14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

PROSODY MODELING FOR AN EMBEDDED TTS SYSTEM IMPLEMENTATION

Dragos Burileanu and Cristian Negrescu

Faculty of Electronics, Telecommunications and IT, "Politehnica" University of Bucharest,
Blvd. Iuliu Maniu 1-3, 061071 Bucharest, Romania
phone: + (40 21) 402 4688, e-mail: bdragos@messnet.pub.ro, negrescu@elcom.pub.ro

ABSTRACT

Prosody quality strongly influences the intelligibility and the perceived naturalness of synthetic speech. But despite the significant progress in prosody modeling from the last years, incomplete linguistic knowledge that can be derived from text and various language-specific issues still limit the quality of today's commercial text-to-speech (TTS) systems. Moreover, obtaining a right pronunciation and intonation for embedded speech applications that have severe resource constraints, is a more challenging task. The paper describes an enhanced version of an embedded TTS system in Romanian language and proposes and discusses an efficient rule-based intonation model for prosody generation. Informal listening tests show that highly intelligible and fair natural synthetic speech can be produced with small memory footprint and low computational resources.

1. INTRODUCTION

Prosody is crucial for speech synthesis because it conveys aspects of meaning and structure that are not implicit in the segmental content of utterances. At the same time, prosody is difficult to predict in TTS synthesis systems because the input text contains little or no explicit information about meaning and structure, and such information is very hard to deduce automatically.

Even when that information is available, in the form of punctuation and special mark-up tags, or through syntactic (and maybe semantic) analysis, its realization as appropriate prosody is still a major challenge: the complex interactions between different aspects of prosody are often poorly understood, and the translation of linguistic categories into precise acoustic parameters is influenced by a large number of factors [8].

In text-to-speech synthesis systems, prosody is usually understood to mean the generation of fundamental frequency contour (F_0), the specification of segmental durations, and sometimes the control of intensity. Since the research presented here is mainly concerned with intonation, the paper focuses predominantly on this subject.

Intonation is certainly the basic prosody attribute and its naturalness directly affects the overall quality of synthetic speech for TTS systems. Essentially described by the fundamental frequency contour, intonation is often considered the combination of a macroprosodic component reflecting the speaker's choice of intonation pattern, and a microprosodic component, which deals with the particularly acoustic realization of phonemes [4], [6].

The previous considerations essentially explain why the ability to produce an appropriate intonation contour for the generated speech represents today a difficult task even for large-scale PC-based TTS system.

Prosody (and particularly intonation) modeling becomes more complex when we deal with embedded speech applications. The new generation of small-scale computing devices (such as PDAs, cellular phones, car kits, and various other consumer electronics products) has severe resource constraints, notably low CPU resources and small memory footprints; in addition, environment noise or poor quality of the built-in acoustic devices represents supplementary difficulties. Therefore, simplification of the current desktop synthesis engines and algorithm optimization at different processing levels (including the syntactic/prosodic analysis) are the main strategies used nowadays in the embedded TTS engines; most of the present solutions are essentially classical ones, adapted to run efficiently on computationally limited embedded platforms [1], [5].

Compared to speech recognition-based products, a relatively small number of embedded TTS solutions are deployed on the market. Most of these products use a concatenative approach and a reduced-size speech segment database (typically 1-2 Mbytes for one language) to meet the memory constraints; moreover, databases are often stored in compressed format, to further lessen the required footprint. Finally, it must be noticed that these systems provide, at the most, only "acceptable" intonation contours.

The main purpose of this paper is to present an enhanced version of an embedded TTS system and to propose a model for the F_0 contour that can produce an adequate intonation shape for prosody generation. The paper is structured as follows. Section 2 briefly discusses the architecture of our embedded TTS system and describes the proposed intonation model. Experiments and basic considerations about the overall system's performance are presented in Section 3. Section 4 concludes the paper with final remarks.

2. AN EMBEDDED TTS SYSTEM

2.1 Overview

In a previous report [3], we presented a first version of an embedded TTS system for the Romanian language. The hardware platform chosen for this embedded implementation was the Motorola MSC8101 ADS development board built around the MSC8101 processor (based on a StarCore SC140 core). This system had actually a simplified architecture, with a minimal concatenation scheme and no generated prosody, and was developed starting from a complete software (or "reference") PC-based TTS system [2].

Taking into account the remaining computational and memory resources, we recently developed a new version of our embedded TTS system, using the same hardware platform, but adding new important facilities: an improved text preprocessor, a text analysis unit, a prosody generation module, and a PSOLA algorithm for diphone concatenation and speech signal generation. This TTS system architecture is presented in Figure 1; several main features are next briefly described.

- The preprocessor performs several basic operations: input text segmentation, punctuation marks detection and interpretation, replacement of some non-native Romanian letters with graphemes corresponding to their basic phonetic values, substitution of upper cases into lower cases, and most common numerals and abbreviations expansion using simple normalization rules.
- The letter-to-phone converter is based on a parallel neural network architecture. Starting from a basic set of 33 phones and an articulatory description of them, we use a number of 30 fully connected feed-forward neural networks. Each of them is associated to one articulatory characteristic and is capable of determining whether the phone associated with the current input grapheme has the corresponding articulatory features or not, based on a specific binary codification table. After training, the converter is able to provide at his output the complete phone string in the testing phase [2], [3].
- We used a minimal acoustic database consisting of 634 diphones; the segmentation from the speech corpus was manually performed. A two-step compression procedure for the diphone database has been used: down-sampling from 16 to 8 kHz and then compressing the resulted database with an *A-law* voice-coding scheme. Finally, the acoustic segments were labeled and stored in digital format.
- Syllabification, lexical stress positioning and duration assignment are performed using reduced set of rules.
- Based on a computed intonation model, as will be shown in the next section, the strings of labeled phones (together with prosodic information) are concatenated and processed by the speech generation stage according to the PSOLA algorithm.

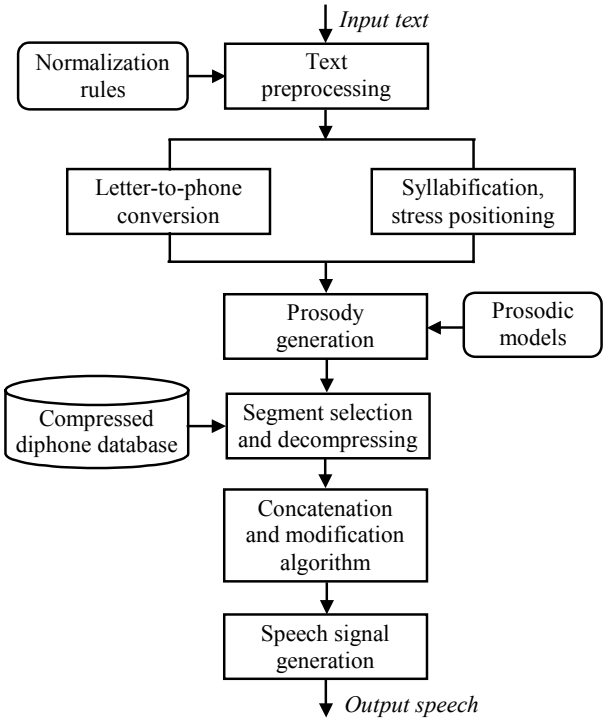

Figure 1 - The embedded TTS system architecture

2.2 Prosody modeling and generation

In a text-to-speech system, prosody generation (the F_0 contour, durations, pauses and intensity) is the intermediate step between text analysis and waveform generation. To generate the F_0 contour for synthesized utterances, one needs first to develop an adequate intonation model that can describe the F_0 behavior; then, an appropriate way is needed to predict model parameters and generate F_0 contours from input (high-level) linguistic information.

Many methods have been proposed to attain this goal, including rule-based and data-driven approaches. Rule-based approaches require sets of heuristic rules written by linguistic experts; this is more difficult to deal with, but can be very efficient and usually can produce consistent intonation contours due to explicit constraints. In contrast, data-driven approaches derive rules or some relations automatically by machine from a corpus; these models require a big amount of linguistic information and are generally dependent on the quality and quantity of available prosodically labeled corpora [9], [12].

Two well-known major approaches deal with the intonation modeling: the tone sequence one, which follows a traditional phonological description of intonation, and the (phonetically oriented) superposition approach [4]. But due to their complexity, both of them are difficult to be handled by a small-sized TTS system implementation (at least theoretically). The few commercial embedded TTS systems use simplified (or partially applied) intonation models [1], [7], [12], or are based on client-server architectures, performing the complete prosodic analysis at the (front-end) server-side [10], [11].

The previous ideas, together with our experience in building statistical models and also with the totally lack of prosody studies in Romanian suitable for TTS synthesis, conducted us to the decision of using a rule-based approach. The F_0 contour generated for a given utterance is described as the sum of a global component, related to the whole intonation unit, and of several local components related to accented syllables. For this research phase, our model is based on a minimal set of linguistic knowledge sources (available at the output of the text analysis module): word segmentation, phones and their corresponding durations, syllabification, lexical stress positions, punctuation information and utterance type (i.e., declarative, interrogative, exclamatory, or imperative). Because this subject will be described in details elsewhere, we only point out that a number of hand-derived rules are used in present to perform the syllabification, to locate the lexical stress, and to assign, for a relatively small number of situations, a sentential (word) stress, which will highlight the relative word prominence with respect to its neighbors.

A *target value* is assigned to each syllable, approximately in the middle of the vocalic part. We defined five levels for F_0 values with respect to the current range and used a simple expression to derive these values:

$$F_0(i) = F_{\min} + (F_{\max} - F_{\min}) \times i/4\,, \quad i = 0, 1, 2, 3, 4$$

where $F_{\min}$ and $F_{\max}$ are the approximate limits of the current F_0 range and can be easily deduced from several natural speech samples for the given speaker (an analogous idea is presented in [8]).

The five levels are mainly related to the following syllable conditions and key-points:

- *level* 1 ($i = 0$): end-point of declarative, exclamatory or imperative sentences; phrase end-point before comma;
- *level* 2 ($i = 1$): syllable without lexical or sentential stress;
- *level* 3 ($i = 2$): sentence start-point and phrase start-point after comma; syllable with lexical stress and without sentential stress;
- *level* 4 ($i = 3$): end-point of interrogative sentences; syllable with sentential stress and without lexical stress;
- *level* 5 ($i = 4$): syllable with lexical and sentential stress.

Finally, the F_0 contour is generated by connecting these target points through a simple linear interpolation. Because our TTS system use a PSOLA algorithm for segment concatenation and speech signal generation [2], at least one F_0 value is needed for each phone at the output of the prosody module. These values can be easily computed from the generated F_0 contour, based on the phone duration information.

As an example, Figure 2 illustrates the discussed procedure for the sentence "Nu se poate, chiar e adevărat?" (*It's not possible, it is really true?*).

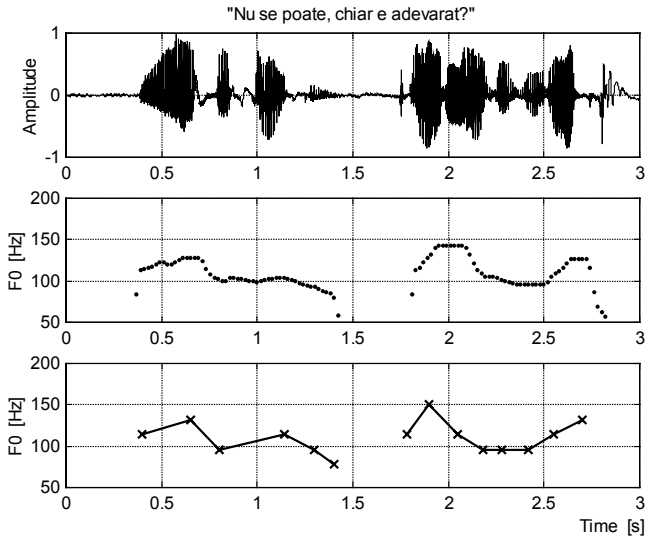

Figure 2 - From top to bottom: waveform, the natural F_0 contour, and respectively the computed target values and the modeled F_0 contour for the same sentence in Romanian.

For this particular (male) speaker, the F_0 range was between 78 Hz and 150 Hz; the annotated phonetic sequence (SAMPA-based) and the coordinates of the corresponding F_0 values are shown in the following.

" #|*nu#se#+po_Xa-te\#|*+k'0ar#+e#a-de-v@-+rat/# "

Time [s]: 0.40 0.65 0.80 1.14 1.30 1.40 1.78 1.90 2.05 2.18 2.28 2.42 2.55 2.70

F_0 [Hz]: 114 132 96 114 96 78 114 150 114 96 96 96 114 132

The prosodic marks used in our formalism and present in the previous example are: '-' for syllable boundary, '+' for lexical stress, '*' for sentential (word) stress, '|' for sentence start-points and phrase start-points after comma, '\' for phrase end-points before comma, '/' for end-point of interrogative sentences, and '#' for word boundary.

3. EXPERIMENTAL EVALUATION

3.1 Tests

Our main testing purpose was to evaluate the proposed intonation model (or, in other words, to investigate the accuracy of the F_0 generation procedure) and consequently to assess the quality of the generated speech, in the framework of the new version of the embedded TTS system.

Data from several speakers were collected, resulting in a large corpus. The speech material contains 370 sentences: professional Romanian speakers on national radio have uttered 320 sentences of various types, lengths and complexities, and 50 sentences (containing several frequently used phrases) were chosen from a locally developed database. The largest part (300 sentences of the corpus) was basically used to infer the model. The remainder (70 sentences) was used to test the TTS system.

A perceptual (formal listening) test was conducted to evaluate the generated speech quality of the complete PSOLA-based TTS system described in Section 2, compared to the older version (and reported in [3]: no generated prosody and a minimal concatenation scheme). We will denote these two versions as 'N-TTS' and respectively 'O-TTS'.

Two groups of 24 subjects each were selected for this experiment, all of them being unfamiliar with speech synthesis systems. First, 60 test sentences were synthesized using both N-TTS and O-TTS systems; the last 10 ones from the test set (speech utterances sampled at 16 kHz, PCM 16-bit) were used to provide to the listeners a reference for a "perfect speech signal". Then, during the test (which was completed in about ten days), the subjects were asked to listen to the synthesized sentences and rate the overall speech quality from 5 to 1 according to the following opinion scale: 5 – excellent (the reference samples), 4 – good (intelligible and natural), 3 – fair (intelligible, but requires hearing effort), 2 – poor (hard to understand), and 1 – bad (unclear, many words missing), with one digit after decimal point (e.g., 2.5). The subjects were told that the score should reflect the speech quality based on clarity and pleasantness. A few times, several subjects asked to listen twice a particular sentence before giving a score.

After the whole test was completed, the Mean Opinion Scores (MOS) for each of the 60 sentences were calculated; finally, a global MOS was calculated for both N-TTS and O-TTS, for each of the two groups of subjects. The results are presented in Table 1.

	MOS (Group 1)	MOS (Group 2
O-TTS	2.84	2.71
N-TTS	3.65	3.48

Table 1 - Evaluation of the two TTS systems

We need to mention that we were not particularly interested in the absolute values of the scores, but mainly in the relative quality improvement - if any. The detailed results of this test are summarized as follows.

- The tests showed clearly that there are obvious differences between the two types of synthesized utterances, and that the proposed prosody generation module (together with PSOLA synthesis) offers an improved naturalness for the synthesized speech.
- Even if the inventory database was compressed at 8 kHz, the intelligibility is still absolutely acceptable.
- Short sentences are less affected by some wrong positioned F_0 values than longer sentences; this is perfectly understandable, because they need less target points.
- Even if it is largely accepted that wrong prosody can cause speech to sound unnatural (or even unintelligible), most subjects perceived positively these situations, compared to no prosody at all (this fact was somehow surprising for us).
- Many of the low scores were obtained due the incomplete or inaccurate linguistic information offered by the current text analysis module; we still have problems with lexical stress assignment, and the present used rule set needs to be improved (in Romanian, the stress is free and variable; in most situations, it falls on one of the last three syllables, but it can also fall on any other syllable, depending on many linguistic factors).

3.2 Overall system's performance

All the main routines of the complete application (except the low-level ones, which are specific to the Motorola processor) are currently implemented in ANSI-C language and fixed-point arithmetic, in order to ensure easy portability on other similar platforms.

The final memory requirement figures are 692 kbytes for the acoustic database, 97 kbytes for the neural networks' coefficients, and respectively 172 kbytes for the complete code-size. Consequently, the total memory required for the program to run is around 960 kbytes (including the memory space for the temporary variables).

Our system is generally comparable in size and CPU with other similar implementations (see, for examples, *Flite* system [1], or the *microDRESS* system [7]). It would be also interesting to make a comparison in prosody modeling and speech quality, but unfortunately the cited references doesn't really provide details about these two subjects, even if they claim some sort of intonation modeling.

A final remark is that the present embedded implementation of the TTS system provides practically an unlimited vocabulary, is capable of synthesizing speech with an extremely good intelligibility, and support real-time synthesis of an input stream of about 100 characters long.

4. CONCLUSIONS

The paper has presented an embedded TTS system for the Romanian language and has introduced an efficient model for the F_0 contour generation. The experiments have demonstrated that using the proposed model one can obtain intonation contours in good agreement with the natural F_0 movements, even if the model cannot capture yet short-term prosodic variations and the prediction is not successful for some particular circumstances (e.g., comma presence, which can lead in Romanian to rising, as well as to falling pitch).

Preliminary experiments performed with the complete TTS system show that most of the wrong prosody situations, especially for long and complex sentences (which are not common, though, for most messages in mobile devices), are due mainly to incorrect lexical stress assignment. By using more accurate linguistic information, we expect further improvement for the current embedded implementation. Work on an improved text analysis module and a particular PDA (HP iPAQ) implementation is currently in progress.

REFERENCES

[1] A. W. Black and K. A. Lenzo, "Flite: A Small Run-Time Synthesis Engine", in *Proc. of The 4th ISCA Workshop on Speech Synthesis*, Perthshire, Scotland, Paper no. 204, 2001.

[2] D. Burileanu, "Basic Research and Implementation Decisions for a Text-to-Speech Synthesis System in Romanian", *International Journal of Speech Technology*, vol. 5, no. 3, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 211-225, Sep. 2002.

[3] D. Burileanu, A. Fecioru, D. Ion, M. Stoica, and C. Ilas, "An Optimized TTS System Implementation Using a Motorola StarCore SC140-Based Processor", in *Proc. of ICASSP 2004*, Montreal, Canada, vol. 5, pp. 317-320, 2004.

[4] Burileanu, D., *The Automatic Synthesis of Speech*. Bucharest: Printech Publishers, 2004.

[5] D. Burileanu, "Spoken Language Interfaces For Embedded Applications", in *Human Factors and Voice Interactive Systems* (D. Gardner-Bonneau, H. Blanchard – Eds.), 2nd Edition, Springer Publishing House, Norwell, 2006 (to appear).

[6] A. Di Cristo, P. Di Cristo, E. Campione, and J. Veronis, "A prosodic model for Text-to-Speech Synthesis in French", in *Intonation. Analysis, Modelling and Technology* (A. Botinis – Ed.), Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[7] R. Hoffmann, O. Jokisch, D. Hirschfeld, G. Strecha, H. Kruschke, U. Kordon, and U. Koloska, "A multilingual TTS system with less than 1 Mbyte footprint for embedded applications", in *Proceedings of ICASSP 2003*, Hong Kong, vol. 1, pp. 532-535, 2003.

[8] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. New Jersey: Prentice Hall, 2001.

[9] K. J. Kohler, "Parametric Control of Prosodic Variables by Symbolic Input in TTS Synthesis", in *Progress in Speech Synthesis* (J.P.H. van Santen et al. – Eds.), Springer-Verlag, New York, 1997.

[10] A. Monaghan, M. Kassaei, M. Luckin, M. Amador-Hernandez, A. Lowry, D. Faulkner, and F. Sannier, "Multilingual TTS for Computer Telephony: The Aculab Approach", in *Proc. of Eurospeech'2001*, Aalborg, Denmark, vol. 1, pp. 513-516, 2001.

[11] H. Sheikhzadeh, E. Cornu, R. Brennan, and T. Schneider, "Real-time speech synthesis on an ultra low-resource, programmable DSP system", in *Proc. of ICASSP 2002*, Orlando, USA, vol. 1, pp. 433-436, 2002.

[12] G. Xydas and G. Kouroupetroglu, "An Intonation Model for Embedded Devices Based on Natural F0 samples", in *Proc. of ICSLP 2004*, Jeju, Korea, pp. 801-804, 2004.