Synthesizing Obama: Learning Lip Sync from Audio

SUPASORN SUWAJANAKORN, STEVEN M. SEITZ, and IRA KEMELMACHER-SHLIZERMAN, University of Washington

Fig. 1. Given input Obama audio and a reference video, we synthesize photorealistic, lip-synced video of Obama speaking those words.

Given audio of President Barack Obama, we synthesize a high quality video of him speaking with accurate lip sync, composited into a target video clip. Trained on many hours of his weekly address footage, a recurrent neural network learns the mapping from raw audio features to mouth shapes. Given the mouth shape at each time instant, we synthesize high quality mouth texture, and composite it with proper 3D pose matching to change what he appears to be saying in a target video to match the input audio track. Our approach produces photorealistic results.

CCS Concepts: •**Computing methodologies** → **Image-based rendering;** *Image manipulation;* Animation; Shape modeling;

Additional Key Words and Phrases: Audio, Face Synthesis, LSTM, RNN, Big data, Videos, Audiovisual Speech, Uncanny Valley, Lip Sync

ACM Reference format:
Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing Obama: Learning Lip Sync from Audio. *ACM Trans. Graph.* 36, 4, Article 95 (July 2017), 13 pages.
DOI: http://dx.doi.org/10.1145/3072959.3073640

1 INTRODUCTION

How much can you infer about someone's persona from their video footage? Imagine learning how to replicate the sound and cadence of a person's voice, how they speak, what they say, how they converse and interact, and how they appear and express themselves.

With tools like Skype, FaceTime, and other video conferencing solutions, we are increasingly capturing video footage of ourselves. In the case of public figures, there is significant video footage available online, in the form of interviews, speeches, newscasts, etc. Analyzing this video is quite challenging, however, as the faces are often shown in a near-profile view, the face region is small, and the lighting, dress, hair, and make-up varies significantly from one interview to the next (also, most of this video is proprietary).

In this paper, we do a case study on President Barack Obama, and focus on the specific task of learning to generate video of Obama from his voice and stock footage. Barack Obama is ideally suited as an initial test subject for a number of reasons. First, there exists an abundance of video footage from his weekly presidential addresses—17 hours, and nearly two million frames, spanning a period of eight years. Importantly, the video is online and public domain, and hence well suited for academic research and publication. Furthermore, the quality is high (HD), with the face region occupying a relatively large part of the frame. And, while lighting and composition varies a bit from week to week, and his head pose changes significantly, the shots are *relatively* controlled with the subject in the center and facing the camera. Finally, Obama's persona in this footage is consistent—it is the President addressing the nation directly, and adopting a serious and direct tone.

Despite the availability of such promising data, the problem of generating mouth video from audio is extremely difficult, due in part to the technical challenge of mapping from a one-dimensional signal to a (3D) time-varying image, but also due to the fact that humans are extremely attuned to subtle details in the mouth region; many previous attempts at simulating talking heads have produced results that look *uncanny*. In addition to generating realistic results, this paper represents the first attempt to solve the audio speech to video speech problem by analyzing a large corpus of existing video data of a single person. As such, it opens to the door to modeling other public figures, or ourselves (through analyzing Skype footage, e.g.,).

Audio to video, aside from being interesting purely from a scientific standpoint, has a range of important practical applications. The ability to generate high quality video from audio could significantly reduce the amount of bandwidth needed in video coding/transmission (which makes up a large percentage of current internet bandwidth). For hearing-impaired people, video synthesis

0730-0301/2017/7-ART95 $15.00
DOI: http://dx.doi.org/10.1145/3072959.3073640

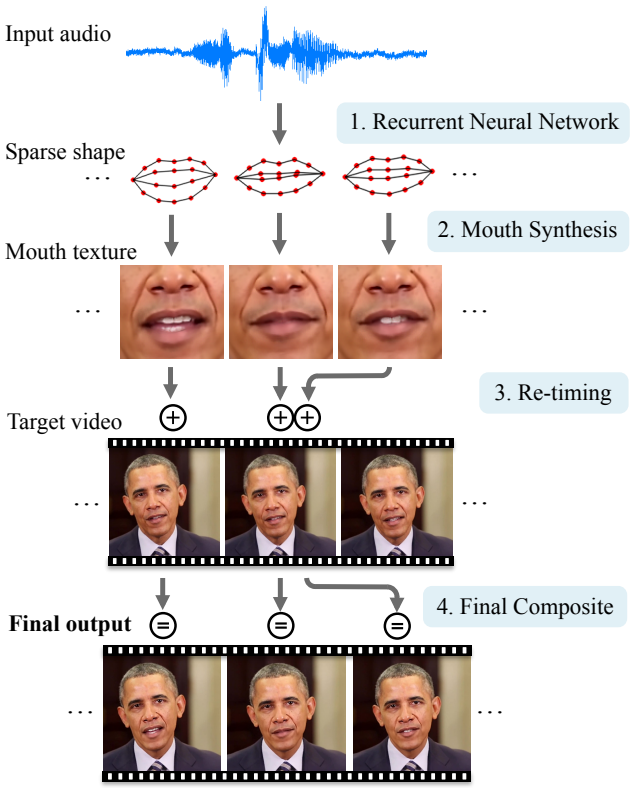


Fig. 2. Our system first converts audio input to a time-varying sparse mouth shape. Based on this mouth shape, we generate photo-realistic mouth texture, that is composited into the mouth region of a target video. Before the final composite, the mouth texture sequence and the target video are matched and re-timed so that the head motion appears natural and fits the input speech.

could enable lip-reading from over-the-phone audio. And digital humans are central to entertainment applications like film special effects and games.

Our approach is based on synthesizing video from audio in the region around the mouth, and using compositing techniques to borrow the rest of the head and torso from other stock footage (Fig. 2). Our compositing approach builds on similar talking head techniques like Face2Face [Thies et al. 2016], although Face2Face *transfers* the mouth from another video sequence whereas we synthesize the mouth shape directly from audio. A main contribution is our recurrent neural network technique for synthesizing mouth shape from audio, trained on millions of video frames, that is significantly simpler than prior methods, yet produces very convincing results. We evaluated many different network architectures to arrive at our solution, but found that a surprisingly simple approach based on standard LSTM techniques produces excellent results. In addition, our approach for generating photorealistic *mouth texture* preserves fine detail in the lips and teeth, and reproduces time-varying wrinkles and dimples around the mouth and chin.

2 RELATED WORK

Creating a photorealistic talking head model – a virtual character that sounds and appears real, has long been a goal both in digital special effects and in the computer graphics research community.

In their seminal paper, Bregler et al. [1997] demonstrated how to "rewrite" a person's lip movement in a video to match a new audio track represented as a phoneme sequence. Their approach was notable in automating all of the key components; face tracking, phoneme detection, mouth synthesis, and compositing, and produced compelling results for a few short sequences. However, the generality of the method in practice was limited due to insufficient phoneme and viseme reference data; as noted by the authors, correct triphones could be found only 6% of the time, and visemes had to be present for each desired pose. Nevertheless, Video Rewrite remains important as one of the very few techniques in the literature that operate on *existing video footage*, e.g., President John F. Kennedy, rather than training on laboratory-captured footage.

Almost all subsequent work that aims to produce photo-realistic speech from audio has required subjects captured in a controlled lab environment, e.g., [Anderson et al. 2013a; Fan et al. 2015a; Mattheyses et al. 2013; Shimba et al. 2015; Wang et al. 2010]. The advantage of the lab environment is that the pose of the subject, their lighting, and the words they utter can all be controlled. Typically, the subject is instructed to say pre-determined, phonetically-rich sentences in a neutral expression (or repeat with up to six different emotions [Anderson et al. 2013a]). In contrast to these lab-based approaches, our goal is to develop methods that can eventually be applied to online video footage, i.e., from interviews, speeches, or Skype feeds.

A requirement of most prior work in this area is the need for phoneme labels with millisecond-accurate timestamps. These labels are either provided manually, from a speech recognition system, or from a text-to-speech module [Fan et al. 2015a; Mattheyses et al. 2013]. Automatic phoneme labeling tends to be error-prone, and thus limits the quality of lip sync. The input is often converted to a sequence of phonemes, or diphones and triphones that encode the surrounding phonemes [Bregler et al. 1997; Fan et al. 2015a]. Additional known contextual information such as stress or position in the sentence may also be provided [Fan et al. 2015a]. Different schemes have been used to solve this regression problem that takes phonemes as input and predicts the visual speech. One is based on Hidden-Markov Models (HMM) [Fu et al. 2005; Sako et al. 2000; Xie and Liu 2007a,b] and the other constructs the final visual speech by concatenating visemes generated from a learned phoneme-to-viseme mapping [Taylor et al. 2012]. Both approaches require a significant amount of linguistic modeling and are quite complex. Voice Puppetry [Brand 1999] is notable as an early (HMM-based) approach that does not require phoneme labels. Recently, regression techniques based on decision tree [Kim et al. 2015], or deep bidirectional long short-term memory [Fan et al. 2015a] have been shown to outperform HMM-based approaches, although these still rely on phoneme labels. Similar to our approach, [Shimba et al. 2015] use an LSTM neural network that does not require phoneme labels and works directly on audio features. However, their network lacks a time-delay, which we have found crucial to producing good results. They unfortunately have no video results available online, and the images in the paper are limited to low-res face images with

mocap dots that have been manually annotated. As an alternative to recurrent neural network, another recent work [Taylor et al. 2016] uses a deep neural network to regress a window of visual features from a sliding window of audio features. In this work, we show that a simple time-shift recurrent network trained on an uncontrolled, unlabeled visual speech dataset is able to produce convincing results without any dependencies on a speech recognition system.

A key aspect of rendering realistic talking heads is synthesizing *visual speech*, i.e., the motion and appearance of the mouth and surrounding areas. While simple effects can be produced based on motion alone, i.e., using morphing techniques to warp the mouth into new poses [Brand 1999], the results often look *cartoony*, as they fail to capture important geometric and shading changes, e.g., creases, dimples, that occur as you move your mouth. Most modern methods therefore attempt to *synthesize* at least the mouth region of the face.

Face synthesis algorithms can be categorized into methods that use 3D face models [Anderson et al. 2013a; Cao et al. 2016, 2005; Thies et al. 2016] and others that operate on 2D images. Even in most 3D-based methods, the mouth and teeth are not explicitly modeled in 3D but are represented with 2D texture, e.g. in [Anderson et al. 2013a]. One of the most common 2D face texture synthesis techniques is Active Appearance Models (AAM) [Cootes et al. 2001] where a face is jointly modeled in a PCA-based representation for both the sparse shape and texture. Due to the low-rank PCA approximation, however, details such as mouth and teeth often appear blurry [Fan et al. 2015a].

Alternatively, several authors have chosen to use a teeth proxy [Anderson et al. 2013b; Garrido et al. 2015; Thies et al. 2015] or to copy teeth texture from original source frames [Thies et al. 2016; Vlasic et al. 2005; Wang et al. 2010]. Neither approach is full-proof, however, often appearing unnatural with artifacts near the lip boundary (see our overview of related work in the accompanying video).

A third source of artifacts in prior art is temporal flickering. To produce smoother results, triphones are sometimes used to find longer subsequences [Bregler et al. 1997], or a visually-smooth path through the original frames is optimized with respect to some similarity cost function [Bregler et al. 1997; Wang et al. 2010]. Another approach is to use optical flow to interpolate in-between frames [Li et al. 2012; Thies et al. 2016]. Note that some of the similarity metrics in these facial reenactment methods require a driving reference face to compare to, which is not available when only audio is given as input. Unfortunately, none of these techniques are full-proof, and some amount of flickering or unnatural warping often remains. In contrast, our mouth synthesis approach is simple, highly realistic, and naturally exhibits temporal continuity without the need for explicit temporal smoothing or interpolation.

We close our related work section by mentioning perhaps the simplest rewrite approach is to take raw video clips of a person talking, chop them up into word-long segments, and reorder the clips to fit the words of any desired new sentence. The popular website `talkobamato.me` does just that; the results can be fun, but also distracting, as the the background, pose, and tone changes rapidly and discontinuously.

3 AUDIO TO VIDEO

Given a source audio track of President Barack Obama speaking, we seek to synthesize a corresponding video track. To achieve this capability, we propose to train on many hours of stock video footage of the President (from his weekly addresses) to learn how to map audio input to video output.

This problem may be thought of as learning a sequence to sequence mapping, from audio to video, that is tailored for one specific individual. This problem is challenging both due both to the fact that mapping goes from a lower dimensional (audio) to a higher dimensional (video) signal, but also the need to avoid the uncanny valley, as humans are highly attuned to lip motion.

To make the problem easier, we focus on synthesizing the parts of the face that are *most correlated* to speech. At least for the Presidential address footage, we have found that the content of Obama's speech correlates most strongly to the region around the mouth (lips, cheeks, and chin), and also aspects of *head motion* – his head stops moving when he pauses his speech (which we model through a retiming technique). We therefore focus on synthesizing the region around his mouth, and borrow the rest of Obama (eyes, head, upper torso, background) from stock footage.

We use the following terms throughout the paper: the many hours of online weekly address video is referred to as *stock* video footage. Stock footage will be used to train our audio-to-shape neural net. The input audio track is the *source*, and the *target video* is a stock video clip into which we composite the synthesized mouth region.

The overall pipeline works as follows (Fig. 2): Given an audio of Obama, we first extract audio features to use as input to a recurrent neural network that outputs, for every output video frame, a sparse mouth shape (Section 3.1). From the sparse mouth shape, we synthesize texture for the mouth and lower region of the face (Section 3.2). The mouth texture is then blended onto a stock video that is modified so that the final head motion appears natural and matches with the given input speech (Section 3.3). During blending, the jaw line is warped to match the chin of the new speech, and the face is composed to a target frame in the original pose. (Section 3.4).

3.1 Audio to Sparse Mouth Shape

Rather than synthesize video directly from audio, we decompose the problem into two steps: 1) map from audio features to sparse shape coefficients, and 2) map from shape to mouth texture. Like [Shimba et al. 2015], we skip the error-prone process of phoneme extraction, and map directly from audio to sparse shape features.

In this step, we represent the audio using standard MFCC coefficients, and the mouth shape by 18 lip fiducials, ranked reduced by a PCA basis, as described next.

Audio Features. For audio features, we use Mel-frequency cepstral coefficients (MFCC) which are computed as follows:

(1) Given a 16KHz mono audio, we normalize the volume using RMS-based normalization in ffmpeg [Bellard et al. 2012; Robitza 2016].
(2) Take the Discrete Fourier Transform on every 25ms-length sliding window over the audio with 10ms sampling interval.
(3) Apply 40 triangular Mel-scale filters onto the Fourier power spectrum, apply logarithm to the outputs.

(4) Apply the Discrete Cosine Transform to reduce dimensionality to a 13-D vector.

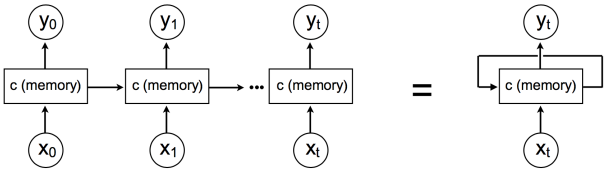
The final 28-D output feature vector consists of the 13-D vector plus the log mean energy to account for volume, and their first temporal derivatives.

Mouth Shape Features. To compute the mouth shape representation, we first detect and frontalize Obama's face in each video frame using the approach in [Suwajanakorn et al. 2014]. For each frontalized face, we detect mouth landmarks using [Xiong and De la Torre 2013] which gives 18 points along the outer and inner contours of the lip. We reshape each 18-point mouth shape into a 36-D vector, apply PCA over all frames, and represent each mouth shape by the coefficients of the first 20 PCA coefficients; this step both reduces dimensionality and decorrelates the resulting feature set. Finally, we temporally upsample the mouth shape from 30Hz to 100Hz by linearly interpolating PCA coefficients, to match the audio sampling rate. Note that this upsampling is only used for training; we generate the final video at 30Hz.

3.1.1 Recurrent Neural Network. We seek to learn a mapping from MFCC audio coefficients to PCA mouth shape coefficients. Let's model this mapping using a Neural Network.

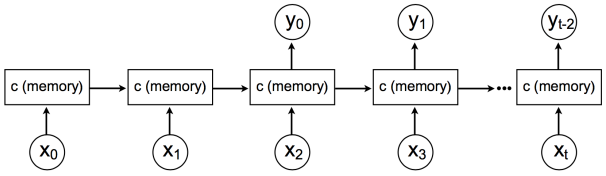
Consider Obama saying the word "America". He begins by making the sound *Uhhh*, which is a cue to the mouth synthesizer that he should start opening his mouth. Clearly our network needs the latest audio features as input to determine the mouth shape. But note also that the current mouth shape also depends on the *previous* shape; he will continue to say *Uhhh* for several milliseconds during which time the mouth will open wider, rather than reopening from a closed state.

These considerations motivate a *recurrent neural network* (RNN): at each moment in time, this network takes the latest audio input



x_t, uses it to modify its hidden state, aka *memory* c, and outputs a new mouth shape vector y_t for that time instant, as well as passing its memory forward in time. In this way, the memory vector (which can be as large as you want), can represent events potentially far into the past. This RNN technique is very popular for learning time-series problems. RNNs are similar to Hidden Markov Models (HMMs), which have been the basis for most prior talking face work [Fu et al. 2005; Sako et al. 2000; Xie and Liu 2007a,b], but RNNs provide a more general memory mechanism, nonlinear transitions, and better performance on many problems. Many variants of RNNs have been proposed, and we use Long Short Term Memory (LSTM) models which provide a more efficient mechanism for modeling long term dependencies. LSTMs work by replacing each hidden unit with a series of gates that are specifically designed to facilitate remembering and forgetting (when useful) information (see [1] for a nice tutorial).

Sometimes, your mouth moves *before* you say something. I.e., by the time Obama says *Uhhh*, his mouth is already open. Hence, it's not enough to condition your mouth shape on past audio input – the network needs to look into the future. Shimba et al. [2015] point this out as a limitation of their LSTM method. One possible solution is to make the network *bidirectional*. Indeed [Fan et al. 2015b] uses a bidirectional LSTM to exploit future context. However, bidirectional networks require much more compute power and memory to train, as they must be unfolded in the backpropagation process, which usually limits not only the length of training examples, but also the length of the output. Instead, a much simpler way to introduce a short future context to a unidirectional network is to add a time delay to the output by shifting the network output forward in time as explored in [Graves and Schmidhuber 2005] as "target delay." While bidirectional LSTMs are popular for speech recognition problems [Graves and Schmidhuber 2005], we find that the simpler time delay mechanism is sufficient for our task, likely due to the need to look less far in the future for audio to video, compared with speech recognition which may require looking multiple words ahead. We find that introducing this time delay dramatically improves the quality of results (Section 4.2), compared to prior architectures like [Shimba et al. 2015] which omit it. A time-delayed RNN (for a delay of $d = 2$) looks like this:



We opt for a simple single-layer unidirectional LSTM [Hochreiter and Schmidhuber 1997]. In Section 4.2, we show a comparison with other architectures such as multi-layer LSTMs, but we did not find significant improvements to merit the additional complexity. Given $[x_1, \ldots, x_n], [y_1, \ldots, y_n]$ as input and output vector sequences, our standard LSTM network is defined by the following functions:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$c_t = c_{t-1} f_t + i_t \tanh(W_j \cdot [h_{t-1}, x_t] + b_j) \quad (4)$$

$$h_t = \tanh(c_t)\sigma(o_t) \quad (5)$$

$$\hat{y}_{t-d} = W_y h_t + b_y \quad (6)$$

where f, i, o, c, h are forget gate, input gate, output gate, cell state, cell output as proposed in [Hochreiter and Schmidhuber 1997]. σ is the sigmoid activation. Note that the cell state and cell output are transformed with tanh. $\hat{y}_{t-d}$ represents the predicted PCA coefficients at time $t - d$ where d is the time-delay parameter. Learned parameters are weight matrices W and bias vectors b. We use a 60 dimensional cell state c and a time delay d of 20 steps (200ms). The

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

network is minimized using L2-loss on the coefficients and trained using Adam optimizer [Kingma and Ba 2014] implemented in TensorFlow [Abadi et al. 2016], on many hours of stock Obama weekly address footage. More details can be found in Section 4.2.

3.2 Facial Texture Synthesis

In this section, we describe our approach for synthesizing high detailed face textures from sparse mouth shapes. We focus on synthesizing the lower face area, i.e., mouth, chin, cheeks, and area surrounding the nose and mouth. Figure 8 shows the mask. The rest of Obama's appearance (eyes, head, torso, background) comes from stock footage of Obama's weekly addresses. Our texture synthesis algorithm is designed to satisfy two key requirements: 1) sharp and realistic appearance per video frame, 2) temporally smooth texture changes across frames.

We explored several approaches for synthesizing the mouth area based on prior art (see Section 2) but found that results were either too blurry (in the case of Active Appearance Models), the teeth were too non-rigid (with warping/flow techniques), or the illumination was mismatched. We compare these different techniques in Figure 10 and the supplementary video. Instead, we propose an approach that combines weighted median and high frequencies from a teeth proxy.

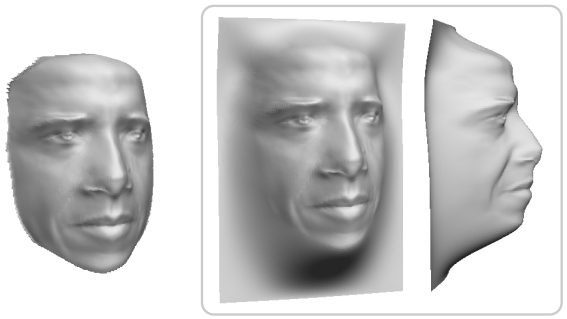
Given a sparse mouth shape sequence and a target video, we process each mouth shape independently. The algorithm overview is as follows: per mouth PCA shape, select a fixed number of target video frames that best match the given shape; apply weighted median on the candidates to synthesize a median texture; select teeth proxy frames from the target video, and transfer high-frequency teeth details from the proxy into the teeth region of the media texture. We describe those steps in detail below.

3.2.1 Candidate frame selection: Given a generated mouth shape, we seek a set of best matching frames from the target video. Candidate frames are selected as follows: we run a landmark detector [Xiong and De la Torre 2013] on the target video, estimate 3D pose, and frontalize every frame using a 3D model of Obama (Figure 3). We compute the 3D face model using [Suwajanakorn et al. 2014], and augment it with rough approximations of chin and background shape. We found that the latter step significantly improved frontalization results. The 3D face model is extended to include the chin by assuming a planar background and solving for a smooth surface that connects the face to the background. Specifically, we minimize the surface's second partial derivatives: suppose the initial surface parametrized on a 2D depth map $f(x, y) : \Omega \to \mathbb{R}$ is only given on the face region $\Omega' \subset \Omega$. We solve for a new f^* on the entire domain Ω by:

$$\min_{f^*} \iint_{\Omega} \left(\frac{\partial^2 f^*}{\partial x^2}\right)^2 + \left(\frac{\partial^2 f^*}{\partial y^2}\right)^2 \mathrm{d}x\,\mathrm{d}y \tag{7}$$

$$\text{subject to } f^*|_{\Omega'} = f|_{\Omega'} \text{ and } \nabla f^*|_{\partial\Omega} = 0 \tag{8}$$

where $\partial\Omega$ denotes the boundary of Ω. The objective and constraints are turned into a linear least squares problem (with soft constraints) on a discrete pixel grid by finite differences. The extended 3D model is shown in Figure 3b. Next, we estimate drift-free 3D pose for each frame using [Suwajanakorn et al. 2014], place the model onto each frame, and back project the head to the frontal view. Even



a) Reconstruction from [Suwajanakorn et al. 2014]

b) **Our extended model with a simple plane**

Fig. 3. We augment an Obama model (a) reconstructed from [Suwajanakorn et al. 2014] with a simple, near-planar background. This extension is used for frontalizing the chin and neck in addition to the face region.

though this extended geometry is inaccurate away from the face area, it suffices as a frontal-warping proxy since the final synthesized texture will be warped back to the original pose using the same geometry.

Texture is synthesized in an area defined by a manually drawn mask that includes the lower face and neck areas in frontal pose. The mask is drawn only once. Additionally, since in some poses the neck is occluded, we automatically mask out the clothing in every video frame (by means of simple thresholding in HSV space; the same threshold is used in all results) and in-paint the masked region using [Telea 2004], and the OpenCV [Bradski 2000] implementation.

Once all frames are frontalized and pose is computed, n frames that have the smallest L^2 distance between frame's mouth shape and target mouth shape are selected.

3.2.2 Weighted median texture synthesis: Given a set of frontal mouth candidate images $\{I_1, \ldots, I_n\}$ with associated mouth shapes $S_1, \ldots, S_n$ where $S_i \in \mathbb{R}^{2\times 18}$ and a target mouth shape S_{target}, we first compute the weighted median per pixel (u, v):

$$\text{median}(u, v) = \arg\min_{c} \sum_{i=1}^{n} w_i |I_i(u, v) - c| \tag{9}$$

$$\text{subject to } \exists k, c = I_k(u, v) \tag{10}$$

This is computed independently for each of the R,G,B channels. c is the output pixel intensity and w_i represents how similar S_i is to S_{target} and is computed by:

$$w_i = e^{\frac{-\|S_i - S_{\text{target}}\|_2^2}{2\sigma^2}} \tag{11}$$

Choosing the right σ is critical. Small σ will create a peak distribution on a few images which can cause temporal flickering, similar to taking a single original frame, and large σ can produce a blurry result. Moreover, the optimal σ for one target shape can be suboptimal for another target shape depending on the number of good candidates, i.e., ones with small $\|S_i - S_{\text{target}}\|$. Because the optimal σ is tied to the number of good candidates, we adaptively select σ such that the weight contribution of n candidates is α-fraction of the weight of all available frames. In other words, we solve for σ

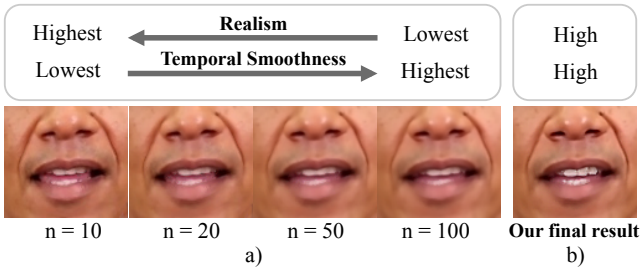


Fig. 4. a) shows the visual quality with respect to the number of candidates (n). Even though averaging (through median) lower numbers produce sharper results, they are not temporally smooth when used in an animation. On the other hand, our final result shown in b) both minimizes blur and is temporally smooth.

for each target shape such that

$$\sum_{i=1}^{n} w_i(\sigma) = \alpha \sum_{i=1}^{N} w_i(\sigma) \tag{12}$$

where N is the total number of video frames. This can be efficiently solved with a binary search on σ. We fix α to 0.9 and tune n for the best balance between the visual quality and temporal smoothness (Figure 4). Once n is selected, only σ will vary for each output frame. With all w_i computed, Equation 9 is solved efficiently by sorting pixel intensities and picking the intensity situated at the half of the total weight.

3.2.3 Teeth Proxy: Synthesizing realistic teeth is surprisingly challenging. The teeth must appear rigid, sharp, pose aligned, lit correctly, and properly occluded by the lip. Our evaluation of prior methods (see Section 2 and submission video) all exhibited problems in one or more of these aspects. In particular, AAM-like models yielded blurry results, while teeth proxy approaches produced compositing artifacts.

We achieved our best results with a new, hybrid technique that combines low-frequencies from the weighted median texture, and high frequency detail from a teeth proxy image. This idea of combining frequencies from different sources is based on [Oliva et al. 2006], which also inspires [Shih et al. 2014; Suwajanakorn et al. 2015]. The key insight is that the median texture provides a good (but blurry) mask for the teeth region, whereas the teeth proxy does a good job of capturing sharp details. Hence, we apply the high frequencies of the teeth proxy only in the (automatically detected) teeth region of the median image.

The teeth proxy reference frame is manually chosen to be one of the target frames where teeth are frontal-facing and highly visible. We need one proxy for the upper and another for the lower set of teeth; these two proxies may be chosen from different frames. This is a step in approach that is manual, and must be repeated for each target sequence.

The teeth region in the median texture, to which we will transfer proxy details, is detected by applying a threshold (low-saturation, high-value) in HSV space within the mouth region given by the landmarks.

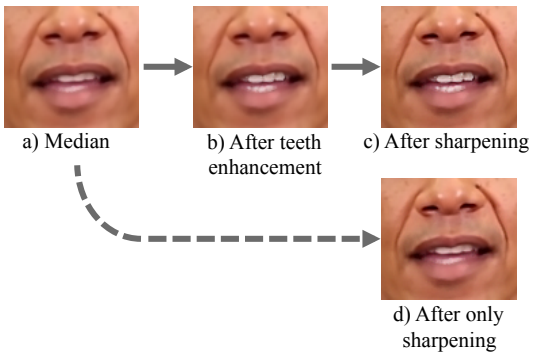


Fig. 5. The effects of proxy-based teeth enhancement. a) shows the weighted median texture computed in Section 3.2.2. b) is after applying proxy-based teeth enhancement in Section 3.2.3. c) is after an additional high-pass filter. d) shows the result of a high-pass filter on the median texture, without the teeth proxy.

Given a teeth proxy reference frame T whose pixel values are converted to be within $[0, 1]^3$, we apply a high-pass filter to T:

$$H_{\sigma,s}(T) = (T - G_\sigma * T) \times s + 0.5 \tag{13}$$

where G_σ is a Gaussian kernel with standard deviation σ, $*$ is the convolution operator, and s is the adjustable strength. We also truncate the value of $H(T)$ to be within $[0, 1]$. Then given a median texture I, we compute the final texture I' for pixel (u, v) in the mouth region as:

$$I'(u,v) = \begin{cases} 2I(u,v)H(u,v) & \text{if } H(u,v) < 0.5 \\ 1 - 2(1 - I(u,v))(1 - H(u,v)) & \text{otherwise} \end{cases} \tag{14}$$

Additionally, we enhance I with multiple $H_{\sigma,s}$ of different σ's to handle various frequency scales. This high-frequency addition, however, only works for the upper teeth, since they are stationary with respect to the face. For the lower teeth, we shift $H(u, v) \leftarrow H(u+\Delta u, v+\Delta v)$ by $\Delta u, \Delta v$ which represent the jaw difference between I and T estimated from the lower lip landmarks. Without accurate landmarks, this can cause flickering. So, instead of using the landmark output from our network or running a landmark detection on I which can be noisy, we compute a weighted average of the lip landmarks of all image candidates using the weights from Equation 11 to obtain an accurate, temporally smooth jaw location. The enhanced teeth texture after an additional spatial sharpening (unsharp masking) is illustrated in Figure 5.

3.3 Video Re-timing for Natural Head Motion

We assume the availability of a target video, into which our synthesized mouth region will be composited. Any of Obama's weekly presidential address videos work well as targets, for example. Since the speech in the target video is different from the source (input) speech, a naive composite can appear awkward. In particular, we've observed that it's important to align audio and visual pauses; if Obama pauses his speech, but his head or eyebrows keeps moving, it looks unnatural. To solve this problem, we use dynamic programming to re-time the target video. We look for the optimal monotonic

mapping between N synthesized mouth animation frames and M target video frames such that:

- it prefers more motion during utterance and minimal motion during silence
- any target video frame may be repeated at most once but never skipped. This limits slow downs to at most 50% and the video cannot be sped up; otherwise a noticeable jump or freeze can occur.
- it prefers sections of the target video where slowing down would be least noticeable, i.e., not during blinking or quick expression changes.

To formulate the dynamic programming objective, we first compute the motion speed for each frame j in the source video, denoted by $V(j)$, using the first derivative of the facial landmark positions as well as a binary flag indicating a blink, denoted by $B(j)$, by applying a threshold on the size of the eye landmarks. Then we assign $V(j) \leftarrow V(j) + \alpha_B B(j)$ where α_B is a balancing weight. For the source speech, we compute a binary flag, denoted by $A(i)$, indicating non-silence by applying a threshold on the audio volume. These binary flag sequences typically contain salt and pepper noise (random 0's or 1's), which we filter out by applying dilation followed by erosion to remove small gaps of 0's. We additionally filter out very short consecutive sequences of 0's or 1's by a second threshold. The recurrence relation is defined as follows:

$$F(i,j,0) = \min(F(i-1,j-1,0), F(i-1,j-1,1)) + G(i,j) \quad (15)$$

$$F(i,j,1) = F(i-1,j,0) + \alpha_s V(j) + G(i,j) \quad (16)$$

$$G(i,j) = \begin{cases} V(j) & \text{if } A(i) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

$$- \begin{cases} \alpha_u V(j) & \text{if } A(i-2) = 1 \text{ and } A(i-3) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

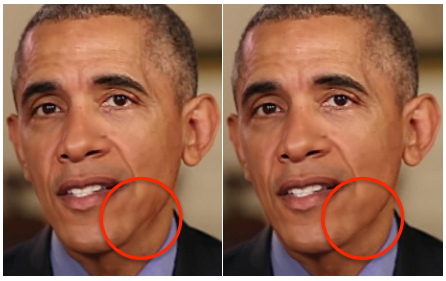
where $F(i,j,k)$ stores the score when the i^{th} mouth shape frame is matched with the j^{th} video frame and k is the number of times this mouth frame has been repeated. α_s penalizes repeating frames during large motion. $G(i,j)$ is an auxiliary function that penalizes large motion during silence and small motion during utterance with adjustable weight α_u. We initialize the base cases as follows:

$$F(0,j,0) = \begin{cases} V(j) & \text{if } A(i) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

$$F(i,0,0) = \begin{cases} \infty & \text{if } i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

$$F(i,j,1) = \infty \text{ if } i = 0 \text{ or } j = 0 \quad (21)$$

The optimal score is $\min_j \left\{ \min(F(N-1,j,0), F(N-1,j,1)) \right\}$ and the optimal mapping is found by back-tracing the minimal path through the 3-dimensional F array with an overall $O(MN)$ time complexity. We set $\alpha_B = 1$ and $\alpha_s = \alpha_u = 2$ in our implementation. Finally, to avoid having a completely static motion for the final composite when a frame is repeated, we warp the repeated frame half way between the previous and next frame. Specifically, suppose frame i is a copy of frame $i-1$, we compute optical flows $F_{(i-1)\to(i+1)}$ and $F_{(i+1)\to(i-1)}$, and define the final frame i as the average of frame $i-1$ warped by $0.5F_{(i-1)\to(i+1)}$ and frame $i+1$ warped by $0.5F_{(i+1)\to(i-1)}$.



a) Without jaw correction b) **With jaw correction**

Fig. 6. a) shows a jawline discrepancy when the mouth texture of a different speech is blended onto a target video frame. b) shows our corrected result where two jawlines are connected.

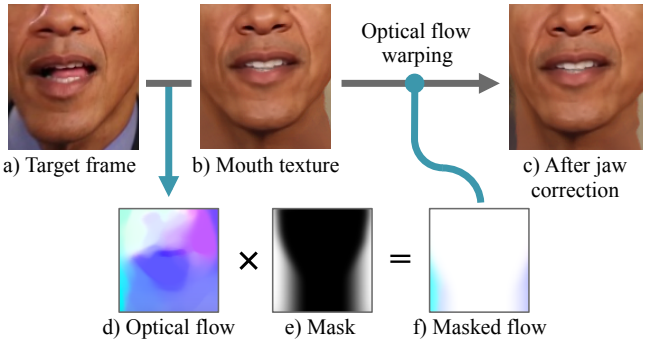


Fig. 7. To prepare the mouth texture so that the final jawline appears seamless in Figure 6, we first compute optical flow between a target video frame (a) and our mouth texture (b). This resulting flow (d) is masked by (e) to produce (f) which is used to warp our mouth texture and produce the final texture in (c).

3.4 Composite into Target Video

Compositing into the target video is the final step of our algorithm. By this point, we have created a lower face texture for each mouth shape corresponding to the source audio. We have also re-timed the target video to naturally fit silence or talking moments in the source audio. The key part of the composition step is to create a natural, artifact-free chin motion and jawline blending of the lower face texture into the target head. Figure 6 illustrates how blending may look if jawlines are not explicitly corrected. The artifacts are especially visible when watching a video. Therefore, we created a jaw correction approach that operates per frame.

3.4.1 Jaw correction: The algorithm is illustrated in Figure 7. Optical flow (d) is computed between the lower face texture frame (b) and target video frame (a). Next, an alpha map is created based on fiducials to focus only on the area of the jawline (e). The flow is masked by the alpha map and then used to warp the lower face texture to fit the target frame.

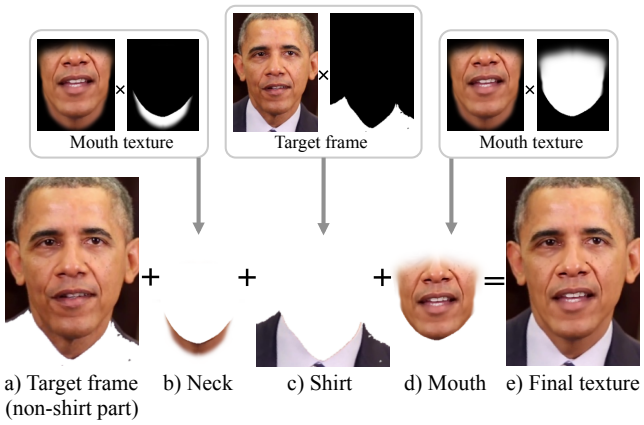


Fig. 8. The final composite is produced by pyramid blending of the following layers from back to front: a) the target frame, b) the neck region under the chin in the mouth texture, c) Obama's shirt from the target frame, d) the mouth.

3.4.2 Final compositing: Figure 8 illustrates the final masking and blending steps. The blending is done using Laplacian pyramids [Burt and Adelson 1983] in a layer based fashion. There are four layers that are blended in the following order from front to back: 1) Lower face texture (excluding the neck), 2) torso (shirt and jacket), 3) Neck, and 4) the rest. Parts 1 and 3 come from the synthesized texture, while parts 2 and 4 come from the target frame. The neck mask is the region under the chin in our synthesized mouth texture and the mouth mask is the region above. The chin is determined by splining face contour landmarks estimated from DLIB library [King 2009]. In some target videos where the background is easy to segment, e.g. when it is a solid black, we create an additional mask for the background (via a color detector) and add it to the shirt mask to have the second layer include both the shirt and background. Although this is optional, it helps prevent the artifact shown in Figure 14b. The final texture is rendered back to the target frame pose using the 3D shape estimated in the synthesis part (Section 3.2).

4 EXPERIMENTS

In this section, we describe implementation details, evaluations, comparisons to prior work, limitations, and applications.

4.1 Running times and hardware:

We report the following runtime based on NVIDIA TitanX for RNN inference, and Intel Core i7-5820K for other computations. For a source audio of 66 seconds in length, on a single CPU core, it took 45 minutes in total to produce a 30fps output video. The breakdown is as follows: 5 seconds to run RNN inference and generate 1980 mouth shapes (30fps for 66 seconds); mouth texture synthesis took 0.1s per frame (3.3 minutes total); and the final composite including chin correction, masking, and rendering took 0.35s per frame (11.5 minutes total). The retiming dynamic programming solution took 0.2s for the entire sequence, with an additional 4s per repeated frame for optical flow interpolation [Liu et al. 2008]. In practice, we parallelized most computations on a 24-core CPU and reduced the runtime from 45 to 3 minutes total (0.1s per frame).

For network training, the time for [Suwajanakorn et al. 2014] to preprocess 17-hour Obama video (pose estimation, frontalization) took around 2 weeks on 10 cluster nodes of Intel Xeon E5530. The network training for 300 epochs on NVIDIA TitanX and Intel Core i7-5820K took around 2 hours.

4.2 LSTM Architecture and Data

For training we downloaded 300 weekly addresses available online[2] spanning 2009 to 2016. Each address lasts about 3 minutes on average, resulting in total of 17 hours of video. We extracted frames at 30fps and obtained around 1.9 million video frames. We randomly split out 20% of the addresses (3 hours) for validation and used 80% (14 hours) for training.

Our network consists of 60 LSTM nodes (dimension of c) and uses a 20 step time-delay d, corresponding to 200ms. We train the network with a batch size of 100 using truncated backpropagation through time with 100 time steps. We use the ADAM optimizer [Kingma and Ba 2014] with learning rate 0.001, implemented in TensorFlow [Abadi et al. 2016]. Each dimension in the input vector is normalized by its mean and variance, but the output is unnormalized to keep the relative importances of the PCA coefficients. Training took 3 hours in total for 300 epochs on a NVIDIA TitanX.

We found that augmenting the LSTM with a time delay was critical for improving validation loss and visual quality. This modification effectively increases the receptive field beyond that of the MFCC window (25ms) to at least 200ms of future audio context. Figure 9 shows validation losses by varying the time delay steps for our single-layer 60-node LSTM network. Without the time delay, both the training and validation losses are high and the visual lip-sync quality is poor. Table 1 shows validations losses for varying time delays as well as the number of LSTM nodes. We found that for our architecture, the time delay of 200ms gives consistently lower validation losses across different numbers of LSTM nodes. Performance decreases beyond 200ms, likely due to the need to propagate information further across time.

Time Delay:	50ms	100ms	200ms	400ms
L1-30	5.824	4.946	4.572	5.147
L1-60	5.782	4.877	4.400	5.089
L1-90	5.726	4.841	4.391	5.009
L1-120	5.732	4.946	4.572	5.147

Table 1. Validation losses for single-layer (L1) networks with varying (30, 60, 90, and 120) LSTM nodes and time delays.

Other network architectures are evaluated in Table 2 by keeping the time delay at 200ms and varying the number of stacked layers, LSTM nodes, and the dropout probability of the standard RNN regularization [Zaremba et al. 2014].

Additionally, we explored other regularization techniques, e.g. variational RNN dropout [Gal 2015] on a few configurations but did not find a major improvement. These validation losses have high

[2]https://www.whitehouse.gov/briefing-room/weekly-address

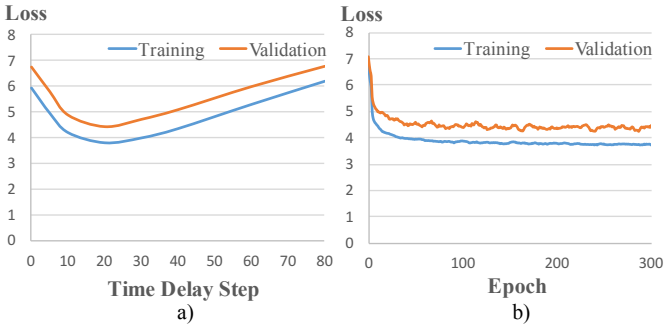


Fig. 9. a) shows the losses at the end of the training of networks with varying time delay steps from 0 to 80 (800ms) trained with 300 unfold time steps. b) plots loss during training of our single-layer LSTM network with 20 time delay steps.

Dropout probability:	0	0.1	0.3	0.5
L2-30	4.449	4.587	4.881	5.252
L2-60	4.389	4.420	4.621	4.923
L2-90	4.403	4.347	4.498	4.754
L3-30	4.409	4.548	4.850	5.237
L3-60	4.402	4.386	4.585	4.881
L3-90	4.439	4.310	4.487	4.718

Table 2. Validation losses for two (L2) and three (L3) layers networks with various LSTM nodes and dropout probability.

variance and do not necessarily translate to better visual lip-sync quality after reaching a certain value. One reason is that we do not have ground-truth mouth landmarks and instead rely on landmark estimates from imperfect algorithms. While we filtered out clear failure cases such as when the mouth shape is irregular or not detected, other error cases remain in both the training and validation sets. As such, zero loss does not equate to perfect visual results. We believe recently developed techniques such as recurrent batch normalization can further improve the loss, but larger accuracy gains may require improving the landmark data. For our purpose, we opt for the simplest network (L1-60, with 200ms delay) that achieves a similar loss and empirically similar visual quality to the more complex models.

4.2.1 Word/phoneme overlap between target and input audio: Here we investigate the amount of phoneme and word overlap between the audio from a target video (used for mouth synthesis) and the input audio. For 4-Obama result (Video E), we compute the percentage of words in input audio that exist in each of the four target videos: Top-left 41%, -right 45%, bottom-left 52%, -right 48%. In other words, half of the words spoken in the input audio do not exist in the target videos. In terms of phoneme, diphone, triphone, tetraphone, pentaphone overlaps (average across 4 targets): 99.9%, 82.9%, 35.7%, 12.1%, 4.9%, respectively. There is less than 5% chance to find similar 5 consecutive phonemes in the target videos. (An average word in the input consists of 3.9 phonemes.)

4.2.2 Training set size: Finally, we evaluated the effect of varying the amount of training data on the quality of the output video. We trained our network with: 0.35% of the data (3 minutes total), 10% (1 hour total), 50% (7 hours), and the full test dataset (14 hours). The supplementary video shows how quality improves with more training data. In particular, with more training data, lip-sync quality improves significantly and mouth jitter is reduced. There is significant improvement at each step, even from 7 hours to 14, indicating that having a large amount of training data is critical.

4.3 Lower Face Synthesis Evaluation

Figure 10 shows comparison of our synthesis algorithm to the classic AAM approach [Cootes et al. 2001] that is prevalent in visual speech synthesis, and to a recent detail-enhancing Laplacian pyramid technique [Suwajanakorn et al. 2015]. We observe that both AAM and [Suwajanakorn et al. 2015] show significant blurriness. The blurriness appears due to use of data captured in uncontrolled and uncalibrated conditions, i.e., faces can be non-frontal, under different lighting, etc. The results in the figure are computed using the same set of frontalized face images as used in our algorithm, and even if lighting-invariant optical flow (i.e., collection flow [Kemelmacher-Shlizerman and Seitz 2012] is performed as in [Suwajanakorn et al. 2015]), the resulted synthesis is blurry in the teeth area due to fine misalignment across photos.

Figure 10 also compares weighted mean, median and mode, for generating facial texture from the same set of image candidates. Mean produces the blurriest result among all three, and mode appears noisy even when the sparse sampling (i.e., low number of candidates) is handled by using larger frequency bins or counting by merging nearby points in color space. This behavior can also be understood in an optimization framework where mean, median, and mode correspond to a minimization with L^2, L^1, L^0 norms, respectively. In the case of mode, the summation of the quasi-convex L^0 has multiple minima and is sensitive to image noise and misalignment, whereas L^2 produces an over-smooth average. Median strikes a good balance between these two, which we've seen in practice translates to better edge-preserving properties than mean and is less noisy than mode.

In Figure 11 we compare to the recent Face2face algorithm [Thies et al. 2016]. We provide the same source video (a weekly presidential address) to both methods. Note that we use only the source audio as input, whereas their technique requires the source video–i.e., they effectively have access to the ground truth mouth appearance. In addition, we provide the same target video (a different weekly presidential address) to both methods. The Face2face were produced by the authors running their original system on our videos. The differences between the two methods are best viewed in the supplementary video. Some observations: our method tends to produce more realistic and *Obama-like* lip and head motion. Additionally, our method captures time-varying wrinkles and creases around the mouth whereas Face2face's texture for the surrounding skin appears more static. The teeth produced by Face2face sometimes appear non-rigid with occasional ghosting artifacts, whereas teeth produced by our method appear rigid and temporally smoother. The differences perhaps expected, since our system is trained on many videos of Obama and tuned to produce high quality realistic results

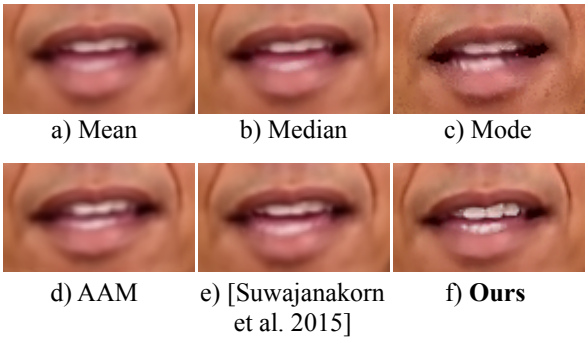


Fig. 10. Mouth synthesis comparison to weighted-mean (a), weighted-median (b), weighted-mode (c), AAM-based techniques (d), and [Suwajanakorn et al. 2015] (e). For all results shown here, we first frontalize all training images using the same frontalization technique as our result, and for (a,b,c,e), we use identical weights to ours computed from Equation 11. Notice how other techniques produce blurry results on our training dataset that contains mouth images from a real speech with natural head motion.

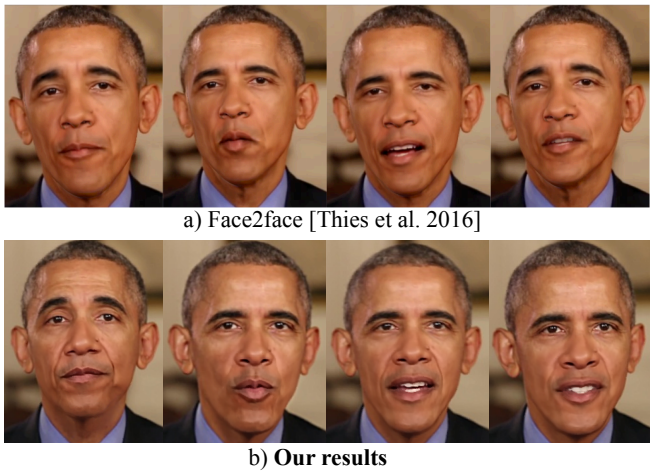


Fig. 11. Comparison to Face2face [Thies et al. 2016] for four different utterances in the same speech using the same source video. Note that [Thies et al. 2016] requires the video of the input speech to drive the animation and focuses on the real-time puppetry application whereas ours aims to synthesize a visual speech given only a recorded audio of Obama. Notice how our method can synthesize more realistic mouths with natural creases around the mouth. The differences between the two approaches are best viewed in the supplementary video.

while Face2face is aimed at a real-time puppetry scenario that uses only two videos (source and target).

In addition to the above comparisons, we compiled a 4 minute video showing related face animation results of the last 20 years. The end of our supplementary video includes this compilation, as a helpful visual reference for our work.

4.3.1 Distribution of candidate frames: We investigate the distribution of the candidate frames selected by our texture synthesis algorithm. We evaluate this on Video A: the average standard deviation of the frame indices of 100 image candidates used to produce one output texture is 931.8 frames (spanning across 31 seconds) with standard deviation 88.5 (3 seconds). The span (or the min and max) of the frame indices are always at least 2,448 frames apart (3,648 total).

4.4 Target Video Re-timing Evaluation

For all results shown, we use the following fixed parameters: The blinking detector was implemented by first normalizing eye landmarks by making the distance between left and right eyes a unit distance. Then blinking is detected if the average eye area is less than 0.025. For indicating silence, we use a threshold of 0.001 on the average audio volume within a window. The resulting binary flags with 0-gap less than 0.5 seconds are connected, and then dilated with kernel of size 0.1 seconds. The dilation and erosion kernels for removing salt and pepper noise for blinking are 3, 3 frames respectively.

In the supplementary video (Video D) we show results generated with and without re-timing (Section 3.3). Since our dynamic programming approach also solves for the best starting frame of the target video, for comparison we start both target videos at this same frame. Without retiming, there are occasional head motion and expression changes during vocal pauses which appears unnatural, as indicated by the red arrow. We also evaluate consistency of the re-timing approach across different target videos (Video E in sup. video), by generating output videos with four different targets. Most of the time, all four results start moving when Obama starts a new sentence and stop during long pauses. Occasionally, Obama moves slightly during small pauses, but fast motion during pauses is avoided.

4.5 Results & Applications

All of the result figures and supplementary videos are generated from input speeches that are not in the training or validation sets. In Figure 12, we run on the audio of Obama weekly address "Standing with Orlando" from YouTube with ID[3] nIxM8rL5GVE and use four different target videos also from weekly addresses (E3gfMumXCjI, 3vPdtajOJfw, 25GOnaY8ZCY, k4OZOTaf3lk). Fully generated videos for each of the targets are presented in supplementary Video E. Our results are realistic with convincing, natural head motion. When compared to prior techniques (Video I), our mouth texture looks sharper than those that rely on AAM and more temporally smooth than those that use frames from original footage. The upper teeth look sharp and realistic. We do note that the lower teeth and tongue appear blurrier and less plausible when compared to the ground-truth footage, e.g., in Video A. In Figure 13 and Video F, we compare our synthesized textures to the original video of the input speech. In Video J, we show the pixel difference map (mean of absolute RGB differences) between the groundtruth video of the input audio and our result using the same input video as the target video (no retiming). Our results show good agreement in terms of the mouth

[3]YouTube video can be accessed by https://www.youtube.com/watch?v=ID given video ID.

Fig. 12. Results for the same input speech using four different target videos. Results along each column are generated from the same utterance.

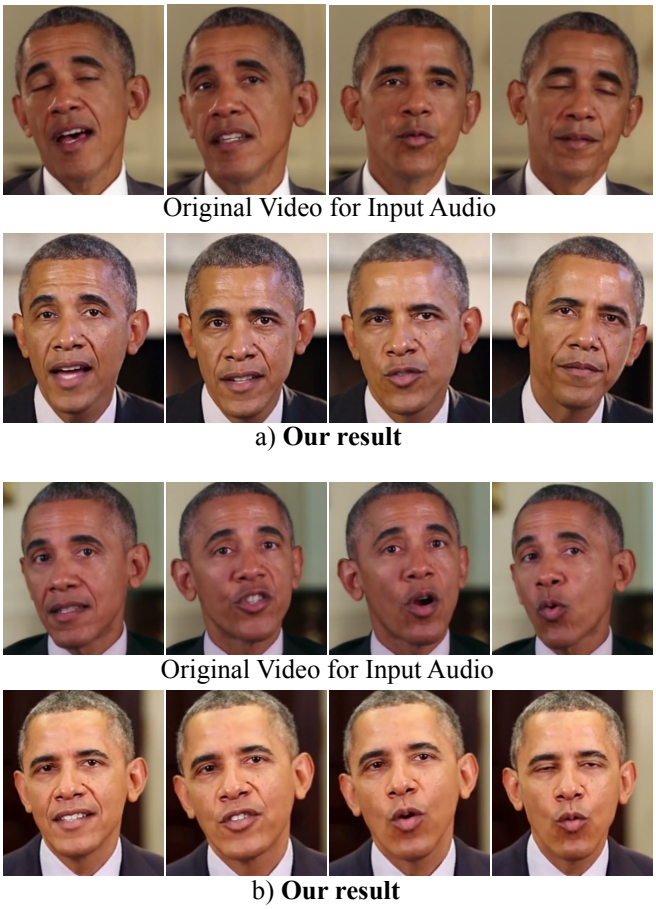


Fig. 13. Comparison of our mouth shapes to the ground-truth footage of the input audio (note good agreement–more results in supplemental video). a) is a weekly address on climate change (cNVzN62l0Yg), and b) is on health care (deF-f0OqvQ4).

shapes, timing, and overall mouth animation, but the mouth region appears less detailed and smoother compared to the groundtruth.

To evaluate generalization to other types of input speech not from the weekly addresses on which we trained, we ran our system on the audio of Obama's interview with Steve Harvey (qMlLjFPCO4M), 60 Minutes (F8MxP9adPO8), and the View (Hdn1iX1a528). Our method can handle casual speech and generates plausible mouth animation even during a mumble or hesitation. We also test on the voice of a quarter century younger Obama from 1990 (7XGi3FGVmA0) and on a voice impressionist (vSAA5GH6OFg). The lipsync quality still looks reasonable although it degrades somewhat as the voice deviates from our training audio.

One useful application of our method is speech summarization, inspired by [Berthouzoz et al. 2012]. I.e., given a long address speech, create a short summary version by manually selecting the desired sections from transcribed text. Our method can then be used to generate a seamless video for the summarized speech shown in Video G. While our result looks similar to [Berthouzoz et al. 2012] for this particular example, an advantage of our system is that we can produce visually seamless cuts even when the head position, lighting, or background has changed between concatenated source footage clips.

4.6 Failure Cases & Limitations

Below are some limitations of our method.

3D geometry errors: During the final composite, our mouth texture is composited over a target frame that may have a different mouth shape and chin location. Usually, our optical flow approach successfully aligns the two chins, but occasionally fails, e.g., when the chin occludes part of his shirt, and produces a double chin artifact shown in Figure 14a. Addressing this problem requires properly modeling the occluded shirt regions. Similarly, when the target pose is non-frontal, imperfect 3D face geometry can cause the mouth texture to be composited outside the face and onto the background (Figure 14b). Even though our method can cope with the reasonable range of head poses presented in the weekly address-style speech, our imperfect head geometry model prevents us from rendering onto a target video with extreme poses such as profiles.

Target video length: Our face texture synthesis approach relies on a full set of mouth shapes being available in the target video,

a) Double chins b) Color bleeding

Fig. 14. a) shows a double chin artifact. This happens when the chin of the target video is lower than our synthesized chin and occludes part of the shirt. b) shows a mouth texture bleeding onto the background.

enough to span the mouth shapes needed for the source audio. This restriction may limit the types and length of target video we can use.

Emotion modeling: Our method does not explicitly model emotions or predict the sentiment of the input speech. Thus Obama's facial expressions in the final output video can appear too serious for a casual speech, or too happy for a serious speech. Some people feel, for example, that our synthesized rendition of Obama's *Standing with Orlando* speech at the beginning of the supplementary video looks too upbeat.

Tongue modeling: Our mouth texture synthesis assumes that the mouth texture can be fully determined by positions of lip fiducials. This may not be entirely true for some sounds such as 'th' that require the use of the tongue, which may be hard to distinguish based purely on lip fiducials.

5 DISCUSSION & FUTURE WORK

We show that by training on a large amount of video of the same person, and designing algorithms with the goal of photorealism in mind, we can create believable video from audio with convincing lip sync. This work opens up a number of interesting future directions, some of which we describe below.

Our pipeline includes one manual step that the user must perform for each target video: selecting and masking a teeth proxy. We believe this step could be automated by training a teeth detector (looking for a large, clear white teeth image)

Our method relies on MFCC audio features, which are not designed specifically for visual synthesis. This suggests that an even more end-to-end network may be able to achieve even better quality by going directly from raw audio waveforms to mouth shapes or textures. For example, [van den Oord et al. 2016] achieves better performance in natural audio generation by replacing MFCC and RNN with dilated convolution. It would be interesting to see how such a network could be applied to our audiovisual synthesis task. Similarly, it would be interesting to see if the network could learn to predict emotional state from audio to produce corresponding visuals (e.g., happy, sad, angry speech, etc.).

Training our system on another person, such as a non-celebrity, can be quite challenging due to the difficulty of obtaining hours of training data. However, the association between mouth shapes and utterances may be, to some extent, speaker-independent. Perhaps a network trained on Obama could be retrained for another person with much less additional training data. Going a step further, perhaps a single universal network could be trained from videos of many different people, and then conditioned on individual speakers, e.g., by giving it a small video sample of the new person, to produce accurate mouth shapes for that person.

While we synthesize only the region around the mouth and borrow the rest of Obama from a target video, a more flexible system would synthesize more of Obama's face and body, and perhaps the background as well. Such a system could enable generating arbitrary length sequences, with much more control of how he moves and acts.

ACKNOWLEDGMENTS

We thank Samsung, Google, Intel, and the University of Washington Animation Research Labs for supporting this research.

REFERENCES

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

Robert Anderson, Björn Stenger, Vincent Wan, and Roberto Cipolla. 2013a. An expressive text-driven 3D talking head. In *ACM SIGGRAPH 2013 Posters*. ACM, 80.

Robert Anderson, Bjorn Stenger, Vincent Wan, and Roberto Cipolla. 2013b. Expressive visual text-to-speech using active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3382–3389.

Fabrice Bellard, M Niedermayer, and others. 2012. FFmpeg. *Availabel from: http://ffmpeg.org* (2012).

Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for placing cuts and transitions in interview video. *ACM Trans. Graph.* 31, 4 (2012), 67–1.

G. Bradski. 2000. *Dr. Dobb's Journal of Software Tools* (2000).

Matthew Brand. 1999. Voice Puppetry. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 21–28. DOI:https://doi.org/10.1145/311535.311537

Christoph Bregler, Michele Covell, and Malcolm Slaney. 1997. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 353–360.

Peter J Burt and Edward H Adelson. 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)* 2, 4 (1983), 217–236.

Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. 2016. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 126.

Yong Cao, Wen C Tien, Petros Faloutsos, and Frédéric Pighin. 2005. Expressive speech-driven facial animation. *ACM Transactions on Graphics (TOG)* 24, 4 (2005), 1283–1302.

Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, and others. 2001. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence* 23, 6 (2001), 681–685.

Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. 2011. Video face replacement. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 130.

Tony Ezzat, Gadi Geiger, and Tomaso Poggio. 2002. *Trainable videorealistic speech animation*. Vol. 21. ACM.

Bo Fan, Lijuan Wang, Frank K Soong, and Lei Xie. 2015a. Photo-real talking head with deep bidirectional LSTM. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4884–4888.

Bo Fan, Lei Xie, Shan Yang, Lijuan Wang, and Frank K Soong. 2015b. A deep bidirectional LSTM approach for video-realistic talking head. *Multimedia Tools and Applications* (2015), 1–23.

Shengli Fu, Ricardo Gutierrez-Osuna, Anna Esposito, Praveen K Kakumanu, and Oscar N Garcia. 2005. Audio/visual mapping with cross-modal hidden Markov models. *IEEE*

Transactions on Multimedia 7, 2 (2005), 243–252.
Yarin Gal. 2015. A theoretically grounded application of dropout in recurrent neural networks. *arXiv preprint arXiv:1512.05287* (2015).
Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormahlen, Patrick Perez, and Christian Theobalt. 2014. Automatic face reenactment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4217–4224.
Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kiran Varanasi, Patrick Perez, and Christian Theobalt. 2015. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 193–204.
Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).
Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 273–278.
Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5 (2005), 602–610.
Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
Masahide Kawai, Tomoyori Iwao, Daisuke Mima, Akinobu Maejima, and Shigeo Morishima. 2014. Data-driven speech animation synthesis focusing on realistic inside of the mouth. *Journal of information processing* 22, 2 (2014), 401–409.
Ira Kemelmacher-Shlizerman and Steven M Seitz. 2012. Collection flow. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 1792–1799.
Taehwan Kim, Yisong Yue, Sarah Taylor, and Iain Matthews. 2015. A decision tree framework for spatiotemporal sequence prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 577–586.
Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.
Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
Kai Li, Feng Xu, Jue Wang, Qionghai Dai, and Yebin Liu. 2012. A data-driven approach for facial expression synthesis in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 57–64.
Shu Liang, Linda G Shapiro, and Ira Kemelmacher-Shlizerman. 2016. Head reconstruction from internet photos. In *European Conference on Computer Vision*. Springer, 360–374.
Ce Liu, William T Freeman, Edward H Adelson, and Yair Weiss. 2008. Human-assisted motion annotation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 1–8.
Wesley Mattheyses, Lukas Latacz, and Werner Verhelst. 2013. Comprehensive many-to-many phoneme-to-viseme mapping and its application for concatenative visual speech synthesis. *Speech Communication* 55, 7 (2013), 857–876.
Wesley Mattheyses and Werner Verhelst. 2015. Audiovisual speech synthesis: An overview of the state-of-the-art. *Speech Communication* 66 (2015), 182–217.
Aude Oliva, Antonio Torralba, and Philippe G. Schyns. 2006. Hybrid Images. *ACM Trans. Graph.* 25, 3 (July 2006), 527–532. DOI: https://doi.org/10.1145/1141911.1141919
Wener Robitza. 2016. ffmpeg-normalize. https://github.com/slhck/ffmpeg-normalize. (2016).
Shunsuke Saito, Tianye Li, and Hao Li. 2016. Real-Time Facial Segmentation and Performance Capture from RGB Input. *arXiv preprint arXiv:1604.02647* (2016).
Shinji Sako, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 2000. HMM-based text-to-audio-visual speech synthesis.. In *INTERSPEECH*. 25–28.
YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. 2014. Style transfer for headshot portraits. (2014).
Taiki Shimba, Ryuhei Sakurai, Hirotake Yamazoe, and Joo-Ho Lee. 2015. Talking heads synthesis from audio with deep neural networks. In *2015 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 100–105.
Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M Seitz. 2014. Total moving face reconstruction. In *European Conference on Computer Vision*. Springer, 796–812.
Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2015. What Makes Tom Hanks Look Like Tom Hanks. In *Proceedings of the IEEE International Conference on Computer Vision*. 3952–3960.
Sarah Taylor, Akihiro Kato, Ben Milner, and Iain Matthews. 2016. Audio-to-Visual Speech Conversion using Deep Neural Networks. (2016).
Sarah L Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. 2012. Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*. Eurographics Association, 275–284.
Alexandru Telea. 2004. An image inpainting technique based on the fast marching method. *Journal of graphics tools* 9, 1 (2004), 23–34.
Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 183.
Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* 1 (2016).
Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. 2005. Face transfer with multilinear models. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 426–433.
Lijuan Wang, Wei Han, and Frank K Soong. 2012. High quality lip-sync animation for 3D photo-realistic talking head. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4529–4532.
Lijuan Wang, Xiaojun Qian, Wei Han, and Frank K Soong. 2010. Synthesizing photo-real talking head via trajectory-guided sample selection.. In *INTERSPEECH*, Vol. 10. 446–449.
Lei Xie and Zhi-Qiang Liu. 2007a. A coupled HMM approach to video-realistic speech animation. *Pattern Recognition* 40, 8 (2007), 2325–2340.
Lei Xie and Zhi-Qiang Liu. 2007b. Realistic mouth-synching for speech-driven talking face using articulatory modelling. *IEEE Transactions on Multimedia* 9, 3 (2007), 500–510.
Xuehan Xiong and Fernando De la Torre. 2013. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 532–539.
Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014).
Xinjian Zhang, Lijuan Wang, Gang Li, Frank Seide, and Frank K Soong. 2013. A new language independent, photo-realistic talking head driven by voice only.. In *INTERSPEECH*. 2743–2747.