Further Developments in LSM–Based Boundary Training for Unit Selection TTS

Jerome R. Bellegarda

Speech & Language Technologies
Apple Computer, Inc., Cupertino, California 95014, USA
jerome@apple.com

Abstract

The level of quality that can be achieved in concatenative text-to-speech synthesis depends, among other things, on a judicious segmentation of all units in the underlying unit selection inventory. We have recently advocated the iterative refinement of unit boundaries based on a data-driven feature extraction framework separately optimized for each boundary region [1]. This paper presents the formal proof of convergence of the iterative algorithm, as well as a detailed analysis of its potential benefits for concatenative TTS synthesis. A formal listening test, in particular, underscores the practical viability of the approach for unit boundary optimization.

Index Terms: speech synthesis, unit selection, segment concatenation, discontinuity perception, boundary optimization.

1. Introduction

In concatenative text-to-speech (TTS) synthesis, 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 [2]. This approach implicitly assumes that all units have been judiciously segmented, because boundary placement critically influences how much discontinuity one is likely to encounter after concatenation, and thus how natural synthetic speech will sound [3].

Automatic segmentation algorithms do not calculate the *globally optimal* cut point between two contiguous units given the entire recorded inventory. Instead, on the basis of general models trained thereon, they seek the best *local* cut point between these two specific units [4]. In many cases, subsequent boundary refinement can make a huge difference in the users' perception of the concatenated acoustic waveform [5]. For highest quality, it would thus be desirable to hand-check every cut point, which is obviously impractical in modern TTS systems. The outcome is often a somewhat uneven performance, where synthetic speech may well sound very good in general but still regularly break down, in ways that are difficult to predict from unit inventory statistics.

We have recently proposed [1] a procedure to systematically optimize all unit boundaries before unit selection, so as to effectively minimize the likelihood of a really bad concatenation. We refer to this (off-line) optimization as the data-driven "training" of the unit inventory, in contrast to the (run time) "decoding" process embedded in unit selection. The method of [1] is based on an alternative TTS feature extraction [6], [7], inspired by the *latent semantic mapping* (LSM) paradigm [8]. This leads to a global discontinuity metric for characterizing the acoustic (dis-)similarity between two candidate segments. Boundary training then leverages this objective function to take into account all potentially relevant units in an iterative manner.

The aim of this paper is to present a formal proof of convergence for the iterative procedure introduced in [1], as well as a detailed analysis of its potential benefits for concatenative TTS synthesis. The next two sections briefly review the underlying LSM framework and the iterative boundary training procedure. Section 4 addresses the convergence of the iterative algorithm, and Section 5 analyzes a representative example, in terms of both inter-unit discontinuity distribution and acoustic waveform differences. Finally, in Section 6 a formal listening test confirms that boundary training can indeed lead to better synthesis.

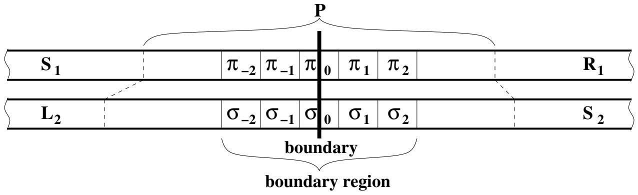


Figure 1: *Speech Segment Notation ($K = 3$).*

2. LSM Framework

To fix ideas, consider among the set of recorded utterances the collection of all possible speech segments ending or starting within the phoneme P, so we can concentrate on a (diphone-style) concatenation within P. Two such acoustic segments, denoted by S_1-R_1 and L_2-S_2, are depicted in Fig. 1. Let $\pi_{-K+1} \dots \pi_0 \dots \pi_{K-1}$ (respectively, $\sigma_{-K+1} \dots \sigma_0 \dots \sigma_{K-1}$) denote the $2K-1$ centered[1] pitch periods associated with the boundary region of S_1-R_1 (respectively, L_2-S_2), such that the boundary between S_1 and R_1 (respectively, L_2 and S_2) falls exactly in the middle of π_0 (respectively, σ_0). For voiced speech segments, each pitch period is defined as the span between two consecutive glottal closure points, and obtained through conventional pitch epoch detection (e.g., [9]). 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 within P. This results in $(2K-1)M$ centered pitch periods in total, encapsulating the entire boundary region. Assuming N denotes the maximum number of samples observed in each of these, we symmetrically zero-pad and appropriately window all instances to N, as necessary. The outcome is the $((2K-1)M \times N)$ matrix W illustrated in the left-hand side of Fig. 2.

At this point we perform the eigenanalysis of W via singular value decomposition (SVD) as [7]:

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

[1] With a *centered* representation, the boundary can be precisely characterized by a single vector in the resulting feature space [7], instead of inferred *a posteriori* from the position of the two vectors on either side.

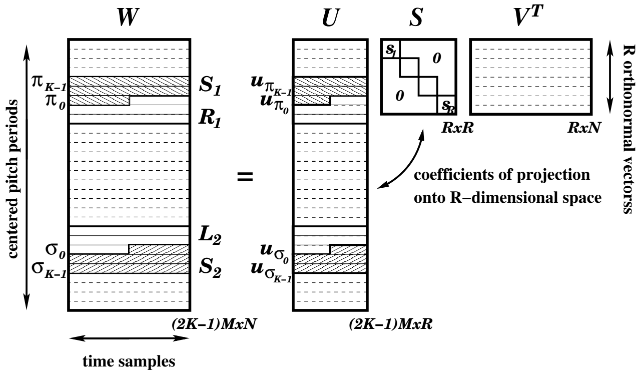


Figure 2: *Decomposition of the Input Matrix.*

where U is the $((2K-1)M \times R)$ left singular matrix with row vectors u_i $(1 \leq i \leq (2K-1)M)$, S is the $(R \times R)$ diagonal matrix of singular values $s_1 \geq s_2 \geq \ldots \geq s_R > 0$, V is the $(N \times R)$ right singular matrix with row vectors v_j $(1 \leq j \leq N)$, $R < \min(N, (2K-1)M)$ is the order of the decomposition, and ^{T} denotes matrix transposition. 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 *LSM space* $\mathcal{L}$ spanned by the (R-dimensional) u_i's and v_j's.

The interpretation of (1) in Fig. 2 focuses on the orthornormal basis obtained from V. Projecting the row vectors of W onto it 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 can then be viewed as feature vectors analogous to, e.g., the usual cepstral vectors.

3. LSM–Based Boundary Training

Consider now the concatenation S_1-S_2, shown as the shaded area in Fig. 2, and denote by δ_0 the concatenated centered period (i.e., consisting of the left half of π_0 and the right half of σ_0). The discontinuity associated with this concatenation is calculated in terms of the trajectory difference before and after concatenation, as expressed in the LSM feature space $\mathcal{L}$.

From [1], the representation of δ_0 in $\mathcal{L}$ is the concatenation vector $\bar{u}_{\delta_0} = \delta_0 V$. Furthermore, the closeness between two individual vectors (cf. [6], [7]) is taken to be:

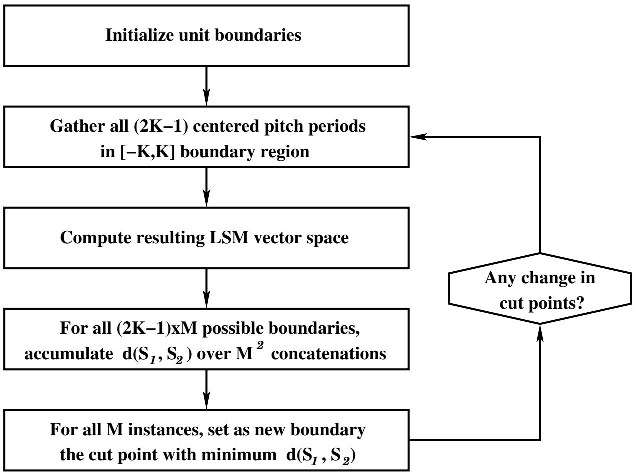
$$c(\bar{u}_k, \bar{u}_\ell) = \cos(u_k S, u_\ell S) = \frac{u_k\, S^2\, u_\ell^T}{\|u_k S\|\,\|u_\ell S\|}\,, \qquad (2)$$

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

$$\tilde{c}(u_{\sigma_{-k}}, u_{\sigma_0}, u_{\sigma_k}) = \frac{c(\bar{u}_{\sigma_{-k}}, \bar{u}_{\sigma_0}) + c(\bar{u}_{\sigma_0}, \bar{u}_{\sigma_k})}{2}\,, \qquad (3)$$

for the average closeness across the boundary σ_0, we therefore specify the *discontinuity score* between S_1 and S_2 as:

$$d(S_1, S_2) = \sum_{k=1}^{K-1} 2\,\tilde{c}(u_{\pi_k}, u_{\delta_0}, u_{\sigma_k}) - \tilde{c}(u_{\pi_k}, u_{\pi_0}, u_{\pi_{-k}}) - \tilde{c}(u_{\sigma_{-k}}, u_{\sigma_0}, u_{\sigma_k})\,. \qquad (4)$$



Initialize unit boundaries

Gather all (2K−1) centered pitch periods in [−K,K] boundary region

Compute resulting LSM vector space

For all (2K−1)xM possible boundaries, accumulate $d(S_1, S_2)$ over M^2 concatenations

For all M instances, set as new boundary the cut point with minimum $d(S_1, S_2)$

Any change in cut points?

Figure 3: *Iterative Training of Unit Boundaries.*

The discontinuity score can be thought of as the relative cumulative change in closeness that occurs during concatenation over the entire boundary region considered. The closer to zero, the more attractive the concatenation. Conversely, the larger the discontinuity score, the more perceptible the concatenation [7].

Once (4) is specified, the iterative boundary training procedure follows the flowchart of Fig. 3. For each phoneme P, M segments from the unit inventory straddle the boundary. These M boundaries must thus be jointly optimized so that all M^2 possible concatenations exhibit minimal discontinuities. 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,[2] but in practice, we have found little difference in behavior based on these various forms of initial conditions [1]. Once this is done, we gather the $2K-1$ centered pitch periods for each unit instance, and derive the resulting LSM space $\mathcal{L}$. This leads to $(2K-1)M$ feature vectors in the space, and hence as many potential new boundaries. For each of them, we compute the associated average discontinuity by accumulating (4) over the set of M^2 possible concatenations. This results in $2K-1$ discontinuity scores for each instance, the minimum value of which yields the cut point to be retained. The new boundaries form the basis for a new boundary region, and the procedure iterates until no change in cut points is necessary.

Since the boundary region shifts from one iteration to the next, the LSM space does not stay static. While this complicates the derivation of a theoretical proof of convergence, it can still be done by exploiting the fact that after each iteration the space remains relatively close to its previous incarnation.

4. Proof of Convergence

First observe that each iteration of training aims at minimizing $d(S_1, S_2)$ for all possible segments S_1 and S_2 within the current boundary region. Thus, from (2)–(4), in order to solve the

[2] For example, the initial boundary for each instance can be placed in the most stable part of the phone (where the speech waveform varies the least), or, more expediently, simply at its midpoint [1].

optimization problem it is sufficient to minimize the Frobenius norm of US^2U^T. Taking into account the fact that U is column-orthogonal, it follows from (1) that the problem is equivalent to minimizing tr(WW^T), where tr$(\cdot)$ denotes matrix trace.

Assume now that at iteration n, tr$(W_n W_n^T)$ is minimized. Due to boundary shifts from iteration n to iteration $n+1$, some pitch periods in W_n are dropped, and replaced by some new ones in W_{n+1}. Denote by Y_n the pitch periods dropped, and by Y_{n+1} the pitch periods added. Exploiting the linearity of the framework and re-arranging the rows yields:

$$W_n = \begin{bmatrix} Y_n \\ Z \end{bmatrix}, \qquad W_{n+1} = \begin{bmatrix} Z \\ Y_{n+1} \end{bmatrix}, \tag{5}$$

where Z represents the (majority of) pitch periods in W_{n+1} that were already present in W_n. Introducing the matrix $\tilde{W}$ as the superset of the pitch periods in W_n and W_{n+1}, we can write:

$$\tilde{W} = \begin{bmatrix} Y_n \\ Z \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} W_n \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} Y_n \\ W_{n+1} \end{bmatrix}. \tag{6}$$

Using the properties of the trace, this in turn entails the three equivalent expressions:

$$\begin{aligned} \text{tr}(\tilde{W}\tilde{W}^T) &= \text{tr}(Y_n Y_n^T) + \text{tr}(ZZ^T) + \text{tr}(Y_{n+1} Y_{n+1}^T)\,, & (7)\\ &= \text{tr}(W_n W_n^T) + \text{tr}(Y_{n+1} Y_{n+1}^T)\,, & (8)\\ &= \text{tr}(Y_n Y_n^T) + \text{tr}(W_{n+1} W_{n+1}^T)\,, & (9) \end{aligned}$$

where all values are non-negative. If the space $\mathcal{L}$ remained perfectly static, then clearly minimizing tr$(W_n W_n^T)$ would also minimize tr$(\tilde{W}\tilde{W}^T)$, otherwise a different segmentation would yield a lower tr$(W_n W_n^T)$ in the first place. But, from (7), this can only happen if:

$$\text{tr}(Y_{n+1} Y_{n+1}^T) \leq \text{tr}(Y_n Y_n^T)\,. \tag{10}$$

Accordingly, we now relax the assumption of perfect stationarity, but constrain any changes in $\mathcal{L}$ to conform to the condition (10). Since, from (8) – (9):

$$\text{tr}(W_{n+1} W_{n+1}^T) \;=\; \text{tr}(W_n W_n^T) + \text{tr}(Y_{n+1} Y_{n+1}^T) - \text{tr}(Y_n Y_n^T)\,, \tag{11}$$

we conclude that, under the constraint (10):

$$\text{tr}(W_{n+1} W_{n+1}^T) \;\leq\; \text{tr}(W_n W_n^T)\,, \tag{12}$$

which completes the proof. The cumulative distance metric (4) is thus guaranteed to converge, in the least-squares sense, to the global minimum tr(ZZ^T), where (at the limit) Z corresponds to the final incarnation of the space $\mathcal{L}$.

The associated final boundaries are therefore globally optimal across the entire set of observations for the phoneme P. Note that, with the choice of the LSM framework, this outcome holds given the exact same discontinuity measure later used in unit selection. Not only does this result in a better usage of the available training data, but it also ensures tightly matched conditions between training and decoding.

5. Experimental Results

We now briefly summarize some of the results we have obtained using male and female voice databases deployed in MacinTalk, Apple's TTS offering on MacOS X. Qualitatively, these databases are fairly similar to the Victoria corpus described in detail in [10]. In particular, recording conditions closely follow those mentioned in [10], though individual utterances generally differ.

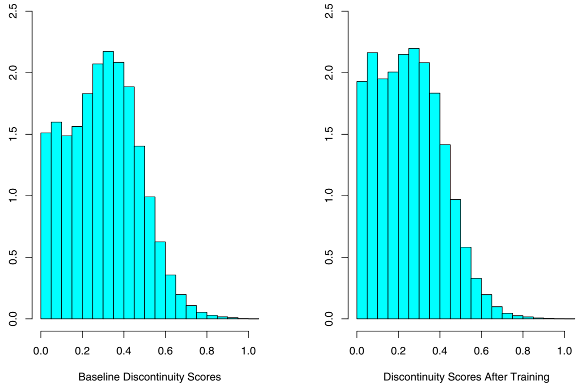


Figure 4: *Distributions in Inter-Unit Discontinuity,* $P = [OI]$.

The phoneme $P = [OI]$ (denoted in SAMPA computer readable phonetic notation, cf. [11]) is especially interesting, because the rapid changes in acoustic targets occurring in the middle of the phoneme tend to complicate the search for an optimal cut point. For this phoneme, boundary training typically yields a consistent reduction of 30% in the average inter-unit discontinuity score across all possible concatenations [1].

Fig. 4 displays the distributions of inter-unit discontinuity scores observed before (left-hand plot) and after (right-hand plot) training the boundaries. Baseline boundaries are (classically) obtained by placing the cut point in the most stable part of the phone, while adjusted boundaries are obtained after, in this case, 16 iterations of boundary training. It can be seen that training shifts the mode of the distribution appreciably to the left: we conjecture that this consistent improvement in all concatenations is largely due to the global scope of the training. Indeed, for $[OI]$ it seems heavily suboptimal to constrain cut points to lie in a (locally) steady state region. Instead, the boundaries are now able to move in an unsupervised manner to attain the relevant global minimum.

To further illustrate the point, the attached files "`Example0_base.aiff`" and "`Example0_new.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 $[OI]$. No signal processing is done beyond the slow down, and, in particular, no manipulation of F_0 is performed. The only difference between the two renditions concerns the boundaries (baseline or adjusted) in the unit inventory.

The analysis of the resulting signals is presented in Figs. 5 and 6. In the first rendition, only slight discontinuities are present in both pitch (blue line) and amplitude (yellow line), indicating that the perception of a bad concatenation comes mostly from the discontinuity related to the particular location of the cut point. In the second rendition, pitch and amplitude have approximately the same profiles as before, again suggesting that they are not significant factors in any perceptual difference between the two renditions. The noticeable improvement that can be heard can therefore be traced directly to a better location for the cut point.

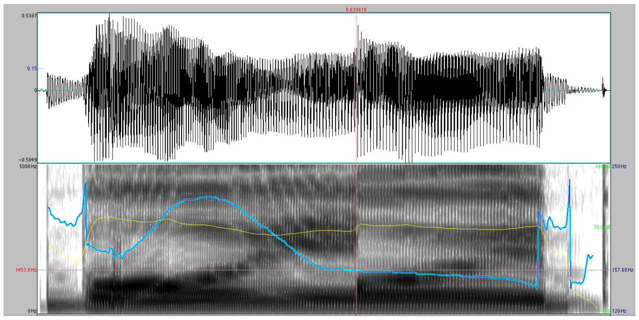


Figure 5: *Analysis of Baseline Rendition of "boyb."*

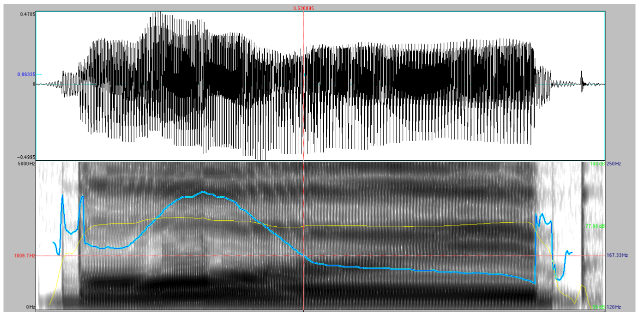


Figure 6: *Analysis of New Rendition of "boyb."*

6. Formal Listening Test

To establish the practical validity of the method, a more formal listening test was performed. As stimuli, we generated a set of whole sentences where the database was segmented entirely using either the baseline or the new method.[3]

Nine participants were selected, including two "naive" users, five generally conversant in speech processing, and two with a more advanced background in psycho-acoustics and phonetics. For each pair of utterances, they were asked to listen sequentially to the two versions, and indicate which version they preferred overall (if any), and why. In each case, the order of presentation was randomized. The results are tabulated in Table I.

Associated files are labelled "`Example$i_base.aiff`" for the baseline and "`Example$i_new.aiff`" for the new boundary training, where $1 \leq i \leq 5$. `Example1` was selected in part because it features a $[bOIb]$ segment similar to the one analyzed above. Close attention to this segment reveals that the same outcome prevails, despite involving a different database recorded by a different voice talent of a different gender. Participants also noted some noticeable improvement in the vicinity of "the purple."

The other examples feature increasingly longer sentences, and, not suprisingly, differences between the two approaches appear to be more pronounced over some segments than others. Segments most often singled out by participants include: in `Example2`, "the cow" and "right away;" in `Example3`, "years ago" and "toy around;" in `Example4`, "expected" and "negatively;" and in `Example5`, "a writer," "insisted in court," and "specific echoes."

Table I shows that, on average, the sentences synthesized from the database featuring the optimal cut points were preferred over three times more often than those synthesized from the database with the baseline cut points. Furthermore, the "Prefer New" outcome is substantially more likely than the combination of "Prefer Base" and "Prefer None" outcomes. We infer that the globally optimal approach described in this paper resulted in boundaries with a smaller amount of perceivable audible discontinuity.

Table I: Listener Preference Results. Maximum Score Achievable is 9.

Utterance	Prefer Base	Prefer None	Prefer New
Example1	2	2	5
Example2	2	4	3
Example3	1	0	8
Example4	4	2	3
Example5	0	0	9
Average Score	1.8	1.6	5.6
95% Confidence	±1.2	±1.3	±2.2

7. Conclusion

We have derived a formal proof of convergence for the iterative boundary training procedure introduced in [1], and analyzed typical distributions in inter-unit discontinuity scores observed before and after training. In addition, a formal listening test has confirmed that utterances synthesized with adjusted boundaries tend to comprise less egregious discontinuities than those synthesized with baseline boundaries. This illustrates the compelling potential of the approach for concatenative TTS synthesis. Future efforts will concentrate on more systematically exploring the influence of the decomposition parameters (particularly K and R), in order to better characterize their relationship to factors such as phoneme identity, number of observations, dominant style of elocution, and overall prosodic context distribution.

8. References

[1] J.R. Bellegarda, "LSM–Based Boundary Training for Concatenative Speech Synthesis," in *Proc. ICASSP*, Toulouse, France, May 2006.

[2] 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.

[3] R. Sproat, Ed., *Multilingual Text–to–Speech Synthesis: The Bell Labs Approach*, Boston, MA: Kluwer Academic Publishers, p. 199, 1997.

[4] F. Malfrere *et al.*, "Phonetic Alignment: Speech–Synthesis–Based Versus Viterbi–Based," *Speech Communication*, Vol. 40, No. 4, pp. 503–517, 2003.

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

[6] 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.

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

[8] 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.

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

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

[11] 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.

[3] Here signal processing was limited to some elementary post-selection diphone blending, which was performed with a very short window (30 samples) for the specific purpose of avoiding particularly egregious (and distracting) "glugs." As before, there was no manipulation of F_0 at all.