COMPRESSION OF LINE SPECTRAL FREQUENCY PARAMETERS USING THE ASYNCHRONOUS INTERPOLATION MODEL

Alexander Kain and Todd Leen

Division of Biomedical Computer Science
Oregon Health & Science University
Portland, Oregon, USA

ABSTRACT

We apply an asynchronous interpolation model (AIM) to line spectral frequency trajectories. AIM represents speech transition features as crossfading between basis vector features, governed by individual interpolation weights per feature component. Basis vectors are initialized from demiphone labels, and then optimized using a local reconstruction error. Using a small diphone acoustic inventory, we reduce the number of parameters by using dimension-reduced latent space weights and a vector quantized pool of basis vectors. The highest compression rate of 1:11 resulted in a log spectral distortion of 4.83 dB.

1. INTRODUCTION

The capabilities of mobile electronic devices are continuously expanding due to improvements in hardware design and manufacturing. Cell phones run complex office productivity software, audio and video players, and Internet browsers. Speech technologies are highly desirable in mobile units, especially for hands-off, eyes-off tasks. Text-to-Speech (TTS) synthesis systems allow the user to receive information by listening rather than reading. Today's most natural sounding TTS systems are based on the *concatenative synthesis* approach, which uses a multitude of pre-recorded speech chunks (a contiguous section of natural speech) of a single speaker, stored in an *acoustic inventory* (AI), to stitch together a new output signal. The quality of the resulting speech relates directly to the size of the database, because the larger the chunks, the fewer the number of concatenation points at which audible artifacts can occur. Moreover, when the prosodic space is not covered by the acoustic inventory, prosodic modification becomes necessary, further degrading the speech signal. Alternatively, the *formant synthesis* approach does not sound very natural, but it is compact in size, gives full prosodic and spectral control over the speech signal, and is highly intelligible. *Statistical synthesis* approaches have emerged that use Hidden Markov Model (HMM) analysis and synthesis techniques to generate speech waveforms that are compact and easily trainable, but are still below the concatenative synthesis approach in quality.

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 systems on embedded hardware is proving costly because of the memory requirements ranging from tens to thousands of megabytes. Since the perceived quality of a concatenative TTS system increases with increasing AI size, a compression technology for AIs would be extremely useful for any storage-limited device (e. g. bluetooth headsets). Another requirement of mobile devices is that the speech generation algorithm must not be CPU-intensive.

No existing speech compression algorithms exists that is 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 computationally complex during offline encoding, as long as decoding is sufficiently fast. The algorithm can also capitalize on the availability of the complete and random-access dataset. In contrast, typical speech coders must provide very low latency and are thus limited to real-time encoding, working on very short signals at a time.
- A high-quality AI contains information in addition to the speech waveform. At a minimum, the speech is segmented and labeled phonetically, but the inventory may also contain prosodic features and pitch marks. The AI also 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 addition, researchers have attempted to improve the problem of audible discontinuities in concatenative synthesis, by interpolating in the formant, waveform, or suitable linear predictive coding domains [e. g. 1, 2, 3, 4]. Commonly, these approaches neither increase synthesis flexibility nor address the issue of compactness. To address these issues, we propose the Asynchronous Interpolation Model (AIM), whose important advantages include (1) a significant reduction in storage requirements, making use of the special properties of an AI, and (2) elimination of concatenation errors, because speech units of the acoustic inventory have identical representations at concatenation points.

2. AIM FORMULATION

The core idea of AIM is to describe a speech region by the varying degrees of influence of preceding and following acoustic events. Previously, we represented a short region (on the order of 5–10 ms) of speech by synthesizing from several non-overlapping feature

This work was supported by NSF Grant 0713617.

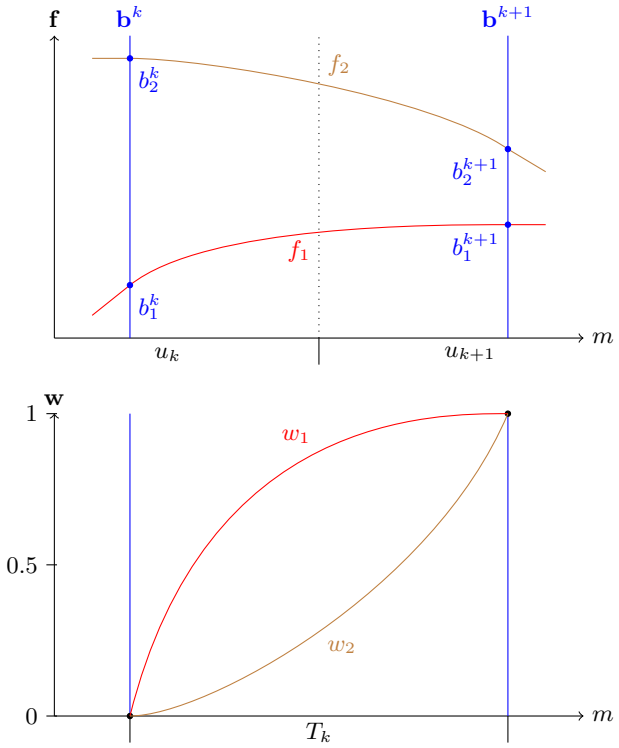


Fig. 1: Transition T_k from acoustic event region u_k to u_{k+1} under AIM, for $N = 2$. Top pane shows feature streams, basis vectors, and their associated acoustic event regions. The dotted line represents the acoustic event region boundary. Bottom pane shows associated interpolation weights. Features and interpolation weights are shown as continuous values for the sake of clarity.

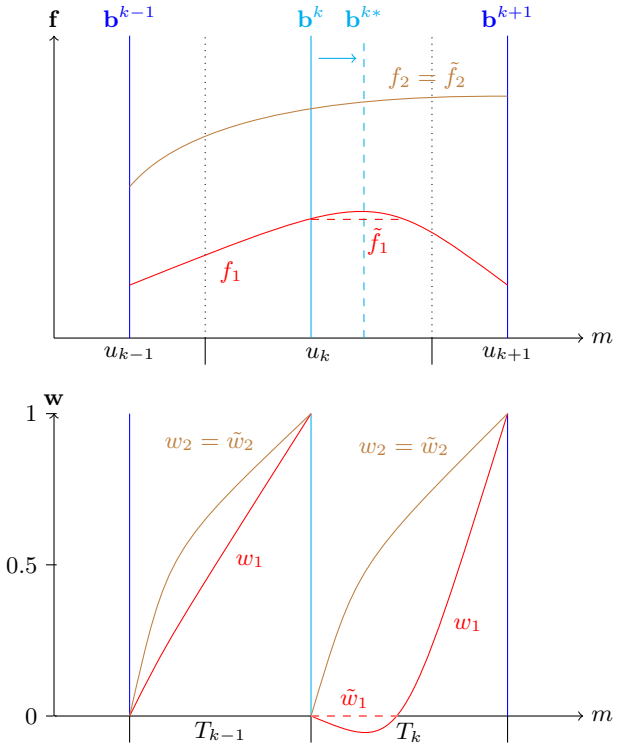


Fig. 2: Optimizing basis vector $\mathbf{b}^k$. The locations of neighboring basis vectors $\mathbf{b}^{k-1}$ and $\mathbf{b}^{k+1}$are fixed, while the best location for the interior basis vector will be determined by minimizing a reconstruction error within the associated acoustic event region u_k (dotted lines represent the acoustic event region boundaries). Original feature streams $\mathbf{f}$ and raw interpolation weights $\mathbf{w}$ are shown with solid, clipped interpolation weights $\tilde{\mathbf{w}}$ and reconstructed feature streams $\tilde{\mathbf{f}}$ in dashed linestyles. We see that the current location of $\mathbf{b}^k$ results in a clipped $\tilde{f}_1$ trajectory (because weights are restricted to be within the range $[0, 1)$, where as the location of $\mathbf{b}^{k*}$, shown as dashed line, would reproduce the original trajectory faithfully.

subsets, each of which were computed by *asynchronous interpolation* (i. e. one interpolation weight per subset) of left and right neighboring *basis vector* features. The locations of basis vectors were associated with particular acoustic event regions such as a phonemes, allophones, or more specialized units, and may contain additional information about phonetic and prosodic context [5, 6, 7, 8]. In this work, we (1) extend the notion of non-overlapping feature subsets to using a special matrix that optimally associates dimension-reduced latent space weights with feature components, and (2) reduce the number of basis vectors by clustering.

Consider a short speech waveform signal that has been divided into M feature frames. Additionally, the location and identity of K acoustic event regions (e. g. phonemes) $u_1, u_2, \ldots, u_K$ associated with the speech signal are known. Given a suitable speech codec, let the short-term speech signal $\mathbf{X}$ at frame m be modeled by a N-dimensional feature vector $\mathbf{f}[m]$. Using a strict diphone assumption, each n^{th} component, or *stream*, of the feature vector $\mathbf{f}$ at frame m is

$$f_n[m] = (1 - w_n[m]) \cdot b_n^k + w_n[m] \cdot b_n^{k+1}, \forall m \in T_k \quad (1)$$

where b_n^k and b_n^{k+1} are the n^{th} components of the *basis vectors* associated with immediately neighboring acoustic events u_k and u_{k+1}, and $w_n[m]$ is the n^{th} *interpolation weight* at frame m, desired to be in the interval $[0, 1]$. The variable T_k denotes the set of frames associated with the transition from basis vector $\mathbf{b}^k$ to $\mathbf{b}^{k+1}$. Figure 1 illustrates the concept.

Other researchers have also represented speech as an interpolation between vectors (e. g. temporal decomposition [9]). Our method differs since we associate basis vectors with a phonemic and prosodic context, and interpolation is carried out asynchronously (i. e. there are several distinct interpolation weight trajectories). Compared to AIM, using an HMM for speech synthesis appears to suffer from several disadvantages, namely (1) the assumption of conditional independence of successive observations is inappropriate for speech (although ameliorated by the usage of delta features), (2) the reliance on relatively large amounts of training data to capture temporal evolution, (3) mathematically complex solutions for trajectory generation [Toda04], and (4) asynchronicity of the data can not be modeled. Whereas an HMM focuses on modeling speech trajectories by dividing them into a sequence of synchronous probabilistic observations, AIM places basis vectors at the beginning and ending of acoustic events and uses asynchronous interpolation weights to explicitly model the transition.

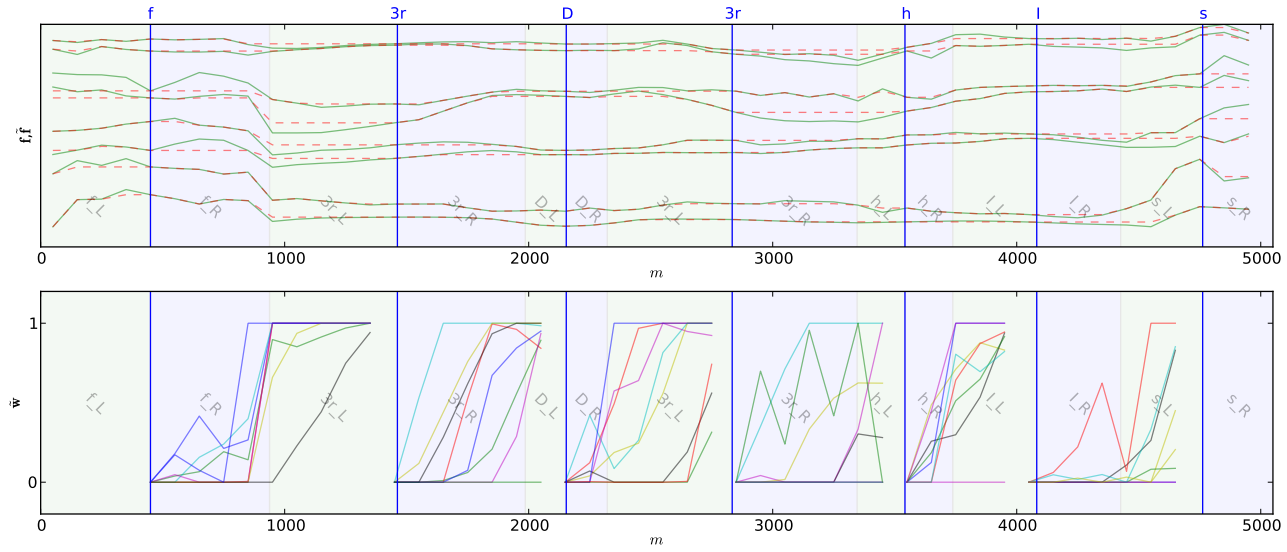


Fig. 3: Example of AIM analysis on the sentence fragment "further his". The top pane shows original features $\mathbf{f}$ (in this case 8$^{\text{th}}$-order line spectral frequencies of a 8000 Hz waveform for illustration clarity) in solid green, and reconstructed features $\tilde{\mathbf{f}}$ in dashed red lines. Basis vector locations are shown with blue vertical lines, with their identity above. The bottom pane shows the associated eight-dimensional clipped interpolation weights $\tilde{\mathbf{w}}$ (with a different color for each component). Demiphone labels are shown with alternating backgrounds.

3. ANALYSIS

3.1. Basis Vector Initialization

For American English, a minimal acoustic event set includes the set of American English phonemes; however, some phonemes contain more than one acoustic event, these are split into several events: diphthongs contain two separate targets (/aI/[1]: "aI1", "aI2"), voiced plosives contain two events for closure and burst (/b/: "bc", "b"), and unvoiced plosives contain three events for closure, burst, and aspiration (/t/: "tc", "t", "th"). Moreover, affricates can be represented as a combination of their constituent events (/tS/: "tc", "t", "S"). The location of acoustic events can be determined manually or from the state locations of a Hidden Markov Model that has been force-aligned to the acoustic inventory.

We initialize basis vector locations to the centers of phoneme labels, or use simple rules for assignement (e. g. for /t/, we actually have phoneme labels /tc/ and /th/, and we initialize the basis vector "tc" to be in the middle of /tc/, "t" 10% into the /th/, and "th" in the middle of the /th/). Given a basis vector location at frame m, its value is equal to the underlying feature streams, or $b_n^k = f_n[m]$.

3.2. Interpolation Weights

Interpolation weights are calculated from Equation 1

$$w_n[m] = \frac{b_n^k - f_n[m]}{b_n^k - b_n^{k+1}} \tag{2}$$

unless $b_n^k - b_n^{k+1}$ is close to zero, in which case all w_n are assigned an arbitrary value. We desire interpolation weight values to be in the interval $[0, 1]$, for several reasons. First, quantization is most efficient when the dynamic range of $\mathbf{w}$ is fixed. Second, the interval guarantees the validity of the resynthesized features; for example, in our LSF-based vocoder, $\mathbf{w}$ values less than zero or greater than one can lead to crossing of LSF trajectories. Third, analysis noise can lead to perceptually insignificant excursions outside of the interval, especially when the denominator of Equation 2 is small. Finally, significant excursions can often be reduced by optimizing the location of basis vectors (see Section 3.3), or are evidence for the necessity to add additional basis vector (outside of the scope of this paper). To address these issues, we clip the interpolation weight values to be inside the interval $[0, 1]$. The resuling clipped interpolation weights are denoted $\tilde{\mathbf{w}}$, and the associated resynthesized features are denoted $\tilde{\mathbf{f}}$, as shown in Figure 2. Figure 3 shows an example analysis using real speech data.

3.3. Basis Vector Optimization

Because the clipping of interpolation weights introduces imperfections during resynthesis, we optimize basis vector locations (and thus also values) as follows: For a speech region containing the frames associated with the set of basis vectors $\{b^{k-1}, b^k, b^{k+1}\}$, we search for the frame location of the center basis vector b^k that minimizes the local reconstruction error over the two transitions T_{k-1} and T_k

$$E = \sum_{m \in T_{k-1},\, T_k} \left| \mathbf{f}[m] - \tilde{\mathbf{f}}[m] \right|^2$$

in a process that first analyzes, then resynthesizes feature trajectories based on the model. Note that since $\mathbf{f}$ is an LSF vector the error E does not directly relate to minimizing spectral distortion, and furthermore the error is unlikely to correlate well with perpectually-judged acoustic distances.

[1] Phoneme names are specified using Worldbet notation, and basis vector names are derived from them.

The search for the best basis vector location of b^k is performed over all frames bounded by the associated acoustic event region u_k and other basis vectors within that region. Figure 2 illustrates the optimization process (for the case of a single basis vector within u_k). We repeat this optimization for $k = 2, \ldots, K-1$, or all interior basis vectors. After all interior basis vectors are consecutively optimized, we iteratively repeat the entire procedure, since the optimal location of a medial basis vector depends on its neighbors. This is performed until convergence (usually two or three passes are sufficient).

3.4. Latent Space Weights

We re-write the interpolation Equation 1 in vector form

$$\mathbf{f}\,[m] = \begin{bmatrix} (1 - w_1[m]) \cdot b_1^k + w_1[m] \cdot b_1^{k+1} \\ (1 - w_2[m]) \cdot b_2^k + w_2[m] \cdot b_2^{k+1} \\ \vdots \\ (1 - w_N[m]) \cdot b_N^k + w_N[m] \cdot b_N^{k+1} \end{bmatrix} \quad . \qquad (3)$$

If all of the interpolation weights $\mathbf{w}[m]$ are independent, we have N degrees of freedom in the weights, with all streams moving asynchronously. At the opposite extreme, if all the weights are identical, then we have a single degree of freedom in the weights, with all streams moving synchronously. This is the minimal set of weights from which to build the stream $\mathbf{f}\,[m]$. More generally, the set of weights can be compressed by positing $1 \le P \le N$ independent degrees of freedom that generate the N-dimensional weights. Maximal compression with synchronous streams corresponds to $P = 1$.

We approximate the clipped interpolation weights by writing a dimension-reduced approximation to the full N-dimensional weights for frame m as

$$\tilde{\mathbf{w}}_{1\times N}[m] \approx \hat{\mathbf{w}}_{1\times N}[m] = \mathbf{\Lambda}_{1\times P}\,[m]\;\; V^T_{P\times N} \qquad (4)$$

where $\mathbf{\Lambda}_{1\times P}\,[m]$ are the P-dimensional *latent-space* weights, and V^T is a $P \times N$ matrix. Equation 4 constructs $\hat{\mathbf{w}}[m]$ as an embedding of the P-dimensional latent weights $\mathbf{\Lambda}\,[m]$ into the N-dimensional space of $\tilde{\mathbf{w}}$.

Previously we constructed the matrix V^T with entries consisting of zeros and ones, hence constraining groups of interpolation weights to be identical; these groups were chosen by minizing a reconstruction error over a large number of possible non-overlapping partitions of a given feature set [8]. Here, we discover the appropriate constraints statistically, using principal component analysis to determine the matrix V^T and the latent-space weights $\mathbf{\Lambda}[m]$. Suppose the feature streams pass through the left and right basis vectors b^k and b^{k+1}, in the first and last frame of a segment respectively. Then, from Equation 3 the weight vector passes through $[0, 0, 0, \ldots]$ and $[1, 1, 1, \ldots]$ in those frames. Hence the diagonal of the N-dimensional hypercube forms a convenient geometric reference for the weight trajectory, from which we measure *deviations* of the weight vectors. (In fact, for the case of a single weight, $P = 1$, the weight trajectory is *along* along this diagonal.) Figure 4 shows an example trajectory for $N = 3$, and $P = 2$. The example trajectory lies in the 2-dimensional hyperplane defined by the diagonal of the cube and one direction orthogonal to the diagonal.

This geometric decomposition constrains the matrix V^T; its first row is the unit N-vector along the hypercube diagonal

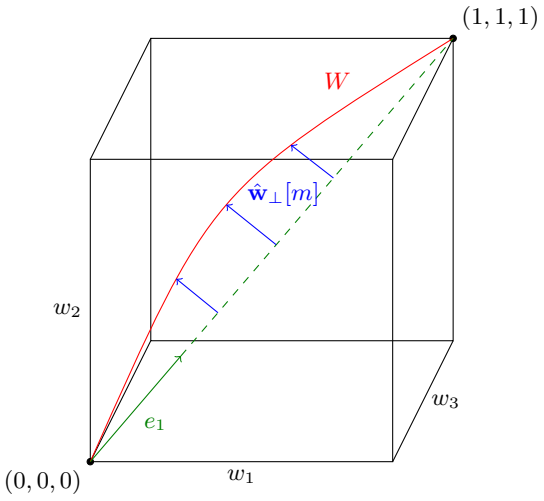


Fig. 4: Geometry of weight trajectories. Three-dimensional weight space with two-dimensional trajectory. The trajectory is in the plane defined by the diagonal vector e_1, and the vectors $\hat{\mathbf{w}}_\perp[m]$ orthogonal to it (and all co-planar here).

$$e_1^T \;=\; \frac{1}{\sqrt{N}}\,[1, 1, 1, \ldots] \quad .$$

We want the dimension-reduced weights $\hat{\mathbf{w}}[m]$ closest in mean-square to the true clipped weights $\tilde{\mathbf{w}}[m]$. These are found by principal component analysis on the deviation of the weights from the hypercube diagonal e_1. First, solve Equation 3 for the true weights and clip to the interval $[0, 1]$. Next project out the components along and orthogonal to the unit diagonal (see Figure 4)

$$\begin{aligned} \Lambda_1 \;&\equiv\; \tilde{\mathbf{w}}[m] \cdot e_1 \\ \mathbf{w}_\perp[m] \;&\equiv\; \tilde{\mathbf{w}}[m]\;\left(1 \;-\; e_1\, e_1^T\right) \end{aligned} \qquad (5)$$

where $\tilde{\mathbf{w}}[m]$ and $\mathbf{w}_\perp[m]$ are N-dimensional row vectors.

Rows 2–P of V^T follow from principal component analysis (PCA) of the *deviation weights* $\mathbf{w}_\perp$. In practice, we implement the PCA by singular value decomposition (SVD) on the $M \times N$ matrix of deviation weights

$$W^\perp_{M\times N} \equiv \left[\;\; \mathbf{w}_\perp[1]^T \quad \mathbf{w}_\perp[2]^T \quad \cdots \quad \mathbf{w}_\perp[M]^T \;\;\right]^T . \qquad (6)$$

We use SVD to decompose $W^\perp$ as

$$W^\perp \;=\; U\,S\,V^T, \qquad (7)$$

where S is the $N \times N$ diagonal matrix of singular values (one of which will be identically zero due to the projection orthogonal to e_1), and V is the $N \times N$ matrix of eigenvectors of the correlation matrix of the $\mathbf{w}_\perp[m]$. Retaining the leading $P-1$ eigenvectors (columns of V in the $N \times (P-1)$ matrix V_{P-1}), we obtain the trailing $P-1$ latent-space weights $\Lambda_\perp$ by projection

$$\Lambda_\perp[m] \;\equiv\; \mathbf{w}_\perp[m] \cdot V_{P-1} \quad .$$

Next, construct the $N \times P$ embedding matrix by pre-pending e_1 to V_{P-1}

$$V_P \;=\; [e_1\; V_{P-1}]$$

and reconstruct the weights by $\hat{\mathbf{w}}[m]$ from the embedding Equation 4 with $\Lambda[m] = [\,\Lambda_1[m]\;\Lambda_\perp[m]\,]$.

P	Parameters	Error [dB]
1	96,863	2.46
2	110,866	2.35
4	138,872	2.21
8	194,884	1.97
18 (uncompressed)	251,730	1.44

Table 1: Number of parameters and average log spectral distortion in dB for different values of latent dimension P. The last row represents the uncompressed case when $P = N = 18$.

4. COMPRESSION EXPERIMENTS

In this section we report on experiments designed to reduce the storage size of acoustic inventory features. Let $\mathbf{F}_{M\times N}$ be M frames of an N-dimensional feature trajectory. The total number of parameters required to model $\mathbf{F}$ using AIM is

$$Parameters = B \cdot N + E \cdot 2 + M \cdot P + N \cdot (P-1) \quad (8)$$

where B is the number of N-dimensional basis vectors in a pool, $E \cdot 2$ is the number of parameters required to store basis vector event times and basis vector identity (indexing into the pool of basis vectors), $M \cdot P$ is the number of latent space weights, and $N \cdot (P-1)$ is the size of the embedding matrix V_{P-1} (see Section 3.4)[2]. We will reduce the total number of parameters by (1) reducing the dimensionality $P \ll N$ of interpolation weights, and (2) clustering similar basis vectors, thus reducing the number of basis vectors, B.

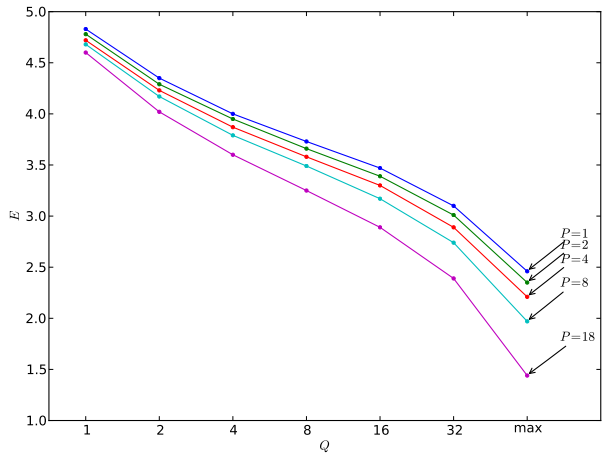
We used a small acoustic inventory consisting of isolated diphones, spoken by an American English male speaker, sampled at 16 kHz, and labeled with demiphones. We analyzed the waveforms using a linear predictive coding (LPC) based vocoder, resulting in voicing, fundamental frequency (F0), energy, and filter parameters for each of the $M = 13,985$ frames. We converted all-pole filter coefficients into $N = 18$ dimensional line spectral frequencies (LSF) and applied AIM to the LSF trajectories, resulting in $B = E = 4143$ basis vectors. The total number of original feature parameters, or the size of $\mathbf{F}$ is $13,985 \cdot 18 = 251,730$.

After calculating clipped interpolation weights, and resynthesizing features $\tilde{\mathbf{f}}$, we converted the LSF trajectories back to the filter coefficient domain. We evaluted the associated original and reconstructed frequency responses $S(w)$ and $\tilde{S}(w)$, and then calculated the mean log spectral distortion (LSD)

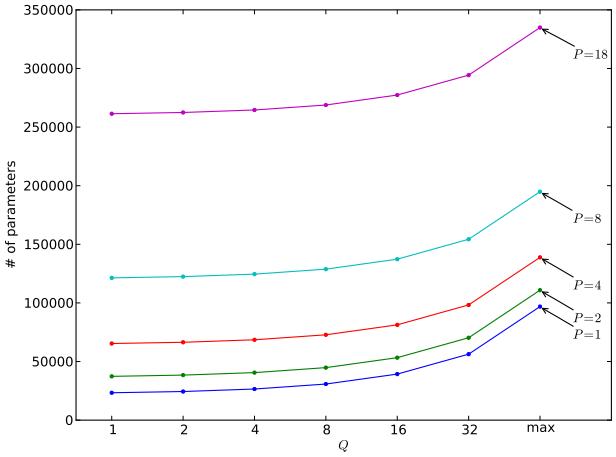
$$E = \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{512}\sum_{w=1}^{512}\left(10\log_{10}\frac{S(w)}{\tilde{S}(w)}\right)^2} \quad (9)$$

over all frames of the acoustic inventory, resulting in an error of 1.4 dB. Informal listening experiments show that acoustic differences between sentences resynthesized from $\mathbf{f}$ versus $\tilde{\mathbf{f}}$ are nearly imperceptible, although this is likely dependent on the choice of vocoder.

[2]For the sake of simplicity we ignore additional compression strategies, such as scalar quantization.



(a) Average log spectral distortion in dB.



(b) Number of parameters.

Fig. 5: The effect of different values of latent dimension P and basis vector classes Q per basis vector symbol. $P = 18$ represents the case when interpolation weights are uncompressed, and $Q = \max$ represents the case when the basis vector pool remains uncompressed.

4.1. Compressing Interpolation Weights

We constructed the deviation weights $W^{\perp}$ according to Equations 5 and 6, and then decomposed the weights using Equation 7.[3] We then constructed the estimate of the original interpolation weights $\hat{\mathbf{w}}$ using Equation 4, and resynthesized an approximation to features $\tilde{\mathbf{f}}$. We calculated the number of parameters using Equation 8 and the average LSD using Equation 9. The results are shown in Table 1. As expected, we see that for lower P the reconstruction error increases, while the number of parameters decreases.

[3]Incidentally, when performing a SVD on $\tilde{\mathbf{w}}$ directly (without the initial projection), the value of the first eigenvector e_1 is very close to the normalized unit vector $\frac{1}{\sqrt{N}}[1,1,1,\ldots]^T$. This means that SVD on the deviation weights $\mathbf{w}_{\perp}$ is close to optimal.

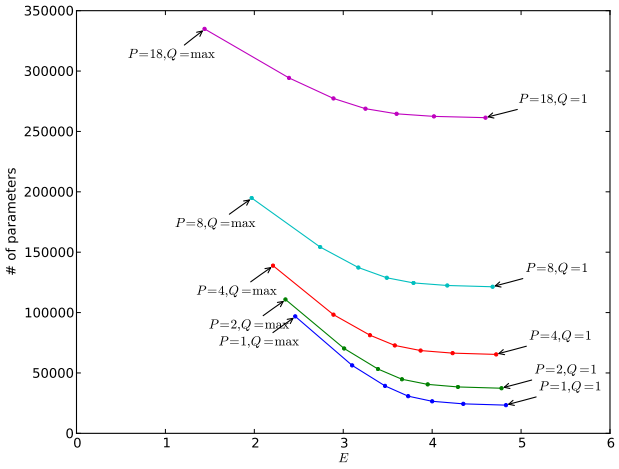


Fig. 6: Relationship between number of parameters and log spectral distortion in dB. For each trajectory, the points represent values of $Q = 1, 2, 4, 8, 16, 32$. $P = 18$ and $Q = \max$ represent the uncompressed interpolation weight and the uncompressed basis vector pool cases, respectively.

4.2. Sharing Basis Vectors

Two different basis vector occurrences with the same acoustic event label can be treated either as identical or as unique. During initial analysis, every basis vector in the inventory is considered unique. This can still provide a high rate of compression because the majority of frames are within transitions and can thus be represented by low-dimensional latent space weights alone. However, for further space savings, we can also share similar basis vector values. In the extreme case, all basis vectors with the same label are shared (i. e. different basis vector locations are represented by a single basis vector value).

We used a vector quantization (VQ) method to explore the effect of basis vector sharing on the average LSF. For each basis vector symbol (e. g. "A"), we first aggregated all basis vectors with that symbol into a pool. We then created a VQ codebook, using multiple random initializations, and kept the one with the lowest average Euclidean distance between data and codewords. Finally, we vector quantized the pool into Q codewords. When using this method, the variable B in Equation 8 is given by $B = Q \cdot U$, where U is the number of unique basis vector symbols. In our case $U = 59$.

For $P = 1, 2, 4, 8, 18$ and for $Q = 1, 2, 4, 8, 16, 32$, we resynthesized an approximation to features $\tilde{\mathbf{f}}$. We calculated the number of parameters using Equation 8 and the average LSD using Equation 9. Results are shown in Figures 5a and 5b. In addition, we have included results from the previous Section, marked as $Q = \max$, which means that we used all available data. (The exact number of datapoints per basis vector symbol class varies, but there are, on average, about 80 examples of each basis vector symbol in our acoustic inventory). As expected, the error is lower for higher values of P, given a fixed Q. Similarly, the error is lower for a higher Q, given a fixed P. We note that for P close to N AIM is inefficient, mostly because of the additional overhead of storing basis vector locations and identities. The relationship between number of parameters and LSD, or the quality-size tradeoff, is shown in Figure 6.

The case of $P = 1$ and $Q = 1$ represents the highest possible compression, and in this configuration there are 59 18-dimensional basis vectors (1062 parameters), 4143 basis vector location and identity values (8286 parameters), 13,985 interpolation weight parameters, and 18 parameters for matrix V_{P-1}, for a total of 22,348 parameters, or a compression ratio of about 1:11. This ratio would increase with increased acoustic inventory size.

5. CONCLUSION

We have extend the notion of non-overlapping feature subsets to using a embedding matrix that maps latent space weights to interpolation weights. The embedding matrix is found automatically through singular value decomposition of the interpolation weight trajectories. Experiments showed that this is effective in compressing the number of parameters. Further compression can be gained by vector quantizing the pool of basis vectors associated with each basis vector symbol. The highest compression ratio of 1:11 results in a log spectral distortion of 4.83 dB relative to uncompressed features.

In future work, we will research a basis vector optimization method that considers all transitions involving a particular basis vector symbol simultaneously during optimization. W will also study whether the vector-quantized basis vectors correspond to specific phonetic contexts. Finally, we plan on improving the overall framework by performing latent-space dimension reduction on the the features directly, which more closely optimizes the final error criterion.

6. REFERENCES

[1] H. Mizuno, M. Abe, and T. Hirowaka. Waveform-based speech synthesis approach with a formant frequency modification. In *ICASSP*, pages 195–198, 1993.

[2] J. Wouters and M. Macon. Control of spectral dynamics in concatenative speech synthesis. *IEEE Trans. Speech and Audio Proc.*, 9(1):30–38, January 2001.

[3] D. T. Chappell and J. H. L. Hansen. A comparison of spectral smoothing methods for segment concatenation based speech synthesis. *Speech Communication*, 36(3):343–373, 2002.

[4] P. H. Low, C. H. Ho, and S. Yaseghi. Using estimated formant tracks for formant smoothing in text to speech synthesis. In *ASRU*, pages 688–693, 2003.

[5] A. Kain and J. van Santen. Compression of acoustic inventories using asynchronous interpolation. In *IEEE Workshop on Speech Synthesis*, pages 83–86, 2002.

[6] A. Kain and J. van Santen. A speech model of acoustic inventories based on asynchronous interpolation. In *EUROSPEECH*, pages 329–332, 2003.

[7] A. Kain and J. van Santen. Unit-Selection Text-to-Speech Synthesis Using an Asynchronous Interpolation Model. *Proceedings of 6th ISCA Workshop on Speech Synthesis*, August 2007.

[8] R. Moldover and A. Kain. Compression of line spectral frequency parameters with asynchronous interpolation. In *Proceedings of ICASSP*, April 2009.

[9] B. Atal. Efficient coding for LPC parameters by temporal decomposition. In *ICASSP*, pages 81–84, 1983.