REDUCING THE FOOTPRINT OF THE IBM TRAINABLE SPEECH SYNTHESIS SYSTEM

Dan Chazan, Ron Hoory, Zvi Kons, Dorel Silberstein and Alexander Sorin

IBM Haifa labs
Mount Carmel, Haifa 31905, Israel.

ABSTRACT

This paper presents a novel approach for concatenative speech synthesis. This approach enables reduction of the dataset size of a concatenative text-to-speech system, namely the IBM trainable speech synthesis system, by more than an order of magnitude. A spectral acoustic feature based speech representation is used for computing a cost function during segment selection as well as for speech generation. Initial results indicate that even with a dataset size of a few megabytes it is possible to achieve quality which is significantly higher than existing small footprint formant based synthesizers.

1. INTRODUCTION

In recent years, major progress has been made in the quality and naturalness of text-to-speech (TTS) systems. Today, high quality speech can be produced by concatenative synthesis systems, where speech segments are selected from a large speech dataset ([1], [2], [3]). The contents of the speech dataset is a critical factor in the synthesis quality. Usually, a large number of speech segments with an overall duration ranging from 30 minutes to several hours is stored in the dataset. Sophisticated search algorithms are used to determine which segments are to be concatenated. The large size is required in order to cover as many phonetic contexts and as many acoustic environments as possible, and hence, avoid discontinuities at the concatenation points and reduce the amount of prosodic modifications. As a result, high quality concatenative TTS systems tend to require a large amount of disk and memory resources, ranging from tens to hundreds of megabytes, even for a single voice. While for server based TTS systems this problem is not a severe one, for small, hand-held and other low resource devices this size makes the concatenative TTS system unsuitable for implementation.

Several approaches have been previously suggested to reduce the size of the speech dataset. The method proposed in [4], is based on off-line pre-selecting segments, in order to determine which subset of segments is the most useful for synthesis. Previous systems have used a single version or small number for each synthesis unit ([5]). The reduction that can be achieved by this approach without quality degradation is usually quite limited. Another approach suggested (not necessarily implemented) in [6], [7], [8] is to use low bit rate speech compression techniques.

The current IBM system [3] uses a compression algorithm that achieves factor 7 compression at any sampling rate with almost no noticeable degradation in speech quality. This, however, is not sufficient, and typically results in a 50MB dataset (for 11 KHz) even after both pre-selection and compression.

The work described in this paper is aimed at significantly reducing the footprint of the IBM trainable synthesis system [3]. The synthesized speech quality should, on one hand, significantly outperform the non-concatenative (formant) small footprint system currently used by IBM. On the other hand the quality should be as close as possible to that obtained before footprint reduction. The advantage of the approach taken here is that the same feature vectors are used for both segment search and speech generation (as in [9]). This makes the method efficient as well as simple, and allows for a TTS scheme that works entirely in the feature domain.

The paper is organized as follows: Section 2 describes the speech representation and compression that are the basis for this work. Section 3 describes the small footprint concatenative text-to-speech system. Section 4 describes some initial results.

2. THE SPEECH REPRESENTATION

The speech representation is based on the mel frequency cepstral coefficients (MFCC) feature vectors commonly used in speech recognition systems. The MFCC features are currently utilized by the IBM concatenative TTS system [3] for automatic speech segment alignment during dataset preparation, as well as for the computation of the cost function used for the segment search and selection during the synthesis process. Recently, a novel technique for speech reconstruction from the MFCC features and pitch has been introduced [10], which makes it possible to reuse the MFCC features for the speech generation as well.
2.1. Using a sinusoidal speech model

The approach presented in [10] uses a sinusoidal model ([11]), where the short-term (ST) speech signal is represented by a sum of sine-waves. The ST-signal is characterized by the frequencies \( \{f_i\}_{i=0}^{N-1} \), amplitudes \( \{A_i\}_{i=0}^{N-1} \) and phase values \( \{\varphi_i\}_{i=0}^{N-1} \) of the component sine-waves. According to the sinusoidal model, the discrete-time Fourier transform of the corresponding windowed short-term signal is given by:

\[
\hat{S}(f) = \sum_{i=0}^{N-1} C_i W(f - f_i),
\]

where \( C_i = A_i e^{j\varphi_i} \) are the complex amplitudes and \( W(f) \) is the Fourier transform of the time domain window (typically a Hamming window). Although the sinusoidal speech model by itself is a very accurate speech representation, and was indeed used in its Harmonic plus Noise (HNM) version for speech synthesis ([6], [7]), it is not very efficient, because of the large number of parameters required.

The synthesis system presented in this paper represents the MFCC features and the original phase values of the speech waves. According to the deviation of the short-term speech signal from a pure sinusoidal model (1), synthesis is carried out using the amplitudes estimated from the MFCC features with phases values \( \varphi_i \) that are obtained from the amplitudes by a method akin to the minimum phase approach presented in [10]. Unlike in the pitch synchronous analysis/synthesis used in [3], analysis/synthesis is carried out every 10 msec with a 25 msec analysis frame.

Two variants of the analysis/synthesis model are explored in this paper. In the first, referred to as the synthetic phase version, the analysis for every frame, consists of MFCC feature extraction, pitch determination, and computation of the degree of voicing (d.o.v). The degree of voicing, ranging from 0 (fully unvoiced) to 1 (fully voiced), is determined according to the deviation of the short-term speech signal from a pure sinusoidal model (1). Synthesis is carried out by combining the amplitudes reconstructed from the MFCC features with phases values which are obtained from the amplitudes by a method akin to the minimum phase approach presented in [10]. The degree of voicing compensates for the lack of phase data. It eliminates "buzziness" effects, by the addition of colored noise with an emphasis on high frequencies, in an amount growing as the degree of voicing becomes lower.

In the second variant of the model, referred to as the original phase version, the analysis includes estimation of the complex amplitudes \( C_i \) by minimizing the model error \([S(f) - \tilde{S}(f)]^2\), where \( \tilde{S}(f) \) is given by (1), and \( S(f) \) is the Fourier transform of the windowed ST signal. The synthesis is carried out using the amplitudes estimated from the MFCC features and the original phase values \( \varphi_i \).

2.2. Speech compression

The speech parameters to be compressed include the MFCC feature vector, the pitch value and the degree of voicing value for the synthetic phase case and MFCC, pitch and phase values for the original phase case. Since the work done on phase compression is still ongoing, we shall focus here on the synthetic phase case. Two compression configurations were tested. In the first, for every frame, the information for the 24 MFCC features, for the pitch frequency and for the degree of voicing is independently coded (this ensures simple random access). In the second, referred to as decimated, the same information is coded every 20 msec (for every second frame). In the interleaved frame only an interpolation factor value \( \alpha_n \) is coded. Given the two quantized feature vectors from the neighboring frames \( X_n', X_{n+1}' \), \( \alpha_n \) is computed to minimize the linear interpolation error \([|(1 - \alpha_n) X_n' + \alpha_n X_{n+1}' - X_n]|^2\).

Coding of the MFCC features is done by split VQ, while the pitch and degree of voicing are coded with scalar quantization. When compressing the speech waves residing in a TTS system dataset, bit rate can be reduced by:

- Designing codebooks using speech data from the dataset alone (single speaker).
- Discarding pitch information, since usually pitch is synthesized at synthesis time.

The following table summarizes the bit rate figures in bps for the case of 11 KHz with synthetic phase. Pitch information was excluded.

<table>
<thead>
<tr>
<th></th>
<th>MFCC</th>
<th>D.O.V</th>
<th>Int. factor</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>7000</td>
<td>300</td>
<td>0</td>
<td>7300</td>
</tr>
<tr>
<td>Decimated</td>
<td>3500</td>
<td>150</td>
<td>150</td>
<td>3800</td>
</tr>
</tbody>
</table>

The two table lines match compression ratios of 1:24 and 1:46 compared to the 1:7 reported by [3].

3. TEXT-TO-SPEECH SYNTHESIS SYSTEM

The IBM trainable speech synthesis system ([2], [3]) is a state-of-the-art concatenative speech synthesis system. The system uses hidden Markov model (HMM) state-sized segments as its basic synthesis units. The segment search and selection is carried out using a decision tree and dynamic programming algorithm. The prosody of the selected segments is modified and they are concatenated, using pitch synchronous information (pitch marks) residing in the dataset.

3.1. The basic small footprint system

The structure of the basic small footprint system is shown in figure 1.

The TTS front-end processes the text including: text to phone conversion, phrase boundary placement, duration,
pitch and energy prediction. The front-end processing is followed by conversion of the phone sequence to target acoustic leaves (HMM states) by dropping the implied sequence of contexts down the decision tree. Prosodic values corresponding to the leaf sequence are also computed. For simplicity of the drawing, this stage was not explicitly depicted, but embedded in the front-end block. In the synthesizer back-end, the segment selection and concatenation block searches the database with the aim of selecting the sequence of segments, which best produces the target sentence. The search is based on a dynamic programming process, which minimizes a cost function. The cost function weighs spectral continuity cost, computed using the MFCC feature vectors, and prosody target cost (difference between the inherent prosody and the target prosody of the segment). The feature vectors of the selected segments are then extracted from the dataset, and simple manipulations are performed to change the prosody (duration is changed by skipping and repeating feature vectors, energy by updating the MFCC values). The modified feature vectors corresponding to the selected segments are then concatenated and fed together with the target pitch into the reconstruction block, which generates the speech signal as in [10].

The main difference between this scheme and the original IBM trainable synthesis system is in the speech generation part. Originally, a pitch synchronous synthesis scheme was used for prosody modification and segment concatenation. Here, these operations are carried out in feature domain, and then followed by a reconstruction step, converting the sequence of feature vectors back to time domain. Some changes were also made in adapting the cost function to this scheme. Initially, the dataset size was 54 MB, including:

a. Speech data (originally compressed with a 1:7 ratio).
b. Pitch marks data.
c. Segment information (decision tree leaf label, duration, pitch etc.).
d. MFCC vectors for spectral continuity cost computation.

In the small footprint scheme described in this paper, the speech data is represented by compressed feature vectors (with 1:24 or 1:46 compression ratios). The feature vectors are used for both speech generation and spectral continuity cost calculation. Pitch marks are not required any more. This enables us to reduce the total dataset size by more than an order of magnitude.

### 3.2. Using original phase

It is possible to further improve the synthesized speech quality by using the original phase data (especially for low frequency harmonics). This adds three steps: aligning and storing the phase data with the MFCC feature during dataset preparation, combining the original phase with the synthetic phase at synthesis time and realigning consecutive frames to reproduce the correct pitch. The phase which will be used in the synthesis process is aligned to make it as smooth as possible. This is required in order to reduce the phase unwrapping problem and to simplify the recombination with the synthetic phase.

The phase alignment is carried out by removing a linear phase term, which is equivalent to a time domain shift by an offset $\tau$. Several methods where suggested for determining the best offset (e.g. the center of gravity of the signal [12]). We obtained the best results by requiring smoothness of the signal’s complex amplitudes $\{C_k\}_{k=1}^{N-1}$ after the time shift. We use $\tau$ that minimizes the following weighted form of the derivative:

$$
\sum_{k=0}^{N-1} \frac{C_{k+1}}{\sqrt{|C_{k+1}|}} e^{2\pi i \tau f_k} - \frac{C_k}{\sqrt{|C_k|}} e^{2\pi i \tau f_k} \right|^2.
$$

(2)

Usually, only phase values of the lower frequency harmonics (typically 20) are stored. During synthesis, the original phase is used for lower harmonics and the synthetic phase for the higher harmonics. In order to realign the segments, the current frame is correlated with the previous one. Both the correlation and the realignment are performed on the Hilbert transform of the signals. This gives us an additional degree of freedom – a constant phase term to be used on top of the linear term, in order to make transition between the segments smoother.

### 4. INITIAL RESULTS

This section presents the results of a formal listening test. The objective of the test was to compare the quality of the small footprint concatenative system to the original IBM trainable synthesis system and to IBM’s current small footprint system based on formant synthesis. For the small footprint concatenative system, synthetic phase was used, and the feature files were not decimated. The overall dataset size was 5MB.
For the test, three sentences were randomly selected and synthesized by all three systems with a male voice at 11KHz sampling rate. Listeners were 25 adults unfamiliar with the TTS system. Each listener was presented with a sequence of sentences played by high bandwidth loudspeakers. The sequence contained sentences from the three systems at random order. The listeners were asked to rate the overall quality, intelligibility, naturalness and pleasantness. We used 1-5 MOS-like grades where 1 is the worst and 5 is the best.

The results are presented in figure 2. It shows the average score over all sentences and all listeners for each system. It can be seen that the small footprint concatenative system (B) outperforms the formant based system (A), but is still not as good as the original large footprint concatenative system (C). All differences are statistically significant (using t-tests with 95% confidence) except for the difference of the intelligibility score between systems A and B.

Informal listening tests showed that for the decimated compression configuration (as described in section 2.2), there is no significant degradation in performance.

5. CONCLUSION

Although the results shown in the paper are initial they already indicate that the concatenative method significantly outperforms the conventional formant synthesis even at small footprints of a few megabytes. Future work will focus on enhancing the quality of the small footprint concatenative system. This can be achieved by improving the speech model (e.g., with phase information as described in section 3.2) and by better adapting the segment selection mechanism to the speech representation used.

6. ACKNOWLEDGEMENTS

Thanks to Michael Picheny, Mahesh Viswanathan and Rob Donovan from IBM T.J. Watson Research center for their help with the IBM trainable speech synthesis system.

7. REFERENCES