Eurographics Symposium on Rendering (DL-only Track) (2019)
T. Boubekeur and P. Sen (Editors)

Puppet Dubbing

O. Fried[1] and M. Agrawala[1]

[1]Stanford University, USA

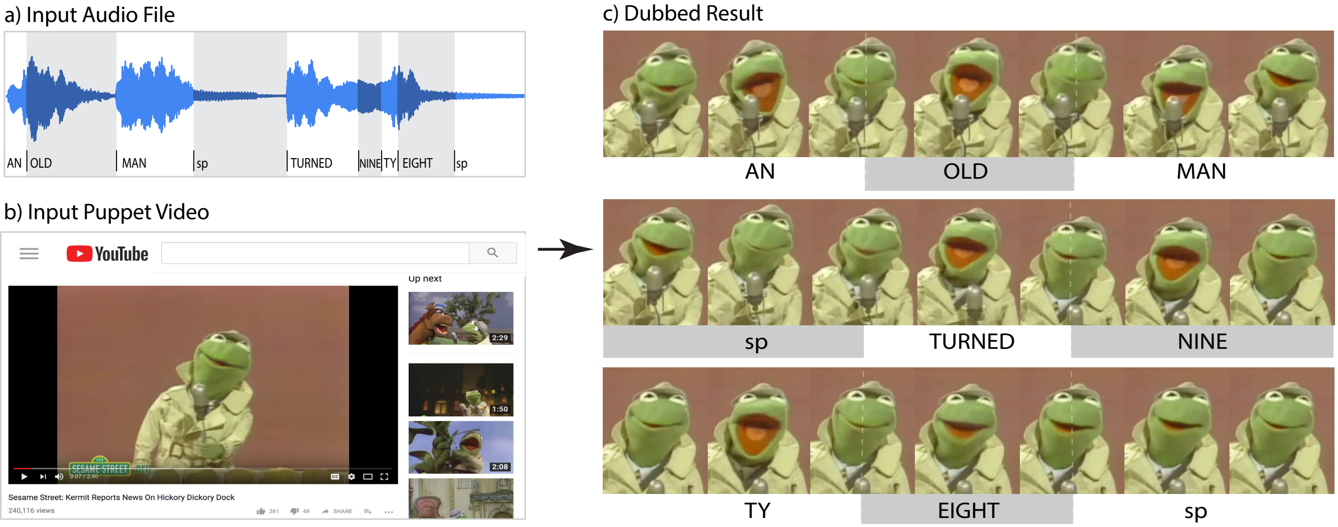

Figure 1: *Given an audio file and a puppet video, we produce a dubbed result in which the puppet is saying the new audio phrase with proper mouth articulation. Specifically, each syllable of the input audio matches a closed-open-closed mouth sequence in our dubbed result. We present two methods, one semi-automatic appearance-based and one fully automatic audio-based, to create convincing dubs.*

Abstract

Dubbing puppet videos to make the characters (e.g. Kermit the Frog) convincingly speak a new speech track is a popular activity with many examples of well-known puppets speaking lines from films or singing rap songs. But manually aligning puppet mouth movements to match a new speech track is tedious as each syllable of the speech must match a closed-open-closed segment of mouth movement for the dub to be convincing. In this work, we present two methods to align a new speech track with puppet video, one semi-automatic appearance-based *and the other fully-automatic* audio-based*. The methods offer complementary advantages and disadvantages. Our appearance-based approach directly identifies closed-open-closed segments in the puppet video and is robust to low-quality audio as well as misalignments between the mouth movements and speech in the original performance, but requires some manual annotation. Our audio-based approach assumes the original performance matches a closed-open-closed mouth segment to each syllable of the original speech. It is fully automatic, robust to visual occlusions and fast puppet movements, but does not handle misalignments in the original performance. We compare the methods and show that both improve the credibility of the resulting video over simple baseline techniques, via quantitative evaluation and user ratings.*

CCS Concepts
• ***Computing methodologies*** *→ Graphics systems and interfaces; Motion processing;* • ***Information systems*** *→ Video search;*

DOI: 10.2312/sr.20191220

1. Introduction

Puppet-based video is widely available online at sites like YouTube and Vimeo in the form of clips from well-known TV shows and films (e.g. *Sesame Street, Barney and Friends, The Muppet Movie,* etc.). The abundance of such clips has led to a vibrant remix culture in which people dub the clips to alter the speech and make the puppets tell jokes [You14c], recite lines from famous films [You14a, You08], speak in a different language or accent [You14b, You14d] or sing rap songs [You16].

But, dubbing such puppet video is challenging as it requires carefully matching the mouth movements of the puppet to a new speech track. Expert puppeteers suggest that puppets are most convincing when each closed-open-closed segment of the puppet mouth corresponds to exactly one syllable of speech [Cur87]. So, the best dubbing efforts usually involve frame-level matching and re-timing of closed-open-closed mouth segments in the video to the syllables in the new speech. Today, such matching and re-timing is largely a manual process and is extremely time-consuming.

In this work, we present two techniques that significantly reduce (or eliminate) the manual effort required to dub a puppet video with a new source speech track (Figure 1). Both methods start by breaking the new speech into a sequence of syllables. Our *appearance-based* method tracks the puppet head and lip motion to identify the closed-open-closed segments. We call these segments *visual syllables* and align them to the syllables in the new speech. Our *audio-based* method assumes that the original puppeteer properly aligned the visual syllables to the original puppet speech. It treats the audio syllables in the original puppet speech as a proxy for the visual syllable boundaries and aligns them to the syllables in the new speech. Both methods then re-time the video and the new speech to best match each other.

Our puppet dubbing methods offer different strengths and weaknesses. The appearance-based method is robust to low quality audio and to the presence of background music in the original video, while the audio-based method is robust to low visual quality and occlusions in the video. The appearance-based method does not assume that the original puppeteering performance was well synced with the audio, and can distinguish between on- and off-camera speech, but at a cost of some manual annotation. The audio-based method requires no annotation, but does not directly identify visual syllables and therefore can produce artifacts when the original speech is misaligned with the original puppet video.

The main contributions of this work are:

- Identification of puppeteering guidelines that produce a convincing puppet performance.
- Instantiaion of these guidelines in two novel methods for dubbing existing puppet video with new audio.
- A novel approach for combined video and audio re-timing, which takes into account the rate of change in audio speed.

We quantitatively measure the accuracy of our puppet-specific visual detectors and the amount of stretching and squeezing performed on the videos, which we try to minimize to reduce artifacts. Finally we conduct a user study which finds that our appearance-based and audio-based methods are seen as having significantly higher quality than simple baseline approaches.

2. Related Work

Our puppet dubbing methods build on three areas of related work.

Dubbing humans. Prior work on dubbing has primarily focused on synthesizing video or 3D animation of human faces that match an input audio speech track. These techniques fall into three main categories. (1) Phoneme-based approaches extract the sequence of phonemes in the input speech, generate the visual counterpart of lip and face motion for each phoneme (called a viseme), and concatenate the visemes to form the final result [BCS97, ELFS16, TMTM12, WHS12]. (2) Machine-learning approaches learn the mapping between low-level features of the input speech to the output frames of video or 3D animation [Bra99, KAL*17, SSKS17, TKY*17] (3) Performance capture approaches require an input RGB or RGB-D video of an actor performing the speech and remap the performance onto the output video, 3D animation or even a single image [PLH*15, KGT*18, FKSM16, TZS*16, WBLP11, FJA*14, BHB*11, CBZB15, AECOKC17]. Because human viewers are well attuned to the nuances of human faces, it is a significant challenge for these approaches to generate believable lip and face motions that avoid the uncanny valley. In contrast, viewers are more forgiving with non-human puppets and, as we will show, there are only a few key characteristics of puppet mouth motion (as opposed to human mouth motion) that must be maintained to produce a convincing dub.

Dubbing puppets is different from dubbing humans: human dubbing methods rely on datasets of scanned human heads and deformable head models, automatic face landmark detectors and other human specific tools which are not available for puppets. For that reason we cannot directly apply human-based methods as-is for puppet dubbing. Furthermore, the simplicity of puppet articulation allows us to use much less raw material compared to human dubbing methods. We typically produce adequate results with less than 1 minute of video, while human dubbing from audio often requires hours of raw data.

Dubbing non-human characters. Researchers have used some of the performance capture techniques mentioned earlier [WBLP11, FJA*14, BHB*11, CBZB15, Ado18] to transfer human performances of speech to non-human characters (either 2D cartoons or 3D models). Others have developed techniques to drive mouth motions of such characters using audio peak detection [BJS*08] or a phoneme-based analysis of the speech [ELFS16]. Unlike our approach, these methods all require rigged models of the mouth and lips of the 2D/3D characters.

Speech-driven audio/video editing. Researchers have recently developed a number of tools for transcript-driven editing of speech-based audio and video content. These tools rely on an accurate text transcript of the input speech, either via speech-to-text tools [IBM16] or crowdsourcing services (e.g. `rev.com`), and phoneme-level time alignment of the transcript to the speech [RBM*13, OH16]. They enable text-based editing of talking-head style video [BLA12, FTZ*19] and podcasts [RBM*13, SLD16], vocally annotating video [PGHA16, TBLA16], indexing lecture [PRHA14] and blackboard video [SBLD15], synthesis of new speech audio to stylistically blend into the input speech [JMD*17] and automatic edit-

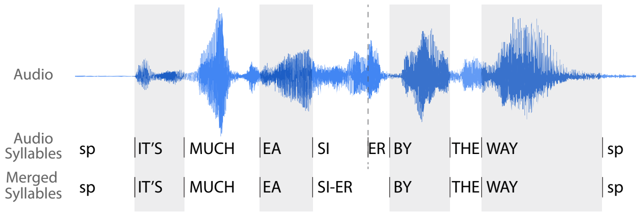

Figure 2: *Automatic audio alignment for the phrase "it's much easier by the way". Given audio with transcript we locate words and phonemes, and derive syllables from them. Each resulting syllable a_i is comprised of a label a_i^{lbl} such as IT'S, MUCH and EA. The label* sp *indicates silence. Each syllable also includes a start time a_i^{in} and end time a_i^{out} indicated by the syllable segment end points. We merge short syllables with their neighbors to produce a final syllable segmentation.*

ing of rough cuts [LDTA17]. We similarly rely on time-aligned text transcripts of the speech in both the input audio and candidate video. However, instead of editing, we focus on using the transcribed syllables for dubbing.

3. Guidelines for Performing Puppet Speech

Our puppet dubbing approach is inspired by puppeteering tutorials and observations of expert puppeteer performances. Based on these observations we have identified three main guidelines for producing visually convincing puppet speech:

(g1) Each syllable in speech should match to one closed-open-closed segment of puppet lip motions. We call such video segments *visual syllables*.

(g2) Lips should be still and ideally closed when the puppet is not speaking as lip motions during silences can be disturbing. We call silences in speech *silence syllables* and closed mouth video segments *visual silence syllables*.

(g3) In rapid speech sequences, instead of a one-to-one match, several spoken syllables may correspond to a single visual syllable.

Guidelines (g1) and (g2) are mentioned in puppeteering training videos [You10]. Although (g3) is less directly documented, we often observed a many-to-one match between multiple syllables in fast speech and a single visual syllable, even in expert performances. This is at times a conscious effort by the puppeteer to simplify the performance, or due to real-world operating constraints on the puppets, and is related to the loss of fricatives and elision of vowels in rapid human speech [Jon11]. In the remainder of this paper we will use the term *syllable* to refer to both silence and non-silence speech syllables and use the specific terms only when necessary for disambiguating between the two types. Similarly, we will use the term *visual syllable* to refer to both silence and non-silence visual syllables.

4. Algorithmic Methods

Our goal is to dub a given puppet video V with new speech audio A according to the guidelines in Section 3. Our approach involves four main steps. (1) We segment the new speech audio track into a sequence of syllables (Section 4.1). (2) We segment the puppet video into a sequence of visual syllables (Section 4.2). (3) We align the audio syllables with the optimal subsequence of visual syllables (Section 4.3). (4) Finally, we re-time the audio and video so that each visual syllable matches the length of a corresponding audio syllable (Section 4.4).

4.1. Step 1: Segment Audio Into Syllables

To segment the new speech audio A into syllables, we first obtain a transcript of it. In practice most of our examples use the closed captions from YouTube, but we have also experimented with automatic speech transcription tools [IBM16, OH16] and crowdsourcing transcription services like `rev.com`. We align the transcript to the audio using P2FA [YL08, RBM*13], a phoneme-based alignment tool, and then combine the phonemes into syllables using the approach of Bartlett et al. [BKC09]. This gives us an ordered sequence $A = (a_1, \ldots, a_n)$ of syllables, each with a label denoting the syllable name, start time and end time $a_i = (a_i^{lbl}, a_i^{in}, a_i^{out})$ (Figure 2). The syllable label *sp* indicates a silence syllable.

While the resulting syllable to speech alignment is usually very good, background music, environmental noise and/or poor enunciation can create some misalignments. Users can optionally fix such misalignments using PRAAT [Boe01] to adjust syllable boundaries. In practice, we've found that 1 minute of speech requires between 0 and 20 minutes of manual tweaking to produce a perfect alignment, and usually less than 5 minutes. However, even with misalignments introduced by P2FA our results are often acceptable. We present results with and without these manual tweaks [supplemental material: audio alignment].

Guideline (g3) suggests that in rapid speech, puppeteers do not articulate every short syllable and instead merge them into a single visual syllable. Thus, if a syllable is shorter than a threshold (set empirically to 150ms) we merge it with its shortest neighboring syllable. We merge in ascending length order, until no more merges are possible. Importantly, we only merge syllables belonging to the same word and not across words, according to puppeteering practices, as the latter produces visible artifacts.

4.2. Step 2: Segment Video Into Visual Syllables

We offer two methods for segmenting the puppet video V into a sequence of visual syllables $V = (v_1, \ldots, v_m)$, one that is appearance-based and one that is audio-based. Like audio syllables, each resulting visual syllable consists of a label, in this case denoting whether the syllable is silent *sp* or non-silent *nsp*, a start time and an end time $v_i = (v_i^{lbl}, v_i^{in}, v_i^{out})$. We describe each segmentation technique and discuss their different strengths and weaknesses

4.2.1. Appearance-based visual syllables.

Our appearance-based algorithm is designed to first classify each frame into one of three categories, *open-mouth*, *closed-mouth* and *invalid* (no visible puppet head) as described below and then use the classification to construct visual syllables as follows. Given the per-frame classification (middle row of Figure 3), we mark a sequence

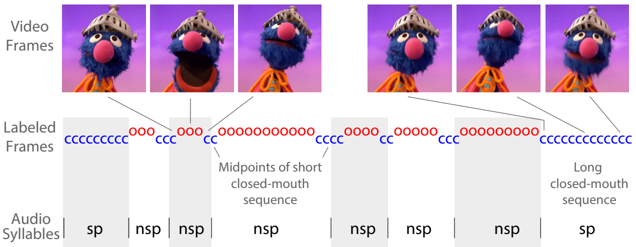
Video Frames
Labeled Frames
ccccccccc ooo ccc ooo cc ooooooooooo cccc oooo cc ooooo ccc ooooooooo ccccccccccccc
Midpoints of short closed-mouth sequence
Long closed-mouth sequence
Audio Syllables
sp
nsp
nsp
nsp
nsp
nsp
nsp
sp

Figure 3: *Converting a sequence of open-mouth (red), closed-mouth (blue), and invalid labels into visual syllables. A sequence of five or more consecutive* closed-mouth *frames is deemed a silent visual syllable. All the remaining valid frames are grouped into non-silent visual syllables. Boundaries between non-silent visual syllables are set in the middle of the remaining short* closed-mouth *sequences. The sequence shown contains two silent syllables (sp) and six non-silent syllables (nsp).*

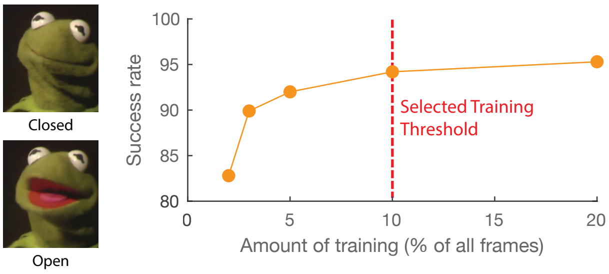

Figure 4: *For appearance-based visual syllables we annotate open and closed mouth frames (left). We also annotate frames with no puppets as invalid (not shown). For a given video we annotate a fraction of the frames and train a neural network to predict missing labels. We experimented with different annotation amounts and found that 10% annotation provides a good trade-off between prediction accuracy and annotation time (right).*

of five or more consecutive *closed-mouth* frames as a silent visual syllable. All the remaining valid frames are grouped into non-silent visual syllables. We set the boundaries between non-silent visual syllables in the middle of the adjacent short *closed-mouth* sequences. Consecutive invalid frames are grouped into invalid sequences, and are not used in later steps (their matching cost is set to infinity).

We created an annotation UI that allows quick keyboard-based annotation. We provide hotkeys for the 3 labels and for frame navigation, requiring one keystroke per annotated label while allowing the annotator to revisit and change previous labels. The average annotation rate using our UI is 200 frames per minute. Thus, for a 1 minute video running at 30fps annotating the complete video would take about 9 minutes. While it is possible to use this interface to manually label short videos, for longer videos we have developed a machine learning approach that significantly lightens the annotation workload.

While researchers have developed a number of facial landmark point detectors for human faces [RPC17,SLC09] these methods are not designed to handle puppets. Therefore, for each puppet video V we train a mouth state classifier to detect *open-mouth*, *closed-mouth* and *invalid* frames. Invalid frames are those without a visible puppet head.

Specifically, we start with the pre-trained GoogLeNet [SLJ*15] model and specialize it to our puppet mouth state detection task. We remove the last three layers of GoogLeNet, which are specific to the original classification task, and replace them with a fully connected layer followed by softmax and a classification output layer that generates one of our three labels. To reduce training time we focus on primarily learning the weights for the new layer; we set the learning rate of the first 110 layers to 0, all other pre-existing layers to 10^{-4} and the new fully connected layer to 10^{-3}.

For training data, we manually annotate a fraction of the frames for a given input video. There is a classification quality vs. annotation effort trade-off in selecting the annotation amount. Figure 4 shows the relationship between the two for a typical video. We find that annotating 10% of the frames is a good threshold allowing a tenfold decrease in annotation time, while maintaining 94% classification accuracy. Using our annotation interface, manually labeling 10% of randomly chosen frames for a 1 minute video (at 30fps) requires 0.9 minutes of annotation time — less than the time it takes to watch the video. We augment the training data by adding random translations and flips, and train for 6 epochs using a batch size of 10. See [supplemental materials: 10% vs. 100% annotation] for results using fully annotated and partially annotated videos. For many examples the results are comparable, with errors in classification manifesting as consecutive visual syllables being detected as one, or one syllable being split into several.

As an optional step, users can correct errors in the classification results using our annotation interface. We find that in practice correcting errors requires watching the video annotated with the labels output by our classifier, pausing it whenever an error is spotted, stepping back a few frames and relabeling around the error. Such correction requires an additional annotation time of about 2 minutes per 1 minute of video, but improves the labels, making them indistinguishable from manual annotation of the complete video.

4.2.2. Audio-based visual syllables.

Our audio-based algorithm assumes that visual syllables in V are closely aligned with the original speech audio track in V. Therefore we can segment the video into visual syllables by first segmenting its original audio track using the approach described in Section 4.1. We then treat the resulting audio syllables as a proxy for the visual syllables of the video.

4.2.3. Comparison.

Table 1 compares our appearance- and audio-based visual syllable extraction approaches. Appearance-based visual syllables are robust to noisy audio and do not rely on good audio-video alignment in the original performance. Appearance-based visual syllables also distinguish between on- and off-camera speech (i.e. if a character is turned away from the camera and speaking, appearance-based syllables will not use it as an audible syllable). Audio-based visual syllables use audio as a proxy for the location of visual syllables.

	Apearance	Apearance with manual correction	Audio
Mean annotation time	0.9x	3x	None
Robust to noisy audio & background music	✓	✓	✗
Robust to audio-video misalignment	✓	✓	✗
Distinguishes on- and off-camera speech	✓	✓	✗
Robust to low-quality video & occlusions	✗	✗	✓

Table 1: *Comparison between appearance-based and audio-based visual syllables. Annotation time (averaged across annotators and media assets) compares between length of media and annotation time. E.g. 3x is 3 minutes annotation time per 1 minute of video. Network training (for appearance-based methods) took 9 minutes for our shortest sequence and 2 hours for the longest.*

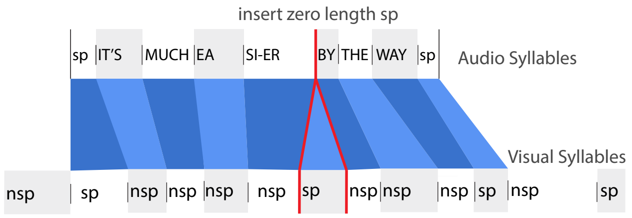

Figure 5: *Alignment. For a sequence of audio syllables* $(a_1,\ldots,a_n)$ *(top) we find the best starting point within the visual syllable* $(v_1,\ldots,v_m)$ *(bottom) such that silence syllables align as much as possible and the difference in syllable lengths is minimized. After finding this starting point we make sure both sequences have the same number of syllables by adding zero length silence syllables opposite any unmatched silence (marked in red).*

They are robust to low quality video and occlusions and do not require any manual annotation. As we will discuss in Section 5.3, The complementary properties of these two approaches make each of them best suited to different dubbing tasks.

4.3. Step 3: Align Audio Syllables to Visual Syllables

The goal of the alignment step is to align the speech syllables to an optimal sequence of visual syllables in the puppet video, such that when possible (1) silence matches to silence, (2) non-silence matches to non-silence and (3) syllable lengths are as similar as possible. Conditions (1) and (2) ensure that when the new speech track is silent, the puppet's mouth is closed, and when there is speech, the puppet's mouth is moving. Condition (3) minimizes the amount of retiming required in Step 4 to match the timing of the visual syllables to the speech, thereby reducing artifacts.

Formally, given A, V and their respective starting indices i, j, we can recursively define the distance $\mathcal{D}(A_i,V_j)$ between the sequences (Equation 1), using $A_i=(a_i\ldots a_n)$ and $V_j=(v_j\ldots v_m)$ as shorthand for the syllable subsequences starting at i and j respectively.

$$\mathcal{D}(A_i,V_j)=\begin{cases}\mathcal{D}(A_{i+1},V_{j+1}) & a_i=sp, v_j=sp\\ \big|\|a_i\|-\|v_j\|\big|+\mathcal{D}(A_{i+1},V_{j+1}) & a_i\neq sp, v_j\neq sp\\ w*\|a_i\|+\mathcal{D}(A_{i+1},V_j) & a_i=sp, v_j\neq sp\\ w*\|v_j\|+\mathcal{D}(A_i,V_{j+1}) & a_i\neq sp, v_j=sp\end{cases} \quad (1)$$

We use $\|a\|$ to indicate the length of syllable a. This equation is defined for $i\leq|A|$ and $j\leq|V|$, and equals 0 when $i>|A|$ or $j>|V|$. The first term incurs no penalty when speech silence matches with visual silence, in accordance with condition (1). The second term penalizes the difference in length of the non-silent syllables, in accordance with conditions (2) and (3). The third and fourth terms heavily penalize alignment of silence to non-silence by adding w times the length of silence. We empirically set w to 1. Notice that in these two terms, only the sequence containing the silent syllable advances. Therefore, we terminate this recursion only when the distance considers an equal number of non-silent syllables in the speech and audio. Equation 1 can be interpreted as a variant of dynamic time warping [SC78], in which insertions and deletions are only permitted for specific types of syllables, and both syllable length and type are used to determine the cost of operations.

We iterate over all possible starting points of V to find the best starting point j, for which $\mathcal{D}(A_1,V_j)$ is minimal. The starting point defines a video sub-sequence $(v_j,\ldots,v_k)$ which best matches the audio $(a_1,\ldots,a_n)$ (Figure 5). These sequences have the same number of non-silent syllables, but may include a different number of silences. For ease of annotation we equalize the number of syllables. Beginning at the starting point, we jointly iterate the two sequences. Whenever one sequence contains silence and the other contains a non-silence, we add a zero length sized silent syllable to produce a matching silent pair. After this syllable equalization, we are left with syllable sequences $(a_1,\ldots,a_n)$ and $(v_j,\ldots,v_{j+n-1})$ where a_1 matches v_j, a_2 matches v_j+1, etc.

4.4. Step 4: Re-time Audio and Video

Given the source audio and target video, along with their matching syllable sequences $(a_1,\ldots,a_n)$, $(v_j,\ldots,v_{j+n-1})$, we must retime each matching pair of syllables so that they are the same length. We first explain how we retime audio and video syllables individually, and then describe our strategy for combining these two methods to minimize visual and audible artifacts.

4.4.1. Audio retiming

Given an audio syllable a_i and a target length L, we speed up or slow down the audio such that $\|a_i\|=L$. We use Waveform Similarity Overlap-add (WSOLA) [VR93] as implemented in Matlab by TSM-Toolbox [DM14] to retime each audio syllable. WSOLA produces a waveform by maximizing local similarity between the generated result and the original waveform in corresponding neighborhoods, as measured by short-time Fourier transform representations. We have also experimented with phase vocoders [FG66], Harmonic-Percussive Separation [DME14] and the commercial Elastique algorithm [zpl18], but found that WSOLA works best for our goal of retiming short speech segments corresponding to syllables. Other methods excelled at retiming music or longer speech segments.

4.4.2. Video retiming

Given a visual syllable v_i and a target length L, our goal is to retime v_i such that $\|v_i\| = L$. Our approach is to treat the original frames of the syllable as samples of a continuous time-varying function parameterized over the time segment $[0,1]$. We then resample L_f evenly spaced frames in $[0,1]$, where L_f is the target length expressed as a number of frames (i.e. if L is in seconds, L_f is set to L times the video framerate). Since resampled frames may not lie on frame boundaries we use either nearest-neighbor sampling (for videos with rapid motion or low frame-rate) or optical flow interpolation [Far03] to generate them. Also note that we assume L_f is an integer value, a property we enforce in Section 4.4.3.

4.4.3. Combined audio-video retiming

Retiming artifacts can appear as speedup/slowdown in the audio or as blur/choppiness in the video. Such artifacts become more prominent as the retiming factor increases. Our approach is to prevent extreme retiming of either the video or the audio by trading off retiming in one channel (audio or video) for retiming in the other.

Given a pair of matching syllable sequences $(a_1,\ldots,a_n)$ and $(v_j,\ldots,v_{j+n-1})$ we define the sequence of lengths for each output syllable $(l_1,\ldots,l_n)$. Since the retiming factor is a multiplicative property (e.g. retiming 1 minute to 2 minutes will have similar quality to retiming 10 minutes to 20 minutes, and is very different from retiming 10 minutes to 11 minutes), we set l_i to be the *geometric* mean of $\|a_i\|$ and $\|v_{j+i-1}\|$,

$$l_i = \sqrt{\|a_i\|\|v_{j+i-1}\|} \tag{2}$$

to evenly distribute retiming between audio and video.

Next we calculate the audio retiming factors for the sequence

$$f_i^{\text{aud}} = \frac{l_i}{\|a_i\|} \tag{3}$$

Extreme audio retiming produces audibly disturbing results. Extreme video retiming is also undesirable, however we found that the visual slowdowns or speedups are often preferable to audio artifacts. Thus, we limit the allowed audio retiming factor

$$\hat{f}_i^{\text{aud}} = \max(\min(f_i^{\text{aud}}, \frac{1}{T}), T) \tag{4}$$

Where T is an audio retiming threshold parameter set empirically to 1.3. We avoid abrupt timing changes between consecutive syllables by convolving the sequence of retiming factors $(\hat{f}_1^{\text{aud}},\ldots,\hat{f}_n^{\text{aud}})$ with a box filter of size 3. The video retime factors f_i^{vid} are set to produce the proper combined retiming amount

$$f_i^{\text{vid}} * \hat{f}_i^{\text{aud}} = \frac{\|a_i\|}{\|v_{j+i-1}\|} \tag{5}$$

As a last step, we slightly update the retiming factors to the closest values that yield integer values for L_f. Given the audio and video retiming factor, we apply the methods in Section 4.4.1 and Section 4.4.2 to produce the final result.

5. Results

Our main motivating application is the creation of puppet videos dubbed with new speech content (Figure 6). Our method can also be used to improve synchronization in existing puppetry videos and to facilitate video translation. We collected 9 puppet videos of Kermit (2x), Big Bird (2x), Grover, Miss Piggy, Fozzie Bear, Cookie Monster, Abby Cadabby and one Japanese anime video, ranging in length from 14 seconds to 68 seconds with an average length of 35 seconds. We also collected 55 audio snippets from the internet containing spoken text, movie quotes, jokes, songs and political speech, ranging in length from 2 seconds to 54 seconds with an average length of 8 seconds. We have generated all combinations of results mentioned in the paper (appearance-based/audio-based, with/without manual correction of audio annotation). For representative results with the various puppets, see [supplemental materials: Representative Results].

Figure 6 shows results of video extraction and retiming. Our tool finds a well aligned video sub-sequence (Section 4.3), retimes it (Section 4.4), and produces a final result synchronized to the user's audio recording. We encourage the reader to look at the supplementary materials for video results. Producing a result using a 1 minute video clip with a 10 seconds audio snippet takes less than 10 seconds for nearest-neighbor sampling of frames, and 3 minutes for optical flow based interpolation on a MacBook Pro with a 2.7 GHz Intel Core i7 processor. We used nearest-neighbor sampling for Cookie Monster, as he moves erratically, causing adjacent frames to be dissimilar and unsuitable for optical-flow based interpolation. All other videos use optical flow.

For comparison between fully annotated visual syllables and 10% annotated visual syllables, see [supplemental materials: 10% vs. 100% annotation]. Many 10% results are comparable to the fully annotated version, with errors arising when frames are mislabeled. A mislabeled *open-mouth* may introduce an erroneous *nsp* syllable, which causes a spoken syllable for a closed mouth sequence. A mislabeled *closed-mouth* may split *nsp* in two or change it to *sp*.

For comparison between appearance-based and audio-based results, see [supplemental materials: appearance vs. audio]. Appearance-based results are generally better than audio-based, as evident by our user study (Section 5.2). Our experience in examining the original puppet videos is that puppeteers often do not perfectly align mouth closed-open-closed sequences with the syllables in their speech. The mouth articulations often lag the speech and these misalignments create artifacts for our audio-based approach.

To evaluate the contribution of good audio annotations, see [supplemental materials: Appearance-based, with/without manual correction of phoneme alignment]. We compare between using the automatically generated phoneme alignment (P2FA [YL08, RBM*13]) and a manually corrected alignment. We find that P2FA produces good alignments at the level of words, but is not perfect at the level of phonemes or syllables. These misalignments appear as extra or missing visual syllables.

We also compare against baseline approaches ([supplemental materials:baselines]). Baselines include random selection of clips and an ablation study in which we omit retiming (Section 4.4) from our algorithm. The baseline methods are often completely misaligned with the new speech as mouth movements occur when there is silence in the speech and vice versa. In the no-retiming case our approach simply aligned the audio and visual syllables, so the first syllable in the speech start with a mouth opening, but since there is no re-timing the video and audio quickly fall out of synchronization with each other.

As far as we know the proposed method is the first to dub puppet videos to new audio, and has no direct comparison in previous literature. Methods such as Video Puppetry [BJS*08] require a puppeteer to explicitly perform, while our method produces the performance from the input audio sequence. Methods such as JALI [ELFS16] require a rigged CG model, while we operate on internet videos. Methods such as Deep Video Portraits [KGT*18] require a driving video to control the result. Lastly, the method of Suwajanakorn et al. [SSKS17] solves the arguably harder task of puppeteering human heads from audio, but requires hours of training video. In contrast, we only require one video that can be a few seconds long.

Other applications. In addition to dubbing a puppet video with new audio, our methods can also be used to improve existing puppet performances and for dubbing a performace in a different language. We show an example from Sesame Street in which the audio and puppet mouth are not well synchronized [supplemental materials: improve synchronization in existing puppet performances]. Interestingly, we found that such misalignments in the original performance often arise, even in professional productions. We can improve the synchronization of the original performance (at a cost of retiming artifacts) by applying our algorithm on the original video and audio content. Thus, our methods can be used as part of a post-production pipeline to improve puppet performances.

5.1. Algorithmic Evaluation

One goal of our method (Section 4) is to minimize extreme speed-ups and slow-downs due to large differences in syllable lengths between the aligned audio and video. We can directly measure these artifacts by calculating the average length ratio between audio and visual syllables (Table 2). We compare between our alignment procedure applied to appearance-based syllables and audio-based syllables. As a baseline, we also compute a random alignment where we randomly select a sequence of appearance-based syllables in the video with the same number of syllables as in the new speech. We obtain less squash/stretch distortion compared to the baseline, even though our alignment has other constraints – not just amount of stretch and squash, but also the fact the silent and non silent regions should align. Our appearance-based method outperforms the audio-based method in stretch/squash minimization.

5.2. User Study

We conduct a user study to evaluate and compare our results. We evaluate the following conditions:

1. ***Appearance 10% + Correction.*** Our appearance-based method with 10% annotation and manual correction of wrong labels.
2. ***Appearance 10%.*** Our appearance-based method with 10% annotation, without manual correction of wrong frame labels.
3. ***Audio.*** Our audio-based method.
4. ***No-retiming.*** Our appearance-based method with 10% annotation but without retiming.

	Random	Ours (audio)	Ours (appearance)
Mean length ratio (aud > vid)	3.70	4.25 (n=612)	2.92 (n=590)
Mean length ratio (vid > aud)	3.13	2.32 (n=615)	2.27 (n=568)
Total	6.83	6.57	**5.19**

Table 2: *Average distortion amount per syllable. We calculate the average ratio between audio and video syllable length. Large values cause speedup/slowdown in the audio and blur/choppiness in the video. We out-perform a random selection of a video starting point, even though such random selection does not need to accommodate other constraints, such as matching silence and non-silence respectively.*

5. ***Random.*** Random selection of a video subsequence to match with the new audio.

We compare to baseline approaches and not to, e.g., human dubbing methods (Section 2) because those require human specific algorithms, a rigged model, a reference motion, or some combination of the above, which are not available for our scenario. We use 5 audio-video pairs and generate the 5 conditions for each, resulting in 25 videos. We asked 6 people to rate each result on a 7-point Likert scale, ranging from 1-extremely bad quality to 7-extremely good quality, for a total of 150 ratings. The participants were free to view the videos in any order and to view a video multiple times. Figure 7 shows the study results. A Kruskal-Wallis test finds that there is a significant difference between these conditions ($p < 10^{-8}$). We then use Tukey's honestly significant difference procedure for pairwise comparisons and find that all pairs of conditions except 1-2, 2-3, 3-4 and 4-5 are significantly different ($p << 0.02$). These results suggest that our appearance-based methods out-perform the other methods, followed by the audio-based method, our method with no retiming and a random baseline.

5.3. Discussion

Together, our results and evaluations suggest that our appearance-based method generally produces more convincing dubs than our audio-based approach and that both outperform baseline dubbing methods. However our appearance-based method does incur some manual annotation cost as reported in Table 1. Thus, the two methods offer a tradeoff between visual quality and annotation time. We recommend using the appearance-based approach when accuracy is most important and users can spend some effort on annotation. We recommend the audio-based approach when users have no time to annotate the video – e.g. if they need to quickly dub a large number of videos.

Non-human dubbing is a relatively under explored task. In this work, we chose to investigate puppet dubbing, which is in a way the polar opposite of human dubbing — composed of relatively stiff open-closed mouth gestures. We have shown that simple approaches and not a lot of data, coupled with domain specific algorithms, suffice to produce convincing results. There are many other

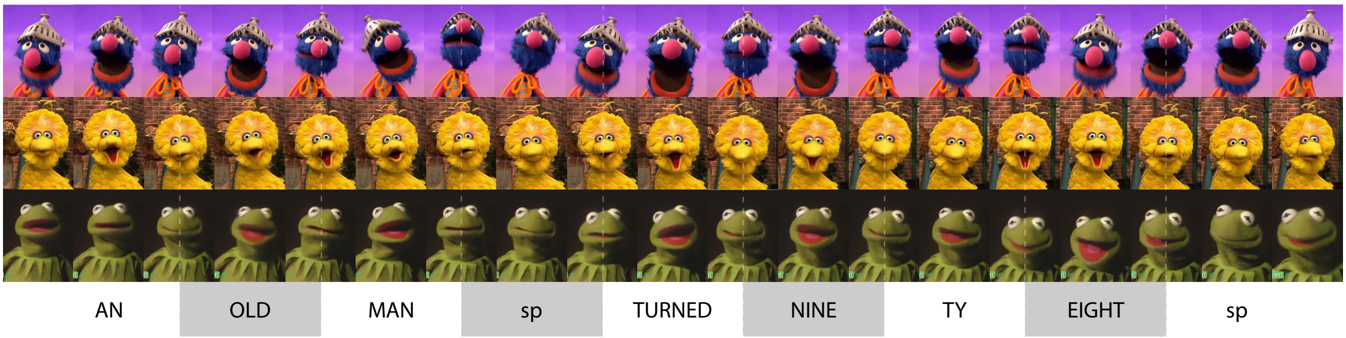

Figure 6: *Given an audio recording and a puppet video, we find the best matching subsequence of the video and produce a retimed result that matches the audio. We show selected frames after our alignment using Kermit, Grover and Big Bird videos. The frames shown are from the start, midpoint and end of each syllable in the new speech (bottom). Notice how the beginning and end of each audible syllable generally corresponds with a closed mouth, and the middle with an open mouth, while silent syllables correspond with a sequence of closed mouth frames. Full videos in supplemental materials.*

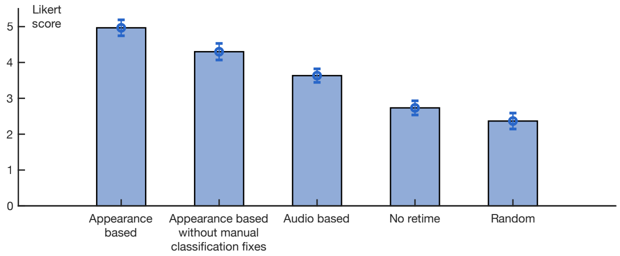

Figure 7: *User study ratings. Our participants rated the appearance-based method with and without manual correction, audio-based method, results without retiming and a random selection of video segments on a 1–7 Likert scale. Bars indicate mean ratings, whiskers are standard error of the mean.*

types of non-human characters. For example, some animations use open-closed cues together with appearance and disappearance of teeth and facial expressions to create more expressive characters. We hope that this work will lead to more non-human dubbing research.

6. Future Work and Conclusion

Our approach has several limitations, suggesting interesting directions for future work.

Automatic puppet mouth state detection. Our appearance-based method requires some manual annotation effort. One direction for future work is to collect a larger set of puppet videos, and use them to train a generic puppet mouth state detectors that can perform well for any unseen puppet videos. Our audio-based method relies on good alignment between the syllables in the original audio track and the puppet's closed-open-closed mouth states. While we have found in practice that most puppet videos are not perfectly aligned, there may be enough alignment to serve as weak supervision to build a puppet mouth state detector without any manual labeling.

Background and camera motion. Our current method does not analyze the background behind the puppet. While we found this to work for most puppet videos, some videos contain regular background motion. For such videos the retiming procedure will produce irregular motion, which can be visually disturbing (e.g. a smooth camera pan can become choppy). Adding background analysis to the matching procedure will alleviate such artifacts, at a cost of more computation. We can also use object segmentation to separate foreground from background. This will allow us to retime puppets and compose the results onto arbitrary backgrounds, but may introduce new artifacts due to segmentation and compositing.

Generalization to cartoonized human faces. Our work is puppet specific and relies on guidelines from expert puppeteers on how to give convincing performances. It is not designed to work on human faces, which are far more expressive than puppets ([supplemental materials: human heads]). We would like to investigate this space between our method and existing human speech manipulation methods [BCS97,SSKS17]. For example, some animated characters rely on mouth openness, but also on mouth contents (teeth visibility) and gestural appendage movements to more expressively convey speech and emotion. We plan to investigate these and other examples in the under-explored spectrum between puppets and humans.

Despite these limitations our tools offer new ways to quickly create dubbed puppet videos. As the amount of video available online increases we believe that such remixing techniques will become more commonplace as they offer efficient approaches to produce high-quality visual stories.

References

[Ado18] ADOBE: Adobe character animator, 2018. [Online; accessed 29-May-2018]. URL: https://helpx.adobe.com/after-effects/character-animator.html. 2

[AECOKC17] AVERBUCH-ELOR H., COHEN-OR D., KOPF J., COHEN M. F.: Bringing portraits to life. *ACM Trans. Graph. 36*, 6 (Nov. 2017), 196:1–196:13. doi:10.1145/3130800.3130818. 2

[BCS97] Bregler C., Covell M., Slaney M.: Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 353–360. 2, 8

[BHB*11] Beeler T., Hahn F., Bradley D., Bickel B., Beardsley P., Gotsman C., Sumner R. W., Gross M.: High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. 30*, 4 (July 2011), 75:1–75:10. doi:10.1145/2010324.1964970. 2

[BJS*08] Barnes C., Jacobs D. E., Sanders J., Goldman D. B., Rusinkiewicz S., Finkelstein A., Agrawala M.: Video puppetry: a performative interface for cutout animation. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 124. 2, 7

[BKC09] Bartlett S., Kondrak G., Cherry C.: On the syllabification of phonemes. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2009), NAACL '09, Association for Computational Linguistics, pp. 308–316. 3

[BLA12] Berthouzoz F., Li W., Agrawala M.: Tools for placing cuts and transitions in interview video. *ACM Trans. Graph. 31*, 4 (July 2012), 67:1–67:8. doi:10.1145/2185520.2185563. 2

[Boe01] Boersma P.: Praat, a system for doing phonetics by computer. *Glot International 5*, 9/10 (2001), 341–345. 3

[Bra99] Brand M.: Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 21–28. 2

[CBZB15] Cao C., Bradley D., Zhou K., Beeler T.: Real-time high-fidelity facial performance capture. *ACM Trans. Graph. 34*, 4 (July 2015), 46:1–46:9. doi:10.1145/2766943. 2

[Cur87] Currell D.: *The complete book of puppet theatre*. Barnes & Noble Imports, 1987. 2

[DM14] Driedger J., Müller M.: TSM Toolbox: MATLAB implementations of time-scale modification algorithms. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)* (Erlangen, Germany, 2014), pp. 249–256. 5

[DME14] Driedger J., MÃijller M., Ewert S.: Improving time-scale modification of music signals using harmonic-percussive separation. *IEEE Signal Processing Letters 21*, 1 (Jan 2014), 105–109. doi:10.1109/LSP.2013.2294023. 5

[ELFS16] Edwards P., Landreth C., Fiume E., Singh K.: JALI: An animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 127. 2, 7

[Far03] Farnebäck G.: Two-frame motion estimation based on polynomial expansion. In *Image Analysis* (Berlin, Heidelberg, 2003), Bigun J., Gustavsson T., (Eds.), Springer Berlin Heidelberg, pp. 363–370. 6

[FG66] Flanagan J. L., Golden R.: Phase vocoder. *Bell Labs Technical Journal 45*, 9 (1966), 1493–1509. 5

[FJA*14] Fyffe G., Jones A., Alexander O., Ichikari R., Debevec P.: Driving high-resolution facial scans with video performance capture. *ACM Trans. Graph. 34*, 1 (Dec. 2014), 8:1–8:14. doi:10.1145/2638549. 2

[FKSM16] Furukawa S., Kato T., Savkin P., Morishima S.: Video reshuffling: Automatic video dubbing without prior knowledge. In *ACM SIGGRAPH 2016 Posters* (New York, NY, USA, 2016), SIGGRAPH '16, ACM, pp. 19:1–19:2. doi:10.1145/2945078.2945097. 2

[FTZ*19] Fried O., Tewari A., Zollhöfer M., Finkelstein A., Shechtman E., Goldman D. B., Genova K., Jin Z., Theobalt C., Agrawala M.: Text-based editing of talking-head video. *ACM Trans. Graph. 38*, 4 (July 2019), 68:1–68:14. doi:10.1145/3306346.3323028. 2

[IBM16] IBM: IBM Speech to Text Service. https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/speech-to-text/, 2016. Accessed 2016-12-17. 2, 3

[JMD*17] Jin Z., Mysore G. J., Diverdi S., Lu J., Finkelstein A.: Voco: Text-based insertion and replacement in audio narration. *ACM Trans. Graph. 36*, 4 (July 2017), 96:1–96:13. doi:10.1145/3072959.3073702. 2

[Jon11] Jones D.: *Cambridge English pronouncing dictionary*. Cambridge University Press, 2011. 3

[KAL*17] Karras T., Aila T., Laine S., Herva A., Lehtinen J.: Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Trans. Graph. 36*, 4 (July 2017), 94:1–94:12. doi:10.1145/3072959.3073658. 2

[KGT*18] Kim H., Garrido P., Tewari A., Xu W., Thies J., Niessner N., Pérez P., Richardt C., Zollhöfer M., Theobalt C.: Deep Video Portraits. *ACM Transactions on Graphics (TOG)* (2018). 2, 7

[LDTA17] Leake M., Davis A., Truong A., Agrawala M.: Computational video editing for dialogue-driven scenes. *ACM TOG 36*, 4 (July 2017), 130:1–130:14. doi:10.1145/3072959.3073653. 3

[OH16] Ochshorn R., Hawkins M.: Gentle: A Forced Aligner. https://lowerquality.com/gentle/, 2016. Accessed 2018-09-25. 2, 3

[PGHA16] Pavel A., Goldman D. B., Hartmann B., Agrawala M.: Vidcrit: Video-based asynchronous video review. In *Proc. of UIST* (2016), ACM, pp. 517–528. 2

[PLH*15] P. G., L. V., H. S., I. S., K. V., P. P., C. T.: Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. *Computer Graphics Forum 34*, 2 (2015), 193–204. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12552, doi:10.1111/cgf.12552. 2

[PRHA14] Pavel A., Reed C., Hartmann B., Agrawala M.: Video digests: A browsable, skimmable format for informational lecture videos. In *Proc. of UIST* (2014), pp. 573–582. 2

[RBM*13] Rubin S., Berthouzoz F., Mysore G. J., Li W., Agrawala M.: Content-based tools for editing audio stories. In *Proc. of UIST* (2013), pp. 113–122. 2, 3, 6

[RPC17] Ranjan R., Patel V. M., Chellappa R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence PP*, 99 (2017), 1–1. doi:10.1109/TPAMI.2017.2781233. 4

[SBLD15] Shin H. V., Berthouzoz F., Li W., Durand F.: Visual transcripts: Lecture notes from blackboard-style lecture videos. *ACM Trans. Graph. 34*, 6 (2015), 240:1–240:10. 2

[SC78] Sakoe H., Chiba S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 26*, 1 (February 1978), 43–49. doi:10.1109/TASSP.1978.1163055. 5

[SLC09] Saragih J. M., Lucey S., Cohn J. F.: Face alignment through subspace constrained mean-shifts. In *2009 IEEE 12th International Conference on Computer Vision* (Sept 2009), pp. 1034–1041. doi:10.1109/ICCV.2009.5459377. 4

[SLD16] Shin H. V., Li W., Durand F.: Dynamic authoring of audio with linked scripts. In *Proc. of UIST* (2016), pp. 509–516. 2

[SLJ*15] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.: Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015). 4

[SSKS17] Suwajanakorn S., Seitz S. M., Kemelmacher-Shlizerman I.: Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph. 36*, 4 (July 2017), 95:1–95:13. doi:10.1145/3072959.3073640. 2, 7, 8

[TBLA16] TRUONG A., BERTHOUZOZ F., LI W., AGRAWALA M.: Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (New York, NY, USA, 2016), UIST '16, ACM, pp. 497–507. doi:10.1145/2984511.2984569. 2

[TKY*17] TAYLOR S., KIM T., YUE Y., MAHLER M., KRAHE J., RODRIGUEZ A. G., HODGINS J., MATTHEWS I.: A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 93. 2

[TMTM12] TAYLOR S. L., MAHLER M., THEOBALD B.-J., MATTHEWS I.: Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics Conference on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2012), EUROSCA'12, Eurographics Association, pp. 275–284. doi:10.2312/SCA/SCA12/275-284. 2

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

[VR93] VERHELST W., ROELANDS M.: An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing* (April 1993), vol. 2, pp. 554–557 vol.2. doi:10.1109/ICASSP.1993.319366. 5

[WBLP11] WEISE T., BOUAZIZ S., LI H., PAULY M.: Realtime performance-based facial animation. *ACM Trans. Graph. 30*, 4 (July 2011), 77:1–77:10. doi:10.1145/2010324.1964972. 2

[WHS12] WANG L., HAN W., SOONG F. K.: High quality lip-sync animation for 3d photo-realistic talking head. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on* (2012), IEEE, pp. 4529–4532. 2

[YL08] YUAN J., LIBERMAN M.: Speaker identification on the scotus corpus. *The Journal of the Acoustical Society of America 123*, 5 (2008), 3878–3878. arXiv:https://doi.org/10.1121/1.2935783, doi:10.1121/1.2935783. 3, 6

[You08] YOUTUBE - MICHAEL HUTZ: Bert and ernie, pesci and deniro, casino, 2008. [Online; accessed 29-May-2018]. URL: https://www.youtube.com/watch?v=NShgvtEro7I. 2

[You10] YOUTUBE - TVLESSONDOTCOM: How to make a puppet lip sync properly, 2010. [Online; accessed 18-January-2018]. URL: https://www.youtube.com/watch?v=_3T7IfdynVw. 3

[You14a] YOUTUBE - CINEMASINS JEREMY: Tarantino movies with muppets, 2014. [Online; accessed 18-January-2018]. URL: https://www.youtube.com/watch?v=644e0h4qyb0. 2

[You14b] YOUTUBE - ISTHISHOWYOUGOVIRAL: I muppet - auguriiiii di buon anno - muppets luca, 2014. [Online; accessed 29-May-2018]. URL: https://www.youtube.com/watch?v=aMBsU3b58fc. 2

[You14c] YOUTUBE - JAMIE HUNTER: Muppets dubbing side-by-side, 2014. [Online; accessed 29-May-2018]. URL: https://www.youtube.com/watch?v=2Hdl7lqQwsQ. 2

[You14d] YOUTUBE - JORDAN LAWRENCE: The voice jamaican kermit the frog, 2014. [Online; accessed 29-May-2018]. URL: https://www.youtube.com/watch?v=rro8RoPjraM. 2

[You16] YOUTUBE - ISTHISHOWYOUGOVIRAL: Warren g ft. nate dogg | regulate | sesame street version, 2016. [Online; accessed 29-May-2018]. URL: https://www.youtube.com/watch?v=aYD3gLCXXuU. 2

[zpl18] ZPLANE: zplane elastique, 2018. [Online; accessed 29-May-2018]. URL: http://licensing.zplane.de/technology#elastique. 5