SPEECH DRIVEN 3-D FACE POINT TRAJECTORY SYNTHESIS ALGORITHM

Levent M. Arslan and David Talkin

Entropic Research Laboratory, Washington, DC, 20003

ABSTRACT

This paper presents a novel algorithm which generates three-dimensional face point trajectories for a given speech file with or without its text. The proposed algorithm first employs an off-line training phase. In this phase, recorded face point trajectories along with their speech data and phonetic labels are used to generate phonetic codebooks. These codebooks consist of both acoustic and visual features. Acoustics are represented by Line spectral frequencies (LSF), and face points are represented with their principal components (PC). During the synthesis stage, speech input is rated in terms of its similarity to the codebook entries. Based on the similarity, each codebook entry is assigned a weighting coefficient. If the phonetic information about the test speech is available, this is utilized in restricting the codebook search to only several codebook entries which are visually closest to the current phoneme (a visual phoneme similarity matrix is generated for this purpose). Then these weights are used to synthesize the principal components of the face point trajectory. The performance of the algorithm is tested on held-out data, and the synthesized face point trajectories showed a correlation of 0.73 with true face point trajectories.

1. INTRODUCTION

There has been a significant interest in the area of face synthesis recently. This topic has numerous applications including film dubbing, computer-based language instruction, cartoon character animation, multimedia entertainment, etc. There is a large effort in developing autonomous software agents that can communicate with humans using speech, facial expression, gestures, and intonation. Katashi and Akikazu [6] employed animated facial expressions in a spoken dialogue system. Other researchers [3, 4]) used various forms of visual agents animating gestures, intonation, and head movements. Lip synching is another application of wide interest. Video Rewrite system [5] uses existing footage to create automatically new video of a person mouthing words that she did not speak in the original footage.

In this study, we propose a new algorithm to synthesize three dimensional face point trajectories corresponding to a novel utterance. The general algorithm does not require any text input. However, the performance of the algorithm significantly improves if phonetic information is known a priori. Therefore, throughout this paper the algorithm will be described assuming phonetic information is available. It will be described in the end how the proposed algorithm can be to the case where phonetic information is not available. The general outline of the paper is as follows. Section 2 describes the proposed face point trajectory synthesis algorithm. In this section, the formulation and automatic generation of a novel visual phoneme similarity matrix is described as well. Section 3 presents the simulations and performance evaluation. Finally Section 4 discusses the results and future directions.

2. ALGORITHM DESCRIPTION

The face synthesis algorithm proposed in this paper is an extension of the STASC voice transformation algorithm which is described in [1]. The flowchart of the proposed face synthesis algorithm is shown in Figure 1. The algorithm requires two on-line inputs: i) a speech file, ii) its corresponding phoneme sequence. It also requires two additional inputs which are generated prior to face synthesis during the training stage: i) an audio-visual codebook, ii) a visual phoneme similarity matrix. First, we will explain how the codebook, and the visual phoneme similarity matrix are generated.

2.1. Audio-Visual Codebook Generation

For the data collection, first synchronized speech and face point trajectories must be recorded from a subject. For this study the point trajectories were recorded using a multi-camera triangulation system yielding 60 samples/sec at a spatial resolution of .254 mm in X, Y, and Z. In the pilot study reported here, 54 points on and around the face were recorded while a single subject uttered approximately 300 TIMIT sentences selected to provide the richest possible phonetic coverage. Unfortunately, tongue movement was not included in the dataset. Speech and EGG (Glottal Enterprises) were also digitized via a DAT recorder at 48kHz, then later digitally down-sampled to 16kHz for more compact storage. In order to model the acoustic and visual features that correspond to the subject talker an audio-visual codebook is used.

Acoustic features used in the codebook are line spectral frequencies (LSF) which provide a compact representation of the speech signal. They have a number of nice properties which make them attractive among speech researchers especially in the speech coding area. The relation of LSFs to visual features have been investigated by Yehia et. al. [7]. They found that 91% of the total variance observed in the orofacial data was accounted for by LSFs.

The visual features are principal components of 162 dimensional face point parameter vector. The principal com-

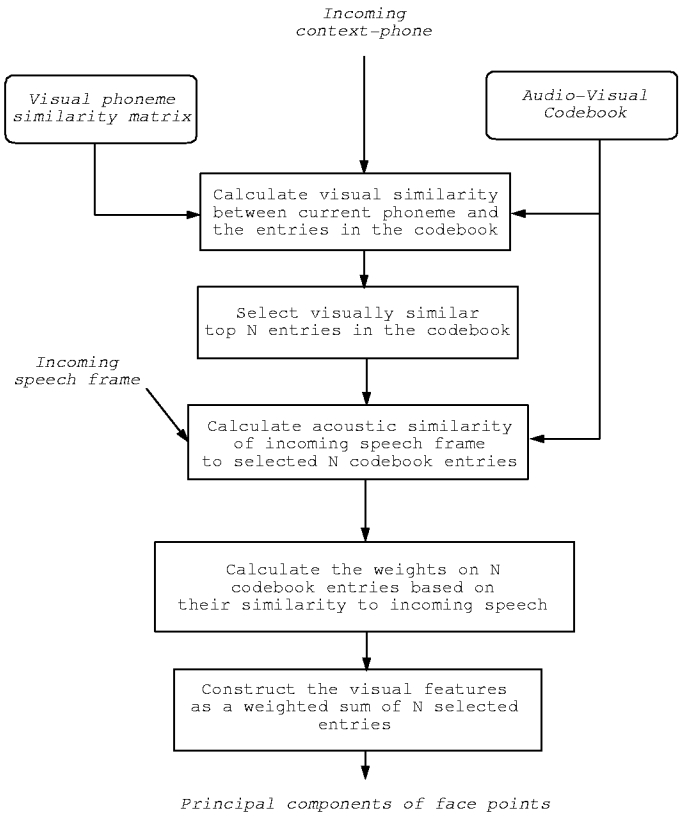

Figure 1: Flow-diagram of the proposed face synthesis algorithm.

ponents can be obtained using the Karhunen-Loeve transformation technique. Since the movements of points on the face are highly correlated, a significant dimensionality reduction can be achieved with minor distortion. Principal components have been used for various applications in image processing and face animation [5].

Each codebook entry in the audio-visual codebook corresponds to a certain context and it consists of an acoustic feature vector, and a visual feature vector. Associated with each specific context there are 5 codewords corresponding to uniformly spaced locations in time across the duration of the phoneme. The audio-visual codebook entries are generated as follows. First, the speech data is segmented into phonemes. Next, each phoneme is tagged with a symbol which we refer to as "context-phone" which represents the left and right context of the phoneme. After the data is tagged this way, each phoneme is labeled with 5 uniformly spaced time locations. The acoustic and visual features corresponding to those 5 locations are then appended to the audio-visual codebook.

2.2. Automatic Generation of Visual Phoneme Similarity Matrix

Since in practice the training data will not include all possible context-phones, we need a way of associating unseen context-phones with the audio-visual codebook. In this paper, a novel procedure for the automatic selection of closest context-phone is developed. The criterion that we chose for visual similarity of phonemes is based on Euclidean distance of principal components of face data. Therefore, initially an average principal component vector is estimated for each phoneme.

$$\mathbf{m}_k = \frac{1}{T_k}\sum_{t=1}^{T_k}\mathbf{P}_{kt}, \qquad k = 1 \ldots K, \tag{1}$$

where K represents the total number of phonemes present in the language, T_k represents the number of tokens for the k^{th} phoneme, and $\mathbf{P}_{kt}$ represents the t^{th} principal component coefficient vector that is associated with k^{th} phoneme. Then, the Euclidean distance between each phoneme pair is calculated as:

$$\mathbf{D}_{ik} = \|\mathbf{m}_i - \mathbf{m}_k\| \qquad i = 1 \ldots K, \quad k = 1 \ldots K. \tag{2}$$

Finally, a similarity measure is derived from the distances using:

$$\mathbf{S}_{ik} = e^{-\nu \mathbf{D}_{ik}} \qquad i = 1 \ldots K, \quad k = 1 \ldots K. \tag{3}$$

This formulation assures that similarity values, $\mathbf{S}_{ik}$, will range between 0 and 1. The constant ν in the equation can be adjusted to control the dynamic range of similarity values appropriately. In the experiments reported in this study we used a value of 10 for ν. In general, it is observed that the entries in the automatically derived matrix agree with intuitive expectations. However, we have not performed subjective tests to verify this statement yet.

Next, we formulated a procedure to pick visually most similar context-phones to an unseen context-phone. It has been shown that visual confusability depends highly on the context of a phoneme [2]. Therefore, we have taken into account the context of a phoneme when selecting the appropriate context-phones in the codebook. We represent the context-phone as $...l_3\_l_2\_l_1\_c\_r_1\_r_2\_r_3...$, where "$l_n$" represents n^{th} phoneme to the left, "r_n" represents n^{th} phoneme to the right, and c represents the center phoneme. The similarity of a test context-phone to each of the context-phones in the codebook can be formulated as:

$$\mathbf{n}_j = \mathbf{S}_{cj} + \sum_{i=1}^{C}\xi^{-i}\mathbf{S}_{l_{ij}} + \sum_{i=1}^{C}\xi^{-i}\mathbf{S}_{r_{ij}} \quad j = 1 \ldots L \tag{4}$$

where C is the level of context information, L is the total number of context-phones in the codebook, $\mathbf{S}_{cj}$ is the similarity between the center phone of the unseen context-phone and the j^{th} context-phone in the codebook, $\mathbf{S}_{l_{ij}}$ is the similarity between the i^{th} left phoneme of the unseen context-phone and the j^{th} context-phone in the codebook, and $\mathbf{S}_{r_{ij}}$ is the similarity between the i^{th} right phoneme of the unseen context-phone and the j^{th} context-phone in the codebook. Since similarity matrix values range between zero and one, by selecting ξ to be greater than 10 one can assure that center phoneme match will always have the highest precedence in the decision procedure, and as we move away from the center the influence of match will decrease.

The next section describes the face synthesis process using the visual phoneme similarity matrix and the audio-visual codebook.

2.3. Face Synthesis

First, the context-phone which corresponds to the incoming speech frame is compared to available context-phones in the codebook in terms of their visual similarity. Using the similarity metric discussed in the previous section, the top N most similar context-phones are selected in the audio-visual codebook. Next, the acoustic feature vector corresponding to the incoming speech frame is compared to all the LSFs that correspond to the top N context-phones. There will be $5N$ such vectors, since each context-phone is represented with 5 uniformly spaced audio-visual vectors. The incoming LSF vector $\mathbf{w}$ is compared with each LSF vector, $\mathbf{L}_i$, in the codebook and the distance, $\mathbf{d}_i$, corresponding to each codeword is calculated. The distance calculation is based on a perceptual criterion where closely spaced line spectral frequencies which are likely to correspond to formant locations are assigned higher weights.

$$\begin{aligned} \mathbf{h}_k &= \frac{1}{\text{argmin}(|\mathbf{w}_k - \mathbf{w}_{k-1}|, |\mathbf{w}_k - \mathbf{w}_{k+1}|)} \qquad k = 1, \ldots, P \\ \mathbf{d}_i &= \sum_{k=1}^{P} \mathbf{h}_k |\mathbf{w}_k - \mathbf{L}_{ik}| \qquad i = 1, \ldots, 5N \end{aligned} \tag{5}$$

where $5N$ is the reduced codebook size based on context. Based on the distances from each codebook entry, an expression for the normalized codebook weights can be obtained as:

$$\mathbf{v}_i = \frac{e^{-\gamma \mathbf{d}_i}}{\sum_{l=1}^{5N} e^{-\gamma \mathbf{d}_l}} \qquad i = 1, \ldots, 5N \tag{6}$$

This set of weights $\mathbf{v}$ allows us to approximate the original LSF vector $\mathbf{w}$ as a weighted combination of codebook LSF vectors:

$$\hat{\mathbf{w}}_k = \sum_{i=1}^{5N} \mathbf{v}_i \mathbf{w}_{ik} \tag{7}$$

The value of γ in the previous equation is found by an incremental search in the range of 0.2 to 2 with the criterion of minimizing the perceptual weighted distance between the approximated LSF vector $\tilde{\mathbf{w}}$ and original LSF vector $\mathbf{w}$. The set of weights $\mathbf{v}$ estimated based on acoustic similarity are used to construct the PCs of face points corresponding to the current speech frame:

$$\hat{\mathbf{p}}(t) = \sum_{i=1}^{5N} \mathbf{v}_i \mathbf{F}_i \tag{8}$$

where $\mathbf{F}_i$ represents the average principal component vector for i^{th} codebook entry. Next, the time sequence of estimated principal component vectors, $\hat{\mathbf{p}}(t)$ is smoothed to provide more natural face point trajectories. We used two different methods for smoothing: i) triangular windowing; and ii) spline interpolation.

3. EVALUATIONS

We used ten minutes of audio-visual training data from a single talker to generate our codebooks and the visual phoneme similarity matrix. Five minutes of data was set aside for testing. The visual data was recorded at a 60 Hz sampling rate. Using the proposed algorithm face point trajectories were synthesized for the test data. In order to test the upper limit on the performance of the algorithm, we resynthesized the training utterances as well. Figure 2 shows an example face trajectory synthesized from one of the test utterances. Here, the middle plot shows the center upper lip point trajectories along y-axis across time for original (dark dotted curve), synthesized with spline smoothing (dark solid curve), and synthesized with triangular smoothing (light curve). The bottom plot shows the center lower lip point trajectories along y-axis across time for original (dark dotted curve), synthesized with spline smoothing (dark solid curve), and synthesized with triangular smoothing (light curve). As can be seen from the figure, both synthesis algorithms are approximating the true face point trajectories reasonably well. For example, for the /f/ phonemes in "often" (at time 0.8 sec) and "farm" (at time 2.0 sec) the synthesized lower lip moves upward following the true trajectory. In fact, the highest error regions correspond to non-speech sections. For speech sections, the performance is significantly better. From the figure it can also be observed that the spline method produces more natural and smooth trajectories when compared to triangular smoothing method. However, it results in relatively larger delays when compared to the triangular smoothing method.

For the evaluations, we used the correlation coefficient between original and synthesized face point trajectories as the performance criterion. In order to make a fair judgment of the performance we used speech-only frames. The silence frames were identified and disregarded in the evaluations based on energy thresholding. In order not to disregard stop consonants a median filtering over a sufficiently long duration (112 ms) on the energy contour was applied before the energy thresholding. The selected frames are marked with dots in a straight line at the center of the lip trajectory plots in Figure 2. No dots are printed for silence frames. Since most of the points on the face do not move significantly, we used upper and lower lip y-axis trajectories to obtain a reference of performance. The average correlation coefficients between face point coordinates for the original and synthesized data are shown in Table 1 both for training and test data. From the table it can be seen that despite the fact that the spline method produces more natural face point trajectories it performs slightly worse when compared to the triangular smoothing method. This result can be explained by the fact that in general the spline method produces relatively larger delays.

In order to determine the optimal number of similar context-phones (N in Equation 5) used in the restricted codebook search, we performed simulations. Figure 3 shows the correlation between synthesized (triangular smoothing) and true face point trajectories as a function of the num-

ber of similar context-phones used in the codebook. After 3 context-phones the curve levels off for held-out data. As can be expected the performance on the training data degrades as more context-phones are used.

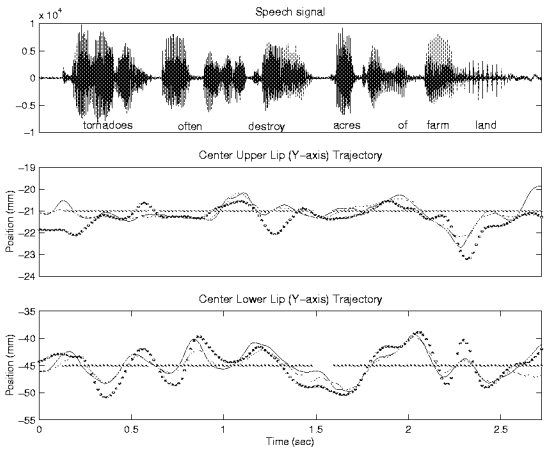

Figure 2: The comparison of original and synthesized face point trajectories for held-out data. The sentence was "Tornadoes often destroy acres of farm land".

In order to determine the optimal number of similar context-phones (N in Equation 5) used in the restricted codebook search, we performed simulations. Figure 3 shows the correlation between synthesized and true face point trajectories as a function of the number of similar context-phones used in the codebook. After 3 context-phones the curve levels off for held-out data. As can be expected the performance on the training data degrades as more context-phones are used.

Our future plans include the development of a more accurate global evaluation measure in terms of its correlation with human judgment.

4. CONCLUSION

In this paper, a novel algorithm for face point trajectory synthesis is described. For the modeling phase, an audio-visual codebook is generated based on context-dependent phonetic labels. In addition, the automatic generation of a visual phoneme similarity matrix is described. The codebook and the matrix are then used in the synthesis stage to select the most likely codebook entries for a given speech segment and phonetic label. The most significant contribution in this paper is the usage of acoustics in synthesizing the fine detail face trajectories. The algorithm can be generalized by not restricting the codebook search using phonetic information. In that case, acoustic information alone can be used to determine the codebook weights across the whole audio-visual codebook. The performance may not be as good when compared to algorithm performance using phonetic information, since acoustically confusable phonemes (e.g., /m/ versus /n/) may create problems in the synthesized face in such a scheme. However, this capability may be useful in practical applications such as video conferencing or where language independence is a requirement.

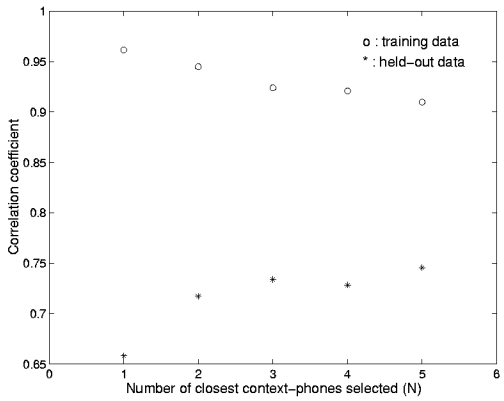

Figure 3: The influence of the number of similar context phones N incorporated in the codebook on the performance of the proposed algorithm.

Face Synthesis Algorithm Performance Evaluation			
Test condition	Whole Face	Lower Lip	Upper Lip
Training Data	0.9239	0.9463	0.9287
Test Data	0.7338	0.8468	0.7230

Table 1: Average Correlation between original and synthetic face point trajectories during speech-only sections using top 3 visually most similar context-phones.

5. REFERENCES

[1] L.M. Arslan and D. Talkin. "Voice Conversion by Codebook Mapping of Line Spectral Frequencies and Excitation Spectrum". In *Proc. EUROSPEECH*, volume 3, pages 1347–1350, Rhodes, Greece, September 1997.

[2] E.T. Auer, L.E. Bernstein Jr., R.S. Waldstein, and P.E. Tucker. "Effects of phonetic variation and the structure of the lexicon on the uniqueness of words". In *Proc. AVSP Workshop*, pages 21–24, Rhodes, September 1997.

[3] J. Bertenstam, J. Beskow, M. Blomberg, R. Carlson, K. Elenius, B. Granstrom, J. Gustafson, S. Hunnicut, J. Hogberg, R. Lindell, L. Neovius, A. de Serpa-Leitao, L. Nord, and N. Strom. "The Waxholm system - a progress report". In *Proc. of Spoken Dialogue Systems*, Vigsø, Denmark, 1995.

[4] J. Beskow. "Animation of Talking Agents". In *Proc. AVSP Workshop*, pages 149–152, Rhodes, September 1997.

[5] C. Bregler, Michele Covell, and Malcolm Slaney. "Video Rewrite: Visual Speech Synthesis from Video". In *Proc. AVSP Workshop*, pages 153–156, Rhodes, September 1997.

[6] N. Katashi and T. Akikazu. "Speech dialogue with facial displays". In *Proc. of the 32nd Annual Meeting of the Assoc. for Comp. Ling.*, pages 102–109, 1994.

[7] H. Yehia, P. Rubin, and E. Vatikiotis-Bateson (1998) (in press). "Quantitative association of acoustic, facial, and vocal-tract shapes". *Speech Communication.*