A speech model of acoustic inventories based on asynchronous interpolation

Alexander B. Kain, Jan P. H. van Santen

Center for Spoken Language Understanding
OGI School of Science & Engineering at OHSU
20000 NW Walker Road, Beaverton, OR 97006, USA
kaina@ohsu.edu

Abstract

We propose a speech model that describes acoustic inventories of concatenative synthesizers. The model has the following characteristics: (i) very compact representations and thus high compression ratios are possible, (ii) re-synthesized speech is free of concatenation errors, (iii) the degree of articulation can be controlled explicitly, and (iv) voice transformation is feasible with relatively few additional recordings of a target speaker. The model represents a speech unit as a synthesis of several types of features, each of which has been computed using non-linear, asynchronous interpolation of neighboring basis vectors associated with known phonemic identities. During analysis, basis vectors and transition weights are estimated under a strict diphone assumption using a dynamic time warping approach. During synthesis, the estimated transition weight values are modified to produce changes in duration and articulation effort.

1. Introduction

The capabilities of mobile electronic devices are continuously expanding due to improvements in hardware design and manufacturing. For example, today's cell phones can run complex office productivity software, audio and video players, and Internet browsers. Obviously, speech technologies are highly desirable in mobile units, which typically feature limited input and output interfaces, and are often used in hands-off, eyes-off tasks. One of these technologies is a text-to-speech (TTS) system which allows the user to receive information by listening.

There are two viable TTS technologies today: *formant* synthesis and *concatenative* synthesis. In formant synthesis, great flexibility is achieved by controlling important speech features explicitly; however, the resulting speech, while highly intelligible, often sounds unnatural. A concatenative synthesizer, on the other hand, takes pre-recorded speech units from its *acoustic inventory* (AI), optionally modifies them prosodically, and joins them together to produce a more natural-sounding speech utterance. However, audible discontinuities at concatenation points or excessive signal processing, due to a mismatch in phonetic or prosodic context, may degrade the speech quality. To feature a variety of phonetic contexts and thus reduce the number of concatenations, an acoustic inventory typically holds many thousands of units, whose lengths may vary from short phoneme pair units (*diphones*) to words and phrases.

With the price of storage continually decreasing, concatenative systems with large AIs have become feasible on personal computers and servers. However, the implementation of such a system on embedded hardware is proving difficult because of the large memory requirements ranging from tens to thousands of megabytes. Since the perceived quality of a concatenative TTS system is directly related to the number and length of available units, a compression technology for AIs is extremely useful for any storage-limited device.

Although the area of speech coding has produced many compression algorithms, none of them are specifically designed to compress an AI, which differs from general speech coding in several respects:

- An AI contains speech from a single speaker in acoustically constant and noise-free conditions. General purpose speech coders, however, are designed to work with a variety of speakers and environments.
- An algorithm designed expressly for the compression of an AI can be highly complex computationally because processing is performed offline, as long as decoding is sufficiently fast. Furthermore, the algorithm can greatly capitalize on the availability of the complete dataset. In contrast, typical speech coders must provide very low latency and are thus limited to real-time solutions working on very short signals at a time.
- An AI contains information about the stored speech signals. At a minimum, the speech is segmented and labeled phonetically, but the inventory may also contain prosodic features and pitch marks. Also, a high quality AI has the *close acoustic match* property, which means that the units are recorded to be maximally compatible in the sense that their concatenation with each other reduces audible spectral discontinuities as much as possible. This additional structure and information is extremely advantageous for compression.

In this paper, we propose a model that describes the speech in an AI. Compression or coding is equivalent to fitting the parameters of the model, whereas decompression or decoding is equivalent to synthesizing speech from the stored parameters. In addition, our speech model forms a conceptual bridge between concatenative and formant synthesizers, merging the natural speech quality of concatenative synthesis with the flexibility of formant synthesis. It has the following characteristics:

- By taking advantage of the special properties of an AI, the minimally required model parameter set makes very high compression ratios possible. However, it is also possible to specify a quality or size criterion for variable compression rates.
- The synthesized speech is free of any concatenation artifacts because the regions of speech that are near potential concatenation points are represented as a single set of parameters, thus exploiting the close acoustic match property.

This research was conducted with support from NSF grant 0117911

- An explicit coarticulation model allows for control over the articulation effort or *degree of articulation*, which is especially important when changing the speaking rate.
- When the number of the model parameters is sufficiently small, information about the segmental characteristics of the original speaker can be easily replaced with that of another speaker, allowing for a transformation of the synthesis voice to sound like any desired target speaker without recording a full AI.

2. Method

2.1. Core Idea

The core idea of our speech model is to represent a speech unit as a synthesis of several types of features, each of which has been computed using *non-linear*, *asynchronous* interpolation of neighboring *basis vectors*. Each basis vector can be associated (labeled) with a particular phoneme, allophone, or smaller unit. Labels may contain additional information about context and prosody.

In its most general form, we approximate the complex speech spectrum $\mathbf{X}$ at frame m of the i^{th} unit by letting

$$\hat{\mathbf{X}}^{u_i|U}(m) = \mathcal{T}\left(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\right) \tag{1}$$

where $U = u_{i-l}, \ldots, u_{i-1}, u_{i+1}, \ldots, u_{i+r}$ represents the context of l left neighbors and r right neighbors. (We drop most indices in the interest of readability.) $\mathcal{T}$ is a synthesis function that is given N interpolated speech feature vectors $\mathbf{p}$, which are calculated using

$$\mathbf{p}_n = \sum_{i=-l}^{r} w_n^{u_i|U}(m) \cdot \mathcal{M}_n\left(\mathbf{b}_n^{u_i}\right) \tag{2}$$

where w are the non-negative *transition weights* for basis vectors $\mathbf{b}$ modified by a modification function $\mathcal{M}$. For each unique label in the AI there exist N types of basis vectors, which are feature vectors describing various aspects of the underlying speech signal. The transition weights, which are subject to the condition

$$\sum_{i=-l}^{r} w_n^{u_i|U}(m) = 1 \quad \forall n, m \tag{3}$$

determine the relative contribution of each of the $l+r+1$ (modified) basis vectors for a particular n and m. The purpose of $\mathcal{M}$ is to introduce additional flexibility into the model, for example allowing an already calculated $\mathbf{p}_n$ to influence the calculation of $\mathbf{p}_{l \neq n}$.

The idea of modeling speech by interpolating basis vectors is not new, as exemplified by Atal's Temporal Decomposition method [1]. However, our proposed method is fundamentally different in that the phonemic identities of the basis vectors are known, and several asynchronous, non-linear interpolations are involved.

2.2. Implementation

In our current work, we experiment on the diphone database "MWM" from the OGIresLPC package [2]. We select a slightly revised set of American English phonemes as labels. Since some phonemes describe more than a single acoustic event, we represent them by two or more transitions involving several basis vectors: we (i) split diphthongs into two parts, e.g. /aI/ = /aI_1/ → /aI_2/ , where the diacritic distinguishes between the two different acoustic targets, (ii) split unvoiced plosives into three parts: the closure, burst, and aspiration, e.g. /t/ = /tc/ → /tb/ → /th/ , and (iii) represent affricates and syllabics as a combination of other basis vectors, e.g. /tS/ = /tb/ → /S/ and /l=/ = /&/ → /l/ .

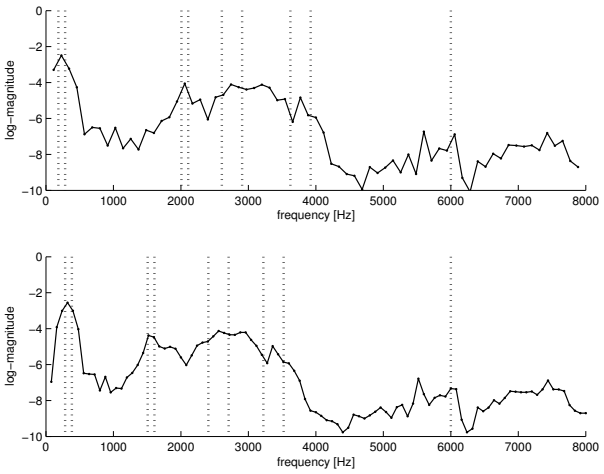

Figure 1: *Example of the effect of modification function $\mathcal{M}$. Given the log-magnitude spectrum of phoneme /i:/ with original landmarks (top), $\mathcal{M}$ nonuniformly resamples the spectrum to align with desired landmarks (bottom). Landmarks are represented as dashed lines.*

We select three types of basis vectors: a list of formant-like frequencies describing spectral "landmarks" ($\mathbf{b}_{\text{form}}$), and short-term magnitude ($\mathbf{b}_{\text{mspec}}$) and phase ($\mathbf{b}_{\text{pspec}}$) spectral information. In our implementation, Eq. (1) reduces to

$$\hat{\mathbf{X}}^{u_i|U}(m) = \mathcal{T}\left(\mathbf{p}_{\text{mspec}}, \mathbf{p}_{\text{pspec}}\right) \tag{4}$$

where the synthesis function $\mathcal{T}$ is equivalent to the construction of a complex spectrum from magnitude and phase spectra. The interpolated speech features are defined as

$$\begin{aligned}
\mathbf{p}_{\text{mspec}} &= \sum_{i=-l}^{r} w_{\text{spec}}^{u_i|U}(m) \cdot \mathcal{M}\left(\mathbf{b}_{\text{mspec}}^{u_i}; \mathbf{b}_{\text{form}}^{u_i}, \mathbf{p}_{\text{form}}\right) \\
\mathbf{p}_{\text{pspec}} &= \sum_{i=-l}^{r} w_{\text{spec}}^{u_i|U}(m) \cdot \mathcal{M}\left(\mathbf{b}_{\text{pspec}}^{u_i}; \mathbf{b}_{\text{form}}^{u_i}, \mathbf{p}_{\text{form}}\right) \\
\mathbf{p}_{\text{form}} &= \sum_{i=-l}^{r} w_{\text{form}}^{u_i|U}(m) \cdot \mathbf{b}_{\text{form}}^{u_i}
\end{aligned} \tag{5}$$

where $\mathcal{M}$ is a function that warps magnitude and phase spectra along the frequency axis. Specifically, given a particular spectrum and its landmarks, $\mathcal{M}$ calculates a non-uniform resampling of the original frequency locations and returns a magnitude spectrum with landmarks at $\mathbf{p}_{\text{form}}$ (see Fig. 1). Note that even though there are 3 types of basis vectors, the transition weights for the spectral parameters are tied.

Eq. (5) is based on the observation that in transitions between most phonemes, formants and the overall spectral shape change asynchronously. For example, a transition from /i:/ to /v/, as in the word "leave", shows a change in formants that starts well before the onset of frication. Another motivation is to model the speech spectrogram as a "morphing" between different speech sounds. Similar to the process of image morphing,

we use formants as salient features to render a good approximation of the transition between two sounds, one which could not be achieved by simply cross-fading. Yet another way to view the system is as a formant synthesizer whose formant trajectories are fitted to the speech in the AI, but one that merges additional information about the overall spectral shape, resulting in more natural synthesis.

3. Analysis

3.1. Estimating basis vectors

For each label, we select a representative location in the AI manually (for an automatic approach see [3]). To estimate spectral basis vectors $\mathbf{b}_{\text{mspec}}$ and $\mathbf{b}_{\text{pspec}}$ a pitch-synchronous sinusoidal analysis is carried out. To estimate $\mathbf{b}_{\text{form}}$, we manually determine formant frequencies f and bandwidths b and construct a sorted list of frequencies of the form $\{f_1 - b_1, f_1 + b_1, f_2 - b_2, f_2 + b_2, \ldots, f_4 - b_4, f_4 + b_4, C\}$, where C is a constant upper limit frequency beyond which no spectral modification is desired. When formants were not visible, formant frequencies from locus theory were used [4].

3.2. Estimating transition weights

During analysis, we assume the close acoustic match property to hold perfectly, i.e. the boundaries of diphone units are assumed to be free of coarticulation. For analysis purposes, we focus on fitting to the magnitude spectrum exclusively. Therefore, we can rewrite Eq. (4) and (5) to approximate the transition

$$|\hat{\mathbf{X}}|^{u_0 \to u_1}(m) = w_{\text{spec}}^{u_0 \to u_1}(m) \cdot \mathcal{M}\left(\mathbf{b}_{\text{mspec}}^{u_0}; \mathbf{b}_{\text{form}}^{u_0}, \mathbf{p}_{\text{form}}\right) + (1 - w_{\text{spec}}^{u_0 \to u_1}(m)) \cdot \mathcal{M}\left(\mathbf{b}_{\text{mspec}}^{u_1}; \mathbf{b}_{\text{form}}^{u_1}, \mathbf{p}_{\text{form}}\right) \quad (6)$$

$$\mathbf{p}_{\text{form}} = w_{\text{form}}^{u_0 \to u_1}(m) \cdot \mathbf{b}_{\text{form}}^{u_0} + (1 - w_{\text{form}}^{u_0 \to u_1}(m)) \cdot \mathbf{b}_{\text{form}}^{u_1} \quad (7)$$

where we require $w(1) = 1$ for the first frame, $w(M) = 0$ for the last frame M, and $w(m) \geq w(m+1)$.

Given an estimate of the basis vectors, we solve for the optimal transition weights by means of a dynamic time warping (DTW) algorithm. First, we construct the three-dimensional *transition cube*

$$\mathbf{Q}_{k,a,b}^{u_0 \to u_1} = a/A \cdot \mathcal{M}\left(\mathbf{b}_{\text{mspec}}^{u_0}; \mathbf{b}_{\text{form}}^{u_0}, \mathbf{p}_{\text{form}}\right) + (1 - a/A) \cdot \mathcal{M}\left(\mathbf{b}_{\text{mspec}}^{u_1}; \mathbf{b}_{\text{form}}^{u_1}, \mathbf{p}_{\text{form}}\right) \quad (8)$$

$$\mathbf{p}_{\text{form}} = b/B \cdot \mathbf{b}_{\text{form}}^{u_0} + (1 - b/B) \cdot \mathbf{b}_{\text{form}}^{u_1} \quad (9)$$

with $a = 0, 1, \ldots, A$ and $b = 0, 1, \ldots, B$, for the k^{th} spectral frequency component. (The general case requires a $n+1$-dimensional hypercube.) The columns along the first dimension are magnitude spectra sampling all possible intermediate states in the transition from u_0 to u_1. After aligning $\mathbf{Q}$ to the original magnitude spectrogram $|\mathbf{X}|$, the optimal DTW path is used to assign final values to the transition weights w_{spec} and w_{form} (see Fig. 2). Appropriate starting, ending, and local path constraints are used to enforce the conditions stated above.

4. Synthesis

Phoneme durations, and thus distances between basis vectors, are generally different from those seen during the analysis step. Therefore, a strategy must be devised for arranging the transition weight values during compression or expansion of a transition. While various strategies have been suggested [5], we propose to preserve the articulation effort during changes to duration. Specifically, transition weights can be expanded by stretching only the center of the transition trajectory (see Fig. 3). However, they cannot be compressed any further: if a desired transition is shorter than what is stored, the influence of the current phoneme will reach beyond the current diphone context. In order to control the articulation effort and make fast speech possible, we introduce a parameter α that scales the timing of a transition around unit boundaries. The behavior is as follows: for $\alpha = 1$, nothing is changed; if the distance between two basis vectors is halved, then a setting of $\alpha = 2$ will increase the articulation effort such that the transitions will complete in the context of the diphone (as originally during analysis). In this manner it is possible to control the degree of articulation at any speaking rate, rendering speech as more hypo-articulate or hyper-articulate. In contrast, changes to duration using common techniques such as PSOLA are coupled with uncontrolled changes to articulation effort [6].

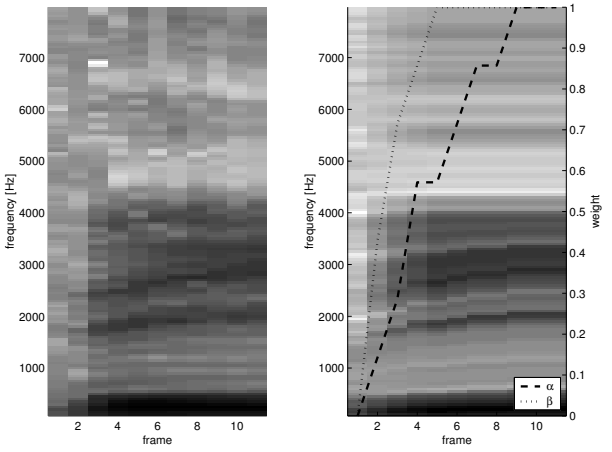

Figure 2: *Log-magnitude spectrogram of the original diphone (left) and estimated diphone with superimposed transition weights (right). The spectral frames of the estimated diphone correspond to columns in the transition cube that were traced out by the DTW path. The distinct role of the two transitions weights is evident: While w_{form} indicates relatively slow formant movements, w_{spec} indicates a relatively fast initial change of spectral balance, as is commonly associated with fricative-vowel transitions.*

Finally, Eq. (4) and (5) are used to calculate the complex spectrogram of the new sentence. The final time domain signal is calculated by a pitch-synchronous sinusoidal synthesis method (see Fig. 4).

5. Voice Transformation

Often it is desirable to customize a TTS system with a new voice, but only few recordings of the new speaker are available or practical. A possible strategy for this in the context of the proposed implementation is to simply replace the basis vector information (complex spectrum and formant frequencies and bandwidths), but leave the transition weights unchanged. This approach is effective in changing the synthesized voice to the degree that transition weights are speaker-independent.

An example application of voice transformation exists in the context of assistive technologies for people who present with a motor speech impairment or *dysarthria*. The ability to

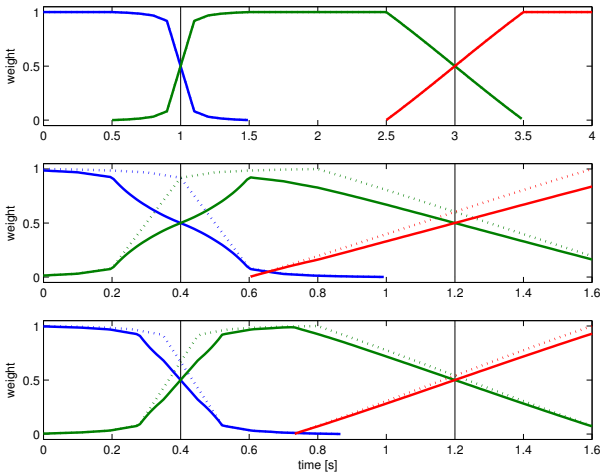

Figure 3: *Illustration of our approach to compress or expand transition weight values. Each plot shows transition weights for three units that have been originally estimated over one second transitions. The new durations for the top plot are 1, 2, and 1, respectively (The vertical bars indicate unit boundaries). Note that in areas between a "left" and "right" transition the weights are set to one. The middle plot shows weight values for shorter durations (dotted lines are weight values before normalization). Note that for a portion during the second unit the previous and next transition weights overlap, and thus a triphone context exists. Finally, we increase the value of α from 1 to 1.15 in the bottom plot, thus simulating an increased articulation effort. Weight values are for illustration only. Zero values are omitted in the interest of clarity.*

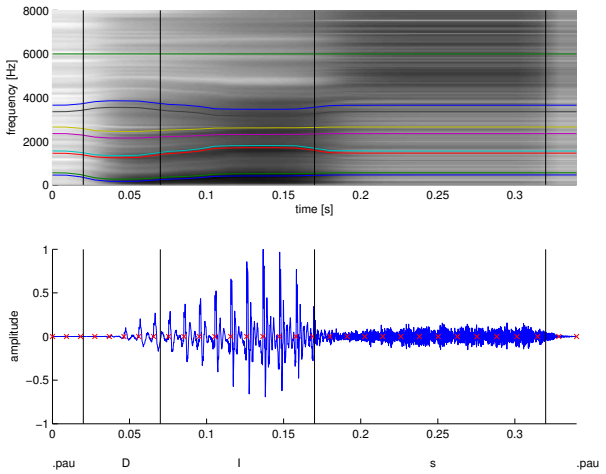

Figure 4: *Pitch-synchronous log-magnitude spectrogram with superimposed values of $\mathbf{p}_{form}$ (top) and time domain waveform with pitch marks (bottom) for the word "this".*

give a speech enhancement system an output voice that preserves the unique characteristics of the dysarthric speaker has important implications for quality of life. The ability to do this with a minimum of recordings is critical, because it allows one to capitalize on the ability of many dysarthric individuals to intermittently, and with great effort, produce almost normal speech.

6. Conclusions

An objective evaluation and coder comparison were carried out in a previous work, in which a 6.5Mb inventory was compressed to 57Kb (compression ratio of 1:114), at a spectral distortion of 8.7dB [3]. Because it is difficult to assess the quality of a TTS system based on objective measurements we plan on performing subjective listening tests in the future. In the meantime, example stimuli are available at our website http://cslu.cse.ogi.edu.

Building on the existing system, we intend to investigate the following extensions:

Automatic discovery of labeling set. Instead of specifying the set of labels to be used as basis vectors manually, it is desirable to devise an automatic method based on a size or quality criterion. When a particular speech unit cannot be modeled accurately enough by transition between existing basis vectors, then new basis vectors need to be inserted at optimal locations and labeled appropriately. Also, existing labels may need to be "split", based on contextual or prosodic effects. For example, the label /l/ may need to be replaced by two labels that correspond to /l_dark/ and /l_light/. Therefore, a search algorithm is needed that derives the optimal labeling set. This approach promises to be highly suitable for the compression of large unit-selection databases, where many concatenation points exist and a manual selection of labels is suboptimal and impractical.

Estimating transition functions. Typically, transition weight functions display a certain regularity which can be exploited by approximating them by a parameterized function, such as a sigmoid. Such a scheme would allow a decrease in the number of model parameters, resulting in a higher compression ratio.

Parameter Sharing. It can be observed that weight trajectories for certain types of transitions, such as vowel to nasal transitions, are similar. Therefore, transition weight trajectories may be shared between all transitions of this type, resulting in a further decrease in the number of model parameters.

7. References

[1] B. Atal, "Efficient coding for LPC parameters by temporal decomposition," *Proc. Int. Conf. Acoust., Speech and Signal Processing*, 1983, pp. 81–84.

[2] M. Macon, A. Cronk, J. Wouters and A. Kain, "OGIresLPC: Diphone synthesizer using residual-excited linear prediction," *Technical Report CSE-97-007, Dept. of Computer Science, Oregon Graduate Institute of Science and Technology, Portland, OR*, Sep 1997.

[3] A. Kain and J. van Santen, "Compression of acoustic inventories using asynchronous interpolation," *Proceedings of IEEE Workshop on Speech Synthesis*, September 2002.

[4] D. Klatt, "Review of text-to-speech conversion for english," *J. Acoust. Soc. Am.*, vol. 82, no. 3, pp. 737–793, September 1987.

[5] D. Broad and F. Clermont, "A methodology for modeling vowel formant contours in CVC context," *J. Acoust. Soc. Am.*, vol. 81, no. 1, pp. 155-165, January 1987.

[6] J. Wouters and M. Macon, "Control of spectral dynamics in concatenative speech synthesis," *IEEE Trans. Speech and Audio Proc.*, vol. 9, no. 1, pp. 30-38, January 2001.