COMPRESSION OF ACOUSTIC INVENTORIES USING ASYNCHRONOUS INTERPOLATION

Alexander B. Kain and 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

ABSTRACT

A compression method is proposed that takes advantage of a powerful property of acoustic unit inventories: In the appropriate acoustic space, units that share a (context-dependent or -independent) phoneme label must be close to a vector *phoneme template* associated with the phoneme. The method approximates units by interpolation between templates. The interpolation operation involves two asynchronous weight functions operating on the template. One is associated with spectral peak locations, the second with spectral balance. This enables approximating transitions such as [i:] $\rightarrow$ [v], in which formant movement precedes frication onset. The algorithm guarantees smooth concatenation points.

1. INTRODUCTION

Recent advances in hardware technologies have introduced fast processors on hand-held devices, such as personal digital assistants and mobile phones. However, what has prevented the most common form of text-to-speech (TTS) synthesis, *concatenative* TTS, from becoming ubiquitous on these devices is the large amount of memory that is required for storing a system's *acoustic inventory*, which contains the necessary acoustic units. Moreover, recent research has shown that substantial voice quality improvements can be obtained when inventories are used that contain not only phoneme pair units, or *diphones*, but also longer units, entire words, and even phrases. Thus, there is a quality/size trade-off in concatenative TTS. It follows that compression technologies are needed for any storage-limited device, even if the cost of storage continues to decline.

Over the last several decades, the area of speech coding has created many technologies for speech compression, and some of these have been used for compressing acoustic units [1]. However, these technologies have not produced the required compression ratios. This is no surprise, because speech coding applications differ in several key respects from synthesis:

- In an acoustic inventory (AI), units are spoken by the same speaker under acoustically constant and noise free conditions. By contrast, speech coding applications are designed to work under a variety of acoustic conditions and cannot be fine-tuned for a specific speaker.
- The AI encoder can be arbitrary complex, since encoding takes place only once, off-line, on a large, static, random access data set. Speech compression algorithms on the other hand must usually encode and decode in real-time, working on very short signals.
- A high quality AI has the *close acoustic match* property, by which is meant that, in an appropriate acoustic feature space, units that share a (context-dependent or -independent) phoneme label (joinable units) are close to a common feature vector, or *template*, associated with the common phoneme. For example, the /p/$\rightarrow$/i:/ and /i:/$\rightarrow$/tS/ diphone units have been selected from a larger corpus of recorded speech to be maximally compatible in the sense that their concatenation does not lead to audible spectral discontinuities.
- Additional information is available in the inventory, such as pitch marks and acoustic segment boundaries, which can be used advantageously by the compression algorithm.

In this paper, we propose a special purpose compression algorithm that achieves a high rate of compression while eliminating spectral discontinuities. We implement and evaluate the algorithm using a diphone inventory, but the approach can be extended to operate on larger and variable size unit inventories as well.

2. METHOD

2.1. Core idea

The core idea of our method is to approximate a diphone by interpolating between a left and a right phoneme template. The general idea of using interpolation for compression is not new, as exemplified by Atal's Temporal Decomposition method [2], a general purpose compression method in which speech frames are approximated by linear combinations of *basis vectors*. However, the proposed method is fundamentally different in that (i) the templates correspond to known phonetic labels instead of arbitrary (and numerous) frames, and (ii) it involves *two asynchronous, nonlinear* interpolation operations. The latter is based on the realization that during transitions between certain phonetic segments, articulators and their acoustic manifestations often change asynchronously. For example, a transition from a high vowel to a /v/, as in the word "leave", shows a lowering of the second formant that starts well before the onset of frication. Simple interpolation between an /i:/ and /v/ would create an unnatural transition in which either the frication onset is too early and smooth or in which the formant movement is too late and sudden. We need a method that can track asynchronous formant and spectral balance changes.

We represent a diphone in the form of $\mathbf{D}_{n,m}^{l \rightarrow r}$, which is the n^{th} frequency component of the complex spectrum of a short segment

This research was conducted with support from NSF Grants 0117911 "Making Dysarthric Speech Intelligible" and 0082718 "Modeling Degree of Articulation for Speech Synthesis". We are grateful to Gilead Cohen and Johan Wouters for helpful discussions, and to Mike Macon for inspiring us to work on this topic.

of speech at a discrete time point, or frame, m of the original diphone with phoneme labels l to r. In our specific method, we let

$$\mathbf{D}^{l\to r}_{n,m} \approx \hat{\mathbf{D}}^{l\to r}_{n,m} = \mathrm{T}\left(\mathbf{P}^l, \mathbf{P}^r, \alpha^{l\to r}_m, \beta^{l\to r}_m\right)_n \quad (1)$$

$$n = 1, 2, \ldots, N \quad m = 1, 2, \ldots, M$$

where $\hat{\mathbf{D}}^{l\to r}_{n,m}$ is the compressed diphone, and T is the *transition function*. T takes as arguments *phoneme templates* $\mathbf{P}^l$ and $\mathbf{P}^r$, as well as *transition weights* $\alpha^{l\to r}_m$ and $\beta^{l\to r}_m$. A phoneme template is defined as the representative complex spectrum of a typical rendition of a phoneme, whereas transition weights govern the temporal evolution of the transition.

Expanding the right hand side of Eq. 1, we define

$$\begin{aligned}&\mathrm{T}\left(\mathbf{P}^l, \mathbf{P}^r, \alpha^{l\to r}_m, \beta^{l\to r}_m\right)_n \\ &= \beta^{l\to r}_m \cdot \mathrm{M}\left(\mathbf{P}^l, \mathbf{v}^{l\to r}_m\right)_n \oplus \left(1-\beta^{l\to r}_m\right) \cdot \mathrm{M}\left(\mathbf{P}^r, \mathbf{v}^{l\to r}_m\right)_n\end{aligned} \quad (2)$$

$$0 \le \beta^{l\to r}_m \le 1$$

where $\mathbf{v}^{l\to r}_m$ is the K-dimensional *feature parameter state* at frame m, defined as

$$\mathbf{v}^{l\to r}_{k,m} = \alpha^{l\to r}_m \cdot \mathrm{V}\left(\mathbf{P}^l\right)_k + \left(1-\alpha^{l\to r}_m\right) \cdot \mathrm{V}\left(\mathbf{P}^r\right)_k \quad (3)$$

$$k = 1, 2, \ldots, K \quad 0 \le \alpha^{l\to r}_m \le 1.$$

The function $\mathrm{V}(\mathbf{P})$ returns the feature parameters of a vocal tract model for $\mathbf{P}$, whereas $\mathrm{M}\left(\mathbf{P}, \mathbf{v}^{l\to r}_m\right)$ is a speech modification function such that

$$\mathrm{V}\left(\mathrm{M}\left(\mathbf{P}, \mathbf{v}^{l\to r}_m\right)\right) = \mathbf{v}^{l\to r}_m.$$

In other words, M modifies $\mathbf{P}$ to have features $\mathbf{v}^{l\to r}_m$. We use the operator $\oplus$ to indicate addition taking place separately in the log magnitude and unwrapped phase domains, with subsequent recombination.

Examining Eq. 3, we observe that $\mathbf{v}^{l\to r}_m$ is calculated by interpolating between feature parameters of the two phoneme template using weight $\alpha^{l\to r}_m$. The result is used in Eq. 2 to modify both phoneme templates accordingly. Finally, the two modified spectra are interpolated using weight $\beta^{l\to r}_m$. Thus, the value of α will correlate with changes in tongue position or place of articulation and height, while β will correlate with spectral balance or the manner of articulation. Because the values of the two weights evolve asynchronously, we call our method *asynchronous interpolation compression* (*AIC*). We will now take a closer look at the various components of the transition function.

2.2. Estimating phoneme templates

Given an acoustic inventory, the initial task is to obtain phoneme templates $\mathbf{P}$ for each phoneme of the database. In order to interpolate between the templates, it is necessary to fix the frequencies of each component of $\mathbf{P}$.

The simplest approach to estimation is to select the appropriate speech locations manually. An automatic approach, taken in this paper, consists of collecting one or more *boundary frames* from all diphones that involve a specific phoneme. Boundary frames are frames that start or terminate a diphone and are typically located in the center of phonemes. For example, to estimate the phoneme template for /i:/, we collect left frames of diphones that begin with /i:/ as well as right frames of diphones that end in /i:/. Then, we set the template equal to the complex spectrum of the frame that has been identified as the centroid in the log-magnitude spectrum domain (see Fig. 1). This may not be the optimal solution; however, we do not expect significant improvements if the acoustic inventory has the close acoustic match property.

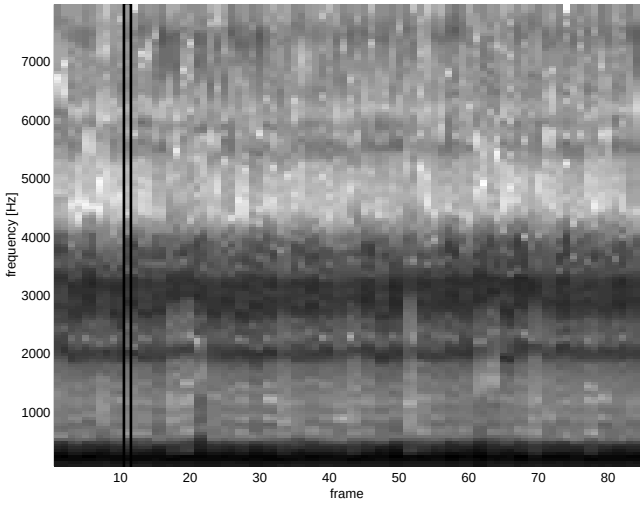

Fig. 1. Log-magnitude spectrogram of concatenated boundary frames for the phoneme /i:/. The phoneme template $\mathbf{P}^{/i:/}$, defined as the centroid, is marked by the rectangle. The close acoustic match property of the acoustic inventory is evident by the similarity across frames.

In certain diphones, acoustic events occur that are spectrally dissimilar to either phoneme template. An obvious example are diphthongs, such as in the diphone /aI/→/@/. In this case, unit-internal templates will have to be used and interpolated in sequence; in our example the templates would be $\mathbf{P}^{\mathrm{a(/aI/)}}$, $\mathbf{P}^{\mathrm{I(/aI/)}}$, and $\mathbf{P}^{/@/}$, where the first and last templates are (joinable) phoneme templates, but the middle template is used only in conjunction with its original diphone (see also Sec. 4).

2.3. Speech model and modification function

The particular implementations of V and M depend on the choice of speech model. In this work, we use the concept of "spectral landmark" features. Since it is possible to pre-compute $\mathrm{V}(\mathbf{P})$, we manually label the first three formant frequencies and bandwidths of each phoneme, resulting in a sorted list of frequencies of the form $\{F_1\text{-}B_1,\ F_1\text{+}B_1,\ F_2\text{-}B_2,\ F_2\text{+}B_2,\ F_3\text{-}B_3,\ F_3\text{+}B_3,\ C\}$, where C is a constant upper limit frequency beyond which no spectral modification is desired. When formants were not visible, we used formant frequencies from locus theory [3].

During synthesis, $\mathrm{M}\left(\mathbf{P}, \mathbf{v}^{l\to r}_m\right)$ has access to both the original features $\mathrm{V}(\mathbf{P})$ and the desired features $\mathbf{v}^{l\to r}_m$, using them to calculate a non-uniform sampling of the original frequency locations. To obtain the final spectrum, the magnitude and unwrapped phases are interpolated at the new frequencies. The original spectral balance is mostly preserved by this frequency warping (see Fig. 2).

The advantage of this method is that of computational efficiency — important for synthesis on a computationally limited device. However, the quality of modified speech spectra may be im-

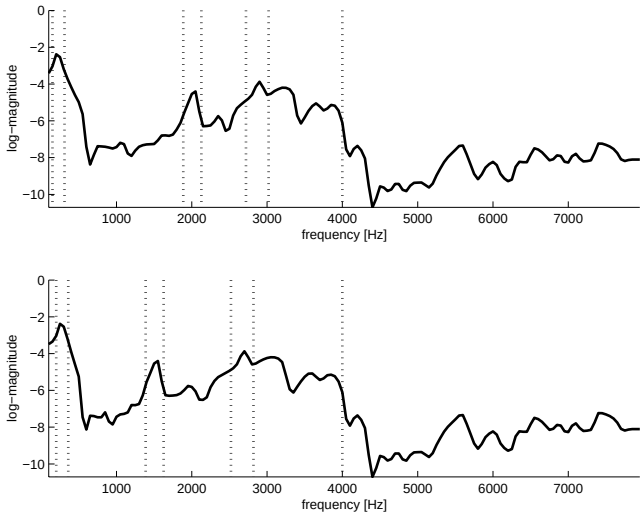

Fig. 2. Log-magnitude spectrum of the phoneme template $\mathbf{P}^{/\text{i:}/}$ before (top) and after modification (bottom). The dashed lines represent vocal tract feature parameters $\mathrm{V}\,(\mathbf{P})$ and $\mathbf{v}_m^{l\to r}$, respectively.

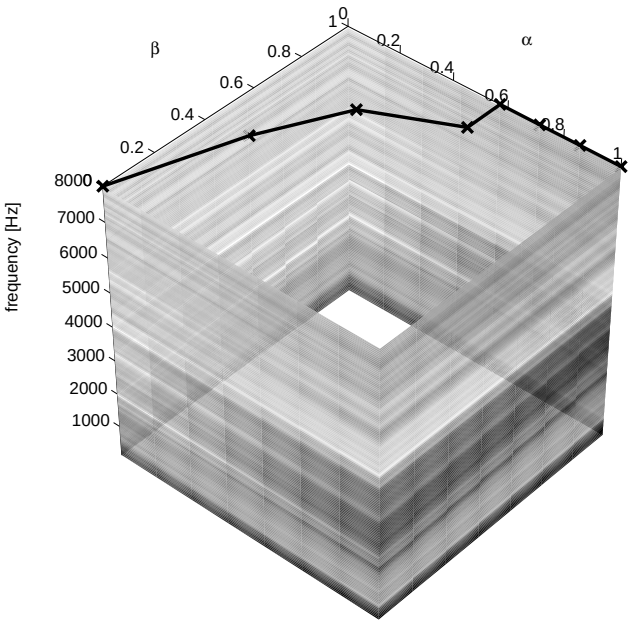

Fig. 3. Transformation cube $\mathbf{Q}$ for the diphone $\mathbf{D}^{/\text{D}/\to/\text{i:}/}$ (only four sides are shown). Each column is a log-magnitude spectrum, the result of a particular choice of α and β in Eq. 4. The left front side corresponds to a change in α only, equivalent to a spectral cross-fade between $\mathbf{P}^{/\text{D}/}$ and $\mathbf{P}^{/\text{i:}/}$. The right front side corresponds to a change in β only, which represents a continuous modification to $\mathbf{P}^{/\text{i:}/}$. The DTW path, superimposed onto the top side, is constrained to start at the left corner and end at the right corner. The transition weight values for each frame of the diphone are associated with the crosses in the path.

proved upon by using parts of the spectral modification method suggested by Wouters et. al [4].

2.4. Estimating transition weights

To guarantee smooth concatenation at the diphone boundaries, we impose conditions $\alpha_1 = \beta_1 = 0$ and $\alpha_M = \beta_M = 1$. It then follows that

$$\hat{\mathbf{D}}_{n,1}^{l\to r} = \mathbf{P}_n^l \quad \hat{\mathbf{D}}_{n,M}^{l\to r} = \mathbf{P}_n^r.$$

Thus the spectrum of two joinable diphones will be identical at their left and right boundaries, respectively, even if the acoustic units in the original inventory do not exhibit the close acoustic match property perfectly. Additionally, it is reasonable to require $\alpha_{m-1} \leq \alpha_m$ and $\beta_{m-1} \leq \beta_m$.

Given phoneme templates $\mathbf{P}$ and their feature parameters $V\,(P)$, we can now optimally estimate the transformation weights α and β to approximate a diphone $\mathbf{D}$ by using a dynamic time warping (DTW) algorithm. Because α and β can evolve independently, we construct the three-dimensional *transformation cube*

$$\mathbf{Q}_{n,a,b}^{l\to r} = \mathrm{T}\left(\mathbf{P}^l, \mathbf{P}^r, a/A, b/B\right)_n \tag{4}$$

$$a = 0, 1, \ldots, A \quad b = 0, 1, \ldots, B.$$

Now we align $\mathbf{Q}$ to the template $\mathbf{D}$ and the resulting DTW path is used to set values for α and β (see Figures 3 and 4). The constraints above are realized by restricting the starting and ending points of the path, as well as using appropriate local path constraints. While we allowed for large jumps in β (allowing for naturally occurring sudden changes in spectral balance), we reduced the maximum rate of change for α (since vocal tract feature parameters evolve relatively slower).

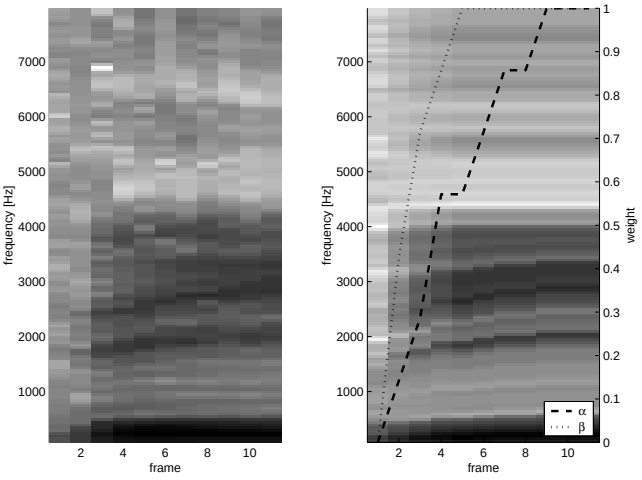

Fig. 4. Log-magnitude spectrogram of the original diphone $\mathbf{D}^{/\text{D}/\to/\text{i:}/}$ (left) and compressed diphone $\hat{\mathbf{D}}^{/\text{D}/\to/\text{i:}/}$ with superimposed transition weights (right). The spectral frames of the compressed diphone correspond to columns in the transformation cube that were traced out by the DTW path. The distinct role of the two transitions weights is evident: While α effects relatively slow formant movements, β causes a relatively fast change of spectral balance, as is commonly associated with fricative-vowel transitions.

3. EVALUATION

Currently, our diphone inventory has not been re-labeled yet to take into account unit-internal templates. Therefore, we do not include diphthongs, plosives, or affricates in our evaluation. As reference, we use the "MWM" database of the OGIresLPC package

Coder	SD	SEGSNR	SPD	Size	Ratio
Wave	-	-	5.9 (5.2)	6,519	1:1
G.722.1	4.1 (3.8)	16 (20)	5.9 (5.0)	638	1:10
G.722.2	7.3 (7.0)	4 (8)	4.9 (4.5)	197	1:33
AIC	8.7 (9.8)	1 (14)	0.0 (0.0)	57	1:114

Table 1. Mean spectral distortion (SD) and segmental signal-to-noise ratio (SEGSNR) across frames against the reference, and discontinuity distortion (SPD) between joinable frames of various coders. Variances are shown in parentheses. The size column refers to the total size of the compressed database in Kbytes, which includes other, uncompressed information such as pitch marks.

(http://cslu.cse.ogi.edu/tts), sampled at 16 kHz. We compare the reference against the ITU-T recommendation G.722.1 for speech coding at 24 Kbit/s [5], G.722.2 at 6.6 Kbit/s [6], and against our AIC approach.

A first evaluation involves frame-by-frame comparison of the reference with the coded speech, using standard spectral distortion (SD) and segmental signal-to-noise ratio (SEGSNR) measures. For each compression scheme, we encoded and decoded (i.e. synthesis without any prosody modification) the recorded diphones. For fricatives, the AIC coder randomizes the phases to avoid tonal artifacts. A second evaluation focuses on the spectral discontinuity (SPD) between the last and first frame of joinable diphones, using standard SD. By construction, spectral discontinuity is zero for the AIC method.

The results are shown in Table 1. While the size of AIC is an order of magnitude smaller than the G.722.2 coder, it also performs worse. However, it completely eliminated any spectral discontinuities that are present in the reference and other coders. Of course these objective measures, while informative, need to be validated with perceptual experiments. For example, the SEGSNR is sensitive to phase differences, yet it is unknown what the perceptual significance of these differences is.

Note that the AIC coding scheme is not fully size-optimized (e.g. there is no compression of phoneme templates, nor did we parameterize the weight trajectories or use parameter sharing between units of the same class; see Section 4).

4. EXTENSIONS

The proposed method is the initial result of a longer-term project, which has the following goals:

Application to longer units. To apply the method to longer units, more unit-internal templates are needed. A key issue is to what degree these unit-internal templates can be shared by different units. If they cannot be shared, then the proposed method still provides compression because a template requires the same amount of storage as a single frame, and the total number of frames in a unit substantially exceeds the total number of phonemes in a unit; presumably, the latter is roughly equal to the number of templates required to generate a unit. If templates can be shared, larger savings will result. This extension will enable hand-held devices to benefit from progress in long-unit based TTS.

Parameterized weight functions. The weights shown in Figure 4 are typical in that they can be approximated by a sigmoid function characterized by a slope and a location parameter. Such a scheme can achieve a further substantial increase in compression.

Parameter sharing. The weight trajectories may be shared by units belonging to certain classes, such as Vowel→Nasal transitions. Further savings would be obtained if the temporal patterns of all units in a given class have been normalized.

Voice adaptation. Changing the TTS voice can be accomplished by changing the phoneme templates while keeping the original transition weights, which appear to be highly speaker independent. This would permit the generation of personalized TTS systems using a minimum of recordings. This can be crucial for applications such as voice transformation for individuals with Dysarthria, where a coupled ASR-TTS system would be used to transform the Dysarthric speech into more intelligible speech. Individuals with Dysarthria are generally not able to create the amount of recordings needed for even a small diphone inventory.

Enhanced spectral control. Our hybrid Formant/Concatenative synthesizer has spectral control while maintaining naturalness. This enables development of synthesis modules that affect speaking rate, articulation effort, etc.

5. CONCLUSION

A compression method has been proposed that takes advantage of properties of acoustic inventories, in particular the property that the end and start points of joinable units must be close in acoustic space. This immediately suggests using some type of interpolation scheme, but the asynchronous character of transitions between phonetic segments makes it impossible to use any straightforward interpolation method. The proposed method captures transition asynchrony by using a two-step process in which spectrum peak locations and spectral balance are asynchronously manipulated.

The method can be considered as a hybrid of traditional formant or rule based systems such as MITTalk and concatenative systems: It uses recorded speech, but generates transitions via (trained) interpolation functions.

6. REFERENCES

[1] A. Acero, "A mixed-excitation frequency domain model for time-scale pitch-scale modification of speech," in *Proc. of the Int. Conf. on Spoken Language Processing*, Sydney, Australia, December 1998.

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

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

[4] 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.

[5] ITU-T, *G.722.1 7 KHz Audio coding at 24 Kbit/s and 32 Kbit/s for hands free operation in systems with low frame loss*, September 1999.

[6] ITU-T, *G.722.2 Wideband coding of speech at around 16 Kbit/s using adaptive multi-rate wideband (AMR-WB)*, January 2002.