Supervised Coordinate Descent Method with a 3D Bilinear Model for Face Alignment and Tracking

Yongqiang Zhang[1], Shuang Liu[2], Xiaosong Yang[2], Jianjun Zhang[2], Daming Shi[1,*]
1 Harbin Institute of Technology, 2 Bournemouth University
seekever@foxmail.com

Abstract

Face alignment and tracking play important roles in facial performance capture. Existing data-driven methods for monocular videos suffer from large variations of pose and expression. In this paper we propose an efficient and robust method for this task by introducing a novel supervised coordinate descent method (SCDM) with 3d bilinear representation. Instead of learning the mapping between the whole parameters and image features directly with a cascaded regression framework in current methods, we learn individual sets of parameters mappings separately step by step by a coordinate descent mean. Since different parameters make different contributions to the displacement of facial landmarks, our method is more discriminative to current whole-parameter cascaded regression methods. Benefiting from a 3D bilinear model learned from public databases, the proposed method can handle the head pose changes and extreme expressions out of plane better than other 2D-based methods. We present the reliable result of face tracking under various head poses and facial expressions on challenging video sequences collected online. The experimental results show our method outperforms state-of-art data-driven methods.

Keywords: supervised coordinate descent method, face alignment, face tracking, facial performance capture

1 Introduction

Face alignment and tracking are essential for tasks such as facial performance capture and facial animation, which have been popular in 3D online games and CG films. It also plays an important role in building person specific avatars for VR or AR applications, such as virtual classroom and remote video chatting. Although many techniques have been developed to achieve facial capture, oftentimes dedicated equipment are needed, such as camera arrays, depth camera and facial markers. For consumer grade applications, an aspiring solution is to develop a facial performance capture system based on automatic face alignment and tracking with a commodity web-camera.

In the past, some model-based methods have been studied for face alignment in 2D image domain. One well-known method is Active Shape Model (ASM) [1], which is one of the earliest data-driven models for shape fitting. As an improvement, Active Appearance Model(AAM) [2] considers the global appearance rather than only local textures in ASM. Although AAM and its variations [3, 4, 5, 6] can fit face well for near-frontal face images, they tend to fail for uncontrolled face images in the wild environment. Recently regression-based methods have been exploited, and cascaded regression with shape-indexed feature is introduced for face alignment, like face alignment by Explicit Shape Regression[7] and Supervised Descent Method [8], as significant achievements in regression-based methods. These methods can be used for real-time face fitting efficiently but they cannot

capture variations of pose and expression out of plane due to lack of 3D information.

With the development of commercial depth acquisition devices, it has been convenient to obtain 3D face data. Benefiting from existing 3D face databases, regression-based methods with 3D information have been proposed. While some methods[9] uses 3D face database for augmenting 2D face images with wilder ranges of poses and expressions, some methods have trained 3D face models for user-specific multi-parameter regression methods[10]. At present this kind of method can deal with slight variations of pose and expression in 3D space, while a post-processing is still needed.

In this paper, we are aiming for low cost commodity webcams which make accurate face alignment and tracking much harder. We propose a novel supervised coordinate descent method combined with a 3D bilinear face model for robust face alignment and tracking across poses and expressions. The main contributions of this paper are described as follows:

- We propose a novel supervised coordinate descent method for learning different types of parameters mappings separately to reduce the cross impact caused by learning a whole-parameter mapping.
- Taking advantage of a 3D bilinear model learned from public databases, our method can handle large pose and expression variations in 3D space with accuracy - overcoming the deficiencies of existing methods.
- Our method is validated and compared in challenging face videos collected online.

Related methods are reviewed in section 2, and the framework and details of our approach are described in section 3. In section 4, we demonstrate how to train and test our model for a face alignment and tracking application, followed by experiments and results in section 5, and conclusion in section 6.

2 Related work

Various kinds of data-driven methods have been developed for face alignment and tracking. Because our method is a regression-based method with a 3D facial bilinear representation model, here we mainly review these following related works, including 2D parameter-based methods, 2D regression-based methods, 3D facial representation models and current 3D regression-based methods.

AAM[2] and its variants[3, 4, 5, 6] are typical 2D parameter-based methods. In this kind of methods, a PCA model is trained for global appearance while a shape PCA is trained at the same time. Matthews et al. [11] improved the optimization algorithm for AAM cost function by well-known project-out inverse compositional algorithm. Cootes et al. [3] proposed view-based AAM to adjust the standard AAM to the multi-view environment. Donner et al. proposed a fast search for parameters of AAM using canonical correlation analysis[12]. Gonzalezmora et al. proposed 2D bilinear AAM [5], and Lee et al. proposed tensor-based AAM [6]. These extensions ease the impact of variations of pose, expression and illumination, but they often tend to unreliable for faces in the wild, due to the sensitivity to initialization.

Another popular type of face alignment methods are 2D regression-based methods. Among them, the cascaded regression framework with shape-indexed features obtained in 2D images is adopted. As a representative method, Cao et al. proposed a real-time Explicit Shape Regression [10]. They designed a two-level boosted rand ferns regression with pixel difference feature in local coordinates. Further, Xiong et al. presented Supervised Descent Method [8, 13], which provides reasonable theory description of the cascaded regression: It can be regarded as a supervised learning method for approximating the iterative gradient descent solution of a nonlinear cost function. It is simple and extensible, which inspires many works under this framework, such as ensemble of regression trees [14], and regularized linear cascaded regression with local binary features[15]. These methods are very fast while maintaining high accuracy, but they often show poor results when the variations of pose and expression are large. Feng et al. proposed a cascaded collaborative regression trained using a mixture of synthetic and real images[9]. A 3D morphable face model[16] is used to generate synthesized 2D faces with various poses. Thus it shows a better capability

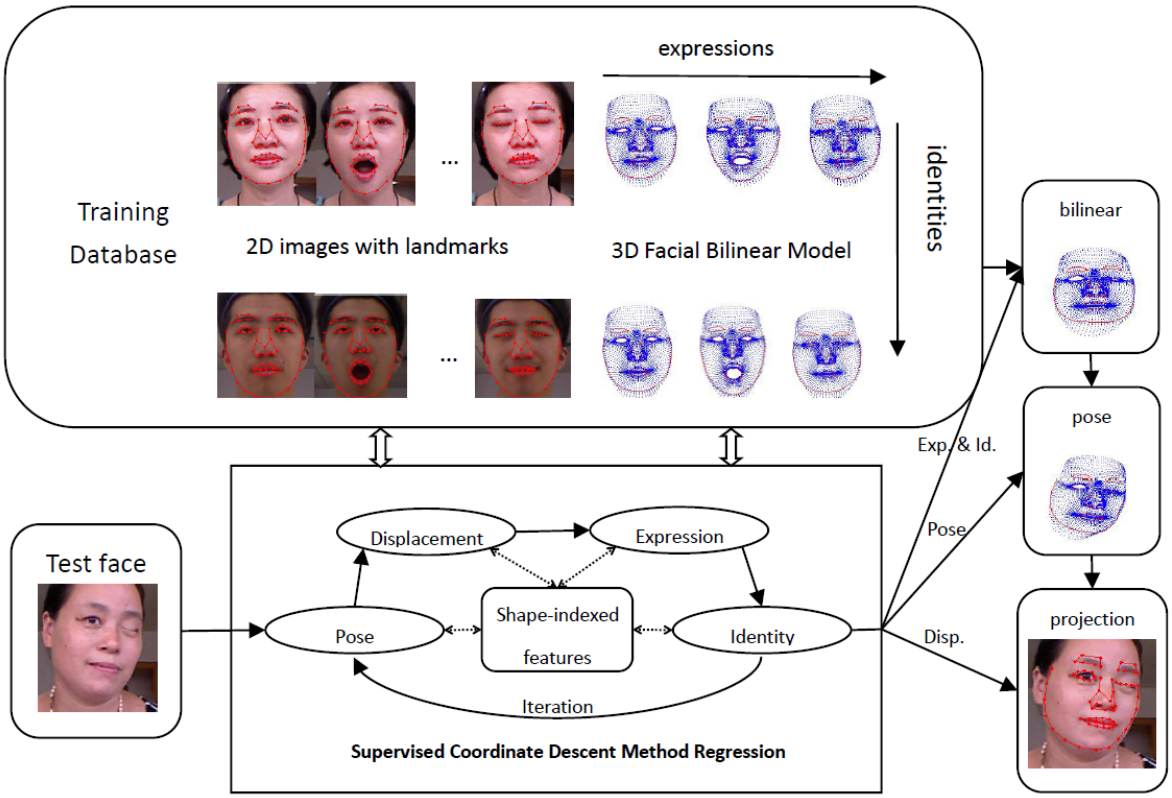

Figure 1: A flowchart of SCDM for face alignment and tracking.

to face alignment with pose variations. However, it is still a hard task to locate and track face landmarks with fast and extreme changes out of plane in the wild environment.

As an important augmentation to handle variations of face shape in 3D space, 3D face representation models have been studied widely.The Blendshape[17] is studied at early stage for describing a range of typical facial expressions in 3D domain, which provides a basic 3D face model for later 3D face regression methods, like 3D PCA faces[16], multi-linear model[18], and bilinear model[9].Blanz et al.[16] trained a PCA model on a 3D face dataset and represented a 3D face as a linear combination of 3D PCA components in their 3D morphable model. Vlasic et al.[18] proposed a multi-linear model to face transfer. In this model, a 3D face mesh is represented as a muliti-linear combination of principal components by visemes $\times$ expressions $\times$ identities modes. Biliear model[19] is a simple multi-linear model, which represents a 3D shape as the combination of principal components by expressions $\times$ identities modes. These methods can represent a face with blendshapes reliably and can be used to drive a avatar easily. Biliear model has been used in recent face tracking and performance capture, such as dynamic expression model[19] and dynamic displacement model [20].

Benefiting from sufficient 2D labeled face images and corresponding 3D models, 3D data-driven methods for face tracking and performance capture have been focused by researchers. Saragih et al. [21] combined 3D constrained local model with MPEG-4 face model to track face and transfer expressions. Yang et al. [22] trained linear PCA model on a public 3D face database offline, and located 2D landmarks by a variant of ASM [23] and solved coefficients of 3D PCA components based on perspective projection when tracking online. Similar to this work, Shuang et al. [24, 25] used a popular 2D landmark detector [14] to localize the facial landmarks and solved the pose, expression coefficients and identity coefficients based on perspective projection and a pre-trained 3D bilinear model. Cao et al. [19] proposed a 3D cascaded regression-based method for user-specific face tracking and animation. It integrates 2D land-

mark detection and the optimization of 3D projection parameters and expression parameters in a cascaded regression, which achieves comparable tracking result to a depth-based method [26]. Moreover, Cao et al. [20] proposed Dynamic Displacement Model by adding 2D displacements to perspective projection of 3D landmarks. In this method, pose matrix, expression coefficients and 2D displacements are regarded as a whole parameter, and a cascaded regression is learned on an augmented database. Then the identity coefficients and camera matrix are optimized based on a 3D bilinear model and perspective projection. Their method can track the facial landmarks more accurately when avoiding pre-processing done in their previous work.

Our work is inspired by 3D Dynamic Displacement Model and 2D-based cascaded regression. Since different parameters have different impact on 2D displacements of landmarks, training a cascaded regression directly for the whole parameters probably leads to cross impact, thus there is still a complex pro-processing needed for refine the coarse regressed result. In this paper a supervised coordinate decent method is proposed to learn the different types of parameters separately in a cascaded regression framework. The details are described in Section 3.

3 Supervised Coordinate Descent Method with a 3D Bilinear Model

3.1 Overview

This paper presents a 3D multi-parameters cascaded regression method based on a 3D Bilinear Model and Dynamic Displacement Model. The 3D Bilinear Model describes a bilinear representation of a 3D face mesh, and the Dynamic Displacement Model describes how to project 3D landmarks onto their corresponding 2D ones on the image plane. The parameters include pose matrix of perspective projection, expression coefficients and identity coefficients based on a trained 3D bilinear model, and 2D displacements based on Dynamic Displacement Model. The key idea in this work is that different kinds of parameters mapping with shape-indexed should be learned one by one through a supervised coordinate descent way. It means that other parameters keep fixed while learning a specific parameter mapping at a regression step. A flowchart of our supervised coordinate descent method is shown in Fig. 1.

3.2 3D Bilinear Model and 2D Displacement Model

Based on Vlasic's work [18] and Cao's work [20], a 3D face mesh can be represented as a bilnear combination by identity $\times$ expression modes. When the expression keeps fixed, a use-specific face can be represented as a linear combination of faces with different identities; When the identity keeps fixed, a face with specific expression can be represented as a linear combination of faces with a series of predefined expressions. A 3D face database with different identities and expressions can be represented as a 3 modes tensor by vertex $\times$ identity $\times$ expression. And N-mode SVD is used to compress the huge tensor to a small core tensor. For a 3D face V, its bilinear representation is described as Eq. 1:

$$V = C_r \times_2 \mathbf{u}^T \times_3 \mathbf{e}^T \tag{1}$$

where C_r is a core tensor, $\mathbf{u}$ is the identity coefficients, $\mathbf{e}$ is the expression coefficients, and $\times_k$ is product operator by mode-k.

The transformation from object coordinate to camera coordinate is obtained by a pose matrix including 3D rotation and translation, as shown in Eq. 2:

$$F = \mathbf{R}V + \mathbf{t} \tag{2}$$

where F is a 3D face mesh in camera coordinate, $\mathbf{R}$ is rotation matrix, and $\mathbf{t}$ is translation vector.

A 3D predefined landmark on a face mesh is projected onto the 2D image plane by perspective projection, but there usually exists difference between the directly projected 2D positions and the real 2D landmarks. The addition of a 2D displacement [20] can supply this drawback. The transformation from 3D landmarks to 2D landmarks is described in Eq. 3:

$$s_k = \prod(F^{(v_k)}) + d_k \tag{3}$$

where s_k is a 2D landmark in image plane, $F^{(v_k)}$ is its corresponding predefined vertex on 3D

face mesh F, d_k is the 2D displacement of s_k, and $\prod$ is the perspective projection operator: For a 3D vertex $\mathbf{p} = [X, Y, Z]^T$, its projected position in normalized 2D image plane is: $[x = X/Z, y = Y/Z]^T$, and the original 2D landmark is normalized by : $[x = (x_o - W/2)/W, y = (y_o - H/2)/W]^T$.

Usually, given L labeled 2D landmarks and corresponding predefined 3D landmarks on its 3D face mesh, the pose matrix $\{\mathbf{R}, \mathbf{t}\}$, identity coefficients $\mathbf{u}$, expression coefficients $\mathbf{e}$ and displacements $\mathbf{D} = \{\mathbf{d_k}\}$ can be solved by minimizing the Huber loss function applied to re-projected error between 2D landmarks and 3D landmarks:

$$arg\min_{P} \sum_{k=1}^{L} \|d_k\|_{\epsilon}^2 \tag{4}$$

where $\mathbf{P} = \{\mathbf{R}, \mathbf{t}, \mathbf{u}, \mathbf{e}, \mathbf{D}\}$,and d_k is computed based on the definition above:

$$d_k = s_k - \prod(\mathbf{R}(C_r \times_2 \mathbf{u}^T \times_3 \mathbf{e}^T) + \mathbf{t})^{(v_k)} \tag{5}$$

In the case that 2D landmarks has been detected, a nonlinear trust region optimization method like a sparse variant of the Levenberg Marquardt algorithm [27] can be used to solve pose and bilinear parameters, as Shuang et al. [24, 25] do. It is obvious is that a reliable landmark detector is necessary, but popular detectors are usually 2D-based and cannot capture large variations of pose and expression out of plane. So we are aiming at solving both 2D landmarks and 3D parameters simultaneously by a cascade regression with shape-indexed features.

3.3 *Supervised Coordinate Descent Method*

In a cascaded regression framework [8, 10, 13], given sufficient training samples, the optimal solution of a nonlinear cost function from a reasonable initialization can be solved by iteratively learning the mappings between current function output and the residual between current input and the optimal solution, as a supervised approximation of the gradient descent method.

As for face alignment and tracking in 2D videos, the nonlinear cost function is based on a feature description like SIFT [8], HOG[9],

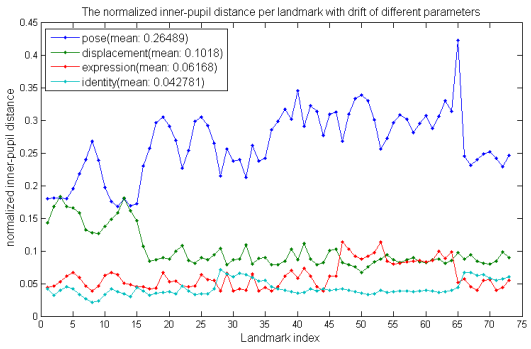

Figure 2: The normalized inner-pupil distance per landmark with drift of different parameters.

binary features[15], extracted around the landmarks. Given a 2D face image $\mathbf{I}$ and L 2D landmarks $\mathbf{s} = [s_1, ...s_k, ..., s_L]^T$, the feature function is denoted as $\mathbf{h}(\mathbf{I}, \mathbf{s})$. Based on the bilinear model and 2D displacement model, our input for the function is $\mathbf{P} = \{\mathbf{R}, \mathbf{t}, \mathbf{u}, \mathbf{e}, \mathbf{D}\}$, which generates the 2D landmarks $\mathbf{s}(\mathbf{P})$ by Eq. 3, then the feature function is represented as $\mathbf{h}(\mathbf{I}, \mathbf{s}(\mathbf{P}))$. Denoting $\mathbf{P}_0 = \{\mathbf{R}_0, \mathbf{t}_0, \mathbf{u}_0, \mathbf{e}_0, \mathbf{D}_0\}$ as the initialization, and $\mathbf{P}_* = \{\mathbf{R}_*, \mathbf{t}_*, \mathbf{u}_*, \mathbf{e}_*, \mathbf{D}_*\}$ as the optimal solution, the learning object is the residual $\Delta\mathbf{P} = \mathbf{P}_* - \mathbf{P}_0 = \{\Delta\mathbf{R}, \Delta\mathbf{t}, \Delta\mathbf{u}, \Delta\mathbf{e}, \Delta\mathbf{D}\}$. Our cost function is defined by minimizing the distance between features of predicted parameters and features of real parameters, as shown in Eq. 6:

$$f(\mathbf{P}_0 + \Delta\mathbf{P}) = \|\mathbf{h}(\mathbf{I}, \mathbf{s}(\mathbf{P}_0 + \Delta\mathbf{P})) - \mathbf{h}(\mathbf{I}, \mathbf{s}(\mathbf{P}_*))\|^2 \tag{6}$$

A standard cascaded regression for the whole parameters is:

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \mathbf{W}_k \mathbf{h}(\mathbf{I}, \mathbf{s}(\mathbf{P}_k)) + \mathbf{b}_k \tag{7}$$

where $\mathbf{W}_k$ is the learned mapping between the residual $\Delta\mathbf{P}_k = \mathbf{P}_* - \mathbf{P}_k$ and the current shape-indexed feature $\mathbf{h}(\mathbf{I}, \mathbf{s}(\mathbf{P}_k))$ on a training dataset, and $\mathbf{b}_k$ is the learned bias. However, it is sensitive to changes out of plane, and probably drift away from the real solution, because taking different types of parameters as whole can lead to the cross impact.

It is known that different parts of $\Delta\mathbf{P}$ make different contributions to the movement of 2D lamdmars $\Delta\mathbf{s} = \mathbf{s}(\mathbf{P}_0 + \Delta\mathbf{P}) - \mathbf{s}(\mathbf{P}_0)$: Pose

residual $\{\Delta\mathbf{R}, \Delta\mathbf{t}\}$ produces large-scale global movement; displacement residual $\Delta\mathbf{D}$ produces large-scale movements of contour landmarks and small-scale inner landmarks; expression residual $\Delta\mathbf{e}$ produces large-scale local movements of parts of landmarks; identity residual $\Delta\mathbf{u}$ produces small-scale global movement. We have evaluated the different movements caused by different parameters on three labeled face databases [28, 29, 30]. For each type of parameter, we fix others and replace it with a mean value, then we generate its 2D landmarks via our bilinear model and 2D displacement model. Following we compute the normalized inter-pupil distance between landmarks generated by real parameters and the replaced ones. As shown in Fig. 2, the errors decrease from pose to identity. An example of movements by different parameters also illustrates different-level contributions of different parameters.

If a cost function is differentiable, it will be a wise choice to solve different parts one by one with a coordinate decent method. Denoting $\mathbf{M} = \{\mathbf{R}, \mathbf{t}\}$, $\mathbf{P} = \{\mathbf{M}, \mathbf{D}, \mathbf{e}, \mathbf{u}\}$, An ideal format of the coordinate decent method for Eq. 7 is:

$$\begin{aligned}
\mathbf{M}_{k+1} &= \mathbf{M}_k - \alpha\mathbf{J}^T_{\mathbf{h},\mathbf{M}}(\phi_{\mathbf{M}_k} - \phi_*) \\
\mathbf{D}_{k+1} &= \mathbf{D}_k - \alpha\mathbf{J}^T_{\mathbf{h},\mathbf{D}}(\phi_{\mathbf{D}_k} - \phi_*) \\
\mathbf{e}_{k+1} &= \mathbf{e}_k - \alpha\mathbf{J}^T_{\mathbf{h},\mathbf{e}}(\phi_{\mathbf{e}_k} - \phi_*) \\
\mathbf{u}_{k+1} &= \mathbf{u}_k - \alpha\mathbf{J}^T_{\mathbf{h},\mathbf{u}}(\phi_{\mathbf{u}_k} - \phi_*)
\end{aligned} \tag{8}$$

where α is the learning ratio, $\mathbf{J}_{\mathbf{h},\mathbf{M}}$, $\mathbf{J}_{\mathbf{h},\mathbf{D}}$, $\mathbf{J}_{\mathbf{h},\mathbf{e}}$ and $\mathbf{J}_{\mathbf{h},\mathbf{u}}$ are Jacobi matrices of different parameters with chain rule by $\mathbf{h}$, $\phi_* = \mathbf{h}(\mathbf{I}, \mathbf{s}(\mathbf{P}_*))$ is the real feature, $\phi_{\mathbf{M}_k}$ is the feature obtained by $\{\mathbf{M}_k, \mathbf{D}_k, \mathbf{e}_k, \mathbf{u}_k\}$, $\phi_{\mathbf{D}_k}$ is obtained by $\{\mathbf{M}_{k+1}, \mathbf{D}_k, \mathbf{e}_k, \mathbf{u}_k\}$, $\phi_{\mathbf{e}_k}$ is obtained by $\{\mathbf{M}_{k+1}, \mathbf{D}_{k+1}, \mathbf{e}_k, \mathbf{u}_k\}$ and $\phi_{\mathbf{u}_k}$ is obtained by $\{\mathbf{M}_{k+1}, \mathbf{D}_{k+1}, \mathbf{e}_{k+1}, \mathbf{u}_k\}$.

Because the function $\mathbf{h}$ is not differentiable, the analytic solutions of Jacobi matrices cannot be computed directly. Alternatively, the decent mappings $\mathbf{W}_{\mathbf{M}_k} \approx \mathbf{J}^T_{\mathbf{h},\mathbf{M}}$, $\mathbf{W}_{\mathbf{D}_k} \approx \mathbf{J}^T_{\mathbf{h},\mathbf{D}}$, $\mathbf{W}_{\mathbf{e}_k} \approx \mathbf{J}^T_{\mathbf{h},\mathbf{e}}$ and $\mathbf{W}_{\mathbf{u}_k} \approx \mathbf{J}^T_{\mathbf{h},\mathbf{u}}$ between different parameters and function output can be learned as approximation of Jacobi matrices when a sufficient number of training samples are provided. It is noticed that a real feature is known during training but unknown for testing, so it is replaced with the mean feature $\overline{\phi}_*$ of all real features for training. Thus our supervised coordinate descent method is described as,

$$\begin{aligned}
\mathbf{M}_{k+1} &= \mathbf{M}_k - \alpha\mathbf{W}_{\mathbf{M}_k}(\phi_{\mathbf{M}_k} - \overline{\phi}_*) \\
\mathbf{D}_{k+1} &= \mathbf{D}_k - \alpha\mathbf{W}_{\mathbf{D}_k}(\phi_{\mathbf{D}_k} - \overline{\phi}_*) \\
\mathbf{e}_{k+1} &= \mathbf{e}_k - \alpha\mathbf{W}_{\mathbf{e}_k}(\phi_{\mathbf{e}_k} - \overline{\phi}_*) \\
\mathbf{u}_{k+1} &= \mathbf{u}_k - \alpha\mathbf{W}_{\mathbf{u}_k}(\phi_{\mathbf{u}_k} - \overline{\phi}_*)
\end{aligned} \tag{9}$$

At the k-th cascaded step, the coordinate decent mappings $\mathbf{W}_{\mathbf{M}_k}$, $\mathbf{W}_{\mathbf{D}_k}$, $\mathbf{W}_{\mathbf{e}_k}$ and $\mathbf{W}_{\mathbf{u}_k}$ are learned with training pairs $S_{\mathbf{M}_k} = \{\Delta\mathbf{M}^i_k, \Delta\phi^i_{\mathbf{M}_k}\}$, $S_{\mathbf{D}_k} = \{\Delta\mathbf{D}^i_k, \Delta\phi^i_{\mathbf{D}_k}\}$, $S_{\mathbf{e}_k} = \{\Delta\mathbf{e}^i_k, \Delta\phi^i_{\mathbf{e}_k}\}$ and $S_{\mathbf{u}_k} = \{\Delta\mathbf{u}^i_k, \Delta\phi^i_{\mathbf{u}_k}\}$, respectively. Each part is learned by a linear regression, as shown in Eq. 10:

$$\begin{aligned}
\mathbf{W}_{\mathbf{M}_k} &= arg\min_{\mathbf{W}_{\mathbf{M}_k}} \sum \|\Delta\mathbf{M}^i_k - \mathbf{W}_{\mathbf{M}_k}\Delta\phi^i_{\mathbf{M}_k}\|^2 \\
\mathbf{W}_{\mathbf{D}_k} &= arg\min_{\mathbf{W}_{\mathbf{D}_k}} \sum \|\Delta\mathbf{D}^i_k - \mathbf{W}_{\mathbf{D}_k}\Delta\phi^i_{\mathbf{D}_k}\|^2 \\
\mathbf{W}_{\mathbf{e}_k} &= arg\min_{\mathbf{W}_{\mathbf{e}_k}} \sum \|\Delta\mathbf{e}^i_k - \mathbf{W}_{\mathbf{e}_k}\Delta\phi^i_{\mathbf{e}_k}\|^2 \\
\mathbf{W}_{\mathbf{u}_k} &= arg\min_{\mathbf{W}_{\mathbf{u}_k}} \sum \|\Delta\mathbf{u}^i_k - \mathbf{W}_{\mathbf{u}_k}\Delta\phi^i_{\mathbf{u}_k}\|^2
\end{aligned} \tag{10}$$

where $\Delta\mathbf{M}^i_k = \mathbf{M}^i_* - \mathbf{M}^i_k$, $\Delta\phi^i_{\mathbf{M}_k} = \overline{\phi}_* - \phi^i_{\mathbf{M}_k}$, $\Delta\mathbf{e}^i_k = \mathbf{e}^i_* - \mathbf{e}^i_k$, $\Delta\phi^i_{\mathbf{e}_k} = \overline{\phi}_* - \phi^i_{\mathbf{e}_k}$, $\Delta\mathbf{u}^i_k = \mathbf{u}^i_* - \mathbf{u}^i_k$, $\Delta\phi^i_{\mathbf{u}_k} = \overline{\phi}_* - \phi^i_{\mathbf{u}_k}$, and $\Delta\mathbf{D}^i_k = \mathbf{D}^i_* - \mathbf{D}^i_k$, $\Delta\phi^i_{\mathbf{D}_k} = \overline{\phi}_* - \phi^i_{\mathbf{D}_k}$.

4 SCDM Application for Face Tracking

In this section we describe the details of a SCDM application to face tracking. A core tensor is first computed for the bilinear model by N-mode SVD on a public database. Then our SCDM regression model is trained on augmented public datasets according to the Section 3. Finally the trained SCDM is used for tracking face landmarks in face image sequences. A flowchart of our method is shown in Fig. 1.

4.1 Preparation for Training

We use 3D part of Facewarehouse database [28] to build the bilinear model. This database consists of 3D meshes of 150 different persons with

47 expressions for each. There are over $11k$ vertexes on each mesh. The original data is organized as a $11k\ vertexes \times 150\ identities \times 47\ expressions$ tensor. We compress it to a $11k\ vertexes \times 50\ identities \times 25\ expressions$ core tensor by N-mode SVD on 2-mode and 3-mode. The core tensor is used in the bilinear model.

There are three public datasets used for training our SCDM: Facewarehouse [28], LFW[29] and GTAV[30]. There are 5904 2D images in Facewarehouse, specially including 1152 images in left pose and 1152 images in right pose. Different with the 3D part of this database, it includes 150 different persons with 24 expressions in frontal pose, and also 48 persons with 24 expressions in left pose and right pose. 7258 images from 3010 persons in the LFW are used, and 1298 images from 44 persons in the GTAV are used. All the images are labeled with 73 facial landmarks and face bounding boxes are detected by a fast face detector[31].

Pose parameters, expression coefficients, identity coefficients, and displacements have not been given in raw databases, so we solve them with labeled 2D landmarks by minimizing the cost function in Eq. 5. The optimization algorithm is an efficient gradient descent method as Shuang et al. adopt [24]. Thus there are totally 14460 samples prepared for training. According to Yan at el.'s work [32] the HOG feature outperforms SIFT and others for shape-indexed feature in a cascaded regression, so HOG is adopted as our feature function.

4.2 Training the SCDM Regression Model

The mean shape $\overline{\mathbf{P}}_*$ of all parameters for training is used as the initialization $\mathbf{P}_0$ at the first training or testing step. At each training step, we learn pose mapping first, and then update current shape-indexed feature with the new pose, then we learn displacement mapping and update feature with the new displacement. Similar operations are followed by expression and identity according to Eq. 9 and Eq. 10. A few number of iterations are executed until the training error is under a threshold or the number of iterations reaches the maximum. 4 iterations are adopted in our application, which are enough for accurate result.

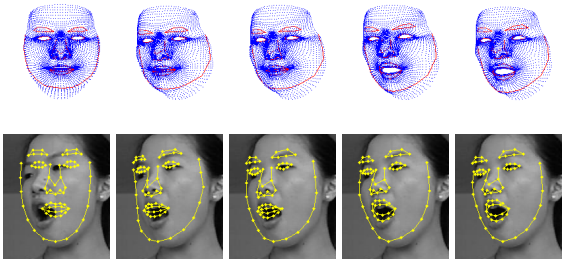

Figure 3: An example of run-time regression at the first iterative step.

4.3 Run-time Tracking

To initialize, the face bounding-box of the the first frame in a user-specific video is scanned by a face detector [31]. Then we use an object tracker [33] to capture the bounding box of the face in the subsequent frames. After that we crop a whole frame by the face bounding box, and conduct face alignment and tracking with our SCDM regression. Testing process also starts from the mean shape $\overline{\mathbf{P}}_*$, and then the different parameters and features are updated one by one with learned coordinate descent mappings based on Eq. 9. Iteration is executed for several times until reaching the maximum. An example of run-time regression at the first iterative step is shown in Fig. 3.

5 Experiments and Results

We collected 14 challenging videos online to evaluate the overall performance. The frame number of each video ranges from 190 to 900, and the total number of all the frames is 5135. The resolution of each video is 640×480. All the frames are labeled with 73 2D landmarks. Pose, expression coefficients, identity coefficients and 2D displacement are also computed based on our 3D bilinear model with labeled 2D landmarks, which are used for reconstruct 3D face meshes of these frames as the ground-truth 3D mesh.

There are different people talking with various expressions, poses, partial occlusions and illuminations among these sequences. The comparison with state-of-the-art 2D regression-based and model-based methods is performed.

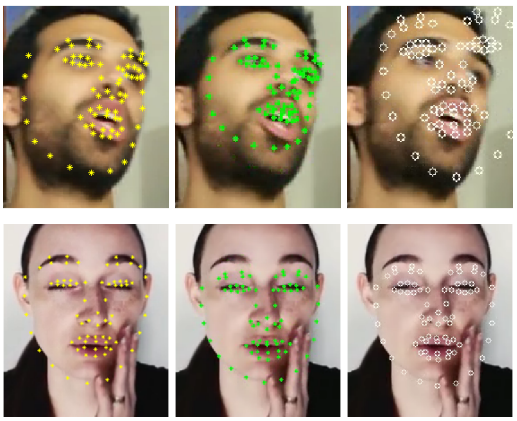

Figure 4: Landmark tracking comparison in challenging videos. From left to right: Ours, ERT, AAM

2D regression-based methods include state-of-the-art Ensemble of Regression Trees (ERT) [14], regressing Local Binary Features (LBF) [15]. 2D model-based methods include popular AAM-based methods: AAM-1[2] and AAM-2 [12]. A recent 3D Robust method [25] is also compared. These methods are also trained on the same three databases.

We evaluate the 3D mesh errors and 2D landmark errors of our SCDM and other methods. The measure metric of 2D landmark errors is the normalized inner-pupil distance, as used in popular methods [14, 15]. Because landmark tracking is usually used for capturing 3D facial performance, it is important to get accurate 3D meshes from 2D/3D tracked result. The mesh error is evaluated by calculating the normalized distance between a ground-truth mesh and its reconstructed one. Since the compared 2D-based methods do not directly provide 3D parameters for reconstructing a 3D mesh, we reconstruct their 3D meshes by: Solve 3D parameters with the predicted 2D landmarks as we do during training data preparation, and then generate 3D meshes with solved parameters. All the Experiments are done on a PC with the same hardware—i5 CPU(2.57 GHz), 16GB RAM and operating system —WIN-10. The mesh errors and landmark errors of different methods are shown in Tab. 1. It indicates that our SCDM can track landmarks more accurately. Moreover, 3D meshes directly generated by our SCDM are much more reliable than the reconstructed ones with 2D-based methods. Fig. 4 illustrates landmark tracking results in these challenging videos. It can be seen that our SCDM still keep stable localization when the pose and expression change drastically, or partial occlusion occurs. 2D-based methods fail to track in the same situation. More examples of our SCDM tracking in different videos are shown in Fig. 5 and a application demo is also presented in our supplementary video.

Table 1: Error comparison in challenging videos

Method	Mesh error	Landmark error
Our SCDM	**25.1**	**5.1**
3D Robust[25]	41.7	7.2
ERT[14]	92.3	19.2
LBF[15]	88.1	11.4
AAM-1[12]	97.3	21.9
AAM-2[2]	87.9	21.3

6 Conclusion

In this paper we present a novel data-driven face alignment and tracking method for monocular videos. A bilinear model is used as the 3D prior to strengthen cascaded regression processing. A novel supervised coordinate descent method is proposed to separately learning the descent mappings between different types of parameters with shape-indexed features individually, which is more stable than the previous whole-parameter regression works. Benefiting from 3D prior of the bilinear model, it shows more reliable capture ability than popular 2D-based methods while tracking face landmarks with variations out of plane. Our work is easy to be extended for performance-based facial animation and expression transfer in many customized AR/VR applications, which is considered as our future work.

References

[1] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.

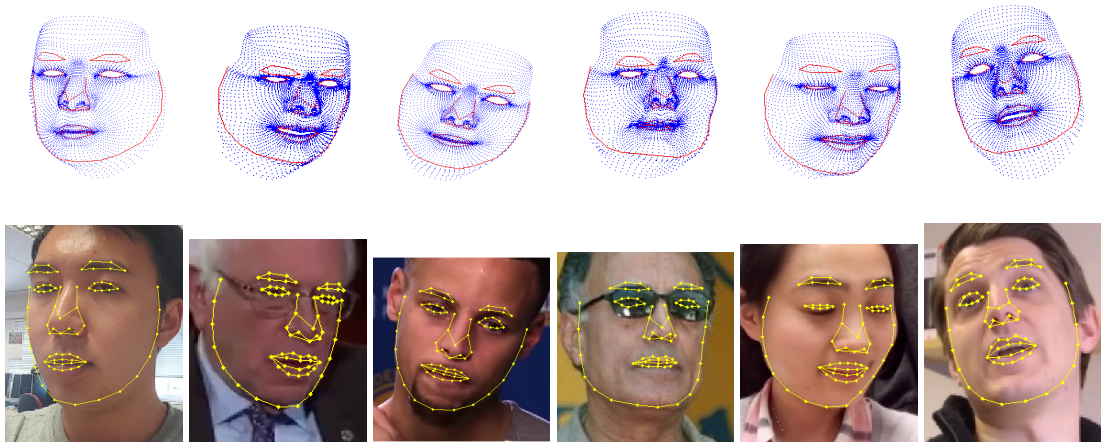

Figure 5: Landmark tracking results.

[2] Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, et al. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.

[3] Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J Taylor. View-based active appearance models. *Image Vis. Comput.*, 20(9):657–664, 2002.

[4] David Cristinacce and Timothy F. Cootes. Feature detection and tracking with constrained local models. *BMVC*, 41:929–938, 2006.

[5] Jose Gonzalezmora, Fernando De La Torre, Rajesh Murthi, Nicolas Guil, and Emilio L. Zapata. Bilinear active appearance models. In *ICCV*, pages 1–8, 2007.

[6] Hyung Soo Lee and Daijin Kim. Tensor-based aam with continuous variation estimation: application to variation-robust face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):1102–16, 2009.

[7] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression, December 27 2012. US Patent App. 13/728,584.

[8] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013.

[9] Zhen-Hua Feng, Guosheng Hu, Josef Kittler, William Christmas, and Xiao-Jun Wu. Cascaded collaborative regression for robust facial landmark detection trained using a mixture of synthetic and real images with dynamic weighting. *IEEE Trans. Image Process.*, 24(11):3425–3440, 2015.

[10] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *Int. J. Comput. Vis.*, 107(2):177–190, 2014.

[11] Iain Matthews and Simon Baker. Active appearance models revisited. *Int. J. Comput. Vis.*, 60(2):135–164, 2004.

[12] Rene Donner, Michael Reiter, Georg Langs, Philipp Peloschek, and Horst Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1690, 2006.

[13] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *CVPR*, pages 2664–2673, 2015.

[14] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, pages 1867–1874, 2014.

[15] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, pages 1685–1692, 2014.

[16] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d

faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[17] P Eckman and Wallace V Friesen. Facial action coding system (facs): A technique for the measurement of facial action, 1978.

[18] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *TOG*, volume 24, pages 426–433. ACM, 2005.

[19] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *TOG*, 32(4):41, 2013.

[20] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *TOG*, 33(4):43, 2014.

[21] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Real-time avatar animation from a single image. In *FG2011*, pages 117–124. IEEE, 2011.

[22] Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. Expression flow for 3d-aware face component transfer. In *TOG*, volume 30, page 60. ACM, 2011.

[23] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *ECCV*, pages 504–513. Springer, 2008.

[24] Shuang Liu, Xiaosong Yang, Zhao Wang, Zhidong Xiao, and Jianjun Zhang. Real-time facial expression transfer with single video camera. *Computer Animation and Virtual Worlds*, 27(3-4):301–310, 2016.

[25] Shuang Liu, Yongqiang Zhang, Xiaosong Yang, Daming Shi, and Jianjun Zhang. Robust facial landmark detection and tracking across poses and expressions for in-the-wild monocular video. *Computational Visual Media*, pages 1–15, 2017.

[26] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *TOG*, volume 30, page 77. ACM, 2011.

[27] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[28] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graph.*, 20(3):413–425, 2014.

[29] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[30] F Tarrés and A Rama. Gtav face database. *GVAP, UPC*, 2012.

[31] Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, and Stan Z Li. Face detection based on multi-block lbp representation. In *International Conference on Biometrics*, pages 11–18. Springer, 2007.

[32] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z. Li. Learn to combine multiple hypotheses for accurate face alignment. In *ICCV Workshops*, pages 392–396, 2013.

[33] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*. BMVA Press, 2014.