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 deformations from detected and tracked 2D facial landmarks. We map the input face into the facial expression space of the 3D head model using blendshape model and formulate a lightweight energy-based optimization problem which is solved by non-linear least squares at 18 frames per second 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, specialized 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

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 virtual avatar generation for films and computer games, such a system can also help in developing affective user interfaces in real world contexts. For example, facial movements of a user can be used to assess his psychological state, intent in reaching for a specific tool or 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 marker-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 everyday 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-driven 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 fulfill real-time constraints, our energy functional relies
been released (some commercial facial performance capture software has deep neural network based facial region segmentation and is facial performance capture from RGB data relies on accurate images. The approach of Saito et al. is adaptable to user-specific data. Their setup requires a pre-processing step from 2D data which is achieved iteratively reweighted least squares solver to achieve a dense photometric consistency measure and use GPU-acceleration. They track facial landmarks relying on presented a real-time photo-realistic facial monocular reenactment approach. Thies et al. [22] encompass the subspace of facial identity, facial expression face rigs from monocular videos. Their pipeline consists of automatic reconstruction and animation of user-specific 3D face models for identity and facial expression representation obtained in a preprocessing step. Our method assumes 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. 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 $f \in \mathbb{R}^{3p}$ 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 $b_k \in \mathbb{R}^{3p}$. The absolute blendshape model is then defined as:

$$f = \sum_{k=0}^{n} w_k b_k,$$  \hspace{1cm} (1)

where $0 \leq w_k \leq 1$ are the blendshape weights [18]. We arrange $n$ blendshape targets into a matrix $B = [b_0, ..., b_n] \in \mathbb{R}^{3p \times n}$ defining the expression semantics transferable to the avatar. $b_0$ denotes a face with neutral expression and $b_i \neq 0$ corresponds to different base expressions. After concatenating $w_k$ into a vector $w \in \mathbb{R}^n$, Eq. (1) can be rewritten as:

$$f = B w.$$  \hspace{1cm} (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 $B$ is composed of offsets w.r.t $b_0$: $B = [b_1 - b_0, ..., b_n - b_0]$. As a result, multiple rows of $B$ are composed of zero or near zero values. Then, Eqs. (1) and (2) read as follows:

$$f = b_0 + \sum_{k=1}^{n} w_k (b_k - b_0) = b_0 + B w.$$  \hspace{1cm} (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

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 the follow-up [11], they use public image datasets to train the regressor. [12] describes 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 has been released (e.g., Apple’s iPhone X app to animate a virtual character with its depth camera [17]).

In this work, we use a monocular 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. We track a sparse set of facial landmarks in every incoming frame 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 expressions 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 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. 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.
applications on a single CPU. To compensate for the slight misalignment of facial expressions, we register the scans from [22] by solving the following constrained orthogonal Procrustes problem:

$$\mathbf{R} = \arg \min_{\mathbf{\Omega}} \| \mathbf{\Omega A} - \mathbf{B} \|_F, \quad \text{s. t.} \quad \mathbf{\Omega}^T \mathbf{\Omega} = \mathbf{I},$$  

(4)

where \( \mathbf{A} \) and \( \mathbf{B} \) are the two blendshape targets to be registered, \( \mathbf{R} \) is the orthogonal matrix that maps \( \mathbf{A} \) to \( \mathbf{B} \) and \( \| \cdot \|_F \) denotes Frobenius norm. For every mesh \( \mathbf{M} \), we extract \( \mathbf{R} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \), where \( \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \text{svd}(\mathbf{M}) \), and \( \mathbf{\Sigma} = \text{diag}(1/\text{det}(\mathbf{V} \mathbf{U}^T)) \). 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 \( \mathbf{E}(\gamma) \) for \( \gamma = (\mathbf{R}, \mathbf{t}, \mathbf{w}) \), where \( \mathbf{R} \) and \( \mathbf{t} \) are the rotation and translation, i.e., the head pose, and \( \mathbf{w} \) is the vector of blendshape weights for the facial expression recovery:

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

(5)

\( \mathbf{E}_{\text{sparse}} \) is the data term that measures the model’s head pose and facial expression from the input 2D facial landmarks. It consists of \( \mathbf{E}_{\text{pose}} \) and \( \mathbf{E}_{\text{fit}} \):

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

(6)

\( \mathbf{E}_{\text{prior}} \) comprises regularization term for the head pose \( \mathbf{E}_{\gamma} \) as well as constraints on the blendshape weights \( \mathbf{E}_{\beta} \) and \( \mathbf{E}_{\sigma} \):

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

(7)

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

Non-rigid tracking. We detect 2D facial landmarks using the off-the-shelf face alignment approach [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 between the 2D facial landmarks and points on 3D blendshape targets are known in advance per design.

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

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

(8)

where \( \pi(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \) denotes the perspective projection operator. \( [\mathbf{R}|\mathbf{t}] \) are the extrinsic camera parameters (camera pose), \( \mathbf{P} \) and \( \mathbf{p} \) are the 3D and 2D corresponding facial landmarks, respectively, and \( i \) is the feature point index. 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 \( \mathbf{E}_{\gamma} \) to enforce temporal smoothness on the head pose:

$$\mathbf{E}_{\gamma}(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{\eta} ||| \mathbf{r} | \mathbf{t}_{i-1} - \mathbf{r} | \mathbf{t}_i ||_2^2, \quad \text{with the angle-axis representation of the rotation \( r = [r_x, r_y, r_z] \) around the \( x, y \) and \( z \)-axes and \( t \) being the timeframe.}$$  

(9)

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

$$\mathbf{E}_{\text{fit}}(\mathbf{w}) = \sum_{k=1}^{n} \| \pi(\mathbf{b}_0 + \mathbf{B}_k \mathbf{w}) - \mathbf{p}_i \|_2^2.$$  

(10)

Since the elements of the blendshape basis are not orthogonal, i.e., not linearly independent, the same facial expression can be recovered using different target combinations. Thus, we include a sparsity prior based on [2] defined as an \( \ell_1 \)-norm:

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

(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. We define such function by adding two smooth Heaviside function approximations [25]:

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

(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 — for keeping the blendshape target weights \( \mathbf{w} \) in the range \([0,1]\).](image-url)

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 for head pose and 44 parameters (the number of blendshape targets) for the facial expression, with a total of \( 68 \times 2 \) residuals for \( \mathbf{E}_{\text{sparse}} \), six for \( \mathbf{E}_{\gamma} \) and one for each \( \mathbf{E}_{\beta} \) and \( \mathbf{E}_{\sigma} \).
5. RESULTS

The pipeline is implemented in C++ using DLib [26] and ceres solver [27]. We test it on a commodity computer with an Intel Xeon(R) W3520 processor and 8GB of RAM. The videos were captured with a Logitech C920 HD Pro webcam at the resolution of $640 \times 480$ pixels. Representative results are shown in Fig. 3 and in the supplemental material.

**Runtime analysis.** The average throughput for $\sim 1000$ frames amounts to 18 frames per second. Face alignment takes 23.7 ms while the energy minimization takes 31.6 ms per frame on average. We also investigate how the internal number of iterations in the energy function affects 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 experiments, we consider the trade-off between accuracy and runtime. In Fig. 4-(right), head pose requires around 15 iterations to converge, while the estimation of the blendshape target weights does not entirely converge during the first 50 iterations. Still, 15 iterations are sufficient to transfer similar facial expressions to the target (see Fig. 3).

**Head pose evaluation.** We evaluate the head pose using the Boston University (BU) head tracking database [28] which contains 45 video sequences of individuals performing different head movements. We use the mean absolute error (MAE) to compare the rotation to other 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 close to the other errors, although the compared methods are intended for 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 (Fig. 3: (a)-(f)). Although the face alignment has limited performance for facial expressions with strong asymmetry (Fig. 3: (b), (h) and (i)), our method transfers such expressions adequately. The performance of our approach is clearly affected by the accuracy of facial landmark detection and tracking, large head rotations and occlusions (Fig. 3: (j)-(l)). Similarly to other methods using RGB data, our method is sensitive to low illumination (Fig. 3-(g)).

6. CONCLUSIONS

We present a real-time pipeline for performance-driven facial animation for monocular systems. The head pose and facial expression recovery are formulated as a lightweight optimization problem with blendshapes. Our pipeline runs at 18 frames per second on a single CPU and does not require training data nor special hardware which makes it suitable for embedded systems, with potential for affective user interfaces.
7. REFERENCES


