FURTHER ANALYSIS OF LSM–BASED UNIT PRUNING FOR UNIT SELECTION TTS

Jerome R. Bellegarda

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

ABSTRACT

The level of quality that can be achieved in concatenative text-to-speech synthesis is primarily governed by the inventory of units used in unit selection. This has led to the collection of ever larger corpora in the quest for ever more natural synthetic speech. As operational considerations limit the size of the unit inventory, however, *pruning* is critical to removing any instances that prove either spurious or superfluous. At last ICASSP we introduced an alternative pruning strategy based on a data-driven feature extraction framework separately optimized for each unit type in the inventory [1]. This paper presents further validation of this strategy, as well as a detailed analysis of its potential benefits for concatenative synthesis.

Index Terms— Concatenative speech synthesis, unit selection, inventory pruning, distinctiveness/redundancy perception.

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 the underlying speech database contains enough distinct segments, with sufficiently varied phonetic and prosodic characteristics, to cover all acoustico-linguistic events to be synthesized. Not surprisingly, this has led to an exponential growth in the size of the average concatenative TTS database. Unit inventories with a footprint close to 1 GB are now routine in server-based applications (cf. [3]). The next generation of unit selection systems could easily bring forth another order of magnitude increase in this footprint. Nevertheless, operational considerations will always limit inventory size to a finite practical value, and thus the level of coverage associated with a given database will always be less than 100% [4].

Hence the need to *prune* the unit inventory, i.e., to decide which units are best kept and which are best discarded, so as to attain the highest possible coverage for a given overall target size. Pruning is not only an engineering requirement for most platforms and systems, but also a critical element of investigating the degrees of freedom within a TTS database. As such, it contributes to our fundamental understanding of concatenative synthesis. The tantalizing pay-off, of course, is that one day it might become unnecessary to record data that would be pruned out anyway.

Pruning is usually based on clustering together units that are "similar," comparing units from each cluster to the relevant cluster center, and removing those instances that are "furthest away" from the cluster center.[1] Pruning 20% of units in this way usually makes no significant difference to (and may even improve) perception, while up to 50% may be removed without seriously degrading quality [5]. The exact outcome, however, tends to be markedly sensitive to the particular distance measure adopted for calculating the impurity of a cluster (as well as, if applicable, to the particular corpus chosen for establishing the relative frequency of units). The selected metrics are usually local in nature, which typically results in sub-optimal (greedy) clustering [5]. Also, in some cases, looking at the distribution of the distances within clusters to quantify what is meant by "close enough" can be a fairly opaque process (cf. [1]). This underscores a certain lack of scalability, and the need for at least some human supervision.

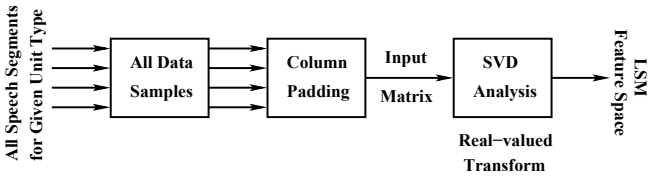

Fig. 1. *Pruning-Specific LSM Feature Extraction.*

At last ICASSP we introduced [1] a pruning approach based on a different signal representation. This solution relies on an alternative TTS feature extraction framework [6], inspired by the *latent semantic mapping* (LSM) paradigm [7]. This leads to a consistent distinctiveness/redundancy measure which can address, in a scalable manner, the (traditionally separate) problems of outliers and redundant units. The aim of this paper is to further validate this solution, and to more fully characterize its behavior and ensuing benefits for concatenative TTS synthesis. The next section briefly reviews the LSM-based unit pruning framework and associated distinctiveness/redundancy measure. Section 3 focuses on a simple case study which exposes in detail the kind of behavior typical of the approach. Finally, in Section 4 a more formal listening test suggests that LSM-based unit pruning can indeed be performed without noticeable degradation in perceived quality.

2. LSM–BASED UNIT PRUNING

Pruning-oriented LSM-based feature extraction is illustrated in Fig. 1, where a *unit type* is any acoustico-linguistic event of interest (be it an individual demi-phone, phoneme, diphone, syllable, word, or sequence thereof, possibly in a specific acoustic and/or prosodic context), and a *unit* is an individual observation, or instance, of that unit type in the unit selection inventory. Assume that for a given unit type, M instances are available. The first step is to gather the time-domain samples associated with each of these M instances. If N denotes the maximum number of samples observed over this collection, we then zero-pad all units to N, as necessary.[2] The outcome is a $(M \times N)$ matrix W with elements w_{ij}, where each row w_i corresponds to a particular unit, and each column t_j corresponds to a slice

[1] The reader is referred to [1] for a review of the various ways to implement this strategy, as well as a discussion of their respective shortcomings.

[2] Among the several length normalization methods we have looked into, column padding seems to work the best, perhaps because it more directly preserves duration information.

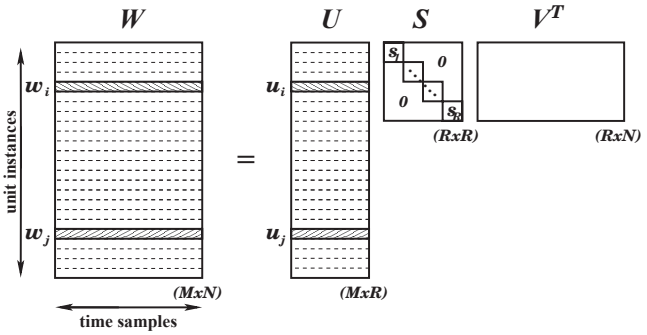

Fig. 2. *Decomposition of the Input Matrix.*

of time samples. This matrix W, illustrated in the left-hand side of Fig. 2, globally encapsulates the unit type, as characterized by *all* of its instances in the database. Typically, M and N are on the order of a few thousands to a few tens of thousands.

At this point we perform the eigenanalysis of W via singular value decomposition (SVD) as [6]:

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

where U is the $(M \times R)$ left singular matrix with row vectors u_i $(1 \leq i \leq M)$, 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 $(1 \leq j \leq N)$, $R \leq \min(M, N)$ 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* 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 that basis defines a representation for the units in terms of their coordinates in this projection, namely the rows of US. Thus, (1) defines a mapping between the set of units and (after appropriate scaling by the singular values) the set of R-dimensional vectors $\bar{u}_i = u_i S$. These can then be viewed as feature vectors analogous to, e.g., the usual cepstral vectors, except that they are obtained through a global, unit-specific, real-valued decomposition instead of a local, signal-independent projection onto a set of complex sinusoids [6].

Given this feature extraction, a natural expression for the closeness between two feature vectors is given by [1], [6]:

$$c(\bar{u}_i, \bar{u}_j) = \cos(u_i S, u_j S) = \frac{u_i\,S^{\,2}\,u_j^{\,T}}{\|u_i S\|\,\|u_j S\|}\,, \tag{2}$$

for any $1 \leq i, j \leq M$. This is the *distinctiveness/redundancy measure* induced over the LSM feature space.

Following [1], the measure (2) allows us to cluster the feature vectors into K clusters, where $K \ll M$. Once these K clusters have been obtained in the feature space, we proceed to eliminate all clusters with n or less vectors, which are most likely to be associated with outlier units. The remaining clusters, by construction, comprise vectors which are very close to one another in the space, and which are therefore good candidates for interchangeability. It is thus safe to replace them by their centroid, or, in practice, the actual unit which maps closest to that centroid in the LSM feature space. All other instances of that unit type in the same cluster can therefore be pruned away. The procedure iterates on the set of unit types until all of them have been processed. The collection of retained vectors then constitutes the pruned unit inventory.

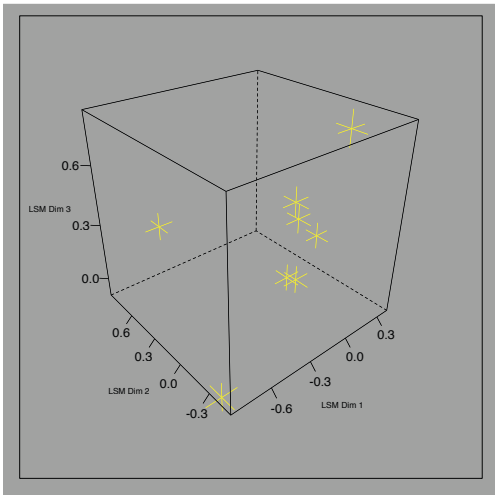

Fig. 3. $M = 8$ *Instances of* $w =$ see *(in 3-D Space).*

3. TYPICAL BEHAVIOR

At first glance such an unconventional approach, based on a simple eigenvalue analysis in the time domain, may not appear ideally suited for the pruning task at hand. After all, it is well known that two speech segments may have similar waveform shapes and still sound quite a bit different, or conversely they may look somewhat unrelated while sounding perceptually close to each other.

To investigate this concern, it seems best to delve in detail into an illustrative case study. We started from a phonetically and prosodically varied voice database currently deployed in MacinTalk, Apple's TTS offering on MacOS X.[3] We then restricted the corpus to a small subset of the database with no obvious outliers. To keep the amount of data to analyze tractably small, and thus allow for close examination of every individual unit, we further focused on only $M = 8$ instances of the word $w =$ see, selected in a semi-supervised way to achieve suitable coverage of durational behavior.

We extracted these units from about 40 "see" segments present in the subset considered. Across the $M = 8$ instances, we observed a maximum number of samples of $N = 10721$, which led to a (8×10721) input matrix. We then computed the SVD of this matrix and obtained the associated feature vectors as described in Section 2. For display purposes, we selected $R = 3$ for the dimension of the LSM space, but in this case values of R in the range $[3, 8]$ all produced a qualitatively similar outcome. A rendition of the ensuing 3-D representation is given in Fig. 3, showing the 8 feature vectors resulting from the mapping.

At this point we clustered these feature vectors via bottom-up clustering using the distinctiveness/redundancy measure (2). In this simple case, the most natural outcome was 3 distinct clusters, for a reduction factor of 2.67. The first cluster regroups the 3 points with positive coordinates (closest to the center of the cube). The second cluster regroups the two points lying near the origin (closest to the bottom of the cube). The third cluster regroups the remaining

[3]Though individual utterances generally differ, the underlying corpus, called Alex, is fairly similar to the Victoria corpus described in detail in [8], especially in terms of recording conditions. The sampling rate is 22.05 kHz throughout, and the voice database comprises approximately 20,000 distinct unit types, with a number of units per unit type varying between 1 and about 20,000. The average hovers around 50.

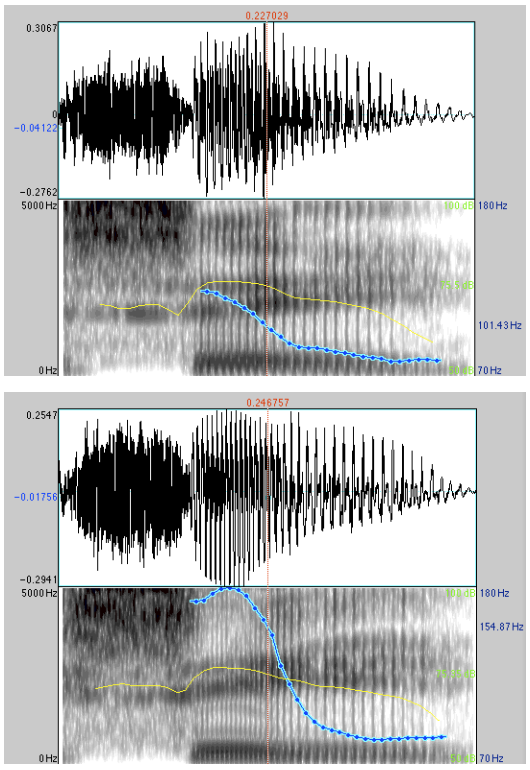

Fig. 4. *Two Speech Segments for "see" from Cluster 1.*

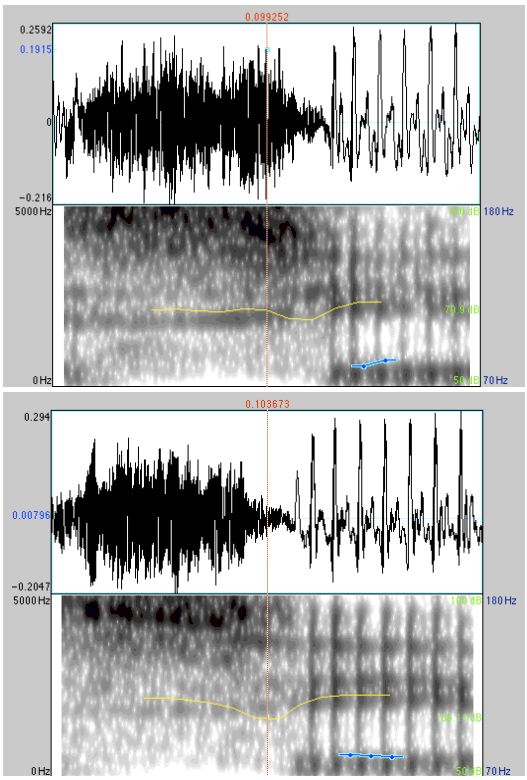

Fig. 5. *Two Speech Segments for "see" from Cluster 2.*

3 points (featuring one or more large coordinate, i.e., closest to the edges of the cube).[4]

Next, each cluster was analyzed in detail for acoustico-linguistic similarities and differences. We found that the first cluster contained instances of "see" spoken with an accented vowel and a falling pitch, as for example would occur when the word is spoken just before an intonational phrase boundary. Two speech segments from this cluster are illustrated in Fig. 4. While distinct, they clearly possess a number of similar characteristics, in terms of duration (both about 500 ms), pitch dynamic (consistent blue lines), intensity contour (comparable yellow lines), etc.

The second cluster contained instances of "see" spoken with an unaccented vowel and a flat or perhaps slightly rising pitch, as for example would occur when the word is spoken between accented material. The two associated speech segments are illustrated in Fig. 5. Again, they bear some resemblance in terms of short duration (both about 200 ms), low pitch, low intensity, etc. More importantly, it is quite evident that both of them are very different from the speech segments of Fig. 4.

Finally, the third cluster contained instances of "see" spoken with a distinctly tense version of the vowel and a flat or slightly falling pitch. Two speech segments from this cluster are illustrated in Fig. 6. Again, they clearly resemble each other more than any speech segment from the other two clusters.

It could be argued that the degree of consistency exemplified in Figs. 4–6 is rather unusual in everyday speech. We conjecture, however, that it may well be the norm rather than the exception when it comes to TTS unit inventories, due to highly monitored recording conditions and the frequent use of professional voice talent.

[4]This last cluster may be harder to visualize at first, because the human eye naturally favors Euclidean distance over the measure (2).

In all cases, it feels that replacing one unit by another from the same cluster would largely maintain the "sound and feel" of the utterance, while replacing it by a unit from a different cluster would be (sometimes seriously) disruptive to the listener.

To illustrate, the attached files "`Cluster3a.aiff`" and "`Cross3a_3b.aiff`" give two renditions of the sentence:

He offered his binoculars so they could see for themselves.

where in the first file the word "see" comes from the original recording, while in the second file it was spliced in from a different recording from the same cluster (Cluster 3). No further processing was done to the second rendition, yet it is hard to perceive a difference when comparing it to the original.

In contrast, the two files "`Cross3a_1a.aiff`" and "`Cross3a_2a.aiff`" give two renditions of the same sentence where the word "see" was spliced in from recordings coming from either Cluster 1 or Cluster 2 (again, without any other processing). It is immediately obvious that the spliced units "stand out," and substantially modify the "sound and feel" of the original utterance.

4. LISTENING TEST

To further establish the practical validity of the method, a more formal listening test was conducted. As stimuli, we generated a set of 5 sentences synthesized from each of 3 different unit inventories: (i) the original inventory, where no pruning was performed, which corresponds to a reduction factor of $RF = 1$, (ii) the inventory obtained by setting the target reduction factor to $RF = 1.25$, which corresponds to a moderate pruning of 20% of all units, and (iii) the inventory obtained by setting the target reduction factor to $RF = 2$, which corresponds to a more aggressive pruning of 50% of all units.

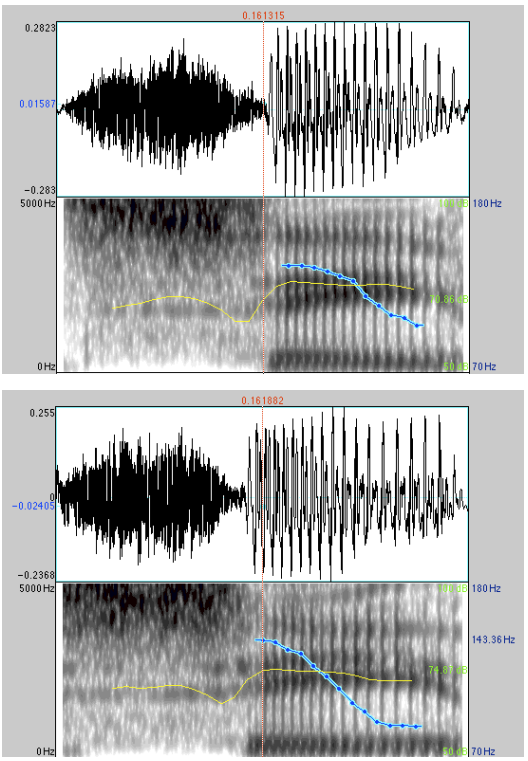

Fig. 6. *Two Speech Segments for "see" from Cluster 3.*

Typically, moderate pruning removes mostly outliers, while aggressive pruning removes redundant and near-redundant units as well.

In order to establish a point of reference for LSM-based pruning, we ran a first set of experiments where pruning was done by removing units at random to achieve the above reduction factors. Four listeners participated in this baseline study. They were asked to score each of the five utterances from the 3 different databases on the standard MOS scale, where 5 is the best. Tabulating the results yields the score distributions presented in Table I. This table shows that a substantial degradation occurs when removing units at random, even in the case of moderate pruning. This probably stems from the well known fact that a single badly rendered unit in a sentence often ruins perception for that entire sentence, even though the rest of the units may well be otherwise acceptable.

The second set of experiments involved the same listeners as above, plus four additional participants, including two with no background whatsoever in speech processing. In this series pruning was performed as detailed in the previous sections, using a maximum of $R = 50$ for the dimension of the LSM space. Again listeners were asked to score each of the five utterances from the 3 different databases on the standard MOS scale. Tabulating the results yields the score distributions presented in Table II. This time, on the average, the sentences synthesized from the pruned inventories were not rated noticeably worse than those synthesized from the baseline inventory. In one instance (utterance 2), the MOS score is even slightly higher when pruning is used, which we conjecture is due to the removal of a borderline outlier unit that just happened to be picked when synthesizing this particular utterance. Overall, the relative degradation in perceived quality seems to be limited to under 3% when removing 20% of the units, and under 5% when removing half of the units in the database.

Table I: Mean Opinion Scores for Baseline Pruning.

Utterance Number	No Pruning $RF = 1$	Moderate Pruning $RF = 1.25$	Aggress. Pruning $RF = 2$
1	3.0	1.5	1.1
2	2.9	2.0	1.3
3	3.9	2.1	1.1
4	2.8	1.2	1.0
5	2.8	1.1	1.0
Average MOS	3.08	1.58	1.10
95% Confid.	± 0.82	± 0.80	± 0.21

Table II: Mean Opinion Scores for LSM-based Pruning.

Utterance Number	No Pruning $RF = 1$	Moderate Pruning $RF = 1.25$	Aggress. Pruning $RF = 2$
1	3.3	3.0	3.0
2	2.4	2.9	2.9
3	4.1	4.0	4.0
4	3.0	2.6	2.5
5	2.9	2.8	2.6
Average MOS	3.14	3.06	3.00
95% Confid.	± 1.10	± 0.96	± 1.04

5. CONCLUSION

We have further illustrated some of benefits of the unit pruning procedure introduced in [1], through the detailed analysis of a simple case study, as well as two sets of listening evaluations involving both moderate and aggressive pruning. These experiments suggest that LSM-based unit-centric pruning can indeed reduce the size of the unit inventory without noticeable degradation in perceived TTS quality. Future efforts will concentrate on more systematically exploring the influence of the decomposition parameters (particularly R), in order to better characterize their relationship to factors such as unit type, number of instances, dominant style of elocution, and overall prosodic context distribution.

6. REFERENCES

[1] J.R. Bellegarda, "LSM–Based Unit Pruning for Concatenative Speech Synthesis," in *Proc. ICASSP*, Honolulu, HI, pp. IV-521–IV-524, April 2007.

[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] 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.

[4] N. Campbell, "CHATR: A High–Definition Speech Re–Sequencing System," in *Proc. 3rd ASA/ASJ Joint Meeting*, Honolulu, HI, pp. 1223–1228, December 1996.

[5] A.W. Black and K. Lenzo, "Optimal Data Selection for Unit Selection Synthesis," in *Proc. 4th ISCA Speech Synth. Workshop*, Perthshire, Scotland, paper 129, August 2001.

[6] J.R. Bellegarda, "A Global, Boundary–Centric Framework for Unit Selection Text–to–Speech Synthesis," *IEEE Trans. ASL*, Vol. ASL–14, No. 3, pp. 990–997, May 2006.

[7] 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.

[8] J.R. Bellegarda, K.E.A. Silverman, K.A. Lenzo, and V. Anderson, "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.