014)
26. Masi, I., Lisanti, G., Bagdanov, A., Pala, P., Bimbo, A.: Using 3D models to recognize 2D faces in the wild. In: Proceedings of IEEE conference on computer vision and pattern recognition, Portland, OR, pp. 775–780 (2013)
27. Phillips, P.J., Scruggs, W.T., O'Toole, A.J., Flynn, P.J., Bowyer, K.W., Schott, C.L., Sharpe, M.: FRVT 2006 and ICE 2006 large-scale experimental results. IEEE Trans. Pattern Anal. Mach. Intell. **32**(5), 831–846 (2010)
28. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification. In: Proceedings of IEEE conference on computer vision and pattern recognition, Columbus, OH, pp. 1701–1708 (2014)
29. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Image Process. **19**(6), 1635–1650 (2010)
30. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B. **58**(1), 267–288 (1996)
31. Toderici, G., Passalis, G., Zafeiriou, S., Tzimiropoulos, G., Petrou, M., Theoharis, T., Kakadiaris, I.A.: Bidirectional relighting for 3D-aided 2D face recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition, San Francisco, CA, pp. 2721–2728 (2010)
32. Toderici, G., Evangelopoulos, G., Fang, T., Theoharis, T., Kakadiaris, I.A.: UHDB11 database for 3D–2D face recognition. In: Proceedings of Pacific-Rim symposium on image and video technology, Guanajuato, Mexico, pp. 73–86 (2013)
33. Vu, P.V., Chandler, D.: A fast wavelet-based algorithm for global and local image sharpness estimation. IEEE Signal Process. Lett. **19**(7), 423–426 (2012)
34. Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., Ma, Y.: Toward a practical face recognition system: robust alignment and illumination by sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 372–386 (2012)
35. Wang, C., Song, X.: Robust head pose estimation via supervised manifold learning. Neural Netw. **53**, 15–25 (2014)
36. Weise, T., Bouaziz, S., Li, H., Pauly, M.: Realtime performance-based facial animation. ACM Trans. Graph. **30**(4), 77 (2011)
37. Wu, Y., Xu, X., Shah, S.K., Kakadiaris, I.A.: Towards fitting a 3D dense facial model to a 2D image: a landmark-free approach. In: Proceedings of international conference on biometrics: theory, applications and systems, Arlington, VA, pp. 1–8 (2015)
38. Wu, Y., Shah, S.K., Kakadiaris, I.A.: Rendering or normalization? An analysis of the 3D-aided pose-invariant face recognition. In: Proceedings of IEEE international conference on identity, security and behavior analysis, Sendai, Japan, pp. 1–8 (2016)
39. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: Proceedings of IEEE conference on computer vision and pattern recognition, Portland, OR, pp. 532–539 (2013)
40. Xiong, X., De la Torre, F.: Global supervised descent method. In: Proceedings of IEEE conference on computer vision and pattern recognition, Boston, MA, pp. 2664–2673 (2015)
41. Yang, H., Jia, X., Patras, I., Chan, K.P.: Random subspace supervised descent method for regression problems in computer vision. IEEE Trans. Signal Process. Lett. **22**(10), 1816–1820 (2015)
42. Yang, H., Mou, W., Zhang, Y., Patras, I., Gunes, H., Robinson, P.: Face alignment assisted by head pose estimation. In: Proceedings of British machine vision conference, Swansea, UK, pp. 1–13 (2015)
43. Zhao, X., Kim, T.K., Luo, W.: Unified face analysis by iterative multi-output random forests. In: Proceedings of IEEE conference

on computer vision and pattern recognition, IEEE, Columbus, OH, pp. 1765–1772 (2014)
44. Zhen, X., Wang, Z., Yu, M., Li, S.: Supervised descriptor learning for multi-output regression. In: Proceedings of IEEE conference on computer vision and pattern recognition, Boston, MA, pp. 1211–1218 (2015)
45. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Proceedings of IEEE conference on computer vision and pattern recognition, Providence, RI, pp. 2879–2886 (2012)
46. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S.: High-fidelity pose and expression normalization for face recognition in the wild. In: Proceedings of IEEE conference on computer vision and pattern recognition, Boston, MA, pp. 787–796 (2015)

Yuhang Wu is a student pursuing a Ph.D. degree in computer science at the University of Houston. He joined the department in 2013. He earned his B.E. degree in computer science and technology from the Capital Normal University, China. His research interests include biometrics and computer vision, specifically on 3D-aided face recognition, face pose-estimation, and face landmark detection.

Shishir K. Shah is a Professor of Computer Science at the University of Houston. He joined the department in 2005. He received his B.S. degree in Mechanical Engineering and M.S. and Ph.D. degrees in Electrical and Computer Engineering from The University of Texas at Austin. He directs research at the Quantitative Imaging Laboratory and his current research focuses on fundamentals of computer vision, pattern recognition, and statistical methods in image and data analysis with applications in multi-modality sensing, video analytics, object recognition, biometrics, and microscope image analysis. He has co-edited two books and authored numerous papers on object recognition, sensor fusion, statistical pattern analysis, biometrics, and video analytics. Dr. Shah currently serves as an Associate Editor for Image and Vision Computing and the IEEE Journal of Translational Engineering on Health and Medicine. He received the College of Natural Sciences and Mathematics' John C. Butler Teaching Excellence Award in 2011 and the Department of Computer Science Academic Excellence Award in 2010.

Ioannis A. Kakadiaris is a Hugh Roy and Lillie Cranz Cullen University Professor of Computer Science, Electrical & Computer Engineering, and Biomedical Engineering at the University of Houston. He joined UH in 1997 after a postdoctoral fellowship at the University of Pennsylvania. He earned his B.Sc. in physics at the University of Athens in Greece, his M.Sc. in computer science from Northeastern University and his Ph.D. at the University of Pennsylvania. He is the founder of the Computational Biomedicine Lab and the Director of the DHS Center of Excellence on Borders, Trade, and Immigration Research (BTI). His research interests include biometrics, video analytics, computer vision, pattern recognition, and biomedical image computing. He is the recipient of a number of awards, including the NSF Early Career Development Award, Schlumberger Technical Foundation Award, UH Computer Science Research Excellence Award, UH Enron Teaching Excellence Award, and the James Muller Vulnerable Plaque Young Investigator Prize. His research has been featured on The Discovery Channel, National Public Radio, KPRC NBC News, KTRH ABC News, and KHOU CBS News.