Optimization of allophone database compression with wavelets for Polish speech synthesis TTS systems

Krzysztof Popowski, Edward Szpilewski

Institute of Computer Sciences
University of Bialystok, Sosnowa str. 64, 15-887 Bialystok, Poland
popowski@ii.uwb.edu.pl

Abstract

This paper presents a new optimized wavelet compression method for allophone databases for speech synthesis using various wavelet functions, effective quantization and coding. This approach allows obtain better compression results still achieving good quality of reconstructed speech signal. The article also presents several types of allophone databases, wavelets used for compression and a short introduction to new TTS systems that can use our encoded databases.

1. Introduction

In the present world of quick developing new information technologies the issue of change textual notation into voice signal which we can hear is very important. One of its main goal is development speech signal databases which containing necessary information to produce output voice that can be transparent to all listeners.

In our overall research we propose synthesis of phonemic characteristics of speech that based on the Allophones Natural Waves (ANW) method of speech signal concatenation [1]. For satisfaction of this process we should develop and create Natural Allophone Databases – two databases for one kind of voice: minimum and maximum. The destination of our Polish universal TTS system is produce three kinds of output voice: men, woman and men professional speaker. That is indicates so we need create six Natural Allophone Databases all containing over ten thousand of allophones.

In other hand we may want to develop TTS applications not only for PC computers but also for other platforms which perhaps don't have enough memory to store all databases (i.e. mobile phones). To resolve this problem we need to compress of our databases.

Several methods of signal compression have been proposed in the literature. In this paper we will work on wavelet compression. The first approaches of allophone database encoding using wavelets on which this new method is based was presented previously in [2] and [3]. The previous articles shows only one allophone database and its compression with only one type of wavelet function named 'Daubechies 10'. This paper describes database encoding optimization by using series types of wavelet functions (Daubechies, Symlet, Coiflet) and choose the best function to allophone's encoding.

Section 2 of this paper describes the new allophone databases creating process. Sections 3 and 4 describe the wavelet functions used to experiments and present encoding algorithm. Section 5 presents obtained experiments results. Findings and their implications are discussed in Section 6.

2. Polish allophone databases construction

Polish language alphabet consists of 32 letters of Latin alphabet:

- 23 usual letters: *A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, R, S, T, U, W, Y, Z*
- 9 letters from additional marks: *Ó, Ś, Ż, Ź, Ć, Ń, Ą, Ł, Ę*

Additionally from adopted words we can come across the letters: *Q, V, X* and we also use 7 digraphs: *CZ, DZ, DŻ, DŹ, RZ, SZ, CH*. Altogether we have 51 phonemes: 8 vowels and 43 consonant [4].

Phoneme in addition to its neighborhood is i.e. what phoneme is before him and after him can have different sound. Therefore one phoneme can have many variants so-called allophones.

To create allophone database first we should find the words list which contain target set of allophones. Further, these words should record without any emotions and then manually cut out suitable allophones and obviously record it under suitable name.

2.1. Recording conditions

The technical characteristics of the collected data were as follows:

- Microsoft wave format (.wav)
- one channel (mono)
- sampling frequency: 22050 Hz
- bits per sample: 16

For the recording of the three databases we additionally employed a male professional radio speaker. Thanks this we obtained three types of voice: two male voices (professional and unprofessional) and one female voice (unprofessional). Professional speech is necessary to profiles comparison with unprofessional voice type. The recordings took place in a professional sound radio studio. Recording the complete database for one type of voice (one speaker) took approximately 3 hours (with corrections). The whole database for one kind of voice contains around 2300 allophone files.

2.2. Resource partition

All the recording in each database were described by a set of label files. Next, the files were partite on groups which we receive two kinds of databases for each kind of voice:

- Minimum set database – 424 allophones
- Maximum set database – 2329 allophones

Thanks to minimum database we can build TTS system with minimal resources (memory, computations) and output speech sound have still quite good quality. Of course the best sound results we obtaining by working with maximum databases.

3. Wavelet functions and transforms

A wavelet is a mathematical waveform function of effectively limited duration that has an average value of zero. Wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet. Any signal can then be represented by translated and scaled versions of the mother wavelet [5].

Our application which should realize optimization of wavelet compression we using standard wavelet functions from three families:

- Daubechies: *db4, db5, db6, db7, db8, db9, db10*
- Coiflet: *coif1, coif2, coif3*
- Symlet: sym4, sym5, sym6, sym7, sym8, sym9, sym10

Following drawing represents several examples of *mother* wavelets and they associated scaling functions.

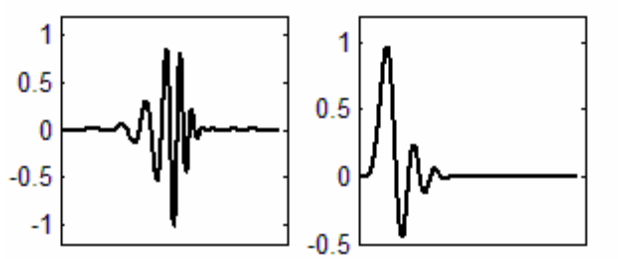

Figure 1: *Mother* wavelet *db10* and her scaling function

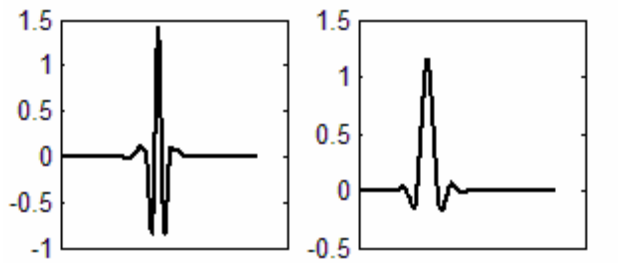

Figure 2: *Mother* wavelet *coif3* and her scaling function

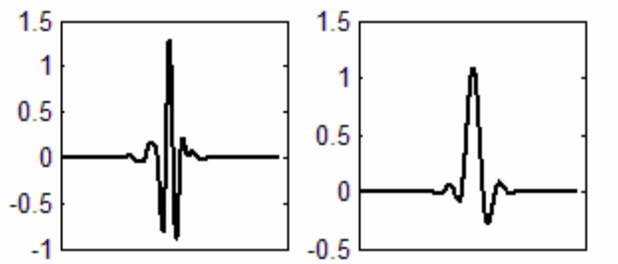

Figure 3: *Mother* wavelet *sym10* and her scaling function

Taking the wavelet transform of a speech signal we receive wavelet coefficients at different scales and translations. These coefficients represent the signal in the wavelet domain and all data operations can be performed using just the corresponding wavelet coefficients.

3.1. DWT – Discrete Wavelet Transform

The wavelet representation of signal x is a function of two variables; making up linear combination calibrated (parameter m) and shifted (parameter n) *mother* wavelet ψ and her scaling function φ:

$$f(x) = \sum_{m}^{L} \sum_{n} d(m,n)\psi(2^{-m}x - n) +$$

$$+ \sum_{n} a(L,n)\varphi(2^{-L}x - k) \qquad (1)$$

The wavelet function ψ is determined by the high-pass filter, which also produces the details of the wavelet decomposition. The scaling function φ is very similar to the wavelet function. It is determined by the low-pass quadrature mirror filters, and thus is associated with the approximations of the wavelet decomposition [6].

The detail coefficients at scale m and approximation coefficients at scale L can be expressed as:

$$d(m,n) = \frac{1}{\sqrt{2^m}} \int f(x)\psi(2^{-m}x - n)dx \qquad (2)$$

$$a(L,n) = \frac{1}{\sqrt{2^L}} \int f(x)\varphi(2^{-L}x - n)dx \qquad (3)$$

The Discrete Wavelet Transform (DWT) coefficients can be computed by using Mallat's Fast Wavelet Transform algorithm [7]. This algorithm is sometimes referred to as the *two-channel sub-band coder* and involves filtering the input signal based on the wavelet function used. This process is given in Fig. 4.

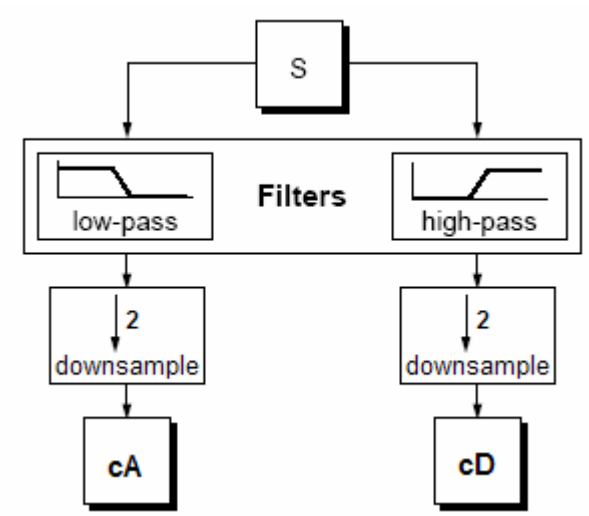

Figure 4: Single step of wavelet decomposition

Starting with a discrete input signal vector S, the first stage of the FWT algorithm decomposes the signal into two sets of coefficients. These are the approximation coefficients cA (low frequency information) and the detail coefficients cD (high frequency information). The coefficient vectors are obtained by convolving S with the low-pass filter for approximation and with the high-pass filter for details. This filtering operation is then followed by dyadic decimation or down sampling by a factor of 2.

If the length of each filter is equal to *2N* and the length of the original signal S is equal to *n*, then the corresponding lengths of the coefficients cA and cD are given by the formula:

$$floor\left(\frac{n-1}{2}\right)+N \tag{4}$$

This shows that the total length of the wavelet coefficients is always slightly greater than the length of the original signal due to the filtering process used. This leads to increase the size of data then compressed and we require better encoding methods.

3.2. DWPT – Discrete Wavelet Packet Transform

The decomposition process can be iterated. The approximation is then itself split into a second-level approximation and detail, and the process is repeated. For *n-level* decomposition, there are *n+1* possible ways to decompose or encode the signal. In wavelet packet analysis, the details as well as the approximations can be split. This is the *wavelet packet decomposition tree* (Fig. 5)

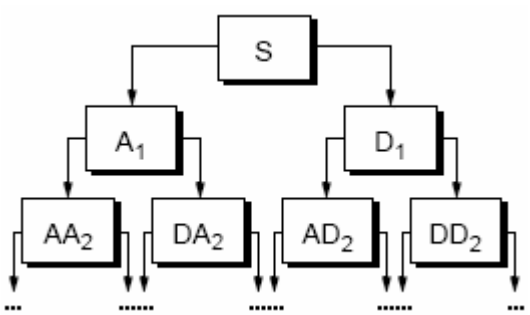

Figure 5: Multilevel wavelet packet decomposition tree

The wavelet decomposition tree is a part of complete binary tree with depth seven. The selection of the optimal decomposition level in the hierarchy depends on the nature of the signal being analysed or some other suitable criterion, such as critical bands, entropy. Every decomposition tree node contains wavelet coefficients which they proceeds form specified frequency band. Frequency bounds (in Hz) for particular bands can be computed form following formulas:

$$f_{\min}(z)=\left(z-2^{h}+1\right)\left(\frac{\frac{f_s}{2}}{2^{h}}\right) \tag{5}$$

$$\mathrm{f}_{\max}(z)=f_{\min}+\frac{\frac{f_s}{2}}{2^{h}} \tag{6}$$

where

z – tree node number,
h – tree node height,
f_s – signal frequency sampling.

3.3. ADWPT – Adaptive Discrete Wavelet Packet Transform [8]

In our project filter bank is created through adaptive tree of wavelet packet transform. Tree's maximum structure be defined as constant because signal analysis in bands, which spread is smaller than critical bands does not take any advantages. Directly in tree control algorithm was implemented psychoacoustic model to calculate of signal masking thresholds. Psychoacoustic model defines set of phenomena during analysis by human's sound system. In this place were used that issues form model like: threshold of absolute hearing, critical band idea (Bark scale) and simultaneous masking [9].

Adaptive decomposition tree steering algorithm, for a given allophone speech signal, can be presented by following steps:

1) **Decomposition tree *s* has only one node *z* (*root* node). We compute wavelet time entropy *WTE(s)* and perceptual entropy *J(z)*.**
2) **If node *z* is not on the list of indivisible nodes (in which also belong maximum structure tree terminal nodes) we split node *z* into two descendant (*s=s+1*) and compute individual entropies.**
   If tree height is greater than 7 → go to step 5.
   If *WTE(s) > WTE(s-1)* → go to step 5.
3) **If $J(z_i^s) < J(z_{2i-1}^{s+1}) + J(z_{2i}^{s+1})$ or $J(z_i^s) == 0$ then add node *z* into list of indivisible nodes.**
4) **Compute *WTE(s)* and *WTE(s-1)* and go to step 2.**
5) **Optimal filter bank has been appointed. End of working the algorithm for a given allophone.**

Tree structure extending on the basis of two entropy: perceptual (calculated from psychoacoustic model) (7) and wavelet time entropy (9).

Perceptual entropy is given by formula:

$$J\left(z_i^s\right)=\sum_{\forall\omega_{s,i}}\log_2\left(SMR\left(z_i^s\right)\right) \tag{7}$$

$$SMR=\left\lceil\frac{\left|\omega_{s,i}\right|}{\sqrt{c\frac{T(z_i^s)}{K(z_i^s)}}}\right\rceil \tag{8}$$

where

SMR – Signal-to-Masking Ratio

⌈ ⌉ - round to nearest integer number,

ω – wavelet transform coefficients,

$T(z_i^s)$ – masking threshold for node *z*,

$K(z_i^s)$ – number of coefficient for node *z*.

Wavelet time entropy is given by formula:

$$WTE(s) = \sum_k \frac{|\omega_{s,k}|}{\sum_j |\omega_{s,j}|} \cdot \ln\left(\frac{|\omega_{s,k}|}{\sum_j |\omega_{s,j}|} \right) \qquad (9)$$

where

ω – wavelet transform coefficients for scale *s*.

4. New optimized wavelet compression

Wavelet compression based lossy compression techniques have three steps in general:

- *Transform*: data are first transformed into wavelet domain (build optimal filter bank with ADWPT tree structure using optimal wavelet function).
- *Quantization*: the wavelet coefficients are quantized for minimize number of bits per sample (introduces quantization noise).
- *Effective Coding*: the resulting symbols after quantization are further coded with Variable Length Coding (VLC) and Right-Left Encoding (RLE) algorithms to reduce the bit rate.

Our allophone compression system has following scheme (Fig. 6):

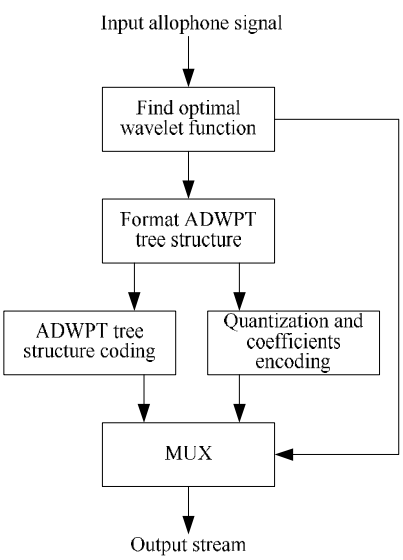

Figure 6: Scheme of encoding process

For every allophone signal from sound file we found the best optimal wavelet function. Next signal was subjected to adaptive wavelet packet transform and we obtain series of wavelet coefficients. Using psychoacoustic model was possible zeroing wavelet coefficients being under masking threshold level, that coefficients which they are inaudible for human sound system. Next, every wavelet coefficients and elements necessary to reconstruction particular allophone: number of wavelet function, number of tree nodes, each nodes number and each node coefficients amount were coded by VLC algorithm. Wavelet coefficient with values 0 was coded by RLE algorithm.

4.1. Choice of optimal wavelet function

The choice of the *mother*-wavelet function used in designing high quality speech coders is of prime importance. Choosing a wavelet that has compact support in both time and frequency in addition to a significant number of vanishing moments is essential for an optimum wavelet speech compressor [10]. Several different criteria can be used in selecting an optimal wavelet function. The objective is to minimize Normalized Root Mean Square Error (NRMSE), and maximize Compression Ratio. Following drawing (Fig. 7) shows all finding optimal wavelet function process.

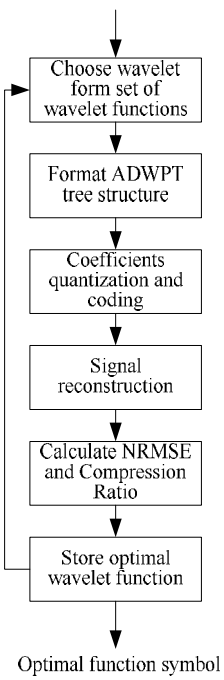

Figure 7: Scheme of finding optimal wavelet function

4.2. Compressed database format

Signal compression is achieved by truncating small-valued coefficients (replaced them with zeros), effective coefficients quantization with hiding quantization noise under masking level and then efficiently encoding them with VLC algorithm.

Another approach to compression is to encode consecutive zero valued coefficients by two values [11]. One value to indicate a sequence of zeros (4 bits code: '0000') in the wavelet coefficients vector and the second value representing the number of consecutive zeros (VLC encoded).

That formed codes and information necessary for signal reconstruction (also encoded) were wrote into a binary file, which represent allophones compressed database. Additionally, to increase finding given allophone process, also was created index file, containing name of given allophone, his offset in database file and length of data occupied by his wavelet representation.

4.3. Allophone reconstruction

Constructing decoder to such system already less complicated. From bit string we should extract information about ADWPT tree structure, used wavelet to encode and then reconstructed wavelet coefficients, insert into suitable tree node. Next we give an invert wavelet transform and on the exit we receive reconstructed signal of the allophone (all process as shown in Fig. 8).

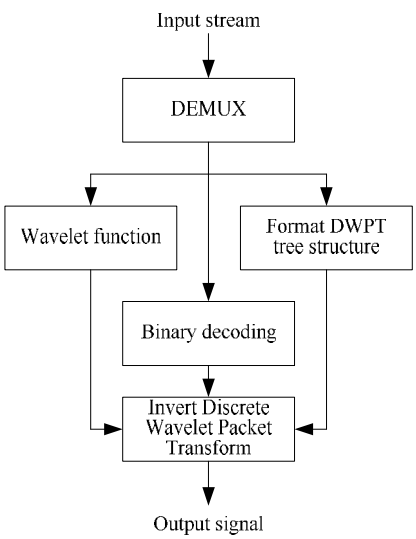

Figure 8: Scheme of decoding process

5. Discussion

To preliminary tests above efficient of coding algorithm compressed one size-kind of Natural Allophone Databases - minimum database for three voice-kinds of voices: male professional, male amateur and female. Summary are three databases which altogether contains over 1250 *.wav files.

To compute performance of our encoding system we use three measures: Normalized Root Mean Square Error (NRMSE), Signal to Noise Ratio (SNR) and Compression Ratio (CR) which are given by following formulas [12]:

$$NRMSE = \sqrt{\frac{\overline{(x(n)-r(n))^2}}{(x(n)-\mu_x(n))^2}} \tag{10}$$

where

$x(n)$ – input allophone's speech signal,
$r(n)$ – reconstructed allophone's signal,
$\mu_x(n)$ – mean of the $x(n)$.

$$SNR = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \tag{11}$$

where

σ_x^2 – mean square of the input allophone's signal,
σ_e^2 – mean square difference between the original and reconstructed allophones signals.

$$CR = \frac{sizeof(x(n))}{sizeof(cDATA)} \tag{12}$$

where

$cDATA$ – compressed allophone's signal.

One allophone database contains allophones from 51 phoneme groups. Percentage of zeroing wavelet coefficients are joins directly with dynamic of data signal and have main influence on compression ratio. Total, were compressed six allophone databases and for obtained results were computed average values for each of 51 groups. Several fragments of the above values for the exampled allophone groups are shown in Table 1.

Table 1: Average results for selected phoneme groups

Phoneme group	NRMSE	SNR	CR	Best wavelet
A	0,28	11,00	1,93	db10
B	0,41	8,05	5,74	sym7
Bi	0,47	7,70	5,85	db8
C	0,13	17,61	2,28	db4
Ci	0,10	20,82	1,74	db6
Cz	0,09	22,21	3,03	sym10
D	0,23	14,29	5,28	sym6
Di	0,34	9,46	3,36	sym10
Dz	0,42	7,51	2,22	sym9
Dź	0,32	10,38	1,95	sym7
Dż	0,35	10,60	2,47	db7

Compression results and specific features of each allophone database are shown in the following Table 2.

Table 2: Encoding results for each allophone database

Database name	Files	DB size [MB]	Encoded DB size [MB]	CR %
DB1min	424	2,85	0,77	73%
DB2min	424	2,72	0,68	75%
DB3min	424	2,43	0,58	76%

For present results clearly, databases were named as follows:

- DB1min – minimum allophone database for male professional speaker voice.
- DB2min – minimum allophone databases for male amateur speaker voice.
- DB3min – minimum allophone databases for female amateur speaker voice.

We obtained average reduction size of the allophone databases about 75% still keeping very good quality of reconstructed signal. Using different wavelet functions types shows that for vowel and consonant phonemes groups optimal wavelet functions are not from same families. For encoding vowels the optimal wavelet family is *Daubechies* and from it most often *db10* wavelet. For consonants phoneme groups optimal are wavelet functions from *Symlet* family, especially *sym10*.

6. Conclusions and Future Work

In this paper we presented the new method of allophone databases wavelet compression for Polish TTS application. Using different types of wavelet functions the compression ratio can be easily varied, while most other compression techniques have fixed compression ratios. However additionally, time of one allophone compression increase significantly because finding optimal wavelet for each allophone signal must compare results obtained from 17 wavelet functions and then, from best results selects the optimal solution. The important advantage is there, so we need only one execute of allophone database compression and TTS system will be work correctly.

Natural Allophone Database creation is a more complicated and more time-consuming process than compression. It happens because each allophone has to be cut out and properly recorded, that is why the recording has to be repeated several times in order to extract a single allophone. Therefore, the next step-in the research is working out a method of automation of the entire process. To achieve this, we need to have a well-prepared standard allophone base which would be a starting point for other databases.

Presented method of wavelet compression used to create binary compressed databases has several advantages. Most importantly are obtaining properly small database size than her original size and joining all natural allophone wave-files into one compressed file. Additionally, creating the index file considerably increase speed of finding suitable encoded allophone, then decoded and played. Usage of the compressed databases is very simple. Build a decoder for this system as small C library that implements methods to accessing and getting information from database engine is very easy.

First step of further work is extension of existing Polish TTS (described in [13]) application about working with multi-voice wavelet compressed databases.

Second important step in future work are design and implementation new Polish TTS system working on mobile devices. Some kinds of mobile devices like mobile phones have limited memory joined with not powerful CPUs. This is very good area to use wavelet transformation and compressed databases. Applications working on mobile phones should be implemented in J2ME technology. Works over such system was started lately. Extension of TTS applications into new areas like mobile devices will be new trends in evolution of such systems.

7. Acknowledgements

This paper was supported by the European Commission under grant INTAS Ref. number 04-77-7404. The author would like to express thanks for their support.

8. References

[1] Lobanov B., Karnevskaya H., *MW Speech Synthesis from Text*. Proc. of the XII International Congress of Phonetic Sciences. Aix-en-Provense, France, 1991, pp.406-409.

[2] Popowski K., Rafalko J., Shpilewski E. *The wavelet compression for speech synthesis,* Proceedings of the 8th International Conferece PRIP'2005, Minsk, Belarus, 2005, pp. 54-58.

[3] Popowski K., Rafałko J., Piórkowska B., Szpilewski E., *Using wavelet compression of allphone database for Polish language TTS system,* Proceedings of the 10th International Conference SPECOM'2005, Patras, Greece, pp. 305-308.

[4] Dłuska M., *Fonetyka polska*, PWN Warszawa – Kraków 1981.

[5] Graps A., *An Introduction to Wavelets*, IEEE Computational Sciences and Engineering, Summer 1995, vol. 2, num. 2.

[6] Białasiewicz J. P., *Falki i Aproksymacje*, WNT, Warszawa, 2000, pp. 253.

[7] Misiti M., Misiti Y., Oppenheim G. and Poggi J., Matlab Wavelet Tool Box, The Math Works Inc., 2000.

[8] Petrovsky Al., Petrovsky A.A., *Audio coding with a masking thresholds adapted wavelet packet based on run-timeconfigurable processor architecture*. 110th AES Convention preprint no 5298, Amsterdam, May 2001.

[9] Painter T., Spanias A., *Perceptual coding of digital audio.* Proceedings of the IDEE, vol. 88, no. 4, April 2000, pp. 451-513.

[10] Kastantin R., Stefanoiu D., Feng G., Martin N. and Mrayati M., *Optimal Wavelets for High Quality Speech Coding*,. *Proc. Of EUSIPCO-94,* European Association for Signal Processing, Edinburgh, Scotland, 1994, pp. 399-402.

[11] Kinsner W. and Langi A., *Speech and Image Signal Compression with Wavelets*, *IEEE Wescanex Conference Proceedings*, IEEE, New York, NY, 1993, pp. 368-375.

[12] Fgee E. B., Phillips W.J., Robertson W., "Comparing Audio Compression using Wavelets with other Audio Compression Schemes". *IEEE Canadian Conference on Electrical and Computer Engineering,* IEEE, Edmonton, Canada, 1999, pp. 698-701.

[13] Shpilewski E., Piurkowska B, Rafalko J., Lobanov B, Kiselov V., Tsirulnik L., *Polish TTS in Multi-Voice Slavonic Languages Speech Synthesis System*, Proceedings of the 9th International Conference SPECOM'2004, Saint Petersburg, Russia, 2004, pp.565-570.