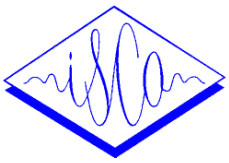

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

A REAL TIME SPEECH SYNTHESIS SYSTEM

P. A. Taylor, I. A. Nairn, A. M. Sutherland, M. A. Jack

Centre for Speech Technology Research, University of Edinburgh, U.K.

ABSTRACT

A real time speech synthesis system for a personal computer is described. Details are given of real time considerations and an explanation is given as to how the system was implemented. The main modules of the linguistic and synthesis component are described. The flexible configuration of the system is demonstrated and two example applications are examined.

1 INTRODUCTION

At CSTR we have developed a speech synthesis system that works in real-time on a personal computer. This paper will describe the implementation of this system, the details of the stages involved and the possible configurations in which the synthesizer can be used.

The speech synthesis system consists of two components. The first component's task is to produce a linguistic and prosodic analysis of the input (eg. text) and produce a parametric representation of the utterance. The second component is then the synthesizer itself, which takes this parametric representation and uses signal processing to produce a speech waveform.

2 COMMON FRAMEWORK

The linguistic component is broken down into well defined modules which read and write from a central multi-level data structure. All levels of this data structure can be accessed by any of the modules allowing free and easy reference to information. This allows easier development and software control but has more important, wider implications when considering the possible configurations in which the system may be used. The main modules in the linguistic component are; text to phoneme conversion, syntactic parsing and intonation provision.

The synthesis component uses a database of 2400 diphones and the PSOLA signal processing technique to produce speech waveforms.

The speech synthesis system is designed to work in a variety of applications areas, some of which needing a full unconstrained TTS capability, others may have different requirements. Often these other applications will have knowledge of how sentences are to be spoken and therefore can guide the system in its processing. The modules have thus been designed so that they can be configured in a variety of ways to suit each application. The operations these modules perform can be guided by a system of escape codes (described below) in situations where the application has knowledge of how the module is to perform. As such, the description of the modules in section 4 should be seen as 'default' operations that are carried out when no application knowledge is present.

Escape codes have been defined to guide modules in their operation. The modules use their default mode of operation when no escape codes are present. The escape codes can be used to by-pass certain processes or to guide what would be otherwise automatic procedures. For example, escape codes can be used as guidance for text anomalisation, to mark new/old information or to make sure intonation accents are placed on the correct words.

3 IMPLEMENTATION

3.1 REAL TIME DESIGN

A major drawback of most TTS systems is that they require additional expensive hardware to cope with the heavy processing required of them. This is disadvantageous because of the cost incurred in building such a system, the difficulty in incorporating new features in the system once built, and the general inflexibility of the system once operational. Because of the ever decreasing cost of computer hardware, we have decided that a dedicated hardware synthesis system is no longer needed and therefore we have decided to implement our system entirely in software. This enables our system to run on a variety of hardware platforms. The ANSI C programming language was chosen for the entire system due to its portability and speed.

The real-time performance of the system has been made possible by using the PSOLA algorithm (Hamon et al 1989). In other systems, the use of frequency domain modifications or parametric models (LPC or formant synthesis) greatly adds to the computational expense of the synthesis. As all the signal processing in our system is performed in the time domain, no transformations of any kind need to be carried out and no floating point arithmetic is needed.

The PC is the target machine for the system, but due to the 'software only' nature of the system, it has been easy to port the system onto other machines including the Sun Sparcstation. Three real time systems currently exist on the PC:

The first, which is the closest to a traditional TTS system, takes text input from the keyboard or file and produces utterances which can be stored as waveforms or alternatively as synthesizer input parameters. The later is preferable as speech waveforms are stored at 40,000 bytes/second against 30 bytes/second for the synthesizer parameter format.

The second system has all the capability of the first but is integrated with a graphical user interface. This interface allows the user to have more control over the way the utterances are produced. The user interface concentrates on the areas where the automatic modules have the most difficulty in producing acceptable output. Via mouse and keyboard interaction, the user can alter the system of escape codes to produce various effects. Text anomalies can be tagged so as dates and times etc can be spoken in any desired way. As intonation is one of the most difficult problems in TTS, the user can also place accents and decide accent prominence via the interface.

The third system is a stand alone synthesizer which can take the parametric input format described above and produce speech waveforms.

3.2 PC REQUIREMENTS

A 386 PC is needed to run the system, but no 387 co-processor is required due to the absence of floating point arithmetic. The PC needs at least 2 Megabytes of RAM to operate effectively. However if only 2 MB RAM is present, the synthesizer cannot store all the diphones in memory at once, so a diphone swap system is employed where the most frequently used diphones are kept in RAM while others are swapped to and from disk as required.The PC needs 5 MB RAM to be able to store all the diphones in RAM simultaneously. By downsampling the diphone database, memory requirements can be greatly reduced. Such an approach would be appropriate in applications where the synthesizer was to operate over the telephone. In this case the size of the database could be reduced by a factor of 2.5. The size of the run-time word lexicon can also be optimised for memory requirements, with any size being used from a few hundred words to 27,000.

4 SYSTEM OVERVIEW

This section will describe the details of the algorithms used in the various modules in the linguistic component.

4.1 TEXT TO PHONEME

The Text to Phoneme system converts raw input text to a phonemic representation, along with part of speech, stress and syllable boundary information. The system comprises of various modules. A major difference between our approach and that of others is the size of our dictionary. Most TTS systems only have a small (few thousand word) exceptions dictionary and rely on the morphological decomposition and letter to sound rules to produce a phonemic transcription of the text. As memory (disk or RAM) is cheap today there is no great expense incurred in using enough RAM to store a large dictionary. The dictionary is more accurate with grapheme to phoneme conversion and is very useful at providing other information such as part of speech and lexical stress. The dictionary can also provide alternate pronunciations or word classes where appropriate. A flow chart of this module is shown in figure 1. Typically, 95% of words follow the left path of the chart.

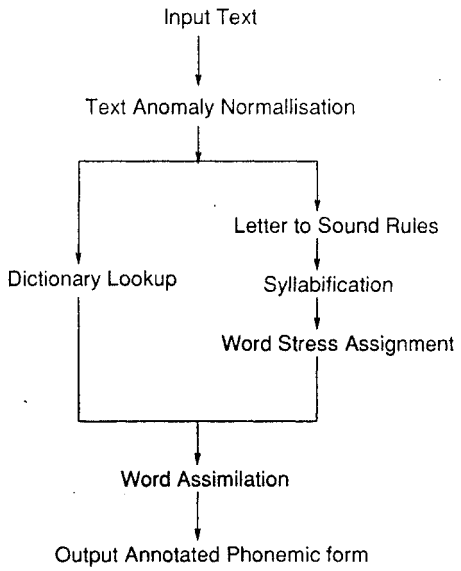

Figure 1. Flow chart for the text to phoneme module

4.11Text anomaly normalisation

This reads in free formatted text and converts it to an internal standard format. A complete sentence is identified, then words and white space are isolated. Any unknown characters, such as graphic characters, are treated as white space. Numbers are initially converted to digit form, (eg 12.3 is converted to one two point three), but via escape codes these can be realised as dates, times, numbers etc. Punctuation is used to indicate candidate clause and sentence boundaries.

4.12 Dictionary Lookup

A 27,000 word lexicon comprising of part of speech information, lexical stress, syllable boundaries and word pronunciations is available. By using a binary 'look up', a word's phonemic entry can be found from the lexicon. Run time lexicons of different sizes can be constructed from the core lexicon for different applications and storage requirements.

4.13 Letter to sound rules

This module converts any words that are not in the lexicon to their phonemic forms, based on a system by McIlroy (1974). It accomplishes this by a number of string rewrite rules which convert a word (or parts of it) to its phonemic form. It also has a number of heuristics to deal with final s's & e's, etc.

4.14 Syllabification.

Once a phonemic form is generated by the letter to sound rules, this module inserts syllable boundaries. The heuristic used is based on rules supplied by Gimson (1984) and O'Connor (1954). In all cases tried (> 22K words) this module obtained the correct syllabification.

4.15 Finite State Rule Formalism

A two level finite state transducer (FST) system is available for interpreting rule formalisms. This is based on the system by Antworth (1990). This FST mechanism is used to generate word stress using a system described by Williams (1987). This mechanism is also used to produce assimilation effects which are needed to modify the phonemic form of an utterance. The main purpose of assimilation is to convert the citation form of words into forms which are more usually found in continuous speech and to deal with word boundary effects.

4.2 MORPHOLOGICAL DECOMPOSITION

A morphological decomposition module (that is currently being ported from LISP to C) will be eventually included in the system to process words that are not in the dictionary. Morphological analysis is a more preferable technique than letter to sound rules due to its greater accuracy in producing the correct phonemes for a word, and its ability to give part of speech information.

4.4 PARSING

At present, only a simple clause based deterministic parser is implemented. This uses part of speech classifications to place non-hierarchical phrase boundaries. Work is being carried out on implementing a more sophisticated parsing architecture in which various parsing strategies and grammars will be tried. Top-down, bottom-up and statistical parsers are being considered. Different grammars may be built for different applications so as to optimise the parsing module for each situation.

4.4 INTONATION

Intonation assignment is possibly the most difficult part of speech synthesis. For fully natural intonation to be produced, an understanding of

the sentence is needed. At present there are no systems which even attempt speech understanding due to the difficulty of the problem. Understanding is very difficult to obtain from the text alone, but if the linguistic component has extra knowledge from the application (such as in the example given in section 5), the production of natural intonation will be made much easier.

A Pierrehumbert style of intonational notation is used throughout the intonation module. (Pierrehumbert 1979).

4.41 Accent Assignment

The present technique assigns pitch, phrase and boundary accents by reference to the deterministic parse of the sentence. This is similar to the system used by Silverman (1987). It is hoped in the future to have more sophisticated assignment algorithms when a full parse of the sentence is available.

Work is under way to allow applications to assign accents by examining any semantic and pragmatic information that an application may have available.

4.42 Accent Realisation

Once a sentence has been annotated with intonation markers, a set of pitch values are determined according to the method of Silverman (1987). Our system is more flexible than the one used by Silverman as our synthesis component can accept as many target points as is needed on each phoneme, and these target points can be at any position and of any frequency value.

4.1 DIPHONE SYNTHESIS

The diphone synthesizer produces speech waveforms according to the specification given by the grapheme to phoneme conversion process and the intonation assignment. The synthesizer first uses diphones from a database of 2400 diphones to generate a waveform of the correct phonemic sequence for the utterance. The prosody of the sentence is then imposed by using the time domain PSOLA signal processing technique (Hamon et al. 1989) to alter the durations and pitch of the sentence.

The diphone database currently being used was generated from a single male British English RP speaker. An automatic method has been developed to facilitate the quick and easy preparation of a diphone database for new speakers (Taylor & Isard 1991). Work is now under way to produce an new diphone set from a female speaker.

4.2 SPEECH CODING

The diphone database consists of a set of speech waveforms. The diphones were recorded at 20 KHz sampling rate and at 16 bit accuracy. The diphone database took up approximately 17 Meg. This was too big to fit into the memory of most target machines so some speech coding was needed to reduce the memory requirement. A simple 4-bit ADPCM coder, which has been optimised for coding diphones, was developed which reduced the size of the diphones to about 4.4 Meg.

5 APPLICATIONS

In assessing the market potential for speech synthesis, we have concluded that the quality of our current synthesizer component is good enough for many commercial applications, but the difficulty lies in taking unconstrained text as the input to the linguistic component. Due to the unconstrained nature of the TTS task, the linguistic component has to cope with all the variability that may occur in any style of the language in question, and then produce an output which allows the synthesizer to produce speech which is natural and intelligible.

By studying potential applications for systems, it became apparent that very few applications would need the ability to process totally unconstrained text. Even in the much quoted case of using TTS for reading newspapers to the blind, the purpose of the task is still well defined and not totally open. It seemed that in most applications a general grammar and vocabulary could be defined ahead of time and if this set up was adequate, the linguistic component could create parameters which could be used by the synthesis component to produce speech with greater naturalness and intelligibility.

5.1 EXAMPLE APPLICATIONS

Two trial applications that do not use an unconstrained TTS approach are described here.

5.11 Message Generation

A message announcement system is a simple application where a set of messages need to be spoken by machine. Where the set of spoken messages can be easily defined before hand, speech coding can be used but when the complexity or number of messages greatly increases the speech coding solution becomes less attractive. The diphone database represents 7 minutes of stored speech, so if the application needs to store more speech than this, a great saving can be made by storing the speech as input synthesis parameters as these only take up 30 bytes/second.

A message generation system could be used in areas such as database query applications where a single variable (such as a person's name) could have thousands of different possibilities.The grammar of the application would be well defined and as such could be prescribed before hand, as could the vocabulary and intonation information. The graphical user interface configuration of the synthesis system (cf. section 3) could be used to generate carrier sentences prior to run time which would ensure that correct pronunciation was used and that the intonation was suitable. At run-time, the numbers or names required by the application could be inserted in the carrier sentences in parametric form and used as input to the stand alone synthesizer.

5.12 Machine Dialogue

In many situations, grammar and vocabulary will not be so easy to define ahead of time and a more flexible approach is needed. Conversational computing is one such area where again the task may be somewhat less than unconstrained, but not so easy to define as in the message generation system. Such a system, in conjunction with a speech recognition system, would generate responses to a user's questions. Here the machine will be only be able to give responses in ways it knows how to, but may have to use unknown words or phrases given by the recognition system. Hence the vocabulary and grammar can not be fully defined prior to run time.

A sentence generation module would be required to generate the responses to the users questions. Such a module would obviously have a knowledge of the grammar of the sentence (thereby eliminating the need for a parser) and could use the system of escape codes to guide the assignment of intonation. In this way the task of synthesis becomes easier as application knowledge can guide the system in its processing.

6 EVALUATION OF SYNTHESIS QUALITY

The synthesis component of the system was used in a large scale evaluation of synthesis intelligibility. (Sydeserff et al.1991). The experiment tested a variety of synthesizers to produce a metric on which to evaluate performance. The experiment showed that the quality of the

synthetic speech was of the same order as the commercially available DECTalk system (Klatt 1982). Since this experiment was carried out, new pitchmarks have been produced for the diphone database. This new set of pitchmarks contains fewer errors and subsequently the quality of the speech has risen.

7 CONCLUSIONS

The real time speech synthesis system described here is a move away from text to speech programs towards a more flexible speech generation system that can be used for a variety of purposes. The quality of the speech produced has been judged to have been good in comparison with other commercially available systems. Due to the software orientated approach taken, our system can be manufactured at a fraction of the cost of other systems.

REFERENCES

Antworth, E. L., (1990) PC-KIMMO: *A two level processor for Morphological Analysis,* Summer Institute of Linguistics.

Gimson, A. C., (1983) *An Introduction to the Pronunciation of English,* Edward Arnold (Publishers) Ltd., Third Edition.

Hamon, C., Moulines, E., Charpentier, F., (1989) *A Diphone Synthesis System based on Time-Domain Modifications of Speech,* Proceedings ICASSP.

Klatt, D. H., (1982) *The Klattalk Text-to-Speech System.* Proc. ICASSP '82 p1589 - 1592.

McIlroy, M. D., *Synthetic English Speech by Rule,* Bell Telephone Laboratories, 1974.

O'Connor, J. D., Trim, J. L. M., (1953) *Vowel, Consonant and Syllable; a Phonological Definition,* Word, Vol 9, August 1953.

Pierrehumbert, J. B., (1980) *The Phonology and Phonetics of English Intonation,* PhD Thesis, Massachussetts Institute of Technology.

Silverman, K. E. A., (1987) *The Structure and Processing of Fundamental Frequency Contours,* PhD Thesis, University of Cambridge.

Sydeserff, H. A., Caley, R. J., Isard, S. D., Jack, M. A., Monaghan, A. I. C., Verhoven J., (1991) *Evaluation of Speech Synthesis Techniques in a Comprehension Task.* EUROSPEECH '91.

Taylor, P. A., Isard, S. D., (1991) *Automatic Diphone Segmentation,* Proceedings EUROSPEECH '91.

Williams, B, (1987) *Word stress assignment in a text-to-speech synthesis system for British English,* Computer Speech and Language, Vol 2, 1987.