FACE IT!: A PIPELINE FOR REAL-TIME PERFORMANCE-DRIVEN FACIAL ANIMATION

Jilliam María Díaz Barros[♭ †] *Vladislav Golyanik*[♭ † §] *Kiran Varanasi*[♯] *Didier Stricker*[♭ †]

[♭] University of Kaiserslautern [†] DFKI [§] MPI for Informatics [♯] HTWK Leipzig

ABSTRACT

This paper presents a new lightweight approach for real-time performance-driven facial animation from monocular videos. We transfer facial expressions from 2D images to a 3D virtual character, by estimating the rigid head pose and non-rigid face deformation from detected and tracked 2D facial landmarks. We map the input face into the facial expression space of the 3D head model using blendshape models and formulate a lightweight energy-based optimization problem, which is solved by non-linear least squares at 18 FPS on a single CPU. Our method robustly handles varying head poses and different facial expressions, including moderately asymmetric ones. Compared to related methods, our approach does not require training data, specialised camera setups or graphics cards, and is suitable for embedded systems. We support our claims with several experiments.

Index Terms— Performance-driven animation, face tracking, head pose estimation, blendshape model

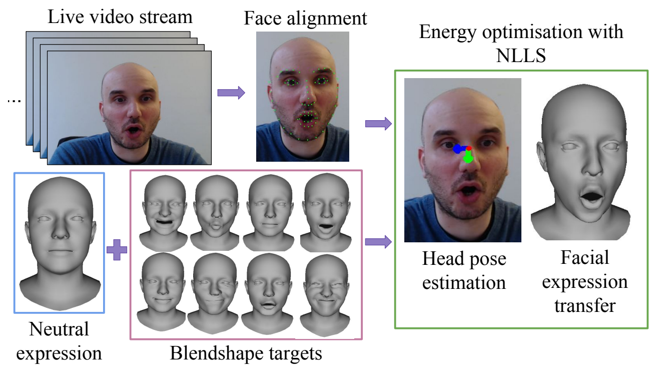

Fig. 1: Overview of the proposed pipeline for 2D-3D facial expression transfer: facial landmarks are detected in every incoming image and used to find the head pose and an optimal blendshape combination resembling the observed appearance. It supports moderately asymmetric expressions and runs on a CPU at real-time rates.

1. INTRODUCTION

Real-time performance-driven facial animation refers to the problem of capturing a live video stream of a person and animating a virtual avatar upon the observed facial expressions. Although this problem was first investigated in the context of the production of virtual avatars in films and computer games, such a system can also help in developing affective user interfaces in real world contexts. For example, the facial movements of the user can be used to assess the psychological state, or the user's intent in reaching for a specific tool, or his response to an interactive computer system. Some applications of such interfaces could be: driver monitoring in automobiles, service kiosks for patients in hospitals, or installations in theme-parks. To facilitate real time interaction, the system has to have very low hardware and data requirements, while being robust to a diverse range of human users.

Depending on the target application, there is always a trade-off between the quality of the input data and the complexity of the acquisition setup [1]. On one side, there are high-end systems used in the movie and gaming industries (*e.g.,* active 3D scanners or markers-based motion capture systems). Even though they provide realistic animations, they are intrusive and require substantial manual intervention. On the other side, there are simple, inexpensive and non-intrusive passive-scanning devices such as conventional monocular RGB cameras. Even though RGB or intensity-based facial-tracking methods have limited operational performance (*e.g.,* under varying illumination), monocular cameras are ubiquitous and flexible in installation and usage. Recently, several approaches based on commodity RGB-D sensors have been proposed [2, 1, 3]. Nevertheless, the most common visual data acquisition technology in every-day life constitutes RGB cameras as those embedded in mobile devices.

We aim at a lightweight method for real-time 3D facial character animation from monocular RGB or intensity images, which can be used in consumer-centric applications. In order to meet these requirements, the 2D facial tracking has to be robust, accurate and lightweight. Moreover, the setup should not rely on specialised hardware or markers. Fig. 1 provides an overview of the proposed pipeline. To summarise, the primary **contributions** of this paper are:

- A new real-time approach for performance-based facial animation from a monocular setup. We formulate 3D character animation as a lightweight energy-based optimization problem solved with non-linear least-squares (Sec. 4).
- To guarantee real-time constraints, our energy func-

This work was partially funded by the project VIDETE (01IW18002) of the German Federal Ministry of Education and Research (BMBF).

tional relies only on a sparse set of 2D facial landmarks, which are used to update the head pose and facial expressions (Sec. 4).
- A novel differentiable energy term for specifying the range of the weights of the blendshape targets (Sec. 4).
- A set of experiments for the validation of different aspects of the proposed method (Sec. 5).

2. RELATED WORK

In this section, we summarise state-of-the-art methods in monocular facial performance capture. For an extensive overview on this topic, we refer the reader to [4].

Several works propose approaches for non-rigid tracking and character animation which require either specialised setups, physical markers, RGB-D cameras or manual intervention [5, 1, 2, 6, 7, 8, 9]. Cao *et al.* [10] introduced a real-time facial animation approach from 2D data which requires a user-specific shape regressor trained in a preprocessing step with manual adjustments. In a follow-up work [11], they use public image datasets to train the regressor. [12] proposed a bilinear face model for identity and facial expression representation based on 2D or RGB-D data which can be used to generate a blendshape model of an actor or animate a 3D face.

Garrido *et al.* [13] introduced an offline approach for automatic reconstruction and animation of user-specific 3D face rigs from monocular videos. Their pipeline consists of three layers, where a parametric shape model is defined to encompass the subspace of facial identity, facial expression and fine-scale details such as wrinkles. Thies *et al.* [14] presented a real-time photo-realistic facial monocular reenactment approach. They track facial landmarks relying on a dense photometric consistency measure and use GPU-based iteratively reweighted least squares solver to achieve real-time frame rates. Liu *et al.* [15] introduced a real-time expression-transfer approach from 2D data which is adaptable to user-specific data. Their setup requires a preprocessing step for the acquisition of target-specific training images. The approach of Saito *et al.* [16] for real-time 3D facial performance capture from RGB data relies on accurate deep neural-network-based facial region segmentation and is robust to occlusions and significant head rotations.

Recently, some commercial facial performance capture software have been released, *e.g.* Apple's iPhone X app to animate a virtual character with its depth camera [17].

In our work, we use a monocular 2D setup and a lightweight energy-based minimization, which can be used in affective user interfaces. Our approach runs on a single CPU at real-time rates, while relying on robust facial landmark extraction. We do not require specialised hardware, preprocessing steps, manual intervention, large collections of training data or pre-trained target-specific regressors. Thus, our method addresses several limitations of existing 2D-to-3D facial expression transfer approaches.

3. OVERVIEW OF THE PROPOSED PIPELINE

An overview of our approach is shown in Fig. 1. In every incoming frame, we track a sparse set of facial landmarks for the recovery of rigid and non-rigid facial motion. Then, we define a linear parametric model with blendshapes and retrieve parameters modeling the head pose and facial expression by solving an energy-based optimization problem. Finally, we map the 2D facial expressions to a virtual 3D character which can be an animatable avatar or a person-specific 3D reconstruction obtained in a preprocessing step. Our method assumes a perspective projection model and known intrinsic camera parameters.

Blendshape Model. Blendshape models provide a simple yet robust technique for facial animation. They allow to parameterize facial expressions by building a linear weighted sum of basis elements [18]. The set of D blendshape targets defines the valid range of expressions and limits face movements to a subspace of dimension D. Unlike PCA-based models, each basis shape encodes a semantically meaningful expression.

The face model is given by a column vector $\mathbf{f} \in \mathbb{R}^{3p}$, composed of p vertices with the coordinates vectorized as $[x_0, y_0, z_0, x_1, y_1, z_1, ..., x_p, y_p, z_p]^T$. Similarly, each blendshape target is denoted by a vector $\mathbf{b}_k \in \mathbb{R}^{3p}$. The absolute blendshape model is then defined as:

$$\mathbf{f} = \sum_{k=0}^{n} w_k \mathbf{b}_k, \tag{1}$$

where $0 \leq w_k \leq 1$ are the blendshape weights [18]. We arrange n blendshape targets into a matrix $\mathbf{B} = [\mathbf{b}_0, ..., \mathbf{b}_n] \in \mathbb{R}^{3p \times n}$ defining the expression semantics transferable to the avatar. $\mathbf{b}_0$ denotes a face with neutral expression and $\mathbf{b}_i \; \forall i \neq 0$ corresponds to different base expressions. Concatenating w_k into a vector $\mathbf{w} \in \mathbb{R}^n$, Eq. (1) can be rewritten as:

$$\mathbf{f} = \mathbf{B}\mathbf{w}. \tag{2}$$

Similarly to commercial animation software such as Maya [19] and state-of-the-art methods [2, 13, 14], we use the delta form of the blendshape model, *i.e.,* each column of $\mathbf{B}$ is composed of offsets w.r.t $\mathbf{b}_0$: $\mathbf{B} = [\mathbf{b}_1 - \mathbf{b}_0, ..., \mathbf{b}_n - \mathbf{b}_0]$. As a result, multiple rows of $\mathbf{B}$ are composed of zero or near zero values. Eqs. (1) and (2) are then read as follows:

$$\mathbf{f} = \mathbf{b}_0 + \sum_{k=1}^{n} w_k(\mathbf{b}_k - \mathbf{b}_0) = \mathbf{b}_0 + \mathbf{B}\mathbf{w}. \tag{3}$$

Alignment of Blendshape Targets. We selected 44 blendshape targets from [20] and modified versions of the scans from [21] provided by [22]. These datasets provide targets with consistent topology and vertex-wise correspondences, with 5023 vertices and 9976 faces. Although the resulting variety of facial expressions is not as high as in [12], the low number of vertices makes them attractive for real-time applications on a single CPU. To compensate for the misalignment of the targets, we register the scans from [22] by solving the following constrained orthogonal Procrustes problem:

$$\mathbf{R} = \arg\min_{\mathbf{\Omega}} \|\mathbf{\Omega}\mathbf{A} - \mathbf{B}\|_{\mathcal{F}}\,,\ \text{s. t. } \mathbf{\Omega}^{\mathsf{T}}\mathbf{\Omega} = \mathbf{I}, \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ are two blendshape targets to register, $\mathbf{R}$ is the orthogonal matrix that maps $\mathbf{A}$ to $\mathbf{B}$ and $\|\cdot\|_{\mathcal{F}}$ denotes Frobenius norm. For every mesh, we extract $\mathbf{R} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^{\mathsf{T}}$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}} = \mathrm{svd}(\mathbf{M})$, and $\mathbf{\Sigma} = \mathrm{diag}(1\,1\,\det(\mathbf{V}\mathbf{U}^{\mathsf{T}}))$. Note that only a subset of points on the back side of the head is used for the alignment.

4. OUR TARGET ENERGY FUNCTIONAL

We propose to minimize a multi-objective energy function $\mathbf{E}(\boldsymbol{\gamma})$ for $\boldsymbol{\gamma} = (\mathbf{R}, \mathbf{t}, \mathbf{w})$, where $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation, *i.e.* the head pose, and $\mathbf{w}$ are the blendshape weights to recover the facial expression:

$$\mathbf{E}(\boldsymbol{\gamma}) = \omega_{\text{sparse}}\, \mathbf{E}_{\text{sparse}}(\boldsymbol{\gamma}) + \omega_{\text{prior}}\, \mathbf{E}_{\text{prior}}(\boldsymbol{\gamma}). \tag{5}$$

$\mathbf{E}_{\text{sparse}}$ is the data term that measures the model's head pose and facial expression, from $\mathbf{E}_{\text{pose}}$ and $\mathbf{E}_{\text{fit}}$, respectively, and the input 2D facial landmarks:

$$\mathbf{E}_{\text{sparse}}(\boldsymbol{\gamma}) = \omega_{\text{pose}}\, \mathbf{E}_{\text{pose}}(\mathbf{R}, \mathbf{t}) + \omega_{\text{fit}}\, \mathbf{E}_{\text{fit}}(\mathbf{w}). \tag{6}$$

$\mathbf{E}_{\text{prior}}$ comprises our regularization terms for the head pose, $\mathbf{E}_{\tau}$, and blendshape weights, $\mathbf{E}_{\beta}$ and $\mathbf{E}_{\sigma}$:

$$\mathbf{E}_{\text{prior}}(\boldsymbol{\gamma}) = \omega_{\tau}\, \mathbf{E}_{\tau}(\mathbf{R}, \mathbf{t}) + \omega_{\beta}\, \mathbf{E}_{\beta}(\mathbf{w}) + \omega_{\sigma}\, \mathbf{E}_{\sigma}(\mathbf{w}). \tag{7}$$

The weights $\omega_{\{\cdot\}}$ in Eqs. (5)-(7) define the contribution of each energy term to $\mathbf{E}(\boldsymbol{\gamma})$.

Non-rigid tracking. We detect 2D facial landmarks using the off-the-shelf face alignment approach proposed by [23], which aligns an ensemble of regression trees. We retrieve 68 facial landmarks around the jawline, lips, nose, eyes and eyebrows. Optical flow is then used to track the landmarks frame by frame. The correspondences of the 2D facial landmarks for every 3D blendshape target are known in advance.

Rigid head pose estimation. An initial estimate of the rigid head pose is computed based on [24]. A set of robust facial landmarks, including eyes canthi, both lateral and medial, and points around the nose, are used to minimize the reprojection error of the 3D-2D correspondences. For the other frames, we minimize the reprojection error of the $\eta = 68$ facial landmarks, using:

$$\mathbf{E}_{\text{pose}}(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{\eta} \|\pi(\mathbf{R}\mathbf{P}_i + \mathbf{t}) - \mathbf{p}_i\|_2^2\,, \tag{8}$$

where $\pi(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}^2$ denotes the perspective projection operator. $[\mathbf{R}|\mathbf{t}]$ are the extrinsic parameters of the camera, *i.e.* the pose, $\mathbf{P}$ and $\mathbf{p}$ are the 3D and 2D corresponding facial landmarks, respectively, and i is the index of the i-th feature point. As the calibration of the camera is known, Eq. (8) is minimized in the least squares sense with respect to the pose parameters $\mathbf{R}$ and $\mathbf{t}$, using Levenberg-Marquardt iteration.

Inspired by [2], we include an additional term, $\mathbf{E}_{\tau}$ to enforce temporal smoothness on the head pose:

$$\mathbf{E}_{\tau}(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{\eta} \|[\mathbf{r}|\mathbf{t}]_{t-2} - 2[\mathbf{r}|\mathbf{t}]_{t-1} + [\mathbf{r}|\mathbf{t}]_{t}\|_2^2\,, \tag{9}$$

with $\mathbf{r} = [r_x, r_y, r_z]$ being the angle-axis representation of the rotation around the x, y and z axes and t the timeframe.

2D-3D Transfer of Facial Expressions. To recover the facial expression, we minimize the reprojection error of the η facial landmarks using the blendshape model in Eq. (3):

$$\mathbf{E}_{\text{fit}}(\mathbf{w}) = \sum_{i=1}^{\eta} \left\|\pi(\mathbf{b}_0^i + \bar{\mathbf{B}}^i\mathbf{w}) - \mathbf{p}_i\right\|_2^2. \tag{10}$$

Since the elements of the blendshape basis are not orthogonal, *i.e.* not linearly independent, the same facial expression could be recovered using different target combinations. Thus, we include a sparsity prior based on [2], defined as a ℓ_1-norm:

$$\mathbf{E}_{\sigma}(\mathbf{w}) = \sum_{k=1}^{n} ||\mathbf{w}||_1. \tag{11}$$

To avoid compensation artifacts, the weights are usually set in the range [0, 1]. This implies that we need a differentiable function so that in the range [0,1] it generates a zero penalty, and a large penalty otherwise. Inspired by [25], we define such function by adding two smooth Heaviside function approximations:

$$\mathbf{E}_{\beta}(\mathbf{w}) = \frac{\pi}{4}\left(\tan^{-1}\left(\frac{\mathbf{w} - a}{b}\right) - \tan^{-1}\left(\frac{\mathbf{w} + a - 1}{b}\right)\right) + c, \tag{12}$$

with $a = 1.002, b = 2 \cdot 10^{-5}$ and $c = 2.5$ (see Fig. 2).

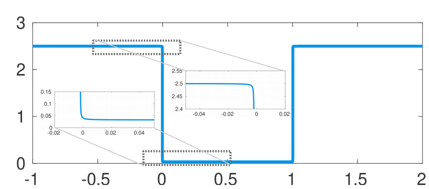

Fig. 2: Our function — a sum of two Heaviside approximations — to limit the blendshape target weights $\mathbf{w}$ to range [0,1].

In contrast to [1, 2], we do not use any temporal coherence constraints on the blendshape weights.

Energy Minimization. We solve an energy-based optimization problem for 50 parameters: 6 DOF head pose and 44 parameters (the number of blendshape targets) for the facial expression, with a total of 68×2 residuals for $\mathbf{E}_{\text{sparse}}$, 6 for $\mathbf{E}_{\tau}$ and 1 for each $\mathbf{E}_{\beta}$ and $\mathbf{E}_{\sigma}$.

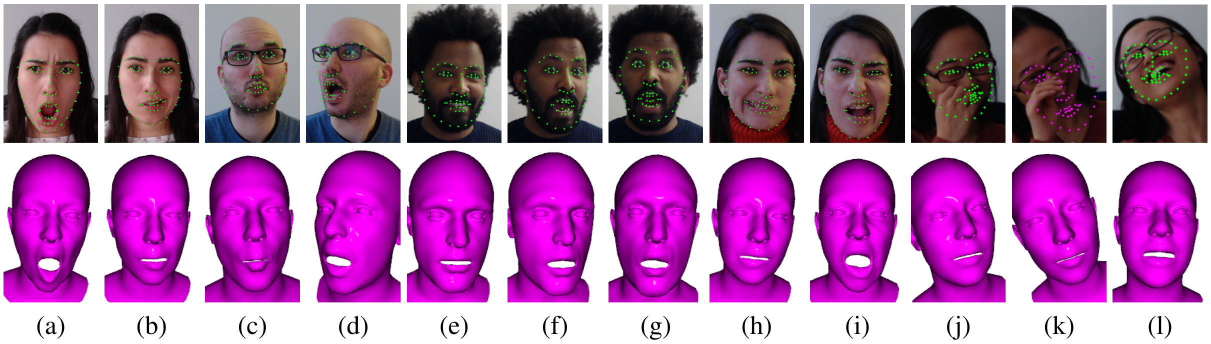

Fig. 3: Results of our performance-driven facial animation approach. Top: Input images with detected facial landmarks. Bottom: Animated 3D characters. (a)-(i) demonstrate the variety of supported facial expressions; (j)-(l) show results under occlusion and extreme poses.

5. RESULTS

The pipeline was implemented in C++, using DLib [26] and Ceres [27]. We used a commodity computer with an Intel Xeon(R) W3520 processor and 8GB of RAM. The videos were captured using a Logitech C920 HD Pro webcam, with a resolution of 640×480. Representative results are shown in Fig. 3 and in the supplemental material.

Runtime analysis. The average runtime for ~1000 frames was 18FPS. Face alignment took 23.7 ms, while the energy minimization took 31.6 ms on average per frame.

We also investigated how the internal number of iterations in the energy function affected the output and runtime. Fig. 4 (left) shows the resulting head poses and facial expressions for one frame. To select a fixed set of parameters for all the experiments, we considered the trade-off between accuracy and time consumption. In Fig. 4 (right), head pose required around 15 iterations to converge, while the estimation of the weights of the blendshape targets did not converge in the first 50 iterations. However, 15 iterations were enough to transfer similar facial expressions to the target (see Fig. 3).

Head pose evaluation. We evaluated the head pose using the Boston University (BU) head tracking database [28], which contains 45 video sequences of individuals performing different head movements. We used the mean absolute error (MAE) to compare the rotation to other methods of the state of the art (see Table 1). We report translation errors (in inches) of 2.27, 0.90 and 2.04 for the X, Y and Z axes, respectively. The errors of our approach are comparable to other methods, although they are intended to face alignment and head pose estimation only, without any facial performance capture.

Discussion. Our pipeline can handle occlusions caused by glasses, long hair and beard (see Fig. 3: (a)-(f)). Although the face alignment has limited performance for facial expressions with strong asymmetry, our method can transfer such expressions sufficiently (see Fig. 3: (b), (h) and (i)). Our approach is constrained by the facial landmarks detection and tracking, particularly under large head rotations and occlusions (see Fig. 3: (j)-(l)). Similarly to other methods using RGB data, the method is sensitive to low illumination (Fig. 3 (g)).

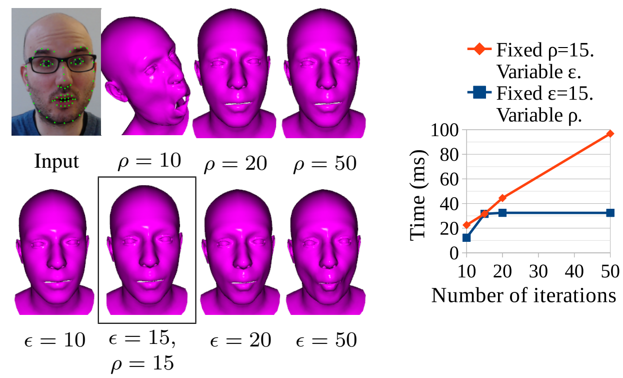

Fig. 4: Left: Energy minimization results. ρ and ϵ are the number of iterations used to estimate the head pose and facial expression, respectively. Top: Head pose for a fixed $\epsilon = 15$. Bottom: Facial expression for a fixed $\rho = 15$. Selected parameters: $\epsilon = 15$ and $\rho = 15$. **Right:** Time consumption of the energy minimization for varying number of iterations of ρ and ϵ, for ~1000 frames.

Method	Roll	Pitch	Yaw	Average
Jeni *et al.* [29]	2.41	**2.66**	3.93	**3.0**
Wu *et al.* [30]	3.1	5.3	4.9	4.43
Gou *et al.* [31]	3.3	4.8	5.1	4.4
Diaz Barros *et al.* [24]	**2.32**	3.41	**3.90**	3.21
Ours	2.35	3.62	4.38	3.45

Table 1: Comparison of rotational MAE on the BU dataset.

6. CONCLUSIONS

We presented a real-time pipeline for performance-driven facial animation based on monocular systems. The head pose and facial expressions are formulated as a lightweight optimization problem, using blendshape models. Our method runs at 18 FPS on a single CPU and does not require training data nor special camera setups. These features make our pipeline suitable for embedded systems, with potential for affective user interfaces.

7. REFERENCES

[1] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly, "Realtime performance-based facial animation," in *ACM Trans. Graph.* ACM, 2011, vol. 30.

[2] Sofien Bouaziz, Yangang Wang, and Mark Pauly, "Online modeling for realtime facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, 2013.

[3] Pei-Lun Hsieh, Chongyang Ma, Jihun Yu, and Hao Li, "Unconstrained realtime facial performance capture," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[4] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt, "State of the art on monocular 3D face reconstruction, tracking, and applications," in *Computer Graphics Forum*. Wiley Online Library, 2018, vol. 37.

[5] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly, "Face/off: Live facial puppetry," in *SIGGRAPH/Eurographics Symposium on Computer animation*. ACM, 2009, pp. 7–16.

[6] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler, "Realtime facial animation with on-the-fly correctives.," *ACM Trans. Graph.*, vol. 32, 2013.

[7] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt, "Real-time expression transfer for facial reenactment.," *ACM Trans. Graph.*, vol. 34, no. 6, 2015.

[8] Roger Blanco i Ribera, Eduard Zell, JP Lewis, Junyong Noh, and Mario Botsch, "Facial retargeting with automatic range of motion alignment," *ACM Trans. Graph.*, vol. 36, no. 4, 2017.

[9] Yudong Guo, Juyong Zhang, Lin Cai, Jianfei Cai, and Jianmin Zheng, "Self-supervised cnn for unconstrained 3d facial performance capture from a single RGB-D camera," *arXiv preprint:1808.05323*, 2018.

[10] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou, "3D shape regression for real-time facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, 2013.

[11] Chen Cao, Qiming Hou, and Kun Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, no. 4, 2014.

[12] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," *IEEE Trans. on Visualization and Computer Graphics*, vol. 20, no. 3, 2014.

[13] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt, "Reconstruction of personalized 3D face rigs from monocular video," *ACM Trans. Graph.*, vol. 35, no. 3, June 2016.

[14] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[15] 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*, vol. 27, 2016.

[16] Shunsuke Saito, Tianye Li, and Hao Li, "Real-time facial segmentation and performance capture from RGB input," in *European Conf. on Computer Vision (ECCV)*. Springer, 2016.

[17] "Use animoji on your iphone x and ipad pro," `https://support.apple.com/en-gb/HT208190`, 2018, [Online; accessed 01-Feb-2019].

[18] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng, "Practice and theory of blendshape facial models.," *Eurographics (State of the Art Reports)*, vol. 1, no. 8, 2014.

[19] Sham Tickoo, *Autodesk Maya 2018: A Comprehensive guide*, CADCIM Techonologies, 2017.

[20] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proceedings of the European Conf. on Computer Vision (ECCV)*, 2018.

[21] Darren Cosker, Eva Krumhuber, and Adrian Hilton, "A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling," in *Int. Conf. on Computer Vision (ICCV)*, 2011.

[22] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. Graph.*, vol. 36, no. 6, 2017.

[23] Vahid Kazemi and Josephine Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[24] Jilliam Maria Diaz Barros, Bruno Mirbach, Frederic Garcia, Kiran Varanasi, and Didier Stricker, "Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation," in *Winter Conf. on Applications of Computer Vision (WACV)*. IEEE, March 2018.

[25] Tony F. Chan and Luminita Vese, "Active contours without edges," *Trans. on Image Processing*, vol. 10, 2001.

[26] Davis E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, 2009.

[27] Sameer Agarwal, Keir Mierle, and Others, "Ceres solver," `http://ceres-solver.org`.

[28] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, 2000.

[29] László A Jeni, Jeffrey F Cohn, and Takeo Kanade, "Dense 3D face alignment from 2D video for real-time use," *Image and Vision Computing*, vol. 58, pp. 13–24, 2017.

[30] Yue Wu, Chao Gou, and Qiang Ji, "Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion," *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] Chao Gou, Yue Wu, Fei-Yue Wang, and Qiang Ji, "Coupled cascade regression for simultaneous facial landmark detection and head pose estimation," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2017.