SPEECH SYNTHESIZER BASED ON TIME DOMAIN SYLLABLE CONCATENATION

Ljubomir Josifovski[1], Dragan Mihajlov[2], Dejan Gorgevik[2]

[1] Faculty of Mechanical Engineering
Karpos II bb, Skopje, Macedonia
e-mail: ljupco@ereb.mf.ukim.edu.mk
tel: +389 91 363 566 ext. 299
fax: +389 91 362 298

[2] Faculty of Electrical Engineering
Karpos II bb, Skopje, Macedonia
e-mail: {dragan|dejan}@cerera.etf.ukim.edu.mk
tel: +389 91 363 566 ext. 156
fax: +389 91 364 262

ABSTRACT

In [2] we have presented a subsystem for text-to-speech (TTS) conversion for macedonian language as a part of a system for support of humans with damaged eyesight [1]. In this paper we present the speech synthesizer which is part of the TTS conversion subsystem. It's based on time-domain syllable concatenation. A novel module for duration and fundamental frequency (F_0) modification is introduced and discussed. We believe that the architecture presented is well suited to the nature of macedonian language, and fits well with prerequisites for real-time operation on a standard, of-the-shelf hardware. The prototype containing inventory of 1275 syllables and implementing modules for duration and F_0 modification was built and tested. The preliminary tests concerning the eligibility of the synthesized speech are most encouraging.

1. INTRODUCTION

The texts from newspapers, magazines, books etc. have to be recorded on audio tapes or printed in Braille writing in order to be presented to the humans with damaged eyesight. An automated system for speeding up the process is under development [1]. It includes automatic reading of printed Macedonian Cyrillic text, its archival, conversion to speech by TTS subsystem and printing on a Braille printer (Figure 1). This paper addresses the speech synthesizer part of the TTS subsystem. The written material is scanned first. OCR on the bitmap containing Cyrillic text [3] is performed next. The resulting ASCII text file can be printed on a Braille printer, feed to a TTS subsystem or archived and reused later.

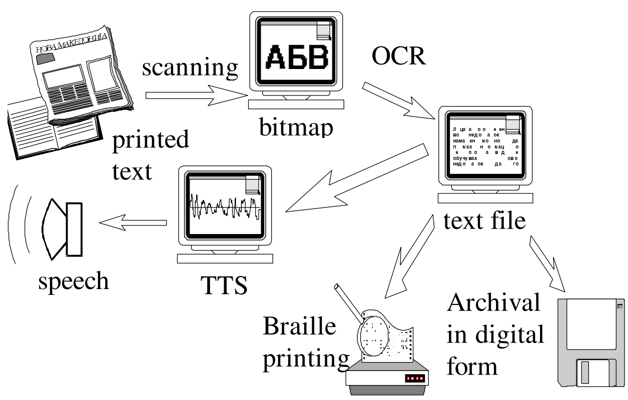

Figure 1. A system for helping people with damaged eyesight

2. TTS SUBSYSTEM OVERVIEW

A number of systems for TTS conversion from unrestricted text, such as MITalk [4], CNET [5], AT&T Bell Labs. multilingual TTS system [6], TTS systems for Welsh [7], Polish [8], Japanese [9] & [10], Slovenian [11] etc., have been developed. However, ideal TTS system, indistinguishable from the human reading, is yet to be constructed.

All of the mentioned systems share the notion that the process of TTS conversion is to be divided into two major steps. In the first step, input text is converted into some form of linguistic representation. Such representation includes information on phonemes to be produced (including pauses) and their duration, pitch (where applicable) and amplitude (or power) over the time. This step deals with natural language processing (NLP) mostly. In the second step, the previous information is converted into speech waveform This is the final output of the whole TTS subsystem. The module performing this step is commonly referred to as a speech synthesizer.

The main aim we were concerned of during our TTS subsystem design was early and rapid bootstrap of a system that will produce intelligible speech from text on a standard, off-the-shelf hardware. During the first step we modeled only the phenomena (phrasing, intonation, lexical stress) with the biggest impact on the speech intelligibility. Our method of choice for the speech synthesizer part was time-domain concatenation of prerecorded syllables with altered duration and F_0.

The NLP part of the TTS subsystem consists of modules for syllabification, phrasing, lexical stress and intonation assessment and F_0 and duration prediction.

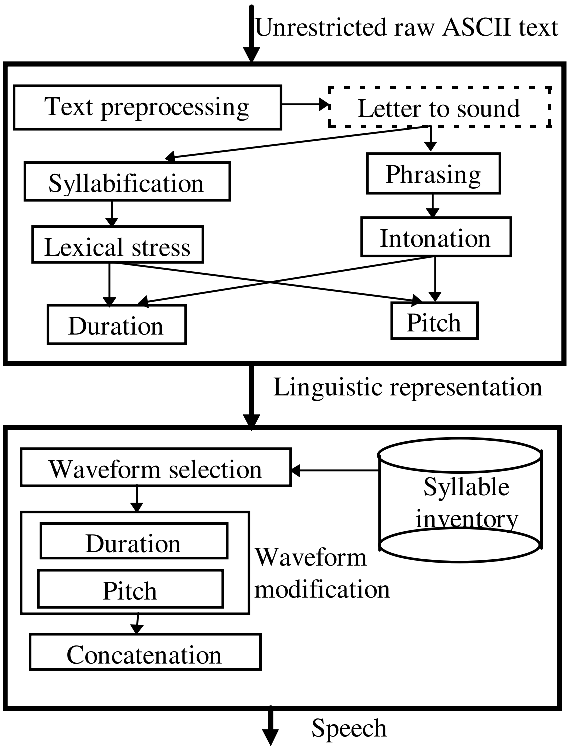

Figure 2. Architecture of our TTS subsystem

Syllabification is necessary for syllable selection and lexical stress assessment. The usage of complete dictionary of presyllabified words or construction and evaluation of a set of rules (together with exceptions dictionary) for syllabification were abandoned in favor of a techniques for automatic induction of rules for syllabification from small presyllabified lexicon (24898 in the latest version) of words. A three layer (186x12x1 in the current incarnation), feed-forward neural network (NN) based estimator of the probability that there is a syllable break after a given letter in a certain context was trained using on-line backpropagation. The training corpus consisted of 55094 training patterns containing the letter in question and context of one letter before, and four letters after together with the correct answer. The error of misplaced syllable breaks on the testing set was less then 3% [2]. In the current implementation, the single network is replaced with 31 smaller networks (three layer, 155x4x1 architecture), each corresponding to a separate letter (as there are 31 letters in macedonian language). Separate networks are much faster and easyier trained.

The module for lexical stress assignment is implemented by rules. The phrasing and intonation modules are still under development.

3. SPEECH SYNTHESIZER BASED ON TIME-DOMAIN SYLLABLE CONCATENATION

The speech synthesizer for macedonian language is based on concatenation of prerecorded syllables in time domain. The synthesizer consists of: an inventory of prerecorded syllables with variable lengths, syllable duration and F_0 changer, and syllable concatenator (Figure 3).

Other popular approaches include parametric representations of speech segments like Linear Predictive Coding (LPC) [6], [5] or formant synthesis [4], [8]. We rejected formant synthesis because it's impossible to acquire data and numerous rules necessary to operate such synthesizer for macedonian at present time. LPC synthesis was dropped as less natural and computationally more expensive compared to time-domain based methods [12], [9].

The choice of syllable as a basic speech inventory unit is somewhat more disputable. Phonemes [9], diphones [5] & [6], demysyllables [13] or multiple instances (with different F_0, duration and phonemic context) of one to several phones [10] are possible basic speech inventory units of choice. The need to eliminate the coarticulation effects on the basic unit in greatest extent (as there is little research in this area for macedonian) was the main factor that influenced our decision.

The syllable inventory contains syllables recorded in time domain in known phonetic context and information describing the context. Syllables are extracted from carrier sentences uttered by a professional speaker. It is important that syllabic environment in carrier words ensures that syllable in question doesn't carry lexical (primary) stress. Thus it can be easily modified in both ways (stressed and unstressed). Small set of the most frequent words is also added to the inventory.

The first inventory was build manually. Tools for automatic syllables extraction and normalization from longer sequences of prerecorded speech are investigated. In the next version we plan to extend the inventory with several instances of the syllables, each in different phonemic context, F_0 and duration. Thus, the process of conversion of syllabified text into the set of syllables from the inventory will take into account recommendations [10] about the allowable F_0 and duration modification.

Thorough review of non-parametric techniques for F_0 and duration modification of speech, both in frequency and time domain, can be found in [14]. We modify duration and F_0 on the baseis of the non-linear oscillator model of speech [15]. The model enables both linear and non linear duration scaling without syllable labeling and tagging (thus can be done without manual work) and is feasible in real-time. Sub/super sampling the syllable with altered duration results in fundamental frequency modification. Both duration and fundamental frequency modification are performed together, in a single step.

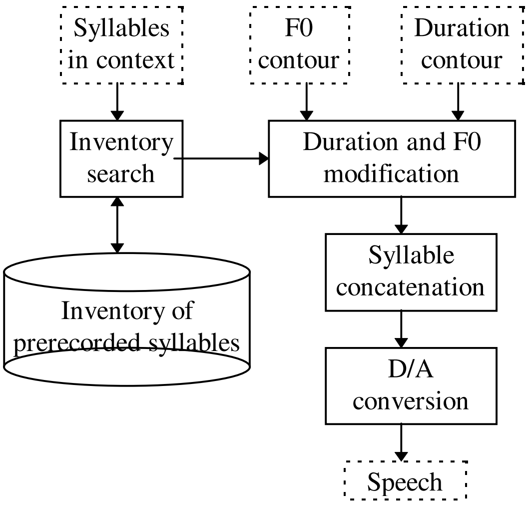

Figure 3. Speech synthesizer for macedonian language

In the next stage, syllable concatenation consisting of cross fading of overlapped junctions of the apposite syllables is performed. Sometimes that leads to unsatisfactory results. Different approach, involving partial labeling of the syllable structure and algorithm that makes use of that information is currently considered (at a cost of additional men work for labeling). In order to smooth spectral discontinuities in the overlapped junction, pitch-synchronous cross fading may be applied. The cross fading region should be at least one pitch period wide. The exact beginning of the region can be determined using fast algorithm to find the minimum absolute error (MAE) [16] between the apposite regions.

The choice of time domain operation ensures real-time operation on a standard personal computer (and contributes to eligibility, too). However, choice of large phonetic units such as syllables leads to high eligibility of the synthesized speech. The choice of syllable as a basic unit nicely fits with the feature of the macedonian language - short syllables with two to four phonemes account for approximately 97% of all syllables. Thus, inventory contains relatively small number (1275 in the current version) of syllables. Additionally, the effects of coarticulation are implicitly taken care of, as they rarely cross over the syllable boundaries.

4. DURATION AND F_0 MODIFICATION

The duration and F_0 are modified according the model and technique introduced in [15]. Speech is modeled as a nonlinear oscillator. Short-time stationary intervals which characterize vowels are interpreted as attractors of the underlying system. Noise-like waveform consonant patterns are transitions from one attractor to another.

The oscillator is defined as nonlinear feedback system using state space representation. The N-dimensional state vector X(n) is obtained from N delayed samples:

$$X(n) = [x(n), x(n-M), ... x(n-(N-1)M)]$$

The delay parameter M defines the "subsampling" factor and is not necessary equal to 1. Product NM defines the duration of the state vector.

The state transition function a(X) is a nonlinear mapping from X(n-1) to X(n). Actually, only the most recent component x(n) of X(n) is produced. Other (N-1) components are shifted components of X(n-1), as shown on Figure 4. The unknown component x(n) of X(n) is estimated using the "nearest-neighbor" method. First, a table of all state transitions observed in given waveform is constructed. Known the current value of the state vector, the output value is the table entry nearest to the state vector. Oscillator stability is guaranteed, at a price of lack of any noise suppression mechanism.

The model is adapted to the short-time nature of speech by adaptation of the state-transition codebook approximately every 1 ms (see Figure 4). Speech is divided in frames of length L (20 ms in the current implementation). Next frame advances over the previous with step K (1 ms long). All samples in a frame are used to build the state-transition codebook of size L x (N+1) samples. Each of the L codebook entries consists of a short waveform segment of length N and an immediately following sample. Once the codebook is built, signals of arbitrary length can be synthesized from free-running oscillator. For time-scale modification, after the frame is advanced over previous and new codebook is built, free-running oscillator produces RK samples (R is the time-scaling factor).

Codebook search computational cost can be greatly reduced if performed only once every J samples. Once the best match is found, J subsequent samples are copied from the original waveform. But clearly audible discontinuity appears in the output waveform on the place where sample is obtained by codebook search. In [15], linear predictive smoothing is used as an alternative to costly time-domain interpolation of the original waveform. However, linear predictive smoothing is itself a costly procedure if the speech coder is not LPC based.

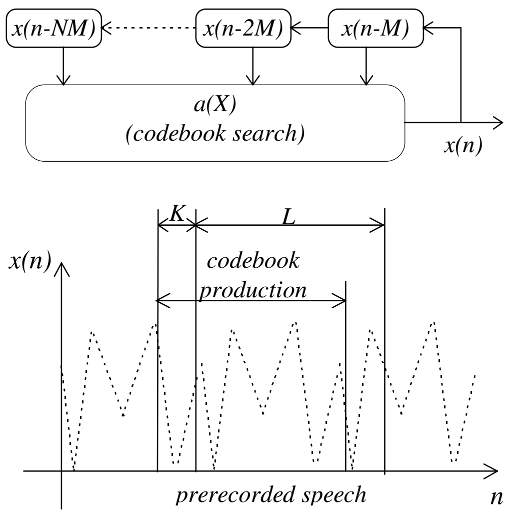

Figure 4. Speech modeled as nonlinear oscillator

Instead, we modified the algorithm to discard "skip-J-codebook-searches" method that leads to discontinuities but still maintain real-time operation. Instead skipping over J codebook searches, we do codebook search for every sample. However, if the result of the previous search was row *u* in the codebook, then, it's most likely (given the continuos nature of speech) that the (*u+1*)-th row will be the result of the next search. Therefore, we can reduce the searching time (up to factor *L/2*) by beginning to search from row *u+1* instead of *1*.

Next, instead of searching for the best match over the whole codebook, it is sufficient to find the "close enough" match in the codebook and end the search. We used the $\sum_{i=1}^{\dim ension} abs\left[codebook(i) - sample(i)\right]$ metrics to measure the "closeness" of the match.

The method is easily extended to ensure non-linear time scaling by simply producing R_iK instead of RK samples in every step. The F_0 modification is also taken into account. It's realized by sub/super sampling. This translates into occasional insertion or deletion (no codebook search in these cases) of the speech samples in the above algorithm.

The use of above modifications of the basic algorithm ensured real time operation while still maintaining acceptable time-scaling ranging from 50% to 400%. We currently use sampling frequency of 22050 Hz, 20 ms long frame L, 1 ms long step K, dimensionallity of 10 and threshold for the metrics of 5000.

5. CONCLUSION

Speech synthesizer based on time-domain syllable concatenation is presented in this paper. The synthesizer is used in a TTS conversion module for macedonian language which is part of system for helping people with damaged eyesight. A novel module for duration and F_0 modification is introduced and discussed in the paper. We believe that the architecture presented is well suited to the nature of macedonian language, and fits well with prerequisites for real-time operation on a standard, of-the-shelf hardware. The prototype containing inventory of 1275 syllables and implementing modules for duration and F_0 modification was built. The preliminary tests concerning the eligibility of the synthesized speech are most encouraging.

REFERENCES

[1] D. Mihajlov, D. Djordjevik, N. Kotevska, “Computer System for Support of Humans with Damaged Sight”, ETAI, Ohrid, 1993

[2] Lj. Josifovski, D. Mihajlov, D. Djordjevik, “Text-to-Speech Conversion for Macedonian as Part of a System for Support of Humans with Damaged Sight”, In Proc. of Int. Conf. on Information Technology Interfaces ITI'96, pp 61-66, Pula, Croatia, 1996

[3] D. Đorđević, D. Mihajlov, Lj. Josifovski, "Kompjuterski sistem za pomoć osobama sa oštećenim vidom: Podsistem za optičko prepoznavanje štampanog ćiriličnog teksta", YU Info, Brezovica, Jugoslavija, april 1995

[4] J. Allen, “Speech Synthesis From Unrestricted Text”, In “Computer Speech Processing”, Eds. F. Fallside & W. A. Woods, Prentice-Hall Int, pp. 461-477, 1985, London

[5] M. Stella, “Speech Synthesis”, In “Computer Speech Processing”, Eds. F. Fallside F. & W. A. Woods, Prentice-Hall International, pp. 421-460, 1985, London

[6] R. W. Sproat, J. P. Olive, “Text-to-Speech Synthesis”, AT&T Technical Journal, Vol. 74, No. 2, pp 35-44, 1995

[7] B. Williams, “Diphone Synthesis for the Welsh Language”, In Proc. of the International Conference on Spoken Language Processing, Sept. 1994, Yokohama, Japan

[8] J. Imiolczyk, I. Nowak, G. Demenko, “High Intelligibility TTS Synthesis for Polish”, Archives of Acoustics, Vol. 19, No. 2, pp 161-172, 1994

[9] T. Hirokawa, K. Itoh, H. Sato, “High Quality System Based on Waveform Concatenation of Phoneme Segment”, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. 76, No. 11, pp 1964-1970, 1993

[10] H. Kawai, et al., “Development of a TTS for Japanese Based on Wave Form Splicing”, In Proc. of IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 1, pp. 569-572, Adelaide, 1994

[11] J. Gros, N. Pavesic, F. Mihelic, "A Text-to-Speech System for the Slovenian Language", EUSIPCO'96, pp. 1043-1046, Trieste, Italy, 1996

[12] T. Dutoit, “High Quality Text-to-Speech Synthesis: A comparison of Four Candidate Algorithms”, In Proc. of IEEE International Conference on Acoustics Speech and Signal Proc, Vol. 1, pp 565-568, Adelaide, 1994

[13] V. Kraft, J. R. Andrews, "Design, Evaluation and Acquisition of a Speech Database for German Synthesis-by-Concatenation", In Proc. of 4th Australian International Conference on Speech Science and Technology, pp 724-729, Brisbane, 1992

[14] E. Moulines, J. Laroche, “Non-parametric Techniques for Pitch-Scale and Time-Scale Modification of Speech”, Speech Communication, Vol. 16, No. 2, pp 175-205, 1995

[15] G. Kubin G, W. B. Kleijn, “Time-Scale Modification of Speech Based on a Nonlinear Oscillator Model”, In Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 1, pp 453-456, Adelaide, Australia, April 1994

[16] G.-J. Lin, S.-G. Chen, T. Wu, “High Quality and Low Complexity Pitch Modification of Acoustic Signals”, In Proc. of IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 5, pp 2987-2990