Volume xx (200y), Number z, pp. 1–11

Real-Time Facial Expression Transformation for Monocular RGB Video

Luming Ma† and Zhigang Deng‡

Department of Computer Science
University of Houston, Houston, TX, USA

Figure 1: Our automated approach can real-time transform the neutral facial expression in input video (top) to happy expression (bottom).

Abstract

This paper describes a novel real-time end-to-end system for facial expression transformation, without the need of any driving source. Its core idea is to directly generate desired and photo-realistic facial expressions on top of input monocular RGB video. Specifically, an unpaired learning framework is developed to learn the mapping between any two facial expressions in the facial blendshape space. Then, it automatically transforms the source expression in an input video clip to a specified target expression through the combination of automated 3D face construction, the learned bi-directional expression mapping, and automated lip correction. It can be applied to new users without additional training. Its effectiveness is demonstrated through many experiments on faces from live and online video, with different identities, ages, speeches, and expressions.

Keywords: real-time face reconstruction, expression transformation, facial animation

CCS Concepts

•**Computing methodologies** → **Animation; Image-based rendering;**

1. Introduction

Realistic facial expression creation and transformation has been a long-standing problem in computer graphics and computer vision. Thus far, popular approaches usually require a driving source or the combination of multiple ones, such as capturing a subject's performance and then transferring it to virtual faces [DCFN06, SLS*12, TZN*15, AECOKC17, VBPP05], and speech-driven facial animation [Bra99,EGP02,DNL*06,DN08a,SSKS17,TKY*17,KAL*17]. However, these methods only provide a way to drive the face to follow the performed expressions, and do not provide the flexibility to synthesize new facial expressions on top of the original, such as being happier or being angry instead of neutral while speaking. In addition, the transferring approaches usually break the synchro-

† lma15@uh.edu
‡ zdeng4@uh.edu

nization between the face reenactment and audio from the source video and thus are unsuitable for speech video.

An ideal solution to the above problem is to generate desired and photo-realistic facial expressions on top of the source expression of an input monocular video clip, without the need of any driving sources. One straightforward way is to per-frame edit the source expression. Clearly, this is quite tedious and time-consuming; furthermore, it is non-trivial to ensure the temporal dynamics of the edited facial expression. Another technical path explored previously is to learn a mapping $\mathbb{M} : X \Rightarrow Y$ between two sequences X and Y that are semantically aligned. For example, image-to-image translation approaches [IZZE16, JAFF16] are used to address this problem by transferring image style or content between image pairs. However, they often require a large number of aligned face images of various identities, expressions and environment lighting as the training data. The problem becomes even more difficult when dealing with speech video, because lip synchronization needs to be preserved in the transformed and re-rendered video, besides the intrinsic complexity and subtlety of facial expressions.

To tackle the above problem, we propose a complete pipeline to real-time transform the source expression of the subject in an input (source) monocular RGB video clip to a user-specified target expression and then photo-realistically re-render the same performance but with the target expression. The generated facial expression sequence is temporally dynamic, coherent, and lip-synchronized to the source audio. Specifically, the main components of our system (Figure 2) are briefly described as follows:

- **3D facial performance reconstruction**. We start the process by reconstructing 3D facial performance from an input video clip. The facial performance is reconstructed by detecting 2D facial landmarks and then estimating the 3D facial deformations (i.e., bilinear identity and expression parameters) as well as 3D head poses and camera parameters. Then, we estimate the lighting conditions and facial albedo using the shading cues (see §3).
- **CycleGAN-based expression mapping**. We train a CycleGAN based expression mapping model to learn the bi-directional mapping between two facial expressions based on an unpaired face video dataset. Different from existing image based approaches using Convolutional Neural Networks (CNN), we learn the model in a facial blendshape space. We will discuss its architecture and training in §4.
- **Expression transformation and lip correction**. We apply the trained CycleGAN-based expression mapping to a small window of neighboring frames for smoother and more confident results in target expression (see §5). We then synchronize lip motion to the source audio by minimizing the distance of key lip vertices before and after transformation in §5.1.
- **Composition**. We re-render the face with the transformed facial expression, original head poses, lighting, and albedo map. Since the shapes of the mouth might be significantly different between the original source expression and the transformed target expression, our approach warps the mouth region from the source frame into the transformed face. Finally, we complete the composition by putting the re-rendered face and the warped mouth regions in different image layers and blending back to the original input video (see §5.2).

The main contributions of this work can be summarized below:

- a complete, real-time pipeline for facial expression transformation on monocular RGB video, without the need of any driving sources;
- a data-driven approach to solve the automatic bi-directional transformation between a pair of facial expressions; and
- a novel optimization formula for temporally coherent and lip synchronized expression transformation.

2. Related Work

Our system reconstructs 3D facial performance of the subject from an input video clip, transforms his/her facial expression to a user-specified target expression, and finally photo-realistically re-renders his/her performance with the target expression. Thus, the literature review in this section is specifically focused on the recent, most related efforts on facial reconstruction, video-based face reenactment, expression manipulation, and image transformation. Readers can refer to [DN08b, LAR*14] for more comprehensive reviews on facial animation.

2.1. Face Reconstruction

Many face reconstruction methods have been proposed over the past decades. A significant portion is data-driven methods stemming from the seminal morphable face model [BV99, BBPV03] where a statistical model is employed to reconstruct facial identity and expression from images and/or video. Vlasic et al. extend this method with a multi-linear model that is constructed along the axes of vertices, identities, expressions, and visemes [VBPP05]. Similarly, the FaceWarehouse [CWZ*14] employs a bilinear face model that consists of 47 FACS-based [EF78] blendshapes for each identity. Recently, Cao and colleagues proposed regression based real-time performance capture from RGB video [CWLZ13, CHZ14, CBZB15]. The works of [SWTC14, GZC*16] capture fine facial details based on the shape-from-shading technique from monocular RGB video in an off-line manner.

On the other hand, physics based methods [LTW95, SNF05, IKKP17] build an anatomically accurate, volumetric model with facial musculature, tissue, and skeleton. Muscle activations in the physical model are capable of simulating contacts and collisions of face with external objects. Ichima et al. introduced volumetric blendshapes combining intuitive control of blendshapes and the capability of realistic physics-based simulation [IKNDP16]. Similarly, Cong et al. create a blendshape system for facial muscles that drives underlying anatomical and biomechanical muscle dynamics [CBF16].

2.2. Video-based Face Reenactment

One type of face reenactment methods replaces the face identity of a target video clip with a source actor while preserving its original performance [DSJ*11, GVR*14]. On the other hand, the works of [XCLT14, TZN*15, TZS*16] transfer the facial expression of a source actor to a target video clip. Li et al. presented a data-driven solution to synthesize the target video from a driving actor by retrieving frames from a pre-recorded dataset

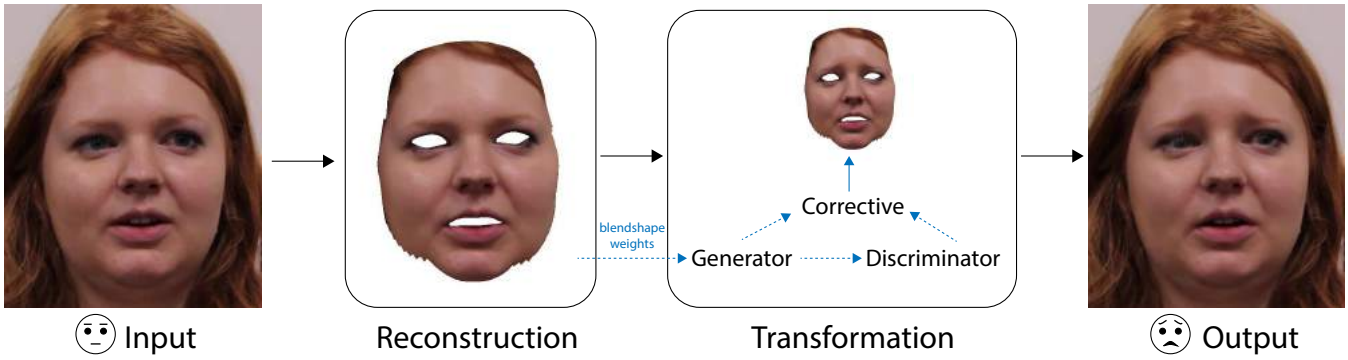

Figure 2: From an input face video clip, our system first reconstructs the head pose, identity, expression, and albedo map of its 3D face, as well as camera parameters and environment lighting. After that, the source expression in the face is transformed to the desired (target) expression through CycleGAN-based expression mapping in the blendshape space and followed by mouth corrective and smoothing. The re-rendered face with the target expression is finally blended back to the input video.

of the target person [LXW*12]. Expression transfer for actor-to-avatar [WBLP11, FJA*14] has also been extensively explored previously. In addition, prior works [CTFP05, GVS*15, SSKS17] produce photo-realistic speech animation in which lip motion is matched with input audio.

2.3. Expression Manipulation

Some previous works manipulate the facial expression or facial components in images or video. For example, Yang et al. transfer a local facial component (e.g., smiling mouth) from one image to another [YWS*11]. Some other methods aim at manipulating eye gaze in 2D video [KPB*12, GKSL16] or editing 3D facial animation crafted by artists at the sequence level [LD08, MLD09]. The work of [YBS*12] achieves exaggeration, attenuation, or replacement of facial expression in parts of 2D video. In this method, the resulting video is synthesized through image warping or frames reordering, and thus it cannot effectively handle illumination changes incurred by the change of expression. It also lacks the capability of creating novel facial expressions while our method can create novel target facial expressions that do not exist in the original source video. Malleson et al. continuously blend facial performance video of an actor, which may contain different facial expressions or emotional states [MBW*15] .

2.4. Image Transformation

We complete this section by reviewing some recent image transformation/style transfer methods developed in computer vision community. “pix2pix" [IZZE16] employs conditional adversarial networks to learn the mapping between image pairs. The works of [JAFF16, GEB16] combine the style and content from two images and synthesize a novel image using CNN. Taigman et al. transfer face images into emojis by training a domain transfer network on millions of face images [TPW16]. Our approach builds on the CycleGAN framework [ZPIE17] that learns a mapping function between two unpaired image domains using two GAN models [GPAM*14]. The works of [LZZ16, SLH*17, CCK*17] developed variants of GAN models for face attributes synthesis, such as gender/age modification and expression transformation. However, those imaged based GAN models are usually limited to generating low resolution images and likely to incur artifacts on face or background change. In addition, none of the above methods can handle photo-realistic and temporal consistent transformations for image sequences and thus they cannot be straightforwardly extended for video-based facial expression transformation.

3. 3D Facial Performance Reconstruction

In this section, we reconstruct 3D facial performance frame by frame from an input face video clip. We employ the real-time facial performance capture method in [WSXC16, SWTC14]. We summarize their method in the following and refer readers to their papers for details. For each frame, we first track a set of 2D facial landmark locations using the local binary feature (LBF) based regression [RCWS14] (Figure 3a). Then, we reconstruct the 3D facial deformation and head pose by minimizing the difference between the projected 3D facial features and the corresponding 2D landmark locations. Finally, we estimate the lighting conditions and facial texture using the first several frames and thus reconstruct the 3D facial performance.

We represent 3D facial models using multi-linear models [VBPP05, CWLZ13]. Specifically, we describe a 3D face using two low-dimensional vectors that control identity and expression, respectively:

$$M = R(C_r \times_2 u^T \times_3 v^T) + T, \quad (1)$$

where M represents the facial geometry of a subject, R and T represent the global rotation and translation of the subject respectively, C_r is the reduced core tensor, and u and v are respectively identity and expression parameters (also called the *identity vector* and the *expression vector* in the remaining writing).

Our multi-linear model is constructed based on the FaceWarehouse dataset [CWZ*14]. In our experiments, the dimensions of

the identity vector u and the expression vector v are set to 50 and 46, respectively. Specifically, we use a Delta blendshape formulation [LAR*14] to represent the face with the expression vector $\{v_i\}_{i=1}^{46}$, where $0 \leq v_i \leq 1$. Given u, its corresponding individual-specific blendshape face is obtained as $B = C_r \times_2 u^T = \{B_0, B_1, ..., B_{46}\}$ and $M = R(B_0 + \sum_{i=1}^{46} v_i(B_i - B_0)) + T$.

Then, using the algorithms in [WSXC16], we first reconstruct 3D head pose and facial deformation for each frame. An example of 3D face construction is shown in Figure 3b. After that, we estimate the lighting condition and texture of the face using the first several frames and assume constant for the rest of the video, following the work of [SWTC14]. In this process, the face surface is assumed Lambertian with albedo $\rho(x,y)$, represented as RGB values in texture space. The lighting of the Lambertian surface is modeled using spherical harmonics [BJ03]. We solve for the lighting coefficients (i.e., lighting coefficients of the harmonic expansion) and texture in an analysis-by-synthesis way such that the difference between the synthesized image and the input video frame is minimized [SWTC14]. To this end, we can synthesize a face image (Figure3c) based on "hypothesized" lighting coefficients $\mathbf{l}$, the albedo map $\rho(x,y)$, and a per-pixel normal estimate $\mathbf{n}(x,y)$:

$$I(x,y) = \mathbf{l}^T \rho(x,y) Y(\mathbf{n}(x,y)). \quad (2)$$

(a) Landmarks (b) Geometry (c) Texture

Figure 3: An example of 3D face reconstruction from an input frame: The detected 2D facial landmarks (a), the resulting face model without (b) and with (c) texture.

4. Cycle-Consistent Expression Mapping

We now describe how to transform the source expression in the input video to a specified (target) expression. Facial expressions can be quite complicated; directly mapping the facial appearance from one expression to another requires a significant amount of training examples to cover the varieties across illuminations, races, ages, genders, etc. Instead, with the above reconstructed facial deformations and texture, we propose to learn this expression mapping in the blendshape weights space $\{v_i\}_{i=1}^{46}$. Since our approach may need to take speech video as the input, theoretically we need to collect a large number of training blendshape pairs that are aligned at the phoneme level to learn an effective mapping between two expressions. Collecting such a training dataset is dauntingly time-consuming and error-prone due to phoneme alignments. Besides, the transformed expression needs to be continuous and smooth.

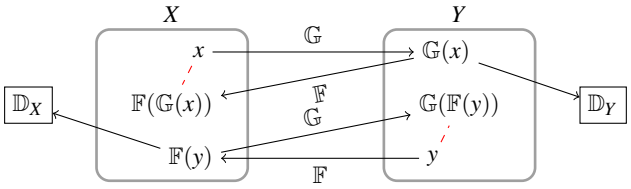

Figure 4: Two mapping functions $\mathbb{G}$ and $\mathbb{F}$ between two expressions X and Y are learned, where $\mathbb{G}: X \to Y$ and $\mathbb{F}: Y \to X$. $\mathbb{D}_X$ and $\mathbb{D}_Y$ are discriminators for X and Y, respectively. The loss of mapping cycle is denoted as red dash lines.

To address the above challenges, at the first step, we employ the Cycle-Consistent Generative Adversarial Network (CycleGAN) [ZPIE17] to learn the mapping between a source expression and a target expression, without phoneme-phoneme alignments. Compared to the original CycleGAN model that can only take an image as input, our model directly consumes blendshape weights, which are transparent to identity, pose, texture, and lighting conditions. Our model training can also converge faster and requires significantly fewer training data than the original CycleGAN model.

4.1. Blendshape Weights Mapping

We now describe how we employ the CycleGAN model to learn the expression mapping in the blendshape weights space. From the reconstructed blendshape weights of the used training video dataset [HJ10], we first sample training expression pairs (x_i, y_i) independently from a source domain and a target domain. Next, given samples in two expression domains X and Y (e.g., neutral and happiness), a mapping function $\mathbb{G}: X \to Y$ is learned, with the expectation that the transformed samples $\mathbb{G}(x)$ are as close as possible to real samples in domain Y.

Our method constructs a CycleGAN-based expression mapping model by learning a backward mapping function $\mathbb{F}: Y \to X$. As illustrated in Figure 4, a real sample x in domain X is transformed to $\mathbb{G}(x)$ in domain Y and then mapped back to $\mathbb{F}(\mathbb{G}(x))$ in domain X. Similarly, a cycle transformation of y is expressed as $\mathbb{G}(\mathbb{F}(y))$. To reduce the space of possible mapping functions, we enforce the result of cycle transformation to be as close as possible to the corresponding real samples, i.e., $\mathbb{F}(\mathbb{G}(x)) \approx x$ and $\mathbb{G}(\mathbb{F}(y)) \approx y$.

This cycle-consistent loss is measured as:

$$E_{cyc}(\mathbb{G}, \mathbb{F}) = \|\mathbb{F}(\mathbb{G}(x)) - x\|_1 + \|\mathbb{G}(\mathbb{F}(y)) - y\|_1. \quad (3)$$

Our model also includes two discriminators $\mathbb{D}_X$ and $\mathbb{D}_Y$ in order to distinguish between the transformed samples $\mathbb{G}(x), \mathbb{F}(y)$ and corresponding real samples y, x. Specifically, $\mathbb{D}_Y$ aims to differentiate the transformed sample $\mathbb{G}(x)$ from the real sample y and $\mathbb{D}_X$ aims to differentiate $\mathbb{F}(y)$ from x. The objective function can be expressed as:

$$\begin{aligned} E_{gan}(\mathbb{G}, \mathbb{D}_Y) &= E_{y \sim p_Y}[\log \mathbb{D}_Y(y)] + E_{x \sim p_X}[\log(1 - \mathbb{D}_Y(\mathbb{G}(x)))] \\ E_{gan}(\mathbb{F}, \mathbb{D}_X) &= E_{x \sim p_X}[\log \mathbb{D}_X(x)] + E_{y \sim p_Y}[\log(1 - \mathbb{D}_X(\mathbb{F}(y)))]. \end{aligned} \quad (4)$$

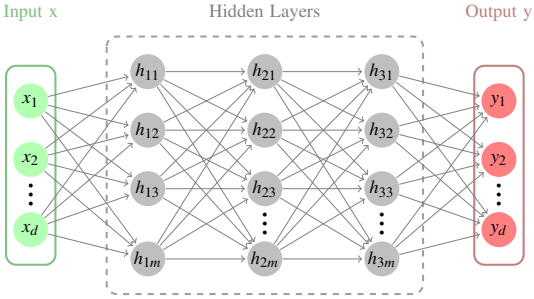

Figure 5: The generator $\mathbb{G}$ maps a source expression x to the target y through a deep neural network. The network consists of input and output layers with dimension $d = 46$, and 3 hidden layers with dimension $m = 100$.

The full energy (Eq. 5) is a summation of Eq. 3 and Eq. 4, with λ (set to 10 in our experiments) controlling the weight for the cycle-consistent loss:

$$E(\mathbb{G},\mathbb{F},\mathbb{D}_X,\mathbb{D}_Y) = E_{gan}(\mathbb{G},\mathbb{D}_Y) + E_{gan}(\mathbb{F},\mathbb{D}_X) + \lambda E_{cyc}(\mathbb{G},\mathbb{F}). \tag{5}$$

The learned generators $\mathbb{G}$ and $\mathbb{F}$ are used for the mapping between the source expression and the target expression. Given a target expression v^g from the generator, a 3D face can be created using Eq. 1.

4.2. Architecture

We developed two generators and two discriminators for the CycleGAN based expression mapping model as presented in Figure 4. A generator is a fully connected neural network (Figure 5) containing an input layer, 3 ReLU hidden layers, and a sigmoid output layer. Each hidden layer has 100 units while both the input and the output layers have 46 units corresponding to the expression vector. The sigmoid function in the output layer helps to regularize each element of the target expression vector to reside in the valid range $[0,1]$. A discriminator has a similar structure except that the output layer has only one unit producing a probability p. This probability p indicates the chance that the input comes from real data samples (i.e., 1-p chance is from a generator). For instance, 1.0 means the 100 percent chance that the input comes from real data samples, while 0.0 means the 100 percent chance that the input comes from a generator. Weights are initialized with a normal distribution $\mathcal{N}(0,0.01)$.

4.3. Training

We used the Surrey Audio-Visual Expressed Emotion (SAVEE) dataset [HJ10] for model training. The dataset contains video clips recorded from 4 male actors with multiple expressions, uttering 120 sentences in English. Each video clip was recorded at 60 FPS, resulting in 110K images in total. We ran our 3D performance reconstruction method described in §3 on each video clip, and manually removed those results with inaccurate appearance, leading to around 12K blendshape weights vectors for each expression category. We fixed neutral as the source domain and trained the above CycleGAN-based expression mapping models to map neutral to other expressions. Each model was trained for 200 epochs with the batch size of 1. The networks were implemented in Tensorflow [AAB*16] and took around 8 hours for training on a nVidia Geforce GTX 1060 GPU. For any expression mapping that does not involve the neutral expression, we concatenate two successive mappings using neutral as the relay node. As shown in Figure 6, an expression mapping *sadness* $\rightarrow$ *anger* is translated as *sadness* $\rightarrow$ *neutral* $\rightarrow$ *anger*. The model training is offline done only once.

Figure 6: A mapping from sadness to anger can be translated as two consecutive mappings: sadness $\rightarrow$ neutral and neutral $\rightarrow$ anger.

5. Expression Transformation

Window-based smoothing: Now we use the trained CycleGAN model for mapping blendshape weights v to the target expression v^*. Simply forward feeding v to the trained generator $\mathbb{G}$, however, cannot guarantee to obtain a dynamically coherent, lip-synced sequence. The reason is that our training data are blendshape weight vectors from unpaired images in two domains. Therefore, smooth transitions of an input blendshape deformation sequence from the source domain cannot guarantee smoothness when mapped to the target domain. Meanwhile, the unpaired sampling mechanism cannot preserve lip sync, since the randomly sampled images do not necessarily have the same utterance in training video clips. Figure 7 shows a failure case: the mouth in the source expression (Figure 7a) turns to be more opened after CycleGAN mapping (Figure 7b). To address the above issue, we propose to minimize a quadratic energy function to solve the target expression v^* per frame:

$$\begin{aligned} &\arg\min_{v^*} \left\| v^* - \frac{\sum_{j=-k}^{k} w_j v_{g,j}}{\sum_{j=-k}^{k} w_j} \right\|_2^2 + \alpha \left\| \bar{B} v^* - m \right\|_2^2 \\ &\text{s.t.} \quad 0 \le v_i^* \le 1, i = 1,...,46 \end{aligned} \tag{6}$$

In Eq. 6, the first term encourages the optimal v^* to be close to the mapped expression v_g after convolution operation by a window of size $2k+1$. j is the index of a frame within the window containing a small number of neighbors around the current frame, and w_j, and $v_{g,j}$ represent the weight and generated expression vector from the generator $\mathbb{G}$ by feeding the source expression v for the j^{th} frame, respectively. To balance the contribution of each $v_{g,j}$ within the window, we set the weight w_j to be the confidence value $\mathbb{D}(v_{g,j})$ by feeding $v_{g,j}$ to the discriminator $\mathbb{D}$. Recall that a higher value of $\mathbb{D}(y)$ indicates it is more likely y is sampled from real data rather than from a generator. The sliding window effectively helps to smooth out jitters in the synthesized target animation. We set $k = 1$ in this paper, hence the resulting sequence has 1 frame delay from the source video stream. The second term in Eq. 6 is the lip-correction term, detailed in the follow-up §5.1.

(a) Source:Neutral (b) Target:Happiness (c) Corrected Lip

Figure 7: We enforce the lip motion to match the source audio by minimizing the 3D distance of key lip vertices between before (green dots) and after (red dots) transformation.

5.1. Lip Correction

We also synchronize lip motion with the source audio. The basic idea is to constrain the lip region with the tracked lip motion while maximally preserving the target expression characteristics. Specifically, we integrate a lip correction term (i.e., the second term) into the energy function Eq. 6, which measures the 3D distance of the selected key lip vertices between the tracked and transformed face models. In Eq. 6, $\bar{B}$ is a matrix consisting of the corresponding rows of blendshape B for the key lip vertices, and m is a vector containing the 3D positions of the key lip vertices in the source expression. As shown in Figure 7, the key lip vertices after correction (Figure 7c) are moved toward their counterpart in Figure 7a. α is a weight to balance the two terms, and is set to 1000 in our experiments.

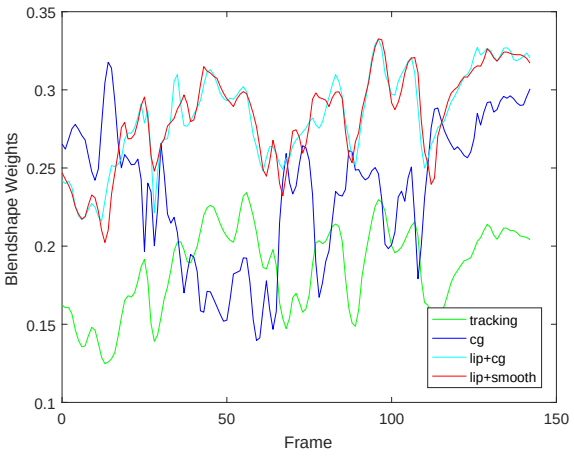

Figure 8: Weight curves of a specific blendshape basis (related to mouth movement) directly tracked from the source video (green, *tracking*), by the CycleGAN-based expression transformation alone (blue, *cg*), by the CycleGAN-based expression transformation + lip correction (cyan, *lip+cg*), and by the CycleGAN-based expression transformation + lip correction + smoothing (red, *lip+smooth*).

As shown in Figure 7, the key lip vertices (or called control points) are pre-defined as the four vertices bounding the upper and lower lips respectively. The main reason why we only choose the four key vertices in the middle of the lips, instead of selecting more vertices on the lips, is as follows: for certain target expressions (e.g., happiness), the activations of certain mouth-relevant blendshape bases are required. Therefore, if we impose more lip control points as a strong constraint for Eq. 6 to ensure the transformed lip shape being as close as possible to the original one, the resulting target facial expression may be less desired. In our experiments, we found that the selection of the four key vertices is a good trade-off to balance the overall expression realism and lip-sync. Figure 8 also illustrates the weight changes of a specific blendshape basis (primarily relevant to mouth movement) by different modules. As shown in this figure, a naive per-frame CycleGAN (cg) generates incoherent dynamics compared to the source animation. Per-frame CycleGAN with lip correction (lip+cg) creates synchronous but noisy movements, and finally our window-based smoothed CycleGAN with lip correction (lip+smooth) creates dynamically consistent and smooth mouth movements.

5.2. Composition

To synthesize a photo-realistic frame with the target expression, we finally re-render the face model with the target expression v^*, together with the head pose, camera parameters, lighting, and albedo texture estimated from the source frame. The texture mapping is pre-defined by artists. As mentioned previously, the mouth shape may be changed after transformation, so we need to accordingly warp the mouth interior of the source frame to fill in the mouth region of the re-rendered face. We pre-define a sparse set of vertices around the lip contour in 3D model space, and their triangulation in the projected 2D image space to cover the mouth region. The mouth region of the source frame is then warped to the new projected positions of the lip contour vertices. To the end, we put the re-rendered face, the warped mouth, and the untouched eyes region, and the background from the source frame onto different layers, and blend them together using the Laplacian pyramid blending algorithm [BA83].

6. Experimental Results

We captured live facial performance of several volunteers using a commodity Microsoft LifeCam HD 5000 running at 30Hz in 640×480 resolution. We also applied our method to some YouTube video clips at resolution 1280×720 to show the generality of our method. We show the transformed results in Figure 9 and in the accompanying demo video.

We implemented our system in C++ using Eigen [GJ*10] for linear algebra and OpenCV for image processing. We solve the nonlinear optimization problem in face tracking using the Trust Region strategy with box constraints for the expression parameters. We use block coordinate descent algorithm for a faster convergence: the pose and identity parameters usually converge in 5 iterations, and the expression parameters terminate in 10 iterations. Although the estimation of lighting and albedo texture takes 1.8s, it is only computed once for the first several frames and will not delay the real-time tracking thereafter. The CycleGAN inference is implemented in Tensorflow running on GPU. The constrained

Figure 9: Results of our system on selected Internet video clips. Both of the input expressions are neutral.

quadratic programming in transformation is solved by the interior point method. Similar to [TZS*16], we run compostion in fragment shaders on GPU with the hardware-generated mipmaps for building image pyramids. We show runtime statistics of our method in Table 1. Our system runs on a desktop computer with two Intel Xeon E5620 CPUs @2.4 GHz and nVidia Geforce GTX 1060 GPU.

Table 1: Runtime statistics for video clips with three different resolutions. From top to bottom: 640×480, 1280×720 and 1920×1080. The CPU and GPU computations run in parallel.

CPU		GPU		FPS
Tracking	Transform	CycleGAN	Composite	
24.89ms	7.62ms	1.79ms	1.52ms	30.76Hz
26.29ms	7.99ms	1.76ms	2.94ms	29.17Hz
29.21ms	7.46ms	1.75ms	5.31ms	27.27Hz

We qualitatively evaluated the effectiveness of our approach on real-time captured facial performance, on publicly available face datasets, and on Internet video clips. Figures 1, 9, 10, and 12 show some results by our approach to transform one expression to another, with various skin colors, ethnicities, and ages. For the video results by our approach, please refer to the supplemental demo video.

6.1. Comparisons

Since we cannot find any previous approaches that are specifically designed to achieve the *same* goal as ours, to the best of our knowledge, we compared our approach with some previous approaches that aim to achieve *similar* goals, including linear mapping, [YBS*12], the original CycleGAN [ZPIE17], and ground truth, described below.

Comparison with linear mapping + lip correction. We compared our approach with the straightforward linear mapping method. Given an input blendshape coefficients vector x from the source expression domain, the target blendshape vector y can be obtained from the following linear function:

$$y = x - \bar{X} + \bar{Y}, \tag{7}$$

where $\bar{X}$ and $\bar{Y}$ are the average blendshape coefficients vectors for the source and the target expressions, respectively. We computed $\bar{X}$ and $\bar{Y}$ from the same dataset that was used for CycleGAN training, and kept them fixed in our experiments. If the above Eq. 7 is directly used, obviously the resulting lip movements could be out-of-sync with audio. Therefore, we add the lip correction module that is proposed in this paper onto the naive linear mapping method. Figure 10b shows a result of linear mapping + lip correction.

Compared to our method (Figure 10f), the linear mapping + lip correction could produce acceptable visual results but with less computation/training time and less implementation effort. However, we observe that when computing the average delta blendshape vector $(\bar{Y} - \bar{X})$ from a large enough dataset, asymmetric expressions may neutralize each other, resulting in balanced delta values for those blendshape bases controlling the left half face and the right half face. This means that if the input expression is symmetric, the generated expression will also be symmetric. By contrast, our model provides more flexibility by learning the nonlinear characteristic of the expression mapping problem. Figure 11 shows such an example: given a neutral input with mouth corners slightly down, the linear mapping method generates flat or slightly up mouth corners, which is less consistent with the relatively strong cheek raise on the resulting face. This is because cheeks and mouth corners are controlled by separate blendshape bases which are uncorrelated in the linear mapping method. Our CycleGAN mapping overcomes this issue by learning the correlation between blendshape bases, and thus generates more holistically consistent target expressions.

Comparison with [YBS*12]. We compared our approach with [YBS*12] on the same sequence "Talking Face Video" [Coo17] that is used in the original work of [YBS*12]. Figure 10d and Figure 10f present the transformation to happiness from the same neutral input (Figure 10a) by [YBS*12] and our approach, respectively. Note that the method by [YBS*12] actually replaces the neutral sub-sequence by manually choosing a happy frame from the input source sequence as the center of gap and interpolates expression towards the two gap boundaries. Therefore, their method cannot transform an expression to a *novel* expression that does *not* exist in the input source sequence. By contrast, our method can transform the expression in the input source sequence to a novel expression for the whole sequence, not limited to the *existing* expressions in the input source sequence.

(a) Input (b) Linear Mapping + Lip Correction (c) CycleGAN [ZPIE17] (d) [YBS*12] (e) Our CycleGAN-based Expression Mapping (f) Result by our approach

Figure 10: Comparisons among (b) the linear mapping + lip correction method, (c) the original image-based CycleGAN model [ZPIE17], (d) [YBS*12], and (e)(f) our method on the same input (a). (b) was generated by a linear blendshape weights mapping, (c) was generated by the original CycleGAN model trained on the images from the same dataset and using the identical weight parameters of the loss function as our CycleGAN-based expression mapping model, and (d) was generated by its original authors' implementation. (e) and (f) were generated by our CycleGAN-based expression mapping model without and with lip correction and smoothing, respectively.

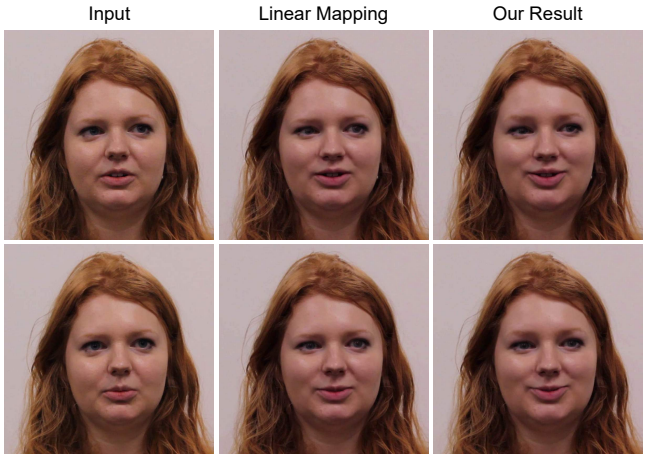

Figure 11: Visual comparisons between Linear Mapping + Lip Correction and our method.

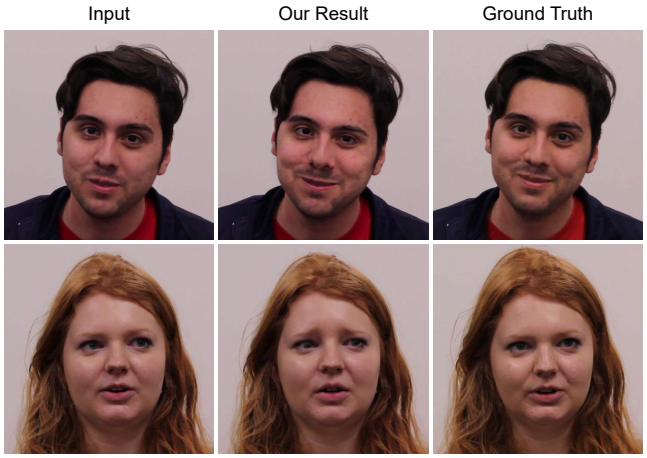

Figure 12: Comparison with ground truth. Both of the input expressions are neutral, and outputs are happiness and sadness from top to bottom. The ground truth frames are manually chosen where the subjects are uttering the same phonemes as our resulting frames.

Comparison with CycleGAN [ZPIE17]. We also compared our method with the original image-based CycleGAN model [ZPIE17]. In this comparison, we trained the latter using the images from the same SAVEE dataset and the same weight parameters for the loss function. The trained network contains three stride-2 convolutions and 9 residual blocks. From Figure 10c we can see that the original CycleGAN model trained on a relative small amount of images cannot generalize well to unseen images. The generated image is not photo-realistic and the face appears blurry, since (i) the identity of the input image is far from the ones in the training data, and (ii) the background color is also changed because the background color in the training data is plain black. In addition, the image-based CycleGAN model fails to generate temporal smooth facial animation, which can be clearly seen in the supplemental demo video. In contrast, Figure 10e shows that our CycleGAN-based expression mapping model works more effective on a small set of training data, generalizes well to various identities, and is more suitable for hallucinating face generation. Figure 10f shows the lip-corrected and smoothed result.

Comparison with [TZS*16] and [AECOKC17]. The works of Face2face [TZS*16] and [AECOKC17] are two state-of-art methods for expression transfer. However, both of them require a driving source video clip from which the expression is transfered to a target video clip. Recording driving video clips with different expressions and with per frame lip-sync to the audio of the target video is practically infeasible, so a fair comparison between our method and the two works cannot be performed. Nonetheless, it is noteworthy that our method can be complementary to them in order to synthesize emotional expressions with little effort in modification.

Comparison with ground truth. We captured volunteer subjects speaking the same sentences with two different expressions. After that, we applied our method to transform one expression to the other and then compared our transformed results with their corresponding ground truth performance. As shown in Figure 12, our method produced similar happiness and stronger sadness compared

to the recorded ground truth of the subjects. It is noteworthy that it is impossible for the subjects to make perfectly audio-aligned performances with two different expressions; therefore, we manually chose ground truth frames where the subjects are uttering the same phonemes as the comparison frames.

7. Limitations

Despite its demonstrated effectiveness, our current work has several limitations described below.

- Our current approach cannot handle extreme head poses in input video. As shown in Figure 13, the target happy expression in a side view produced a concave face contour compared to the source image.
- Our current method lacks the capability to generate personalized expressions. In reality, different people may have their own different ways to express the same emotion even when speaking the same sentence multiple times. Moreover, people exhibit various facial dynamic details such as creases and wrinkles around the eyes or on the forehead. The above individual-specific expression characteristics are ignored in the current work.
- Similar to many prior related works, our currrent method relies on an accurate facial landmark tracker. Inaccurate detected landmarks can lead to smoothing effect at the texture estimation step (§3) as different pixels from input images could be mapped to the same uv position in texture space.
- Our current system can produce a consistent sequence with the target expression. However, the transitions between different expressions are also important, which are not yet taken into consideration in our current method.
- Due to the forced synchronization of lip motion in real-time, our method cannot produce highly exaggerated expressions, such as surprise with mouth largely deformed compared to the source.

(a) Input (b) Mesh Overlay

Figure 13: Our method cannot handle extreme head poses in input video. (b) shows a failure case in which the face contour is changed in the new expression (red circle).

8. Conclusion

In this paper, we present a complete pipeline to photo-realistically transform the facial expression for monocular video in real-time. We train a CycleGAN model in blendshape weights space using less data and training time. We present a real-time smooth transformation algorithm to retain lip sync with the source audio. We tested our system on many face video clips with different identities and speeches and for different target expressions.

As the future work, we plan to improve our system to automatically identify the exhibited facial expression from the source video, instead of identified by users in the current work. Our current system generates target video with untouched audio and head motion. However, for certain target expressions such as anger, keeping the original audio and/or head motion from the input source video may seriously affect the perception of the transformed emotion, because the emotions conveyed by the facial expression, head motion, or audio channels could be substantially different or even conflict with each other. We plan to incorporate emotional transformation for speech (e.g., [GDOP17]) and head motion (e.g., [DSD17]) to the current framework in the future.

Acknowledgments

The authors would like to thank Dr. Fuhao Shi for his thoughtful discussion and numerous helps for this work. This work was in part supported by US National Science Foundation (NSF) IIS-1524782. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the agencies.

References

[AAB*16] ABADI M., AGARWAL A., BARHAM P., BREVDO E., CHEN Z., CITRO C., CORRADO G. S., DAVIS A., DEAN J., DEVIN M., ET AL.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016). 5

[AECOKC17] AVERBUCH-ELOR H., COHEN-OR D., KOPF J., COHEN M. F.: Bringing portraits to life. *ACM Trans. Graph. 36*, 6 (Nov. 2017), 196:1–196:13. 1, 8

[BA83] BURT P., ADELSON E.: The laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31*, 4 (1983), 532–540. 6

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating faces in images and video. *Computer Graphics Forum 22*, 3 (2003), 641–650. 2

[BJ03] BASRI R., JACOBS D. W.: Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 2 (2003), 218–233. 4

[Bra99] BRAND M.: Voice puppetry. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 21–28. 1

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 187–194. 2

[CBF16] CONG M., BHAT K. S., FEDKIW R.: Art-directed muscle simulation for high-end facial animation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), SCA '16, Eurographics Association, pp. 119–127. 2

[CBZB15] CAO C., BRADLEY D., ZHOU K., BEELER T.: Real-time high-fidelity facial performance capture. *ACM Trans. Graph. 34*, 4 (July 2015), 46:1–46:9. 2

[CCK*17] CHOI Y., CHOI M., KIM M., HA J.-W., KIM S., CHOO J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint arXiv:1711.09020* (2017). 3

[CHZ14] CAO C., HOU Q., ZHOU K.: Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph. 33*, 4 (July 2014), 43:1–43:10. 2

[Coo17] COOTES T.: Talking face video, 2017. URL: http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html. 7

[CTFP05] CAO Y., TIEN W. C., FALOUTSOS P., PIGHIN F.: Expressive speech-driven facial animation. *ACM Trans. Graph. 24*, 4 (Oct. 2005), 1283–1302. 3

[CWLZ13] CAO C., WENG Y., LIN S., ZHOU K.: 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July 2013), 41:1–41:10. 2, 3

[CWZ*14] CAO C., WENG Y., ZHOU S., TONG Y., ZHOU K.: Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2014), 413–425. 2, 3

[DCFN06] DENG Z., CHIANG P.-Y., FOX P., NEUMANN U.: Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games* (2006), I3D '06, ACM, pp. 43–48. 1

[DN08a] DENG Z., NEUMANN U.: Expressive speech animation synthesis with phoneme-level controls. *Computer Graphics Forum 27*, 8 (2008), 2096–2113. 1

[DN08b] DENG Z., NOH J.: Computer facial animation: A survey. In *Data-driven 3D facial animation*. Springer, 2008, pp. 1–28. 2

[DNL*06] DENG Z., NEUMANN U., LEWIS J. P., KIM T.-Y., BULUT M., NARAYANAN S.: Expressive facial animation synthesis by learning speech coarticulation and expression spaces. *IEEE transactions on visualization and computer graphics 12*, 6 (2006), 1523–1534. 1

[DSD17] DING Y., SHI L., DENG Z.: Perceptual enhancement of emotional mocap head motion: An experimental study. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)* (2017), pp. 242–247. 9

[DSJ*11] DALE K., SUNKAVALLI K., JOHNSON M. K., VLASIC D., MATUSIK W., PFISTER H.: Video face replacement. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 130:1–130:10. 2

[EF78] EKMAN P., FRIESEN W.: Facial action coding system: a technique for the measurement of facial movement. *Palo Alto: Consulting Psychologists* (1978). 2

[EGP02] EZZAT T., GEIGER G., POGGIO T.: Trainable videorealistic speech animation. *ACM Trans. Graph. 21*, 3 (July 2002), 388–398. 1

[FJA*14] FYFFE G., JONES A., ALEXANDER O., ICHIKARI R., DEBEVEC P.: Driving high-resolution facial scans with video performance capture. *ACM Trans. Graph. 34*, 1 (Dec. 2014), 8:1–8:14. 3

[GDOP17] GRINSTEIN E., DUONG N., OZEROV A., PEREZ P.: Audio style transfer. *arXiv preprint arXiv:1710.11385* (2017). 9

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 2414–2423. 3

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. http://eigen.tuxfamily.org, 2010. 6

[GKSL16] GANIN Y., KONONENKO D., SUNGATULLINA D., LEMPITSKY V.: Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *Computer Vision – ECCV 2016* (Cham, 2016), Springer International Publishing, pp. 311–326. 3

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems 27* (2014), Curran Associates, Inc., pp. 2672–2680. 3

[GVR*14] GARRIDO P., VALGAERTS L., REHMSEN O., THORMAEHLEN T., PEREZ P., THEOBALT C.: Automatic face reenactment. In *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition* (June 2014), pp. 4217–4224. 2

[GVS*15] GARRIDO P., VALGAERTS L., SARMADI H., STEINER I., VARANASI K., PÉREZ P., THEOBALT C.: Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. *Computer Graphics Forum 34*, 2 (2015), 193–204. 3

[GZC*16] GARRIDO P., ZOLLHÖFER M., CASAS D., VALGAERTS L., VARANASI K., PÉREZ P., THEOBALT C.: Reconstruction of personalized 3d face rigs from monocular video. *ACM Trans. Graph. 35*, 3 (May 2016), 28:1–28:15. 2

[HJ10] HAQ S., JACKSON P. J.: Multimodal emotion recognition. *Machine Audition: Principles, Algorithms and Systems*, 17 (2010), 398–423. 4, 5

[IKKP17] ICHIM A.-E., KADLEČEK P., KAVAN L., PAULY M.: Phace: Physics-based face modeling and animation. *ACM Trans. Graph. 36*, 4 (July 2017), 153:1–153:14. 2

[IKNDP16] ICHIM A.-E., KAVAN L., NIMIER-DAVID M., PAULY M.: Building and animating user-specific volumetric face rigs. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar Germany, Germany, 2016), SCA '16, Eurographics Association, pp. 107–117. 2

[IZZE16] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004* (2016). 2, 3

[JAFF16] JOHNSON J., ALAHI A., FEI-FEI L.: Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European Conference on Computer Vision* (2016), pp. 694–711. 2, 3

[KAL*17] KARRAS T., AILA T., LAINE S., HERVA A., LEHTINEN J.: Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Trans. Graph. 36*, 4 (July 2017), 94:1–94:12. 1

[KPB*12] KUSTER C., POPA T., BAZIN J.-C., GOTSMAN C., GROSS M.: Gaze correction for home video conferencing. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 174:1–174:6. 3

[LAR*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F. H., DENG Z.: Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports) 1*, 8 (2014). 2, 4

[LD08] LI Q., DENG Z.: Orthogonal-blendshape-based editing system for facial motion capture data. *IEEE Computer Graphics and Applications 28*, 6 (2008). 3

[LTW95] LEE Y., TERZOPOULOS D., WATERS K.: Realistic modeling for facial animation. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, ACM, pp. 55–62. 2

[LXW*12] LI K., XU F., WANG J., DAI Q., LIU Y.: A data-driven approach for facial expression synthesis in video. In *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), IEEE, pp. 57–64. 3

[LZZ16] LI M., ZUO W., ZHANG D.: Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586* (2016). 3

[MBW*15] MALLESON C., BAZIN J., WANG O., BRADLEY D., BEELER T., HILTON A., SORKINE-HORNUNG A.: Facedirector: Continuous control of facial performance in video. In *Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV)* (Dec 2015), IEEE, pp. 3979–3987. 3

[MLD09] MA X., LE B. H., DENG Z.: Style learning and transferring for facial animation editing. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), SCA '09, ACM, pp. 123–132. 3

[RCWS14] REN S., CAO X., WEI Y., SUN J.: Face alignment at 3000 fps via regressing local binary features. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (June 2014), IEEE, pp. 1685–1692. 3

[SLH*17] SONG L., LU Z., HE R., SUN Z., TAN T.: Geometry guided adversarial facial expression synthesis. *arXiv preprint arXiv:1712.03474* (2017). 3

[SLS*12] SEOL Y., LEWIS J., SEO J., CHOI B., ANJYO K., NOH J.: Spacetime expression cloning for blendshapes. *ACM Trans. Graph. 31*, 2 (Apr. 2012), 14:1–14:12. 1

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (July 2005), 417–425. 2

[SSKS17] SUWAJANAKORN S., SEITZ S. M., KEMELMACHER-SHLIZERMAN I.: Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph. 36*, 4 (July 2017), 95:1–95:13. 1, 3

[SWTC14] SHI F., WU H.-T., TONG X., CHAI J.: Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 222:1–222:13. 2, 3, 4

[TKY*17] TAYLOR S., KIM T., YUE Y., MAHLER M., KRAHE J., RODRIGUEZ A. G., HODGINS J., MATTHEWS I.: A deep learning approach for generalized speech animation. *ACM Trans. Graph. 36*, 4 (July 2017), 93:1–93:11. 1

[TPW16] TAIGMAN Y., POLYAK A., WOLF L.: Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200* (2016). 3

[TZN*15] THIES J., ZOLLHÖFER M., NIESSNER M., VALGAERTS L., STAMMINGER M., THEOBALT C.: Real-time expression transfer for facial reenactment. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 183:1–183:14. 1, 2

[TZS*16] THIES J., ZOLLHOFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), IEEE, pp. 2387–2395. 2, 7, 8

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3 (July 2005), 426–433. 1, 2, 3

[WBLP11] WEISE T., BOUAZIZ S., LI H., PAULY M.: Realtime performance-based facial animation. *ACM Trans. Graph. 30*, 4 (July 2011), 77:1–77:10. 3

[WSXC16] WANG C., SHI F., XIA S., CHAI J.: Realtime 3d eye gaze animation using a single rgb camera. *ACM Trans. Graph. 35*, 4 (July 2016), 118:1–118:14. 3, 4

[XCLT14] XU F., CHAI J., LIU Y., TONG X.: Controllable high-fidelity facial performance transfer. *ACM Trans. Graph. 33*, 4 (July 2014), 42:1–42:11. 2

[YBS*12] YANG F., BOURDEV L., SHECHTMAN E., WANG J., METAXAS D.: Facial expression editing in video using a temporally-smooth factorization. In *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), IEEE, pp. 861–868. 3, 7, 8

[YWS*11] YANG F., WANG J., SHECHTMAN E., BOURDEV L., METAXAS D.: Expression flow for 3d-aware face component transfer. *ACM Trans. Graph. 30*, 4 (July 2011), 60:1–60:10. 3

[ZPIE17] ZHU J., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 2242–2251. 3, 4, 7, 8