Speech-driven Face Reenactment for a Video Sequence

Yuta Nakashima†, Takaaki Yasui†, Leon Nguyen†,‡, Noboru Babaguchi (member)†

Abstract We present a system for reenacting a person’s face driven by speech. Given a video sequence with the corresponding audio track of a person giving a speech and another audio track containing different speech from the same person, we reconstruct a 3D mesh of the face in each frame of the video sequence to match the speech in the second audio track. Audio features are extracted from such two audio tracks. Assuming that the appearance of the mouth is highly correlated to these speech features, we extract the mouth region of the face’s 3D mesh from the video sequence when speech features from the second audio track are close to those of the video’s audio track. While retaining temporal consistency, these extracted mouth regions then replace the original mouth regions in the video sequence, synthesizing a reenactment video where the person seemingly gives the speech from the second audio track. Our system, coined S2TH (speech to talking head), does not require any special hardware to capture the 3D geometry of faces but uses the state-of-the-art method for facial geometry regression. We visually and subjectively demonstrate reenactment quality.

Key words: Face reenactment, talking head, speech-driven, 3D geometry

1. Introduction

Talking head synthesis, or face reenactment, is a technique to synthesize a person’s face with the facial expression and mouth changing over time. Various methods have been proposed for different types of input for controlling the facial expression and mouth, e.g., an audio signal containing speech1). For example, the state-of-the-art 3D face reconstruction technique has enabled facial expression transfer from a face in one video to that in another video3). Thanks to the recent development of deep neural networks (DNNs), or more specifically, generative adversarial networks (GANs), face reenactment from a single image of a speaker has been proposed to synthesize the mouth region as if it were uttering the sounds from given speech3). In this paper we refer to techniques that reenact faces from a speech as speech-driven face reenactment (e.g. video rewrite5).

Speech-driven face reenactment offers a variety of potential applications, ranging from entertainment to a remedy for people who have difficulties in facial motor functions. For instance, it may be used for foreign language dubbing in order to make the mouth movement in a video agree with the new audio track. It might also be used for fraud, making someone believe that a certain person is actually speaking.

In this paper, supposing that there exists a sufficient amount of video recordings for a certain speaker to observe the commonly used mouth movements and idiosyncrasies in delivering speech, we present a speech-driven face reenactment system, coined S2TH (speech to talking head), that synthesizes a talking head given a new speech signal, as illustrated in Fig. 1. Given a new target audio signal containing the speaker’s speech,
Our S2TH system retrieves the source video frames that agree with the utterance in the target audio signal. We then replace only the mouth region of the speaker in the source video frames with those from the corresponding retrieved video frames, giving the impression that the speaker is uttering the new speech signal.

Our main contribution is the use of the state-of-the-art DNN-based 3D depth map regressor\(^5\) and the face 3D morphable model (3DMM)\(^2\) to represent the face in each video frame, which facilitates mouth region retrieval and replacement. This approach drastically relaxes the hardware requirements, e.g., the usage of a stereo camera, and eliminates the manual annotation of facial images.

2. Related work

Recently, face reenactment has been widely studied. To the best of our knowledge, the video rewrite system by Bregler et al.\(^7\) is the first to implement automated speech-driven face reenactment in a data-driven approach, although it only handles 2D transformations. Schödl et al.\(^7\) presents video texture, which can reenact the face given a short video sequence. This approach does not provide control over the facial expression since it basically repeats frames from the input video sequence. Pighin et al.’s method synthesizes facial expressions of a person by capturing her/his face with certain facial expressions (happy, anger, etc.) with multiple cameras\(^6\). After manually labeling some landmarks, a 3DMM is fitted to the landmarks to regain the 3D geometry. Arbitary facial expressions can be obtained by morphing different 3D geometries.

Face2Face\(^2\) is an epoch-making work, which can fit a 3DMM to a sequence of 2D RGB video frames in real-time. The 3D geometry and lighting are simultaneously optimized in the fitting process. The obtained face representations can be used to control the facial expression of different speakers from other video sequences. The same group further extends it to make synthesized faces as well as the body more photo-realistic\(^9\). In this work, we borrow the 3DMM used in their method. Garrido et al.\(^10\) also propose a method to transfer mouth shape of a dubber to a target actor in the video so that it matches mouth motion of the target actor to the different audio track. This approach is applicable to dubbing on different languages and different speakers, but it requires dubber’s footage and 3D face mesh while our S2TH system just requires a dubbed audio track.

Some of early works on speech-driven face reenactment use hidden Markov models (HMMs)\(^11\)\(^12\)\(^13\). Typically a HMM is trained to synthesis mouth movements with audio/visual training database, and on inference stage, the trained HMM is sampled to predict visual trajectory. Wang et al.\(^14\) apply PCA to lip images to get visual feature vector for training a HMM, and then they search the lip image closest to the predicted visual trajectory from lip sample database so as to guarantee a photo-real output sequence. Anderson et al.\(^15\) use an extended HMM and adapted active appearance model representing face shape and appearance. Recently some approaches were proposed that use DNNs to generate faces\(^3\)\(^16\)\(^17\)\(^18\)\(^19\).

Our work also takes the speech-driven approach but it presumes that a video sequence of the target speaker is available. We do this without special hardware so that it can be applied to existing video sequences. Instead of acquiring 3D geometry via lighting, we employ the volumetric regression network by Jackson et al.\(^6\) to obtain a depth map of the speaker’s face for each frame of the video sequence. We then fit the 3DMM\(^2\) to the depth maps. For the audio features, we use mel-frequency cepstral coefficients (MFCCs)\(^20\).

3. Overview

Figure 2 shows an overview of our S2TH system. Given source video containing a speaker’s face as well as a target speech signal from the same speaker, our system firstly reconstructs the 3D face mesh \(M_k\) of the face in the \(k\)-th source video frame \((k = 1, \ldots, K)\). The audio track of the source video and the target speech are divided into audio frames and are converted to audio feature vectors, denoted by \(u_k\) and \(v_k\), respectively. For each target audio frame \(v_k\), we retrieve an appropriate source video frame based on the source audio frame \(u_k\) and a method based on Face2Face\(^2\), consequently forming a reordered sequence of the 3D face meshes \(\tilde{M}_k\). The original face mesh \(M_k\) is then modified to replace its mouth region with that of \(\tilde{M}_k\). The modified face mesh, \(\tilde{M}_k\), is then rendered using both the \(k\)-th original video frame and the \(k\)-th reordered sequence frame to synthesize a single frame for face reenactment.

4. 3D Face Mesh Reconstruction

To gain the 3D geometry of the speaker’s face, we fit a 3DMM to each frame in the source video\(^2\). For doing this, we first regres a volumetric representation of the 3D face geometry in each source video frame using\(^6\),
which is then converted to a depth map. This enables us to fit 3DMM without explicitly estimating the lighting while adjusting the 3DMM. We use the first 80 and 76 identity and expression eigenvectors, respectively, of the Basel Face Model 2017 dataset\(^{23}\). The model \(M_k\) for the \(k\)-th frame has \(N_v\) vertices and each vertex is in \(\mathbb{R}^3\). \(M_k\) is given by

\[
M_k = \mathcal{M} + \mathbf{L}_{id} \cdot \alpha_k + \mathbf{L}_{exp} \cdot \beta_k,
\]

where \(\mathcal{M} \in \mathbb{R}^{3 \times N_v}\) is the mean identity shape matrix, each of whose column vectors represent a vertex; \(\mathbf{L}_{id} \in \mathbb{R}^{3 \times N_v \times 80}\) and \(\mathbf{L}_{exp} \in \mathbb{R}^{3 \times N_v \times 76}\) are respectively the identity and expression eigenvectors organized into tensors; and \(\alpha \in \mathbb{R}^{80}\) and \(\beta \in \mathbb{R}^{76}\) are respectively the identity and expression coefficients. The operator \(\cdot\) represents the tensor dot product. \(M_k\) is the matrix whose column vectors are vertices, being the same as \(\mathcal{M}\). The subscript \(k\) of \(M_k\), \(\alpha_k\), and \(\beta_k\) indicating the frame index is omitted hereinafter for notational simplicity unless it is ambiguous.

**Preprocessing.** Prior to fitting, facial landmarks are extracted using the OpenPose face landmark detector\(^{23}\). We also process the video frames with the volumetric regression network of Jackson et al.\(^3\) to obtain the depth map through the volumetric representation for the face in each frame. Since we use the orthogonal projection to map an arbitrary 3D point to the image plane, the depth map can be easily converted to a set of 3D points \(D = \{d\}\) where \(d \in \mathbb{R}^3\). The three images from left in Fig. 3 illustrate this process.

**3DMM fitting.** We then use the depth map \(D\), facial landmark feature points projected onto the depth map, and the eigenvalues of the 3DMM to form an objective with these three terms for our 3DMM fitting problem (the right-most image in Fig. 3). Because this fitting problem only involves 3D geometry, the parameters to be optimized are \(\alpha\), \(\beta\), and the rigid transformation (i.e. rotation matrix \(R \in \mathbb{R}^{3 \times 3}\) and translation vector \(t \in \mathbb{R}^3\)) with a scaling factor \(s \in \mathbb{R}\), where the rotation matrix \(R\) is represented by the Euler angles \(\tau \in \mathbb{R}^3\).

The first term of our objective, \(C_{ver}\), involves the residual between the 3DMM vertices and the points on the depth map. For every optimization step, we use the nearest neighbors algorithm to correspond a depth map point to each 3DMM vertex. Taking the Euclidean norm of the residuals, the term is

\[
C_{ver} = \frac{1}{N_v} \sum_{i=1}^{N_v} \| sR M_i + t - d_i \|_2^2,
\]

where \(M_i\) is the \(i\)-th column vector of \(M\) and \(d_i \in \mathbb{R}^3\) is the nearest neighbor to \(M_i\) in \(D\).

The second term, \(C_{lan}\) involves similar residuals, but between the \(N_L\) corresponding landmarks on the 3DMM and the depth map. We use the landmark detector\(^{23}\) to find a set \(D_L = \{d_j\}_{j=1}^{N_L}\) of \(N_L\) landmarks for each frame in the video, where \(d_j\) is the 3D point obtained from the depth map that corresponds to the \(j\)-th landmark. The term is thus

\[
C_{lan} = \sum_{j=1}^{N_L} \| sR M_j + t - d_j \|_2^2,
\]

where \(M_j \in \mathbb{R}^3\) is the vertex in \(M\) corresponding to the \(j\)-th landmark, which can be preliminarily and manually found since the topology of 3DMM does not change.

The last term, \(C_{reg}\) is the statistical regularizer over the identity and expression parameters \(\alpha\) and \(\beta\) by their
rithmation and translation parameters via the Kabsch algo-

0. Additionally, we obtain an initial guess for the ro-
solution of the optimization problem with Gauss-Newton

\[ \sum_j \| sR x_j - d'_j \|^2_2, \]

which leads to 

\[ s = \frac{\text{Tr}(D'_L R X^\top)}{\text{Tr}(X X^\top)}. \]

5. Mouth Region Retrieval

Using the 3DMM fitted to each video frame and the features \( u_k \) from the source video’s audio track as well as the features \( v_k \) from the target speech, we retrieve 3DMM suitable for corresponding utterance in those obtained from the source video. We employ a method similar to Face2Face\(^2)\) but use audio features instead of 3DMM features.

Audio feature extraction. We process the au-
do track and the target speech to obtain features

\[ \text{MFCCs} \]

from the pre-emphasized audio in approxi-
mately 20 ms audio frames with 50% overlap. We use
the first 13 coefficients of the MFCCs, replacing the ze-
"roth coefficient, which corresponds to the sum of
the MFCC coefficients, with the RMS energy of the audio
frame. We take the first-order derivative of these 13
features using a nine-frame window to capture tempo-
ral features, such as the change in speech formants, of
the audio track, resulting in 26 total audio features.

Because the frame rate of these audio feature vectors is
higher than the 24 fps of video frame rate, we extract
the feature vector that is temporally nearest to each
video frame so that the audio features and the 3DMMs
are in correspondence, resulting in \( u_k \) for the source
video’s audio track and \( v_k \) for the target speech.

3DMM retrieval. We employ a method based on
Face2Face\(^3)\) for finding an appropriate 3DMM se-
quence. In the paper\(^2)\), appropriate 3DMMs are re-

respective eigenvalues \( \sigma^2_{id} \in \mathbb{R}^{80} \) and \( \sigma^2_{exp} \in \mathbb{R}^{76} \):

\[
C_{reg} = \sum_{n=1}^{80} \frac{\sigma^2_{id,n}}{\sigma^2_{id}} + \sum_{n=1}^{76} \frac{\sigma^2_{exp,n}}{\sigma^2_{exp}}.
\]

where the subscript \( n \) indicates the \( n \)-th component.

Following the 3DMM literature\(^{24,25)\), we use the
Gauss-Newton algorithm to fit these parameters. The
total cost of the optimization problem becomes

\[ C = C_{ver} + C_{lan} + C_{reg}. \]

To optimize for the parameters \( \Theta = [\alpha, \beta, r^\top, s^\top, t] \in \mathbb{R}^{163} \), we form the residual vector
\( e \in \mathbb{R}^N \), where \( N = 3N_v + 3N_L = 80 + 76 \) and
which contains each element of the residuals of \( C_{ver} \) and \( C_{lan} \) along with
each element in the summations of \( C_{reg} \). The Jacobian
matrix \( J \in \mathbb{R}^{N \times 163} \) of \( e \) thus defines the gradient vector
\( J^\top e \) of \( C \) with respect to \( \Theta \), allowing for the iterative
solution of the optimization problem with Gauss-Newton
steps:

\[ \partial \Theta = (J^\top J)^{-1} J^\top e. \]

Basically, we need to loop the above outlined opti-
mization process for all source video frame. However,
because our video only features one speaker, we optimi-
ize the identity parameters \( \alpha \) only during the first 20
frames, when the speaker’s expression is relatively un-
changed, and we keep the 20-th frame’s \( \alpha \) constant for
the remainder of the frames. We initialize \( \alpha \) and \( \beta \) to
0. Additionally, we obtain an initial guess for the ro-
rotation and translation parameters via the Kabsch algo-
rithm\(^{26)\), using the correspondence between the mean-
centered 3DMM landmarks \( X = (x_1, \ldots, x_{N_L}) \) (a
matrix consisting of \( N_L \) 3D points in \( M \) associated with
the landmarks, whose gravity center is translated so that it
coincides with the origin) and the mean-centered depth
map landmarks \( D'_L = (d'_1, \ldots, d'_L) \) (a matrix consisting
of \( N_L \) points in \( D \), translated in the same way as \( X \)).
The scaling factor can be found by minimizing the sum
of the Euclidean distances

[Image 301x675 to 421x781]
retrieved mainly based on 3DMM coefficients. In this work, however, the target is speech only and so the 3DMM coefficients are not available. Instead, we use audio features \( \{u_k\} \) and \( \{v_k\} \).

Specifically, our problem is to find an appropriate 3DMM sequence \( \{M_k\}_k \) for given target audio feature \( \{v_k\}_k \), and \( M_k \) is superimposed on the \( k \)-th source video frame as background. We presume the following four conditions:

(i) For frame \( k \) of given target audio feature \( v_k \), an appropriate 3DMM \( M_k \)’s corresponding audio feature \( u_k \) must be similar to \( v_k \).

(ii) For consistent lighting and shading, the rigid transformation (i.e. rotation and translation) applied to \( M_k \) must be similar to that of \( M_k \) in background frame.

(iii) For consistent texture between the background frame and the replaced mouth region, the landmark of the \( M_k \)’s face and \( M_k \)’s face must be similar.

(iv) For smooth temporal transition, conditions (i) and (ii) must hold true for the next frame.

As in\(^2\), we cast the problem of finding appropriate 3DMM sequence into shortest path finding over a directed acyclic graph (DAG). For target audio feature \( v_k \) whose appropriate 3DMM \( M_k \) is superimposed on the \( k \)-th source video frame, we find top 20 3DMMs from all 3DMMs, each of whose corresponding audio feature is closest to \( v_k \). These 20 3DMMs serve as nodes in our DAG, and each node has edge to all nodes in the adjacent time step \( k + 1 \). Let \( M_{k,l} \) denote the \( l \)-th node for the \( k \)-th frame. The edge weight \( w_{kl\ell} \) between \( M_{k+1,l} \) is given by

\[
w_{kl\ell} = \text{Dist}_A(v_k, u_l) + \text{Dist}_A(v_{k+1}, u_{\ell'}) + \text{Dist}_T(T_k, T_l) + \text{Dist}_T(T_{k+1}, T_{\ell'}) + \text{Dist}_L(D_{L,k}, D_{L,l}),
\]

where \( u_l \) is the source audio feature associated with the top 20 3DMMs for the \( k \)-th frame, \( u_{\ell'} \) is that for the \( k + 1 \)-th frame, and \( \text{Dist}_A \) is the Euclidean distance. Similarly, \( T_k \), \( T_l \), and \( T_{\ell'} \) are global transformation \( (r \) and \( t \), and we define their distance \( \text{Dist}_T \) by the root of squared sum of rotation’s and translation’s element-wise differences. \( D_{L,k} \) and \( D_{L,l} \) are sets of landmark 3D points of the \( k \)-th frame and the \( l \)-th node of the \( k \)-th frame. The last term \( \text{Dist}_L \) is the sum of the Euclidean distances.

The objective to be minimized is given by \( \sum_k w_{kl\ell} \), and the \( \text{A}^* \) shortest path algorithm minimizes the objective over \( l \) and \( \ell' \).

6. Mouth Region Replacement

To synthesize the talking head, replacing the mouth region in the \( k \)-th source video frame with that from \( M_k \), we merge \( M \) and \( M' \) into \( M' \). The talking head is rendered from \( M' \) and is superimposed onto the frame.

Mesh merging. To replace the mouth region of the \( k \)-th source video frame, we merge the mouth region of the 3DMM \( M \) from the reordered sequence with the original \( M \). This can be done based on a similar idea as Poisson image editing\(^27\). We preserve the displacement vectors of adjacent vertices in \( M \) around the mouth region, and the boundary condition is given by the surrounding vertices. Specifically, we formulate the problem into the following minimization:

\[
\arg\min_{\{M_i\}} \sum_{i \in \partial \Omega} ||M_i - \bar{M}_i||_2^2 + \alpha \sum_{(i,j) \in A} ||(M_i - M_j) - (\bar{M}_i - \bar{M}_j)||_2^2,
\]

where \( \bar{M}_i \) is the \( i \)-th vertex in merged 3DMM \( \bar{M} \). \( \Omega \) is the set of vertices in the mouth region and \( \partial \Omega \) is the set of boundary vertices (i.e., vertices that are not in \( \Omega \) but are connected to those in \( \Omega \)), and \( A \) is the set of adjacent pairs of vertices in \( \Omega \cup \partial \Omega \). The above minimization can be cast to a sparse linear system and can be solved easily.

Rendering. We then render the mouth region in \( M' \) with the standard rendering pipeline, but with two different textures applied to it: the first texture is the original video frame, and the second texture is corresponding video frame in the reordered sequence. Let \( I_k \) denote the rendered images with the first and second textures, respectively. We blend these two images with weights determined based on the distance from the mouth region. The mouth region is then superimposed onto the original video frame.

More specifically, in the standard rendering pipeline, we can render a 3D mesh with applying texture by using a mapping from a vertex in the mesh to the corresponding point in the texture image. For \( I_k \), we use the mapping between \( M_k \) and the source video’s \( k \)-th frame to render \( M \). For \( I_{\ell'} \), we use the mapping between \( M_k \) and its corresponding video frame (i.e., the \( k \)-th frame of the reordered sequence \{\( M \}_k \}). With this, \( I_k \) has a more natural rendering result around the mouth region, while \( I_{\ell'} \) is more natural in regions other than mouth.
region while the mouth region may be significantly distorted.

To blend these two images, we determine the weight for each pixel. We first extract mouth region $m$ from $\tilde{M}$, where the mouth region is preliminary identified. Extracted mesh $m$ is then rendered to find the hole region between lips (note that 3DMMs have no triangles between the lips and thus rendered images has a hole in the face, which is filled when rendering $I_k$ and $\bar{I}_k$). We generate a binary image, in which pixels in the hole region are 1 and 0 otherwise. We apply a distance transform to the binary image, which assigns to each pixel the distance from the nearest pixel whose value is 1. Let $\delta(p)$ denote the value of pixel $p$ in the distance image. We computer weight $\omega(p)$ for this pixel by

$$\omega(p) = \max(\tau - \delta(p), 0)/\tau,$$

(10)

where $\tau$ controls the region that $\bar{I}_k$ affects. That is, the value $I'(p) \in \mathbb{R}^3$ of pixel $p$ in the resulting image $I'$ is given by

$$I'(p) = (1 - \omega)I_k(p) + \omega(p)\bar{I}_k(p).$$

(11)

Temporal texture blending. The above texture blending can alleviate the discontinuities or distortions due to mouth region replacement. For temporal continuity in replaced mouth regions, we apply moving average over three successive frames to the textures of the reordered sequence, where the mouth regions are aligned based on their corresponding meshes.

7. Results

For visual comparison, we used videos on YouTube for both source videos and target speeches. For target speech, we excerpted the audio track of the YouTube videos. We picked out three politicians (Obama,

---

Fig. 4 Reenactment results. From top to bottom: target video from which the speech were excerpted, reenactment with the same person (Obama), reenactment with other person (Trump and Trudeau).

Fig. 5 Example video frames from the Trump sequences. Top: Target video (audio track were excerpted as target speech). Bottom: Reenactment.
Trump, and Trudeau) and used two different videos for each as the source and target∗, except Trudeau. All target speeches and source videos are about 120 sec, and we normalized the volume of the audio tracks.

Figure 4 shows reenactment results, in which three source videos (Obama, Trump, and Trudeau) were reenacted by Obama’s target speech. Since our S2TH system does not assume that the target speech comes with video, it can be used even when the speakers of the source video and the target speech are different, although it is designed for the same person. With these results, we show how our system generalizes to different speakers. The result demonstrates that our speech-driven reenactment works for different people. This can be because the MFCCs have the capability to encode the characteristics of phonemes regardless of speakers, which is reasonable as the MFCCs have been used in the classic speech recognition frameworks20). We consider that our system synthesizes plausible reenactment to a certain extent, although we can find some flaws. For example, mouth regions sometimes change discontinuously and frequently. This may be because the audio

feature-based mouth region retrieval is not well-posed due to one-to-many correspondences among a certain utterance and mouth regions. Also, mesh fitting failed with small errors, which causes inconsistent superimposition of mouth regions. We also reenacted Trump’s source video with his speech (Fig. 5). With this source video, the variations in facial expression and lip shapes were relatively small, and the result looks well. Figure 6 shows the source video frames and their corresponding synthesized frames. We can see that the synthesized frames have different mouth shapes; yet, they look visually plausible.

Figure 7 shows some typical failure results. Our S2TH system replaces the mouth region in the original 3DMM $M$ with that of the retrieved 3DMM $\bar{M}$. However, errors in 3DMM fitting cause displacement of replaced mouth regions, which are shown in the two video frames from left in Fig. 7. This problem was significant when face poses in the video frames are not frontal. Our system also takes the moving average of successive three frames for temporally more continuous reenactment; however, this causes blurring results when retrieved 3DMMs change a lot over time. The two video frames from right in Figure 7 demonstrate this problem, in which teeth are duplicated.

We conducted user study to evaluate the subjective quality of synthesized faces as well as the original face with 34 participants, mainly in their 20s, from some labs in two universities. They evaluated the following 6 Obama videos: (a) The source video and the original audio track. (b) The synthesized video without temporal texture blending and the original audio track (empirically, temporal texture blending affects a lot on the visual quality). (c) The synthesized video with temporal texture blending (the full system and the original audio track. (d)–(f) are the same video but with the tar-

∗ Following YouTube videos were used in our experiments.
Sources:
https://www.youtube.com/watch?v=a161Dv1dlJ4 (Obama)
https://www.youtube.com/watch?v=HfUKg5x50bE (Trump)
https://www.youtube.com/watch?v=-GUU7zt2t_g (Trudeau)
Targets:
https://www.youtube.com/watch?v=xXH5af77xw (Obama)
https://www.youtube.com/watch?v=Owz5D3wYo (Trump)
get speech. For easier comparison for the participants, (a)–(c) and (d)–(f) were played simultaneously and they could play back, stop, and pause it as many times as they want. We asked them two questions for each video, i.e., the naturalness of lip movement for given speech (lip movement) and the quality of the video itself (visual quality). The participants responded these questions a Likert scale ranging from 1 (bad) to 5 (good). Since (d)–(f) are the same video as (a)–(c), respectively, evaluation of (a)–(c) is enough for visual quality; however, we evaluated all videos.

Figure 8 shows the subjective evaluation results with MOS’s and standard deviations (error bars) over all participants. As for lip movement, the MOS for (a) is almost perfect, and synthesized faces (b) and (c) get low scores, as we can expect. Interestingly, the source video is the best among (d)–(e) when played with the target speech. Our full system (f) is better than (e) and as well as (b) and (c), which means that our full system can generate better lip movement compared to irrelevant speech ((c) vs. (f)) in some extent, although the difference is not significant. As for visual quality, our full system is better with temporal texture blending, although the difference is small. The deterioration of (d) would be caused by participants’ bias that they assumes lip movement and speech match.

8. Conclusion

This paper proposed a system for speech-driven face reenactment based on retrieval of suitable mouth regions using audio features, coined S2TH. Our system only requires speech of a certain person as a source and a video sequence that contains the person’s speech as target. We visually demonstrated that our speech-driven face reenactment works well and subjective evaluation showed that synthesized faces suit to the target speech in some extent; however, our experimental results also highlighted some flaws in visual quality, especially in 3DMM fitting and mouth region retrieval. Our future work includes to improve our mesh fitting algorithm for more stability and mouth region retrieval for smoother transitions. We also work on synthesizing a jaw movement that matches to the lip movement.

Acknowledgement. This work was supported by Japan Society for the Promotion of Science, Grants-in-Aid for Scientific Research (JSPS KAKENHI) under Grant No. 16H06302.

References

1) Matthew Brand. Voice puppetry. In SIGGRAPH, pages 21–28, 1999.
2) Justus Thies, Michael Zollhoefer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-time face capture and reenactment of RGB videos. In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pages 2387–2395, 2016.
3) Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. ACM Trans. Graphics, 36(4):95:1–95:13, 2017.
4) Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? In Proc. British Machine Vision Conference (BMVC), pages 1–12, 2017.
5) Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In Proc. ACM SIGGRAPH, pages 353–360, 1997.
6) Aaron S. Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Taimiroupolou. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In Proc. Int. Conf. Computer Vision (ICCV), pages 1–10, 2017.
7) Arno Schödl, Richard Szelski, David H. Salesin, and Irfan Essa. Video textures. In Proc. ACM SIGGRAPH, pages 489–498, 2000.
8) Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szelski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. In Proc. ACM SIGGRAPH, pages 75–84, 1998.
9) Justus Thies, Michael Zollhefer, Christian Theobalt, Marc Stamminger, and Matthias Niessner. HeadOn: Real-time reenactment of human portrait videos. ACM Trans. Graphics, 37:164:2–164:13, 2018.
10) Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kian Varanasi, Patrick Perez, and Christian Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In Computer Graphics Forum, volume 34, pages 193–204, Wiley Online Library, 2015.
11) Shengli Fu, Ricardo Gutierrez-Osuna, Anna Esposito, Praveen Kakumanu, K. and Oscar Garcia, N. Audio/visual mapping with cross-modal hidden markov models. IEEE Transactions on Multimedia, 7(2):243–252, 2005.
12) Lei Xie and Zhi-Qiang Liu. A coupled hmm approach to video-realistic speech animation. Pattern Recognition, 40(8):2325–2340, 2007.
13) Lei Xie and Zhi-Qiang Liu. Realistic mouth-synching for speech-driven talking face using articulatory modelling. IEEE Transactions on Multimedia, 9(3):500–510, 2007.
14) Lijuan Wang, Xiaojun Qian, Wei Han, and Frank Soong. K. Synthesizing photoreal talking head via trajectory-guided sample selection. In INTERSPEECH, volume 10, pages 446–449, 2010.
15) Robert Anderson, Bjørn Stenger, Vincent Wan, and Roberto Cipolla. Expressive visual text-to-speech using active appearance models. In Proc. CVPR, page 31823189, 2013.
16) Bo Fan, Lei Xie, Shan Yang, Lijuan Wang, and Frank K. Soong. A deep bidirectional LSTM approach for video-realistic talking head. Multimedia Tools and Applications, 75(9):5287–5309, 2016.
17) Taehwan Kim, Yisong Yue, and Matthew Iain Taylor, Sarah. A decision tree framework for spatiotemporal sequence prediction. In Proceedings of the 21th ACM SIGKDD International
Yuta Nakashima received the B.E. and M.E. degrees in communication engineering and the Ph.D. degree in engineering from Osaka University, Osaka, Japan, in 2006, 2008, and 2012, respectively. He is currently an Associate Professor at the Institute for Datability Science, Osaka University. His research interests include computer vision and machine learning and their applications. His main research includes video content analysis using machine learning approaches. He is a member of ACM, IEICE, and IPSJ.

Takaaki Yasui received the B.E. from Ritsumeikan University, Shiga, Japan, in 2017 and the M.E. in engineering from Osaka University, Osaka, Japan in 2019. His research interests include face synthesis.

Leon Nguyen visited Osaka University as an internship student from Sep. 2017 to Feb. 2018 while he was a master degree student at University of California at San Diego.

Noboru Babacuhi received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, in 1979, 1981, and 1984, respectively. From 1996 to 1997, he was a Visiting Scholar with the University of California, San Diego. He is currently the Dean of the Graduate School of Engineering, Osaka University. His research interests include image/video analysis, multimedia computing, and intelligent systems. Dr. Babaguchi is a fellow of the IEICE. He is also a member of the ACM, IPSJ, ITE, and JSAI.