Data-Driven 3D Neck Modeling and Animation

Yilong Liu, Chengwei Zheng, Feng Xu, Xin Tong, and Baining Guo, *Fellow, IEEE.*

Abstract—In this paper, we present a data-driven approach for modeling and animation of 3D necks. Our method is based on a new neck animation model, that decomposes the neck animation into local deformation caused by larynx motion and global deformation driven by head poses, facial expressions, and speech. A skinning model is introduced for modeling local deformation and underlying larynx motions, while the global neck deformation caused by each factor is modeled by its corrective blendshape set, respectively. Based on this neck model, we introduce a regression method to drive the larynx motion and neck deformation from speech. Both the neck model and the speech regressor are learned from a dataset of 3D neck animation sequences captured from different identities. Our neck model significantly improves the realism of facial animation and allows users to easily create plausible neck animations from speech and facial expressions. We verify our neck model and demonstrate its advantages in 3D neck tracking and animation.

Index Terms—Neck modeling, neck animation, speech-driven animation.

1 Introduction

Modeling and reconstructing realistic 3D facial animation play an important role in many graphics applications, such as movie production, game design, and virtual reality. In the past years, a number of methods have been proposed for modeling and animating 3D faces [1]–[7] and variant organs, such as hair [8]–[10], eyeballs [11]–[14], eyelids [15], [16], teeth [17], and lips [18]. These methods significantly improve the realism of facial animation. However, there is still one important part that has not been well studied: the neck. Although the neck is not part of the head, its motion still gives significant visual signals for people to recognize a subject's expressions and emotions. For example, when people talk, the neck will deform according to expression, pose, and speech. Apart from that, when people are nervous or telling lies, they tend to swallow and it leads to strong neck deformation. Therefore, modeling and animating the neck become a crucial task for realistic 3D facial animation.

Anatomically, the underneath muscles, vocal tract, and larynx cause neck deformation. As a result, the neck deformation is correlated to head poses, facial expressions, larynx motions, and voice. The main focus of previous methods [19]–[22] that model neck animation is mainly on head poses and facial expressions. The neck deformation driven by larynx and speech is ignored.

In this paper, we propose a data-driven approach for modeling the shape and full deformation of 3D necks of variant identities. In our method the neck deformation is decomposed into local deformation, which is caused by larynx motions, and global deformation correlated with head poses, facial expressions, and speech voice. For local deformation, we introduce a novel skinning scheme to model larynx motions. For global deformation, we first utilize the blendshape bases of the FLAME model [22] for head poses and facial expressions, and then introduce a set of corrective blendshapes to model the neck deformation caused by speech. In creating the neck model, we capture 3D face and neck motion sequences of different subjects using a multi-view setup. The captured sequence of each identity consists of 3D facial and neck animation with different head poses, facial expressions, and speeches. To learn our neck model from the captured dataset, we develop an optimization algorithm. Based on the neck model, we also introduce a regression method for mapping speech audio to the neck larynx motions and the weights of speech blendshapes. As a result, users can directly create neck animations based on speech input. We train the regression model from the motion sequences and the associated speech audios in the captured dataset.

Our neck model can be considered as an enhanced FLAME model with a new larynx skinning model. Similar to FLAME, our neck model is user-independent and can be directly applied to different identities. Modeling neck deformation based on different facial animation factors separately aids our model to easily generate neck animations consistent with the facial expressions and speech of different users. We validate our neck model by 3D neck tracking and animation. As a result, it has been shown that our neck model not only improves the realism of character animation, but also enables users to easily create reasonable neck animations from speech and facial expressions. To the best of our knowledge, this is the first work that aims to achieve larynx motion modeling and speech-driven neck animation.

In summary, the main contributions of our work are as follows:

- A FLAME-based 3D neck model and a larynx skinning model that model 3D neck deformations caused by larynx motion and speech, as well as head poses and expressions
- A 3D neck and facial animation dataset, which will be released to the community for research use
- A novel optimization method for constructing the neck model from the data
- A regression model for generating 3D neck animation from the speech audio

- *Y. Liu, C. Zheng and F. Xu are with Tsinghua University. E-mail: liuyilong.thu@gmail.com, l1l11012@qq.com, xufeng2003@gmail.com. F. Xu is the corresponding author.*
- *X. Tong is with Microsoft Research Asia. Email: xtong@microsoft.com.*
- *B. Guo is with Microsoft Research Asia and Tsinghua University. Email: bainguo@microsoft.com.*

2 Related work

In this section, we discuss the previous work that is directly related to this paper. Please refer to [23] for a comprehensive survey on 3D face modeling and animation.

2.1 Facial performance capturing and reconstruction

There have been a set of methods proposed for capturing and reconstructing high-quality 3D facial performance of real subjects. Beeler et al. [1] developed a camera rig in an environment with controlled lighting to achieve fine-scale facial motion reconstruction. Huang et al. [2] use face markers in a pre-processing step to solve the temporal consistency in the reconstruction. Shape from Shading (SfS) techniques [24] are also utilized in reconstructing facial details from monocular RGB videos [25], [26], which largely simplify the setup for facial detail reconstruction. Garrido et al. [18] reconstruct accurate lip motions through lip tattoos and multi-view capture system. Since the neck region lacks texture, we follow the method in [18] and also use tattoos and multi-view capture system for capturing the high-quality 3D face and neck animation sequences for model regression.

2.2 Models for 3D facial animation

Data-driven face models build a generic representation of face shapes and expressions of different identities from a face dataset. Blanz and Vetter [27] propose a Morphable Model to represent the shape variations of different identities, while a Blendshape model [28] is used to model facial expressions. To represent face shape and expression together, multilinear models [29], [30] are developed with aligned face data, and are able to fit the input of arbitrary users with arbitrary expressions. Recently, Li et al. [22] present a FLAME model for modeling the full head and neck region with global blendshapes and corrective pose-dependent blendshapes. Physically-based models construct an anatomic structure of a human face and generate facial animations with physical simulation. Ichim et al. [31] propose a volumetric model to handle passive motions on faces. An extended version [32] explores more physics and can be used for both reconstruction and animation. A number of methods have been presented to construct personalized blendshapes of a specific user. Ichim et al. [33] use multi-view images and an expression sequence to reconstruct user's face rig. Garrido et al. [34] achieve a similar goal with a monocular sequence only. Recently, Hu et al. [7] use a single image to build the user's face rig with vivid face and hair. Our method is different from the aforementioned methods that focus on the face region, as it is designed for modeling neck animation.

2.3 Neck modeling and animation

There have been a few methods introduced for modeling the neck and the upper body. Lee et al. [19] propose a biomedical model to model the anatomical structure of the neck-head and a control model to generate neck motions. Later, they extend the biomedical model to the upper body and combine the model with FEM simulation of soft tissues for generating body animations [21]. Bender et al. [20] propose a physical model for neck deformation with more deformation details and fast simulation speed. These methods can generate physically correct neck animation, but the physically based models proposed in these methods are designed manually for specific subjects, making it difficult to be used in other identities. Apart from that, it is also unclear how to integrate these techniques with other data-driven facial animation methods for generating realistic and consistent face and neck animation. All these methods mainly focus on neck motions driven by head and shoulder motions and ignore the larynx motion and detailed neck deformation resulting from speech.

Recently, Li et al. [22] propose a data-driven FLAME model to represent full head and neck animation using a set of blendshapes. Since the modeling of face and neck animation uses the same set of blendshapes, their method can automatically generate consistent face and neck deformation for specific facial expressions and head poses. However, the FLAME model also ignores the neck deformation caused by larynx and vocal tract and thus cannot model the neck deformation resulting from speech and swallowing. We expand the FLAME model by including a novel skinning scheme for neck deformation caused by larynx motion and corrective blendshapes for detailed neck deformation caused by speech. We also develop a new optimization algorithm to construct our new model from the captured neck animation dataset and a speech animation model for generating the neck animations from speech.

2.4 Speech animation

A number of techniques have been developed for generating facial animations from speech. Liu et al. [35] combine both audio and video input for real-time 3D facial animation, in which a user-independent phoneme feature sequence is extracted from speech audio input via a deep-learning network and then used for finding the corresponding 3D lip motions from a pre-captured database. Taylor et al. [36] apply a deep learning technique to directly regress mouth motions from the audio signal. Karras et al. [37] further consider the correlation between speech and emotions to produce more vivid facial animations. Aside from 3D facial motion sequences, 2D videos are also been edited with speech signal in a data-driven manner [38]. In contrast with these methods that generate facial animation from speech input, our method develops the neck animation from speech. The results of our experiments show that there are strong correlations between the speech signal and the neck deformation, and that the potential combination of neck and mouth regions may generate more vivid speech-driven animations in the future.

3 Model formulation

In this section, we illustrate our proposed novel face model. The latest FLAME model [22] works well in representing identity, poses and expression variations of a human head. However, it does not deal with the rich deformation of the neck and larynx. Our model expands the FLAME model with the inclusion of speech-related correctives, which consist of a larynx model designed based on our anatomical observation and a speech-related blendshape model .

We create our model based on the FLAME topology. Since its original resolution is not high enough to reproduce the variation, we apply the butterfly subdivision [39] on the face and front neck area. The eyeballs and corresponding joints are eliminated as they are not related to the main focus of this paper. Therefore, our model has a topology of $N = 13232$ vertices, and $K = 2$ joints (neck and jaw).

3.1 Larynx model

In anatomy, it is known that the larynx can slide beneath the neck skin and push the skin to deform accordingly. Such a motion is quite different from commonly modeled facial deformation, where there are fixed correspondences between vertices and geometric features. There are two important observations about the larynx motion: firstly, the bump driven by the larynx is moving as a whole, and the shape of the bump is basically the same at different larynx positions. Secondly, the correspondence between the larynx and the skin is varying. Based on these observations, we decouple a neutral shape $\mathbf{T}$ into a larynx-removed virtual base shape $\mathbf{T}_0$, and its larynx shape represented as an offset map in the UV space. To make sure the 3D larynx shape is preserved when the weight map slides in the UV space, an isometric parameterization is required when building the larynx model. Specifically, we crop out the mesh in the front of the neck region and use SLIM [40] with isometric distortion optimization to generate the UV map for the larynx model. Since the deformation is caused by the underneath larynx shape, and should be independent to the surface normal of the skin, we apply a 1D-per-pixel weight map $\mathcal{W}_l$ and a constant 3D unit vector $\mathbf{t}_s$ to represent the larynx shape. In this way, the neutral shape $\mathbf{T}$ can be formulated as:

$$\mathbf{T} = \mathbf{T}_0 + \mathcal{W}_l \cdot \mathbf{t}_s, \tag{1}$$

where $\cdot$ calculates a 3D vector for each vertex i by $\mathcal{W}_l^i \mathbf{t}_s$, and $\mathcal{W}_l^i$ denotes the value of the corresponding pixel of vertex i on $\mathcal{W}_l$.

To enable the sliding motion, a larynx motion factor $\mathbf{t}_d$ is applied to the larynx weight map, causing all the nonzero values of $\mathcal{W}_l$ to translate simultaneously in the UV space. It means that the weight value on pixel $\mathbf{u}$ will move to pixel $\mathbf{u} + \mathbf{t}_d$. Thus, the overall delta shape of the larynx is maintained, and the corresponding surface vertices are changed due to the sliding, which fits our physical observation. The larynx motion vector $\mathbf{t}_d$ is represented as a 2D vector in the UV space, but naturally the larynx should not slide horizontally. Therefore, we enforce the x-coordinate of $\mathbf{t}_d$ to be zero, and only solve the y-coordinate, which will be denoted as a scalar τ, so the final sliding larynx weight map will be denoted as $\mathcal{W}_l(\tau)$. We also observe that the larynx sliding direction is not strictly parallel to the surface of $\mathbf{T}_0$, introducing some slight variations on the larynx shape. In most cases, the neck is slightly leans forward at the rest pose; therefore, the magnitude of the protrusion onto the skin caused by the larynx tends to be larger when it drops down, and smaller when it raises. To approximate this effect, we further apply a scalar α on $\mathbf{t}_s$. A more intuitive demonstration about the relationship of τ and α will be shown in Section 7.2. Putting together, a larynx-sliding mesh at rest pose is expressed as:

$$\mathbf{T} = \mathbf{T}_0 + \mathcal{W}_l(\tau) \cdot \alpha\, \mathbf{t}_s \tag{2}$$

With different τ and α, the larynx shape will deform accordingly, as is shown in Figure 1.

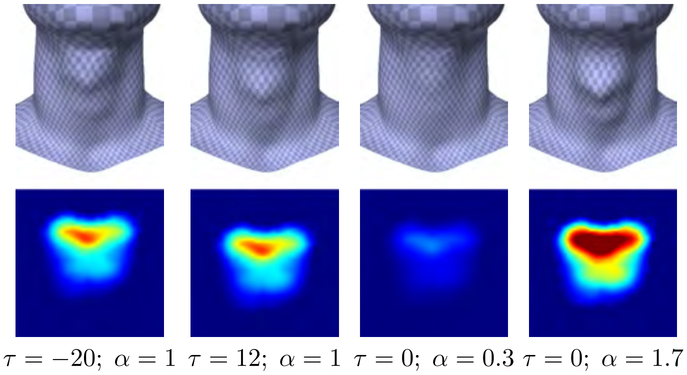
$\tau = -20;\ \alpha = 1$ $\tau = 12;\ \alpha = 1$ $\tau = 0;\ \alpha = 0.3$ $\tau = 0;\ \alpha = 1.7$

Fig. 1: The effects of larynx coefficients. The color-coded map in the second row is the larynx weight map $\mathcal{W}_l(\tau) \cdot \alpha$, synthesized with τ and α values from below.

3.2 Blendshape correctives

Pose and expression blendshapes The pose blendshapes $B_P(\boldsymbol{\theta}; \mathcal{P})$ and the expression blendshapes $B_E(\boldsymbol{\psi}; \mathcal{E})$ in our neck model are formulated with consistent notations and representations as in [22]. Notice that even though the formulations are the same with those in FLAME, we will fine-tune these blendshapes using our data to include more detailed motion in our model, especially on the neck region.

Speech blendshapes The expression blendshapes can model some neck deformation caused by different mouth motions, however, the neck may have additional deformation at different states of phonation because the vocal tract keeps configuring itself to pronounce different sounds. During speaking, the neck deformation may behave differently compared to non-speaking scenarios. Besides the mentioned larynx model, we also propose to train a set of speech-related blendshapes to model the additional variation of the neck. Like other blendshape correctives, the speech blendshapes are also orthogonal basis of displacements:

$$B_N(\boldsymbol{\phi}; \mathcal{N}) = \sum_{n=1}^{|\boldsymbol{\phi}|} \phi_n \mathbf{N}_n \tag{3}$$

where $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_{|\boldsymbol{\phi}|}]^\top$ denotes the speech blendshape coefficients, and $\mathcal{N} = [\mathbf{N}_1, \ldots, \mathbf{N}_{|\boldsymbol{\phi}|}] \in \mathbb{R}^{3N \times |\boldsymbol{\phi}|}$ denotes the speech blendshape basis.

3.3 Linear blend skinning

With all correctives and larynx shape added, the shape of any specific identity at rest pose can be expressed as:

$$\begin{aligned} \mathbf{M}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}, \tau, \alpha, \mathcal{W}_l) =& \mathbf{T}_0 + B_P(\boldsymbol{\theta}; \mathcal{P}) + B_E(\boldsymbol{\psi}; \mathcal{E}) \\ &+ B_N(\boldsymbol{\phi}; \mathcal{N}) + \mathcal{W}_l(\tau) \cdot \alpha\, \mathbf{t}_s \end{aligned} \tag{4}$$

in which user-specific base shape $\mathbf{T}_0$ is modeled using shape blendshapes the same way as FLAME. For convenience, we use $\mathbf{M}$ to indicate a general mesh, and $\mathbf{T}$ to indicate a mesh that could be modeled purely by the larynx model, but not the pose, expression and speech blendshapes. The following descriptions will follow this rule.

As we also use standard Linear Blend Skinning [41] function $W(\mathbf{M}, \mathbf{J}, \boldsymbol{\theta}, \mathcal{W})$ to rotate the rest pose mesh on the joints

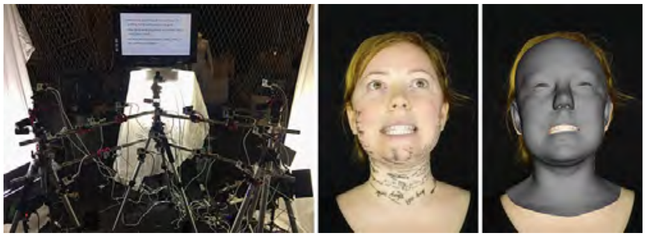

Fig. 2: Data capture. Left: The multi-view setup. Middle: The performer with tattoos attached. Right: A reconstructed mesh for the input frame.

$\mathbf{J}$ as FLAME does, we directly use the joint position $\mathbf{J}$ and skinning weight $\mathcal{W}$ of FLAME. The final model is as follows:

$$\mathbf{M}'(\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\phi},\tau,\alpha,\mathcal{W}_l) = W\left(\mathbf{M}(\boldsymbol{\theta},\boldsymbol{\psi},\boldsymbol{\phi},\tau,\alpha,\mathcal{W}_l),\mathbf{J},\boldsymbol{\theta},\mathcal{W}\right) \tag{5}$$

4 Data capture

Our model aims to represent more detailed motion on the neck region; therefore, we need high-fidelity 3D mesh sequences as training data. In this section, we will discuss how we acquire the training data.

4.1 Capture setup

We set up a multi-view capture system that consists of $C = 16$ GoPro cameras, each recording a 1080p video at 60 fps. To start and stop the recording, all cameras are connected to one remote controller through WiFi. Three diffuse lighting sources are carefully arranged to make sure the face and the neck are properly lit with least shadows. The cameras are calibrated at the beginning of each capture, and a clapperboard is used for manual synchronization in post-processing.

To ensure a good temporal correspondence, we attach some tattoo textures to the cheek and neck of each actor, which provide clues for visual tracking. After attaching the tattoos, we scan a high-quality template mesh of the actor's neutral expression at rest pose using a handheld Artec Eva scanner. The scan mesh will be deformed to fit the multi-view information to generate deformation at each frame.

During the recording, we ask the users to perform three kinds of motions: pure pose changes to train the pose blendshapes, pure expression changes to train the expression blendshapes and pure speech changes to train the larynx model and the speech blendshapes. An illustration of our data capture is shown in Figure 2.

4.2 Data reconstruction

After the recording procedure, the scan template mesh will be resampled to the same topology as in our model, with N vertices, and will be denoted as $\mathbf{M}^*$. After synchronization, we have image sequences from the C cameras: $\{\mathcal{I}_c^t\}_{c=0}^{C-1}$, and the calibrated intrinsic and extrinsic parameters for each camera: $\{\boldsymbol{\pi}_c, \mathbf{R}_c, \mathbf{t}_c\}_{c=0}^{C-1}$. We first reconstruct a 3D point cloud $\{\mathbf{p}_i^t\}$ for each frame t using PMVS [42] algorithm. The point cloud will be used to deform the template mesh $\mathbf{M}^*$.

For each frame t, we need to find the correspondence between the PMVS point cloud and the template mesh. For simplicity, we omit the superscript t in the following notations.

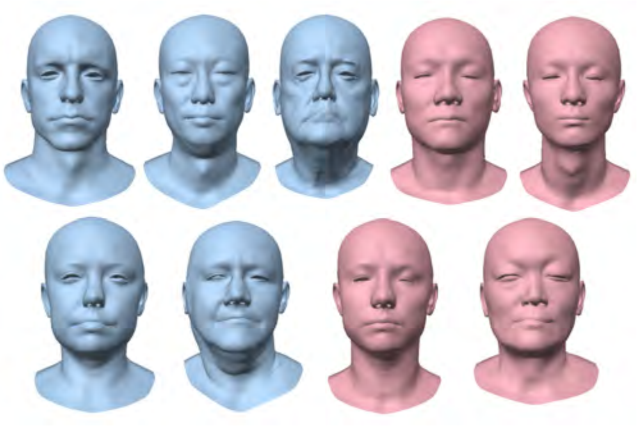

Fig. 3: Captured subjects. The blue ones are used for model training while the red ones are used for fitting tests.

Firstly the initial mesh $\tilde{\mathbf{M}}$ is rendered to each view with its texture, generating images $\{\tilde{\mathcal{I}}_c\}_{c=0}^{C-1}$. Dense optical flow $f_c : \mathcal{I}_c \mapsto \tilde{\mathcal{I}}_c$ is estimated at each view [43]. After that, each $\mathbf{p}_i$ will be projected to the view c_i whose view angle is the closest to the normal of the point, and then the corresponding pixel is decided using the optical flow of the view c_i:

$$\tilde{\mathbf{u}}_i = f_{c_i}(\boldsymbol{\pi}_{c_i}(\mathbf{R}_{c_i}\mathbf{p}_i + \mathbf{t}_{c_i})) \tag{6}$$

By inversely rendering $\tilde{\mathbf{u}}_i$, we can directly get the $\{\mathbf{p}_i\}$'s corresponding point $\mathbf{v}_i$ on the template mesh.

For each frame, we use the result of the previous frame as $\tilde{\mathbf{M}}$ to fit the current point cloud. The fitting result serves as the initial guess for the next iteration. For the first frame of a sequence, $\tilde{\mathbf{M}}$ is obtained by rigidly deforming $\mathbf{M}^*$ with some manually selected feature correspondences on faces. Given the dense correspondences obtained by the aforementioned methods, we deform the template mesh out of the FLAME space by the method introduced in [44].

The texture of the template mesh is initialized by the scan of the first frame. After the fitting of the first frame, the texture is updated from all the cameras and will be fixed for all the rest frames. Specifically, each pixel contributes to the color of its corresponding vertex, and the weight of the contribution is decided by the dot product of the viewing direction and the normal direction of the vertex.

As a result, our reconstructed data contains 9 subjects, each with three datasets, the pose, expression and speech datasets, which usually contain 3000, 3000 and 12,000 frames, respectively. For each frame, we have 16 images and a reconstructed mesh model. For the 9 captured subjects, we utilize the animations of 5 subjects for training, while animations of the remaining 4 subjects are used for testing. Figure 3 shows all the captured subjects.

5 Model training

In this section, we will train our model with our recorded data. Based on the motions of the actor, we classify all the recorded data into three datasets: pose dataset $\mathcal{D}_{pose}$, expression dataset $\mathcal{D}_{exp}$ and speech dataset $\mathcal{D}_{speech}$. For simplicity, in the following formulations, we assume the data is captured from one actor, unless stated otherwise. However, all the algorithms can be extended to multi-person data in a straightforward way, as we actually do in our experiments.

5.1 Refinement of pose and expression blendshapes

For $\mathcal{D}_{pose}$, we assume there is no deformation caused by expression and speech, and for $\mathcal{D}_{exp}$, we assume that the actor is performing silently without larynx motion. As a consequence, we can use $\mathcal{D}_{pose}$ and $\mathcal{D}_{exp}$ to train the pose and expression blendshapes with the methods introduced in [22]. We train the corrective pose blendshape with 5 subjects only and we found the result model has good generality. To train the corrective blendshape for expression, we randomly sample 10 identities from the FLAME identity space and generate their neck deformation sequences by following the deformation sequence of 5 subjects. Since details are lacking in the neck deformation generated by FLAME model, we then transfer the neck deformation details from the 5 subjects to the 10 FLAME identities via deformation transfer [45]. The pose and expression blendshapes generated in our method consist of both face and neck regions, and can be regarded as a refined version of FLAME blendshapes. In the experimental results, we will show the refined blendshapes outperform the original FLAME blendshapes in modeling the deformation, especially on the neck regions.

5.2 Speech model training

Once the pose and expression blendshape are refined, we train the speech related blendshapes and larynx model from the speech dataset $\mathcal{D}_{speech}$. Before the training, for each mesh model, we first compensate the deformation caused by pose and expression by estimating the pose and expression coefficients. So in this subsection, a 3D mesh $\mathbf{M}$ has no pose and expression-related deformation, and can be represented purely by our speech model as

$$\{\mathbf{M} \mid \mathbf{M} = \mathbf{T}_0 + B_N(\boldsymbol{\phi};\mathcal{N}) + \mathcal{W}_l(\tau)\cdot\alpha\,\mathbf{t}_s\}. \qquad (7)$$

Here, $\mathcal{N}$ is a set of corrective blendshapes for modeling neck deformation driven by speech, $\mathbf{T}_0, \mathcal{W}_l, \mathbf{t}_s$ construct the larynx model, and $\{\alpha_i\}, \{\tau_i\}, \{\boldsymbol{\phi}_i\}$ are model parameters for each 3D mesh.

We train the speech model by minimizing the difference between the captured mesh and the mesh reconstructed by our model, as shown in Algorithm 1. First, we manually separate the neck area into the larynx region $L(\mathbf{T})$ and the non-larynx region $N(\mathbf{T})$ as in Figure 4. After model initialization, we then optimize the model $\mathbf{T}_0, \mathcal{W}_l, \mathbf{t}_s, \mathcal{N}$ and estimate the coefficients $\{\alpha_i\}, \{\tau_i\}, \{\boldsymbol{\phi}_i\}$ for each frame in an iterative way. To this end, we further divide all unknowns into four groups: the larynx model $\mathbf{T}_0, \mathcal{W}_l, \mathbf{t}_s$, the larynx model coefficients: τ, α; the speech blendshape basis $\mathcal{N}$, and the blendshape coefficients $\boldsymbol{\phi}$. In each optimization step, we update each group of unknowns with the other three fixed. The algorithm is stopped after a predefined number of iterations. In the subsequent parts of this subsection, we first describe each step of our training algorithm and then discuss the implementation details of the algorithm.

Initialization In the initialization stage, we calculate the initial $\mathbf{T}_0, \mathcal{W}_l, \mathbf{t}_s$ from a mesh $\hat{\mathbf{T}}$ recorded with neutral pose, neutral expression and no speech. Since $\tau = 0$ and $\alpha = 1$ in this situation, the formulation for $\hat{\mathbf{T}}$ is reduced to $\hat{\mathbf{T}} = \mathbf{T}_0 + \mathcal{W}_l \cdot \mathbf{t}_s$. The best guess of the base mesh $\mathbf{T}_0$ is to remove the bump at the larynx region. Using the predefined region masks $L(\mathbf{T})$ and $N(\mathbf{T})$, we initialize $\mathbf{T}_0$ by reconstructing the larynx region using the boundary condition and zero Laplacian constraints:

$$\mathbf{T}_0 = \arg\min_{\mathbf{T}_0} \|B\big(L(\mathbf{T}_0)\big) - B\big(L(\hat{\mathbf{T}})\big)\|^2 + \lambda\|\mathbf{L}\cdot L(\mathbf{T}_0)\|^2 \qquad (8)$$

$$\text{s.t.} \quad N(\mathbf{T}_0) = N(\hat{\mathbf{T}})$$

where $B(\cdot)$ selects the one-ring boundary vertices of a given mesh region and $\mathbf{L}$ is the Laplacian operator. An illustration of the reconstructed base mesh is shown in Figure 5. And the direction of $\mathbf{t}_s$ is initialized with the average direction of the offsets between $\hat{\mathbf{T}}$ and $\mathbf{T}_0$:

$$\mathbf{t}_s = \sum_{i\in L} \frac{\hat{\mathbf{T}}^i - \mathbf{T}_0^i}{\|\hat{\mathbf{T}}^i - \mathbf{T}_0^i\|} \qquad (9)$$

where $\hat{\mathbf{T}}^i$ is the i-th vertex on the larynx region of mesh $\hat{\mathbf{T}}$. To ensure the actual offset is along the direction of $\mathbf{t}_s$, we shoot each vertex on $L(\hat{\mathbf{T}})$ along $-\mathbf{t}_s$ direction, and the intersection point on $\mathbf{T}_0$ becomes the new position of the corresponding vertex of $\mathbf{T}_0$. And by definition, the initial larynx weight map can be calculated as

$$\mathcal{W}_l^i = \|\hat{\mathbf{T}}^i - \mathbf{T}_0^i\|. \qquad (10)$$

The speech blendshapes are initialized to zero in our method, and so are the corresponding coefficients.

```
Algorithm 1: Algorithm for speech model training
Data: Speech-related mesh at each frame $\{\mathbf{M}_j\}$ and
      neutral mesh $\hat{\mathbf{T}}$
Result: Model parameters $\mathbf{T}_0, \mathcal{W}_l, \mathbf{t}_s, \mathcal{N}$; Per-frame
        coefficients $\{\alpha_j\}, \{\tau_j\}, \{\boldsymbol{\phi}_j\}$
begin
    Initialize $\mathbf{T}_0$ (Equation 8);
    Initialize $\mathbf{t}_s$ (Equation 9);
    Initialize $\mathcal{W}_l$ (Equation 10);
    Initialize $|\mathcal{N}| = 2$, and update blendshapes $\mathcal{N}$;
    Calculate blendshape coefficients $\{\boldsymbol{\phi}_i\}$;
    while $|\mathcal{N}| <$ threshold and max iteration number
     not reached do
        Update larynx model $\mathbf{T}_0, \mathcal{W}_l, \mathbf{t}_s$;
        Update larynx coefficients $\{\tau_i\}, \{\alpha_i\}$;
        $|\mathcal{N}| += 2$, and update blendshapes $\mathcal{N}$;
        Update blendshape coefficients $\{\boldsymbol{\phi}_i\}$;
    end
end
```

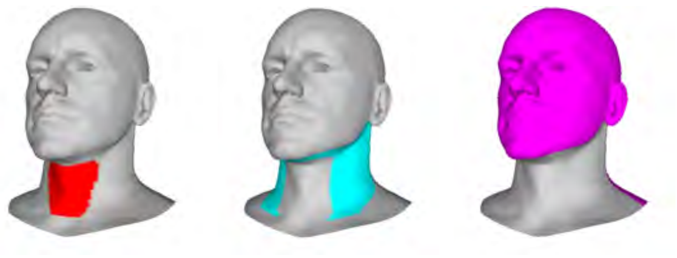
(a) Larynx region (b) Non-larynx region (c) Face region

Fig. 4: The masks used in our algorithm.

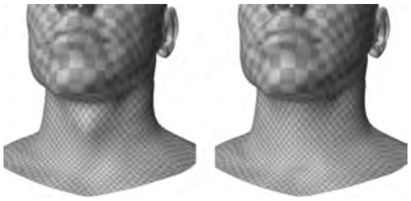

Fig. 5: The neutral mesh and its corresponding base mesh calculated by Equation 8.

Larynx coefficient estimation With known $B_N(\boldsymbol{\phi};\mathcal{N})$, we use $\mathbf{T_j}$ to denote a mesh purely represented by the larynx shape, and it can be represented by $\mathbf{M}_j$ as

$$\{\mathbf{T_j} \mid \mathbf{T_j} = \mathbf{M_j} - B_N(\boldsymbol{\phi_j};\mathcal{N})\}. \tag{11}$$

The larynx translation for each frame τ_j is estimated by finding the translation parameter τ that maximizes the correlation between the larynx offset that is extracted from the captured 3D mesh by $\mathbf{T}_j - \mathbf{T}_0$ and the translated weight map $\mathcal{W}_l(\tau) \cdot \mathbf{t}_s$:

$$\tau_j = \arg\max_{\tau} \mathrm{NCC}\big(\mathbf{T}_j - \mathbf{T}_0,\ \mathcal{W}_l(\tau) \cdot \mathbf{t}_s\big), \tag{12}$$

where NCC measures the Normalized Cross-Correction (NCC) between two offset maps. We use the NCC because it is robust to the shape difference of two offset maps caused by the unoptimized larynx model in the current iteration. To find the best τ, we slide the initial offset map vertically with each possible τ value and find the one that generates the maximal NCC value.

After τ_j is determined, the scalar α_j for each frame is calculated by solving the following least-squares optimization:

$$\alpha_j = \arg\min_{\alpha} \mathrm{Dist}\big(\mathbf{T}_j - \mathbf{T}_0,\ \mathcal{W}_l(\tau_j) \cdot \alpha\, \mathbf{t}_s\big), \tag{13}$$

where $\mathrm{Dist}(\mathcal{W}_1, \mathcal{W}_2)$ refers to the L^2 distance of two offset maps that are computed by the sum of L^2 distance of displacements on each offset map pixel.

Blendshape update In this step, all the recorded meshes $\mathbf{M}_j$ are used to re-estimate the speech blendshape. We first calculate all the residual of blendshape correctives by $\mathbf{M}_j - (\mathbf{T}_0 + \mathcal{W}_l(\tau_j) \cdot \alpha_j\, \mathbf{t}_s)$. The residual vectors of each frame j are stacked together to form a residual matrix, and Singular Value Decomposition (SVD) is performed to get an orthogonal representation. Given the pre-assigned blendshape dimensions $|\mathcal{N}|$, the left $|\mathcal{N}|$ columns of the left orthogonal matrix in SVD (which is commonly noted as the U matrix) are extracted as the new speech blendshape $\mathcal{N}$. The capacity of $\mathcal{N}$ is set to 2 in the initialization stage, and is increased by 2 at every iteration. In this way, the high-frequency component of larynx motion is avoided from being captured by the blendshapes.

Larynx model update First we use Equation 11 again to get all $\mathbf{T}_j$. Then we update the base mesh $\mathbf{T}_0$ and the weight map $\mathcal{W}_l$ together, given the current estimation of $\mathbf{t}_s$, τ_j and α_j:

$$\begin{aligned}\{\mathbf{T}_0, \mathcal{W}_l\} = &\arg\min_{\mathbf{T}_0,\mathcal{W}_l}\ E_{data} + \lambda_1 E_{Lap} + \lambda_2 E_{reg} + \lambda_3 E_{bdry} \\ &\text{s.t.} \quad N(\mathbf{T}_0) = N(\mathbf{T})\end{aligned} \tag{14}$$

The data term is

$$E_{data} = \sum_j \|\mathbf{T}_0 + \mathcal{W}_l(\tau_j) \cdot \alpha_j\, \mathbf{t}_s - \mathbf{T}_j\|^2. \tag{15}$$

The Laplacian constraints are

$$E_{Lap} = \|\mathbf{L} \cdot L(\mathbf{T}_0)\|^2. \tag{16}$$

The regularization constraints are defined as

$$E_{reg} = \lambda_T \|\mathbf{T}_0 - \mathbf{T}_0'\|^2 + \lambda_W \|\mathcal{W}_l - \mathcal{W}_l'\|^2 \tag{17}$$

where $\mathbf{T}_0'$ and $\mathcal{W}_l'$ are the respective results of the previous iteration. And the boundary constraints are

$$E_{bdry} = \|B\big(L(\mathbf{T}_0)\big) - B\big(L(\mathbf{T})\big)\|^2. \tag{18}$$

Then $\mathbf{t}_s$ can be estimated in a least square manner:

$$\mathbf{t}_s = \arg\min_{\mathbf{t}_s} \sum_j \|\mathbf{M}_j - \mathbf{T}_0 - B_N(\boldsymbol{\phi}_j;\mathcal{N}) - \mathcal{W}_l(\tau_j) \cdot \alpha_j \mathbf{t}_s\|. \tag{19}$$

Blendshape coefficient estimation Again, we calculate the blendshape correctives by $\mathbf{M}_j - (\mathbf{T}_0 + \mathcal{W}_l(\tau_j) \cdot \alpha_j\, \mathbf{t}_s)$, and perform blendshape fitting to calculate $\boldsymbol{\phi}_j$ for each recorded mesh:

$$\boldsymbol{\phi}_j = \arg\min_{\boldsymbol{\phi}_j} \|\mathbf{M}_j - \mathbf{T}_0 - \mathcal{W}_l(\tau_j) \cdot \alpha_j\, \mathbf{t}_s - B_N(\boldsymbol{\phi}_j;\mathcal{N})\|. \tag{20}$$

Details The speech blendshapes and the larynx correctives will both deform the base mesh $\mathbf{T}_0$. Therefore the blendshapes and the larynx model should be optimized together. However, this is not easy to achieve. In our iterative method, we first impose the speech blendshape to be very low dimensions, set to 2 for the first iteration. After each iteration, the dimension of $\mathcal{N}$ will increase by 2. Setting a maximum iteration number controls the dimension of the speech blendshapes. In our experiments we enforce 10 iterations and therefore get 20 basis for the speech blendshape.

Aside from building our neck model, the methods in this subsection are also used in fitting input data with the trained neck model. For fitting, we only need to iteratively perform the *larynx coefficient estimation* step and the *blendshape coefficient estimation* step for a few times. Since all the optimizations are linear with a small number of unknowns, our model is suitable for real-time fitting tasks.

6 Speech model regression

Although the neck deformation and the larynx motion are modeled using our technique, it is not intuitive for animators to directly control the coefficients to generate animation. In this section, we propose a regression method to predict the coefficients of the speech-related model from the speech audio input.

6.1 Feature consideration

In our model, the coefficients to be predicted can be grouped into two types: the speech blendshape coefficients $\boldsymbol{\phi}$ which represent the global neck deformation caused by the motion of the vocal tract, and the larynx coefficients τ and α, which represent the local larynx motion driven by the throat and vocal folds. Therefore, the blendshape coefficients should be related to the content of the speech, while the larynx position is considered to be highly related to the pitch and volume of the phonation. This decoupling fits the idea of the source-filter model, which is widely-used in speech synthesis and speech analysis, and recently shows its promising applications in speech-driven animations [37]. In the source-filter model, the speech wave is modeled as an excitation signal from vocal folds passed from linear filters of vocal tracts. By applying Linear Predictive Coding (LPC) analysis, the source signal can be separated from the filter response (the formants).

Similar to [37], we use the standard LPC formulation for audio feature extraction. For the audio preprocessing and configuration of audio frame length, we also follow the settings of [37]. Details about calculating the audio feature can be found in the result section. To sum up, for each audio frame, we have filter features $\mathbf{a} \in \mathbb{R}^{32}$ and source signal features $\mathbf{b} = [b_0, b_1, b_2]^\top \in \mathbb{R}^3$. Takeing the temporal information into account, we use a temporal window of 64 audio frames to form a feature vector that corresponds to a visual frame.

6.2 Regression

The regressors we are expecting are formulated as the blendshape coefficient regressor

$$R_b : \bar{\mathbf{a}} \mapsto \boldsymbol{\phi} \tag{21}$$

where $\bar{\mathbf{a}} = [\mathbf{a}_{t-32}^\top, \dots, \mathbf{a}_t^\top, \dots, \mathbf{a}_{t+31}^\top]^\top \in \mathbb{R}^{2048}$ is the filter feature vector, and the larynx coefficient regressors

$$R_\tau : \bar{\mathbf{b}} \mapsto \tau \tag{22}$$

and

$$R_\alpha : \bar{\mathbf{b}} \mapsto \alpha \tag{23}$$

where $\bar{\mathbf{b}} = [\mathbf{b}_{t-32}^\top, \dots, \mathbf{b}_t^\top, \dots, \mathbf{b}_{t+31}^\top]^\top \in \mathbb{R}^{192}$ is the source feature vector.

Since the dimensionality of the result is quite low, we use linear regression to model the mapping.

$$R_b(\bar{\mathbf{a}}) = \bar{\mathbf{a}}^\top \cdot \mathbf{R}_b \tag{24}$$

and the regression matrix $\mathbf{R}_b$ is estimated by least square method:

$$\mathbf{R}_b = \arg\min_{\mathbf{R}} \sum_i \|\bar{\mathbf{a}}_i^\top \cdot \mathbf{R} - \boldsymbol{\phi}_i\|^2 + \lambda \|\mathbf{R}\|_F^2 \tag{25}$$

The methods of estimating R_τ and R_α are similar, except that a bias is also estimated to balance the mean larynx position and scale.

7 Experimental results

In this section, we first formulate some experimental details. Then we present the comparison to FLAME, the current state-of-the-art parametric face model, to demonstrate the benefit in generating more realistic motion in neck regions. Following that, we add some experiments related to the model parameters. Finally, we present the power of our speech-related regressors, which generate reasonable neck motions from speech input. The experiments show that the regressors are not sensitive to users, indicating the generality of our techniques in the application of speech-driven facial animation. More results of our technique can be found in the accompanying video.

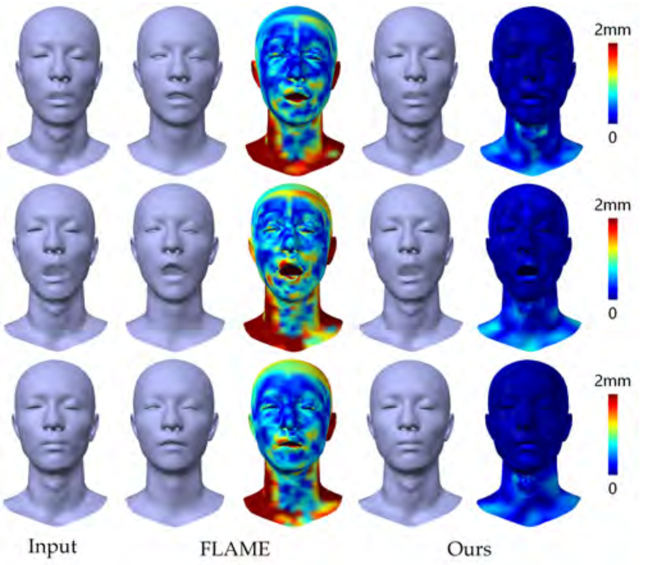

Fig. 6: Comparison with FLAME on a new user

Audio feature We use *filter features* for the blendshape coefficient regression, and *signal features* for the larynx regression. For the LPC formants, we use a 32-order all-pole filter. After estimating the filter responses, the inverse filtering is performed to reveal the signal. We use Prony's Method to calculate the filter coefficients $\mathbf{a}$ and the gain b_0 of the signal. Furthermore, the pitch and the voiced status of the signal are inferred using auto-correlation methods. The estimation of pitch b_1 is constrained to 80 Hz to 350 Hz which is a widely-adopted range of the fundamental frequency of human voice, and is normalized to the range of $[0, 1]$. To disambiguate the unvoiced frames and the 80 Hz pitch voiced frames, the voiced flag b_2 is used, which is set to 1 if it is a voiced frame and 0 otherwise.

Numbers The original FLAME model has 18 pose blendshapes and 100 expression blendshapes. To be comparable, we set the dimensions of our pose and expression blendshapes to be the same with those of FLAME. Our speech blendshapes have a dimension of 20. Our larynx model only has 2 parameters but performs well in modeling the larynx motions. The training process takes about 3 hours. However, this only needs to be performed once to build the model. The fitting of facial landmarks and 3D dense input can be performed in a real-time manner because only linear equations with a limited number of unknowns are needed to be solved. The speech-driven regression is also very fast due to the linearity and the dimensionality.

7.1 Comparison to FLAME

We compare our model to the original FLAME model by fitting a 3D speech sequence of a new user, which shows that our model performs better in reproducing the rich variations in neck regions. Figure 6 presents some keyframes of the experiment. It is worth mentioning that the fitting errors in the facial part also decrease due to the refinement of pose and expression blendshapes (Section 5.1), which affect the whole mesh. More comparisons are shown in our video. The error curves are shown in Figure 7.

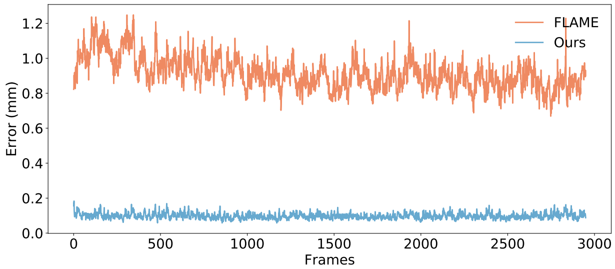

Fig. 7: Fitting errors of our full model and the original FLAME model. (measured by average 3D errors of neck region vertices)

TABLE 1: Error drops of each component validation tests on all 4 subjects (unit: mm), in which test #1 is the subject demonstrated in the paper.

Test ID	Pose Fitting	Exp Fitting	Speech Fitting
#1	0.23	1.57	0.26
#2	0.46	1.25	0.25
#3	0.32	1.42	0.28
#4	0.39	1.21	0.19

7.2 Validation of model components and generality

To demonstrate that each component of our model contributes to the lower fitting error, we calculate the error drop at each fitting stage, which is shown as Row #1 in Table 1. The fitting error is computed as the average 3D distance differences in the neck region, the same way as in the previous subsection. The values in Table 1 are error drops, calculated by the error of FLAME minus that of our model. Additionally, we repeat the experiments on all the other three test subjects. The error drops of each component on each subject are shown in Table 1. We can see that similar results can be seen on different test cases, which indicates that our model generalizes well.

Speech blendshape coefficients In Figure 8, we show a plot of time varying values of speech coefficients in a speaking sequence. For clarity reasons, we only show the first three dimensions. Rich variations exist in the coefficient values. In the highlighted regions, the speaker is between sentences and the expressions are nearly neutral. The coefficient values tend to be near zero. However, small variations still exist because of the influence of the context. This is often known as the coarticulation effect.

Relationship between τ and α We introduce the factor α to compensate the scaling variation occurred when the larynx sliding direction in 3D is not parallel to the neck skin surface. To demonstrate this, we plot the values of τ and α of each frame from a fitting sequence, as shown in Figure. 9. In the first part of the sequence (frame $0-150$), the actor is performing the swallowing motion at rest pose. We can see that when τ drops, which means the larynx is raising to a higher position, α tends to decrease, scaling down the overall shape of the bump, and vice versa. Such a relationship agrees with our observations in real cases, because naturally the neck is slightly leaning forward at rest pose, and when the larynx goes up, it is a little farther away from the skin. However, this is not necessarily the case in non-rest poses. This can be seen in the latter half of the plot (after frame 150), when the actor is rotating his head drastically. In such cases, the larynx shape may become more obvious even when it is raised, for example when the head is being thrown back, stretching the anterior neck skin and pushing the trachea forward.

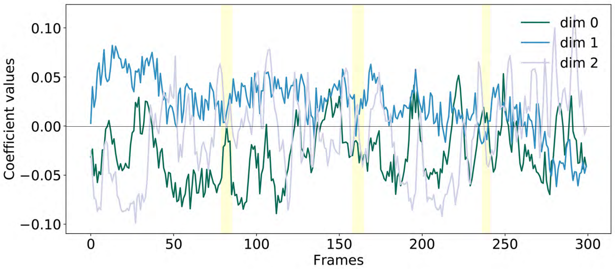

Fig. 8: Values for the first 3 dimensions of speech blendshape coefficients in a fitting sequence. The highlighted regions mark the near-neutral expressions when the speaker rests between sentences.

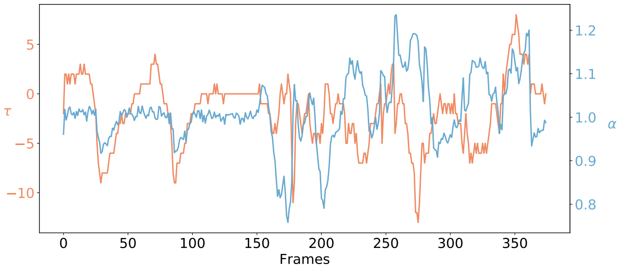

Fig. 9: Values for τ and α in a fitting sequence. The unit of τ is pixel distance in the UV space, and smaller τ corresponds to vertically higher layrnx position.

7.3 Speech-driven animation

The speech-driven regressor enables the generation of model coefficients from the extracted audio features. We built two types of regressors: user-specific regressors, applied to the training user only, and a generic regressor, which provides speech driven animation for new users.

User-specific speech animation We trained three user-specific regressors. The training and testing configurations are listed in Table 2. To synthesize the result meshes, pose and expression coefficients are generated by fitting the input. The coefficients of the speech model, including the speech-related blendshape coefficients and the larynx coefficients, are predicted by the regressors with the corresponding audio signal. Figure 11 shows some keyframes of the regression results of one user. We can see that the deformations in these keyframes are consistent with those of the input.

Generic speech animation We trained two generic regressors for male and female respectively. Each with the data of three users. Since the larynx shape is not obvious for most females, only the regressor for blendshape coefficients is trained for the female, as shown in Figure 10. Although the regression model does not make a big difference visually, the quantitative errors are still reduced significantly. Figure 12

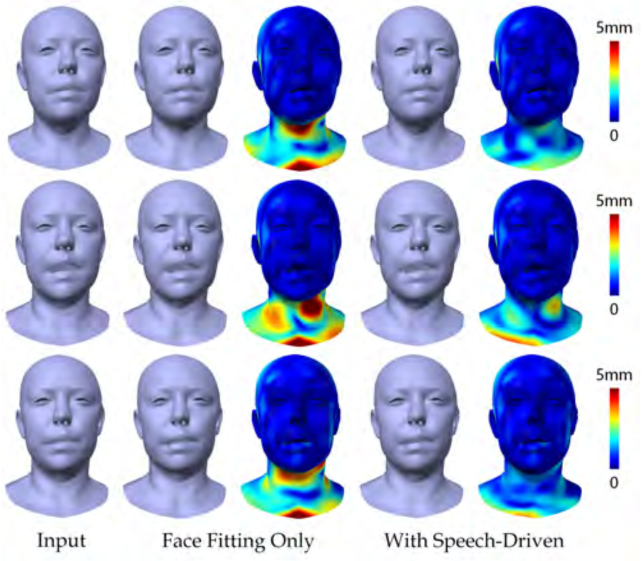

Fig. 10: Results of speech-driven animation of a female generic model

shows the prediction of the motion of a new user using the male generic model. Obviously, the generic model on a new user does not work as well as the specific model on this user, but the former still generates motion consistent with the input. So, in general, our technique is not sensitive to users and can generate plausible results.

7.4 Applications

The proposed model decouples the deformation of the face and the neck into several sets of user-independent coefficients, enabling the application of speech-driven and expression-driven animation. Also, it has the potential to achieve user-friendly neck motion editing, as our larynx model is carefully designed following the anatomical mechanism, and thus could be easily understood and used by animators.

Neck animation generation There are many existing ways to generate facial animations without expensive data capture, for example the image-based tracking and blendshape keyframing. However, usually, there are only few texture features on the neck for image tracking, and there are no well-defined semantics for neck blendshapes. With our model and regressor, it is easier to track and animate neck animation based on facial animation. Figure 13 shows the examples of the face-driven neck animation.

Swallowing modeling Swallowing is one of the most important types of neck and larynx motion. Although swallowing is not related to speech, our model can regenerate the motion given the correct coefficients. Figure 14 shows our results of modeling swallowing, in which the coefficients are generated from fitting the 3D reconstruction data. Our model compresses the data needed to reproduce the swallowing motion, and also introduces easy ways to manipulate.

Neck motion editing Since our model parameterizes the neck shape and motion using decoupled components and coefficients, motion editing becomes as easy as simply adjusting the coefficients. The bottom row of Figure 14 shows a further editing of the neck motion which exaggerates the larynx shape and motion range.

8 Limitations

Our model can generate plausible neck animations for different users. However, since the number of subjects we currently captured is quite limited, it may not be representative enough for a large and diverse population. This can be improved by capturing more subjects. In our regression method, we are modeling the user-independent neck motion. But there may be some subtle yet special motion characteristics among different users. Such user-specific characteristics are not represented in the model. The training data of our method is obtained by nonrigid motion tracking, which is unable to reconstruct high-frequency motion details, leading to the lack of such details in our results. In our model, we simplify the variation on the shape of the larynx and assume the overall shape is maintained up to a global scalar α during sliding. This reduces computational complexity while ensuring good result quality. It might be possible to build a more accurate model with more informative data like dynamic MRI. We do not model secondary motion. It would be interesting to do so by incorporating physically based simulation.

9 Conclusion

This paper proposed a novel dynamic neck model for 3D neck reconstruction and animation. The model combined local skinning and global blendshape representations to achieve user-independent neck motion modeling. The paper also introduced a dataset with 3D head motion sequences of different identities, poses, expressions and speech, and a comprehensive technique to build the dynamic model from the dataset. We will make the dataset available online for research use. Considering the pronunciation mechanism, the paper trained two regressors with different audio features to produce plausible neck animations from speech data. We believe this work has made a step further to achieve full head modeling and animation.

References

[1] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, "High-quality passive facial performance capture using anchor frames," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 75. 1, 2

[2] H. Huang, J. Chai, X. Tong, and H.-T. Wu, "Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 74. 1, 2

[3] F. Xu, J. Chai, Y. Liu, and X. Tong, "Controllable high-fidelity facial performance transfer," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 42, 2014. 1

[4] M. Sela, E. Richardson, and R. Kimmel, "Unrestricted facial geometry reconstruction using image-to-image translation," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 1585–1594. 1

[5] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz," in *Proceedings of Computer Vision and Pattern Recognition (CVPR 2018)*, 2018. 1

[6] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou, "Real-time facial animation with image-based dynamic avatars," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 126, 2016. 1

TABLE 2: Configurations of regressor training.

Regressor	Training Frames	Test Frames	Speech Blendshape Regressor Test Error[1]	τ Regressor Test Error[2]	α Regressor Test Error[2]
User #1	2630	320	0.1188	2.5766	0.1066
User #2	1880	500	0.1245	3.7709	0.1227
User #3	2720	640	0.1034	2.5341	0.0615
Male	8050	6j40	0.1131	1.3792	0.0737
Female	6214	1925	0.1776	-	-

[1] Measured by the L2 norm of the prediction and the ground truth coefficients.
[2] Measured by the squared difference between the prediction and the ground truth τ or α value.

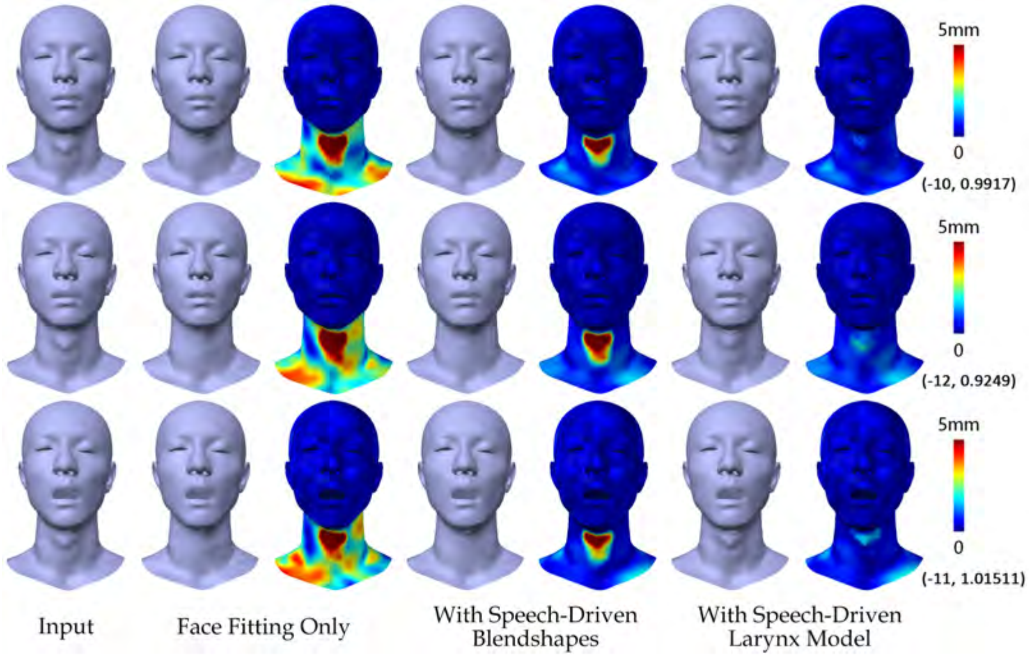

Fig. 11: Results of speech-driven animation a user specific model. (τ, α) values for each row are listed on the right.

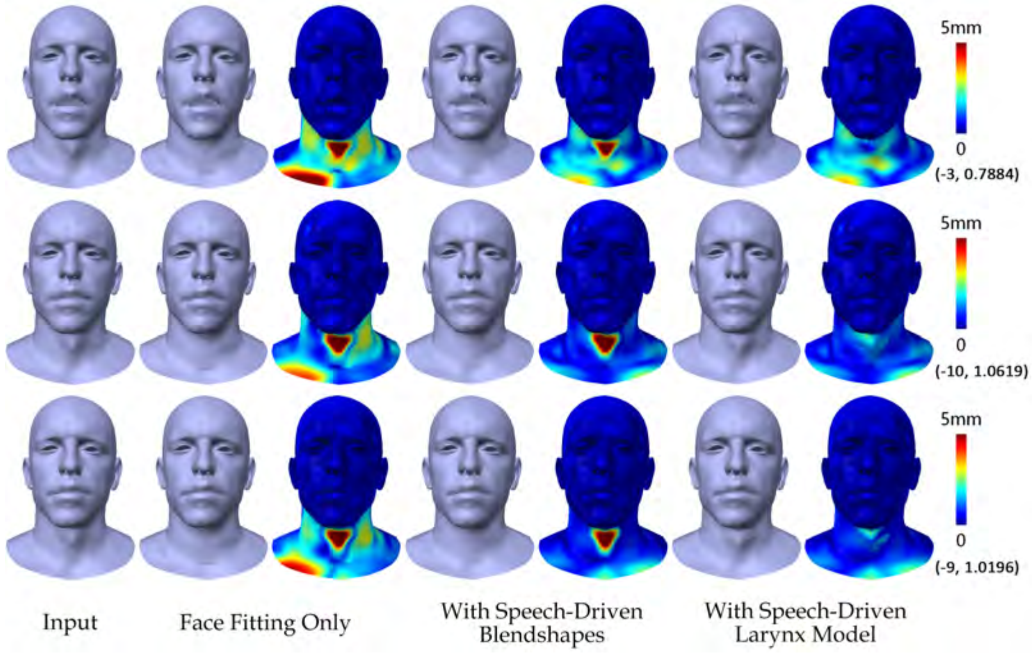

Fig. 12: Results of speech-driven animation of a male generic model. (τ, α) values for each row are listed on the right.

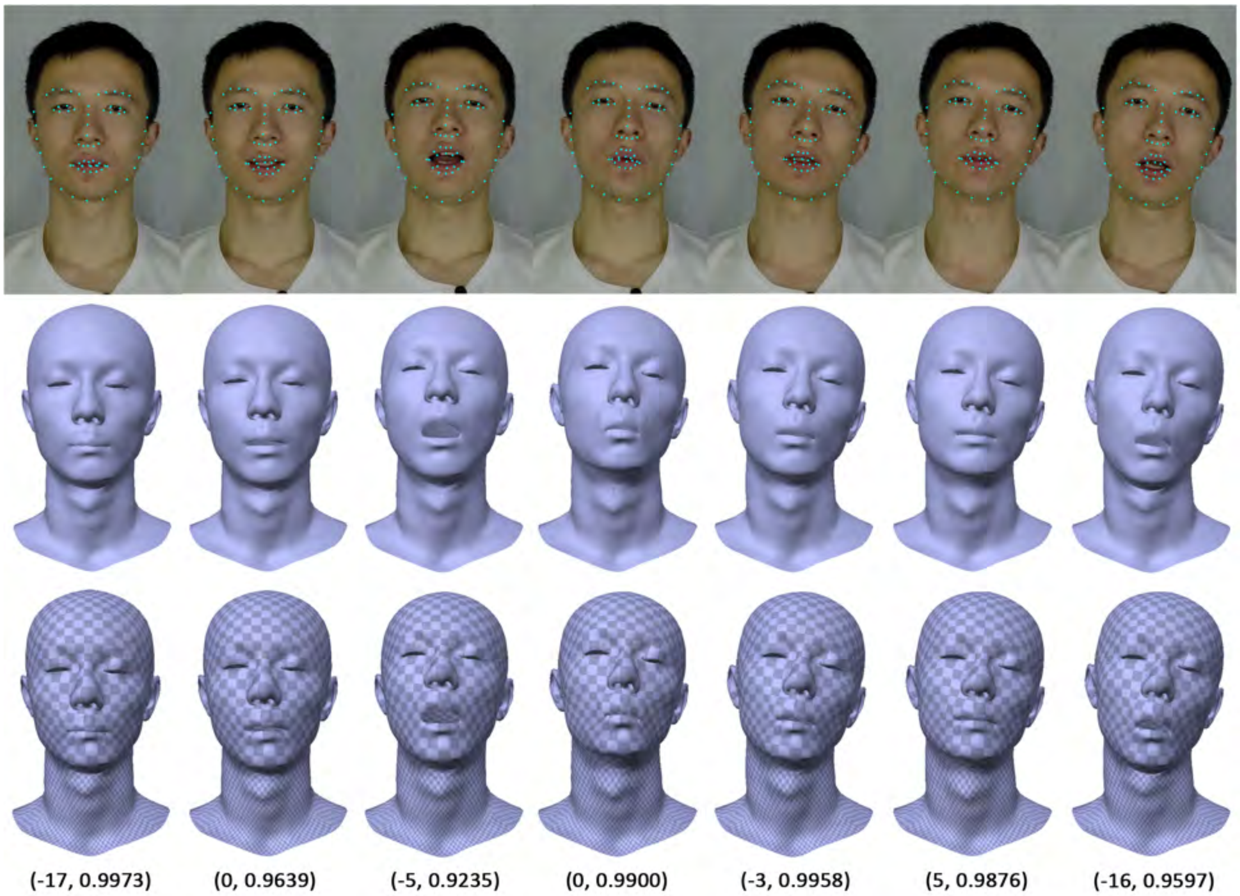

Fig. 13: Results of face-driven animation. 1st row: input video with facial landmarks; 2nd and 3rd rows: obtained face and neck animation. (τ, α) values for each column are listed on the bottom.

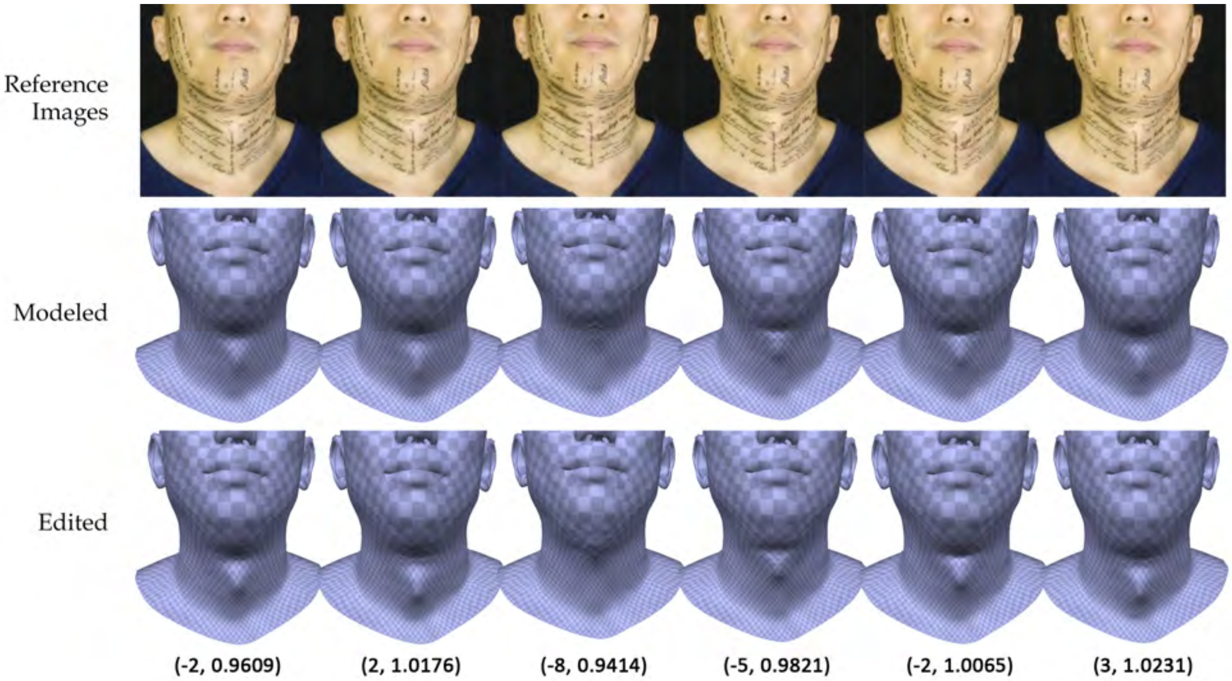

Fig. 14: Modeling and editing of swallowing motion. (τ, α) values for each column's modeling results are listed on the bottom.

[7] L. Hu, S. Saito, L. Wei, K. Nagano, J. Seo, J. Fursund, I. Sadeghi, C. Sun, Y.-C. Chen, and H. Li, "Avatar digitization from a single image for real-time rendering," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 195, 2017. 1, 2

[8] M. Chai, L. Wang, Y. Weng, X. Jin, and K. Zhou, "Dynamic hair manipulation in images and videos," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 75, 2013. 1

[9] M. Chai, C. Zheng, and K. Zhou, "A reduced model for interactive hairs," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 124, 2014. 1

[10] L. Hu, C. Ma, L. Luo, and H. Li, "Single-view hair modeling using a hairstyle database," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 125, 2015. 1

[11] P. Bérard, D. Bradley, M. Nitti, T. Beeler, and M. H. Gross, "High-quality capture of eyes." *ACM Trans. Graph.*, vol. 33, no. 6, pp. 223–1, 2014. 1

[12] C. Wang, F. Shi, S. Xia, and J. Chai, "Realtime 3d eye gaze animation using a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 118, 2016. 1

[13] P. Bérard, D. Bradley, M. Gross, and T. Beeler, "Lightweight eye capture using a parametric model," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 117, 2016. 1

[14] Q. Wen, F. Xu, and J.-H. Yong, "Real-time 3d eye performance reconstruction for rgbd cameras," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 12, pp. 2586–2598, 2017. 1

[15] A. Bermano, T. Beeler, Y. Kozlov, D. Bradley, B. Bickel, and M. Gross, "Detailed spatio-temporal reconstruction of eyelids," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 44, 2015. 1

[16] Q. Wen, F. Xu, M. Lu, and J.-H. Yong, "Real-time 3d eyelids tracking from semantic edges," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 193, 2017. 1

[17] C. Wu, D. Bradley, P. Garrido, M. Zollhöfer, C. Theobalt, M. Gross, and T. Beeler, "Model-based teeth reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 220, 2016. 1

[18] P. Garrido, M. Zollhöfer, C. Wu, D. Bradley, P. Pérez, T. Beeler, and C. Theobalt, "Corrective 3d reconstruction of lips from monocular video." *ACM Trans. Graph.*, vol. 35, no. 6, pp. 219–1, 2016. 1, 2

[19] S.-H. Lee and D. Terzopoulos, "Heads up!: biomechanical modeling and neuromuscular control of the neck," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 1188–1198, 2006. 1, 2

[20] J. Bender, J. Dequidt, C. Duriez, and G. Zachmann, "Physically-based human neck simulation," 2013. 1, 2

[21] S.-H. Lee, E. Sifakis, and D. Terzopoulos, "Comprehensive biomechanical modeling and simulation of the upper body," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 4, p. 99, 2009. 1, 2

[22] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4d scans," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 194, 2017. 1, 2, 3, 5

[23] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3d face reconstruction, tracking, and applications," *Computer Graphics Forum (Eurographics State of the Art Reports 2018)*, vol. 37, no. 2, 2018. 2

[24] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 8, pp. 690–706, 1999. 2

[25] F. Shi, H.-T. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 222, 2014. 2

[26] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video." *ACM Trans. Graph.*, vol. 32, no. 6, pp. 158–1, 2013. 2

[27] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. 2

[28] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. H. Pighin, and Z. Deng, "Practice and theory of blendshape facial models." *Eurographics (State of the Art Reports)*, vol. 1, no. 8, 2014. 2

[29] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 426–433, 2005. 2

[30] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3d facial expression database for visual computing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, 2014. 2

[31] A. E. Ichim, L. Kavan, M. Nimier-David, and M. Pauly, "Building and animating user-specific volumetric face rigs." in *Symposium on Computer Animation*, 2016, pp. 107–117. 2

[32] A.-E. Ichim, P. Kadleček, L. Kavan, and M. Pauly, "Phace: physics-based face modeling and animation," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 153, 2017. 2

[33] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3d avatar creation from hand-held video input," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 4, p. 45, 2015. 2

[34] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt, "Reconstruction of personalized 3d face rigs from monocular video," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 3, p. 28, 2016. 2

[35] Y. Liu, F. Xu, J. Chai, X. Tong, L. Wang, and Q. Huo, "Video-audio driven real-time facial animation," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 182, 2015. 2

[36] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews, "A deep learning approach for generalized speech animation," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 93, 2017. 2

[37] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 94, 2017. 2, 7

[38] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 95, 2017. 2

[39] N. Dyn, D. Levine, and J. A. Gregory, "A butterfly subdivision scheme for surface interpolation with tension control," *ACM transactions on Graphics (TOG)*, vol. 9, no. 2, pp. 160–169, 1990. 2

[40] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung, "Scalable locally injective mappings," *ACM Trans. Graph.*, vol. 36, no. 4, Apr. 2017. [Online]. Available: http://doi.acm.org/10.1145/3072959.2983621 3

[41] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 72, 2007. 3

[42] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010. 4

[43] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 25–36. 4

[44] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5. ACM, 2009, p. 175. 4

[45] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1015706.1015736 5