Text-based Editing of Talking-head Video

OHAD FRIED, Stanford University
AYUSH TEWARI, Max Planck Institute for Informatics
MICHAEL ZOLLHÖFER, Stanford University
ADAM FINKELSTEIN, Princeton University
ELI SHECHTMAN, Adobe
DAN B GOLDMAN
KYLE GENOVA, Princeton University
ZEYU JIN, Adobe
CHRISTIAN THEOBALT, Max Planck Institute for Informatics
MANEESH AGRAWALA, Stanford University

Fig. 1. We propose a novel text-based editing approach for talking-head video. Given an edited transcript, our approach produces a realistic output video in which the dialogue of the speaker has been modified and the resulting video maintains a seamless audio-visual flow (i.e. no jump cuts).

Editing talking-head video to change the speech content or to remove filler words is challenging. We propose a novel method to edit talking-head video based on its transcript to produce a realistic output video in which the dialogue of the speaker has been modified, while maintaining a seamless audio-visual flow (i.e. no jump cuts). Our method automatically annotates an input talking-head video with phonemes, visemes, 3D face pose and geometry, reflectance, expression and scene illumination per frame. To edit a video, the user has to only edit the transcript, and an optimization strategy then chooses segments of the input corpus as base material. The annotated parameters corresponding to the selected segments are seamlessly stitched together and used to produce an intermediate video representation in which the lower half of the face is rendered with a parametric face model. Finally, a recurrent video generation network transforms this representation to a photorealistic video that matches the edited transcript. We demonstrate a large variety of edits, such as the addition, removal, and alteration of words, as well as convincing language translation and full sentence synthesis.

CCS Concepts: • **Information systems** → *Video search*; *Speech / audio search*; • **Computing methodologies** → *Computational photography*; *Reconstruction*; *Motion processing*; *Graphics systems and interfaces.*

Additional Key Words and Phrases: Text-based video editing, talking heads, visemes, dubbing, face tracking, face parameterization, neural rendering.

ACM Reference Format:
Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. 2019. Text-based Editing of Talking-head Video. *ACM Trans. Graph.* 38, 4, Article 68 (July 2019), 14 pages. https://doi.org/10.1145/3306346.3323028

1 INTRODUCTION

Talking-head video – framed to focus on the face and upper body of a speaker – is ubiquitous in movies, TV shows, commercials, YouTube video logs, and online lectures. Editing such pre-recorded video is challenging, but can be needed to emphasize particular content, remove filler words, correct mistakes, or more generally match the editor's intent. Using current video editing tools, like Adobe Premiere, skilled editors typically scrub through raw video footage to find relevant segments and assemble them into the desired story. They must carefully consider where to place cuts so as to minimize disruptions to the overall audio-visual flow.

Authors' addresses: Ohad Fried, Stanford University; Ayush Tewari, Max Planck Institute for Informatics; Michael Zollhöfer, Stanford University; Adam Finkelstein, Princeton University; Eli Shechtman, Adobe; Dan B Goldman; Kyle Genova, Princeton University; Zeyu Jin, Adobe; Christian Theobalt, Max Planck Institute for Informatics; Maneesh Agrawala, Stanford University.

0730-0301/2019/7-ART68 $15.00
https://doi.org/10.1145/3306346.3323028

Berthouzoz et al. [2012] introduce a text-based approach for editing such videos. Given an input video, they obtain a time-aligned transcript and allow editors to cut and paste the text to assemble it into the desired story. Their approach can move or delete segments, while generating visually seamless transitions at cut boundaries. However, this method only produces artifact-free results when these boundaries are constrained to certain well-behaved segments of the video (e.g. where the person sits still between phrases or sentences).

Neither conventional editing tools nor the text-based approach allow synthesis of new audio-visual speech content. Thus, some modifications require either re-shooting the footage or overdubbing existing footage with new wording. Both methods are expensive as they require new performances, and overdubbing generally produces mismatches between the visible lip motion and the audio.

This paper presents a method that completes the suite of operations necessary for transcript-based editing of talking-head video. Specifically, based only on text edits, it can synthesize convincing new video of a person speaking, and produce seamless transitions even at challenging cut points such as the middle of an utterance.

Our approach builds on a thread of research for synthesizing realistic talking-head video. The seminal Video Rewrite system of Bregler et al. [1997] and the recent Synthesizing Obama project of Suwajanakorn et al. [2017] take new speech recordings as input, and superimpose the corresponding lip motion over talking-head video. While the latter state-of-the art approach can synthesize fairly accurate lip sync, it has been shown to work for exactly one talking head because it requires *huge* training data (14 hours). This method also relies on input audio from the same voice on which it was trained – from either Obama or a voice impersonator. In contrast our approach works from text and therefore supports applications that require a different voice, such as translation.

Performance-driven puppeteering and dubbing methods, such as VDub [Garrido et al. 2015], Face2Face [Thies et al. 2016] and Deep Video Portraits [Kim et al. 2018b], take a new talking-head performance (usually from a different performer) as input and transfer the lip and head motion to the original talking-head video. Because these methods have access to video as input they can often produce higher-quality synthesis results than the audio-only methods. Nevertheless, capturing new video for this purpose is obviously more onerous than typing new text.

Our method accepts text only as input for synthesis, yet builds on the Deep Video Portraits approach of Kim et al. [2018b] to craft synthetic video. Our approach drives a 3D model by seamlessly stitching different snippets of motion tracked from the original footage. The snippets are selected based on a dynamic programming optimization that searches for sequences of sounds in the transcript that should *look* like the words we want to synthesize, using a novel *viseme*-based similarity measure. These snippets can be re-timed to match the target viseme sequence, and are blended to create a seamless mouth motion. To synthesize output video, we first create a synthetic composite video in which the lower face region is masked out. In cases of inserting new text, we retime the rest of the face and background from the boundaries. The masked out region is composited with a synthetic 3D face model rendering using the mouth motion found earlier by optimization (Figure 5). The composite exhibits the desired motion, but lacks realism due to the incompleteness and imperfections of the 3D face model. For example, facial appearance does not perfectly match, dynamic high-frequency detail is missing, and the mouth interior is absent. Nonetheless, these data are sufficient cues for a new learned recurrent video generation network to be able to convert them to realistic imagery. The new composite representation and the recurrent network formulation significantly extend the neural face translation approach of Kim et al. [2018b] to text-based editing of existing videos.

We show a variety of text-based editing results and favorable comparisons to previous techniques. In a crowd-sourced user study, our edits were rated to be real in 59.6% of cases. The main technical contributions of our approach are:

- A text-based editing tool for talking-head video that lets editors insert new text, in addition to cutting and copy-pasting in an existing transcript.
- A dynamic programming based strategy tailored to video synthesis that assembles new words based on snippets containing sequences of observed visemes in the input video.
- A parameter blending scheme that, when combined with our synthesis pipeline, produces seamless talking heads, even when combining snippets with different pose and expression.
- A recurrent video generation network that converts a composite of real background video and synthetically rendered lower face into a photorealistic video.

1.1 Ethical Considerations

Our text-based editing approach lays the foundation for better editing tools for movie post production. Filmed dialogue scenes often require re-timing or editing based on small script changes, which currently requires tedious manual work. Our editing technique also enables easy adaptation of audio-visual video content to specific target audiences: e.g., instruction videos can be fine-tuned to audiences of different backgrounds, or a storyteller video can be adapted to children of different age groups purely based on textual script edits. In short, our work was developed for storytelling purposes.

However, the availability of such technology — at a quality that some might find indistinguishable from source material — also raises important and valid concerns about the potential for misuse. Although methods for image and video manipulation are as old as the media themselves, the risks of abuse are heightened when applied to a mode of communication that is sometimes considered to be authoritative evidence of thoughts and intents. We acknowledge that bad actors might use such technologies to falsify personal statements and slander prominent individuals. We are concerned about such deception and misuse.

Therefore, we believe it is critical that video synthesized using our tool clearly presents itself as synthetic. The fact that the video is synthesized may be obvious by context (e.g. if the audience understands they are watching a fictional movie), directly stated in the video or signaled via watermarking. We also believe that it is essential to obtain permission from the performers for any alteration before sharing a resulting video with a broad audience. Finally, it is important that we as a community continue to develop forensics, fingerprinting and verification techniques (digital and non-digital) to identify manipulated video. Such safeguarding measures would

reduce the potential for misuse while allowing creative uses of video editing technologies like ours.

We hope that publication of the technical details of such systems can spread awareness and knowledge regarding their inner workings, sparking and enabling associated research into the aforementioned forgery detection, watermarking and verification systems. Finally, we believe that a robust public conversation is necessary to create a set of appropriate regulations and laws that would balance the risks of misuse of these tools against the importance of creative, consensual use cases.

2 RELATED WORK

Facial Reenactment. Facial video reenactment has been an active area of research [Averbuch-Elor et al. 2017; Garrido et al. 2014; Kemelmacher-Shlizerman et al. 2010; Li et al. 2014; Liu et al. 2001; Suwajanakorn et al. 2017; Vlasic et al. 2005]. Thies et al. [2016] recently demonstrated real-time video reenactment. Deep video portraits [Kim et al. 2018b] enables full control of the head pose, expression, and eye gaze of a target actor based on recent advances in learning-based image-to-image translation [Isola et al. 2017]. Some recent approaches enable the synthesis of controllable facial animations from single images [Averbuch-Elor et al. 2017; Geng et al. 2018; Wiles et al. 2018]. Nagano et al. [2018] recently showed how to estimate a controllable avatar of a person from a single image. We employ a facial reenactment approach for visualizing our text-based editing results and show how facial reenactment can be tackled by neural face rendering.

Visual Dubbing. Facial reenactment is the basis for visual dubbing, since it allows to alter the expression of a target actor to match the motion of a dubbing actor that speaks in a different language. Some dubbing approaches are speech-driven [Bregler et al. 1997; Chang and Ezzat 2005; Ezzat et al. 2002; Liu and Ostermann 2011] others are performance-driven [Garrido et al. 2015]. Speech-driven approaches have been shown to produce accurate lip-synced video [Suwajanakorn et al. 2017]. While this approach can synthesize fairly accurate lip-synced video, it requires the new audio to sound similar to the original speaker, while we enable synthesis of new video using text-based edits. Mattheyses et al. [2010] show results with no head motion, in a controlled setup with uniform background. In contrast, our 3D based approach and neural renderer can produce subtle phenomena such as lip rolling, and works in a more general setting.

Speech animation for rigged models. Several related methods produce animation curves for speech [Edwards et al. 2016; Taylor et al. 2017; Zhou et al. 2018]. They are specifically designed for animated 3D models and not for photorealistic video, requiring a character rig and artist supplied rig correspondence. In contrast, our approach "animates" a real person speaking, based just on text and a monocular recording of the subject.

Text-Based Video and Audio Editing. Researchers have developed a variety of audio and video editing tools based on time-aligned transcripts. These tools allow editors to shorten and rearrange speech for audio podcasts [Rubin et al. 2013; Shin et al. 2016], annotate video with review feedback [Pavel et al. 2016], provide audio descriptions of the video content for segmentation of B-roll footage [Truong et al. 2016] and generate structured summaries of lecture videos [Pavel et al. 2014]. Leake et al. [2017] use the structure imposed by time-aligned transcripts to automatically edit together multiple takes of a scripted scene based on higher-level cinematic idioms specified by the editor. Berthouzoz et al.'s [2012] tool for editing interview-style talking-head video by cutting, copying and pasting transcript text is closest to our work. While we similarly enable rearranging video by cutting, copying and pasting text, unlike all of the previous text-based editing tools, we allow synthesis of new video by simply typing the new text into the transcript.

Audio Synthesis. In transcript-based video editing, synthesizing new video clips would often naturally be accompanied by audio synthesis. Our approach to video is independent of the audio, and therefore a variety of *text to speech* (TTS) methods can be used. Traditional TTS has explored two general approaches: *parametric methods* (e.g. [Zen et al. 2009]) generate acoustic features based on text, and then synthesize a waveform from these features. Due to oversimplified acoustic models, they tend to sound robotic. In contrast, *unit selection* is a data driven approach that constructs new waveforms by stitching together small pieces of audio (or *units*) found elsewhere in the transcript [Hunt and Black 1996]. Inspired by the latter, the VoCo project of Jin et al. [2017] performs a search in the existing recording to find short ranges of audio that can be stitched together such that they blend seamlessly in the context around an insertion point. Section 4 and the accompanying video present a few examples of using our method to synthesize new words in video, coupled with the use of VoCo to synthesize corresponding audio. Current state-of-the-art TTS approaches rely on deep learning [Shen et al. 2018; Van Den Oord et al. 2016]. However, these methods require a huge (tens of hours) training corpus for the target speaker.

Deep Generative Models. Very recently, researchers have proposed *Deep Generative Adversarial Networks (GANs)* for the synthesis of images and videos. Approaches create new images from scratch [Chen and Koltun 2017; Goodfellow et al. 2014; Karras et al. 2018; Radford et al. 2016; Wang et al. 2018b] or condition the synthesis on an input image [Isola et al. 2017; Mirza and Osindero 2014]. High-resolution conditional video synthesis [Wang et al. 2018a] has recently been demonstrated. Besides approaches that require a paired training corpus, unpaired video-to-video translation techniques [Bansal et al. 2018] only require two training videos. Video-to-video translation has been used in many applications. For example, impressive results have been shown for the reenactment of the human head [Olszewski et al. 2017], head and upper body [Kim et al. 2018b], and the whole human body [Chan et al. 2018; Liu et al. 2018].

Monocular 3D Face Reconstruction. There is a large body of work on reconstructing facial geometry and appearance from a single image using optimization methods [Fyffe et al. 2014; Garrido et al. 2016; Ichim et al. 2015; Kemelmacher-Shlizerman 2013; Roth et al. 2017; Shi et al. 2014; Suwajanakorn et al. 2017; Thies et al. 2016]. Many of these techniques employ a parametric face model [Blanz et al. 2004; Blanz and Vetter 1999; Booth et al. 2018] as a prior to better constrain the reconstruction problem. Recently, deep learning-based approaches have been proposed that train a convolutional network

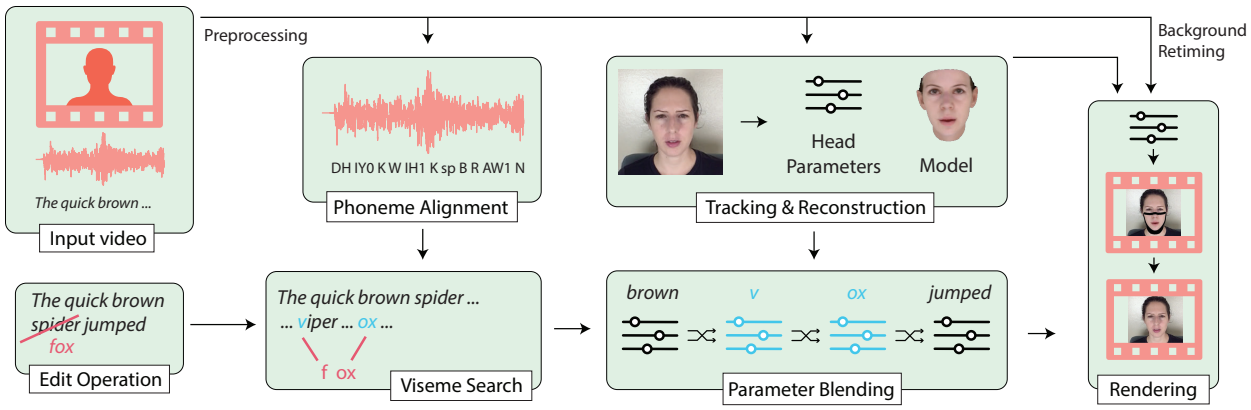


Fig. 2. Method overview. Given an input talking-head video and a transcript, we perform text-based editing. We first align phonemes to the input audio and track each input frame to construct a parametric head model. Then, for a given edit operation (changing *spider* to *fox*), we find segments of the input video that have similar visemes to the new word. In the above case we use ***v**iper* and ***ox*** to construct *fox*. We use blended head parameters from the corresponding video frames, together with a retimed background sequence, to generate a composite image, which is used to generate a photorealistic frame using our neural face rendering method. In the resulting video, the actress appears to be saying *fox*, even though that word was never spoken by her in the original recording.

to directly regress the model parameters [Dou et al. 2017; Genova et al. 2018; Richardson et al. 2016; Tewari et al. 2018a, 2017; Tran et al. 2017]. Besides model parameters, other approaches regress detailed depth maps [Richardson et al. 2017; Sela et al. 2017], or 3D displacements [Cao et al. 2015; Guo et al. 2018; Tewari et al. 2018b]. Face reconstruction is the basis for a large variety of applications, such as facial reenactment and visual dubbing. For more details on monocular 3D face reconstruction, we refer to Zollhöfer et al. [2018].

3 METHOD

Our system takes as input a video recording of a talking head with a transcript of the speech and any number of edit operations specified on the transcript. Our tool supports three types of edit operations;

- ***Add new words:*** the edit adds one or more consecutive words at a point in the video (e.g. because the actor skipped a word or the producer wants to insert a phrase).
- ***Rearrange existing words:*** the edit moves one or more consecutive words that exist in the video (e.g. for better word ordering without introducing jump cuts).
- ***Delete existing words:*** the edit removes one or more consecutive words from the video (e.g. for simplification of wording and removing filler such as "um" or "uh").

We represent editing operations by the sequence of words $\mathcal{W}$ in the edited region as well as the correspondence between those words and the original transcript. For example, deleting the word "wonderful" in the sequence "hello wonderful world" is specified as ('hello', 'world') and adding the word "big" is specified as ('hello', 'big', 'world').

Our system processes these inputs in five main stages (Figure 2). In the phoneme alignment stage (Section 3.1) we align the transcript to the video at the level of phonemes and then in the tracking and reconstruction stage (Section 3.2) we register a 3D parametric head model with the video. These are pre-processing steps performed once per input video. Then for each edit operation $\mathcal{W}$ we first perform a viseme search (Section 3.3) to find the best visual match between the subsequences of phonemes in the edit and subsequences of phonemes in the input video. We also extract a region around the edit location to act as a background sequence, from which we will extract background pixels and pose data. For each subsequence we blend the parameters of the tracked 3D head model (Section 3.4) and then use the resulting parameter blended animation of the 3D head, together with the background pixels, to render a realistic full-frame video (Section 3.5) in which the subject appears to say the edited sequence of words. Our viseme search and approach for combining shorter subsequences with parameter blending is motivated by the phoneme/viseme distribution of the English language (Appendix A).

3.1 Phoneme Alignment

Phonemes are perceptually distinct units that distinguish one word from another in a specific language. Our method relies on phonemes to find snippets in the video that we later combine to produce new content. Thus, our first step is to compute the *identity* and *timing* of phonemes in the input video. To segment the video's speech audio into phones (audible realizations of phonemes), we assume we have an accurate text transcript and align it to the audio using P2FA [Rubin et al. 2013; Yuan and Liberman 2008], a phoneme-based alignment tool. This gives us an ordered sequence $\mathrm{V} = (\mathrm{v}_1, \ldots, \mathrm{v}_n)$ of phonemes, each with a label denoting the phoneme name, start time, and end time $\mathrm{v}_i = (\mathrm{v}_i^{lbl}, \mathrm{v}_i^{in}, \mathrm{v}_i^{out})$. Note that if a transcript is not given as part of the input, we can use automatic speech transcription tools [IBM 2016; Ochshorn and Hawkins 2016] or crowdsourcing transcription services like rev.com to obtain it.

3.2 3D Face Tracking and Reconstruction

We register a 3D parametric face model with each frame of the input talking-head video. The parameters of the model (e.g. expression, head pose, etc.) will later allow us to selectively blend different aspects of the face (e.g. take the expression from one frame and pose

from another). Specifically, we apply recent work on monocular model-based face reconstruction [Garrido et al. 2016; Thies et al. 2016]. These techniques parameterize the rigid head pose $\mathbf{T} \in SE(3)$, the facial geometry $\boldsymbol{\alpha} \in \mathbb{R}^{80}$, facial reflectance $\boldsymbol{\beta} \in \mathbb{R}^{80}$, facial expression $\boldsymbol{\delta} \in \mathbb{R}^{64}$, and scene illumination $\boldsymbol{\gamma} \in \mathbb{R}^{27}$. Model fitting is based on the minimization of a non-linear reconstruction energy. For more details on the minimization, please see the papers of Garrido et al. [2016] and Thies et al. [2016]. In total, we obtain a 257 parameter vector $\mathbf{p} \in \mathbb{R}^{257}$ for each frame of the input video.

3.3 Viseme Search

Given an edit operation specified as a sequence of words $\mathcal{W}$, our goal is to find matching sequences of phonemes in the video that can be combined to produce $\mathcal{W}$. In the matching procedure we use the fact that identical phonemes are expected to be, on average, more visually similar to each other than non-identical phonemes (despite co-articulation effects). We similarly consider visemes, groups of aurally distinct phonemes that appear visually similar to one another (Section 3.3), as good potential matches. Importantly, the matching procedure *cannot* expect to find a good coherent viseme sequence in the video for long words or sequences in the edit operation. Instead, we must find several matching subsequences and a way to best combine them.

We first convert the edit operation $\mathcal{W}$ to a phoneme sequence $\mathrm{W} = (\mathrm{w}_1, \ldots, \mathrm{w}_m)$ where each w_i is defined as $(\mathrm{w}_i^{lbl}, \mathrm{w}_i^{in}, \mathrm{w}_i^{out})$ similar to our definition of phonemes in the video v_i. We can convert the text $\mathcal{W}$ to phoneme labels w_i^{lbl} using a word to phoneme map, but text does not contain timing information $\mathrm{w}_i^{in}, \mathrm{w}_i^{out}$. To obtain timings we use a text-to-speech synthesizer to convert the edit into speech. For all results in this paper we use either the built-in speech synthesizer in Mac OS X, or Voco [Jin et al. 2017]. Note however that our video synthesis pipeline does not use the audio *signal*, but only its *timing*. So, e.g., manually specified phone lengths could be used as an alternative. The video generated in the rendering stage of our pipeline (Section 3.5) is mute and we discuss how we can add audio at the end of that section. Given the audio of $\mathcal{W}$, we produce phoneme labels and timing using P2FA, in a manner similar to the one we used in Section 3.1.

Given an edit W and the video phonemes V, we are looking for the optimal partition of W into sequential subsequences $\mathrm{W}_1, \ldots, \mathrm{W}_k$, such that each subsequence has a good match in V, while encouraging subsequences to be long (Figure 4). We are looking for long subsequences because each transition between subsequences may cause artifacts in later stages. We first describe matching one subsequence $\mathrm{W}_i = (\mathrm{w}_j, \ldots, \mathrm{w}_{j+k})$ to the recording V, and then explain how we match the full query W.

Matching one subsequence. We define $\mathrm{C}_{\mathrm{match}}(\mathrm{W}_i, \mathrm{V}_\star)$ between a subsequence of the query W_i and some subsequence of the video $\mathrm{V}_\star$ as a modified Levenshtein edit distance [Levenshtein 1966] between phoneme sequences that takes phoneme length into account. The edit distance requires pre-defined costs for insertion, deletion and swap. We define our insertion cost $\mathrm{C}_{\mathrm{insert}} = 1$ and deletion cost $\mathrm{C}_{\mathrm{delete}} = 1$ and consider viseme and phoneme labels as well as phoneme lengths in our swap cost

$$\mathrm{C}_{\mathrm{swap}}(\mathrm{v}_i, \mathrm{w}_j) = \mathrm{C}_{\mathrm{vis}}(\mathrm{v}_i, \mathrm{w}_j)(|\,\mathrm{v}_i\,| + |\,\mathrm{w}_j\,|) + \chi\big||\,\mathrm{v}_i\,| - |\,\mathrm{w}_j\,|\big| \quad (1)$$

where $|a|$ denotes the length of phoneme a, $\mathrm{C}_{\mathrm{vis}}(\mathrm{v}_i, \mathrm{w}_j)$ is 0 if v_i and w_j are the same phoneme, 0.5 if they are different phonemes but the same viseme (Section 3.3), and 1 if they are different visemes. The parameter χ controls the influence of length difference on the cost, and we set it to 10^{-4} in all our examples. Equation (1) penalized for different phonemes and visemes, weighted by the sum of the phoneme length. Thus longer non-matching phonemes will incur a larger penalty, as they are more likely to be noticed.

We minimize $\mathrm{C}_{\mathrm{match}}(\mathrm{W}_i, \mathrm{V})$ over all possible $\mathrm{V}_\star$ using dynamic programming [Levenshtein 1966] to find the best suffix of any prefix of V and its matching cost to W_i. We brute-force all possible prefixes of V to find the best match V_i to the query W_i.

Matching the full query. We define our full matching cost C between the query W and the video V as

$$C(\mathrm{W}, \mathrm{V}) = \min_{\substack{(\mathrm{W}_1, \ldots, \mathrm{W}_k) \in \mathrm{split}(\mathrm{W}) \\ (\mathrm{V}_1, \ldots, \mathrm{V}_k)}} \sum_{i=1}^{k} \mathrm{C}_{\mathrm{match}}(\mathrm{W}_i, \mathrm{V}_i) + \mathrm{C}_{\mathrm{len}}(\mathrm{W}_i) \quad (2)$$

where split(W) denotes the set of all possible ways of splitting W into subsequences, and V_i is the best match for W_i according to $\mathrm{C}_{\mathrm{match}}$. The cost $\mathrm{C}_{\mathrm{len}}(\mathrm{W}_i)$ penalizes short subsequences and is defined as

$$\mathrm{C}_{\mathrm{len}}(\mathrm{W}_i) = \frac{\phi}{|\mathrm{W}_i|} \quad (3)$$

where $|\mathrm{W}_i|$ denotes the number of phonemes in subsequence W_i and ϕ is a weight parameter empirically set to 0.001 for all our examples. To minimize Equation (2) we generate all splits $(\mathrm{W}_1, \ldots, \mathrm{W}_k) \in \mathrm{split}(\mathrm{W})$ of the query (which is typically short), and for each W_i we find the best subsequence V_i of V with respect to $\mathrm{C}_{\mathrm{match}}$. Since the same subsequence W_i can appear in multiple partitions, we memoize computations to make sure each match cost is computed only once. The viseme search procedure produces subsequences $(\mathrm{V}_1, \ldots, \mathrm{V}_k)$ of the input video that, when combined, should produce W.

Table 1. Grouping phonemes (listed as ARPABET codes) into visemes. We use the viseme grouping of Annosoft's lipsync tool [Annosoft 2008]. More viseme groups may lead to better visual matches (each group is more specific in its appearance), but require more data because the chance to find a viseme match decreases. We did not perform an extensive evaluation of different viseme groupings, of which there are many.

v01	AA0, AA1, AA2	v09	Y, IY0, IY1, IY2
v02	AH0, AH1, AH2, HH	v10	R, ER0, ER1, ER2
v03	AO0, AO1, AO2	v11	L
v04	AW0, AW1, AW2, OW0, OW1, OW2	v12	W
		v13	M, P, B
v05	OY0, OY1, OY2, UH0, UH1, UH2, UW0, UW1, UW2	v14	N, NG, DH, D, G, T, Z, ZH, TH, K, S
v06	EH0, EH1, EH2, AE0, AE1, AE2	v15	CH, JH, SH
v07	IH0, IH1, IH2, AY0, AY1, AY2	v16	F, V
v08	EY0, EY1, EY2	v17	sp

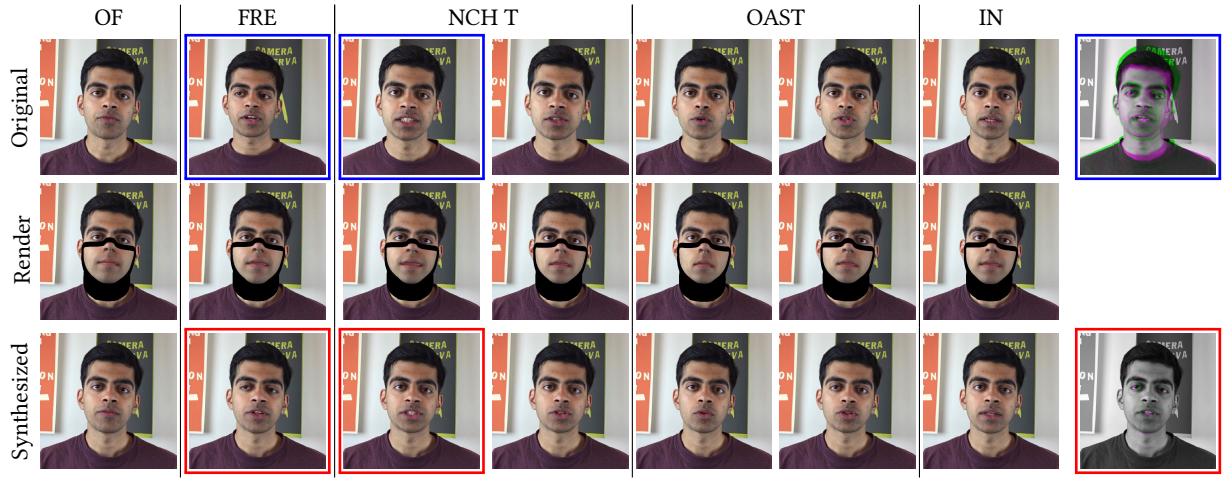


Fig. 3. Our parameter blending strategy produces a seamless synthesized result from choppy original sequences. Above, we insert the expression "french toast" instead of "napalm" in the sentence "I like the smell of napalm in the morning." The new sequence was taken from different parts of the original video: F R EH1 taken from "fresh", N CH T taken from "drenched", and OW1 S T taken from "roast". Notice how original frames from different sub-sequences are different in head size and posture, while our synthesized result is a smooth sequence. On the right we show the pixel difference between blue and red frames; notice how blue frames are very different. *Videos in supplemental material.*

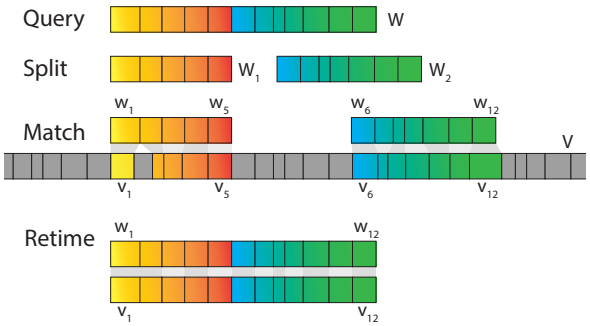


Fig. 4. Viseme search and retiming. Given a query sequence W, We split it into all possible subsequences, of which one $(W_1, W_2) \in \text{split}(W)$ is shown. Each subsequence is matched to the input video V, producing a correspondance between query phonemes w_i and input video phonemes v_i. We retime in parameter space to match the lengths of each v_i to w_i.

3.4 Parameter Retiming & Blending

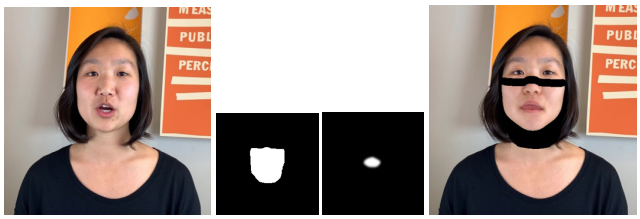
The sequence $(V_1, \ldots, V_k)$ of video subsequences describes sections of the video for us to combine in order to create $\mathcal{W}$. However, we cannot directly use the video frames that correspond to $(V_1, \ldots, V_k)$ for two reasons: (1) A sequence V_i corresponds to part of $\mathcal{W}$ in viseme identity, but not in viseme length, which will produce unnatural videos when combined with the speech audio, and (2) Consecutive sequences V_i and V_{i+1} can be from sections that are far apart in the original video. The subject might look different in these parts due to pose and posture changes, movement of hair, or camera motion. Taken as-is, the transition between consecutive sequences will look unnatural (Figure 3 top).

To solve these issues, we use our parametric face model in order to mix different properties (pose, expression, etc.) from different input frames, and blend them in parameter space. We also select a background sequence $\mathcal{B}$ and use it for pose data and background pixels. The background sequence allows us to edit challenging videos with hair movement and slight camera motion.

Background retiming and pose extraction. An edit operation $\mathcal{W}$ will often change the length of the original video. We take a video sequence (from the input video) $\mathcal{B}'$ around the location of the edit operation, and retime it to account for the change in length the operation will produce, resulting in a retimed background sequence $\mathcal{B}$. We use nearest-neighbor sampling of frames, and select a large enough region around the edit operation so that retiming artifacts are negligible. All edits in this paper use the length of one sentence as background. The retimed sequence $\mathcal{B}$ does not match the original nor the new audio, but can provide realistic background pixels and pose parameters that seamlessly blend into the rest of the video. In a later step we synthesize frames based on the retimed background and expression parameters that *do* match the audio.

Subsequence retiming. The phonemes in each sequence $v_j \in V_i$ approximately match the length of corresponding query phonemes, but an exact match is required so that the audio and video will be properly synchronized. We set a desired frame rate $\mathcal{F}$ for our synthesized video, which often matches the input frame-rate, but does not have to (e.g. to produce slow-mo video from standard video). Given the frame rate $\mathcal{F}$, we sample model parameters $\mathbf{p} \in \mathbb{R}^{257}$ by linearly interpolating adjacent frame parameters described in Section 3.2. For each $v_j \in V_i$ we sample $\mathcal{F}|w_j|$ frame parameters in $[v_j^{in}, v_j^{out}]$ so that the length of the generated video matches the length of the query $|w_j|$. This produces a sequence that matches $\mathcal{W}$ in timing, but with visible jump cuts between sequences if rendered as-is (Figure 4 bottom).

Parameter blending. To avoid jump cuts, we use different strategies for different parameters, as follows. Identity geometry $\boldsymbol{\alpha} \in \mathbb{R}^{80}$ and reflectance $\boldsymbol{\beta} \in \mathbb{R}^{80}$ are kept constant throughout the sequence



(a) Ground Truth ($\mathbf{f}_i$) (b) Face & Mouth (c) Synth. Comp. ($\mathbf{r}_i$)

Fig. 5. Training Corpus: For each ground truth frame $\mathbf{f}_i$ (a), we obtain a 3D face reconstruction. The reconstructed geometry proxy is used to mask out the lower face region (b, left) and render a mouth mask $\mathbf{m}_i$ (b, right), which is used in our training reconstruction loss. We superimpose the lower face region from the parametric face model to obtain a synthetic composite $\mathbf{r}_i$ (c). The goal of our expression-guided neural renderer is to learn a mapping from the synthetic composite $\mathbf{r}_i$ back to the ground truth frame $\mathbf{f}_i$.

(it's always the same person), so they do not require blending. Scene illumination $\boldsymbol{\gamma} \in \mathbb{R}^{27}$ typically changes slowly or is kept constant, thus we linearly interpolate illumination parameters between the last frame prior to the inserted sequence and the first frame after the sequence, disregarding the original illumination parameters of V_i. This produces a realistic result while avoiding light flickering for input videos with changing lights. Rigid head pose $\mathbf{T} \in SE(3)$ is taken directly from the retimed background sequence $\mathcal{B}$. This ensures that the pose of the parameterized head model matches the background pixels in each frame.

Facial expressions $\boldsymbol{\delta} \in \mathbb{R}^{64}$ are the most important parameters for our task, as they hold information about mouth and face movement — the visemes we aim to reproduce. Our goal is to preserve the retrieved expression parameters as much as possible, while smoothing out the transition between them. Our approach is to smooth out each transition from V_i to V_{i+1} by linearly interpolating a region of 67 milliseconds around the transition. We found this length to be short enough so that individual visemes are not lost, and long enough to produce convincing transitions between visemes.

3.5 Neural Face Rendering

We employ a novel neural face rendering approach for synthesizing photo-realistic talking-head video that matches the modified parameter sequence (Section 3.4). The output of the previous processing step is an edited parameter sequence that describes the new desired facial motion and a corresponding retimed background video clip. The goal of this synthesis step is to change the facial motion of the retimed background video to match the parameter sequence. To this end, we first mask out the lower face region, including parts of the neck (for the mask see Figure 5b), in the retimed background video and render a new synthetic lower face with the desired facial expression on top. This results in a video of *composites* $\mathbf{r}_i$ (Figure 5d). Finally, we bridge the domain gap between $\mathbf{r}_i$ and real video footage of the person using our neural face rendering approach, which is based on recent advances in learning-based image-to-image translation [Isola et al. 2017; Sun et al. 2018].

3.5.1 Training the Neural Face Renderer. To train our neural face rendering approach to bridge the domain gap we start from a paired

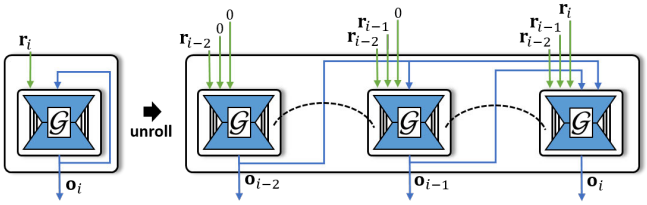


Fig. 6. We assume the video has been generated by a sequential process, which we model by a recurrent network with shared generator $\mathcal{G}$. In practice, we unroll the loop three times.

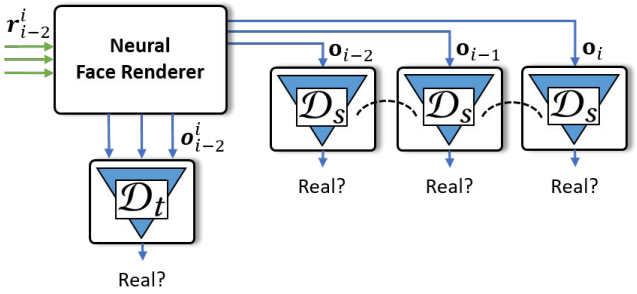


Fig. 7. We employ a spatial discriminator $\mathcal{D}_s$, a temporal discriminator $\mathcal{D}_t$, and an adversarial patch-based discriminator loss to train our neural face rendering network.

training corpus $\mathcal{T} = \{(\mathbf{f}_i, \mathbf{r}_i)\}_{i=1}^{N}$ that consists of the N original video frames $\mathbf{f}_i$ and corresponding synthetic composites $\mathbf{r}_i$. The $\mathbf{r}_i$ are generated as described in the last paragraph, but using the ground truth tracking information of the corresponding frame (Figure 5), instead of the edited sequence, to render the lower face region. The goal is to learn a temporally stable video-to-video mapping (from $\mathbf{r}_i$ to $\mathbf{f}_i$) using a recurrent neural network (RNN) that is trained in an adversarial manner. We train one person-specific network per input video. Inspired by the video-to-video synthesis work of Wang et al. [2018a], our approach assumes that the video frames have been generated by a sequential process, i.e., the generation of a video frame depends only on the history of previous frames (Figure 6). In practice, we use a temporal history of size $L = 2$ in all experiments, so the face rendering RNN looks at $L + 1 = 3$ frames at the same time. The best face renderer $\mathcal{G}^*$ is found by solving the following optimization problem:

$$\mathcal{G}^* = \arg\min_{\mathcal{G}} \max_{\mathcal{D}_s, \mathcal{D}_t} \mathcal{L}(\mathcal{G}, \mathcal{D}_s, \mathcal{D}_t) \ . \tag{4}$$

Here, $\mathcal{D}_s$ is a per-frame spatial patch-based discriminator [Isola et al. 2017], and $\mathcal{D}_t$ is a temporal patch-based discriminator. We train the recurrent generator and the spatial and temporal discriminator of our GAN in an adversarial manner, see Figure 7. In the following, we describe our training objective $\mathcal{L}$ and the network components in more detail.

Training Objective. For training our recurrent neural face rendering network, we employ stochastic gradient decent to optimize the following training objective:

$$\mathcal{L}(\mathcal{G}, \mathcal{D}_s, \mathcal{D}_t) = \mathbb{E}_{(\mathbf{f_i}, \mathbf{r_i})}\left[\mathcal{L}_r(\mathcal{G}) + \lambda_s \mathcal{L}_s(\mathcal{G}, \mathcal{D}_s) + \lambda_t \mathcal{L}_t(\mathcal{G}, \mathcal{D}_t)\right] \ . \tag{5}$$

Here, $\mathcal{L}_r$ is a photometric reconstruction loss, $\mathcal{L}_s$ is a per-frame spatial adversarial loss, and $\mathcal{L}_t$ is our novel adversarial temporal consistency loss that is based on difference images. Let $\mathbf{f}_{i-L}^{i}$ denote the tensor of video frames from frame $\mathbf{f}_{i-L}$ to the current frame $\mathbf{f}_i$. The corresponding tensor of synthetic composites $\mathbf{r}_{i-L}^{i}$ is defined in a similar way. For each of the $L+1$ time steps, we employ an ℓ_1-loss to enforce the photometric reconstruction of the ground truth:

$$\mathcal{L}_r(\mathcal{G}) = \sum_{l=0}^{L} \left\| m_{i-L+l} \otimes \left(\mathbf{f}_{i-L+l} - \mathcal{G}(\mathbf{c}_{i,l})\right) \right\|_1 ,$$
$$\text{with } \mathbf{c}_{i,l} = \left(\mathbf{r}_{i-L}^{i-L+l}, \mathbf{o}_{i-L}^{i-L+l-1}\right) . \quad (6)$$

Here, the $\mathbf{c}_{i,l}$ are the generator inputs for the current frame i and time step l, with $\mathbf{o}_{i-L}^{i-L+l-1}$ being the tensor of output frames for the previous time steps. $\otimes$ is the Hadamard product and m_{i-L+l} is a mouth re-weighting mask that gives a higher weight to photometric errors in the mouth region (Figure 5). The mask is 1 away from the mouth, 10 for the mouth region, and has a smooth transition in between. Note the same generator $\mathcal{G}$ is shared across all time steps. For each time step, missing outputs of non existent previous frames (we only unroll 3 steps) and network inputs that are in the future are replaced by zeros (Figure 6). In addition to the reconstruction loss, we also enforce a separate patch-based adversarial loss for each frame:

$$\mathcal{L}_s(\mathcal{G}, \mathcal{D}_s) = \sum_{l=0}^{L} \Big[\log(\mathcal{D}_s(\mathbf{r}_{i-L+l}, \mathbf{f}_{i-L+l})) + \log(1 - \mathcal{D}_s(\mathbf{r}_{i-L+l}, \mathcal{G}(\mathbf{c}_{i,l})))\Big] . \quad (7)$$

Note there exists only one discriminator network $\mathcal{D}_s$, which is shared across all time steps. We also employ an adversarial temporal consistency loss based on difference images [Martin-Brualla et al. 2018]:

$$\mathcal{L}_t(\mathcal{G}, \mathcal{D}_t) = \log(\mathcal{D}_t(r_{i-L}^{i}, \Delta_{i,l}(\mathbf{f}))) + \log(1 - \mathcal{D}_t(r_{i-L}^{i}, \Delta_{i,l}(\mathbf{o}))) . \quad (8)$$

Here, $\Delta_{i,l}(\mathbf{f})$ is the ground truth tensor and $\Delta_{i,l}(\mathbf{o})$ the tensor of synthesized difference images. The operator $\Delta(\bullet)$ takes the difference of subsequent frames in the sequence:

$$\Delta_{i,l}(\mathbf{x}) = \mathbf{x}_{i-L+1}^{i} - \mathbf{x}_{i-L}^{i-1} . \quad (9)$$

Network Architecture. For the neural face rendering network, we employ an encoder-decoder network with skip connections that is based on U-Net [Ronneberger et al. 2015]. Our spatial and temporal discriminators are inspired by Isola et al. [2017] and Wang et al. [2018a]. Our network has 75 million trainable parameters. All sub-networks ($\mathcal{G}$, $\mathcal{D}_s$, $\mathcal{D}_t$) are trained from scratch, i.e., starting from random initialization. We alternate between the minimization to train $\mathcal{G}$ and the maximization to train $\mathcal{D}_s$ as well as $\mathcal{D}_t$. In each iteration step, we perform both the minimization as well as the maximization on the same data, i.e., the gradients with respect to the generator and discriminators are computed on the same batch of images. We do not add any additional weighting between the gradients with respect to the generator and discriminators as done in Isola et al. [2017]. The rest of the training procedure follows Isola et al. [2017]. For more architecture details, see Supplemental W13.

Table 2. Input sequences. We recorded three sequences, each about 1 hour long. The sequences contain ground truth sentences and test sentences we edit, and also the first 500 sentences from the TIMIT dataset. We also downloaded a 1.5 hour long interview from YouTube that contains camera and hand motion, and an erroneous transcript. Seq2 and Seq3 are both 60fps. Seq1 was recorded at 240fps, but since our method produces reasonable results with lower frame rates, we discarded frames and effectively used 60fps. Seq4 is 25fps, and still produces good results.

	Source	Transcript	Length
Seq1	Our recording	Manually verified	~1 hour
Seq2	Our recording	Manually verified	~1 hour
Seq3	Our recording	Manually verified	~1 hour
Seq4	YouTube	Automatic (has errors)	~1.5 hours

The rendering procedure produces photo-realistic video frames of the subject, appearing to speak the new phrase W. These localized edits seamlessly blend into the original video, producing an edited result, all derived from text.

Adding Audio. The video produced by our pipeline is mute. To add audio we use audio synthesized either by the built in speech synthesizer in Mac OS X, or by VoCo [Jin et al. 2017]. An alternative is to obtain an actual recording of the performer's voice. In this scenario, we retime the resulting video to match the recording at the level of phones. Unless noted otherwise, all of our synthesis results presented in the performer's own voice are generated using this latter method. Note that for move and delete edits we use the performer's voice from the original video.

4 RESULTS

We show results for our full approach on a variety of videos, both recorded by ourselves and downloaded from YouTube (Section 4). We encourage the reader to view video results (with audio) in the supplemental video and website, since our results are hard to evaluate from static frames.

Runtime Performance. 3D face reconstruction takes 110ms per frame. Phoneme alignment takes 20 minutes for a 1 hour speech video. Network training takes 42 hours. We train for 600K iteration steps with a batch size of 1. Viseme search depends on the size of the input video and the new edit. For a 1 hour recording with continuous speech, viseme search takes between 10 minutes and 2 hours for all word insertion operations in this paper. Neural face rendering takes 132ms per frame. All other steps of our pipeline incur a negligible time penalty.

4.1 Video Editing

Our main application is text-based editing of talking-head video. We support moving and deleting phrases, and the more challenging task of adding new unspoken words. A few examples of replacing one or more words by unspoken word(s) are shown in Figure 1 and Figure 9. Our approach produces photo-realistic results with good audio to video alignment and a photo-realistic mouth interior including highly detailed teeth (Figure 10). For more examples of adding new words, and results for moves and deletes we refer to the supplemental video and Supplemental W1–W4.

Fig. 8. Comparison of different neural face rendering backends: We compare the output of our approach with a baseline that is trained based on input data as proposed in Deep Video Portraits (DVP) [Kim et al. 2018b]. DVP does not condition on the background and thus cannot handle dynamic background. In addition, this alternative approach fails if parts of the foreground move independently of the head, e.g., the hands. Our approach explicitly conditions on the background and can thus handle these challenging cases with ease. In addition, our approach only has to spend capacity in the mouth region (we also re-weight the reconstruction loss based on a mouth mask), thus our approach gives much sharper higher quality results. *Video credit (middle): The Mind of The Universe.*

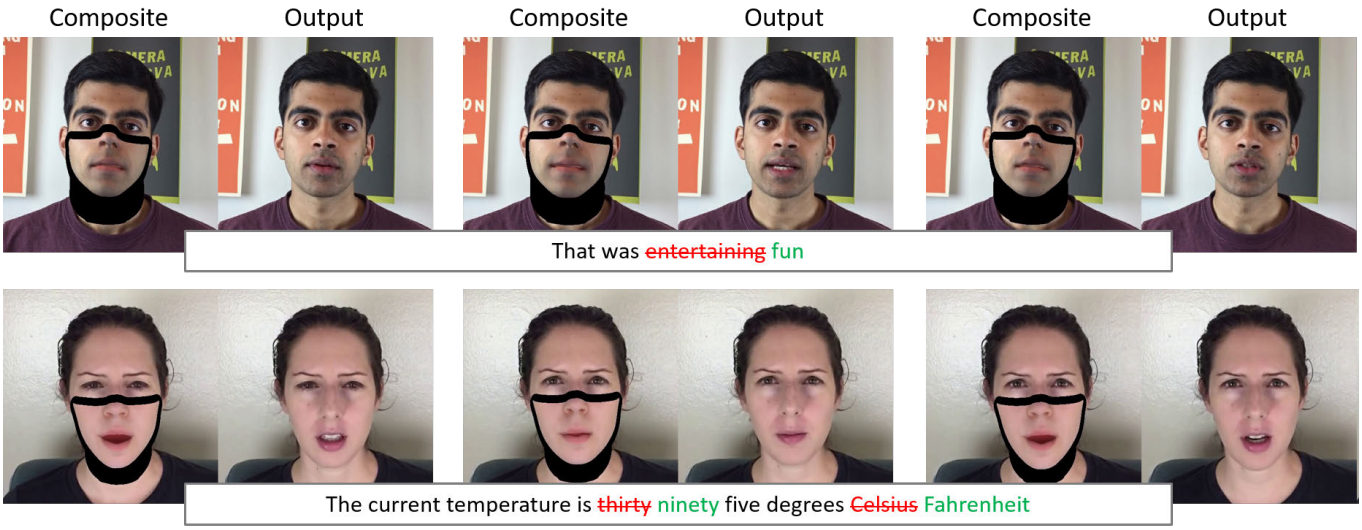


Fig. 9. Our approach enables a large variety of text-based edits, such as deleting, rearranging, and adding new words. Here, we show examples of the most challenging of the three scenarios, adding one or more unspoken words. As can be seen, our approach obtains high quality reenactments of the new words based on our neural face rendering approach that converts synthetic composites into photo-real imagery. For video results we refer to the supplemental.

Our approach enables us to seamlessly re-compose the modified video segments into the original full frame video footage, and to seamlessly blend new segments into the original (longer) video. Thus our approach can handle arbitrarily framed footage, and is agnostic to the resolution and aspect ratio of the input video. It also enables localized edits (i.e. using less computation) that do not alter most of the original video and can be incorporated into a standard editing pipeline. Seamless composition is made possible by our neural face rendering strategy that conditions video generation on the original background video. This approach allows us to accurately reproduce the body motion and scene background (Figure 11). Other neural rendering approaches, such as Deep Video Portraits [Kim et al. 2018b] do not condition on the background, and thus cannot guarantee that the body is synthesized at the right location in the frame.

4.2 Translation

Besides text-based edits, such as adding, rearranging, and deleting words, our approach can also be used for video translation, as long as the source material contains similar visemes to the target language. Our viseme search pipeline is language agnostic. In order to support a new language, we only require a way to convert words into individual phonemes, which is already available for many languages. We show results in which an English speaker appears to speak German (Supplemental W5).

4.3 Full Sentence Synthesis Using Synthetic Voice

With the rise of voice assistants like Alexa, Siri and the Google Assistant, consumers have been getting comfortable with voice-based interactions. We can use our approach to deliver corresponding video. Given a video recording of an actor who wishes to serve as the face of the assistant, our tool could be used to produce the video for any utterance such an assistant might make. We show results of full sentence synthesis using the native Mac OS voice synthesizer

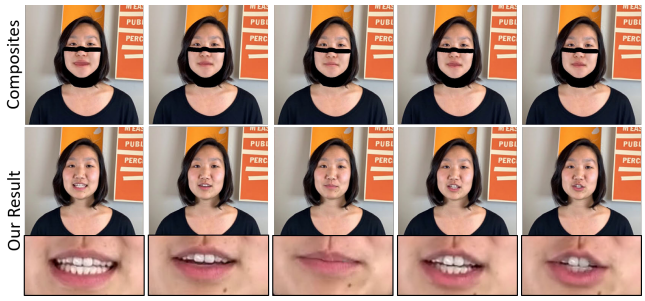


Fig. 10. Our approach synthesizes the non-rigid motion of the lips at high quality (even lip rolling is captured) given only a coarse computer graphics rendering as input. In addition, our approach synthesizes a photorealistic mouth interior including highly detailed teeth. The synthesis results are temporally coherent, as can be seen in the supplemental video.

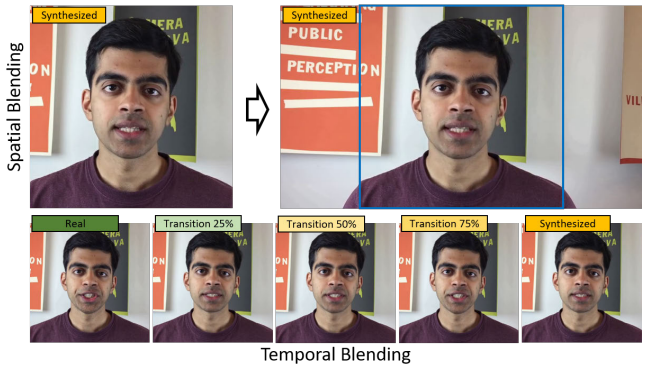


Fig. 11. Our approach enables us to seamlessly compose the modified segments back into the original full frame input video sequence, both spatially as well as temporally. We do this by explicitly conditioning video generation on the re-timed background video.

(Supplemental W7). Our system could also be used to easily create instruction videos with more fine-grained content adaptation for different target audiences, or to create variants of storytelling videos that are tailored to specific age groups.

5 EVALUATION, ANALYSIS & COMPARISONS

To evaluate our approach we have analyzed the content and size of the input video data needed to produce good results and we have compared our approach to alternative talking-head video synthesis techniques.

5.1 Size of Input Video

We performed a qualitative study on the amount of data required for phoneme retrieval. To this end, we iteratively reduced the size of the used training video. We tested our retrieval approach with 5%, 10%, 50%, and 100% of the training data (Supplemental W8). More data leads in general to better performance and visually more pleasing results, but the quality of the results degrade gracefully with the amount of used data. Best results are obtained with the full dataset.

We also evaluate the amount of training data required for our neural face renderer. Using seq4 (our most challenging sequence),

Fig. 12. Evaluation of Parameter Blending: Without our parameter blending strategy, the editing results are temporally unstable. In this example, the mouth unnaturally closes instantly between two frames without blending, while it closes smoothly with our blending approach.

we test a self-reenactment scenario in which we compare the input frames to our result with varying training data size. We obtain errors (mean RMSE per-image) of 0.018 using 100%, 0.019 using 50% and 0.021 using only 5% of the data (R,G,B $\in [0, 1]$). This result suggests that our neural renderer requires less data than our viseme retrieval pipeline, allowing us to perform certain edits (e.g., deletion) on shorter videos.

5.2 Size of Edit

We tested our system with various synthesized phrases. We randomly select from a list of "things that smell" and synthesize the phrases into the sentence "I love the smell of X in the morning" (Supplemental W11). We found that phrase length does not directly correlate with result quality. Other factors, such as the visemes that comprise the phrase and phoneme alignment quality influence the final result.

5.3 Evaluation of Parameter Space Blending

We evaluate the necessity of our parameter blending strategy by comparing our approach to a version without the parameter blending (Figure 12 and Supplemental W12). Without our parameter space blending strategy the results are temporally unstable.

5.4 Comparison to MorphCut

MorphCut is a tool in Adobe Premiere Pro that is designed to remove jump cuts in talking-head videos, such as those introduced by moving or deleting words. It is based on the approach of Berthouzoz et al. [2012], requires the performer to be relatively still in the video and cannot synthesize new words. In Figure 13, we compare our approach to MorphCut in the word deletion scenario and find that our approach is able to successfully remove the jump cut, while MorphCut fails due to the motion of the head.

We also tried to apply MorphCut to the problem of word addition. To this end, we first applied our phoneme/viseme retrieval pipeline to select suitable frames to compose a new word. Afterwards, we tried to remove the jump cuts between the different phoneme subsequences with MorphCut (Figure 14). While our approach with

Fig. 13. We compare our approach in the word deletion scenario to *MorphCut*. MorphCut fails on the second, third, and forth frames shown here while our approach is able to successfully remove the jump cut. *Video credit: The Mind of The Universe.*

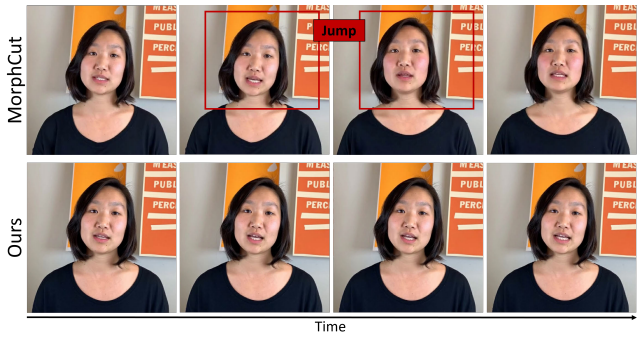


Fig. 14. We tried to stitch retrieved viseme sequences with MorphCut to generate a new word. While our approach with the parameter space blending strategy is able to generate a seamless transition, MorphCut produces a big jump of the head between the two frames.

parameter space blending is able to generate seamless transitions, MorphCut produces big jumps and can not smooth them out.

5.5 Comparison to Facial Reenactment Techniques

We compare our facial reenactment backend with a baseline approach that is trained based on the input data as proposed in Deep Video Portraits [Kim et al. 2018b] (Figure 8). For a fair comparison, we trained our recurrent generator network (including the temporal GAN loss, but without our mouth re-weighting mask) with Deep Video Portraits style input data (diffuse rendering, uv-map, and eye conditioning) and try to regress a realistic output video. Compared to Deep Video Portraits [Kim et al. 2018b], our approach synthesizes a more detailed mouth region, handles dynamic foregrounds well, such as for example moving hands and arms, and can better handle dynamic background. We attribute this to our mouth re-weighting mask and explicitly conditioning on the original background and body, which simplifies the learning task, and frees up capacity in the network. Deep Video Portraits struggles with any form of motion that is not directly correlated to the head, since the head motion is the only input in their technique. We refer to the supplemental video for more results.

We also compare our approach to Face2Face [Thies et al. 2016], see Figure 15. Our neural face rendering approach can better handle the

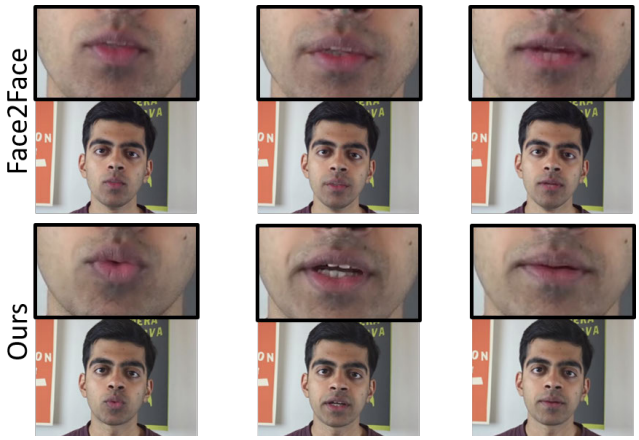


Fig. 15. Comparison to the Face2Face [Thies et al. 2016] facial reenactment approach. Our approach produces high quality results, while the retrieval-based Face2Face approach exhibits ghosting artifacts and is temporally unstable. We refer to the supplemental video for more results.

complex articulated motion of lips, e.g., lip rolling, and synthesizes a more realistic mouth interior. The Face2Face results show ghosting artifacts and are temporally unstable, while our approach produces temporally coherent output. We refer to the supplemental video for more results.

5.6 Ablation Study

We also performed an ablation study to evaluate the new components of our approach (see Figure 16). We perform the study in a self-reenactment scenario in which we compare our result to the input frames. To this end we compare our complete approach (Full) with two simplified approaches. The first simplification removes both the mouth mask and background conditioning (w\o bg & mask) from our complete approach, while the second simplification only removes the mouth mask (w\o mask). As shown in Figure 16, all components positively contribute to the quality of the results. This is especially noticeable in the mouth region, where the quality and level of detail of the teeth is drastically improved. In addition, we also show the result obtained with the Deep Video Portraits (DVP) of Kim et al. [2018a]. We do not investigate alternatives to the RNN in our ablation study, as Wang et al. [2018a] have already demonstrated that RNNs outperform independent per-frame synthesis networks.

5.7 User Study

To quantitatively evaluate the quality of videos generated by our text-based editing system, we performed a web-based user study with $N = 138$ participants and collected 2993 individual responses, see Table 3. The study includes videos of two different talking heads, *Set 1* and *Set 2*, where each set contains 6 different base sentences. For each of the base sentences, we recorded a corresponding target sentence in which one or more words are different. We use both the base and target sentences as ground truth in our user study. Next, we employed our pipeline to artificially change the base into the target sentences. In total, we obtain $2 \times 3 \times 6 = 36$ video clips.

Table 3. We performed a user study with $N = 138$ participants and collected in total 2993 responses to evaluate the quality of our approach. Participants were asked to respond to the statement "This video clip looks real to me" on a 5-point Likert scale from 1 (*strongly disagree*) to 5 (*strongly agree*). We give the percentage for each score, the average score, and the percentage of cases the video was rated as 'real' (a score of 4 or higher). The difference between conditions is statistically significant (Kruskal-Wallis test, $p < 10^{-30}$). Our results are different from both GT-base and from GT-target (Tukey's honest significant difference procedure, $p < 10^{-9}$ for both tests). This suggests that while our results are often rated as real, they are still not on par with real video.

	GT Base Videos							**GT Target Videos**							**Our Modified Videos**						
	Scores							Scores							Scores						
	5	4	3	2	1	Σ	'real'	5	4	3	2	1	Σ	'real'	5	4	3	2	1	Σ	'real'
Set 1	45.3	36.3	7.9	10.0	0.5	4.1	81.6%	47.0	31.9	9.7	10.1	1.4	4.1	78.9%	31.9	25.2	10.9	23.9	8.2	3.5	57.1%
Set 2	41.6	38.1	9.9	9.2	1.2	4.1	79.7%	45.7	39.8	8.7	5.4	0.4	4.3	85.6%	29.3	32.8	9.4	22.9	5.7	3.9	62.1%
Mean	43.5	37.2	8.9	9.6	0.9	4.1	80.6%	46.4	35.9	9.2	7.7	0.9	4.2	82.2%	30.6	29.0	10.1	23.4	7.0	3.7	59.6%

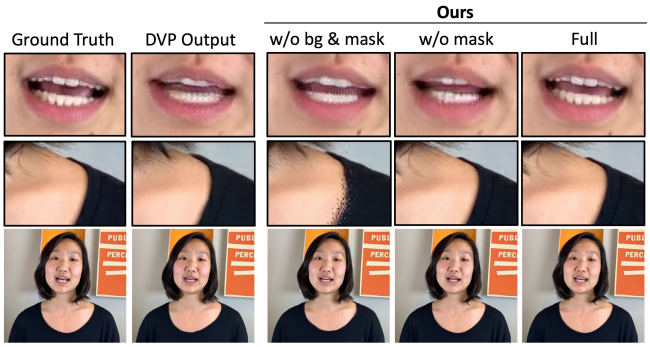


Fig. 16. Ablation study comparing ground truth with several versions of our approach: a simplified version without providing the mouth mask and the background conditioning (w/o bg & mask); a simplified version that provides the background but not the mouth mask (w/o mask); and our complete approach with all new components (Full). In addition, we show a result from the Deep Video Portraits (DVP) of Kim et al. [2018a]. All components of our approach positively contribute to the quality of the results, and our full method outperforms DVP. This is especially noticeable in the hair and mouth regions.

In the study, the video clips were shown one video at a time to participants $N = 138$ in randomized order and they were asked to respond to the statement "This video clip looks real to me" on a 5-point Likert scale (5-*strongly agree*, 4-*agree*, 3-*neither agree nor disagree*, 2-*disagree*, 1-*strongly disagree*). As shown in Table 3, the real ground truth base videos were only rated to be 'real' 80.6% of the cases and the real ground truth target videos were only rated to be 'real' 82.2% of the cases (score of 4 or 5). This shows that the participants were already highly alert, given they were told it was a study on the topic of 'Video Editing'. Our pipeline generated edits were rated to be 'real' 59.6% of the cases, which means that more than half of the participants found those clips convincingly real. Table 3 also reports the percentage of times each score was given and the average score per video set. Given the fact that synthesizing convincing audio/video content is very challenging, since humans are highly tuned to the slightest audio-visual misalignments (especially for faces), this evaluation shows that our approach already achieves compelling results in many cases.

6 LIMITATIONS & FUTURE WORK

While we have demonstrated compelling results in many challenging scenarios, there is room for further improvement and follow-up work: (1) Our synthesis approach requires a re-timed background video as input. Re-timing changes the speed of motion, thus eye blinks and gestures might not perfectly align with the speech anymore. To reduce this effect, we employ a re-timing region that is longer than the actual edit, thus modifying more of the original video footage, with a smaller re-timing factor. For the insertion of words, this could be tackled by a generative model that is able to synthesize realistic complete frames that also include new body motion and a potentially dynamic background. (2) Currently our phoneme retrieval is agnostic to the mood in which the phoneme was spoken. This might for example lead to the combination of happy and sad segments in the blending. Blending such segments to create a new word can lead to an uncanny result. (3) Our current viseme search aims for quality but not speed. We would like to explore approximate solutions to the viseme search problem, which we believe can allow interactive edit operations. (4) We require about 1 hour of video to produce the best quality results. To make our method even more widely applicable, we are investigating ways to produce better results with less data. Specifically, we are investigating ways to transfer expression parameters across individuals, which will allow us to use one pre-processed dataset for all editing operations. (5) Occlusions of the lower face region, for example by a moving hand, interfere with our neural face renderer and lead to synthesis artifacts, since the hand can not be reliably re-rendered. Tackling this would require to also track and synthesize hand motions. Nevertheless, we believe that we demonstrated a large variety of compelling text-based editing and synthesis results. In the future, end-to-end learning could be used to learn a direct mapping from text to audio-visual content.

7 CONCLUSION

We presented the first approach that enables text-based editing of talking-head video by modifying the corresponding transcript. As demonstrated, our approach enables a large variety of edits, such as addition, removal, and alteration of words, as well as convincing language translation and full sentence synthesis. We believe our approach is a first important step towards the goal of fully text-based editing and synthesis of general audio-visual content.

ACKNOWLEDGMENTS

This work was supported by the Brown Institute for Media Innovation, the Max Planck Center for Visual Computing and Communications, ERC Consolidator Grant 4DRepLy (770784), Adobe Systems, and the Office of the Dean for Research at Princeton University.

REFERENCES

Annosoft. 2008. Lipsync Tool. (2008). http://www.annosoft.com/docs/Visemes17.html

Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. 2017. Bringing Portraits to Life. *ACM Transactions on Graphics (SIGGRAPH Asia)* 36, 6 (November 2017), 196:1–13. https://doi.org/10.1145/3130800.3130818

Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recycle-GAN: Unsupervised Video Retargeting. In *ECCV*.

Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for Placing Cuts and Transitions in Interview Video. *ACM Trans. Graph.* 31, 4, Article 67 (July 2012), 8 pages. https://doi.org/10.1145/2185520.2185563

Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. 2004. Exchanging Faces in Images. *Computer Graphics Forum (Eurographics)* 23, 3 (September 2004), 669–676. https://doi.org/10.1111/j.1467-8659.2004.00799.x

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 187–194. https://doi.org/10.1145/311535.311556

James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. 2018. Large Scale 3D Morphable Models. *International Journal of Computer Vision* 126, 2 (April 2018), 233–254. https://doi.org/10.1007/s11263-017-1009-7

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 (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 353–360. https://doi.org/10.1145/258734.258880

Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. 2015. Real-time High-fidelity Facial Performance Capture. *ACM Transactions on Graphics (SIGGRAPH)* 34, 4 (July 2015), 46:1–9. https://doi.org/10.1145/2766943

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. 2018. Everybody Dance Now. *arXiv e-prints* (August 2018). arXiv:1808.07371

Yao-Jen Chang and Tony Ezzat. 2005. Transferable Videorealistic Speech Animation. In *Symposium on Computer Animation (SCA)*. 143–151. https://doi.org/10.1145/1073368.1073388

Qifeng Chen and Vladlen Koltun. 2017. Photographic Image Synthesis with Cascaded Refinement Networks. In *International Conference on Computer Vision (ICCV)*. 1520–1529. https://doi.org/10.1109/ICCV.2017.168

Pengfei Dou, Shishir K. Shah, and Ioannis A. Kakadiaris. 2017. End-To-End 3D Face Reconstruction With Deep Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. 2016. JALI: An Animator-centric Viseme Model for Expressive Lip Synchronization. *ACM Trans. Graph.* 35, 4, Article 127 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925984

Tony Ezzat, Gadi Geiger, and Tomaso Poggio. 2002. Trainable Videorealistic Speech Animation. *ACM Transactions on Graphics (SIGGRAPH)* 21, 3 (July 2002), 388–398. https://doi.org/10.1145/566654.566594

Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. 2014. Driving High-Resolution Facial Scans with Video Performance Capture. *ACM Transactions on Graphics* 34, 1 (December 2014), 8:1–14.

J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. 1993. DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM. (1993). http://www.ldc.upenn.edu/Catalog/LDC93S1.html

Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormaehlen, Patrick Pérez, and Christian Theobalt. 2014. Automatic Face Reenactment. In *CVPR*. 4217–4224. https://doi.org/10.1109/CVPR.2014.537

Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2015. VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track. *Computer Graphics Forum (Eurographics)* 34, 2 (May 2015), 193–204. https://doi.org/10.1111/cgf.12552

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Transactions on Graphics* 35, 3 (June 2016), 28:1–15. https://doi.org/10.1145/2890493

Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. 2018. Warp-guided GANs for Single-photo Facial Animation. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. ACM, New York, NY, USA, Article 231, 231:1–231:12 pages. http://doi.acm.org/10.1145/3272127.3275043

Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. 2018. Unsupervised Training for 3D Morphable Model Regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*.

Y. Guo, J. Zhang, J. Cai, B. Jiang, and J. Zheng. 2018. CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), 1–1. https://doi.org/10.1109/TPAMI.2018.2837742

Andrew J Hunt and Alan W Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, Vol. 1. IEEE, 373–376.

IBM. 2016. IBM Speech to Text Service. https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/speech-to-text/. (2016). Accessed 2016-12-17.

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Transactions on Graphics (SIGGRAPH)* 34, 4 (July 2015), 45:1–14. https://doi.org/10.1145/2766974

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Zeyu Jin, Gautham J Mysore, Stephen Diverdi, Jingwan Lu, and Adam Finkelstein. 2017. VoCo: text-based insertion and replacement in audio narration. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 96.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations (ICLR)*.

Ira Kemelmacher-Shlizerman. 2013. Internet-Based Morphable Model. In *International Conference on Computer Vision (ICCV)*. 3256–3263.

Ira Kemelmacher-Shlizerman, Aditya Sankar, Eli Shechtman, and Steven M. Seitz. 2010. Being John Malkovich. In *European Conference on Computer Vision (ECCV)*. 341–353. https://doi.org/10.1007/978-3-642-15549-9_25

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. 2018a. Deep Video Portraits. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 163.

H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, N. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. 2018b. Deep Video Portraits. *ACM Transactions on Graphics 2018 (TOG)* (2018).

Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational Video Editing for Dialogue-driven Scenes. *ACM Trans. Graph.* 36, 4, Article 130 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073653

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

Kai Li, Qionghai Dai, Ruiping Wang, Yebin Liu, Feng Xu, and Jue Wang. 2014. A Data-Driven Approach for Facial Expression Retargeting in Video. *IEEE Transactions on Multimedia* 16, 2 (February 2014), 299–310.

Kang Liu and Joern Ostermann. 2011. Realistic facial expression synthesis for an image-based talking head. In *International Conference on Multimedia and Expo (ICME)*. https://doi.org/10.1109/ICME.2011.6011835

L. Liu, W. Xu, M. Zollhoefer, H. Kim, F. Bernard, M. Habermann, W. Wang, and C. Theobalt. 2018. Neural Animation and Reenactment of Human Actor Videos. *ArXiv e-prints* (September 2018). arXiv:1809.03658

Zicheng Liu, Ying Shan, and Zhengyou Zhang. 2001. Expressive Expression Mapping with Ratio Images. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 271–276. https://doi.org/10.1145/383259.383289

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. 2018. LookinGood: Enhancing Performance Capture with Real-time Neural Re-rendering. *ACM Trans. Graph.* 37, 6, Article 255 (December 2018), 14 pages.

Wesley Mattheyses, Lukas Latacz, and Werner Verhelst. 2010. Optimized photorealistic audiovisual speech synthesis using active appearance modeling. In *Auditory-Visual Speech Processing*. 8–1.

Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. (2014). https://arxiv.org/abs/1411.1784 arXiv:1411.1784.

Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. 2018. paGAN: Real-time Avatars Using Dynamic Textures. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. ACM, New York, NY, USA, Article 258, 12 pages. https://doi.org/10.1145/3272127.3275075

Robert Ochshorn and Max Hawkins. 2016. Gentle: A Forced Aligner. https://lowerquality.com/gentle/. (2016). Accessed 2018-09-25.

Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. 2017. Realistic Dynamic Facial Textures from a Single Image using GANs. In *International Conference on Computer Vision (ICCV)*. 5439–5448. https://doi.org/10.1109/ICCV.2017.580

Amy Pavel, Dan B Goldman, Björn Hartmann, and Maneesh Agrawala. 2016. VidCrit: Video-based Asynchronous Video Review. In *Proc. of UIST*. ACM, 517–528.
Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video Digests: A Browsable, Skimmable Format for Informational Lecture Videos. In *Proc. of UIST*. 573–582.
Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.
Elad Richardson, Matan Sela, and Ron Kimmel. 2016. 3D Face Reconstruction by Learning from Synthetic Data. In *International Conference on 3D Vision (3DV)*. 460–469. https://doi.org/10.1109/3DV.2016.56
Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. 2017. Learning Detailed Face Reconstruction from a Single Image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 5553–5562. https://doi.org/10.1109/CVPR.2017.589
Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 234–241.
Joseph Roth, Yiying Tong Tong, and Xiaoming Liu. 2017. Adaptive 3D Face Reconstruction from Unconstrained Photo Collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 11 (November 2017), 2127–2141. https://doi.org/10.1109/TPAMI.2016.2636829
Steve Rubin, Floraine Berthouzoz, Gautham J Mysore, Wilmot Li, and Maneesh Agrawala. 2013. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 113–122.
Matan Sela, Elad Richardson, and Ron Kimmel. 2017. Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. In *International Conference on Computer Vision (ICCV)*. 1585–1594. https://doi.org/10.1109/ICCV.2017.175
Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*. IEEE, 4779–4783.
Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. 2014. Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM Transactions on Graphics (SIGGRAPH Asia)* 33, 6 (November 2014), 222:1–13. https://doi.org/10.1145/2661229.2661290
Hijung Valentina Shin, Wilmot Li, and Frédo Durand. 2016. Dynamic Authoring of Audio with Linked Scripts. In *Proc. of UIST*. 509–516.
Qianru Sun, Ayush Tewari, Weipeng Xu, Mario Fritz, Christian Theobalt, and Bernt Schiele. 2018. A Hybrid Model for Identity Obfuscation by Face Replacement. In *European Conference on Computer Vision (ECCV)*.
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. https://doi.org/10.1145/3072959.3073640
Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. 2017. A Deep Learning Approach for Generalized Speech Animation. *ACM Trans. Graph.* 36, 4, Article 93 (July 2017), 11 pages. https://doi.org/10.1145/3072959.3073699
Ayush Tewari, Michael Zollhöfer, Florian Bernard, Pablo Garrido, Hyeongwoo Kim, Patrick Perez, and Christian Theobalt. 2018a. High-Fidelity Monocular Face Reconstruction based on an Unsupervised Model-based Face Autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), 1–1. https://doi.org/10.1109/TPAMI.2018.2876842
Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. 2018b. Self-supervised Multi-level Face Model Learning for Monocular Reconstruction at over 250 Hz. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. 2017. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *ICCV*. 3735–3744. https://doi.org/10.1109/ICCV.2017.401
Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2387–2395. https://doi.org/10.1109/CVPR.2016.262
Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. 2017. Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 1493–1502. https://doi.org/10.1109/CVPR.2017.163
Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. Quickcut: An interactive tool for editing narrated video. In *Proc. of UIST*. 497–507.
Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. In *SSW*. 125.
Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. 2005. Face Transfer with Multilinear Models. *ACM Transactions on Graphics (SIGGRAPH)* 24, 3 (July 2005), 426–433. https://doi.org/10.1145/1073204.1073209
Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018a. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.
Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018b. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *CVPR*.
O. Wiles, A.S. Koepke, and A. Zisserman. 2018. X2Face: A network for controlling face generation by using images, audio, and pose codes. In *European Conference on Computer Vision*.
Jiahong Yuan and Mark Liberman. 2008. Speaker identification on the SCOTUS corpus. *The Journal of the Acoustical Society of America* 123, 5 (2008), 3878–3878. https://doi.org/10.1121/1.2935783 arXiv:https://doi.org/10.1121/1.2935783
Heiga Zen, Keiichi Tokuda, and Alan W Black. 2009. Statistical parametric speech synthesis. *speech communication* 51, 11 (2009), 1039–1064.
Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. 2018. Visemenet: Audio-driven Animator-centric Speech Animation. *ACM Trans. Graph.* 37, 4, Article 161 (July 2018), 161:1–161:10 pages.
M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. 2018. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)* 37, 2 (2018).

A PHONEME & VISEME CONTENT

Our matching algorithm (Section 3.3) is designed to find the longest match between subsequences of phonemes/visemes in the edit and the input video. Suppose our input video consists of all the sentences in the TIMIT corpus [Garofolo et al. 1993], a set that has been designed to be phonetically rich by acoustic-phonetic reseachers. Figure 17 plots the probability of finding an exact match anywhere in TIMIT to a phoneme/viseme subsequence of length $K \in [1, 10]$. Exact matches of more than 4-6 visemes or 3-5 phonemes are rare. This result suggests that even with phonetically rich input video we cannot expect to find edits consisting of long sequences of phonemes/visemes (e.g. multiword insertions) in the input video and that our approach of combining shorter subsequences with parameter blending is necessary.

Figure 17 also examines the variation in individual viseme instances across the set of 2388 sentences in the TIMIT corpus. We see that there is variation both between different visemes and within a class of visemes. These observations led us to incorporate viseme distance and length in our search procedure (Section 3.3) and informed our blending strategy (Section 3.4).

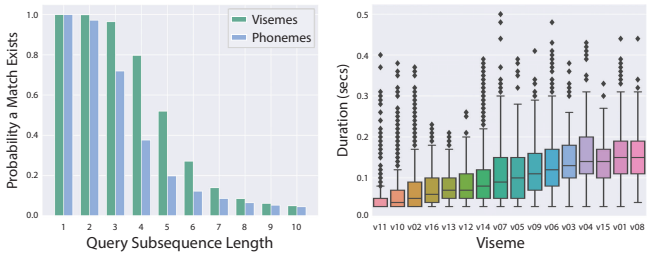


Fig. 17. Left: probability of matching phoneme/viseme subsequences of length $K \in [1, 10]$ in the TIMIT corpus. To ensure that the query subsequences reflect the distribution of such sequences in English we employ a leave-one-out strategy: we choose a random TIMIT sequence of length K, and look for an exact match anywhere in the rest of the dataset. Exact matches of more than 4-6 vismes and 2-3 phonemes are uncommon. Right: variation in viseme duration in TIMIT. Different instances of a single viseme vary by up to an order of magnitude. Between different visemes, the median instance length varies by a factor of five.