FSM and K-Nearest-Neighbor for Corpus based Video-Realistic Audio-Visual Synthesis

Christian Weiss

Institute for Communication Research & Phonetics (IKP)
University of Bonn, Germany
cwe@ikp.uni-bonn.de

Abstract

In this paper we introduce a corpus based 2D video-realistic audio-visual synthesis system. The system combines a concatenative Text-to-Speech (TTS) System with a concatenative Text-to-Visual (TTV) System to an audio lip-movement synchronized Text-to-Audio-Visual-Speech System (TTAVS). For the concatenative TTS we are using a Finite State Machine approach to select non-uniform variable-size audio segments. Analogue to the TTS a k-Nearest-Neighbor algorithm is applied to select the visual segments where we perform image filtering previous to the selection process to extract features which are used for the Euclidian distance measure to minimize distortions while concatenating the visual segments. We consider only the particular start-frame and end-frame between potential video-frame sequences for the Euclidian metric. The selection of the visual equivalence of the selected segments is based on a visemic transcription according to the phonemic transcription of the given input text. Due to using independent source databases for speech and video we synchronize the generated signals in a linear way. The resulting audio-visual utterance is audio lip-movement synchronized audio-visual speech. The system is adaptable easily to new speakers whether using a different speech or video source.

1. Introduction

For more than a decade, researchers have been exploring and experimenting with the relationship between speech and facial expressions that correspond to articulation. McGurk and McDonald [12] have shown that the perception of speech by a person not only depends on acoustic cues, but also on visual cues such as lip movements. In this manner [11] for example showed in his work that the facial gesture plays a major role in face-to-face-communication and in Human-Machine-Interaction by improving the intelligibility of a spoken utterance especially in noisy environments (also known as the cocktail party effect) or for hearing impaired people. Because of the importance of the face and speech in all human-to-human and human-to-machine interactions, this research is considered essential in numerous areas such as speech-reading education, E-commerce, customer relations as well as e-care services. For this reason audio-visual synthesis becomes more and more interesting to research and industry.

In this paper we show our approach building a corpus based video-realistic “Talking Head” that is based on non-uniform variable sized speech and video segment selection for generating audio lip-movement synchronized audio-visual speech. So far two major approaches are currently deployed: the model-based approach of building a Talking Head such as Baldi and Synface [17, 20] and the data-driven audio-visual approach [8] which is based on the well known unit-selection algorithm [4] and photo-realistic image sequences. So far though, there has hardly been any research or success in creating a German video-realistic audio-visual synthesis. For this reason, a framework was developed [14, 15] to produce a corpus based 2D video-realistic audio-visual “Talking Head”, which can be used as a Human-Machine Interface. Our can be seen as a variation of the Video Rewrite algorithm by [5] and the photo-realistic unit selection approach by [8] but using a different database as well as new segment selection algorithms. The Video Rewrite algorithm uses viseme-triphone mapping sequences and stores them in a database to retrieve viseme–triphones when a corresponding sequence is required for the generation of the visual utterance of underlying speech. While generation the facial parts are placed in the according frames of the video sequence. The photo-realistic unit selection algorithm by [8] extracts mouth and eye regions of the speaker and fits them into a base image of the speakers head. To create an appropriate mouth movement of the spoken utterance, the mouth and eye region units which fit best in the base image according to the phonetic transcription, are selected from a database.

In distinction to these two approaches, we extract sequences not only of facial regions like mouth or eyes but use the recorded speaker as a video-frame sequence with no preprocessed manipulation of the video source analogue to our concatenative TTS system. Our speech synthesis follows the unit-selection algorithm by [4] but uses a FSM to select the appropriate speech segments for audio signal generation. K-Nearest-Neighbor is used to select the visual segments according to the visemic transcription of the given input text. We select those potential visual segments which minimizes our concatenation costs computed by a Euclidian distance measure where we address the metric to facial features. The attributes for distance measure are extracted by applying image filtering to the start-frame and the end-frame of the corresponding video sequence where first a luminance and afterwards a Kirsch edge detection filter are applied. A pixel-based search provides us the eye-brows, nostrils and mouth region from where we extract coordinates. To synchronize the lip-movement with the audio stream, we compute a transition factor between the single 2D-video-frames and the speech segments to control the audio-lip synchrony in a linear way.

This paper is organized as follows. In section two we introduce our system and describe the used corpora. In section three we show how our speech segment selection works and in section four we explain our video segment selection for concatenation. Further on in section five we describe the
audio-video synchronization and end up with a conclusion in section six.

2. System Overview

Our audio-visual synthesis system consists of two parts. In Figure 1 we show a schematic overview of our online-module where any given text can be insert to generate an audio-visual utterance. For building our databases we use an offline processing module which provides us all the information we need during runtime. Please see section 2.1 and 2.2 for details.

2.1. Audio and Video Corpora

Our applied corpora consist of recorded speech and recorded video. The speech data comprises about 3500 sentences of spontaneous speech. The recorded video corpus contains one hundred sentences, one hundred word bigrams and 100 triphone- and diphone-sequences extracted of the speech corpus text. The recorded video has the format PAL 25 fps, 720 x 576 pixels and has been stored as an uncompressed AVI file. The speaker was told to read the sentences in a news announcer style moving the head as less as possible. We recorded the speaker in front of a neutral blue background to simplify fast and robust facial feature position detection.

2.2. Data Segmentation and Automatic Labeling

The initial step to prepare the recorded audio-visual data corpus to setup our database is to label the data. We are using a semi automatic approach where the preprocessed data is being manually corrected afterwards. The audio and video streams are identified in the video and are automatically extracted from the recorded video corpus.

2.2.1. Speech Database

Our speech database is phonetically transcribed speech where we labeled the speech hierarchically top-down from sentence, word, and syllable, phone to half-phone. The acoustical features are F0 and Mel-Cepstrum Coefficients. We also include context information, linguistic and phonological information. The data is organized in an xml-style format.

2.2.2. Video Database

By using a HTK-based [19] audio segmentation, we get the time marks of word and phone boundaries in the extracted audio-stream to segment the video stream and time-label the video-stream according to the spoken utterance. A post processing manual correction is essential to get video segment time marks in a correct way. This is due the fact that a video frame normally lasts 40 milliseconds and for that reason it sometimes does not match the phone duration boundaries. Our phonetic and visemic transcription follows [2,13] where we use a modified version of the Maximum Entropy Tagger developed by [12]. The video data is organized as the speech data in xml-style format. Additionally we keep facial features. Please see section 4.2 for a detailed description of extracting facial features.

3. Finite State Machine for Speech Segment Selection

To synthesize any ascii input text, we use a corpus based unit selection Text-to-Speech synthesis approach. Our system runs in a 3-way process where first symbolic text preprocessing provides a text-normalized, phonetically transcribed text enriched with additionally static quantitative, linguistic and phonological feature information. A CART predicts the dynamic feature duration and F0. According to the phonetic transcription we select the potential speech segments out of our speech database and concatenate the speech segments to complete an utterance. The unit selection follows a finite state machine approach which ends in a Markov-Chain. To keep the naturalness of a speaker’s voice we do not apply any manipulation of the speech signal after selection and concatenation.

3.1. Speech Segment Selection

To select the speech segments out of our speech database we use a Markov-Chain shown in Figure 2 of the potential speech segments and an A* algorithm for shortest-path. Each state of our time-discrete Markov-Chain represents a speech segment which is needed for concatenation at that time corresponding to the phonetic transcription.

Figure 2: Markov-Chain for speech-segment selection

We pre-computed all start probabilities and transition probabilities of the segment sequences as they occur in our database where the segments are represented by their quantitative, prosodoc and phonological features. This is done in a hierarchical way starting with bigrams, bigrams-word, word-syllable, and syllable-diphone and diphone-phone transition. The start probability

\[ H_s(A) = \frac{a_n(A)}{n} \]  

is the frequency of our feature-vector describing the particular speech segment according to all speech segments with the
same label but different features. We cumulate this probability (1) with the probability
\[ H_w(A_z) = \frac{a(A_z)}{m} \]  
(2)
of the label-specific probability (2) according to the occurrence of all labels in the corpus.
Finally we compute the transition probability (3) for each speech segment
\[ P(A_1 | A_2) = \frac{P(A_1 \cap A_2)}{P(A_2)} \]  
(3)
that we can do a graph-search during runtime and select the best state sequence that provides those speech segments which are best for concatenating to an utterance. The overall probability (4)
\[ P(X_1)P(X_2 | X_1)P(X_3 | X_2)\ldots P(X_T | X_{T-1}) \]  
(4)
which we compute doing a inverse shortest-path search with the A* algorithm where the transition probabilities are the distances between the states and the best speech segment sequence is (5):
\[ S = \arg\max_{x_{1:T}} \pi x_1 \prod_{t=1}^{T-1} a_{x_t,x_{t+1}} \]  
(5)
We concatenate the speech segments and apply no signal manipulation post-processing to the utterance.


After we generated our speech signal we have to generate the visual frame-sequence for the video signal that we are using for synthesizing the audio lip-movement synchronized audio-visual utterance. We select the video segments according to the visemic transcription [13] of the input text out of our pre-recorded video database. This selection is applied to variable-sized video segments. That means that the longest video-frame sequence we can find in our database is preferred but we have to take account for the distortion at concatenation point. Following this restriction the last video-frame of the first video segment serves as a reference for the first video-frame of the following video segment concerning the speakers head-position and accordingly the facial reference points to minimize the distortions while concatenation. For a smooth transition between the video segments we use K-Nearest-Neighbor to select the appropriate video segment among the considered ones.

4.1. Image Filtering For Head-Position Determination

Previous to the selection of the video segments we have to determine the position of the head respectively the position of facial region-of-interests in the video-frame that we take into consideration to compute a metric, here the Euclidian distance of consecutive frames. Figure 3 shows the chronological image filtering which we are using for feature extraction.

We first apply a luminance filter to the image and after that a Kirsch edge-detection is processed. When applied the Kirsch edge-detection we do a pixel-based search for the nostrils, and mouth which we can locate in a predefined region-of-interest as seen in Figure 4. Due to the fact that we instruct the speaker to stay in the same position as possible while reading the text-corpus for video recording we have a rough knowledge of this area and can avoid search errors.

4.2. K-Nearest-Neighbor Video Segment Selection

The K-Nearest-Neighbor method uses those observations in our training set which is closest in the input space to our target feature vector to form the according class output (6).
\[ f(x_q) \leftarrow \arg\max_{x \in V} \sum_{i=1}^{3} \delta(v, f(x_i)) \]  
(6)
The target feature vector is in each case the last video-frame of the previous video segment.
As soon as we extracted those features we average the coordinates for each facial mark we take into account.

As we only need the best matching video segment according to minimize the distance between the two consecutive video frames we use a 1-Nearest-Neighbor classifier. Each instance of the according start-frame respectively the according end-frame is a training instance. Given \( x_q \) as target we choose \( x_i \) which is closest to \( x_q \). Our metric is Euclidian distance measure (7)
\[ d(x_q, x_i) = \sqrt{\sum_{i=1}^{3} (a_i(x_q) - a_i(x_i))^2} \]  
(7)
where \( a \) denotes our averaged coordinates of the facial region-of-interest. As soon we have determined the video segment which we want to choose for concatenation we have to extract the video-frame sequence out of the video-stream. The frame sequence extraction is described in (8).
\[ a) \quad T_{\delta} = \sum_{i=0}^{T} \Delta t(S_i) \quad b) \quad F_{\delta} = T_{\delta} \cdot f_{\rho} \quad c) \quad F_{\text{off}} = K_{\delta} + F_{\delta} + K_{\delta} \]  
(8)
First in a) we denote the length \( T_{\delta} \) of the required segment where we get the time of the speech segment \( S_i \) in recording \( S_k \). In b) \( F_{\delta} \) determines the number of frames we have to extract.
while multiplying $T_g$ with the frames per second $f_{ps}$. The frames $F_g$ we want to extract do sometimes not match the speech segment boundary. Due to this fact we introduce frame-constant values $K_L$ for the left-end and $K_R$ for the right-end of the sequence to fill-up the frame-sequence in order to match the length of the speech signal. Once we have extracted all required video segments we concatenate them to a new visual utterance.

5. Audio-Video Synchronization

As we are using independent audio and video source recordings we have to synchronize our synthesized audio-signal with the synthesized video-signal to obtain an audio-lip-movement synchronized audio-visual utterance. The synchronization of audio and video-frame sequence is done linear according to the synthesized speech.

$$T_F = \frac{T_{s_0}}{C_{r_{ps}}}$$  \hfill (9)

As denoted in (9) we compute a transition factor $T_F$ for the frame transition in the video sequence. In a normal video of 25 fps the transition factor is 40 milliseconds between the frames. In the way we decrease or increase this transition factor we are capable to do a linear synchronization according to our video-frame sequence with the underlying synthesized speech.

6. Conclusion

In this paper we described a corpus-based approach to generate video-realistic audio-visual speech synthesis that is synchronized in audio and lip-movement. To synthesize speech we used a probabilistic variable-sized unit-selection approach to select the appropriate speech segments for concatenation. The probabilistic approach follows a Finite State Machine. For the visual synthesis we also refer to variable-sized unit-selection and use a video database to select the video-frames according to the visemic transcription of the given input text. To minimize the distortions at concatenation points we applied the K-Nearest-Neighbor algorithm to select those video-segments which have minimal distance in their features. This metric is computed using the Euclidian distance measure. Due to the fact of using independent source segments we have to synchronize the synthesized audio and synthesized video stream to an audio-visual speech utterance. In a normal video of 25 fps the transition factor is 40 milliseconds between the frames.

7. References

[16] Baldi: <http://cslu.cse.ogi.edu/toolkit/>
[18] SAMPA: <http://www.phon.ucl.ac.uk/home/sampa/german.htm>