FlexVoice: A Parametric Approach to High-Quality Speech Synthesis

Gy. Balogh, E. Dobler, T. Grőbler, B. Smodics, and Cs. Szepesvári

Mindmaker Ltd., Budapest, Hungary
{bogyom,dobler,grobler,sbela,szepes}@mindmaker.hu
http://www.flexvoice.com

Abstract. FlexVoice, an integrated text-to-speech (TTS) system is presented in this paper. Its most distinctive feature is its low memory and CPU load while preserving the high quality of leading TTS systems. FlexVoice uses a hybrid approach that combines diphone concatenation with LPC-based parametric synthesis. Major improvements of speech quality are achieved by the careful design of each module at all synthesis levels (such as selection of training data for the various machine learning methods and that of the basic synthesis units for the parametric synthesizer). FlexVoice currently supports US English with two male and two female voices.

1 Introduction

The general idea of developing FlexVoice was to build a TTS system that has significantly lower memory and CPU time requirements than nowadays' high-quality text-to-speech systems, while preserving their intelligibility and naturalness. Time-domain concatenative synthesis has been judged recently as the method that can achieve the highest quality [1]. Time-domain methods, however, have some serious shortcomings, such as limited ability to modify pitch and duration values of segments, which leads to either a quality drop or huge segment inventories. FlexVoice uses concatenative synthesis from a parametric approach. This method has advantages over time-domain methods like easy prosody matching, smaller database, low computational load and large-scale voice manipulations.

2 General Structure

FlexVoice is a concatenative, diphone-based LPC synthesis system. It is a hybrid system in the sense that it concatenates fundamental units but segment concatenation takes place in a parametric space instead of the conventional wave space. This hybrid property ensures the small size and good quality of FlexVoice. FlexVoice has a modular structure. Modules perform diverse tasks but each module can be categorized as either language-dependent or language-independent. Language-independent modules can be re-used in languages other than English without modification.

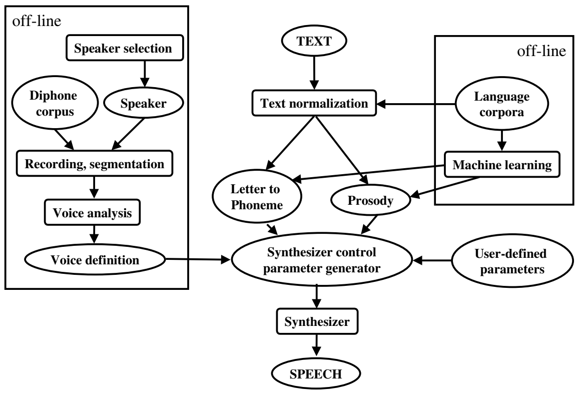

Fig. 1. Flow chart of the FlexVoice TTS system

3 Low-Level Modules and Procedures

Low-level modules are those that produce the final wave output from a sequence of phoneme, pitch, and duration values. All low-level modules are language-independent except the design procedure of the diphone corpus.

3.1 Speaker Selection

Speaker selection is a crucial point because all of the following steps may suffer from an inappropriate selection. Not only the voice properties but also the linguistic abilities of the speaker are rigorously examined. Though speaker selection is strongly subjective, there are some guidelines which should be followed. The speaker should

- be native speaker of the language (US English)
- have a "standard" dialect that is acceptable for the language community
- be able to articulate clearly and naturally at the same time
- have a pleasant voice when synthesized

Though these constraints are not easy to fulfill, they do not necessarily guarantee good results. A preliminary subjective listening test was constructed to select speakers from seven female and six male voices. For the test, a sentence was constructed that contained all phoneme types. The necessary diphones to synthesize the sentence were recorded from each speaker (approximately 50 diphones per speaker). A diphone is recorded as part of a sample word, and no sample word was allowed to be the same as in the test sentence. The test sentence was then synthesized in all speakers' voices, using both natural and synthetic prosody. Subjective evaluation was carried out by ten subjects to evaluate the quality of the synthesized voices.

3.2 Diphone Corpus

Extra care must be taken when designing the diphone database. This is even more true because all diphones are represented with a single word (or phrase) that is recorded only once. The phone alphabet used in FlexVoice contains 56 sounds. This is the adapted version of the SAMPA English system to US English, plus allophonic variations such as aspirated stops, syllabic consonants, diphthongs, etc. are included. Taking into account that some phone pairs are excluded by allophonic rules, all possible diphones have been looked for. The total number of diphones used in FlexVoice is about 2200. The remaining 900 diphones are theoretically possible but hardly ever appear in English speech.

3.3 Recording and Segmentation

Evidently, recording strongly influences the quality of synthesized speech. The analysis algorithms in FlexVoice turned out to be robust against noise and pitch fluctuations. Nevertheless, high quality studio recording cannot be omitted. Care must be taken to ensure that an appropriate dynamic range is used, the recording loudness is kept constant and side effects like pops and chirps are filtered out. Recording the whole diphone corpus takes approximately 5 hours.

After recording is done, the diphone positions should be marked in the corpus words. No automatic algorithm has been found which performed the segmentation task with the desired accuracy, segmentation is thus done semiautomatically: A specific tool has been developed to display the spectra and waveform of corpus words, helping the expert place the diphone markers. Three markers are used to mark a diphone, the first is inside the first phone, the second is at the border of the two phones, the third is inside the second phone. The exact positions of the first and last markers inside the phones are determined by the phonetic properties of the given sound to ensure smooth concatenation during synthesis, still considerable expertise is needed to place the markers correctly. Segmentation of a whole diphone corpus takes approx. 10 man-days.

3.4 Voice Analysis

Voice analysis is a process that creates a speaker-definition file from the segmented corpus. The model used in FlexVoice is a source–filter model. The goal of voice analysis is to separate the filter part from the source, to determine the filter parameters (spectral analysis), and to determine the amplitudes of the harmonic and noise parts of the source (source estimation).

Spectral Analysis The filter model of FlexVoice is an autoregressive (AR) model. The parameters of the all-pole filter are determined with the Levinson-Durbin algorithm. Experiments have shown no significant impact of pitch synchronous analysis, therefore Gauss-windows with constant window shift (6 ms) are used. The order of the all-pole filter is 20, and an additional normalization factor is computed for every filter. Filter normalization is a critical issue because the loudness values of different diphones coming from different sample words must be equalized. The pitch-dependence of the normalization factor is ignored; filter normalization can thus be an off-line process. The estimation of the normalization factor is done by measuring the energy of the synthesized wave.

Loudness The loudness values of different diphones coming from different corpus words may differ but these diphones should be concatenated smoothly during synthesis. FlexVoice specifies a loudness target value for each phoneme and diphones are equalized so that the loudness value at the beginning of the diphone equals the target value of the first phoneme while the loudness value at the end equals the target value of the second phoneme. Since filters are normalized, loudness equalization can be performed by modifying the source amplitudes only.

Source Estimation Source estimation determines the amplitudes of the harmonic and the noise sources. Since the overall loudness is determined by loudness equalization, the only task is to determine the harmonic/noise ratio. Some phoneme-specific constraints are also added to eliminate recording and estimation errors.

3.5 Synthesizer Control Parameter Generator

The task of this module is to create a series of synthesizer control parameters from the phoneme string given by high-level modules. This is done by concatenating analyzed diphones. Smoothing is performed at diphone boundaries. Source amplitudes are interpolated linearly while filter coefficients are converted to area parameter space, interpolated linearly, then converted back.

3.6 Synthesizer

The synthesizer is a modified version of the Klatt synthesizer [2]. The main differences are that FlexVoice uses linear filters instead of resonators, and uses a single filter (the vocal tract in [2], omitting the parallel tract). There are two sources for filter input: a harmonic (voicing) and a noise source, both normalized to have unit energy density. The voicing source is based on the one in [2] but can optionally be supplemented with a coloured noise component. The noise source emits Gaussian white noise.

4 High-Level Modules

The input of the high-level part is ASCII text, the output is a string of phoneme, duration and pitch values. The high-level part uses a multi-level data structure to represent the different linguistic, phonetic and prosodic information. Every high level module uses this data structure for both input and output. FlexVoice uses various machine-learning techniques as well as rule-based technologies for completing the high-level tasks. The main modules are described in this chapter in order of execution.

4.1 Text Normalization

FlexVoice uses a modified bottom-up chart-parser to solve the text normalization problem. A tokenizer analyses the input text and finds the word boundaries. The frequently used patterns are placed in a grammar that is applied on the output of the tokenizer. The parser may give multiple results with a given probability, therefore further analysis is requred to select the best solution. After a pattern is recognized, FlexVoice creates tokens representing parts that should be converted into words. There is a dictionary containing patterns that describe how to create words from tokens. This dictionary contains further instructions in order to help the extraction process. Each keyword can have different possible results called variants depending on context, style, or the speaker. The dictionary also contains dialect information to enable FlexVoice to differentiate between British and US English. With this technology, FlexVoice can handle the vast majority of text normalization problems.

4.2 Part-of-Speech Tagger

FlexVoice uses a Part-of-Speech (POS) tagger as the first step of linguistic analysis. The tagger is a simple n-gram language model which was trained on an untagged text corpus.

4.3 Word Prominence Computation

FlexVoice assigns a discrete-valued word prominence information for each word. The prominence class is computed from the POS tags using a table created by human experts.

4.4 Letter-to-Phoneme Conversion

FlexVoice uses a machine-learning approach for the letter-to-phoneme conversion. For training, a pronunciation dictionary is used that contains approximately 60,000 words with their American English phonetic transcriptions and word stress information. The phoneme and grapheme strings are aligned with a probabilistic model which uses a phoneme-to-letter transition matrix created by human experts to limit the search space. A C4.5 decision tree algorithm is used to predict the phoneme and stress information using features like context and history. The generalization performance of the model is 70% on word level and 93% on phoneme level. FlexVoice uses an editable pronunciation and POS-tag dictionary. Every word is looked up in this dictionary first, the decision tree is used only if the word is not found. The dictionary is also used for resolving the homograph ambiguity. Finally, there are some hand-driven rules for context-dependent word pronunciations such as “the” before consonants/vowels.

4.5 Prosody Models

The whole prosody generation module is based on neural networks. Four separate multi-layer perceptron models were trained with a variant of the back-propagation algorithm on a large annotated corpus: one for intonational phrase boundary prediction, one for phoneme duration prediction, and two for pitch contour prediction.

Phrase Boundaries FlexVoice splits sentences into intonational phrases that are separated with pauses in human speech. The neural network predicts a probability value for every word pair being a phrase boundary using features such as POS tags of the word context and punctuation information. For the yes/no decision for a word pair being a phrase boundary, a threshold is used on the network output. The analog output of the network is used to determine the duration of the intonational pause. The duration of the pause depends linearly on the output of the neural network.

Duration The neural network for duration predicts the z-score of the duration for every phoneme using features such as phonetic context and phrase break information. The generalization performance of the model is 0.59 (correlation coefficient) measured in the z-score space.

Pitch Contour Two separate neural networks are used for the pitch contour prediction. One for declarative sentences and one for questions. The networks predict a normalized pitch value for each phoneme using features such as phonetic context and POS information of the word context.

5 Conclusions

Subjective listening tests have shown that FlexVoice can produce high-quality synthesized speech that compares favorably with competing products, using only a fraction of resources of a nowadays' average PC. The current version, in its unoptimized state, uses 2 MB RAM per base voice and about 15 percent of CPU time of a PIII 450 MHz processor. In addition, a significant number of different manipulated voices can be derived from a base voice. Future improvements include, among others, the development of an automatic segmentation tool and improvement of the voicing source model.

References

1. Dutoit, T.: An Introduction to Text-To-Speech Synthesis. Kluwer Acad. Publ., Dordrecht (1997)
2. Klatt, D.H., Klatt, L.C.: Analysis, Synthesis, and Perception of Voice Quality Variations among Female and Male Talkers. J. Acoust. Soc. Am. **87** (1990) 820–857