SPEECH SYNTHESIS BY PHONOLOGICAL STRUCTURE MATCHING

Paul Taylor and Alan W Black

Centre for Speech Technology Research, University of Edinburgh,
80, South Bridge, Edinburgh, U.K. EH1 1HN
http://www.cstr.ed.ac.uk
email: {pault, awb}@cstr.ed.ac.uk

ABSTRACT

This paper presents a new technique for speech synthesis by unit selection. The technique works by specifying the synthesis target and the speech database as phonological trees, and using a selection algorithm which finds the largest parts of trees in the database which match parts of the target tree. The technique avoids many of the errors made by prosody generation modules by incorporating their operation in the selection implicitly. A technique for using signal processing only when it is needed most is also described. The technique produces better quality speech than previous approaches and is also significantly faster.

1. INTRODUCTION

It is common in any overview of a speech synthesis system (e.g. [13], [8]) to see the system broken down into a number of components, which nearly always include things such as text normalisation, lexical lookup, intonation, duration, diphone concatenation and signal processing. A standard model of waveform generation over the last years has been for the higher level modules to provide a narrow phonetic transcription, which specifies phonetic content, pitch and duration, and then to use diphones combined with signal processing to realise this as speech.

What is less common these days is to see speech synthesis systems which actually deal with the phonetics explicitly (compare the amount of space devoted to phonetics in [1] compared to that in [13]). The main reason for the success of diphone synthesis over formant or rule based phonetic synthesis is that the complexity of the interactions involved in phonetic speech production becomes overwhelming to the system developer, making natural sounding rules very difficult to write. The adoption of diphone synthesis has allowed developers to bypass this whole complex area of synthesis by implicitly modelling all the phonetics affects within the diphones. To put it another way, developers have given up on trying to understand how low level speech production actually works, and have chosen a data driven technique that solves (or attempts to solve) the problem for them.

In more recent years, many improvements on the basic diphone model have been proposed ([11], [6], [3], [5] [12], [7], [2], [4]) which have often been called "unit selection". While the details of these systems differ, they all adopt a similar strategy. In standard diphone synthesis, a single diphone is used for each phonetic specification, and its pitch and duration are modified by signal processing to meet the target specification. In unit selection synthesis, several units with different pitch, durations and prosodic contexts are compared, and the most appropriate is then used. Some systems leave it at that [11] while others further modify the pitch and duration to make sure it exactly fits the target specification [5].

So substantial progress has been made with regard to improving the segmental side of synthesis by using natural units in appropriate environments. However, in nearly all systems the F0 contour and durations are specified the way they always have been, by a combination of models and rules. While on one had we admit that segmental information is too hard to generate explicitly, on the other we continue to believe that explicit models/rules can be used to generate prosody. Unit selection algorithms are often successful at finding a unit of the pitch and duration that were specified in the target description, but often these targets are simply bad, and hence unnatural sounding speech is produced. Furthermore, as unit selection techniques use a narrow phonetic transcription as input, phenomena such as vowel reduction and assimilation have to be modelled before unit selection operates, and the modules which govern this may make errors in the same way.

This paper proposes a new synthesis technique, whereby we move the waveform generation one level higher again and select units based on phonological information only. That is, information such as canonical pronunciation, phrase-final position and accentuation is used for selection, rather than phonetic transcriptions, millisecond durations and F0 values. In effect we have moved the low level prosody component into the waveform generation part of the system, and abdicated responsibility for modelling this explicitly. This paper describes how this is done and why this is beneficial.

2. PHONOLOGICAL STRUCTURE MATCHING

2.1. Phonological Trees

The Phonological Structure Matching (PSM) algorithm works as follows. The high level component of the system produces a target representation as a tree which contains only phonological information - an example is shown in figure 1. Each utterance in the database is de-

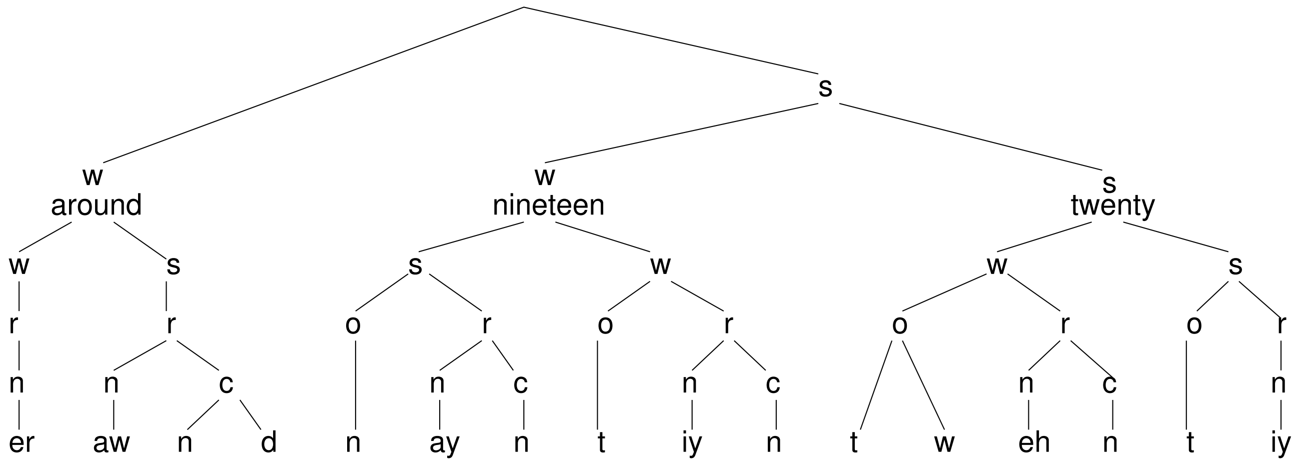

Figure 1. A fragment of an utterance's phonological tree. The higher nodes are in the form of a binary metrical tree in which relative strength relations of strong (s) and weak(w) are expressed between every sibling node. Above the word, this tree is derived from the syntax tree with rules used to assign s and w to nodes. Beneath the word, lexical stress in conjunction with stress shifting rules are used to assign s and w. Beneath the syllable, the tree takes the form of a standard syllabic structure model with nodes representing onset, rhyme, nucleus and coda. Phones form the leaf nodes of the tree.

scribed in exactly the same way, i.e. as a phonological tree reaching from a sentence node to the phones. Each node in the tree is an attribute value list, so that phone nodes have information such as place manner and voice features, whereas word nodes have their lexical head word and part-of-speech. Intonation is specified in terms of an accented/unaccented attribute on each syllable.

In the PSM algorithm "units" are simply nodes in the tree. The PSM algorithm first assigns units in the database to nodes in the target tree and then concatenates these units to form speech. In a given sentence, units may represent a word (e.g. "nineteen"), a phrase (e.g. "nineteen twenty") a syllable ("nine"), a syllabic node (e.g. a onset as in (/t/ /w/)) or a single phone.

The algorithm has three stages, finding candidates, selection between them, and concatenation/signal processing.

2.2. Finding Candidate Units

Candidate finding is performed by comparing nodes in the target tree to nodes in the database. A matching function is defined which when given two arbitrary nodes, will return true if the trees under each node match, and false otherwise. The definition of this function is externally specified, but in our current setup, the function returns true if the trees beneath each node match exactly in structure and have the same phones as terminal nodes.

The algorithm starts by assigning the root node in the target tree to be the target node. The following operation is then performed:

```
PSM(target)
    find candidates in db matching target
    if num of cands > 0
        assign candidates to target
    else
        for each daughter in target
            PSM(daughter)
```

Each node in each utterance in the database is examined, and if the matching function returns true, the unit in the database is assigned to the target node as a candidate. The entire database is searched every time, so that all units which match are assigned as candidates to the target node. If no candidates are found, the current target node's daughters are set to be the target node, and the database is searched again. This continues until all the nodes in the target tree are dominated by a node which has at least one match (there is a possible fail scenario whereby no matches are found - but this would only happen in the highly unlikely situation whereby the database contained less than one example of each phone). When this process is complete, the target tree will have candidate units at various positions in the tree. These units relate to arbitrary sized units in the database, and thus can be phones, syllables, words or groups of words in a phrase.

2.3. Selecting Units

The selection process then decides which of the candidates on each node is best. While all the candidates on a node will dominate a tree of the same structure and the same leaf phones, they may differ in other ways, such as stress patterns and intonation (e.g. words like "nineteen" can have their main stress on the first or second syllable). Also, they may differ in terms of their original phonetic, word and phrasal contexts. For example, if we want to synthesize the sentence "Around nineteen twenty, jazz became suddenly more popular", and had several candidates for the first three words, a candidate from the sentence "Around nineteen twenty the suffragette movement became a strong force" would be better than the same words from "The suffragette movement became a strong force around nineteen twenty" due to the difference in having the words in a phrase initial or phrase final position.

A scoring function is used to judge how well candidate units match the target unit in terms of context and secondary information. The candidate with the best score is chosen as the unit and the rest are discarded.

2.4. Back-off Strategy and Signal Processing

From here, it is a simple matter to concatenate the waveforms of the various selected units to form a synthetic speech utterance. In this case the PSM algorithm corresponds to a "pure" unit selection algorithm of the Hunt and Black [11] type in which no signal processing is used. The pure unit selection paradigm has been proposed as a means of eliminating the inevitable distortion caused by using signal processing to change the pitch and duration of a signal. The argument is, that by selecting units of the correct target pitch and duration, signal processing should not be needed. Unfortunately, it is a simple fact of the combinatorics that many units will not exist in the database (consider how many units would be needed to have every phone in every context in every duration at every pitch - billions). This is one of the main reasons why pure unit selection sometimes makes very bad mistakes - there simply isn't an appropriate unit available and no amount of increasing the database will make significant inroads into this.

In the PSM algorithm, we bypass this problem by making use of signal processing in cases where units with inappropriate pitch and duration are found. When scoring the candidates in the selection process, a normalised factor α can be calculated as a measure of how well that unit fits the target, 0 being a perfect fit and 1 being the worst fit (remember this is in terms of prosody only - all candidates have the same phonemic identities). A set of automatic duration and intonation targets is also generated using the standard techniques described in the introduction. If α is further modified according to the perceived distortion that signal processing entails, an optimal balance between the target and source prosody can be calculated. For instance, the final duration of a segment is calculated as $d_{final} = \alpha d_{target} + (1-\alpha) d_{source}$. The intonation parameters can be calculated in the same way. Once the final duration and intonation targets are chosen, signal processing is used to change the duration and pitch appropriately (currently by a residual excited linear prediction technique). In this way it is possible to have the best of both worlds, natural sounding prosody copied from the database when it is possible, mixed with artificially generated prosody when it is needed.

2.5. Parameters and Training

The PSM algorithm has remarkably few parameters to set, in fact the most difficult "parameter" to set is to decide the exactly what phonological information should be encoded in the trees. In the more normal sense the main parameters are weights which decide the relative importance of stress, intonation and context when choosing units. So far these have been set by hand on held our training data. This is possible because the number of parameters is small, but in the future perceptual tests will be used in conjunction with reinforcement learning algorithms.

3. COMPARISON WITH "STANDARD" UNIT SELECTION

PSM works on phonological representations rather than acoustic/phonetic ones. This is advantageous for a number of reasons, not least of which is the simple fact that the high level components can generate more reliable phonological specifications that phonetic specifications.

In a traditional model, a lexical string of phones for a word is converted into a surface string by using post-lexical rules which take into consideration factors such as the position of the word in the phrase and the function that word plays in the overall stress structure of the sentence. Duration targets are then calculated for each phone using the same higher level factors. Modelling lexical to surface changes in this way is crude, as vowel reduction is not simply a case of replacing a full vowel with a schwa and assigning a shorter duration: many subtle complex spectral and timing relations are involved.

In a phonological tree, words are described in terms of their canonical lexical pronunciation - surface level differences are encoded by virtue of the word's position in the tree and the stress and phrasing relationships that go along with that. Thus units are selected based on these features and phenomena like vowel reduction and co-articulation are modelled implicitly.

Phonological representations have further advantages in that they are more compact representation than acoustic/phonetic ones. This has many advantages in reducing the dimensionality of the functions with perform the matching and scoring. Because of the reduced dimensionality and the independence of phonological features, it is much easier for algorithms (and humans) to train parameters optimally.

The PSM algorithm tries to assign the biggest units possible to nodes and this has significant advantages for minimising concatenation distortion. Experimentation has confirmed the common sense idea that co-articulation is stronger within constituents than across constituent boundaries at any level in the tree. After working with the system, it is clear to us that synthesizing phrases from whole phrase units is better than doing so from originally non-contiguous words, synthesizing words from whole word units is better than doing so from originally non-contiguous syllables, and so on for syllables and syllabic constituents. Hence preserving constituents as complete units helps reduce discontinuities at unit boundaries.

Larger units are better because they allow us to bypass the errors caused by enforcing a strict segmentation of the signal into units such as phones, syllables and words. As yet we do now how to effectively describe the assimilation, co-articulation and reduction affects observed in common phrases like "going to" realised as "gonna", but by modelling this as a single unit we don't have to fully understand this process.

When we look at the computational complexity of the phonological structure matching over the Hunt and Black algorithm [11] there is significant improvement. In the Hunt and Black case, every occurrence in the database of a target phone must be measured so the best n candidates can be found and carried forward to next part of the selection process.

In the PSM algorithm, if the database contains candidates in the correct context, they will automatically be chosen first and searching will not be performed on candidates in inappropriate contexts. Thus if the target is the syllable /m a n/, and a syllable /m a n/ exists in the database, it will be chosen as a candidate. Other occurrences of /m a n/ will also be chosen, but then the search will stop. Importantly, /a/ units of different contexts (e.g. /b a d/) will simply be ignored.

Exceptionally, when a target utterance has no overlapping contexts with any constituent in the database (except at the phone level) PSM will require the same amount of comparisons as Hunt and Black. However this extreme situation is both unusual and would only occur when the target domain badly matches the database itself. A second important complexity issue is that PSM can select units longer than single phone thus the number of combination that need be checked will be less that in the Hunt and Black case. This substantial increase in speed is made possible by building in the assumption that longer matching units are always better than shorter matching units.

4. LIMITED DOMAIN SYNTHESIS

While the PSM algorithm was initially formulated for use in a unconstrained TTS system, we have also been exploring is use as a limited domain synthesizer. It is clear that a major use for speech synthesis is in spoken message generation systems which only operate in a limited domain. As the full flexibility of a standard TTS system isn't need, the hope is that by concentrating on the specifics of the domain, higher quality synthetic speech can be generated. Yi and Glass [14] achieved very good results to this by carefully designing a set of phrase, word and segment sized units which give optimum coverage of the domain they are dealing with. We adopt a slightly looser approach and instead of trying to explicitly design a set of units, we simply record and annotate a representative set of utterances from the domain in question, and add these to the database.

We have tried this in two domains, the MIT Jupiter weather report domain [9] and the more unrestricted ILEX museum guide domain [10]. PSM is ideally suited to these domains, as frequently occurring phrases such as "In Boston today" or "partly cloudy skies" occur several times in the databases as whole phrases and hence sound very natural when synthesized.

However, a simple slot and filler approach would not work as it is often the case that novel phrases or words occur. Hence the flexibility of an approach like PSM is needed. For both these domains, a small amount of domain specific speech (say 1/2 hour) is enough to greatly increase the quality of the output speech as compared to a TTS system. There is no chance of domain specific data getting "lost" in the standard database - because the PSM algorithm always looks for the longest units domain specific units will usually be found first. Building a domain specific PSM system is simply a case of recording and annotating some in-domain data: no retraining or system restructuring are required. As the provision of general in-domain data is all that is needed, it is not the case that a linguistic expert is needed to adapt the system to a new domain.

5. CONCLUSION

We have presented a new synthesis algorithm and have argued that it is superior to acoustics/phonetic phone based unit selection in several ways: it bypasses errors made by the post-lexical rules and low level prosody models, it doesn't suffer from sparse data problems, it has a fast search strategy and it is easily usable for limited domain synthesis. Furthermore, we have added the capability to use signal processing in places where it is needed most.

Future work will concentrate on providing better phonological tree representations for the target and database utterances, adding new signal processing techniques and developing a reinforcement training technique for setting the parameters.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of the UK Engineering and Physical Science Research Council (grants GR/L53250 and GR/K54229) and Sun Microsystems. We would like to thank Joe Polifroni for providing us with the text of the Jupiter sentences.

REFERENCES

[1] J. Allen, S. Hunnicut, and D. Klatt. *From Text to Speech: the MITalk System*. Cambridge University Press, 1987.

[2] A. Black and P. Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Eurospeech97*, volume 2, pages 601–604, Rhodes, Greece, 1997.

[3] Andrew P. Breen and Peter Jackson. A phonologically motivated method of selecting non-uniform units. In *ICSLP 98*, 1998.

[4] 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*, pages 279–282. Springer Verlag, 1996.

[5] Alistair Conkie. A robust unit selection system for speech synthesis. In *137th meeting of the Acoustical Society of America*, 1999.

[6] Andrew Cronk and Michael Macon. Optimized stopping criteria for tree-based unit selection in concatenative synthesis. In *ICSLP 98*, 1998.

[7] R. Donovan and P. Woodland. Improvements in an HMM-based speech synthesiser. In *Eurospeech95*, volume 1, pages 573–576, Madrid, Spain, 1995.

[8] Thierry Dutoit. *An Introduction to Text to Speech Synthesis*. Kluwer Academic Publishers, 1997.

[9] Victor Zue et al. From interface to content: translingual access and delivery of online information. In *Eurospeech97*, volume 4, pages 2227–2230, Rhodes, Greece, 1997.

[10] Janet Hitzeman, Chris Mellish, and Jon Oberlander. Dynamic generation of museum web pages: The intelligent labelling explorer. *Archives and Museum Informatics*, 11:107–115, 1997. Also presented at the Museums and the Web Conference, Los Angeles, March 1997.

[11] Andrew J. Hunt and Alan W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *International Conference on Acoustics Speech and Signal Processing*. IEEE, 1996.

[12] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura. ATR – ν-TALK speech synthesis system. In *Proceedings of ICSLP 92*, volume 1, pages 483–486, 1992.

[13] Richard Sproat. *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Kluwer Academic Publishers, 1998.

[14] Jon R. W. Yi and James R. Glass. Natural sounding speech synthesis using variable length units. In *ICSLP 98*, 1998.