UNIT SELECTION IN A CONCATENATIVE SPEECH SYNTHESIS SYSTEM USING A LARGE SPEECH DATABASE

Andrew J. Hunt and Alan W. Black

ATR Interpreting Telecommunications Research Labs.
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
andrew,awb@itl.atr.co.jp

ABSTRACT

One approach to the generation of natural-sounding synthesized speech waveforms is to select and concatenate units from a large speech database. Units (in the current work, phonemes) are selected to produce a natural realisation of a target phoneme sequence predicted from text which is annotated with prosodic and phonetic context information. We propose that the units in a synthesis database can be considered as a state transition network in which the state occupancy cost is the distance between a database unit and a target, and the transition cost is an estimate of the quality of concatenation of two consecutive units. This framework has many similarities to HMM-based speech recognition. A pruned Viterbi search is used to select the best units for synthesis from the database. This approach to waveform synthesis permits training from natural speech: two methods for training from speech are presented which provide weights which produce more natural speech than can be obtained by hand-tuning.

1. INTRODUCTION

Synthesized speech can be produced by concatenating the waveforms of units selected from large, single-speaker speech databases. The primary motivation for the use of large databases is that with a large number of units available with varied prosodic and spectral characteristics it should be possible to synthesize more natural-sounding speech than can be produced with a small set of controlled units (e.g. diphones) [1].

In previous work at this laboratory on the ATR ν-Talk Japanese speech synthesis system, the selection of units from a large database was based on minimising acoustic distortions between selected units and the target spectrum [2, 3]. The research presented in this paper has been carried out within the CHATR speech synthesis system which also selects units from large, single-speaker databases but which extends the ATR ν-Talk principle to take into account both the prosodic and phonetic appropriateness of units. The primary goal of introducing prosodic information to the selection criteria is to reduce the extent of signal processing required to correct the prosodic characteristics of the units (e.g. using PSOLA, [4]) because increased prosodic modifications tend to reduce the quality of the speech output [1].

The unit selection procedure of CHATR has also proved to be flexible. CHATR has been used to synthesize speech from a wide range of databases including male and female speakers, Japanese and English, and isolated words and continuous read speech [5]. These databases have varied in size from 10 minutes to 150 minutes.

This most important issue to be solved to make this concatenative synthesis approach effective is the selection of appropriate units from the database. This paper advances previous research on CHATR by characterising the selection procedure as Viterbi decoding of a state transition network consisting of all units in a synthesis database, as described in Section 2. This new view of the synthesis database permits automated training using natural speech. In Section 3, two novel methods are presented for training the cost functions which control unit selection. In Section 4, results from subjective assessment of the synthetic speech are presented and discussed.

2. UNIT SELECTION

The input to CHATR is typically text, though this may be augmented with structural and discourse information. The first stages of synthesis transform this input into a *target specification* (or simply *target*). The target for an utterance defines the string of phonemes required to synthesize the text, and is annotated with prosodic features (pitch, duration and power) which specify the desired speech output in more detail. This paper is not concerned with the procedures required to produce the target specification, but instead focuses on the selection of appropriate units from a database to synthesize the target.

Unit selection in CHATR is based on the two *cost functions* shown in Figure 1 [5]. The *target cost*, $C^t(u_i, t_i)$, is an estimate of the difference between a database unit, u_i, and the target, t_i, which it is supposed to represent. The *concatenation cost*, $C^c(u_{i-1}, u_i)$, is an estimate of the quality of

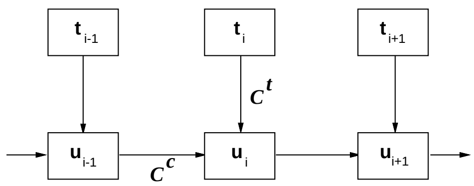

Figure 1. Unit Selection Costs

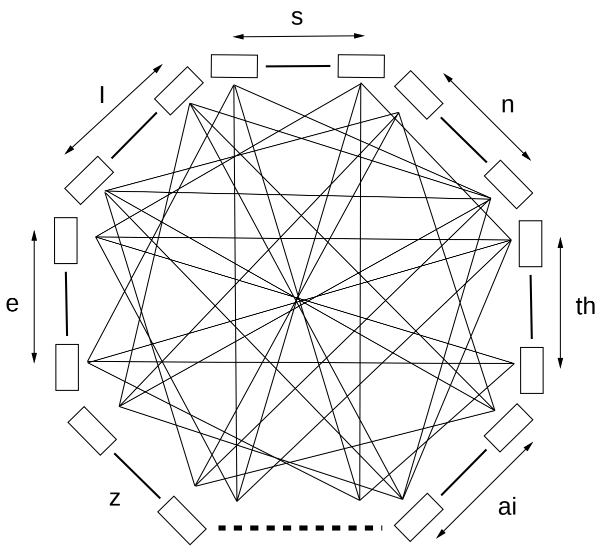

Figure 2. Phoneme Network for a Database

a join between consecutive units (u_{i-1} and u_i). Section 2.1 describes how the database units can be treated as a *state transition network* which is decoded by these costs. Section 2.2 descibes how the costs are calculated and Section 3 describes the training of the costs.

2.1. A Database as a State Transition Network

Given the target specification, the sequence $t_1^n = (t_1, ..., t_n)$, we need to select the set of units, $u_1^n = (u_1, ..., u_n)$, which are closest to the target. By selecting units close to the target we can minimise the extent of the signal processing required to produce prosodic characteristics and thus minimise distortion of the natural waveforms. The speech database containing the candidate units can be viewed as a state transition network with each unit in the database represented by a separate state. The state occupancy cost is given by the target cost, and the state transition cost is given by the concatenation cost. Because any unit can potentially be followed by any other, the network is fully connected. A target phoneme is always synthesised by a database unit with the same phonemic identity.

Figure 2 illustrates a speech database as a state transition network (showing the units required to produce the word "synthesize"). The states (boxes) represent all the phonemes in the database (organised according to phonemic identity) and the transitions (lines) are all possible concatenation sequences. When synthesizing the word "synthesize", we have a target specification t_1^8 with the phonemes */s-I-n-th-e-s-ai-z/* and with each target having a desired pitch, duration and power. The task of waveform synthesis is to find the path through the state transition network, i.e. the sequence of database units, with the minimum cost.

It is worth pointing out that this treatment of a synthesis database has many similarities to HMM-based speech recognition systems [6]. In HMM terminology, the state transition network of database units is an ergodic first-order HMM in which the observation sequence is the target specification, and the best sequence of selected units is the hidden path. The important distinction is that Markov models are probabilistic, whereas the current work uses cost functions.

2.2. Cost Functions and Unit Selection

The selection of good units for synthesis requires an appropriate definition of the target and concatenation costs and effective training of these costs. As was described above, each target phoneme has a target pitch, power and duration. From the sequence of targets we can also determine the prosodic characteristics and the phonetic identity of the preceding and following phonemes. Similarly, given phonetic labelling of the synthesis database (by forced alignment or hand-labelling), we can use standard signal processing techniques to obtain identical information about each phoneme in the database. Thus, each target phoneme and each candidate in the synthesis database is characterised by a multidimensional *feature vector*.

Special treatment is required for the phonetic context as it is by nature a discrete (non-numeric) feature. The following distinctive features are used to characterise the preceding and following phonemes: vowel vs. consonant, voicing, consonant type, point of articulation, and vowel height, length, and rounding. When comparing the distinctive features of two phonemes a comparison function is used: zero if they are the same or one if they differ.

The target cost is calculated as the weighted sum of the differences between the elements of the target and candidate feature vectors: these differences are the p *target sub-costs*, $C_j^t(t_i, u_i)$ $(j = 1, ..., p)$. In the current implementations p varies between 20 and 30. The target cost, given weights w_j^t for the sub-costs, is calculated as follows:

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) \qquad [1]$$

The *concatenation cost*, $C^c(u_{i-1}, u_i)$, is also determined by the weighted sum of q *concatenation sub-costs*, $C_j^c(u_{i-1}, u_i)$ $(j = 1, ..., q)$. The sub-costs can be determined from the unit characterisations of u_{i-1} and u_i (as with the target cost), but may additionally be derived from signal processing of the units. Three sub-costs were used in the current work (i.e. $q = 3$): cepstral distance at the point of concatenation and the absolute differences in log power and pitch. The concatenation cost, given weights w_j^c, is calculated as follows:

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \qquad [2]$$

As a special case, if u_{i-1} and u_i are consecutive units in the synthesis database, then their concatenation is natural and therefore has a cost of zero. This condition encourages the selection of multiple consecutive phonemes from the synthesis database, referred to as *non-uniform units* in the ATR ν-Talk system.

The total cost for a sequence of n units (i.e. a path through the state transition network) is the sum of the target and concatenation costs:

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad [3]$$

where S denotes silence, and $C^c(S, u_1)$ and $C^c(u_n, S)$ define the start and end conditions given by the concatenation of the first and last units to silence. Expanding Equation 3 to include the sub-costs we obtain the following:

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) + \sum_{i=2}^{n} \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad [4]$$

The unit selection procedure is the task of determining the set of units $\overline{u}_1^n$ so that the total cost defined by Equation 4 is minimised:

$$\overline{u}_1^n = \min_{u_1, \ldots, u_n} C(t_1^n, u_1^n) \quad [5]$$

Optimal unit selection can be performed with a Viterbi search. However, to obtain near real-time synthesis on large speech databases, containing tens of thousands of units, the search space must be pruned. This has been implemented by multiple pruning steps. Initially, units with phonetic contexts similar to the target are identified. Next, the remaining units are pruned with the target cost and finally with the concatenation cost [5]. With a beam width of 10-20 units, the search can be performed in near real-time on a database with around 100,000 units (on a Sun SPARCStation 20). Synthesis is faster than real time for smaller databases (less than 50,000 units). Pruning appears to have little effect on the output quality.

3. TRAINING THE COST FUNCTIONS

The most complex issue to be addressed is the training of the weights of the cost functions (w_j^t and w_j^c). This section describes two novel approaches to determining the weights. The first approach, *weight space search* can be characterised as a limited search of the weight space. The second approach, *regression training* involves exhaustive comparison of the units in the database and multiple linear regression.

Both training methods use targets from natural utterances held out from the synthesis database. The role of training is to determine weights which minimise the difference between the natural waveform and the waveform output of the synthesizer given the target specification of the natural utterance. The target specification for a natural utterance is accurately specified by the set of feature vectors of the database units which make up that utterance.

An *objective distance measure* is used to determine the difference between synthesized and natural utterances being withheld from the database. The objective distance measure should reflect as much as possible the *perceptual* similarity of the utterances. Currently, the cepstral distance between the waveforms is used as the objective distance measure.

3.1. Weight Space Search

Given a set of weights we determine the best set of units from the database using Equation 5 (i.e. the Viterbi search), synthesize the waveform, and determine its distance from the natural waveform using the objective distance function. This process is repeated for a range of weight sets and for multiple utterances. The best weight set is chosen as the one that performs most consistently across the utterances.

In the current work 3-5 possible weight values were tested for the prosodic and phonetic context features. All possible combinations of these values were tested on at least 10 training utterances requiring the synthesis and comparison of possibly 100,000's of waveforms.

This training method has some important limitations. Most importantly, the computational requirements grow exponentially with the number of weights being trained and with the number of values used for each weight. This problem has been reduced by training the weights in multiple passes, but still requires 150+ hours of training time for a speech database of 40,000 units (approximately 1 hour of speech) on a Sun SPARCStation 20.

3.2. Regression Training

Regression determines the weights for the concatenation cost and the weights for the target cost separately. A previous experiment by the authors studied the perception of the concatenation of speech segments in Japanese [7]. Analysis of the concatenation of a range of units showed that a linear combination of cepstral distance and difference in power at the point of concatenation is a reasonable predictor of the perceptual quality of a joint. The results from this experiment determined the weights, w_j^c, of the concatenation cost. As mentioned in Section 2.2., the difference in pitch at the point of concatenation was also included in the concatenation cost. The earlier experiment controlled the influence of pitch, so the weight for pitch difference was determined by hand-tuning.

The weights for the target cost were obtained using the objective distance function and multiple linear regression. The training process can be applied to obtain a phoneme-dependent weight set (different weights, w_j^t, for selecting different phonemes), and can also generate different weights for sets of phonemes (e.g. all nasals) or all phonemes together. The steps in regression training are as follows:

1. For each example unit in the synthesis database from the phoneme set currently being trained, perform steps a-d.
   (a) Treat the example unit as a target unit.
   (b) Calculate the acoustic difference between this target and all other instances of the same phoneme in the synthesis database using the objective distance function.
   (c) Identify the set of n-best matches to this target (n=20).
   (d) Determine the target sub-costs $C_j^t(t, u_i)$ for the target unit and the n-best matches.
2. Collect the objective distances and target sub-costs across all the targets and all the n-best matches.

3. Use linear regression to predict the objective distance by a linear weighting of the t target sub-costs. Use the weights determined by linear regression as the weights for the target sub-costs, w_j^t, for current phoneme set.
4. Repeat steps 1-3 for each phoneme set.

The goal of this training algorithm is to determine weights for the target sub-costs which select units that are close to those which would be selected if the objective cost function could be used directly in unit selection. In this way, the process takes advantage of the waveforms of the natural speech which are available in the synthesis database.

The regression training method has many advantages over the weight space search. In particular, it is able to efficiently generate separate weights for different phoneme classes for which the influence of prosodic and phonetic-context factors may be different, and it can train many more weights (training time scales linearly with the number of sub-costs, rather than exponentially as is the case for weight space search). Regression training is also computationally more efficient; training time is reduced by as much as one hundred times. Typical training times are between 1 and 10 hours, depending on the size of the database. Moreover, training time can be reduced to 5 to 30 minutes if the acoustic distances (see step 1b) are pre-calculated permitting rapid evaluation of different sub-costs.

4. DISCUSSION

Both training methods have been applied to a range of synthesis databases including Japanese and English, and male and female speech. Synthesized speech produced from weights of either training method is consistently better than that produced with hand-tuned weights. However, hand-tuning of global unit selection parameters can improve the quality of synthesis with automatically trained weights. For example, an overall increase or decrease in the concatenation weights can be used to trade-off between prosodic correctness and the smoothness of concatenation of units. This trade-off becomes less important as the size of the synthesis database increases.

Several tests have been carried out to compare the synthesised speech produced by the two training methods described in this paper. The results show a consistent but small preference for weights obtained by regression training. We had expected regression training to provide more substantial improvements because of its greater sophistication and because it can train separate weights for different phoneme classes. There is no clear explanation for this result.

Nevertheless, regression training is now the preferred training method because of its substantially lower computational requirements and greater flexibility. A number of areas of regression training are currently under consideration for improvement. Modifications to the objective distance function to include power and pitch (as additions to the cepstral parameters) appear promising. Enhancement of the target sub-costs is another avenue for improving unit selection, in particular through extension of the statistical framework of training to permit the inclusion of discrete parameters into the feature vectors. Additionally, the statistical framework may be improved by using step-wise linear regression or other robust techniques. Finally, we are investigating extensions of regression training to automatically prune synthesis databases (i.e. the removal of poor quality or redundant units from the synthesis database).

5. CONCLUSIONS

This paper has presented a new view of a synthesis database for use in unit concatenative speech synthesis. The units in a synthesis database can be treated as states in a state transition network with the state occupancy costs given by the target cost, and the state transition costs given by the concatenation cost which is an estimate of the quality of concatenation of pairs of units. Given the two costs, the network can be decoded using a pruned Viterbi algorithm. Two methods have been presented for training the target and concatenation costs, weight space search and regression training. Both methods use natural speech to train the weights used in selection costs and provide weights which produce better quality synthesis than hand-tuned weights. Although there is little difference in the quality of output using the two training methods, the regression training method is more effective because of its substantially lower computational requirements and greater flexibility.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. Yasuhiro Yamazaki for his support.

REFERENCES

[1] N. Campbell and A. Black. Prosody and the selection of source units for concatenative synthesis. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in Speech Synthesis*. Springer Verlag, 1995.

[2] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura. ATR ν-talk speech synthesis system. In *Proc. 1992 Intl. Conf. on Spoken Language Processing*, pages 483–486, Banff, Canada, 1992.

[3] N. Iwahashi, N. Kaiki, and Y. Sagisaka. Concatenative speech synthesis by minimum distortion criteria. In *ICASSP '92*, pages II–65–68, 1992.

[4] E. Moulines and Charpentier F. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9(5/6):453–467, 1990.

[5] A. Black and N. Campbell. Optimising selection of units from speech databases for concatenative synthesis. In *EUROSPEECH '95*, pages 581–584, Madrid, Spain, 1995.

[6] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE Proceedings*, 77 No 2:257–285, 1989.

[7] A.J. Hunt and A.W. Black. An investigation of the quality of concatenation of speech waveforms. Technical report, ATR Interpreting Telecommunications Research Laboratories: TR-IT-0137, 1995.