LSM–BASED BOUNDARY TRAINING FOR CONCATENATIVE SPEECH SYNTHESIS

Jerome R. Bellegarda

Speech & Language Technologies
Apple Computer, Inc.
Cupertino, California 95014

ABSTRACT

The level of quality that can be achieved in concatenative text-to-speech synthesis depends, among other things, on a judicious chiseling of the inventory used in unit selection. Unit boundary optimization, in particular, can make a huge difference in the users' perception of the concatenated acoustic waveform. This paper considers the iterative refinement of unit boundaries based on a data-driven feature extraction framework separately optimized for each boundary region. Such unsupervised boundary training guarantees a globally optimal cut point between any two matching units in the inventory. This optimization is objectively characterized, first in terms of convergence behavior, and then by comparing the average inter-unit discontinuity obtained before and after training. Experimental results and listening evidence both underscore the viability of this approach for unit boundary optimization.

1. INTRODUCTION

In concatenative text-to-speech (TTS) synthesis, the acoustic signal is generated from pre-recorded speech segments, normally extracted from a large database with varied phonetic and prosodic characteristics. The selection of the best unit sequence is cast as a multivariate optimization task, where the unit inventory is searched to minimize suitable cost criteria across the whole target utterance [1]. In practice, it is often necessary to modify the chosen units in order to reduce audible discontinuities, and/or more precisely match the target prosody [2]. Because any such manipulation is liable to degrade signal quality, it is highly desirable to select units for which the minimum amount of post-processing is required [3]. This in turn has implications on inventory construction.

Audible smoothness is arguably most affected by how speech units are cut after recording: boundary placement critically influences how much discontinuity one is likely to encounter after concatenation, and thus how natural synthetic speech will sound [4]. Boundaries are typically derived using automatic segmentation algorithms operating at the level of the smallest unit considered, be it a phone, diphone, syllable, etc. Such algorithms tend to rely on either dynamic time warping (e.g., [5]) or hidden Markov modeling (e.g., [6]). None of them, however, calculates the *globally optimal* cut point between two contiguous units given the entire recorded inventory. Instead, they yield a *locally optimal* cut point given these two specific units (and the underlying models). As a result, the set of boundaries so obtained may not be particularly well suited to the task of concatenating different units.

For highest TTS quality, it would thus be desirable to hand-check every cut point for global consistency.[1] In recent years, however, the number of units has grown too large for such exhaustive optimization. The outcome is often a somewhat uneven performance, where the TTS system may well sound very good in general but still regularly break down, in ways that are difficult to predict from unit inventory statistics.

[1] In the case of diphone synthesis, for example, manual or semi-manual boundary adjustment has been shown to make a huge difference in the users' perception of the concatenated acoustic waveform [7].

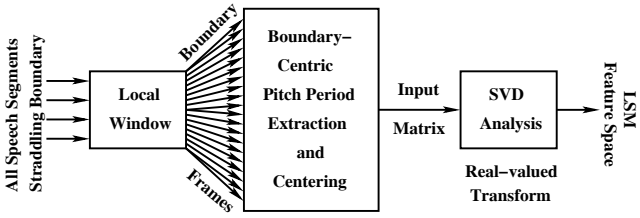

Fig. 1. *LSM Feature Extraction Framework.*

What seems to be needed is a method to systematically optimize all unit boundaries before unit selection, so as to effectively minimize the likelihood of a really bad concatenation. This would guarantee that at run time, uniformly high quality units are available to choose from. We refer to this (off-line) optimization as the data-driven "training" of the unit inventory, by analogy to the (run time) "decoding" process embedded in unit selection. Note that, since such unsupervised training inherently requires a global criterion, the intrinsically local cost functions typically used for unit selection (see, e.g., [8]) do not seem adequate.

On the other hand, we have recently introduced an alternative TTS feature extraction framework [9], based on the *latent semantic mapping* (LSM) paradigm [10], which leads to a global discontinuity metric for characterizing the acoustic (dis-)similarity between two candidate segments [11]. By leveraging this global outlook, it becomes possible to take into account all potentially relevant units in an iterative manner, and thereby carry out the desired training of individual boundaries.

The paper is organized as follows. The next section briefly reviews the LSM framework. In Section 3, we describe in greater details the global criterion used to evaluate inter-unit concatenations. Section 4 presents the iterative procedure followed for boundary training. Finally, in Section 5 we report on experimental evaluations which illustrate the salient characteristics of the approach.

2. LSM FEATURE EXTRACTION

LSM features are derived using the framework illustrated in Fig. 1, in which a modal analysis of the signal is carried out through a pitch synchronous real-valued transform in each boundary region of interest [9]. To fix ideas, consider among the set of recorded utterances the collection of all possible speech segments ending and starting within the phoneme P, so we can concentrate on a

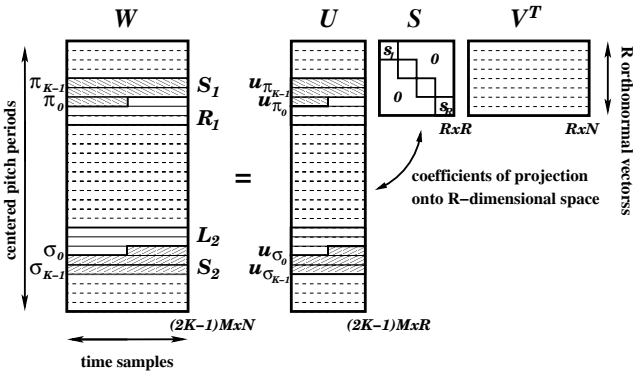

Fig. 2. *Decomposition of the Input Matrix.*

(diphone-style) concatenation within P. Denote by S_1-R_1 and L_2-S_2 two such acoustic segments, where S_1 ends somewhere within P and R_1 comprises the contiguous second half of the segment, and S_2 starts somewhere within P and L_2 comprises the contiguous first half of the segment. We focus on the concatenation S_1-S_2, which is not available in the unit inventory.

Let $\pi_{-K+1} \ldots \pi_0 \ldots \pi_{K-1}$ denote the $2K-1$ centered[2] pitch periods associated with the boundary region of S_1-R_1, such that the boundary between S_1 and R_1 falls exactly in the middle of π_0. Similarly, let $\sigma_{-K+1} \ldots \sigma_0 \ldots \sigma_{K-1}$ denote the $2K-1$ centered pitch periods associated with the boundary region of L_2-S_2, such that the boundary between L_2 and S_2 falls exactly in the middle of σ_0. For voiced speech segments, each pitch period is obtained through conventional pitch epoch detection (see, e.g., [12]). For voiceless segments, the time domain signal is similarly chopped into analogous, albeit constant-length, portions.

Further assume that there are M segments like S_1-R_1 and L_2-S_2 present in the unit inventory, i.e., with a boundary in the middle of the phoneme P, and that for each of these we have extracted the relevant centered pitch periods as described above. This results in $(2K-1)M$ pitch periods in total, encapsulating the entire boundary region. Assuming N denotes the maximum number of samples observed in each of these centered pitch periods, we symmetrically zero-pad and appropriately window all centered pitch periods to N, as necessary. The outcome is a $((2K-1)M \times N)$ matrix W with elements w_{ij}, where each row c_i corresponds to a centered pitch period, and each column t_j corresponds to a slice of time samples. This matrix W is illustrated in the left-hand side of Fig. 2. Typically, M is on the order of a few thousands, N of a few hundreds, and a reasonable value for K is $K = 3$.

At this point we perform the eigenanalysis of W via singular value decomposition (SVD) as [11]:

$$W = U\,S\,V^{\,T}\,, \tag{1}$$

where U is the $((2K-1)M \times R)$ left singular matrix with row vectors u_i $(1 \le i \le (2K-1)M)$, S is the $(R \times R)$ diagonal matrix of singular values $s_1 \ge s_2 \ge \ldots \ge s_R > 0$, V is the $(N \times R)$ right singular matrix with row vectors v_j $(1 \le j \le N)$, $R < \min(N, (2K-1)M)$ is the order of the decomposition, and ^{T} denotes matrix transposition. As is well known, both left and right singular matrices U and V are column-orthonormal, i.e., $U^{\,T}U = V^{\,T}V = I_R$ (the identity matrix of order R). Thus, the column vectors of U and V each define an orthornormal basis for the space of dimension R spanned by the (R-dimensional) u_i's and v_j's. By analogy with the latent semantic analysis framework,[3] the resulting feature space is called the *LSM space* $\mathcal{L}$ [10].

The interpretation of (1) in Fig. 2 focuses on the orthornormal basis obtained from V. Projecting the row vectors of W onto that basis defines a representation for the centered pitch periods in terms of their coordinates in this projection, namely the rows of US. Thus, (1) defines a mapping between the set of centered pitch periods and (after appropriate scaling by the singular values) the set of R-dimensional feature vectors $\bar{u}_i = u_i S$. These vectors can then be viewed as feature vectors analogous to, e.g., the usual cepstral vectors.

In contrast to such traditional Fourier-derived features, the relative positions of the LSM vectors in the space $\mathcal{L}$ are determined by the overall characteristics observed in the relevant pitch periods, as opposed to an analysis restricted to a particular instance (be it frequency domain processing or otherwise). Hence, two vectors $\bar{u}_k$ and $\bar{u}_\ell$ "close" (in some suitable metric) to one another in $\mathcal{L}$ can be expected to reflect a high degree of similarity in the relevant pitch periods, and thus potentially a small amount of perceived discontinuity in the ensuing concatenated acoustic signal [9], [11].

3. CONCATENATION EVALUATION CRITERION

Since, for the phoneme P, M segments from the unit inventory straddle the boundary, there are M boundaries to consider. These M boundaries must be jointly optimized so that all M^2 possible concatenations exhibit minimal discontinuities.[4] In the space $\mathcal{L}$, the unit boundary optimization problem therefore boils down to minimizing the convex hull of all vectors associated with all possible π_0. To carry out this task, we first have to express the concatenation point (or, more precisely, the centered pitch period straddling the concatenation) in $\mathcal{L}$.

Note that the feature space comprises, in particular, the vectors $\bar{u}_{\pi_k}$ and $\bar{u}_{\sigma_k}$, representing the centered pitch periods π_k and σ_k, respectively (for $-K+1 \le k \le K-1$). Consider now the potential concatenation S_1-S_2 of these two units, shown as the shaded area in Fig. 2. This concatenation can be expressed as $\pi_{-K+1} \ldots \pi_1\ \delta_0\ \sigma_1 \ldots \sigma_{K-1}$, where δ_0 represents the concatenated centered period (i.e., consisting of the left half of π_0 and the right half of σ_0). This sequence will in turn have a corresponding representation in $\mathcal{L}$ given by:

$$\bar{u}_{\pi_{-K+1}}\ \ldots\ \bar{u}_{\pi_1}\ \bar{u}_{\delta_0}\ \bar{u}_{\sigma_1}\ \ldots\ \bar{u}_{\sigma_{K-1}}\,. \tag{2}$$

The only vector not directly associated with a row in the original input matrix W is $\bar{u}_{\delta_0}$. However, it can easily be calculated by treating δ_0 (a row vector of dimension N) as an additional row of the matrix W. From [11], the representation of that additional row is obtained as:

$$\delta_0 = u_{\delta_0}\,S\,V^{\,T} = \bar{u}_{\delta_0}\,V^{\,T}\,, \tag{3}$$

[2] As will become clear shortly, the main advantage of a *centered* representation is that the boundary can be precisely characterized by a single vector in the resulting feature space [11]. This is in contrast with usual representations where the boundary is normally inferred *a posteriori* from the position of the two vectors on either side.

[3] This is where the expression "semantic" in LSM comes from, although in the present context "global" would be a more accurate terminology.

[4] Clearly, M of these concatenations already exhibit minimal discontinuity, since they occur between contiguous segments (such as S_1-R_1 and L_2-S_2 mentioned earlier). Without loss of generality, we do not explicitly remove them from the analysis.

where the R-dimensional vector u_{δ_0} acts as an additional row of the matrix U. Hence the *concatenation vector*: $\bar{u}_{\delta_0} = u_{\delta_0} S = \delta_0 V$ corresponds to the representation of δ_0 in $\mathcal{L}$.

Given $\bar{u}_{\delta_0}$, the discontinuity brought about by this concatenation can be expressed as the cumulative difference in closeness between the vectors composing the two segments before and after concatenation. Recall from [9], [11] that the expression for the closeness between two individual vectors is given by:

$$c(\bar{u}_k, \bar{u}_\ell) = \cos(u_k S, u_\ell S) = \frac{u_k \, S^2 \, u_\ell^T}{\|u_k S\| \, \|u_\ell S\|} \,, \qquad (4)$$

for any $1 \leq k, \ell \leq (2K-1)M$. Introducing the shorthand notation:

$$\tilde{c}(u_{\sigma_{-k}}, u_{\sigma_0}, u_{\sigma_k}) = \frac{c(\bar{u}_{\sigma_{-k}}, \bar{u}_{\sigma_0}) + c(\bar{u}_{\sigma_0}, \bar{u}_{\sigma_k})}{2} \,, \qquad (5)$$

for the average closeness across the boundary σ_0, we can therefore write:

$$d(S_1, S_2) = \sum_{k=1}^{K-1} 2\,\tilde{c}(u_{\pi_k}, u_{\delta_0}, u_{\sigma_k}) - \tilde{c}(u_{\pi_k}, u_{\pi_0}, u_{\pi_{-k}}) - \tilde{c}(u_{\sigma_{-k}}, u_{\sigma_0}, u_{\sigma_k}) \,. \qquad (6)$$

This entity can be thought of as the relative cumulative change in similarity that occurs during concatenation over the entire boundary region considered. This, in essence, corresponds to the trajectory difference before and after concatenation, as expressed in the LSM feature space. An important special case is when the two segments considered are in fact contiguous in the database, i.e., the σ's are identically equal to the π's. In this situation, it can be easily verified that, in particular, $\delta_0 = \sigma_0 = \pi_0$. We conclude that this metric exhibits the property: $d(S_1, S_2) \geq 0$, with equality if and only if $S_1 = S_2$. We refer to (6) as the *discontinuity score* between S_1 and S_2. The closer the discontinuity score is to zero, the more attractive the concatenation. Conversely, a large value of the discontinuity score tends to be correlated with a perceptibly bad concatenation [11].

4. ITERATIVE PROCEDURE

Once the global concatenation criterion (6) is specified, the iterative procedure follows the flowchart of Fig. 3. The basic idea is to focus on each possible boundary region in turn, compute the LSM space associated with this region, adjust individual boundaries one at a time in that space, update the boundary region accordingly, and iterate until convergence.

The initialization step can be performed in a number of different ways. For example, in the case of diphone cut points, the initial boundaries can be obtained directly from diphone segmentation (cf. [5]), or inferred indirectly after phoneme alignment (cf. [6]). In the latter case, they can be determined in the usual manner based on where the speech waveform varies the least, or simply taken to be at the midpoint of the phone. In practice, we have found little difference in behavior based on these various forms of initialization. It is worth pointing out that the last solution is the most expedient, since it does not require a very accurate phoneme aligner: only a reasonable estimate of the phoneme boundaries is required to calculate a plausible mid-phoneme cut point.

Once initialization is completed, we proceed as detailed earlier to gather the $2K-1$ centered pitch periods and derive the resulting

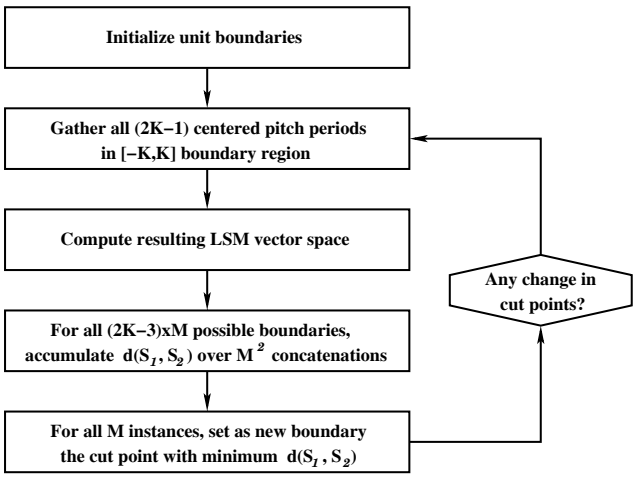

Fig. 3. *Iterative Training of Unit Boundaries.*

LSM space $\mathcal{L}$. This results in $(2K-1)M$ feature vectors in the LSM space, and hence $(2K-3)M$ potential new boundaries.[5] For each of them, there are M^2 possible concatenations, hence we compute the average discontinuity associated with the potential new boundary by accumulating (6) over this set of M^2 possible concatenations. At this point, we obtain $2K-3$ discontinuity measures for each unit instance. As new boundary for each of them, we retain the cut point associated with the minimum value over these $2K-3$ discontinuity scores. The new boundaries form the basis for a new boundary region, and the procedure iterates until no change in cut points is necessary. The last boundaries are therefore globally optimal across the entire set of observations for the phoneme P.

5. EXPERIMENTAL RESULTS

We now briefly summarize some of the results we have obtained on a phonetically and prosodically varied voice database currently deployed in MacinTalk, Apple's TTS offering on MacOS X. This database is fairly similar to the Victoria corpus described in detail in [13]. In particular, recording conditions closely follow those mentioned in [13], though individual utterances generally differ.

For the purpose of illustration, we focus on five representative phonemes (denoted below in SAMPA computer readable phonetic notation, cf. [14]): an example of a steady spectrum vowel, $P = [U]$; an example of a steady spectrum fricative, $P = [Z]$; an example of a nasal, $P = [N]$; an example of a varying spectrum fricative, $P = [h]$; and an example of a varying spectrum vowel (diphtong), $P = [OI]$. The last two instances are especially challenging: $[OI]$ because the rapid changes in acoustic targets occurring in the middle of the phoneme tend to complicate the search for an optimal cut point, and $[h]$ because it often exhibits analogous behavior, compounded by unpredictable voicing characteristics throughout each observation.

The first item of interest is the convergence of the iterative procedure proposed in Fig. 3. Note that, as boundaries shift from one iteration to the next, so does the boundary region for the phoneme considered. Since there is a one-to-one mapping between the bound-

[5] For simplicity, and without loss of generality, we do not allow a boundary at either extreme of the boundary region.

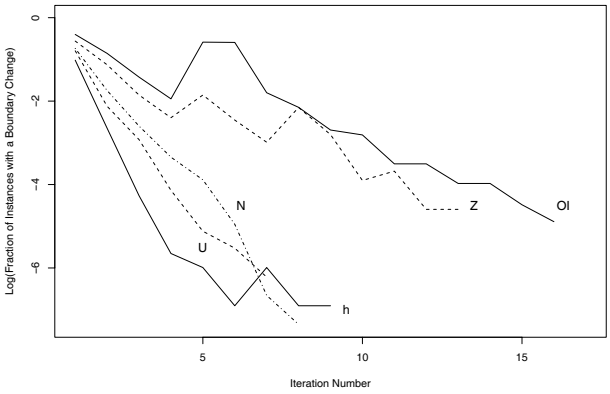

Fig. 4. *Convergence Behavior (Number of Boundary Changes).*

ary region and the LSM space (cf. Section 2), this in turn implies that the LSM space does not stay static. This makes it particularly challenging to derive a theoretical proof of convergence. On the other hand, as long as after each iteration the LSM space remains "sufficiently close" to its previous incarnation, it is intuitively reasonable to expect the procedure to indeed converge.

For each of the five representative phonemes selected, we extracted from the recorded set of utterances all instances of speech segments (in this case, diphones) with a left or right boundary falling in the middle of the phoneme P. There were a total of 502 such instances for $[U]$, 198 instances for $[Z]$, 1559 instances for $[N]$, 1997 instances for $[h]$, and 266 instances for $[OI]$. Then, for each instance, we extracted $K = 3$ pitch periods on the left and $K = 3$ pitch periods on the right of the boundary, leading to $2K - 1 = 5$ centered pitch periods. For $[OI]$, for example, this led to a (1330×350) input matrix. We then followed the iterative procedure described in the previous section, with a value $R = 5$, to derive the globally optimal cut point in each instance.

We assessed convergence by measuring, after each iteration, the fraction of instances for which a boundary change was observed. The outcome is plotted in Fig. 4 (note the log scale on the y-axis). This plot shows that in all cases the boundary training procedure does converge exponentially, though not necessarily monotonically. The occasional loss of monotonicity can be directly traced to disturbances introduced in the LSM space from one iteration to the next. Convergence seems to be roughly twice as slow for $[Z]$ and $[OI]$ as for the other phonemes, although this behavior could have more to do with small sample size than intrinsic difficulty in finding the global optimum. In any event, convergence is reliably attained across all phonemes within a relatively small number of iterations.

After each iteration we also computed the average inter-unit discontinuity score across all possible concatenations. We observed that boundary training yields a consistent reduction of 20 to 30% in the inter-unit discontinuity, suggesting a comparatively reduced likelihood of producing a bad concatenation. To illustrate, the attached files "Before.aiff" and "After.aiff" give two renditions of the nonsense word "boyb" (pronounced $[bOIb]$), slowed down to a speaking rate of approximately 10 words per minute for emphasis. In both cases, the only concatenation between non-contiguous segments occurs within the phoneme $[OI]$. The only difference between the two renditions concerns the boundaries in the unit inventory. In the first case, the inventory featured baseline boundaries (placed, as commonly done in the case of diphone-style cut points, where the phone varies the least). In the second one, the inventory featured optimized boundaries (produced as detailed above). The noticeable improvement that can be heard exemplifies the general benefits of the technique for concatenative TTS synthesis.

6. CONCLUSION

We have proposed an iterative solution to the data-driven training of unit boundaries in unit selection TTS utterances. This approach leverages the LSM decomposition of information gathered across each boundary region, which we first introduced in [9]. Compared to standard Fourier analysis, the LSM framework allows all relevant units to be mapped onto the same, separately optimized feature space of relatively low dimension. Iterative training using a global discontinuity criterion then produces the desired adjusted boundaries. This procedure was empirically observed to converge within a relatively small number of iterations. The outcome is a globally optimal cut point between any two matching units in the available unit inventory, which in turn reduces the likelihood of encountering an egregiously bad concatenation between two segments. Experimental evaluations indeed show a consistent decrease in average inter-unit discontinuity due to boundary adjustment, and the trained boundaries seem to correlate well with reduced perceived discontinuity.

7. REFERENCES

[1] A. Hunt and A. Black, "Unit Selection in a Concatenative Speech Synthesis System Using Large Speech Database," in *Proc. ICASSP*, Atlanta, GA, pp. 373–376, 1996.

[2] W.N. Campbell and A. Black, "Prosody and the Selection of Source Units for Concatenative Synthesis," in *Progress Speech Synth.*, J. van Santen, R. Sproat, J. Hirschberg, and J. Olive, Eds., New York: Springer, pp. 279–292, 1997.

[3] M. Balestri *et al.*, "Choose the Best to Modify the Least: A New Generation Concatenative Synthesis System," in *Proc. 6th Eurospeech*, Budapest, Hungary, pp. 2291–2294, September 1999.

[4] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next–Gen TTS System," in *Proc. 137th Meeting Acoust. Soc. Am.*, pp. 18–24, 1999.

[5] F. Malfrere and T. Dutoit, "An Alignment System for Prosodic Parameter Extraction of a French Text," in *Lang. Multim.*, T. Brondsted and I. Lytje, Eds., Aalborg University Press, pp. 139-150, 1997.

[6] Y.-J. Kim and A. Conkie, "Automatic Segmentation Combining an HMM-based Approach and Spectral Boundary Correction," in *Proc. ICSLP*, Denver, CO, pp. 145-148, 2002.

[7] A. Conkie and S. Isard, "Optimal Coupling of Diphones," in *Progress Speech Synth.*, J. Van Santen, R. Sproat, J. Olive, and J. Hirschberg, Eds., New York: Springer, pp. 293–304, 1997.

[8] J. Vepa and S. King, "Join Cost for Unit Selection Speech Synthesis," in *TTS Synth.: New Paradigms & Advances*, S. Narayanan and A. Alwan, Eds., Upper Saddle River, NJ: Prentice Hall, pp. 35–62, 2004.

[9] J.R. Bellegarda, "A Novel Discontinuity Metric for Unit Selection Text–to–Speech Synthesis," in *Proc. 5th ISCA Speech Synth. Workshop*, Pittsburgh, PA, pp. 133–138, June 2004.

[10] J.R. Bellegarda, "Latent Semantic Mapping," *Signal Proc. Magazine, Special Issue Speech Technol. Syst. Human–Machine Communication*, L. Deng, K. Wang, and W. Chou, Eds., Vol. 22, No. 5, pp. 70–80, September 2005.

[11] J.R. Bellegarda, "A Global, Boundary–Centric Framework for Unit Selection Text–to–Speech Synthesis," *IEEE Trans. Speech Audio Proc.*, to appear, Vol. SAP–14, No. 4, July 2006.

[12] D. Talkin, "Voicing Epoch Detection Determination with Dynamic Programming," *J. Acoust. Soc. Am.*, Vol. 85, Supplement 1, 1989.

[13] J.R. Bellegarda *et al.*, "Statistical Prosodic Modeling: From Corpus Design to Parameter Estimation," *IEEE Trans. Speech Audio Proc., Special Issue Speech Synthesis*, N. Campbell, M. Macon, and J. Schroeter, Eds., Vol. SAP–9, No. 1, pp. 52–66, January 2001.

[14] Speech Assessment Methods Phonetic Alphabet (SAMPA), "Standard Machine–Readable Encoding of Phonetic Notation," ESPRIT project 1541, 1987–89, cf. http://www.phon.ucl.ac.uk/home/sampa/home.htm.