ISCA Archive
http://www.isca-speech.org/archive

The Third ESCA/COCOSDA Workshop
(ETRW) on Speech Synthesis
Jenolan Caves House, Blue Mountains,
NSW, Australia, November 26-29, 1998

GENERALIZATION AND DISCRIMINATION IN TREE-STRUCTURED UNIT SELECTION

Michael W. Macon *Andrew E. Cronk* *Johan Wouters*[*]

Center for Spoken Language Understanding
Oregon Graduate Institute, Portland OR 97291-1000
http://www.cse.ogi.edu/CSLU/tts

ABSTRACT

Concatenative "selection-based" synthesis from large databases has emerged as a viable framework for TTS waveform generation. Unit selection algorithms attempt to predict the appropriateness of a particular database speech segment using only linguistic features output by text analysis and prosody prediction components of a synthesizer. All of these algorithms have in common a training or "learning" phase in which parameters are trained to select appropriate waveform segments for a given feature vector input. One approach to this step is to partition available data into clusters that can be indexed by linguistic features available at runtime. This method relies critically on two important principles: *discrimination* of fine phonetic details using a perceptually-motivated distance measure in training and *generalization* to unseen cases in selection. In this paper, we describe efforts to systematically investigate and improve these parts of the process.

1 INTRODUCTION

Since the late 1980's [1], "selection-based" concatenative synthesis from large databases has received increased interest as a potential improvement upon fixed diphone (or demiphone, etc.) inventories. In principle, a greater number of units should be able to more accurately realize several types of variability in the output (e.g., vowel reduction, end-of-phrase voice quality changes), and for very large databases even appropriate F_0 contours and segmental durations. In practice, these large-corpus methods tend to result in output speech quality that can be very good or very bad depending on the similarity of the input text to items in the synthesizer database. Thus, many unsolved problems exist and further research effort is needed.

All unit selection algorithms have in common a training phase in which parameters are trained to provide selection of appropriate waveform segments. These methods rely critically on two important principles: (*i*) discrimination of fine phonetic details using a perceptually-motivated distance measure in training and (*ii*) generalization to unseen cases in runtime selection. In this paper, we discuss several issues related to design and optimization of a tree-based clustering algorithm for unit selection. Section 2 reviews current approaches to unit selection including the tree-based method and discusses their mechanisms for learning, generalization, and discrimination. In Section 3, we discuss a perceptual experiment designed to test the correlation of several well-known distance measures to human judgements of similarity between speech segments. In Section 4, we describe a method for creating selection trees and show that it can be improved by incorporating a measure of its ability to generalize in the training process.

2 UNIT SELECTION APPROACHES

All speech synthesis algorithms attempt to generate acoustic features of speech using as input only the linguistic target features produced by the text/prosodic analysis components of the synthesizer. Unit selection algorithms do this by attempting to predict the appropriateness of a particular database speech segment using only these linguistic targets. A "unit" can be any quantal size from a word to phone to sub-phone. Features can include categorical or numeric values like place/manner of articulation of N units to the left and right, stress, F_0 targets, duration, etc. Denoting the linguistic features of a target and database candidate unit $\mathbf{l^t}$ and $\mathbf{l^c}$, respectively, and the acoustic features of a target and database candidate unit $\mathbf{a^t}$ and $\mathbf{a^c}$, respectively, the the task in synthesis of an utterance is the following:

> Given a sequence of linguistic target features $\{\mathbf{l^t}_0, \mathbf{l^t}_1, ..., \mathbf{l^t}_N\}$, find the sequence of candidate segments with linguistic features $\{\mathbf{l^c}_0, \mathbf{l^c}_1, ..., \mathbf{l^c}_N\}$, that will minimize an acous-

[*] This work was supported by grants from Texas Instruments and Intel Corporation.

tic feature distance

$$d(\{\mathbf{a^t}_0, \mathbf{a^t}_1, ..., \mathbf{a^t}_N\}, \{\mathbf{a^c}_0, \mathbf{a^c}_1, ..., \mathbf{a^c}_N\}).$$

The acoustic target features $\mathbf{a^t}$ of the speech segments to be synthesized are not known at run-time; these must be predicted automatically in some way after training on a database. Since it is impossible to create a speech database that contains all possible combinations of linguistic feature contexts, some of the inputs $\mathbf{l^t}$ will not have been seen previously in the database, thus the algorithm must generalize to find outputs for these cases. The function $d(\cdot,\cdot)$ can take into account both "target cost" between target $\mathbf{a^t}_i$ and candidate $\mathbf{a^c}_i$ and "concatenation cost" between selected units $\mathbf{a^c}_i$ and $\mathbf{a^c}_{i+1}$ [2].

In the discussions that follow, we focus mainly on target cost. That is, given an input linguistic feature vector $\mathbf{l^t}$, find the database candidate with linguistic features $\mathbf{l^c}$ that should minimize $d(\mathbf{a^t}, \mathbf{a^c})$, even though we don't know $\mathbf{a^t}$ explicitly.

2.1 Training a distance function

The work of Hunt, Black, Campbell, *et al.* in the CHATR system [2, 3] uses a linear regression technique to train a function

$$D(\mathbf{l^t}, \mathbf{l^c}) = \sum_i w_i \left| f(l_i^t) - f(l_i^c) \right|^2, \qquad (1)$$

where $f(\cdot)$ is a mapping from categorical linguistic features like "place of articulation" to numeric values. This mapping can also be structured as a table lookup $f(l_i^t, l_i^c)$. The weights w_i are trained to minimize the average difference between $D(\mathbf{l^t}, \mathbf{l^c})$ and $d(\mathbf{a^t}, \mathbf{a^c})$, the true acoustic distance, over the entire database. Selection at runtime is based on choosing the N database candidates with smallest cost $D(\mathbf{l^t}, \mathbf{l^c})$ for each target unit, then considering concatenation cost between them. A dynamic programming search is used to jointly minimize these costs.

Ultimately, the function $D(\mathbf{l^t}, \mathbf{l^c})$ should predict human judgements of similarity between units $\mathbf{l^t}$ and $\mathbf{l^c}$. Whether or not this is true depends on two fundamental assumptions. First, the acoustic distance measure $d(\mathbf{a^t}, \mathbf{a^c})$ used to train $D()$ must mimic human judgements of difference between segments $\mathbf{l^t}$ and $\mathbf{l^c}$. In [2], the root-mean-squared (RMS) mel-cepstral distance over the segments was used. Second, the algorithm must be able to generalize to accurately predict $D(\mathbf{l^t}, \mathbf{l^c})$ for unseen $\mathbf{l^t}$ and select an appropriate unit in these cases. Gradual deviations of $\mathbf{l^t}$ away from a particular $\mathbf{l^c}$ should correspond to gradual increases in $D(\mathbf{l^t}, \mathbf{l^c})$ in a way that also mimics ordering of human judgements of phonetic quality difference. Because the linguistic features are for the most part categorical, the mapping function $f()$ plays a critical role in determining whether or not this is true.

2.2 Clustering

Another approach to the training step is to partition the candidates (each described by linguistic feature vector $\mathbf{l^c}_i$ and acoustic feature vector $\mathbf{a^c}_i$) into clusters containing "similar" units, using a distance measure $d(\mathbf{a^c}_i, \mathbf{a^c}_j)$. Unit selection at run-time corresponds to indexing into these clusters using target features $\mathbf{l^c}$ and choosing one member of the best cluster for each target.

A binary decision tree is a useful and computationally-efficient mechanism for performing this clustering and indexing. Most approaches to binary tree clustering draw on elements of the Classification and Regression Trees (CART) methodology proposed by Breiman, *et al.* [4]. In the training process, for a particular node in the tree, all binary splits along dimensions of the linguistic feature vector $\mathbf{l^c}$ are considered. The splitting question that results in the most "compact" child clusters (i.e., those having the minimum entropy or variance) is kept at each stage, and the process is repeated.

In work by Black and Taylor [5] and others [6, 7], and in the system described in Section 4, a weighted average of a distance measure $d(\mathbf{a^c}_i, \mathbf{a^c}_j)$ over all pairs of units in the cluster is used as a measure of intra-cluster variance. The power of the algorithm to find judicious splits of the data and provide appropriate synthesis units is based directly on the ability of the distance measure $d()$ to mimic human judgments of acoustic similarity.

In work by Donovan [8], the measure of cluster variance is based on log-likelihood of the data when a Gaussian distribution is fit to the candidates in the child clusters. This is similar to using a distance weighted by the inverse covariance matrix of the cluster (i.e., Mahalanobis distance). Here it is assumed that a squared difference of the feature vectors (e.g., MFCC's) will mimic human judgements of acoustic difference. A similar phonetic decision tree approach has been followed by others in both speech recognition [9] and synthesis [10].

In the clustering approach, an input vector not seen in the training data will still move down the tree until it resides in some terminal node. By virtue of the tree-growing procedure, a fundamental assumption is that the candidate units residing in this terminal node will be acoustically similar to the hypothetical target acoustic target vector $\mathbf{a^t}$. Thus the generalization power of the algorithm depends on when the partitioning of the data is stopped. If tree growth is not stopped soon enough, the tree will become biased towards the units in the training data, and too many good matches to a previously-unseen input will be excluded from consideration. If it is stopped too early, too many poor units will be considered. Most published approaches to binary tree clustering have used stopping thresholds (e.g., stop splitting when the relative improvement in variance is less than $x\%$) to halt tree growth. In Section 4, we discuss the merits of using cross-validation during tree-growing to optimize the tree's power to generalize to unseen cases.

2.3 Training a generation function

The methods described above generate acoustic features of speech using only the linguistic target features of the synthesizer, but they do so in a very indirect manner. It is also possible to structure the problem as a *direct* mapping

$$\mathbf{a^t} = g(\mathbf{l^t}). \tag{2}$$

Rule-based formant synthesizers accomplish the linguistic-acoustic mapping $\mathbf{a^t} = g(\mathbf{l^t})$ by using hand-coded tables of formant targets and smoothing functions. Articulatory synthesizers do the same to generate articulator positions, and add the extra step of computing the output of a vocal tract physical model. In these cases, the distance measure $d()$ enters the problem through the metric used by the developer tweaking the rules – i.e., listening to the output, which *really does* mimic human judgements. Generalization capability is controlled by the "correctness" and comprehensiveness of the smoothing rules to behave appropriately for cases the developer has not checked in the development cycle.

Data-driven approaches to a direct mapping have also been proposed, often based on neural networks (e.g., [11, 12]). In this case, the neural network learns a correspondence between input vectors $\mathbf{l^t}$ and feature targets $\mathbf{a^t}$, as well as implicit context-dependent smoothing functions realized by including time-delayed feedback terms in the network. Phonetic distance measures enter the training algorithms through the network weight update measures – a perceptually well-motivated measure will optimize the weights most effectively. The advantage of this approach is a much increased power to generalize in comparison to waveform-based unit selection. The direct mapping can, to a limited degree, create new units for contexts that were not recorded in the database at all, whereas in unit selection this is only accomplished by assuming that a unit from a different context can be substituted in its place.

The drawback is of course, that it is difficult to learn a direct mapping, and this means that a relatively simplified speech model must be used – e.g., a Klatt formant synthesizer, pulse-excited LPC or cepstral coefficients. Many difficult-to-model details of the signal captured in waveform concatenative synthesis are thrown away to create a simplified model with few enough parameters to be reliably estimated.

3 DISCRIMINATION EXPERIMENT

The preceding section has established the fact that a critical component of all unit selection algorithms is a measure of distance between speech segments. This measure must be designed to correspond to human judgements of similarity/difference.

Furthermore, the requirements on distance measures for synthesis are somewhat more stringent than in automatic speech recognition (ASR) algorithms. In ASR, the task at hand is to statistically discriminate *phones* in order to decode the underlying phoneme sequence. In TTS, on the other hand, the task is to choose the most perceptually appropriate item from a set of similar *allophones* – a much more difficult task.

We have examined the discriminative power of several well-known distance metrics through a test of perceptual judgements of allophonic variations. Others have investigated distance measures in the context of speech coder evaluation (e.g., [13]), improving ASR performance [14, 15], and in more general studies of distance perception [16]. Some recent work studied the use of an auditory model to predict concatenation discontinuities in sine-wave artificial formants [17]. In contrast, the test to be described here is based on substituting segments of speech in an otherwise normal utterance (it tests "target cost"), using conditions very similar to those found in a selection-based synthesizer.

3.1 Perceptual test data

The test database consisted of 166 pairs of CVC words, each pair containing a reference word and a modified version of the same word. The reference words were created with a diphone synthesizer [18], with care taken that there were no noticeable spectral discontinuities at

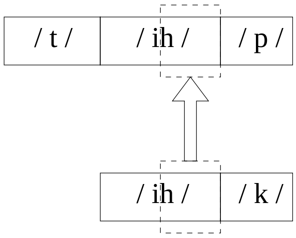

Figure 1. Illustration of segment substitution in modified version of word "tip" (Category III).

the join point in the vowel. The modification consisted of replacing half of the vowel by a instance of the same vowel taken from a different context, as illustrated in Figure 1. To make the experiment a manageable size, the reference words were limited to three categories:

Category I : consonant–vowel–nasal (/n,m,ng/)

Category II : glide (/w,r,l,y/)–vowel–consonant

Category III : cons–vowel–voiceless stop (/p,t,k/)

All were actual English words. In Category I and III, the second half of the vowel was substituted; in Category II, the first half of the vowel was substituted. In 38 of the pairs, the two words were identical, and these were used as a control group. Listeners were presented with each pair and asked to rate their distance on a scale from 0 (identical) to 4 (very different). Fifteen subjects participated in the test (one subject's responses were rejected because they fell significantly outside the distribution of the other fourteen).

3.2 Objective measure data

We considered the correlation of the perceptual data to five different feature extraction methods: LPC- and FFT-based cepstral coefficients, line spectral frequencies, log area ratios, and the Itakura distance (see [19] for descriptions). These were coupled with two different spectral pre-warping schemes: (*i*) mel-scale frequency warping and (*ii*) a frequency-dependent loudness scaling and Bark-scale frequency warping used in the perceptual linear prediction (PLP) method [20]. These features (all LPC-based measures used 12 dimensions) were extracted from the speech segments at regular intervals. Features from the shorter of two segments of different length were linearly warped to the longer duration The objective measure of distance used was the average squared difference of the time-aligned feature vectors.

	linear	PLP	Mel
FFT cepstra	0.49	0.62	**0.64**
LPC cepstra	0.48	0.61	**0.64**
line spectral frequencies	0.34	0.57	0.58
log area ratio	0.28	0.55	0.52
Itakura distance	0.50	0.61	**0.64**

Table 1. Correlation between perceptual distances and several objective measures. Boldface numbers are the best results.

3.3 Results

Table 1 shows the weighted average of correlation coefficients computed in each of the phonetic categories above (using a Fischer transformation [21]). The best correlations were found using a mel-scale cepstral representation (MFCC), however PLP-based measures were also within the margin of error of the test (±0.05). When delta (slope) features were added to the representation, increases to a correlation of 0.68 were found for the best case. This is in the neighborhood of the best correlations found in speech coding tests described in [13]. For more detailed analysis of the results, please see [22].

Although "higher is better," it is difficult to discern whether a correlation of 0.68 is good enough for unit selection. An interesting alternative measure is to set a threshold in the perceptual responses (0,1='good' and 2,3,4='bad'), and investigate the *detection versus false alarm characteristic* of the MFCC measure. That is, if we set an MFCC distance threshold, what percentage of 'good' units will be classified as 'bad' and vice-versa? For unit selection, this describes the probability of not selecting good units versus choosing inappropriate units.

The plots of this characteristic for each test category (I, II, III) and for four different vowels (/ae,aa,iy,uw/) in the CVC exemplars are given in Figure 2. A curve that is squeezed into the upper left corner of the plot is near-optimal; a diagonal line with a slope of 1.0 shows that the measure gives no information. For example, for the vowel /iy/, the measure seems to perform quite poorly. For the 'glide-/ae/-C' exemplars, the performance is better. Because these results are based on relatively few data points, we feel that further experiments are needed to draw quantitative conclusions. However, these preliminary results suggest that simple, well-known measures like the ones considered in this study are *not sufficiently reliable* to guarantee optimal results in unit selection. Development of measures

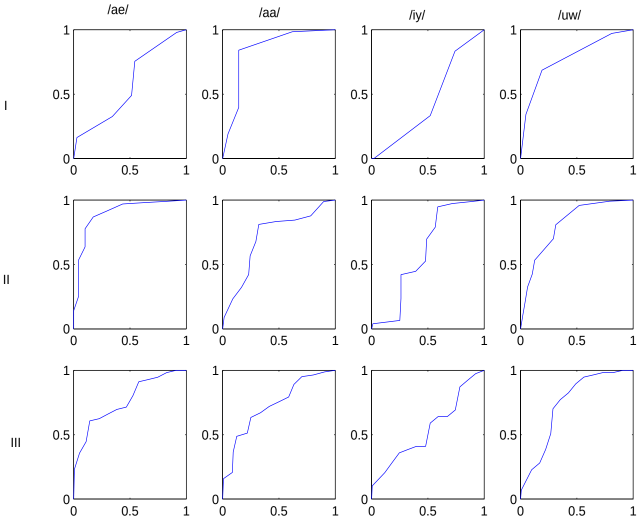

Figure 2. *Probability of detection* versus *probability of false alarm* characteristics for segment classification into ’good’ or ’bad’ perceptual categories. Each point on the curve describes the probability of selecting good units versus the probability of choosing inappropriate units.

better suited to allophonic discrimination is needed.

4 GENERALIZATION EXPERIMENT

In Section 2, we argued that all synthesis algorithms, including unit selection methods, must show some power to generalize to unseen cases. In the clustering approach to unit selection, this is accomplished by stopping the growth of the tree before it becomes too biased toward the training data. In most previous approaches, the growth of binary trees is halted through the use of a stopping threshold. Manual setting of thresholds is tedious, and provides no criterion of optimality in generalizing to unseen inputs.

In this section, we consider a cross-validation method for optimizing tree size automatically based on the tree's role in the unit selection process. The method is reminiscent of the cross-validation technique used for classification in the CART method [4], with some modifications.

In this approach, a “development set” is held out from the database during tree-growing (clustering). Instead of using a fixed stopping criterion, the tree is first grown to its maximum size. At this point, a recombination process is begun by considering the effect of reducing the size of the tree on the average distance from the development set. This is done by dropping into the tree the linguistic description vector $\mathbf{l^c}$ of each unit in the development set, and computing the average distance of the selected cluster to the acoustic features $\mathbf{a^c}$ for each unit. At each stage, the terminal nodes that, when combined upward worsen the training error the least are pruned. In this process, we are guiding the recombination by measuring the power of the tree to select appropriate units for the unseen cases. The recombination is continued until the tree consists of only a single node. This produces a curve similar to Figure 3, from which a minimum can be found. This minimum corresponds to the tree that is “optimally pruned” with respect to the development set. The process can be repeated after shuffling the development/training set division.

Using a single-speaker database of phonetically-balanced sentences, the performance of the cross-validation method was compared with the performance of two common stopping criteria: (a) a *minimum units* threshold, which sets a lower limit for the number of units in a node and stops splitting at that point, and (b) a *minimum improvement* threshold which stops splitting when measures of intra-cluster variance fail to improve more than a specified percentage.

In these tests, the cross-validation method acheived approximately 15% better objective distortion scores than threshold methods, while requiring no hand-tuning. More details of the results of this experiment can be found in [23]. In related experiments for ASR using log likelihood as the distance metric [9], cross-validation performed about the same as a threshold method, but was preferred for the fact that it was automatic.

Of course, the real question remaining to be answered is whether or not the objective distance measure improvements seen in this experiment will translate into clear improvements in the subjective quality of the speech produced.

5 DISCUSSION

This paper has reviewed several recent approaches to unit selection-based concatenative synthesis, and pointed out the need to consider both measures of phonetic discrimination and generalization in their design.

Future efforts in this area should be directed in several areas, including more detailed perceptual studies applicable to synthesis, specifically tests of concatenation detection in various environments. Better measures need to be devised, and these will probably need to incorporate time- and frequency-domain masking principles [24]. More detailed study of invariances in

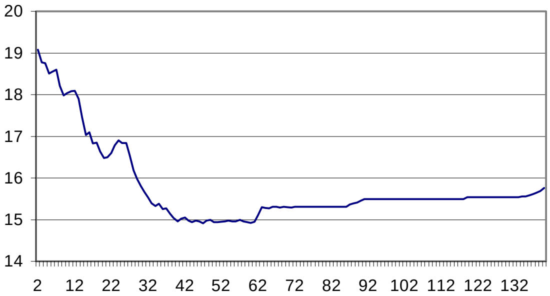

Figure 3. Average distance of units in selected cluster to units in the development set during cross validation, as a function of the number of leaves in the tree. The tree is maximally grown, then pruned back (right to left across the x-axis). (Lower is better on the y-axis.)

speech data should be considered, and "feedback" of these results into database design must happen to maximize coverage of important variability. Finally, further attention should be paid to using signal models that can modify more than F_0 and duration (perhaps phonetic reduction, voice quality changes, and other effects). In this way, efforts can be made to combine the advantages of parametric models and waveform-based unit selection.

REFERENCES

[1] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, pp. 679–682, 1988.

[2] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 373–376, 1996.

[3] A. Black and W. N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," in *Proc. EUROSPEECH*, pp. 581–584, September 1995.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Monterey, CA: Wadsworth and Brooks, 1984.

[5] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proc. EUROSPEECH*, 1997.

[6] W. J. Wang, W. N. Campbell, N. Iwahashi, and Y. Sagisaka, "Tree-based unit selection for English speech synthesis," in *Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. II, pp. 191–194, 1993.

[7] S. Nakajima, "Automatic synthesis unit generation for English speech synthesis based on multi-layered context oriented clustering," *Speech Communication*, vol. 14, pp. 313–324, September 1994.

[8] R. E. Donovan, *Trainable Speech Synthesis.* PhD thesis, Cambridge University, 1996.

[9] H. J. Nock, M. J. F. Gales, and S. J. Young, "A comparitive study of methods for phonetic decision-tree state clustering," in *Proc. EUROSPEECH*, vol. 1, pp. 111–114, 1997.

[10] X. D. Huang, A. Acero, *et al.*, "Whistler: A trainable text-to-speech system," in *Proc. of the Int'l Conf. on Spoken Language Processing*, pp. 2387–2390, 1996.

[11] O. Karaali, G. Corrigan, and I. Gerson, "Speech synthesis with neural networks," in *Proc. of World Congress on Neural Networks*, pp. 45–50, September 1996.

[12] C. Tuerk and T. Robinson, "Speech synthesis using artificial neural networks trained on cepstral coefficients," in *Proc. EUROSPEECH*, pp. 1713–1716, 1993.

[13] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, *Objective Measures of Speech Quality.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[14] N. Nocerino, F. K. Soong, L. R. Rabiner, and D. H. Klatt, "Comparative study of several distortion measures for speech recognition," *Speech Communication*, vol. 4, pp. 317–331, 1985.

[15] H. Hermansky and J. C. Junqua, "Optimization of perceptually-based ASR front-end," in *Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, pp. 219–222, 1988.

[16] O. Ghitza and M. M. Sondhi, "On the perceptual distance between two speech segments," *Journal of the Acoustical Society of America*, vol. 101, no. 1, pp. 522–529 1997.

[17] J. H. L. Hansen and D. T. Chappell, "An auditory-based distortion measure with application to concatenative speech synthesis," *IEEE Trans. on Speech and Audio Processing*, vol. 6, pp. 489–495, September 1998.

[18] M. Macon, A. Cronk, J. Wouters, and A. Kain, "`OGIresLPC`: Diphone synthesiser using residual-excited linear prediction," Tech. Rep. CSE-97-007, Department of Computer Science, Oregon Graduate Institute of Science and Technology, Portland, OR, September 1997. available from `www.cse.ogi.edu/CSLU/tts`.

[19] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals.* Macmillan, 1993.

[20] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Acoustics*, vol. 2, pp. 587–589, October 1994.

[21] A. L. Edwards, *An Introduction to Linear Regression and Correlation.* San Francisco: W. H. Freeman and Co., 1993.

[22] J. Wouters and M. W. Macon, "Phonetic distance measures for speech synthesis." to be presented at the *International Conference on Spoken Language Processing*, December 1998.

[23] A. E. Cronk and M. W. Macon, "Optimized stopping criteria for tree-based unit selection in concatenative synthesis." to be presented at the *International Conference on Spoken Language Processing*, December 1998.

[24] B. C. J. Moore, *An Introduction to the Psychology of Hearing.* Academic Press Limited, 3rd ed., 1989.