jSynt: A Czech Text-to-Speech System written in JAVA

Pavel Král and Kamil Ekštein
Dept. of Computer Science & Engineering,
University of West Bohemia, Plzeň, Czech Republic
{pkral, kekstein}@kiv.zcu.cz

Abstract

This paper deals with a speech synthesis. Speech synthesis is an artificial production of speech by a computer. Speech synthesis from a text is called Text-to-Speech (TTS) synthesis. The main goal of this paper is to propose and implement a TTS system based on the MBROLA (Multiband Resynthesis Overlap-Add) project. We implement jSynt tool, a TTS system written in java. The first version includes two main languages, Czech and English. However the design of the system is general enough to fit other languages. The resulting synthesized speech is understandable, but not very natural.

1. Introduction

Human speech is the most natural form of communication between people. Therefore, it would be very beneficial using natural speech for the interaction between human and computer. An artificial production of speech by a computer is called speech synthesis, speech synthesis from a text is called Text-to-Speech (TTS) synthesis.

Many applications of speech synthesis exist, we only mention some examples: In telecommunications services, speech synthesis can replace the operator's voice, short messages can be simply pronounced by the synthesis. In public transport, synthesized speech is used instead human speech to inform the passengers about the arrivals/departures or to provide other important information. Speech synthesis could be also used for language education, but to our best knowledge, this is not done yet, due to the low quality of existing systems.

The main goal of this paper is to propose and implement a TTS system based on the MBROLA [1] (Multiband Resynthesis Overlap-Add) project.

This paper is organized as follows. Section 2 presents a short overview of speech synthesis. Section 3 describes briefly the MBROLA system that is used in our work. Next section deals with the design and implementation of the jSynt TTS system. In the last section, we discuss the results and we propose some future research directions.

2. Short Review on the Speech Synthesis

We can divide methods of speech synthesis into three following categories:

1. Articulatory synthesis: direct modeling of the human speech production system;
2. Formant synthesis: modeling the pole frequencies of speech signal or transfer function of vocal tract based on source-filter-model;
3. Concatenative synthesis: use of the prerecorded samples derived from natural speech.

Articulatory synthesis [2] is usually not used in current synthesizers, because it is too complicated for high quality implementations. However, it may be a favorable method in the future.

In current systems, formant and concatenative methods are mostly used. The formant approach [3] has been dominant for long time, but today the concatenative method [4, 5] is becoming more and more popular. Therefore, we present the details of these methods.

The main issue of the concatenative synthesis is to find correct length of the speech units that will be concatenated. Word units are practical when they are pronounced in isolation. However, the sound of the continuous sentence is not natural. The current synthesizers are thus mostly based on shorter units such as phonemes, diphones, demisyllables or on combinations of these.

Several methods of concatenative synthesis exist, we mention here the interesting ones only:

- Microphonemic method: units of variable length derived from natural speech are used [6];
- Linear Prediction (LP) based methods: based on the source-filter-model [7], filter coefficients estimated automatically from the speech;
- Sinusoidal models: assumption that the speech signal can be represented as a sum of sine waves with time-varying amplitudes and frequencies [8, 9];

- PSOLA methods [10]: very popular method, allows prerecorded speech samples smoothly concatenated and provides good controlling for pitch and duration, used in some synthesis systems as for example in ProVerbe, HADIFIX, MBROLA (system that is used in this work), etc.

3. MBROLA

MBROLA represents a speech synthesizer based on concatenative synthesis of diphones. Therefore, a database of diphones adapted to the MBROLA format is needed to run the synthesizer. This database has been created for about twenty languages for example for English, Czech, French, etc.

Input of the system is a list of phonemes with prosodic information such as duration of phonemes and a linear description of pitch. It is thus not a Text-To-Speech (TTS) synthesizer, because it does not accept raw text as an input. Output of MBROLA are speech samples on 16 bits, at the sampling frequency of the diphone database.

MBROLA is distributed for non-commercial applications for free and uses the TD-PSOLA algorithm [11] for speech synthesis.

4 jSynt TTS system

In this section, we describe design and implementation of the jSynt system. We require that jSynt could synthesize several languages (language independent) and synthesized speech would be comprehensible.

4.1 Design

jSynt TTS system operates as follows: the input text (a text file or a user keyboard input) is firstly phonetized by phonetizer module into phone sequence. The second step consists in including the relative phone duration and pitch information in the synthesized word into the phone code (using the prosodic module). Finally, the MBROLA system is called with these parameters in order to synthesize the speech.

The main part of the system is the phonetizer that encodes phones using the SAMPA alphabet [12]. Two options of phonetization are proposed:

- from a phonetic dictionary (when the word to phonetize is known);
- from phonetic rules of the target language.

The principle of the system is schematized in Figure 1.

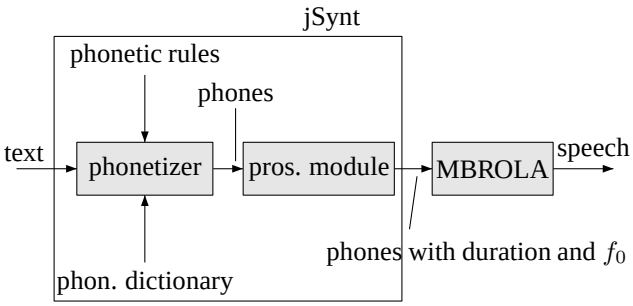

Figure 1: Principle of the jSynt TTS system: *phonetizer*: transformation of the text into the phone sequence using the phonetic dictionary and phonetic rules; *prosodic module*: including of some prosodic information (phone duration and fundamental frequency) into the each phone; *MBROLA*: speech synthesis from the prepared parameters.

4.2 Implementation

The jSynt system is implemented in Java programming language mainly due to its platform independence and its object orientation. We chose the three-layer architecture to isolate the Graphics User Interface (GUI), application and the data. This design is used,because it is simple to replace the individual components when necessary. The GUI is created by javax.swing package.

The main screen of the jSynt system is shown in Figure 2.

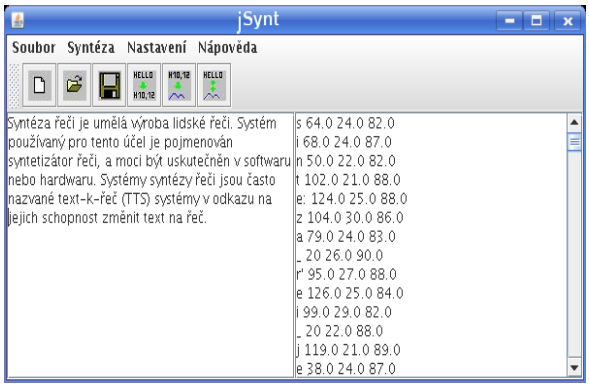

Figure 2: Main screen of the jSynt TTS system

The main configuration of the system (actual and eventual languages to synthesize) and lists of phonetic rules are stored in the XML format. Phonetic vocabularies are represented by the text file in the cp1250 charset. The first version of the system includes two main languages: Czech and English.

4.2.1 Phonetizer

As mentioned previously, this module is responsible for encoding the input text into the phone sequence. If the word is found in the vocabulary, then its phonetic representation is used, otherwise the phonetic rules are used. These rules are relatively simple for Czech language, but much more complicated for English. Therefore, in the first version of our system, only Czech rules were created. Quality of Czech synthesized speech is superior to English, because the out-vocabulary English words are phonetized wrongly.

4.2.2 Prosodic Module

The input of this module is the phone sequence in the SAMPA alphabet. Each phoneme ph is completed by its duration in *ms* (mandatory), and by a series (optional) of pitch targets composed of two float numbers each: the position of the pitch target within the phoneme (in % of its total duration), and the pitch value at this position in Hz.

The phoneme durations are described by the m and v values (mean and variance of the duration in the chosen language). Our duration value is computed as: $m \pm vr$; vr value is chosen randomly from the interval $]0.0; v]$.

Our position p and the fundamental frequency (f_0) values are computed by the analogical way as the phone duration: $x + \delta$. where x value represents average values of the position or of the pitch, δ represents the relative difference of the average values and is chosen randomly from interval $\pm 0.1 \times x$. x values are in the first version of the system constant and have been found experimentally.

The output line has thus the format: $ph\ \ d\ \ p\ \ f_0$.

4.2.3 MBROLA

Hereafter, MBROLA system is called with the previously created parameters and with the location of the diphone database. Diphone databases are available at the MBROLA project web pages[1] for several languages. The output of the system is the synthesized speech.

5. Conclusions & Perspectives

In this work, we propose and implement jSynt system, a TTS synthesizer that uses MBROLA for speech synthesis. The main requirements are the language independence and understandability of the pronounced speech.

The first version of our system includes two main languages, Czech and English. However the design of the system is general enough to fit other languages. The resulting synthesized speech is understandable, but not very natural.

Our first perspective is thus to include more complex prosodic model in order to make synthesized speech more natural. In the current implementation of the system, English phonetic rules are not available. Therefore, next perspective is to include these rules. Our actual system operates with two languages only. Therefore, our last perspective is including another languages (French, German, etc.) into our system and evaluation of the resulting synthesized speech.

[1] http://tcts.fpms.ac.be/synthesis/

Acknowledgment

This work has been partly supported by the grant NPV II-2C06009.

References

[1] T. Dutoit and H. Leich, "Text-to-Speech Synthesis based on a MBE Re-synthesis of Segments Database," *Speech Communication*, vol. 13, pp. 435–440, 1993.

[2] Klatt D., "Review of Text-to-Speech Conversion for English," *Journal of the Acoustical Society of America (JASA)*, vol. 82, no. 3, pp. 737–793, 1987.

[3] J. Allen, S. Hunnicutt, and Klatt D., "From Text to Speech: The MITalk System," *Cambridge University Press, Inc*, 1987.

[4] H. Dettweiler and W. Hess, "Concatenation Rules for Demisyllable Speech Synthesis," in *ICASSP'85*, Tampa, Florida, USA, 1985, pp. 752–755.

[5] Jordi Adell and Antonio Bonafonte, "Towards Phone Segmentation for Concatenative Speech Synthesis," in *5th ISCA Speech Synthesis Workshop*, Pittsburgh, USA, June 2004, pp. 139–144.

[6] K. Lukaszewicz and M. Karjalainen, "Microphonemic Method of Speech Synthesis," in *ICASSP'87*, Dallas, Texas, April 1987, vol. 3, pp. 1426–1429.

[7] M. Karjalainen, T. Altosaar, and M. Vainio, "Speech Synthesis using Warped Linear Prediction and Neural Networks," in *ICASSP'98*, Seattle, WA, USA, May 1998, vol. 2, pp. 877–880.

[8] Macon M., *Speech Synthesis Based on Sinusoidal Modeling*, Ph.D. thesis, Georgia Institute of Technology, 1996.

[9] K. Kleijn and K. Paliwal, "Speech coding and synthesis," *Elsevier Science B.V., The Netherlands*, 1998.

[10] Donovan R., *Trainable Speech Synthesis.*, Ph.D. thesis, Cambridge University Engineering Department, England, 1996.

[11] T. Dutoit, "High Quality Text-to-Speech Synthesis : A Comparison of Four Candidate Algorithms," in *ICASSP'94*, Adelaide, Australia, April 1994, pp. 565–568.

[12] Dafydd Gibbon, Roger Moore, and Richard Winski, *Handbook of Standards and Resources for Spoken Language Systems*, Berlin and New York: Mouton de Gruyter, 1997, Part IV, section B: SAMPA Computer Readable Phonetic Alphabet.