Al-Quds University

Deanship of Graduate Studies

Master of Electronics and Computer Engineering

Thesis Approval

Diphone-Based Arabic Speech Synthesizer for Limited Resources Systems

Prepared By: Nuha Salem Ishtaiwi Odeh

Registration No: 20913062

Supervisor: Dr. Hanna Abdel Nour

Master thesis submitted and accepted, Date: 9/2/2013

The name and signatures of examining committee members are as follows:

1-	Head of committee	Dr. Hanna Abdel Nour	Signature: .....
2-	Internal Examiner	Dr. Ali Jamoos	Signature: .....
3-	External Examiner	Dr. Radwan Tahboub	Signature: .....

Jerusalem-Palestine

1434/2013

Abstract

This research deals with building an Arabic concatenative speech synthesizer for limited resources systems such as mobile applications. This is done by using a database of diphones stored in Code Excited Linear Prediction (CELP) coding format and predefined pitch contours. Diphones are chosen due to the limited necessary storage memory requirements on mobile and in the same time to the fact that diphones have the advantage of modeling coarticulation by including the transition to the next phoneme inside the diphone itself. Due to the problem of unbounded regions between diphones, this research proposes a method to smooth these regions by using a time domain overlap-add (TD-OLA) method. CELP coding scheme is used as an appropriate coding algorithm with appropriate memory requirements to cope up with a limited resources system. This algorithm is modified to be flexible for prosodic modifications. Prosodic features are implemented using linear pitch contour models. These pitch contours follow pitch variations in the affirmative and interrogative types of sentences in Arabic speech. So this research aims to build a complete offline Arabic speech synthesizer on mobile using: diphone-based concatenative synthesis, modified CELP algorithm for speech coding, TD-OLA for smoothing and linear pitch contours for prosodic features. Finally, all synthesis steps are implemented on mobile systems with Android platform under JAVA environment.

Keywords: Concatenative Speech Synthesis, Diphones, CELP coding, TD-OLA, Prosody, Pitch Contours, Limited Resources, Android.

Table of Contents

List of Figures

Introduction

In the past 25 years significant research has been undertaken in the field of Modern Standard Arabic (MSA) Text-to-Speech (TtS) algorithms development in terms of methodology, coding, and quality of produced speech. Zaki et al. [Zaki00] presented an Arabic TtS system based on synthesis by rules using formants and proposed rules for the generation of intonative contours for the interrogative sentences in order to improve the naturalness of the synthesized speech. Up to the knowledge of the writer, concatenative synthesis by diphones has become and by far the most used for unlimited vocabulary applications; the synthesis engine is language independent with an appropriate quality of speech. In 1992, Ghazali et al. presented a TtS synthesis system for Arabic where the synthesis engine included a dictionary of diphones and the algorithm used for the synthesis was Time Domain Pitch Synchronous Overlap Add (TD-PSOLA) for adjustment of pitch as well as duration when required [Ghazali92]. In 2002, H. Al-Muhtaseb et al. also proposed a diphone/sub-syllable method for Arabic TtS systems [Al-Muhtaseb02]. The proposed approach exploited the particular syllabic structure of the Arabic words. For good quality, the boundaries of the speech segments were chosen to occur only at the sustained portion of vowels.

To improve the quality of speech researchers have introduced prosodic parameters. This is done using models of pitch contours for different types of sentences of different lengths. In [Zaki01], the authors proposed a linguistic model to generate fundamental frequency (F0) contours using neural networks. In [El-Imam08], the author proposed pitch contours that were characterized by four attributes: fluctuations around a mean pitch value that lies

along either a declining or a constant line, a narrowing dynamic pitch range, an isochrony period between successive accented syllables, and a two-phase pitch rise in case of certain interrogatives.

The most widely commercialized system for MSA is Sakhr®[1]. But Sakhr TtS engine is mainly desktop application using huge database of recorded speech segments (mainly words) itself without any coding scheme. Sakhr can be a mobile application but any written text will be sent to a server to complete the procedure and then the server will send the synthesized speech to the mobile which means that Sakhr is not a standalone mobile application. Also, Sakhr engine does not take into account prosodic features of the text.

This research intends to go further by designing MSA TtS system by applying a certain number of constraints. The constraints have been identified as those of a limited resources mobile platform. This imposes a number of challenges concerning the unlimited vocabulary system, coding scheme of speech signals, memory requirements and prosody. In unlimited vocabulary system we need to have parts of speech stored as database. This database can be words, syllables, diphones or triphones. Because there are hundreds of thousands of different words and proper names in each language, storing parts such as complete words requires huge database of wav files. Although hardware resources of mobile systems are nowadays comparable to PCs, but this large database imposes the existing applications to work online. One of the known applications of TtS system for Android are iSpeech®[2] and Android TTS®[3]; they are online TtS applications. The challenge is to have an offline TtS application for MSA. In addition to the fact that using words requires huge database, words are not suitable units for any kind of Arabic unlimited

[1] http://www.sakhr.com/tts.aspx
[2] http://www.ispeech.org/api
[3] https://play.google.com/store/apps/details?id=com.ostrobar.tts

vocabulary TtS system because Arabic language is famous for its extremely large and complex vocabulary. As a result, using diphone-based database (with estimated number of diphones of 1700 in MSA) is more efficient for both memory requirements and unlimited vocabulary system challenges. But storing diphones as wav files without any coding scheme still requires large memory comparable with a complete offline Arabic TtS system for mobile. In this research a coding scheme is used; CELP coding. The CELP codec is to be modified in order to cope with a diphone-based speech synthesizer and in the same time have an acceptable perceived quality. This research introduces linear pitch contours models for MSA that carry important information relating utterance type (question vs. statement) which is very efficient in improving the naturalness of the system. CELP coding is also modified to adopt these models. Hence, the existing Android TtS applications don't deal with prosodic features.

In the first chapter we provide a general description of speech synthesis types focusing on diphone-based concatenative synthesis. Also, we present a method for smoothing between diphones boundaries called Time Domain Overlap Add (TD-OLA). In the last section we define prosody and give details about some pitch contours models for MSA. Our purpose in chapter 2 is to go in details with CELP coding algorithm describing its analysis-by-synthesis criteria, which consists of excitation generator, synthesis filter, error weighting filter and error minimization. Stochastic and adaptive codebooks for CELP will be presented in this chapter. Also, we present all the enhancements that we did on the CELP algorithm by deleting the pitch predictor for the unvoiced speech segment and adding the adaptive post filter to reduce the quantization noise. The last section summarizes the enhanced CELP algorithm parameters and how many bits are used to encode each parameter of the enhanced algorithm. In chapter 3, we introduce the Android platform, its operating system. In addition to, we present building the DSP library for speech synthesis

for Android and give an estimation of the required database size for our synthesizer and how it copes with the desired mobile memory specifications. Chapter 4 describes the challenges of this research and how each challenge is solved.

نظام تركيب الصوت للغة العربية بالاعتماد على المقاطع ثنائيات الأصوات للأنظمة محدودة الموارد

إعداد: نهى سالم شتيوي عودة

إشراف: د. حنا عبد النور

الملخص

يهدف هذا البحث إلى بناء نظام لتركيب الصوت للغة العربية باستخدام تقنية التركيب المتسلسل لغرض استخدامها في الأنظمة محدودة الموارد مثل تطبيقات الهواتف المحمولة. يتم هذا عن طريق استخدام قاعدة بيانات من المقاطع الصوتية المعروفة باسم ثنائيات الأصوات والتي تم اختيارها لأن حجمها يتلاءم مع ذاكرة الهاتف المحمول المحدودة بالإضافة إلى انه سيتم تخزين هذه المقاطع باستخدام ترميز CELP. تتميز تلك المقاطع الصوتية بأنها تحتفظ بالتداخل بين كل اثنين من الوحدات الصوتية الفردية داخل كل ثنائي أصوات. لكن وبسبب صعوبة الربط بين ثنائيات الأصوات، يقترح هذا البحث طريقة لتمهيد عملية الربط وضمان سلاستها تدعى التراكب والجمع في المجال الزمني. تم استخدام طريقة الترميز باستخدام CELP كخوارزمية مناسبة لتقليل حجم التخزين لتتلاءم مع الأنظمة محدودة الموارد ولأنها مرنة لتطبيق خصائص نغمات الصوت. يتم إدماج نغمات الصوت مع الإشارة الصوتية من خلال استخدام منحيات خطية لنغمة الصوت. تتبع هذه المنحنيات التغيير في نغمة الصوت في الجمل الاسمية والاستفهامية في اللغة العربية. أخيراً يتم تنفيذ كل خطوات تركيب الصوت على جهاز الهاتف المحمول باستخدام نظام الأندرويد ولغة البرمجة جافا.

Chapter 4

Conclusion

The aim of this research is to design and implement an offline Arabic Text-to-Speech Synthesizer into limited resources system such as mobile applications. In this research we suppose that the grapheme-to-phoneme conversion and the calculation of the pitch contour are ready for used. To achieve this aim several challenges must be solved. First, we want the system to be an unlimited vocabulary synthesizer of Modern Standard Arabic. For this we chose diphone-based concatenative synthesis. The diphones in Arabic do not exceed 1700. Unfortunately there is no open-source database for Modern Standard Arabic diphones, so our system was tested using diphones that form the sentence " هذا من مسؤوليات دافع الضرائب", which consists of a total of 33 diphones.

To create this database, 33 carrier sentences that contained these diphones were recorded in a professional studio of Al-Quds Educational Television by the voice of Dr. Hanna Abdel Nour. These sentences were recorded at sampling rate of 48 kHz. To abide by the standard of the CELP codec system, the first step was to down-sample the recorded sentences to 8 kHz. Then we extracted the required diphones by a well defined segmentation process, and using CoolEdit®[5] Software. The segmentation process took into consideration the spectral properties and perception of the transition between two

[5] http://www.softpedia.com/get/Multimedia/Audio/Audio-Editors-Recorders/Cool-Edit-Pro.shtml

consecutive phonemes, special segmentation rules for special phones like plosives, and the duration of each phoneme. The following figure shows the spectral view of part of the sentence that contains the diphone /مَ/ and how the diphone was extracted.

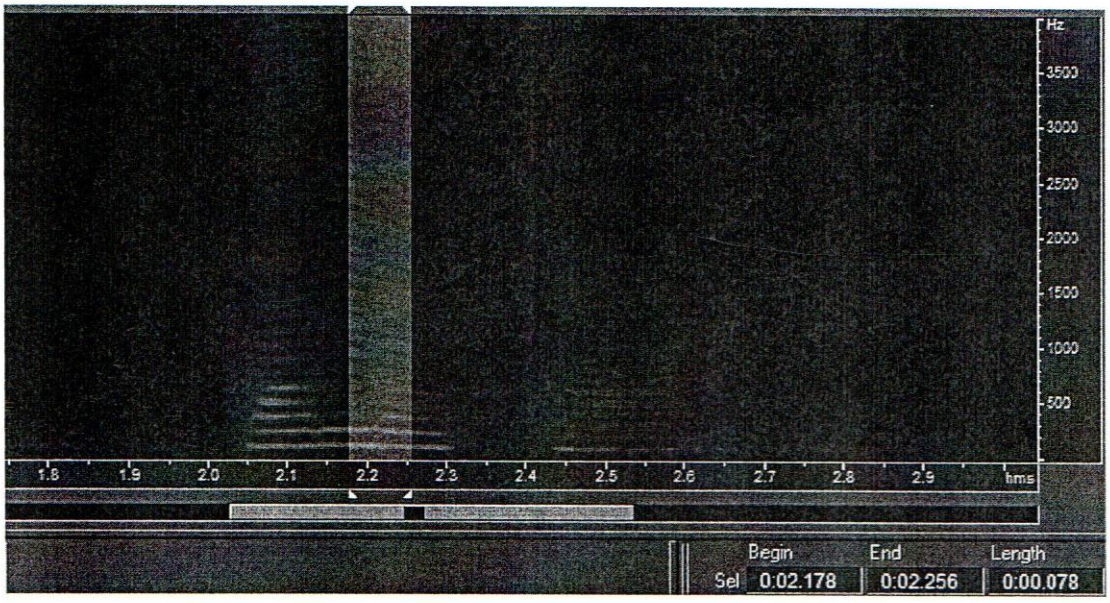

Figure 4.1: Extracting /مَ/ diphone from carrier sentence using CoolEdit®.

After creating a database of 33 wave files containing 33 different diphones for the desired sentence, the second challenge was to design an appropriate coding algorithm with minimum bit rate to cope up with a limited resources system. This algorithm must also be flexible for prosodic modifications and in the same time achieve acceptable perceived quality of speech. CELP coding is an appropriate choice for all these purposes.

Each wave signal representing a diphone had been analyzed to find its CELP parameters. A CELP analyzer with some modifications (for more details read chapter 2) was designed and implemented using MATLAB®[6] software. The inputs of the analyzer were:

1- The recorded diphone (.wav file).

2- The stochastic codebook which was a Gaussian noise of 1024x40 samples, each coded using a 16-bit word.

[6] www.mathworks.com

3- The weighted filter coefficient of 0.8. This value was chosen by listening tests for acceptable speech quality.

The output of the CELP analyzer for each one of the diphones was a text file that included the following CELP parameters:

1- Reflection coefficients, of 10 coefficients/frame.
2- The gain g, the codebook index k and the pitch predictor coefficient β (12 coefficients/frame).
3- Voiced/unvoiced bit index per frame.

The details of the number of bits that represent each coefficient are in Table 2.2.

Prosodic modifications were taken into account by implementing the linear pitch contours model that was designed by Nasser Eldin [Nasser Eldin03]. These pitch contours were designed for affirmative and interrogative types of sentences; short affirmative, short interrogative, long affirmative and long interrogative were implemented by estimating the pitch variations along the sentences depending on the length of the sentence and using the linear property of the model. For more details please read section 2 of chapter 1.

The third challenge was to design the CELP synthesizer to accept prosody modifications in addition to entries from the diphones database. The inputs of the synthesizer were: the same stochastic codebook that used in the analyzer, the CELP parameters for each diphone extracted from its related text file and the pitch variations along the sentence which were estimated by the implementation of pitch contours model depending on the type of the sentence. These input parameters were used to produce the synthesized signal according to the implementations of pitch predictor and synthesis filter.

The fourth challenge was to implement pitch contours and CELP synthesizer on mobile using Android platform. But for testing purposes, our design was first implemented using MATLAB® prior to implementing the same algorithm on mobile using JAVA®[7], and this was because MATLAB® has very good libraries for signal and speech processing.

The total synthesized speech must be modified to solve the problem of the discontinuities between diphones so a modified Time Domain Overlap Add (TD-OLA) algorithm was implemented to smooth the boundaries between diphones. In this algorithm, the synthesized signal was divided into overlapped segments of length 160 samples with overlapped regions of 80 samples of adjacent segments (40 samples each). Each segment was then multiplied by a Hamming widow of length 160 samples. Then these resultant segments were overlapped and added.

After the smoothing process, the resultant speech had some quantization noise that should be reduced to have an acceptable quality. So the adaptive post filter was also implemented (read chapter 2 section 3.4). The adaptive post filter parameters were chosen by listening tests as follows:

$\mu = 0.3$, $\alpha = 0.85$, $\beta = 0.6$

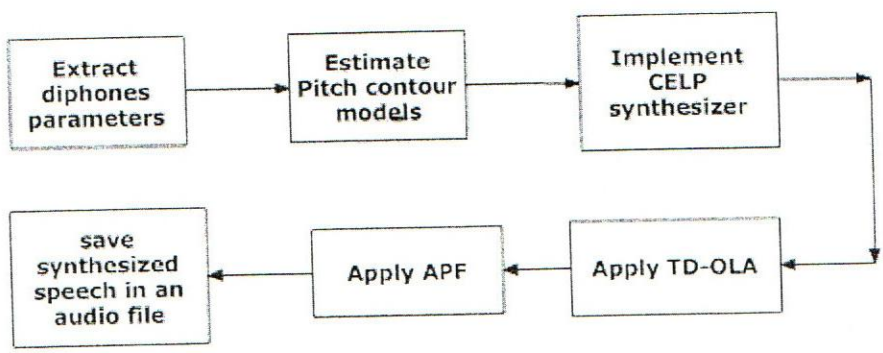

Figure 4.2: The synthesizer implementation steps.

[7] http://www.eclipse.org/downloads/packages/eclipse-ide-java-and-dsl-developers/junosr1

As this system is to be implemented on mobile, the synthesizer steps as shown in Figure 4.2 were re-implemented using JAVA® for Android platform. Each of the MATLAB® built-in functions were re-written in JAVA® by reference to the original purpose and equations of that function. This was due to the lack of Android libraries that supports speech processing. The results of each new function in JAVA® were carefully compared to its original MATLAB® built-in function. The resultant final speech was saved into a .wav file by using the audio library in JAVA® and after that it was ready to send to the speaker. The final interface of the system in mobile phone is shown in Figure 4.3.

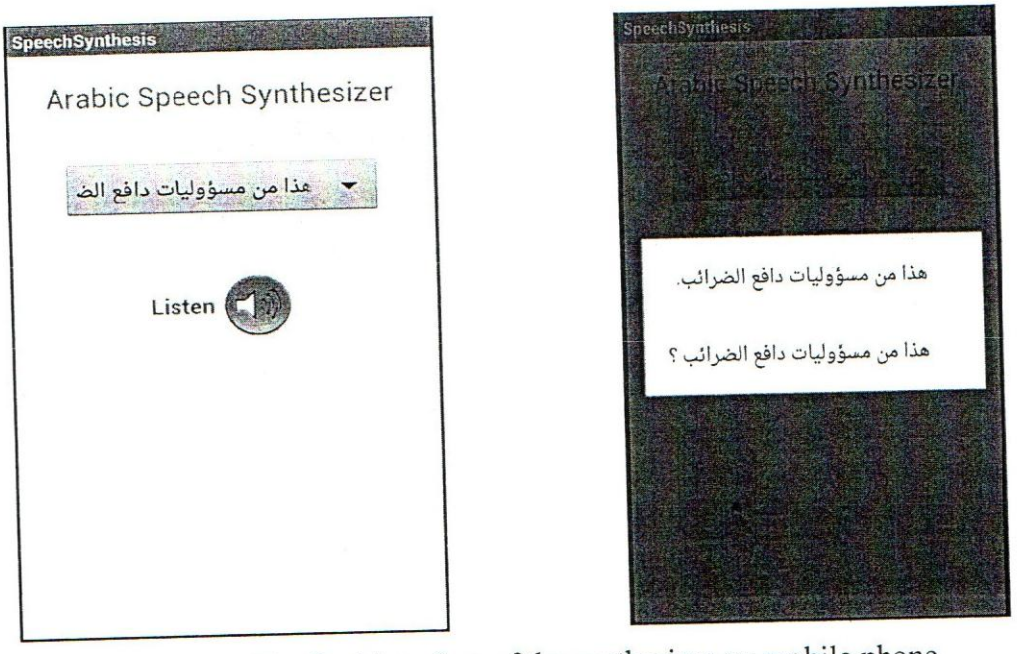

Figure 4.3: The final interface of the synthesizer on mobile phone.