SCIEMCEE
PUBLISHING LONDON

VILIAM ĎURIŠ - VLADIMIR I. SEMENOV - SERGEY G. CHUMAROV

2021

APPLICATION OF CONTINUOUS FAST WAVELET TRANSFORM FOR SIGNAL PROCESSING

SCIEMCEE
PUBLISHING LONDON

APPLICATION OF CONTINUOUS FAST WAVELET TRANSFORM FOR SIGNAL PROCESSING

Viliam Ďuriš - Vladimir I. Semenov - Sergey G. Chumarov

London 2021

Application of continuous fast wavelet transform for signal processing

Published by Sciemcee Publishing.

LP22772, 20-22 Wenlock Road
London, United Kingdom
N1 7GU

Sciemcee Publishing is part of SCIEMCEE.
It furthers the SCIEMCEE's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

First Edition: 2021

A catalogue record for this publication is available from British Library.

ISBN 978-1-9993071-9-6

Includes bibliographical references and index.

CONTENT

PREFACE

Any signal can be decomposed into different bases. The most well-known and familiar decomposition of a signal by sine and cosine is called harmonic (frequency) analysis. When the signal is decomposed by harmonic functions, the basis sequences are nonzero at all points and the expansion coefficients are proportional to the amplitudes of the sines and cosines. In this sense the basis is "global". The basis of sinuses and cosines is best for frequency analysis of signals, that is, for the accurate determination of frequency, and worst for determining time. If the basis sequences differ from zero only strictly at one point and the expansion coefficients are equal to the values of the original signal, then this expansion is "strictly local". Such a basis consists of functions in which all points are zero except for one equal to the Dirac delta function, and for each subsequent delta function is shifted by one count. Such a basis is best for the time analysis of signals, that is, for the exact determination of time, and worst for the determination of frequency. In the wavelet decomposition of a signal, the basis is "local" in the sense that the coefficients of the decomposition are nonzero at several points in the basis sequence. The wavelet basis occupies an intermediate position between the strictly local basis of delta functions and the global Fourier basis. The basis functions, called wavelets, have the property of time-frequency localization, and the analysis itself is called time-frequency. In classical Fourier analysis, frequencies are well distinguished, but there is no information about the time of occurrence of these frequencies in the case of a non-stationary signal. The basis of the Dirac delta function is very good at highlighting the time of appearance of the signal, but it does not highlight the frequency, that is, it has a uniform frequency spectrum. During the wavelet transform, both the frequencies and the time of occurrence of these frequencies are distinguished. Wavelet analysis is an effective tool for studying the local properties of signals for non-stationary signals with rapidly changing local frequencies. The frequency analog for the wavelet transform is the inverse of the scale factor. In order to cover all possible time positions of the signal with the basis functions, it is necessary to shift the basis functions along the time axis, that is, it is necessary to calculate the correlation of the signal with the wavelet, and this requires a lot of time. For real-time signal processing, algorithms for fast numerical calculation of the forward and inverse continuous wavelet transform are required, so the development of such algorithms is relevant. It is only possible to develop fast algorithms for the continuous wavelet transform using the fast Fourier transform. In this regard, in the monograph, in the first chapter, in addition to the Proni method, the wavelet transform and the Fourier transform are considered. In the second chapter, the principles and algorithms for calculating the forward and inverse continuous wavelet transform in the frequency domain using the fast Fourier transform, an algorithm for multiple-scale analysis of a signal in the frequency domain are considered. The third chapter indroduces application of algorithms for numerical calculation of fast continuous wavelet transform for speech recognition in Russian and the fourth chapter provides examples of using these algorithms to compress one-and two-dimensional signals, determines the average size of micro-and macro-objects in an image, and compares the results of multi-scale image analysis in the frequency domain with the results of multi-scale analysis in the MatLab computer mathematics system.

In case of a deeper interest, we refer the readers to references at the end of a book, for easier, more accurate and faster orientation in the given topic.

Authors

INTRODUCTION

Currently, various types of transformations are used to process stationary and non-stationary signals. Among them, the most commonly used are the Fourier transform (FT), the Proni transform, and the Wavelet Transform (WT). The FT decomposes a complex signal into many simple signals and determines the proportion of each simple signal in this complex signal, that is, it performs a spectral analysis. In stationary signals, the signal spectrum does not change over time, and FT is traditionally used to study such signals. The Proni transform and the WT are generalizations of classical spectral analysis, that is, the signal is also decomposed into simple components, but in different bases. For the study of non-stationary signals, the Proni transform and WT are better suited, where the basis functions for the decomposition are not infinite sines and cosines, as in the case of FT, but functions localized in space or in time. These transformations make it easier to extract information from the signal, identify features, and track the change in the frequency composition of the signal over time. Until the 1950s, signal processing in radio engineering was usually performed using analog devices. The use of digital signal processing was promoted by the development of large integrated circuits, the associated reduction in the cost, and size of digital devices, increasing their speed, and the creation of new efficient algorithms for numerical calculation of FT, Proni and wavelet decomposition. For the analysis of signals with the use of electronic computers, discrete analogs of these transformations are used. In turn, continuous signals from the analog form are converted into a discrete form, quantized, and entered into the computer in the form of digits. Digital signal processing technologies are increasingly being used in various fields of science and technology. The analog of FT in digital processing is the discrete Fourier transform (DFT) and its variant - the fast Fourier transform (FFT). The widespread use of the Proni transform for short attenuated signals in seismic exploration became possible with the advent of more powerful computers, which contributed to a better spectral resolution of these signals. A discretized version of the continuous (integral) WT and discrete WT (DWT) is used for signal processing on a computer. The DWT is not a discretized version of the continuous wavelet transform. If the continuous WT uses wavelet shifts with any arbitrarily small step, then the

DWT is based on the use of integer shifts and setting the scales of the power of two. Each transformation has its own advantages and disadvantages, and it is used for the spectral analysis of one-and two-dimensional signals.

1 CLASSICAL AND GENERALIZED SPECTRAL ANALYSIS OF SIGNALS

1.1 Fourier transform

Methods for analyzing signals in the frequency domain are widely used since they allow us to effectively use the properties of signals based on the mathematical apparatus of the FT [3, 4, 58, 61]. If the theory of signal processing was limited to the time approach, it would never have received such rapid development. The theory of FT since the 1920s has become a powerful theoretical basis for the development of radio engineering, radio electronics, electrical engineering and other fields of science and technology. The feasibility of switching to the frequency domain is also associated with the search for a variety of DFT-FFT, which reduces the signal processing time by many times with a large sample [4, 25, 58, 64]. To move from classical spectral analysis to generalized spectral analysis of FT, it is necessary to know the theory of functional analysis, the Laplace transform, and the z-transform as a generalization of the discrete Fourier transform. These sections are not included in the monograph.

If a function $f(t)$ integrable with a square has a period of 2π and is piecewise monotone and bounded in the interval $[-\pi, \pi]$, then its Fourier series converges to the function $f(t)$ at each point of continuity

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nt + b_n \sin nt)$$

and to the value

$$\int (t) = \frac{f(t+0) + f(t-0)}{2}$$

unevenly at the points of discontinuity (Gibbs phenomenon) [3, 58, 61].

The constants a_n and b_n are called Fourier coefficients and are determined by the Euler-Fourier formulas [3, 14, 61]

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos nt \, dt$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin nt \, dt$$

$n = 1,2,3 \ldots$

Both in the theory of Fourier series itself and in other areas of analysis, the closure equation finds numerous applications [61]

$$\frac{a_0^2}{2} + \sum_{m=1}^{\infty} (a_n^2 + b_n^2) = \frac{1}{\pi} \int_{-\pi}^{\pi} f^2(t) \, dt$$

For functions integrable with a square, this equation was first proved by A. M. Lyapunov [61].

So, if two functions $f(t)$ and $y(t)$ are given, integrable in the interval $[-\pi, \pi]$ with a square, having Fourier coefficients a_n, b_n and α_n, β_n, respectively, the generalized closure equation applies [61]

$$\frac{a_0 \alpha_0}{2} + \sum_{m=1}^{\infty} (a_n \alpha_n + b_n \beta_n) = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \, y(t) dt$$

These relations are called Parseval's formulas [61].

The function $f(t)$ can be defined in any interval: in a very wide range of cases, a function arbitrarily defined in an arbitrary interval turns out to be decomposable into a trigonometric series, i.e. the function is represented by a single analytical expression – a trigonometric series – in the entire domain of the function definition. The apparatus of trigonometric series turns out to be a universal tool for "gluing" functions, finally blurring the line between functions that allow a single analytical expression in the entire domain of definition and functions defined using several analytical expressions [3, 61].

The complex form of writing the Fourier series of the function $f(t)$ with the period T is often used

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t}$$
$$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{-i\frac{2\pi n}{T}t}\,dt, n = 1,2,3 \ldots \tag{1.1}$$

where the cosines and sinuses are replaced by Euler's formulas. The Fourier series of a periodic function consists of terms containing harmonic functions with discretely varying frequencies $\nu_0 = \frac{1}{T}, 2\nu_0, 3\nu_0, \ldots,$, i.e. the spectrum of the periodic function is linear [3, 56, 58, 61].

In nature, not all phenomena are periodic. Let us now consider the question of decomposition into the spectrum of non-periodic processes. Let there be a non-periodic absolutely integrable function $f(t)$. Such a function can be represented as periodic, reflecting a process with an infinitely large period. Denote $\frac{n}{T}$ by ν, replace temporarily t with α in formula (1.1), and substitute in the Fourier series

$$f(t) = \sum_{-\infty}^{\infty}\left[\int_{-\frac{T}{2}}^{\frac{T}{2}} f(\alpha)e^{-i2\pi\nu\alpha}\,d\alpha\right]e^{i2\pi\nu t}\,\Delta\nu$$

Changing the frequency

$$\Delta\nu = \frac{n+1}{T} - \frac{n}{T} = \frac{1}{T}$$

When the period T tends to infinity, $\Delta\nu$ tends to zero. Then, as proved in mathematical manuals [3, 14, 27, 56], the sum turns into an integral

$$f(t) = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(\alpha)e^{-i2\pi\nu\alpha}\,d\alpha\right]\sum_{-\infty}^{\infty}\left[\int_{-\frac{T}{2}}^{\frac{T}{2}} f(\alpha)e^{-i2\pi\nu\alpha}\,d\alpha\right]e^{i2\pi\nu t}\,\Delta\nu$$

that is, for a non-periodic function, the Fourier series turns into the Fourier integral. In contrast to the Fourier series, in the Fourier integral, the frequencies change continuously, so we have a continuous spectrum [3, 14, 27, 56, 61].

We write the transformation of the function $f(t)$ in the form

$$f(t) = \int_{-\infty}^{\infty} F(\nu)\, e^{i2\pi\nu t} d\nu \tag{1.2}$$

where

$$F(\nu) = \int_{-\infty}^{\infty} f(\alpha) e^{-i2\pi\nu\alpha}\, d\alpha$$

Returning to the previous notation, i.e. replacing α with t, we get

$$F(\nu) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi\nu t}\, dt \tag{1.3}$$

The function (1.3) is called the frequency spectrum or spectral density of the signal (function) $f(t)$. Expressions (1.3) and (1.2) are called the forward and inverse FT of the signal $f(t)$, respectively. At break points

$$f(t) = \frac{f(t+0) + f(t-0)}{2}$$

Thus, the signal can be described both in the time domain and in the frequency domain. Both representations correspond to each other unambiguously $f(t) \leftrightarrow F(\nu)$.

The basic properties of the Fourier transform can be summarized as follows. Denote the WT of the function $f(t) F(\nu) \leftrightarrow FF\{f(t)\}$.

1. The linearity property

$$FF\{af_1(t) + bf_2(t)\} \leftrightarrow aF_1(\nu) + bF_2(\nu)$$

for the functions $f_1(t)$ and $f_2(t)$ and any constants a and b.

2. Shift theorem

$$FF\{f(t-\xi)\} \leftrightarrow \exp(-i2\pi\nu\xi)\, F(\nu)$$

The shift of the signal in the region of the independent variable causes a phase change proportional to the frequency value of each spectral component of the signal.

3. Re-performing the Fourier transform

$$FF\{F(\nu)\} \leftrightarrow f(-t)$$

restores the original signal with the sign inversion of the independent variable.

4. The derivative theorem

If $FF\{f(t)\} \leftrightarrow F(\nu)$ then $FF\{d^n f(t)/dt^n\} \leftrightarrow (i2\pi\nu)^n F(\nu)$

5. The properties of parity and odd.

If $F(\nu) = F_c(\nu) - iF_s(\nu)$ in the case when $f(t)$ is an even function, $F(\nu) = F_c(\nu)$ is an even function; if $f(t)$ is odd $F(\nu) = F_s(\nu)$ is an odd function.

6. Similarity property

$$FF\{f(at)\} \leftrightarrow (1/|a|)F(\nu/a)$$

where a is a constant.
When a signal is compressed by a factor of a $(a > 1)$ on the time axis, its spectrum will expand by the same factor on the frequency axis, and the spectral density modulus will decrease by a factor of a. When the signal is stretched in time $(a < 1)$, its frequency spectrum is compressed and the spectral density modulus increases.

7. Energy conservation:

$$\int_{-\infty}^{\infty} f^2(t)\,dt = \int_{-\infty}^{\infty} |F(\nu)|^2\,d\nu$$

(Parseval's equality)

1.1.1 Discrete Fourier transform

A discrete signal $f(t_k) = f_k$ can be obtained from a continuous $f(t)$ by taking samples at certain moments t_k. The time interval between two adjacent samples is called the sampling step T_d. To convert a signal into a digital form, in addition to sampling, it is also quantized by level.

The sampling process can be considered as the multiplication of a continuous signal $f(t)$ by a periodic sequence of delta functions [58, 64]

$$s_d(t) = \sum_{k=-\infty}^{\infty} \delta\,(t - kT_d)$$

The resulting signal is called the lattice function $f^*(t)$ (hereafter, the symbol $*$ does not mean the complex conjugation operation)

$$f^*(t) = f(t)s_d(t) = \sum_{k=-\infty}^{\infty} \delta\,(t - kT_d)$$

If the continuous signal $f(t)$ has a bounded spectrum $F(\nu)$, then the sampled signal $f^*(t)$ has a spectrum

$$F^*(\nu) = \frac{1}{T_d}\sum_{k=-\infty}^{\infty} F\left(\nu - k\frac{1}{T_d}\right)$$

That is, the spectrum of a discrete signal is the sum of shifted copies of the spectrum of a continuous signal. The copies are located on the frequency axis at the same $\nu_d = \frac{1}{T_d}$ intervals equal to the sampling frequency. If the sampling frequency satisfies the conditions of Kotelnikov's theorem, i.e. $\nu_d > 2\nu_m$, where ν_m is the maximum frequency of a continuous signal $f(t)$, then the summed copies do not overlap. Then it is possible to restore the original signal and determine the spectrum of the analog signal $f(t)$ from the set of its samples. Otherwise $(\nu_d < 2\nu_m)$, due to the effect of overlapping copies of the spectrum (frequency mixing), the selection of the spectrum $F(t)$ from the spectrum $F^*(\nu)$, as well as the unambiguous determination of the signal, becomes impossible.

Let's get a set of frequency samples from discrete samples of the signal. Let there be a periodic discrete signal given on the segment $[0, T]$ in the form [58, 64]

$$f^*(t) = \sum_{k=0}^{N-1} f_k\, \delta(t - kT_d)$$

where $T = NT_d$, N is the number of discrete values per period. Let's represent this signal as a Fourier series with a period T:

$$f^*(t) = \sum_{n=-\infty}^{\infty} c_n\, e^{i\frac{2\pi n}{T}t}$$

Find the coefficients of the series

$$c_n = \frac{1}{T}\int_0^T f^*(t)\, e^{-i\frac{2\pi n}{T}t} = \frac{1}{T}\int_0^T \left[\sum_{k=0}^{N-1} f_k \delta(t - kT_d)\right] e^{-i\frac{2\pi n}{T}t}$$

Using the filtering property of the delta function, we get

$$c_n = \frac{1}{T}\sum_{k=0}^{N-1} f(k)\, e^{-i\frac{2\pi nkTd}{T}} = \frac{1}{NTd}\sum_{k=0}^{N-1} f(k)\, e^{-i\frac{2\pi nk}{N}} \tag{1.4}$$

The expression included in equality (1.4)

$$F(n) = \frac{1}{N}\sum_{k=0}^{N-1} f(k)\, e^{-i\frac{2\pi nk}{N}}$$

is called the DFT of a sequence of counts $\{f(k)\}$.

Denote by $W_N = e^{-i\frac{2\pi}{N}}$ a quantity called the transformation kernel. Then the DFT formula takes the form

$$F(n) = \frac{1}{N}\sum_{k=0}^{N-1} f\,(k) W_N^{nk} \tag{1.5}$$

The formula for the inverse discrete Fourier transform (IDFT) is written as

$$f(k) = \sum_{k=0}^{N-1} F\,(n) W_N^{-nk} \tag{1.6}$$

In matrix form, the DFT has the form [86]

$$F = N^{-1} W f$$

where W is the matrix of the basis functions

$$W = \begin{pmatrix} W_N^0 & W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{2(N-1)} \end{pmatrix}$$

The rows of the matrix W are pairwise orthogonal, and when each row is divided by $\sqrt{N}$, they become orthonormal. The IDFT matrix is obtained from the matrix W by replacing all elements with complex conjugates and then transposing. The elements of the matrix W_N^k and W_N^l have the same values if the numbers k and l have the same residuals when divided by N, i.e. if $k = l \pmod N$. For example

$$W_N^0 = W_N^N = 1$$
$$W_{16}^1 = W_{16}^{17} = 0{,}9238 - i\,0{,}3826$$

The formula (1.5) is, up to a scale factor, a discrete approximation of the transformation (1.1), in which the function $f(t)$ is replaced by a step function $f(k) = f(t_k)$ within the length of the sampling element, where $t_k = kT_d$, i.e. the integral is replaced by a finite sum. Expression (1.5) is an approximation, the quality of which improves with an increase in N and a corresponding decrease in the sampling step T_d. For the DFT, all the properties given for the FT are satisfied.

1.1.2 Fast Fourier transform

Consider the DFT $F(n)$ of a sequence of length $N = 2^m$ samples, where m is a positive integer. In this case, (1.5) can be reduced to the transformation of two $(N/2)$-point sequences with even $f(2p)$ and odd $f(2p+1)$ numbers [58, 64]

$$F(n) = \sum_{p=0}^{\frac{N}{2}-1} f(2p)\, W_N^{n2p} + \sum_{p=0}^{\frac{N}{2}-1} f(2p+1)\, W_N^{n(2p+1)}$$

Taking into account the equality of $W_N^{\frac{N}{2}} = -1$, we get

$$F\left(n + \frac{N}{2}\right) = \sum_{p=0}^{\frac{N}{2}-1} f(2p)\, W_N^{n2p} - \sum_{p=0}^{\frac{N}{2}-1} f(2p+1)\, W_N^{n(2p+1)}$$

Since

$$W_N^2 = W_{\frac{N}{2}}$$
$$F(n) = \sum_{p=0}^{\frac{N}{2}-1} f_0\,(p) W_{\frac{N}{2}}^{np} + W_N^n \sum_{p=0}^{\frac{N}{2}-1} f_1\,(p) W_{\frac{N}{2}}^{np} \qquad (1.7)$$

$$F\left(n+\frac{N}{2}\right)=\sum_{p=0}^{\frac{N}{2}-1} f_0\,(p)W_{\frac{N}{2}}^{np}-W_N^n\sum_{p=0}^{\frac{N}{2}-1} f_1\,(p)W_{\frac{N}{2}}^{np}$$

where $f_0(p)$ is an even sequence, $f_1(p)$ is an odd sequence, $n = 0, 1, \dots, \frac{N}{2} - 1$.

Hence, the original N-point transformation reduces to two $\left(\frac{N}{2}\right)$- point transformations, N additions and $\frac{N}{2}$ multiplications by W_N^n. Next, we can replace $\left(\frac{N}{2}\right)$- point transformations with $\left(\frac{N}{4}\right)$- point transformations, etc. This substitution is carried out until $\frac{N}{2}$ - sequences of two elements are formed. As a result, the elements of the original sequence are rearranged according to the rule of binary-inverse permutation according to the new number (index). The index corresponds to the reflection ("mirror" reflection) of the binary code of the element number in the original sequence.

It should be noted that data processing is constructed as a recursive procedure. As a result, the DFT is reduced to $m = \log_2 N$ steps, at each of which 2^k transformations on 2^{m-k} points are performed as 2^{k+1} transformations on 2^{m-k-1} points for N additions and $\frac{N}{2}$ multiplications. The basic operation at the m-th step is the so-called "butterfly" (formula (1.7)). Therefore, the number of computational operations for multiplying complex numbers M and for adding complex numbers A is equal [4, 64]

$$M = \left(\frac{N}{2}\right)\log_2 N \tag{1.8}$$

$$A = N\log_2 N \tag{1.9}$$

The algorithm under consideration is called the Cooley-Tukey algorithm, also called the time-thinning algorithm. For efficient construction of the time-thinning FFT algorithm, the samples of the input sequence should be arranged in binary-inverse order. Then the output sequence will be linear.

The original sequence of $N = 2^m$ points can be divided into two sequences in a different way, namely, the first $\frac{N}{2}$ samples and the last $\frac{N}{2}$ samples. This partitioning is the basis of the Sandy – Tukey algorithm, also called the frequency-thinning algorithm. In this case, the original sequence is written in linear order, and the frequency sequence is obtained in binary-inverse order. In order to rearrange the sequence in linear order, it is necessary to rearrange the elements with binary-inverse numbers after the transformation. In this case, the DFT is carried out in the form of calculating two $\left(\frac{N}{2}\right)$-point transformations for N addition operations and $\frac{N}{2}$ multiplication operations of complex numbers. Since the transformations considered can also be calculated recursively, the total number of operations is determined by the formulas (1.8, 1.9). The number of computational operations is significantly smaller compared to the usual DFT, which requires N^2 operations of multiplying complex numbers. Therefore, the algorithms discussed above are called FFT algorithms [4, 25, 58, 64].

The FFT algorithm can be interpreted as a representation of the DFT matrix of (1.5) in the form of a work sparsely populated (i.e. consisting mainly of zero elements) of the matrix:

$$W = W_n W_{n-1} W_{n-2} \dots W_1$$

where each matrix W_m corresponds to m-th step of the algorithm the FFT and contains in each row, only two non-zero element 1 and $W_M^k, k = 0,1, \dots, 2^{m-1} - 1$. The representation of the DFT matrix as a product of weakly filled matrices is called factorization [58, 64].

If the length of the sequence is determined by a prime number that cannot be decomposed into mutually prime factors, then a different approach is required, which consists of switching to calculations in the system of residual classes, which can significantly reduce the number of multiplication operations. The most well-known algorithm of this type is the GRAPE algorithm (named after the author Sh. Grapes, USA). The GRAPE algorithm provides DFT calculation almost an order of magnitude faster (especially for real data sequences) than when using traditional FFT algorithms, but it requires large amounts of computer RAM. Other disadvantages of the GRAPE algorithm are the significant complication of the rules for rearranging the elements of the processed sequence and the increase in the number of addition operations. In most sources, this algorithm is not even mentioned [4, 25].

1.2 Proni transform

In scientific literature, the Proni transform is sometimes called the Proni method, the Proni decomposition, the Proni transform or the Proni filtration. The French mathematician Gaspard Richet (Baron de Prony) in 1795 proposed a method for modeling a sequence of data samples using a linear combination of exponential functions. Currently, this method is generalized to a model consisting of decaying sinusoids (complex exponents). In addition, it uses a procedure for estimating the model parameters using the least squares method to approximate the model fit in cases where the number of data points exceeds the minimum required number of exponents. In this case, the algorithm is called the generalized Proni method [13, 16, 18, 73]. Often, the response of a linear system to a pulse action is a set of damped sinusoids. For example, short seismic signals reflected from different horizons are similar to attenuated sinusoids. For processing and interpreting such data, the Proni transform is well suited, since the FT in the analysis of short signals does not provide high resolution in the spectral region. The main reason is the discrepancy between the short signals being studied and the infinite sines and cosines that are used in Fourier decomposition. The Proni method uses the decomposition of the signal into decaying sines or cosines of the form

$$\lambda = Ae^{-\delta t}\cos(2\pi ft + \varphi) \qquad (1.10)$$

and the amplitude, attenuation coefficient, frequency and phase are determined $(A, \delta, f\varphi)$.

The approximation of discrete samples of the signal y_n by the base function (1.10) has the form

$$x_n = \sum_{k=1}^{m} A_k \exp(j2\pi f_k \Delta tn + j\varphi_k)\exp(-\delta\Delta tn)\,, n = 1, N,$$

where n is the reference number, N is the number of samples, m is the depth of decomposition, and Δt is the sampling period. The real signal is modulated by complex-conjugate exponents. The minimum root-mean-square error of the approximation of the original function y_n at N observation points is determined

$$\sigma^2 = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - x_i)^2$$

to find the amplitude, attenuation coefficient, frequency, and initial phase of each decomposition level, using the least squares method. Total number of parameters

$$(A_k, \delta_k, f_k, \varphi_k)_{k=1}^{k=m}$$

is a discrete Proni spectrum, by analogy with the discrete Fourier spectrum. This spectrum allows a high degree of approximation of the original non-stationary signal. In contrast to the DFT, where the frequency has uniform samples, the frequency parameter in the case of the Proni transform can have arbitrary values and is one of the estimated parameters. Therefore, in the case of a discrete Proni spectrum, we get an irregular frequency domain for each signal. As a result, for some frequency bands, the values of the transmission parameters will be missing, and the width of these bands may vary significantly depending on the type of signals or data being analyzed [20]. The Proni method is most often used in the seismic exploration of minerals, i.e. in the processing and interpretation of seismic data for solving various geological problems and problems of developing hydrocarbon deposits. It acts as a tool that allows us to localize the areas of anomalous scattering and absorption of seismic energy (depending on the time frequency), when studying the properties of target horizons and productive zones. The analysis and interpretation of these areas make it possible to better understand the properties of the target horizons, and the correlation of these properties with the reservoirs under consideration makes it possible to assess their prospects [20]. It is also used in the determination and identification of biogenic signals, for the analysis of seismic signals of a person who is walking. The parametric description of the signals makes it possible to simplify the formation of diagnostic signs for the recognition of an intruder or a group of violators for security alarm systems [22]. It is also used to compress information.

For practice, the frequency and the attenuation constant are usually of greatest interest. When processing a signal using the Proni transform, difficulties arise and to overcome them, it is necessary to solve some questions, such as: how long should the processing window be selected, how to determine the beginning of the signal, what happens if the beginning of the processing window does not coincide with the beginning of the pulse, how to get rid of interference, etc.

1.3 Wavelet transform

The fundamentals of wavelet analysis were developed in the mid-1980s as an alternative to FT for the study of time (spatial) series with pronounced heterogeneity. Wavelet transforms are usually divided into discrete WT and continuous WT.

The development of wavelets is associated with several directions which were initiated by the work of Haar at the beginning of the 20th century. A significant contribution to the theory of wavelets was made by Guppilaud, Grossman and Morlet, who formulated the main ideas of continuous WT (1982), J. Olaf-Stromberg with early works on discrete wavelets (1983), I. Daubechies, who developed orthogonal wavelets with a compact carrier (1988), Mallat, who proposed a multiple-scale method (1989), N. Delprat, who created a time-frequency interpretation of continuous WT (1991), Newland, who developed harmonic WT, et al. [23, 66, 72].

Unlike FT, which localizes frequencies but does not give a time resolution of the process, and the delta function apparatus, which localizes moments of time but does not have a frequency resolution, WT, which has a self-adjusting mobile frequency-time window, equally well detects both low-and high-frequency characteristics of the signal at different time intervals. This versatility has ensured that wavelet analysis is widely used in various fields of knowledge. Discrete WT is usually used for signal encoding, while continuous WT is used for signal analysis [9, 23, 72]. Discrete WT is widely used in engineering and programming, and continuous WT is widely used in scientific research. Families of analyzing functions, called wavelets, are used in the analysis of various images, to study the structure of turbulent fields, to compress large amounts of information, to use in image recognition problems, in signal processing and synthesis, to determine the characteristics of fractal objects. They are used in astrophysics, geophysics, optics, quantum mechanics, for the analysis of blood pressure, pulse and ECG and DNA analysis. They are also used in protein research and climate research.

1.3.1 Continuous (integral) wavelet transform

Wavelet analysis is the decomposition of the signal under study by functions localized both in physical space (time, coordinate) and in Fourier space (frequency). The wavelet decomposition projects a one-dimensional signal onto the time-frequency half-plane, which makes it possible to separate different-scale events and to investigate the dependence of spectral characteristics on time. The family of wavelet functions $\psi_{ab}(t)$ is generated from a single "parent" function $\varphi(t)$ by stretching (compression) and shifting

$$\psi(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-b}{a}\right)$$

due to the operation of shifting in time b and changing the time scale a [1, 9, 66]. For the given values of the parameters a and b, the function $\psi_{ab}(t)$ is the wavelet. The multiplier $\frac{1}{\sqrt{a}}$ ensures that the norm of these functions is independent of the scaling number a. In general, this multiplier is written as a^k, where the parameter k is the exponent of the scale factor. The specific choice of this parameter depends on the purpose of the analysis. The exponent of the scale factor $k = -\frac{1}{2}$ is used to ensure that the signal at each scale has the same energy and if the results of the wavelet analysis are supposed to be compared with the Fourier representation of the signal.

The parameter $k = -1$ is widely used, in which equal values of the wavelet coefficients $W(a, b)$ correspond to equal amplitudes of the signal ripples, regardless of the ripple scale [2]. Wavelets are special functions in the form of short waves (bursts) with a zero integral value and localization along the axis of the independent variable (t or x), capable of shifting along this axis and scaling (stretching and compression). Any of the most commonly used types of wavelets generate a complete orthogonal system. In the case of wavelet analysis (decomposition) of a process (signal), by changing the scale, the wavelets are able to detect differences in the characteristics of the process on different scales, and by wavelets shifting the properties of the process can be analyzed at different points over the entire interval under study. It is due to the completeness property of this system that it is possible to restore (reconstruct or synthesize) the process by means of the inverse WT [9, 24, 66].

Thus, in the frequency domain, the wavelet spectra are similar to bursts (waves) with a peak at the frequency ω and a band $\Delta\omega$, i.e. they have the form of a bandpass filter; in this case, ω and $\Delta\omega$ decrease with the growth of the parameter a. Hence, the wavelets are localized in both the time and frequency domains. It should be noted that the spectral representation (image) of the wavelets is similar to the window setting in the window Fourier transform. However, the difference is that the properties of the window (its width and frequency movement) are inherent in the wavelets themselves. In this regard, with the help of wavelets, it is possible to analyze and synthesize the local feature of any signal $S(t)$ (the function $S(x)$). In WT, the term "shift" means that in the window Fourier transform, it also refers to the location of the window. This term refers to the temporal information present as a result of the WT signal $S(t)$. With WT, we do not have a frequency parameter, as with the windowed Fourier transform, instead, we use a scale factor a, which can be defined as the inverse of the frequency [1, 9, 24, 66].

The continuous WT of a one-dimensional signal $S(t)$ is its representation as a Fourier integral over a system of basis functions $\psi(t)$ [1, 9, 66]:

$$W(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} S(t)\,\psi\left(\frac{t-b}{a}\right)dt \tag{1.11}$$

It follows from (1.11) that the wavelet spectrum $W(a,b)$, in contrast to the Fourier spectrum, is a function of two arguments: the first argument a (time scale) is analogous to the oscillation period, and the second argument b is the signal offset along the time axis. The spectrum $W(a,b)$ of a one-dimensional signal $S(t)$ is a surface in three-dimensional space. The three-dimensional image of the spectrum allows us to analyze the properties of the signal simultaneously in the physical and frequency spaces.

In the course of using wavelets for signal analysis, continuous WT is more convenient; its certain redundancy, associated with a continuous change in the scale factor a and the shift parameter b, becomes a positive quality, since it allows us to more fully and clearly present and analyze the information contained in the data [1, 9, 57, 58].

Like the inverse FT, there is an inverse continuous WT [2]:

$$S(t) = C_{\psi}^{-1} \int_{0}^{\infty} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) W(a,b) \frac{dadb}{a^{3+k}} \tag{1.12}$$

where C_{ψ} is normalizing coefficient:

$$C_{\psi} = \int_{-\infty}^{\infty} \left|F_{\psi}(\omega)\right|^{2} \cdot \omega^{-1} d\omega < \infty,$$

$F_{\psi}(\omega)$ Fourier spectrum of the basis function, and the parameter k is the exponent of the scale factor.

In some cases, at the stage of signal reconstruction (synthesis), it is possible to use a different wavelet than the one used at the stage of decomposition (analysis), compensating for the "shortcomings" of the original wavelet [66, 72].

1.3.2 Discrete wavelet transform

The main difference between discrete and continuous WT is the use of different types of wavelets. As a rule, discrete wavelets have no derivatives and, when decomposed into a Fourier series, they have long "tails". For discrete WT, in addition to the wavelet, the scaling function is also used. The scaling function and the wavelet are evaluated recursively and have no analytical expressions. The origins of discrete WT go back to 1976, when the method of sub-band (pyramidal) encoding of the speech signal was developed. With this encoding, the signal is passed through a tree-like connection of RF and LF filters. Let the sequence $x[n]$ be obtained by sampling the continuous signal $x(t)$. First, the signal is passed through a low-pass filter with a pulse response g

$$y[n] = (x^{*}g)[n] = \sum_{k=-\infty}^{\infty} x[k]\, g[n-k],$$

that is, the convolution is calculated [30].

At the same time, the signal is decomposed using the high-pass filter h. The result is the detailing coefficients (after the RF filter) and the approximation coefficients (after the LF filter, Fig. 1.3.2.1). These two filters are related and are called quadrature mirror filters (QMF). Since half of the frequency range of the signal was filtered out, then, according to Kotelnikov's theorem, the signal counts can be reduced by two times [30]:

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]\, g[2n-k]$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[2n-k]$$

This decomposition halved the time resolution due to the thinning of the signal. However, each of the resulting signals represents half the frequency band of the original signal, so the frequency resolution is doubled.

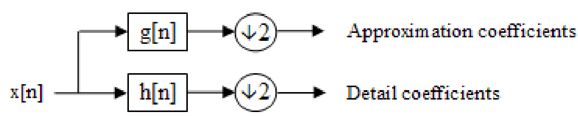


Figure 1.3.2.1. Signal decomposition scheme for discrete WT

This decomposition can be repeated several times to further increase the frequency resolution with further thinning of the coefficients after low-pass and high-pass filtering.

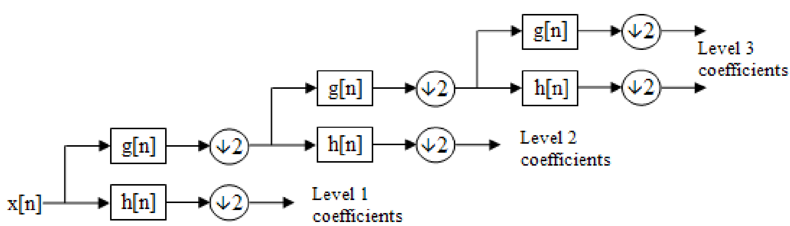


Figure 1.3.2.2. Three-level filter bank (comb)

This decomposition can be represented as a binary tree, where the leaves and nodes correspond to spaces with different time-frequency localization. This tree represents the structure of the filter bank (comb) (Fig. 1.3.2.2) [30]. At each level of the diagram above, the signal is decomposed into low and high frequencies. After double decimation, the signal length must be a multiple of 2^n, where n is the number of decomposition levels. The signal is restored in the inverse order, i.e. zero elements are added to the detailing and approximating coefficients, passed through mirror filters, and added together.

For the synthesis of scaling and wavelet functions, a system of equations for the pulse characteristics of filters is solved, and the scaling and wavelet functions are recursively calculated from these characteristics. The larger the order of the wavelet, the more complex the system of equations. I. Daubechies managed to find a method that allows us to construct an infinite series of orthogonal wavelets, each of which is determined by a finite number of coefficients. It became possible to construct an algorithm that implements the fast wavelet transform (FWT) of discrete data (Mall's algorithm). The disadvantages of discrete wavelets include the asymmetry of the shape and the sharp boundaries of the basis function. Not all discrete wavelets have FWT algorithms [59].

1.4 Wavelet functions

A wide set of wavelets can be used as basis functions. For practical application, it is important to know the features that the original function must necessarily have in order to become a wavelet.

Let us consider the main features [1, 9, 66].

1. Limitation. The square of the norm of the function must be finite:

$$||\psi||^2 = \int_{-\infty}^{\infty} |\psi(t)|^2 \, dt < \infty$$

2. Localization. WT, unlike FT, uses the original function, localized in both time and frequency. To do this, it is enough that the conditions are met:

$$|\psi(t)| \leq C(1+|t|)^{-1-\varepsilon}$$
$$\left|S_{\psi}(\omega)\right| \leq C(1+|\omega|)^{-1-\varepsilon}, \varepsilon > 0$$

3. Zero average. The graph of the original function should oscillate (be alternating) around zero on the time axis and have zero area [1, 9]

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0$$

From this condition, the choice of the name "wavelet" – a small wave-becomes clear. If the area of the function $\psi(t)$, i.e. the zero moment, is equal to zero, the FT $S(\omega)$ of this function is equal to zero at $\omega = 0$ and has the form of a bandpass filter. For different values of a, this is a set of bandpass filters.
Often, for applications, it is necessary that all the first n moments are equal to zero [1, 9]:

$$\int_{-\infty}^{\infty} t^{n}\psi(t)\, dt = 0$$

Wavelets of n-th order allow one to analyze the finer (high-frequency) structure of the signal, suppressing its slowly changing components.

4. Self-modality. A characteristic feature of a wavelet is its self-similarity. All the wavelets of a particular family $\psi(t)$ have the same number of oscillations as the parent wavelet $\psi(t)$ because they are obtained from it by means of scale transformations (a) and shift (b).

The choice of the analyzing wavelet is largely determined by what information needs to be extracted from the signal. Taking into account the characteristic features of different wavelets in the time and frequency spaces, it is possible to identify certain properties and features in the analyzed signals that are invisible on the signal graphs, especially in the presence of strong noise. At the same time, the task of signal reconstruction may not be set, which expands the family of used wavelet functions. To construct such wavelets, derivatives of the Gaussian function are often used [9, 66]:

$$\psi_m(t) = (-1)^m \frac{\partial^m}{\partial t^m}\left[\exp\left(-\frac{t^2}{2}\right)\right]$$

The higher derivatives of the Gaussian function have more zero moments and allow us to extract information about the higher-order features contained in the signal. We will present some wavelets used in the study of signals.

The *MHAT*-wavelet is calculated from the second derivative $(m = 2)$ of the Gaussian function. The Wavelete equation is:

$$\psi(t, a, b) = \frac{1}{\sqrt{a}} = \left(1 - \left(\frac{t-b}{a}\right)^2\right)\exp\left[-\frac{1}{2}\cdot\left(\frac{t-b}{a}\right)^2\right]$$

The wavelet is symmetric, the wavelet spectrum is represented only by the real part, and it is well localized in frequency. The first and last moments of the wavelet are zero. The wavelet is used for analyzing complex signals.

Figure 1.4.1 shows the *MHAT*-wavelet for the scale factor $a = 3$. Fig. 1.4.2 shows the *MHAT*-wavelet for the scale factor $a = 15$.

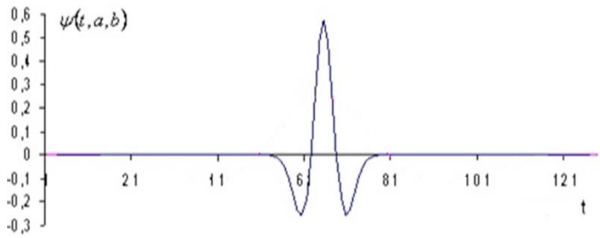


Figure 1.4.1. *MHAT*-wavelet for $a = 3$

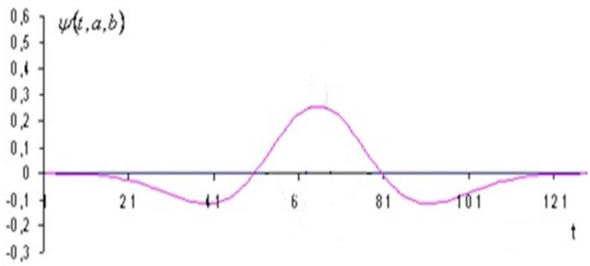


Figure 1.4.2. *MHAT*-wavelet for $a = 15$

Figures 1.4.3 and 1.4.4 show the second-order wavelet spectra for $a = 3$ and $a = 15$.

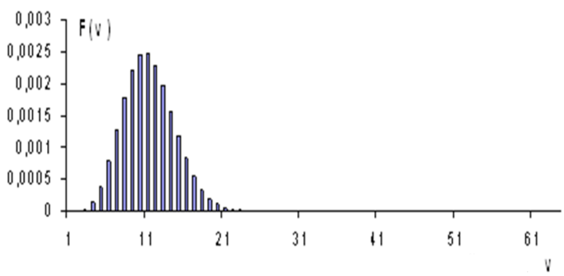


Figure 1.4.3. *MHAT*-wavelet spectrum for $a = 3$

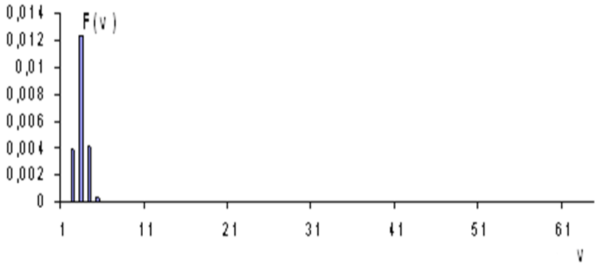


Figure 1.4.4. *MHAT*-wavelet spectrum for $a = 15$

Figure 1.4.5 shows a graph of the third-order derivative of the Gaussian function for the scale coefficients $a = 3$ (blue) and $a = 15$ (purple).

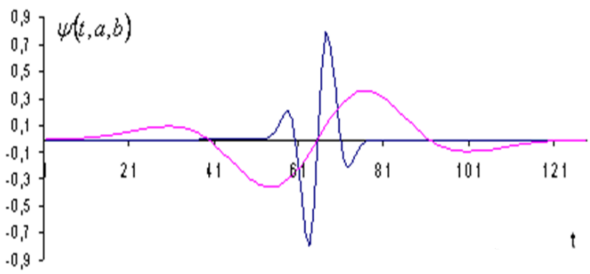


Figure 1.4.5. The third-order derivative of the Gaussian function for $a = 3$ and $a = 15$

Figure 1.4.6 shows a graph of the fourth-order derivative of the Gaussian function for the scale coefficients $a = 3$ (blue) and $a = 15$ (purple).

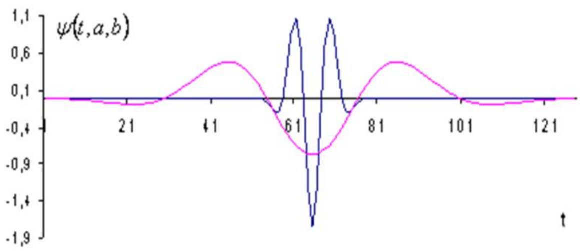


Figure 1.4.6. The fourth-order derivative of the Gaussian function for $a = 3$ and $a = 15$

A complex Morlaix wavelet is a plane wave modulated by a Gaussian of unit width. The Wavelet equation is

$$\psi(t,a,b) = \frac{1}{\sqrt{a}} \cdot \exp(i\omega_0 t)\exp\left[-\frac{1}{2}\cdot\left(\frac{t-b}{a}\right)^2\right]$$

Figure 1.4.7 shows the graph of the real part of the Morlaix function for $a = 15$.

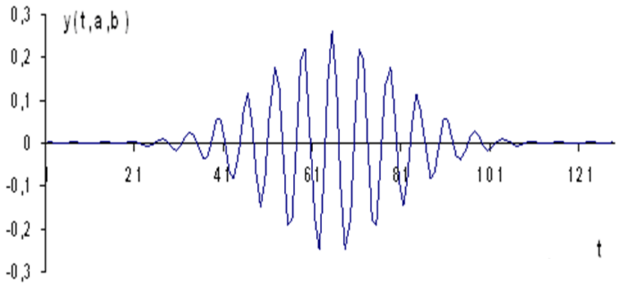


Figure 1.4.7. Complex Morlet wavelet

Figures 1.4.8, 1.4.9 show the spectra of the real part of the Morlet wavelet for different scale coefficients.

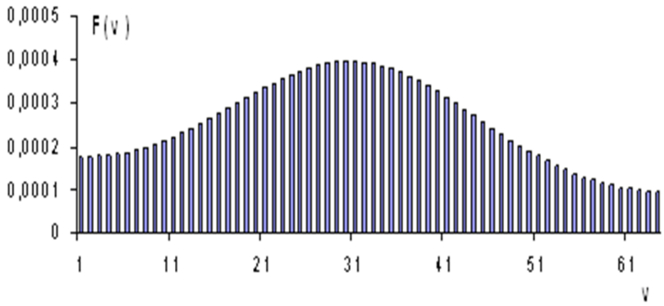


Figure 1.4.8. Spectrum of the real part of the Morlet wavelet for $a = 1$

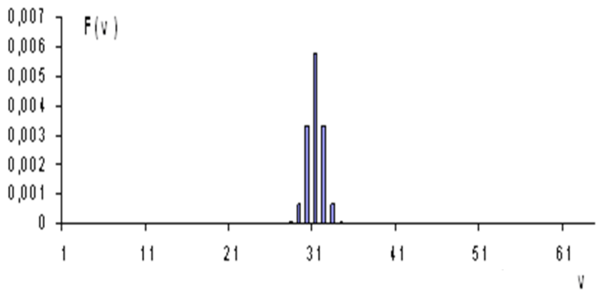


Figure 1.4.9. Spectrum of the real part of the Morlet wavelet for $a = 15$

If we characterize the width of the wavelet spectrum $\Delta\nu$ by the value

$$(\Delta\nu)^2 = \int_{-\infty}^{\infty} \nu^2 F(\nu)^2 \, d\nu$$

which is a measure of the "spread" of energy in the frequency domain, and as a measure of the duration of the wavelet Δt, we take the value

$$(\Delta t)^2 = \int_{-\infty}^{\infty} (t - m_t)^2 \psi(t, a, b)^2 \, dt$$

where $m_t = \int_{-\infty}^{\infty} t \, \psi(t, a, b)^2 \, dt$. Then the uncertainty principle is fulfilled for them

$$\Delta t \, \Delta\nu \geq \frac{E}{4\pi}$$

where $E = \int_{-\infty}^{\infty} f(t)^2 \, dt$.

Figures illustrate the uncertainty principle, which states that it is impossible to achieve localization of energy simultaneously in both the time and frequency domain. The wider the wavelet, the narrower the spectrum, and the narrower the wavelet, the wider the spectrum. Wavelets have the properties of variable time-frequency resolution, which compares favorably with windowed FT. For an effective study of non-stationary signals, just such basis functions are needed.

1.4.1 Properties of wavelet analysis

The forward WT contains the combined information about the analyzed signal and the analyzing wavelet. Wavelet analysis allows us to obtain objective information about the analyzed signal, since the properties of WT (linearity, invariance with respect to shear and invariance with respect to stretching (compression)) do not depend on different basis functions [1, 9, 66].

1. Linearity. It follows from the scalar product

$$W[\alpha S_1(t) + \beta S_2(t)] = \alpha W_1(a, b) + \beta W_2(a, b)$$

2. Shift. A shift of the signal in the time domain by b_0 leads to a shift of the wavelet image also by b_0:

$$W[S(t - b_0)] = W[a, b - b_0]$$

3. Scaling. Stretching (compression) of the signal also leads to stretching (compression) of it in the region $W(a, b)$:

$$W\left[S\left(\frac{t}{a_0}\right)\right] = \frac{1}{a_0} W\left[\frac{a}{a_0}, \frac{b}{b}\right]$$

4. Differentiation:

$$W[d_t^m S] = (-1)^m \int_{-\infty}^{\infty} S(t)\, d_t^m[\psi_{ab}(t)]dt$$

where $d_t^m = \frac{d^m[\ldots]}{dt^m}, m \geq 1$.

It follows from this property that it is possible to ignore large-scale components and analyze high-order features or small-scale variations of the $S(t)$ signal by differentiating either the wavelet or the signal itself the required number of times. This means that the wavelet spectrum of the derivatives is easy to calculate using the derivatives of the basis functions.

5. Scale-time localization. It is due to the fact that the elements of the WT basis are well localized and have a mobile time-frequency window. By changing the scale (increasing the coefficient a leads to a narrowing of the Fourier spectrum of the function $\psi(t)$), wavelets are able to detect differences in characteristics on different scales (frequencies), and by shifting-to analyze the properties of the signal at different points over the entire studied interval. Therefore, when analyzing non-stationary signals, due to the locality property, WT has a significant advantage over the Fourier transform, which gives only global information about the frequencies (scales) of the analyzed signal since the system of functions used in this case (the complex exponent or the sines and cosines) is defined on an infinite interval.

1.4.2 Advantages and disadvantages of the wavelet transform

The wavelet transform of signals is a generalization of spectral analysis, whose typical representative is the Fourier transform. WT has almost all the advantages of FT. However, the FT basis functions do not satisfy the necessary condition of simultaneous localization in time space and frequency space. They cover the entire time axis, so they do not allow us to get localized information, such as when the signal frequency changes. Some of the problems are removed when using the window FT. But the basis functions of the windowed FT have the same frequency and time resolution for all points of the transform plane while the WT basis functions have different resolutions. When using the window Fourier transform, the narrow window has the best time resolution, and the wide window has the best frequency resolution. The problem is that we have to choose a fixed window for the entire signal, whereas different parts of the signal may require different windows.

Unlike windowed FT, continuous WT has good time resolution and poor frequency resolution at high frequencies, while at low frequencies it has good frequency resolution and poor time resolution. The basis function WT is well localized and quickly tends to zero outside of a small interval. This property of WT gives it a great advantage in signal analysis, since fast signal variations (high-frequency characteristics) are well-localized, and a good low-frequency resolution is sufficient to detect slowly changing characteristics. When the signal has high-frequency components of short duration and low-frequency components of extended duration, the use of WT is most effective. The wavelet transform analyzes the signal at different frequencies and at different resolutions simultaneously.

The wavelet bases can be well localized in both frequency and time. When identifying well-localized multi-scale processes in the signals, only those scale levels of decomposition that are of target interest can be considered.

Wavelet bases, in contrast to FT, have quite a lot of different basis functions, the properties of which are focused on solving various problems. The properties of the wavelet coefficients as functions of both the scale and the position of the point represent a unique opportunity to describe point singularities. The higher the order of the analyzing wavelet, the more zero moments it has, and the better WT differentiates the singularities. Wavelet analysis allows us to find fractional Helder exponents, because it can be used to investigate, characterize, and easily distinguish some specific local properties of generalized functions, and it also allows us to determine the fractal dimensions of sets of points in which the function is singular [1, 9].

The disadvantage of WT is its relative complexity, the difficult interpretation of WT results. In [1], it is noted that WT is by no means a substitute for Fourier analysis, and it has no fewer advantages. It simply allows us to look at the analyzed processes from a slightly different point of view. WT makes it possible to effectively study signals that do not have a clearly defined periodicity. Thus, these two types of analysis are more complementary to each other than competitors.

1.5 Conclusions

1. Traditional spectral analysis based on the Fourier transform is inefficient for non-stationary signals, when the time of frequency change is much less than the duration of the implementation intended for analysis.

2. The Proni transform is well-suited for processing short, fast-changing signals, but for calculating long, non-stationary signals, FFT cannot be used to reduce the calculation time.

3. The wavelet transform is well suited for analyzing non-stationary signals of any duration and allows us to isolate the frequencies and the time of the signal frequency change.

4. The disadvantages of discrete wavelets include the asymmetry of the shape and the sharp boundaries of the basis function. The discrete WT is calculated in the time domain, which requires a lot of time for a large sample of the signal.

5. Continuous WT can also be performed in the frequency domain, which allows the use of FFT algorithms to speed up calculations.

2 DEVELOPMENT OF ALGORITHMS FOR NUMERICAL CALCULATION OF THE FORWARD AND INVERSE FAST CONTINUOUS WAVELET TRANSFORM WITH AN ARBITRARY SCALING FACTOR

2.1 Algorithm for numerical calculation of the direct fast continuous wavelet transform in the frequency domain

Continuous WT has a number of positive properties (symmetry, smoothness of the basis function, the possibility of analytical description), which are necessary for the analysis and synthesis of the signals under study. However, the impossibility of practical implementation of the transformation in real time negates all the positive properties of the continuous WT.

WT algorithms are represented in widely used computer mathematics systems, such as *MathCAD, MATLAB* and *Mathematica*. Continuous WT is usually performed by direct numerical integration [1]. Calculating WT by direct numerical integration for large time sequences takes a long time. The direct calculation of WT requires N^2 multiplication operations [25]. The number of computational operations is significantly higher compared to the WT calculation algorithm using FFT, in which the number of multiplication operations increases almost linearly. To increase the performance, the algorithm of continuous fast WT using FFT is developed.

2.1.1 The principle of calculating continuous WT in the frequency domain

The goal is to obtain the formula for calculating the Fourier spectrum WT $W(a,b)$ using the Fourier spectrum of the signal $S(t)$ and the Fourier spectrum of the wavelet $\psi(t)$.

The wavelet spectrum $W(a,b)$ is the correlation between the wavelet at different scales a and the signal

$$W(a,b) = \int_{0}^{2\pi} S(t)\,\psi_a(t-b)dt$$

Let's decompose the signal $S(t)$ into a Fourier series

$$S(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left(a_n \cos\left(\frac{2\pi}{T}nt\right) + b_n \sin\left(\frac{2\pi}{T}nt\right)\right)$$

Let $T = 2\pi$, then

$$S(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_n \cos(nt) + b_n \sin(nt))$$

From the generalized closure equation, it follows that the Fourier series of the signal $S(t)$ can be directly integrated by multiplying by the function $\psi_a(t-b)$.

$$W(a,b) = \int_0^{2\pi} \frac{a_0}{2}\psi_a(t-b)dt + \sum_{n=1}^{\infty}\int_0^{2\pi}(a_n \cos nt + b_n \sin nt)\,\psi_a(t-b)dt =$$
$$= \int_0^{2\pi} \frac{a_0}{2}\psi_a(t-b)dt + \int_0^{2\pi} a_1 \cos t\,\psi_a(t-b)dt + \int_0^{2\pi} a_2 \cos 2t\,\psi_a(t-b)dt +$$
$$+ \int_0^{2\pi} a_3 \cos 3t\,\psi_a(t-b)dt + \cdots + \int_0^{2\pi} a_1 \sin t\,\psi_a(t-b)dt +$$
$$+ \int_0^{2\pi} a_2 \sin 2t\,\psi_a(t-b)dt + \int_0^{2\pi} a_3 \sin 3t\,\psi_a(t-b)dt + \cdots$$

Decompose the wavelet $\psi_a(t-b)$ into a Fourier series:

$$\psi_a(t-b) = \frac{c_0}{2} + \sum_{k=1}^{\infty}(c_n \cos(nt-nb) + d_n \sin(nt-nb)) =$$
$$= \frac{c_0}{2} + \sum_{k=1}^{\infty}\big(c_n(\cos(nt)\cos(nb) + \sin(nt)\sin nb)\big) +$$
$$+d_n(\sin(nt)\cos(nb) - \cos(nt)\sin(nb))$$

Substituting the Fourier series of the wavelet $\psi_a(t-b)$ into each integral, we obtain

$$W(a,b) = \int_0^{2\pi} \left(\frac{a_0 c_0}{4}\right) dt + \int_0^{2\pi} \left(a_1 \frac{c_0}{2}\right) \cos(t)\, dt + \int_0^{2\pi} \left(a_1 \frac{c_0}{2}\right) \sin(t)\, dt +$$

$$+ \int_0^{2\pi} \left(\frac{a_0}{2} c_1\right) \cos(t)\cos(b)\, dt + \int_0^{2\pi} \left(\frac{a_0}{2} c_1\right) \sin(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (a_1 c_1) \cos(t)\cos(t)\cos(b)\, dt + \int_0^{2\pi} (a_1 c_1) \cos(t)\sin(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (a_1 d_1) \cos(t)\sin(t)\cos(b)\, dt - \int_0^{2\pi} (a_1 d_1) \cos(t)\cos(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (b_1 c_1) \sin(t)\cos(t)\cos(b)\, dt + \int_0^{2\pi} (b_1 c_1) \sin(t)\sin(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (b_1 d_1) \sin(t)\sin(t)\sin(b)\, dt - \int_0^{2\pi} (b_1 d_1) \sin(t)\cos(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (a_2 c_1) \cos(2t)\cos(t)\cos(b)\, dt + \int_0^{2\pi} (a_2 c_1) \cos(2t)\sin(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (a_2 d_1) \cos(2t)\sin(t)\cos(b)\, dt - \int_0^{2\pi} (a_2 d_1) \cos(2t)\cos(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (b_2 c_1) \sin(2t)\cos(t)\cos(b)\, dt + \int_0^{2\pi} (b_2 c_1) \sin(2t)\cos(t)\sin(b)\, dt +$$

$$+ \int_0^{2\pi} (b_2 d_1) \sin(2t)\sin(t)\sin(b)\, dt - \int_0^{2\pi} (b_2 d_1) \sin(2t)\cos(t)\sin(b)\, dt + \cdots$$

Since the system of functions 1, $\cos(x)$, $\sin(x)$, $\cos(2x)$, $\sin(2x)$, $\cos(3x)$, $\sin(3x)$, ... is orthogonal:

$$\int_0^{2\pi} \sin(kt)\,dt = 0 \text{ for any } k$$
$$\int_0^{2\pi} \cos(kt)\,dt = 0, \text{ if } k \neq 0$$
$$\int_0^{2\pi} \cos(px)\sin(kt)\,dt = 0 \text{ for any } p, k$$
$$\frac{1}{\pi}\int_0^{2\pi} \cos(px)\cos(kt)\,dt = \begin{cases} 0, & \text{if } k \neq p \\ 1, & \text{if } k = p \neq 0 \end{cases}$$
$$\frac{1}{\pi}\int_0^{2\pi} \sin(px)\sin(kt)\,dt = \begin{cases} 0, & \text{if } k \neq p \\ 1, & \text{if } k = p \neq 0 \end{cases}$$

We obtain

$$\begin{aligned} W(a,b) = \pi\Big(&\frac{a_0c_0}{2} + a_1c_1\cos b - a_1d_1\sin b + b_1c_1\sin b + b_1d_1\cos b \\ &+ a_2c_2\cos 2b - a_2d_2\sin 2b \\ &+ b_2c_2\sin 2b + b_2d_2\cos 2b + \cdots\Big) \\ = \pi\Big(&\frac{a_0c_0}{2} + (a_1c_1 + b_1d_1)\cos b + (b_1c_1 - a_1d_1)\sin b \\ &+ (a_2c_2 + b_2d_2)\cos 2b + (b_2c_2 - a_2d_2)\sin 2b \\ &+ (a_3c_3 + b_3d_3)\cos 3b \\ &+ (b_3c_3 - a_3d_3)\sin 3b + (a_4c_4 + b_4d_4)\cos 4b \\ &+ (b_4c_4 - a_4d_4)\sin 4b + \cdots\Big) \end{aligned}$$

The resulting amount is briefly written as follows:

$$W(a,b) = \pi\left(\frac{a_0c_0}{2} + \sum_{k=1}^{\infty}(a_nc_n + b_nd_n)\cos nb + (b_nc_n - a_nd_n)\sin nb\right)$$

Denote

$$a_0' = a_0c_0,\ a_n' = (a_nc_n + b_nd_n),\ b_n' = (b_nc_n - a_nd_n)$$

Then

$$W(a,b) = \pi\left(\frac{a_0'}{2} + \sum_{k=1}^{\infty}(a_n' \cos nb + b_n' \sin nb)\right)$$

That is, the Fourier coefficients of the wavelet spectrum $W(a,b)$ are calculated using the Fourier coefficients of the signal $S(t)$ and the Fourier coefficients of the wavelet $\psi(t)$. Using the inverse transformation of the sum and the difference of the product of the Fourier coefficients of the signal $S(t)$ and the wavelet $\psi(t)$, the wavelet spectrum $W(a,b)$ of the signal $S(t)$ is calculated.

Thus, in order to calculate the WT of a signal in the frequency domain, it is necessary to obtain the Fourier spectra of the signal and the wavelet for different scale coefficients a, to find the complex conjugate spectrum, and to inverse the Fourier transform of the complex conjugate spectra to obtain the wavelet spectrum of the signal.

2.1.2 Algorithm for numerical calculation of direct fast continuous WT

The algorithm for numerical calculation of the direct continuous fast WT signal $S(t)$ in the frequency domain includes the following steps.

1. The coefficients of the trigonometric series $a_1(n)$ of the signal $S(k)$ are calculated using the direct fast Fourier transform according to the formula

$$a_1(n) = \frac{1}{N}\sum_{k=0}^{N-1} S(k)\cos\left(\frac{2\pi nk}{N}\right)$$

2. The coefficients of the trigonometric series $b_1(n)$ of the signal $S(k)$ are calculated using the direct fast Fourier transform according to the formula

$$b_1(n) = \frac{1}{N}\sum_{k=0}^{N-1} S(k)\sin\left(\frac{2\pi nk}{N}\right)$$

3. The coefficients of the trigonometric series $a_2(n)$ of the wavelet $\psi(k)$ are calculated using the direct fast Fourier transform according to the formula

$$a_2(n) = \frac{1}{N}\sum_{k=0}^{N-1} \psi(k)\cos\left(\frac{2\pi nk}{N}\right)$$

4. The coefficients of the trigonometric series $b_2(n)$ of the wavelet $\psi(k)$ are calculated using the direct fast Fourier transform according to the formula

$$b_2(n) = \frac{1}{N}\sum_{k=0}^{N-1} \psi(k)\sin\left(\frac{2\pi nk}{N}\right)$$

5. The complex conjugate spectrum is calculated:

$$c_1(n) = a_1(n)\cdot a_2(n) + b_1(n)\cdot b_2(n) \tag{2.1}$$
$$c_2(n) = b_1(n)\cdot a_2(n) - a_1(n)\cdot b_2(n) \tag{2.2}$$

Most continuous wavelets are either even or odd functions. For even – numbered wavelets, the series consists of one cosine, and for odd-numbered ones-of one sine. For even wavelets by property 5, $b_2(n) = 0$ and

$$c_1(n) = a_1(n)\cdot a_2(n) \tag{2.3}$$
$$c_2(n) = b_1(n)\cdot a_2(n) \tag{2.4}$$

For odd wavelets by property 5, $a_2(n) = 0$ and

$$c_1(n) = b_1(n)\cdot b_2(n) \tag{2.5}$$

$$c_2(n) = -a_1(n) \cdot b_2(n) \tag{2.6}$$

6. For an even wavelet with M different scale coefficients, the wavelet spectrum $W(a,b)$ (the matrix of $M \times N$ wavelet coefficients) for the input signal of N samples is obtained by calculating the M inverse Fourier transforms of the complex conjugate spectrum (2.3), (2.4) by the formula

$$W(a,n) = \sum_{k=0}^{N-1} \big(c_1(k) + ic_2(k)\big) \exp\left(i\frac{2\pi nk}{N}\right)$$

7. For an odd wavelet with M different scale coefficients, the wavelet spectrum $W(a,b)$ (the matrix of $M \times N$ wavelet coefficients) for the input signal of N samples is obtained by calculating the M inverse Fourier transforms from the complex conjugate spectrum (2.5), (2.6) according to the formula

$$W(a,n) = \sum_{k=0}^{N-1} \big(c_1(k) + ic_2(k)\big) \exp\left(i\frac{2\pi nk}{N}\right)$$

For even-numbered wavelets, clauses 4 and 7 are not met. For odd-numbered wavelets, clauses 3 and 6 are not met.

The coefficients $a_2(n)$, $b_2(n)$ are calculated only with the wavelet, and not with the signal under study. In this regard, it is possible to calculate them in advance and store the results of calculations in RAM or ROM. Due to the parity (odd) of continuous wavelets, the number of multiplications and additions is reduced by 2 times for each scale according to the formulas (2.3 – 2.6). The amount of memory required to store the Fourier coefficients of the wavelets for each scale is also reduced by a factor of 2. Figures 1.4.3 - 1.4.4 (Chapter 1) show the wavelets and their spectra, where it is clearly seen that when the scale factor a increases, the wavelet spectra narrow, and the wavelets themselves expand. We can use this property to reduce memory when storing data. For small scale coefficients a, less memory is needed when storing wavelets, and for large ones-when storing their spectra. Also, fewer multiplications are needed by the formulas (2.3–2.6) in the frequency domain because only the low-frequency components of the spectrum are different from zero.

Using decimation when increasing the scale factor allows us to further reduce the number of addition and multiplication operations. Decimation corresponds to a decrease in the sampling rate, or the removal of some samples from the signal. For example, a double decimation means that every second sample is removed from the signal. At high scales (low frequencies), the sampling rate can be lowered according to Kotelnikov's theorem. When the scale factor a is increased by eight times, the sampling rate can be lowered by eight times, since when the scale is increased, the time resolution decreases and the frequency resolution increases, i.e. for a low-frequency component, we can specify the frequency value more accurately, but less accurately specify its time position.

Decimation for large scale coefficients does not degrade the time resolution of the signal since the deleted samples are redundant and do not carry information. There is no need to describe the signal in too much detail, because when using WT with a large scale factor, the signal is multiplied by the wavelet and integrated over a wider time interval than when using a wavelet with a small scale factor. For example, the wavelet coefficients for large scale coefficients have the same values for a large offset interval b if the same sampling rate is used as for small scale coefficients. Therefore, the use of decimation allows, without reducing the amount of information contained in the signal, i.e. without degrading the resolution, to reduce the number of calculations.

In order to study the wavelet spectrum in the same plane $a \times b$, after decimation, it is necessary to use interpolation, i.e. increase the sampling rate. The interpolation corresponds to increasing the sampling rate by adding new samples between the calculated wavelet coefficients. As new samples, we can use zeros or the values of the calculated wavelet coefficients. The use of decimation and interpolation reduces the calculation time of the wavelet spectrum, since, if we apply uniform sampling, the calculation time of the wavelet coefficients increases in proportion to the number of scale coefficients a.

The structural diagram of the direct fast continuous WT device is shown in Figure 2.1.2.1. The analyzed signal $S(t)$ is fed to the ADC(Block 1), from the output of which a discrete sample $S(n)$ with a length of N samples is fed to the input of the FFT calculator (block 2).

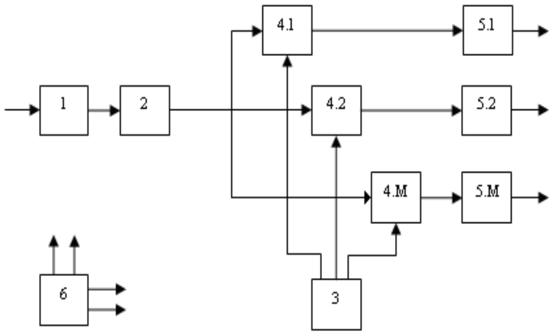


Figure 2.1.2.1. The structural diagrams of the direct fast wavelet transform: 1 – analog-to-digital converter (ADC); 2 – FFT calculator; 3 – read-only memory; 4.1 – 4.M – multipliers; 5.1 – 5.M – inverse FFT calculators; 6 – control device

From the output of block 2, the coefficients of the series $a_1(n)$, $b_1(n)$ of the signal simultaneously enter the first inputs of M multipliers (blocks 4.1 – 4.M). From the ROM (block 3), the coefficients of the series $a_2(n)$ (for even), $b_2(n)$ (for odd) wavelets go to the second inputs of M multipliers (blocks 4.1 – 4.M), from the outputs of which the multiplication results go to the inputs of the inverse FFT calculators (blocks 5.1 – 5.M).

From the outputs of the M inverse FFT calculators (blocks 5.1-5.M), the results of the WT signal are taken in the form of an array of values of the wavelet coefficients with the size of M scales on N shifts $W(m,n)$. The control device (block 6) synchronizes the operation of ADC units (block 1), FFT calculators (block 2), multipliers (blocks 4.1 – 4.M), and inverse FFT calculators (blocks 5.1 – 5.M). This device allows us to select various types of wavelet functions with an arbitrary sampling step of scale coefficients stored in ROM (block 3) for analyzing the input signal [33, 35, 36, 43, 48].

The time characteristics of calculating a continuous WT are shown below. It is shown that the speed of calculating continuous WT in the frequency domain at large scale coefficients a is 3 times higher than in the traditional calculation in the frequency domain. Thus, the algorithm for calculating fast continuous WT combines the advantages of FFT and continuous WT.

2.2 Algorithm for numerical calculation of the inverse fast continuous wavelet transform in the frequency domain

According to the references [77, 78, 79, 80, 82, 83, 86, 87, 89, 90], the use of continuous wavelets for signal analysis is considered preferable to discrete WT. The use of continuous wavelets for data analysis is more convenient, because their certain redundancy, associated with a continuous change in the scale factor a and the shift parameter b, becomes a positive quality in this case, as it allows one to more fully and clearly present and analyze the information contained in the signal.

It is assumed that the possibility of reconstructing signals using continuous wavelets based on the derivatives of the Gaussian function is not guaranteed. With the need to have the inverse WT (or the reconstruction formula) most of the constraints imposed on the wavelet are related [1, 2, 8, 9, 23, 24, 59, 60, 65, 66, 72]. In the vast majority of applications used in the natural sciences, engineering, for reconstruction, coding of signals, discrete wavelets are used.

The chapter presents an algorithm for calculating the inverse continuous fast WT, which allows one to reconstruct the signal with high speed and accuracy.

2.2.1 The principle of inverse continuous WT in the frequency domain

In order to develop an algorithm for calculating the inverse continuous WT in the frequency domain, some changes are made in the formula (1.12) of the inverse continuous wavelet transform. The multiplier that is given in the literature in the form $\frac{1}{\sqrt{a}}$ is replaced by the multiplier $\frac{1}{a}$. The normalizing coefficient in formula (1.12) is replaced by another coefficient. The normalizing coefficient C in the developed algorithm is calculated from the analogue of Parseval's theorem for the wavelet coefficients:

$$\int S(t)S^*(t)\,dt = C^{-1}\iint W(a,b)\,W^*(a,b)\,\frac{dadb}{a^2} \tag{2.7}$$

After determining the normalizing coefficient C from (2.7), it is substituted into the formula

$$S(t) = C^{-1} \int_0^{\infty} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) W(a,b) \frac{dadb}{a^2} \qquad (2.8)$$

The theoretical basis for calculating the inverse continuous fast wavelet transform of the signal $S(t)$ in the frequency domain is the use of formulas (2.8) and (2.7). The inverse transform of the product of the wavelet spectrum $W(a,b)$ and the wavelet $\psi(t)$ calculates the integral with respect to the variable b. By summing the resulting integral with respect to the scale factor a, the reconstructed signal $S(t)$ is calculated.

2.2.2 Algorithm for numerical calculation of the inverse fast continuous WT

The algorithm for numerical calculation of the inverse continuous wavelet transform according to the formula (2.8) in the frequency domain includes the following steps:

1. The coefficients of the trigonometric series $a_1(n)$ of the wavelet spectrum $W(a,b)$ are calculated using the direct FFT according to the formula

$$a_1(n) = \frac{1}{N} \sum_{k=0}^{N-1} W(a,k) \cos\left(\frac{2\pi nk}{N}\right)$$

2. The coefficients of the trigonometric series $b_1(n)$ of the wavelet spectrum $W(a,b)$ are calculated using the direct FFT formula

$$b_1(n) = \frac{1}{N} \sum_{k=0}^{N-1} W(a,k) \sin\left(\frac{2\pi nk}{N}\right)$$

3. The coefficients of the trigonometric series $a_2(n)$ of the wavelet $\psi(t)$ are calculated using the direct FFT using the formula

$$a_2(n) = \frac{1}{N} \sum_{k=0}^{N-1} \psi(k) \cos\left(\frac{2\pi nk}{N}\right)$$

4. The coefficients of the trigonometric series $b_2(n)$ of the wavelet $\psi(t)$ are calculated using the direct FFT using the formula

$$b_2(n) = \frac{1}{N}\sum_{k=0}^{N-1} \psi(k) \sin\left(\frac{2\pi nk}{N}\right)$$

5. The complex conjugate spectrum is calculated using the formulas (2.1), (2.2). For even wavelets – by formulas (2.3), (2.4); for odd ones – by formulas (2.5), (2.6).

6. For an even wavelet, with the $M + 1$ inverse FT of the complex conjugate spectrum (2.3), (2.4), the formula (1.3) is calculated (matrix $\big((M + 1) \times N\big)s'_m(t)$), where $N = 2^m$ (the symbol ' does not mean differentiation)

$$s'_m(n) = \sum_{k=0}^{N-1} \big(c_1(k) + ic_2(k)\big) \exp\left(i\frac{2\pi nk}{N}\right)$$

7. For an odd wavelet, with the $M + 1$ inverse Fourier transforms of the complex conjugate spectrum (2.5), (2.6), using the formula (1.3), we calculate matrix $\big((M + 1) \times N\big)s'_m(t)$, where $N = 2^m$

$$s'_m(n) = \sum_{k=0}^{N-1} \big(c_1(k) + ic_2(k)\big) \exp\left(i\frac{2\pi nk}{N}\right)$$

8. Using the formula (2.7), the normalizing coefficient C is calculated.

9. By the formula

$$S(n) = C\sum_{m=0}^{M} s'_m(n)$$

the signal is being reconstructed.

Just as with direct WT, the coefficients $a_2(n), b_2(n)$ are calculated only for the wavelets, and not for the signal under study. Therefore, they can be calculated in advance, and the results of the calculations can be stored in RAM.

For even – numbered wavelets, clauses 4 and 7 are not met, and for odd-numbered ones, clauses 3 and 6 are not met.

When calculating the wavelet spectrum $W(a,b)$ before using the reconstruction formula (2.8) to synthesize the signal $S(t)$, the normalizing factor $\frac{1}{\sqrt{a}}$ is applied in the form of $\frac{1}{a}$ with a scale factor 2 to the power of m, and found $m+1$ wavelet coefficients $W(a,b)$ using the direct WT algorithm. Just as with direct WT, the amount of memory needed to store the wavelet coefficients is twice as small for each scale factor a due to being odd or even. Increasing the scale factor a to the power of $2m$ allows us to reduce the memory even more, since for large a, the wavelet coefficients are different from zero only for a narrow interval.

The structural diagram of the inverse continuous WT device is shown in Figure 2.2.2.1.

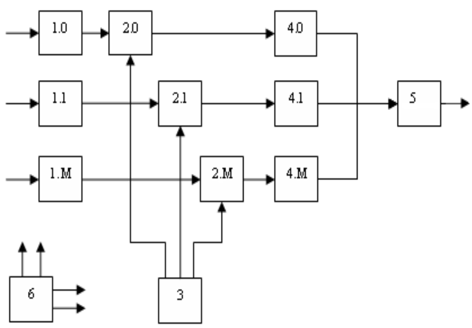


Figure 2.2.2.1. The structural diagram of the inverse wavelet transform: 1.0 – 1.M – FFT calculators; 2.0 – 2.M – multipliers; 3 – read-only memory; 4.0 – 4.M – inverse FFT calculators; 5 – adder; 6 – control device

The wavelet spectrum $W(m,n)$ enters the inputs of the FFT calculators (blocks 1.0 – 1.M), from the output of which the coefficients of the series $a_1(n), b_2(n)$ simultaneously enter the first inputs of the $M+1$ multipliers (blocks 2.0 – 2.M). From the ROM (block 3), the coefficients of the series $a_2(n)$ (for even), $b_2(n)$ (for odd) wavelets go to the second inputs of the $M+1$ multipliers (blocks 2.0 – 2.M), from the outputs of which the multiplication results go to the inputs of the inverse FFT calculators (blocks 4.0 – 4.M).

From the M outputs of the inverse FFT calculators (blocks 4.0 – 4.M), the coefficients are fed to the inputs of the adder (block 5), where the results of the inverse FT are summed, and from the output the result of the inverse WT signal $S(n)$ is removed.

The control device (block 6) synchronizes the operation of the blocks of FFT calculators (blocks 1.0 – 1. M), multipliers (blocks 2.0 – 2.M), inverse FFT calculators (blocks 4.0 – 0.M) and the adder (block 5).

This device allows us to select different types of wavelet functions with a scale factor of the 2nd degree M and with a normalizing factor of $\frac{1}{a}$, stored in ROM 1 (block 3), for signal reconstruction [31, 32, 34].

The results of acoustic signal synthesis suggest that the use of continuous WT is no worse than discrete wavelet synthesis. The use of continuous inverse WT in the frequency domain is preferable for a large sample of an acoustic signal, when the calculation speed is much higher than when calculating WT by direct numerical integration. Also, as the signal sample increases, the accuracy of the signal reconstruction increases. If the reconstructed acoustic signal is checked by ear, it is impossible to distinguish it from the original. For quantitative comparison, a measure of the type of correlation between the reconstructed signal and the original signal was used. The Pearson correlation coefficient is calculated using the well-known formula

$$r = \frac{\sum_{k=1}^{n}(x_k - x_c)(y_k - y_c)}{\sqrt{\sum_{k=1}^{n}(x_k - x_c)^2}\sqrt{\sum_{k=1}^{n}(y_k - y_c)^2}} \tag{2.9}$$

The dependence of the Pearson correlation coefficient between the synthesized acoustic signal and the original signal on the signal sample is shown in Figure 2.2.2.2. The acoustic signal sample is plotted on a logarithmic scale based on base two. A fragment of an acoustic signal with a duration of 62.5 ms was compared, which corresponds to the number of 500 samples.

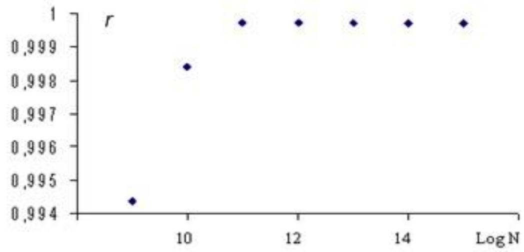


Figure 2.2.2.2. Dependence of the Pearson correlation coefficient from the number of counts

The results of the comparison show that for a signal with a number of samples greater than 2048, the Pearson correlation coefficient r almost does not change. If the same fragment of the acoustic signal is "stitched" from four fragments of 128 samples, then the Pearson correlation coefficient r decreases by 4.5%, i.e. on the graph this point is located even lower than for 512 samples.

To accurately reconstruct the $S(t)$ signal, it is necessary that the frequency band occupied by the signal is smaller than the frequency band occupied by the $m + 1$ wavelets used. It is also necessary that the Fourier image of the wavelet satisfies the relation [24]

$$F'(\nu) = \sum_{m \in Z} \left(\psi_m(a_0^m \nu)\right)^2$$

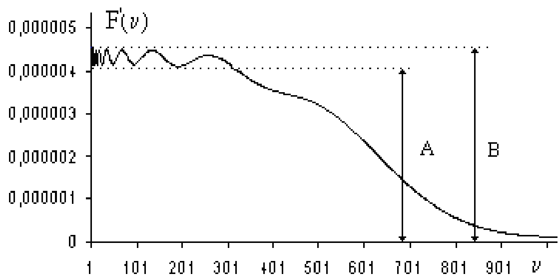


Figure 2.2.2.3. Frequency response of a set of *MHAT*-wavelets

Since for each value of the scale factor, the wavelet is a bandpass filter, the set (sum) of the wavelets is a block of filters with an uneven frequency response determined by the constants A and B (Figure 2.2.2.3). The smaller the difference between A and B, the smaller the recovery error. For a more accurate reconstruction, we can construct dual wavelets that reduce the difference between A and B. The Fourier image of the dual wavelet has the form [24]

$$\tilde{\psi}(\nu) = \frac{\psi(\nu)}{\sum_{m \in Z} \left(\psi_m(a_0^m \nu)\right)^2}$$

By the inverse Fourier transform, we obtain dual wavelets. A set of such wavelets is a block of filters with a uniform frequency response. The frequency response of a set of dual *MHAT* wavelets is shown in Figure 2.2.2.4.

Studies show that the use of dual wavelets can increase the correlation coefficient r.

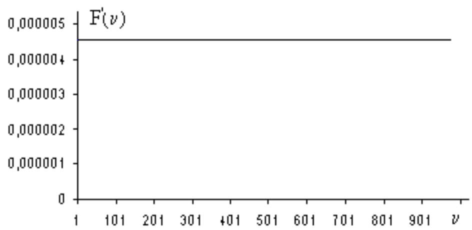


Figure 2.2.2.4. Frequency response of a set of dual *MHAT* wavelets

2.3 Profiling of the program for numerical calculation of the continuous wavelet transform in the frequency domain and calculations by direct numerical integration

The results of the WT speech signals were compared for direct calculation and FFT calculation. The time of calculation by direct numerical integration and calculation of WT using FFT was compared in *Visual Basic for Applications* in an Excel spreadsheet, as well as in *Visual C++*. In *Visual Basic for Applications*, the WT calculation time was measured for 50 cycles with different scale factors. A timer with a resolution of 1 s was used for profiling. Profiling is the measurement of the performance of both the entire program as a whole and its individual fragments. When using a 64-sample signal sample, the calculation time by direct numerical integration is less than the calculation time of WT using the FFT. When the signal sample is increased from 64 to 1024 samples, the time of direct calculation of WT increases by 35 times, and the time of calculation using FFT increases by 3 times and becomes less than the time of direct calculation of WT. When the sample is increased to 8192 samples, the conversion time using the FFT increases by 21 times, and with a direct calculation, the conversion time lasts several tens of minutes, which is unacceptable for processing an acoustic signal.

For more accurate measurements of small intervals in *Visual C++*, a real-time tag counter is used, which is accessed using the *RDTSC (ReaD from Time Stamp Counter*) assembly command. The *TSC (Time Stamp Counter)* is a 64-bit register whose contents are incremented with each clock cycle of the processor core. Each time a hardware reset occurs (with the *RESET* signal), the *TSC* counter starts from zero. The bit depth of the register provides a countdown without overflow for hundreds of years. The resolution of the counter is determined by the processor clock speed. The minimum time interval between the two measurements is equal to the inverse of the clock frequency. For a processor using a clock frequency of 2.54 GHz, the resolution is 0.39 ns. The clock frequency is determined using the *TSC* real-time counter.

The *RDTSC* command returns the number of clock cycles since the processor was started, putting the result in a pair of general-purpose registers *EDX:EAX*. To measure the WT calculation time, a program is written using the built-in *C++*assembler. The counter is graded using the standard OS function *Sleep* [79]. The *Sleep* function suspends the execution of the thread for 1000 ms if the function parameter is 1000. The *TSC* counter is read before calling the *Sleep* function and after returning from it. The difference between these readings is stored in the `t_time` variable. The number of machine clock cycles that have passed in one second of `t_time` is divided by 1000000 and stored in the `n_count` coefficient, which determines how many clock cycles are contained in one microsecond. Since the `t_time` variable is not equal to zero even in the absence of the *Sleep* function, we need to make adjustments to the `t_time` variable. After calculating the calibration coefficient `n_count`, the WT is profiled as follows. The *TSC* counter is read before calculating the WT and stored in the `t_time` variable after the conversion is completed. The resulting `t_time` value is divided by the calibration factor `n_count` and stored in a variable that shows the WT execution time in microseconds. The section of the program that performs WT using the FFT was profiled for a single scale factor, i.e. for a single cycle. The signal sample varied from 64 to 32768 samples. The execution time of 3-6 points of the forward WT algorithm was measured, because when calculating the wavelet spectrum for different scale coefficients a, most time is spent finding the complex conjugate spectrum and calculating the inverse FFT, since The Fourier coefficients of the signal are calculated once. For example, when calculating a wavelet spectrum with 50 scale factors, the numerical calculation time of 1 point is less than 1 % of the total time. Figure 2.3.1 shows the dependence of the time WT in the frequency domain on the number of samples.

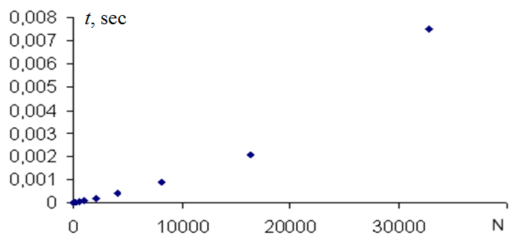


Figure 2.3.1. Dependence of the time WT in the frequency domain on the number of samples

The scale factor a for each sample is of great importance. For each reference, the time was measured 5 times, with a relative error of 4%. When profiling WT, we must take into account the compilation mode. In *Visual Studio*, the mode is determined by the project configuration. We can use the debug (*Win32 Debug*) and Release (*Win32 Release*) configuration. For example, when using the debug configuration, the WT time in the frequency domain for sampling 64 samples is 10.1 microseconds, and when using the output configuration, it is 4.9 microseconds. When using a debug configuration, the compiler inserts additional debug instructions. Profiling small fragments of programs leads to gross errors, since even in the absence of profiled code, the TSC real-time counter increases by a certain value. For the used processor, this value is on average equal to 700 clock cycles.

Figure 2.3.2 shows the dependence of the time WT in the frequency domain for a sample of 32768 samples on the scale factor a for three algorithms.

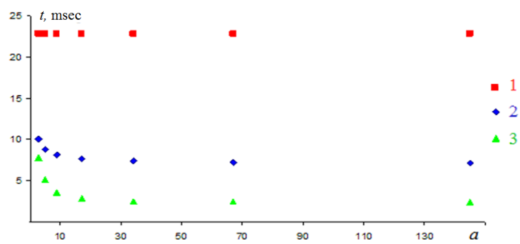


Figure 2.3.2. Dependence of the time WT in the frequency domain on the scale factor *a*

The first algorithm does not use the property of wavelet symmetry. The second algorithm uses the property of symmetry and the ability to calculate the wavelet spectrum for large scale coefficients a using a sample with a smaller number of samples. In Fig. 1.4.3 - 1.4.4 (Chapter 1), it can be seen that when the scale factor a increases, the width of the wavelet spectrum narrows and only the Fourier coefficients of the lower frequencies are different from zero. In this regard, it is sufficient to calculate non-zero coefficients. For example, for the scale factor $a = 477$ for 32768 samples, we can get the Fourier spectrum of the wavelet using a sample for 512 samples. To do this, we need to calculate the Fourier spectrum of the wavelet with a sample of 512 with a different scale factor a and a normalizing factor of $\frac{1}{\sqrt{a}}$. Then the wavelet spectrum with double precision matches the wavelet spectrum for the sample of 32768 samples. The FFT time for 512 samples, respectively, is many times less than for 32768 samples. Thus, the time WT for large scale coefficients a will be determined only by the time of the inverse FFT of the complex conjugate spectrum of the signal and the wavelet.

In the third algorithm, in addition to the second case, the possibility of reducing the time of the inverse FFT by reducing the number of multiplication operations is used. For large scale coefficients a, it is sufficient to calculate the wavelet spectrum over a certain offset interval b, because the correlation of the signal with the wavelet over a wide interval is calculated. In order to produce decimation, we must first pass the signal through a low-pass filter. When calculating WT in the frequency domain, obtaining a complex conjugate spectrum of a signal and a wavelet with a large value of the scale factor a is equivalent to passing the signal through a low-pass filter. In this regard, the inverse FFT is sufficient to calculate for certain values of the offsets b. These calculated wavelet coefficients are sufficient to reconstruct the signal. The implementation of the inverse FFT, as well as the direct FFT, is carried out by means of three nested loops, only in the inverse sequence. By changing the number of multiplication operations in the loop, we can calculate the wavelet coefficients through certain values of the b offsets. For example, for a signal with a sample of 32768 samples, when calculating WT with an offset of $b = 128$, the number of multiplication operations is reduced by almost 50 times compared to $b = 1$. Increasing the offset b by a factor of 2 leads to almost a twofold decrease in multiplication operations. For the offset $b = 4096$, the number of multiplication operations is reduced by 1203 times compared to $b = 1$. And the calculation time WT is reduced by 9.5 times compared to the first algorithm. In Fig. 2.3.2 it can be seen that a slight increase in the scale factor a for the initial section leads to a sharp decrease in the conversion time for the 2nd and 3rd algorithms.

The time for calculating the continuous WT in the frequency domain is reduced even when using an algorithm in which the signal and wavelet spectra are not multiplied by the formulas (2.1–2.6), but the inverse transformation of the signal spectrum is performed in such a way that the result is equivalent to a continuous WT. It is only necessary to know the pattern of using the signal spectrum. Since WT is a frequency analysis of a signal with a constant Q-factor, as opposed to the Fourier transform, we can choose the width of the signal spectrum so that the Q-factor is constant, and study it with different scale coefficients. This algorithm reduces the conversion time from 3 to 13.8 times, depending on the scale factor a.

It was shown that to perform the FFT of a signal with a sample of N samples, $\frac{N}{2}\log_2 N$ computational operations of multiplying complex numbers are necessary. To calculate WT with a sample of N samples with direct numerical integration, it is necessary to have N^2 computational multiplication operations. The reduction in the number of multiplication operations WT in the frequency domain relative to WT in the time domain is estimated by the ratio N^2 to $N\frac{\log_2 N}{2}$ equal to $\frac{2N}{\log_2 N}$. It is also necessary to take into account the reduction of addition operations by the same order. Therefore, for signals with a large sample, the speed of calculating WT in the frequency domain is much higher than when calculating in the time domain.

2.4 Multiple-scale signal analysis

The developed algorithms for numerical calculation of the forward and inverse fast continuous wavelet transform in the frequency domain allow us to represent the signal as a set of its successive approximations. The separation (decomposition) of signals into different types of components is the basis of multiple-scale analysis (MSA). The concept of "multiscale analysis" (multiscale) was formulated in 1986 by Malla and Meyer. Wavelets became more popular after Mall's introduction of the MSA concept for discrete wavelets. He was the first to use wavelets to encode images. The popularity of WT is largely due to the fact that it can be successfully used for image compression. The wavelets are directly related to the MSA of the signals. The idea of MSA is that the signal is decomposed according to the basis formed by shifts and multiple-scale copies of the wavelet function. When performing MSA, the signal space $L^2(R)$ is represented as a system of nested subspaces V_m. The decomposition of functions into wavelet series at a given resolution level m for a discrete WT is performed by the formula [9, 23, 66]

$$S(t) = \sum_{k} C_{mk}\,\varphi_{mk}(t) + \sum_{mk} D_{mk}\,\psi_{mk}(t)$$

where $\varphi_{mk}(t)$ is a scaling function, or scaling function, $\psi_{mk}(t)$ is a discrete wavelet. Coefficient values

$$C_{mk} = \int S(t)\,\varphi_{mk}(t)dt$$
$$D_{mk} = \int S(t)\psi_{mk}(t)\,dt$$

in practice, they are determined using discrete fast WT (Mall's algorithm).

The scaling function and the wavelet are obtained using functional equations. As a rule, they do not have an analytical expression. The mathematical foundations of multiple-scale analysis are described in many sources on discrete WT [9, 23, 66, 72, 105].

The algorithm developed in this paper for the inverse fast continuous WT allows any signal of duration $N = 2^m$ to be represented as

$$S(t) = \sum_{m=1}^{m+1} s_m\,(t)$$

where $s_m(t) = Cs'_m(t)$.

The constant C can be determined more easily using the corollary of formula (2.7) (Parseval's theorem). In the space of real functions, the energy density of the signal is

$$E_W(a,b) = W_l^2(a,b)$$

Local energy density at a point t_0 is

$$E_\delta(a,t_0) = W_l^2(a,t_0)$$

Then

$$S(t_0) = C\sum_{m=1}^{m+1} s'_m\,(t_0) \qquad (2.10)$$

The constant C calculated by formula (2.7) is the same as the constant found by formula (2.10). To avoid division by zero or multiplication by a negative number when calculating using the formula (2.10), it is better to calculate the constant C for the function at the maximum.

By analogy with the discrete WT, the entire signal space $L^2(R)$ as a whole can be represented as a sequence of nested closed subspaces of the corresponding levels m of the signal decomposition:

$$\ldots \subset W_m \subset W_{m-1} \ldots W_0$$

The "dimensions" of the subspaces expand continuously as the value of m decreases, and the union of all the subspaces in the limit gives the space $L^2(R)$. Wavelets in subspaces are formed by the scale transformation ψ_{0k}

$$\psi_{0k}(t) = \psi(t-k), k \in I$$

where k is the integer shift. A wavelet in the subspace m is represented by

$$\psi_{mk}(t) = a^m \psi(a^m t - k), k \in I$$

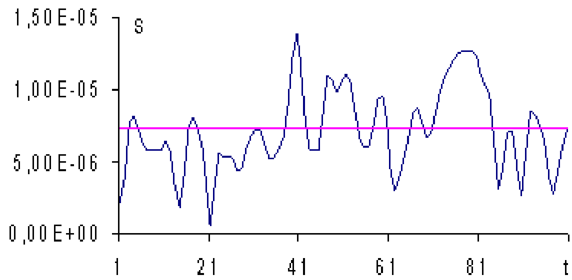
The value of parameter a is 2.

We form from $s_m(t)$ functions $s_m''(t)$ such that (the symbol $''$ does not mean double differentiation)

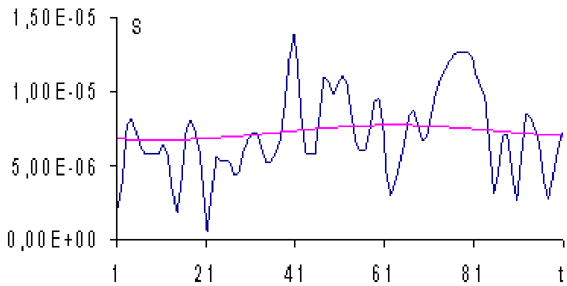
$$s_m''(t) = s_m(t)$$
$$s_{m-1}''(t) = s_m''(t) + s_{m-1}(t) \text{ etc.}$$

If the signal $s_{m-1}''(t)$ belongs to the space W_{m-1}, then at the same time it enters the space W_m, and along with it in this space is the signal $s_m''(t)$. Reducing the space number allows us to study more and more fine details and features of the signal with higher frequency components, i.e. to move from a rough approximation to a higher resolution approximation. Then the signal with the largest time resolution is $S(t) = s_0''(t)$.The variable m is called, in the same way as for a, the scale factor, or the level of analysis. If the value of m is large, then the function $s_m''(t)$ is a rough approximation of $S(t)$, in which there are no details. As the values of m decrease, the accuracy of the approximation increases.

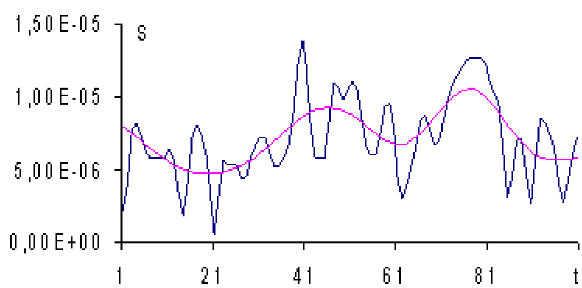
Let's illustrate this with an example. Figure 2.4.1 shows the graphs of the function $S(t)$ and its various approximations, i.e. the function $s''_m(t)$. Signal $S(t)$ is decomposed into 12 levels of decomposition. Figure 2.4.1 shows the $\frac{1}{20}$ part of the signal. In the literature on discrete WT, an m-step discrete WT is called a MSA. The maximum value of m is called the depth of decomposition of the signal.



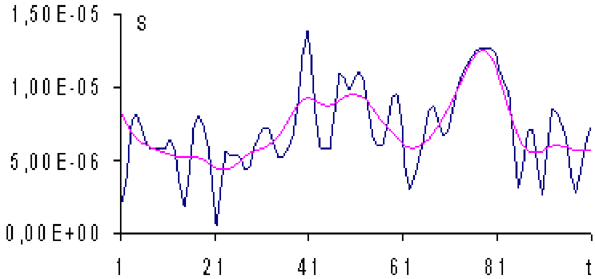
a



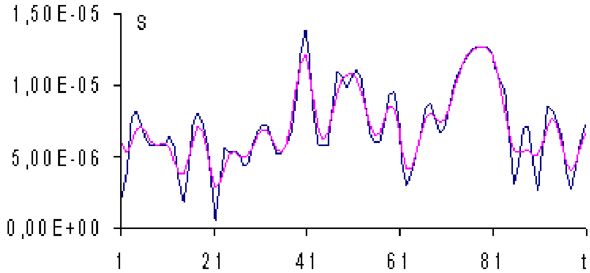
b



c



d



e

Figure 2.4.1. Decomposition of the signal into different levels

For the developed reconstruction algorithm, the depth of the signal decomposition is equal to the value $m+1$. In Figure 2.4.1, $a\ m = 11$ is the roughest approximation of the signal. Throughout the signal, $s''_m(t)$ has an almost constant value. In Figure 2.4.1,$b\ m = 6$. In the other graphs, the value of m decreases from 3 to 1. It can be seen that reducing the scale factor leads to a more detailed description of the signal. For $m = 0$, the Pearson correlation coefficient is 0.999. The reconstructed signal exactly follows the contours of the original, and it is impossible to distinguish them on the graph [31].

The signal can also be examined in inverse order, i.e. first present small-scale components, and then add larger details to these components, gradually approaching the original signal, as in Figure 2.4.1, in inverse order.

The graphs show that the signal can be approximated with some accuracy, depending on the limitation of the number of values of the scaling factor m. Then it is possible to analyze the function or signal at different levels of resolution, or scale, also for filtering and smoothing. For example, by removing small-scale functions, we can isolate a low-frequency useful signal, or, conversely, by removing large-scale functions, we can isolate a high-frequency signal. If we use FT to filter the signal, it is not possible to remove local noise, and WT allows us to remove it as well.

Unlike discrete WT, this algorithm is convenient and simple. There is no need to calculate approximating and detailing coefficients for scaling and wavelet functions. There is no need to find spline and packet wavelets, coiflets, and do all sorts of "tricks" (in the terminology of I. Daubechies).

2.5 Conclusions

1. Algorithms for forward and inverse continuous fast WT based on fast FT with an arbitrary choice of scaling coefficients are developed. Studies of the developed algorithms on real signals are carried out.

2. The developed algorithm of continuous WT in the frequency domain is much more efficient than calculating WT by direct numerical integration. The application of the developed algorithm of the inverse continuous WT allows one to reconstruct one-and two-dimensional signals with high speed and accuracy.

3. An algorithm for one- and two-dimensional multiple-scale analysis using fast continuous WT is developed. An algorithm for multiscale signal filtering using fast continuous WT is developed.

4. In contrast to discrete wavelets, signal compression using continuous wavelets is possible both in the region of the complex conjugate spectrum and in the region of the wavelet coefficients, which gives the developed signal reconstruction algorithm more options to choose from.

5. A comparison of the calculation time of continuous WT in the frequency domain with the calculation time by direct numerical integration shows that the developed algorithms for continuous fast WT in the frequency domain can increase the speed by four orders of magnitude compared to the algorithms for calculating continuous WT in the time domain.

3 APPLICATION OF ALGORITHMS FOR NUMERICAL CALCULATION OF FAST CONTINUOUS WAVELET TRANSFORM FOR SPEECH RECOGNITION IN RUSSIAN

3.1 Analysis of automatic speech command recognition systems

In connection with the development of computer technology, work is actively underway to create systems for automatic recognition of speech commands. Being the main component of any friendly interface between a machine and a person, the system of automatic recognition of speech commands can be integrated into various applications, for example, voice control systems, voice access to information resources, language training using a computer, assistance to the disabled, access to secret objects through voice verification or identification systems. It will be possible to use automatic speech command recognition systems more often if it becomes possible to control the human voice of various machines in real time.

Currently, most existing systems for automatic recognition of speech commands, both at the acoustic and semantic levels, use probabilistic models based on statistical language features. The most popular of them are based on the formal mathematical apparatus of hidden Markov models (HMM). The disadvantage of such models is that to train recognition systems based on methods of statistical (probabilistic) modeling of speech and language processes, huge amounts of speech data (acoustic, text) are required, which requires large financial resources and time. Also, the disadvantages of probabilistic models include the fact that the appearance of new words in the dictionary leads to a sharp increase in speech recognition errors.

In this regard, a promising approach to the analysis of an acoustic signal is the use of multiscale processing methods, in particular those based on WT, which allow analyzing and identifying dependencies, or tracking changes in the characteristics of an acoustic signal at different scales. Multiscale analysis (multi-resolution) allows us to get good time resolution (bad frequency) at high frequencies and good frequency resolution (bad time) at low frequencies. In other words, the low-frequency details of the signal are better localized in the frequency domain, and the high-frequency ones are better localized in the time domain. Obtaining additional information of different time scales and different signal resolution scales can improve the accuracy of speech recognition, so the development of new speech analysis methods based on the latest advances in digital signal processing, such as the theory of multiple-scale analysis and WT, is relevant [78, 80, 82, 86, 87, 90].

In addition to the possibility of multiscale signal representation, WT combines the advantages of spatial and frequency filtering methods. The hardware analog of WT is multi-channel, bandpass filtering of signals with a constant ratio of the filter bandwidth to the central frequency of the signal [90].

One of the main difficulties in speech recognition is the indefinite temporal organization of speech. Obviously, the accuracy of word recognition significantly depends on the accuracy of determining the boundaries of phonemes. The high efficiency of wavelets in the problems of filtering a non-stationary signal in comparison with the Fourier transform leads to the problems of creating wavelet algorithms that allow us to distinguish certain speech sounds in an acoustic signal at a certain scale factor and determine the boundaries between different speech sounds.

Multiscale signal processing with an arbitrary choice of scaling coefficients gives a more complete picture of the local features of a non-stationary signal, since using only integers as scaling coefficients leads to significant information loss when moving from one level of decomposition to another. The signal must be processed quickly, and therefore the development and research of fast algorithms for multiscale processing of an acoustic signal with an arbitrary choice of scaling coefficients remain relevant. The specifics of the development and use of fast algorithms for multiscale processing of an acoustic signal with an arbitrary choice of scaling coefficients are due to the structure of the human ear – in the processing of an audio signal, according to I. Daubechies [82], it transmits a wavelet image of the signal to the brain.

The automatic speech command recognition system is an element of the speech processing process, the purpose of which is to provide a convenient dialogue between the user and the machine. In a broad sense, these are systems that perform phonemic decoding of the speech acoustic signal when pronouncing speech messages in a free style, by an arbitrary speaker, without taking into account the problem orientation and restrictions on the volume of the dictionary. Therefore, instead of the term "speech command recognition", we will use the term "speech recognition".

Speech recognition as one of the components of artificial intelligence has long attracted researchers. Despite some progress made in this area, a number of issues remain unresolved. Combined with the problem of speech synthesis, they are very interesting issues for research. Sharing such systems is at the heart of a full-fledged voice interface. Relatively recently (about thirty years ago), speech recognition and synthesis subsystems were considered part of a single complex of speech interfaces. However, interest in the synthesis disappeared quite quickly.

First, developers did not solve even a tenth of the difficulties they encountered when creating recognition systems. Secondly, unlike speech recognition, speech synthesis does not demonstrate significant advantages over other means of information output from a computer. Instead of the passive use of speech systems "in bulk" (recognition separately, synthesis separately), the tasks of interactive human interaction with various systems are brought to the fore.

For a person, it is a dialogue, not a monologue, that is natural and familiar. Currently, in practice, a one-way voice interface is often used with an overwhelming advantage in the direction of speech recognition, despite the fact that the recognition process exceeds the complexity of the speech synthesis process. The preference for speech recognition is due to the urgent needs of human civilization, which largely depend on the natural features of the communicative functions of the human body. On the one hand, they provide rich opportunities for the rapid reception of information through the visual and auditory organs, on the other, the delivery of information is possible through verbal organs (verbal communication) and significantly inferior to them in the speed of non-linguistic communication. "Communication" with mechanical devices, equipment and devices shows even more lag in the speed of non-linguistic means of communication in comparison with verbal contact. For example, it is faster for us to give a command by voice, dictate a text, or communicate our decision by speech than to do it with our hands using the device controls. Of course, not every device is justified by this comparison, but many devices would be better controlled by voice.

The priority is the development of speech recognition systems. Traditionally, speech recognition is understood as the entire range of services for the transformation of a speech signal into a complete and functional set of defining information about the transmitted message. However, what is currently used in the voice interface of different devices, in principle, is not a set of such services. In fact, often we are talking about a system for recognizing sounds and phonemes, converting speech into text, recognizing and executing certain commands, extracting certain characteristics from speech (for example, identifying the speaker, determining his emotional state, gender, age, etc.) or certain sound patterns in the speech signal.

Formally, the process of speech recognition can be described in just a few phrases. The analog signal generated by the microphone is converted into numbers, and then the so-called phonemes are distinguished in speech, i.e. elementary fragments that make up all the spoken words. Then it is determined which word, which combination of phonemes corresponds. To recognize a word means to find it in the reference dictionary by the pronounced combination of phonemes. However, there are several problems with speech recognition by reference matching, among which the most common are the following:

1. Temporary changes in the characteristic images of speech. The reason for the changes is the different speed of pronouncing the same sounds, i.e. the inconstancy of the duration of the sounds. Even the same words spoken by the same person change in duration each time. If the same words are spoken by different people, their duration may differ even more.
2. The influence of the size of the organ of speech on images. The size of the speech organs in humans varies. In this regard, even if the words are pronounced by organs of the same shape, their resonant frequencies may differ. In the images, this manifests itself as an individual feature of a person. There is a problem of articulation conjugation, i.e. differences in the same sound due to the influence of different preceding and subsequent sounds, the problem of accent, which arises due to differences in the manner of speaking and in the living conditions of the speakers [11].

Thus, the sound recognition system faces difficulties of a fundamental nature. First, the features of the speaker's voice (timbre, noise inclusions associated with the structure of the speech tract), the distinctive manner of pronouncing certain sounds (acceleration or deceleration of the tempo, "swallowing" some sounds, a temporary shift in tonality, the unconscious insertion of insignificant sounds between words), specific articulation – all this leaves an imprint on the spectral composition of the speech signal. And the spectrum under such conditions changes significantly. Moreover, on its basis, the sound recognition system tries to distinguish the transitions of sound into sound, so it is difficult to form universal sound standards, comparisons which would not depend on unforeseen distortions in the spectrum. Secondly, a person usually does not pause between words, and when speaking together, the recognition task is also added to the task of selecting words from the speech stream, which is obviously more difficult. The most important case both for the practitioner and for the researcher is still fused free speech without limiting the lexicon (or even better, involving several people at the same time and against the background of noise).

The ultimate goal of most speech recognition research is speaker-independent speech fusion recognition systems, i.e. systems that could understand any person and could recognize every word of ordinary speech. Training a computer to understand human speech and "voice" various synthesized messages is still an extremely tempting task. To solve it means to make significant progress towards the implementation of a natural user interface. In addition, full-scale voice interaction between a person and a computer will allow a completely new approach to the problem of remote access to databases. A person will be able to receive voice information synthesized from the search results in the database using his phone. The construction of a speech interface involves three components [75]:

1. The first task is that the computer can "understand" what a person is saying to it, i.e. it must be able to extract useful information from a person's speech. At this stage, this task is reduced to extracting the semantic part of speech, the text (understanding such components as intonation, is not yet considered at all).
2. The second task is to make the computer understand the meaning of what is said. As long as the speech message consists of a standard

set of commands that are understandable to the computer, there is nothing complicated in its implementation.

3. The third task is that the computer can convert the information it operates with into a speech message that a person can understand.

Of the three problems listed, a fairly clear and definitive solution exists only for the third. In fact, speech synthesis is a purely mathematical problem that is currently solved at a fairly good level and in the near future, most likely, only its technical implementation will be improved.

The obstacle to the final solution of the first problem is that no one still knows exactly how to dissect our speech in order to extract from it those components that contain meaning. In the sound stream that we give out when talking, one cannot distinguish any individual letters or syllables.

Speech from a physical point of view consists of a sequence of speech sounds with pauses between their groups. At a normal rate of speech, pauses appear between fragments of phrases, since the words are pronounced together (although the ear, as a rule, perceives the words separately). In slow-motion speech, such as dictation, pauses can be made between words and even parts of them. Prepositions and conjunctions always sound together with the following word. Different people pronounce the same speech sound in different ways, and each person has his own way of pronouncing speech sounds. The pronunciation of speech sounds depends on the accent, neighboring sounds, etc. In any language, there is a certain set of sounds that is involved in the formation of the sound appearance of words. As a rule, the sound outside of speech does not matter, it acquires it only as an integral part of the word, helping to distinguish one word from another. The elements of this set of sounds are called phonemes. However, with all the variety in their pronunciation, they are physical realizations (utterances) of a limited number of generalized speech sounds, phonemes. By phonemes we mean only that part of the speech signal that creates the sensation of the elementary sound of natural language speech [17].

For example, in Russian there are 42 basic and 3 indefinite phonemes. Speech sounds are divided into voiced and deaf ones. Ringing sounds are formed with the participation of the vocal cords, which in this case are in a tense state. Under the pressure of air coming from the lungs, they periodically move apart, resulting in an intermittent flow of air. The air flow pulses generated by the vocal cords can be considered periodic with sufficient accuracy. The corresponding period of repetition of the pulses is called the period of the main tone of the voice (TV). The inverse of that, i.e. 1/TV is called the pitch frequency. If the ligaments are thin and strongly stressed, the period is short and the pitch frequency is high; for thick, weakly stressed ligaments, the pitch frequency is low. The pitch frequency for all voices is in the range of 70-450 Hz. When speaking, the frequency of the main tone continuously changes in accordance with the stress and emphasis of sounds and words, as well as for the manifestation of emotions (question, exclamation, surprise, etc.). The change in the frequency of the main tone is called intonation. Each person has his own range of pitch changes (usually it is a little more than an octave) and their own intonation, which is of great importance for the recognition of the speaker. The basic tone, intonation, oral handwriting and timbre of the voice are used to identify a person, and the degree of reliability of identification is higher than by fingerprints.

The formation of speech sounds occurs by giving commands to the muscles of the articulatory organs of speech from the speech center of the brain. When uttering speech sounds, either a tonal pulse signal, a noise signal, or both, pass through the speech apparatus. The speech tract is a complex acoustic filter with a number of resonances created by the cavities of the mouth, nose and nasopharynx, i.e. with the help of the articulatory organs of speech. As a result, a uniform tonal or noise spectrum turns into a spectrum with a number of maxima and minima. The maxima of the spectrum are called formants, and the zero dips are called antiformants. For each phoneme, the envelope of the spectrum has an individual and well-defined shape. When speech is spoken, its spectrum continuously changes, and formant transitions are formed. The frequency range of speech is in the range of 70-7000 Hz. Voiced speech sounds, especially vowels, have a high level of intensity, while deaf ones have the lowest level. When speech is spoken, its volume changes continuously. It changes especially sharply when uttering explosive sounds of speech. The dynamic range of speech levels is in the range of 35-45 dB. Vowel sounds of speech have an average duration of about 0.15 seconds, consonants-about 0.08 seconds (the sound of "п" ["p"] – about 30 ms).

Speech sounds are not equally informative. Thus, vowel sounds contain little information about the meaning of speech, and deaf consonants are the most informative. Therefore, speech intelligibility is reduced by the action of noise, primarily due to the masking of deaf sounds.

Almost all information about speech sounds is contained in the spectral envelope of speech and its temporal change (part of the information about speech sounds is contained in the transitions from the tonal spectrum to the noise spectrum and back – these transitions are used to learn about the change of ringing sounds to deaf sounds and back). All these changes occur slowly (in the pace of speech) [26]. In the perception of speech by a person, the mechanisms of associative analysis are used, while the sounds heard are not just analyzed and compared, but phonemes are collected into verbal images, the most suitable ones are selected not only by sound similarity, but also by intonation, emotional coloring, the context of the word, phrase, sentence and the entire text. Therefore, a person is able to recognize speech even with a large lack of carrier information. For example, a person is more demanding on the sound quality when listening to speech in a foreign language, with poor knowledge of it, than when perceiving native speech.

Humans can easily distinguish words by understanding the context in which they are spoken, whereas for computer systems to distinguish such close sets of sounds it is almost an unsolvable task.

To improve the reliability of speech recognition in conditions of a small signal-to-noise ratio, with a large volume of the dictionary, or in situations where it is necessary to work without pre-tuning to the speaker, attempts are made to use additional (non-speech) information about the speaker during recognition. Thus, the following types of information are used: visual information about the movement of the speaker's lips and jaw, entered into a computer using a video camera; information about the speaker's location, obtained using a video camera and used to orient the directional microphone; information about the movement of the speaker's lips and jaw, entered by cheaper methods (for example, using a reflective photo sensor); information about the speaker's emotional state (skin-galvanic reaction), used to refine the results of speech recognition; use of the bone conduction sound channel; registration of exhaled air.

The speech recognition system consists of two parts. These parts can be allocated into blocks or subroutines. The speech recognition system consists of an acoustic and a linguistic part [28, 75]. The linguistic part may include phonetic, phonological, morphological, syntactic, and semantic models of the language.

The acoustic model is responsible for the representation of the speech signal. The linguistic model interprets the information received from the acoustic model and is responsible for presenting the recognition result to the consumer.

3.1.1 Acoustic model

Both approaches have their advantages and disadvantages. When developing technical systems, the choice of approach is of paramount importance. There are two approaches to building an acoustic model: inventive and bionic. The first one is based on the results of the search for the mechanism of functioning of the acoustic model. In the second approach, the developer tries to understand and simulate the operation of natural systems. The acoustic signal processing sequence includes these steps:

1. Convert the input speech signal into a set of acoustic parameters. Typically, the audio signal is divided into windows of the same length and converted to the frequency domain using a discrete Fourier transform or a more complex transformation, after which the frequency parameters are factorized to reduce the dimension.
2. Reduction of the acoustic waveform to the internal alphabet of the reference elements. If the dictionary of the recognition system contains a larger number of words, in order to save memory, it is advisable to consider not words, but the corresponding phonemic elements as the standards of the recognition system. The set of such reference elements forms a phonetic codebook.
3. Recognize a sequence of phonemes and convert it to text. After determining the probable sequence of reference elements in the input signal, it is necessary to restore an unknown sequence of phonemes, which is a transcription of one of the words of the dictionary. This problem is solved using the method of dynamic programming, when at each moment the most probable expected sequence of phonemes in the signal from the beginning of the word to this moment is determined. If the acoustic parameters are

converted to probabilistic ones using a codebook, and the transcription standards of words are set in the form of probabilistic automata, hidden Markov chains (HMM) are usually used for recognition [77]. The use of HMM for speech recognition is based on the following provisions:

a) Speech can be divided into segments (states), within which the speech signal can be considered stationary. The transition between these states is instantaneous.
b) The probability of the observation symbol generated by the model depends only on the current state of the model and does not depend on the previous generated symbols. In fact, neither of these two assumptions is true for the speech signal. Nevertheless, standard HMMs are the basis for most modern speech recognition systems [26, 28].

In the case of setting word standards in the form of a sequence of acoustic parameter values without using a code book (usually for small dictionaries), a different model of speech recognition is used: dynamic programming, called dynamic time warping.

3.1.2 Linguistic model

The linguistic block is divided into the following levels: phonetic, phonological, morphological, lexical, syntactic, and semantic. All tiers are a priori information about the structure of natural language, and as known, any a priori information about the subject of interest increases the chances of making the right decision. Natural language carries highly structured information, which means that each natural language may require its own unique linguistic model. In accordance with this model, at the first – phonetic-level, the input (for the linguistic block) representation of speech is converted into a sequence of phonemes as the smallest units of the language. It is generally assumed that in a real speech signal, only allophones can be detected – variants of phonemes that depend on the sound environment. At the phonological level, restrictions are imposed on the combinatorics of phonemes (allophones). Not all combinations of phonemes (allophones) occur, and those that do occur have a different probability of occurrence, depending also on the environment. To describe this situation, the mathematical apparatus of Markov chains is used.

Further, at the morphological level, they operate with speech units of a higher level than the phoneme. Sometimes they are called morphemes. They impose a restriction already on the structure of the word, obeying the laws of the simulated natural language. The lexical layer covers the words and word forms of a particular natural language, i.e. the vocabulary of the language, also providing important a priori information about which words are possible for a given natural language. Semantics establishes relations between the objects of reality and the words that denote them. It is the highest level of language. With the help of semantic relations, the human intellect compresses the speech message into a system of images, concepts that represent the essence of the speech message.

Hence, there is the conclusion that the system must be "smart". The better the model of semantic connections (the equivalent of a "mental image system") is constructed, the more likely it is to correctly recognize speech.

Currently available speech recognition systems are based on the collection of all available (sometimes even redundant) information necessary for word recognition. Nevertheless, at present, even when recognizing small messages of normal speech, it is still impossible to perform a direct transformation into linguistic symbols after receiving a variety of real signals, which is the desired result.

There is a class of understanding models, the so-called frame models (or a subset of them – situational models), implemented on neural network algorithms and using relational knowledge bases. Currently, relational knowledge bases are becoming increasingly popular, the essence of which is the existence of certain internal latent relationships that can be used depending on the query. Neural network algorithms are used for their successful implementation in understanding models. These databases are complex systems with hidden structures that can be used to interpret a particular frame. For example, such a database may contain some rules describing the situation, associative or semantic links. This class of understanding models is more promising because it implements implicit relationships between knowledge base elements. The appearance of new information does not lead to a stoppage of the algorithm, but to the search for associative links due to the study of the context and can lead to the formation of a new frame [17].

These speech recognition systems can be classified according to different characteristics: purpose, type of speech, consumer qualities, type of structural unit, mechanism of functioning, and method of distinguishing features.

By appointment, they are allocated as:

- text dictation systems
- speech-to-speech conversion systems
- command systems

According to the type of speech, there are:

- recognizers of isolated words
- merge speech recognizers

According to consumer qualities, systems are distinguished as:

- voice-dependent
- voice-independent

By type of structural unit:

- allophone
- phoneme
- diphon
- trifon
- word
- phrase

According to the mechanism of functioning, there are:

- the simplest (correlation) detectors

- probabilistic network models of decision-making (HMM, dynamic programming, neural network methods).

According to the method of feature selection, there is:

- spectral analysis
- cepstral analysis
- wavelet analysis
- the coding of linear prediction coefficients

3.2 Speech command recognition software

The creation of automatic speech recognition systems is one of the most relevant areas of development of modern computer technologies. Currently, all over the world, work is underway to create more natural means of communication with a computer, and foremost is the speech input of information into a computer. The problem of speech input is complicated by a number of factors: the difference of languages, the specifics of pronunciation, noise, accents, accents, etc. Despite the fact that the first developments in the field of speech recognition date back to the 1920s, the first system was created only in 1952 by *Bell Laboratories* (today it is part of *Lucent Technologies*). At the moment, there are not dozens, but hundreds of research teams working in this direction in scientific and educational institutions, as well as in large corporations. This can be judged by such international forums of scientists and specialists in the field of speech technologies as *ICASSP, EuroSpeech, ICPHS*, etc. For a number of years, voice navigators, or command recognition systems, have been successfully used in various fields of activity. In expensive cars, such as Infinity and Jaguar, oral control of the control panel has been used for several years: the radio, temperature control and navigation system "understand" the voice of the owner of the car and unquestioningly listen to the owner. Voice recognition technology is increasingly being used in middle-class cars.

Currently, users are far from being able to conduct a reasonable dialogue with the computer. However, speech recognition tools for personal computers (PCs) have already appeared, which have reached a level of development sufficient to change the way people and machines interact. The simplest of modern products, capable of responding to words, allow us to control the execution of standard application programs in *MS Windows* and *Linux* systems by means of oral commands (instead of a mouse and keyboard). More complex systems with large dictionaries allow one to perform oral text input or serve as the basis for your own applied speech recognition systems.

All speech recognition software products are divided into three main types: speech navigators, speech text input tools, and development tools. Among the products of these three types, navigation programs are the most numerous. They allow one to give verbal commands to control the operation of the system, for example, the launch and execution of application programs. So, in computers running the *Windows* operating system, one can use voice control to work with individual programs and groups of programs, to copy or move files, etc. A striking example is the *IBM ViaVoice1* and *Microsoft Voice2* programs. For *Linux*, use the program *CVoiceControl3*, which is distributed in the source code. On low-power computers, the developers of such packages require the mandatory use of specialized coprocessor boards. An example of this group of dictation systems is the *Dragon Dictate4* complex of *Dragon Systems* for *MS Windows* and *SPHINX5* for *Linux*. There is a localized version of the *Dragon Dictate program: Gorynych*.

Development tools allow us to create both universal and specialized application programs that use speech recognition methods. Basically, these systems are aimed at programmers who create programs in *C++* and *MS Visual Basic*, and contain basic recognition programs, as well as application program interfaces (*APIs*) and the programming language libraries necessary for their use. One example is *Microsoft Speech API6*.

Currently, well-known companies, such as *Microsoft, IBM, Dragon Systems, Lernout&Hauspie,* and *Digalo*, supply speech recognition and synthesis engines to the software market. The term "engine" refers to a set of software tools that are specially designed and optimized to perform a specific operation. These engines can work with all standard European languages: English, German, French, Italian, Spanish, etc. Support for Russian is still quite rare.

Speech recognition is finding more and more new applications, ranging from applications that convert speech information into text, and ending with on-board car control devices [11]. All the variety of existing speech recognition systems can be divided into the following groups:

1. Software cores for hardware implementations of speech recognition systems.
2. Sets of libraries and utilities for developing applications that use speech recognition.
3. Independent user applications that perform speech control and speech-to-text conversion.
4. Specialized applications that use speech recognition.
5. Devices that perform recognition at the hardware level.
6. Theoretical research and development.

3.2.1 Software cores for hardware implementations of speech recognition systems

At the heart of any speech technology is the so-called "*engine*", or the core of the program, a set of data and rules by which data is processed. Depending on the purpose of this core, the *TTS* and *ASR engine* are distinguished. The *TTS (Text-to-Speech)* engine provides text-to-speech synthesis, and the *ASR (Automatic Speech Recognition) engine* provides speech recognition.

There are several major manufacturers involved in the creation of *ASR* cores, including companies such as *SPIRIT, Advanced Recognition Technologies*, and *IBM*.

The modern market offers automatic speech recognition systems (ASRS) of various companies.

Aculab. Recognition accuracy is 97%. It is a voice-independent system that supports the most common languages, such as British and American English, French, German, Italian, and North American Spanish. The dictionary can be configured for any of these languages, but it is not possible to use multiple languages simultaneously as part of a single dictionary.

Babear SDK Version 3.0. It is a voice-independent system that does not require training for a specific user. The system supports the following languages: English, Spanish, German, French, Danish, Swedish, Turkish, Greek, Icelandic and Arabic.

Loquendo ASR. It is a voice-independent system optimized for use in telephony. It is possible to recognize individual words and speech and search for keywords (with a dictionary of up to 500 words).

LumenVox. It is a voice-independent system that does not require training, but after adapting to a specific user, the recognition results become much better: the recognition accuracy exceeds 90%.

Nuance. The system is optimized for the lowest consumption of memory and other system resources. The recognition accuracy is up to 96%, and remains high even in a noisy room. There is a possibility of self-learning of the system and its adjustment for each user.

SPIRIT. The *SPIRIT ASR Engine* speech recognition system is designed for a wide range of practical tasks. Such tasks include, for example, the organization of an automatic call center (voice control of the menu system, dialing a *PIN* code and phone number), security systems, voice control systems, etc. The system is capable of real-time voice-independent recognition of chains of spoken words and individual speech commands, including in noise conditions with a signal-to-noise ratio of up to +5 dB.

In the *SPIRIT ASR Engine* system, both well-known solutions, such as HMM, and non-standard approaches were implemented, which significantly increased the reliability of recognition in real acoustic conditions. The use of HMM is currently the most popular and successfully applied approach to the problem of speech recognition. Standard HMMs are the basis for most modern speech recognition systems.

SPIRIT develops software for digital telephony, speech compression, speaker identification, *VoIP* and *GPS* technologies. *SPIRIT's ASR engine* is designed for speech command recognition and is used in various applications, such as voice control of devices, voice dialing in *hands-free* devices, and entering personal identification codes (*PINS*) in security systems. This core is built into any DSP or RISC platform and is delivered as object code.

VoiceWare. The system can work in both voice-dependent and voice-independent modes, so special training of the system for working with a specific user is not required. *VoiceWare* supports English and Korean.

IBM Corporation has been working on automatic speech recognition for more than 30 years and has achieved great success in this field. So, the company "*ProVox Technologies*" based on the software core *ViaVoice®*

from IBM has created a system for dictating reports of radiologists *VoxReports*. According to the results of testing, this system with an accuracy of 95-98% recognizes the combined speech of a normal tempo (up to 180 words per minute), regardless of the speaker. However, the system's vocabulary is limited to a set of specific medical terms.

There is an agreement between *Opera Software* and *IBM* on the integration of *Embedded ViaVoice* speech recognition technology into *Opera* browsers. Using *Embedded ViaVoice* will allow users to control the browser not only with the mouse and keyboard, but also with their voice.

Speech recognition technology is increasingly used in mobile communications. So, the company "*Advanced Recognition Technologies*" has created a *smARTspeak NG* system that is embedded in mobile phones. Currently, the *smARTspeak NG* system is used in non-keyboard phones of *Siemens*, *Panasonic* phones of the *TDMA* standard in the United States and other countries.

Sakrament ASR Engine is a software development of the Belarusian company "*Sacrament*", designed for use in various hardware systems and software applications that use speech recognition technologies. Declared characteristics: recognition accuracy of 95-98%; voice independence; language independence; recognition of merged speech in the form of expressions and small sentences. However, there is no possibility of training in this system – additional dictionaries are created by order of the company "*Sacrament*" itself [81, 88, 91].

3.2.2 Sets of libraries, utilities for developing applications that use speech recognition

With the development of speech technologies and the increasing adoption of mobile devices, the idea of using speech control in building network applications arose. To do this, it was necessary to develop a unified standard for the integration of speech technologies.

One of the open standards based on the *XML* language-*VoiceXML (Voice eXtensible Markup Language)*, the first version was published in May 2000 by the international *World Wide Web Consortium (W3 Consortium)* – is intended for the development of *Interactive Voice Response (IVR)* applications for managing media resources. The purpose of creating the standard is to use all the advantages of web programming in the development of *IVR* applications.

Various companies are developing packages for creating speech applications, the so-called *Software Development Kit (SDK)*, that support a particular standard. For example, *Philips* has created the *Speech SDK*. This package supports the *VoiceXML* specification and is designed to communicate with the *C/C++ API*.

CompTek and *Philips* jointly created *SpeechPearl*, a product that is a set of software modules, libraries, and utilities for developing speech recognition systems with Russian support for phone applications [63, 78, 82, 86, 105].

3.2.3 Devices that perform recognition at the hardware level

To use speech recognition functions in various devices, robots, and toys, hardware methods for solving this problem are being developed. For example, the American company "*Sensory Inc*" has developed the *Voice Direct™364* integrated circuit, which performs voice-dependent recognition of a small number of commands (about 60) after preliminary training. Before starting operation, the module must be trained in all the commands used in the work [77, 89].

3.2.4 Theoretical research and development

Many research groups around the world are engaged in the development of the theoretical basis in the field of speech technologies. First of all, these are large corporations such as IBM, Intel, Microsoft, and AT&T. These companies have been engaged in the theory of recognition for more than a decade and are recognized authorities in this field. From all the variety of scientific developments, we will consider in detail the work of domestic research groups.

The Laboratory of Automated Queuing Systems of the Institute of Management Problems of the Russian Academy of Sciences has been conducting research in the field of speech recognition for more than 30 years. The main scientific and practical direction of the laboratory's activity is currently the use of computer recognition of merged speech in public service systems with the possibility of using Russian and other languages.

Mathematical models have been developed to describe the processes in speech recognition systems.

The Institute of System Analysis of the Russian Academy of Sciences is engaged in research in the field of speech recognition, which is focused on solving the following tasks: the development of the theoretical base, the development and software implementation of methods for automatic analysis of speech signals in real time. The principal novelty of the proposed directions consists of the use of the island neural network analysis of the speech signal in correlation with the allocation of stable features and the application of phonological and other "engineering" knowledge (i.e. the use of speech signals) and knowledge based on a meaningful study of the process of utterance or the process of perception about the fine structure of the speech signal.

Since 1996, "СТЭЛ–Компьютерные Системы" ["Stel - Computer Systems"], in cooperation with leading specialists of the Faculty of Philology of the Lomonosov Moscow State University, the Computing Center of the Russian Academy of Sciences and a number of other organizations, has been implementing a project to create a prototype of a voice-independent system for recognizing Russian speech. From a methodological point of view, the project is based on the application of modern methods of speech signal processing and the HMM apparatus for describing the phonetic and semantic-syntactic patterns of Russian [62, 77, 78, 79, 80, 83, 84, 85, 86, 87, 89, 90, 92].

Developers of speech recognition systems face the various tasks. The most popular approach in recent years is based on the priority of solving applied problems. The popularity of this approach is determined by the active development of the market of so-called speech technologies, i.e. systems for automatic speech recognition, synthesis and compression. At the same time, natural science knowledge about the object under study is replaced by utilitarian goals. The corresponding utilitarian approach has a right to exist, but, of course, has nothing to do with science. The most promising areas of use of speech technology devices are related to their interaction with a person and require, in fact, the repetition in the technological system of the methods of working with speech, speech information used by a person.

We can give several examples of the use of methods in speech technologies that are quite effective, but lead science away from actual knowledge. In the field of automatic recognition, such an example is the use of HMM. It is obvious that speech communication is built according to some rather complex rules, about which we do not know everything, or rather, we know very little. Using a probabilistic model (HMM) is an attempt not to learn these rules, but to replace them with random iteration. The disadvantages of this method are obvious: the lack of noise immunity, the need for large statistics [88].

At the moment, the most difficult elements in the construction of a speech recognition system are not recognizing algorithms, their detailed descriptions can be found in monographs and patents, and the construction of an acoustic model of the language and the initial training of standards of dictionary words, most often probabilistic Markov automata. As a rule, to build a reliable model of a particular language from a probabilistic point of view, it is necessary to conduct long-term work of large teams to collect and analyze the acoustic data of a huge number of native speakers of a given language. It is necessary to carefully consider all types of voices and accents available to native speakers, and for each variety of voice and accent to get a reliable assessment of the elements of the code book of this language. An equally difficult task is the construction of word standards. To do this, it is necessary that each word of the dictionary (and there can be about 100,000 of them) is pronounced by each representative of this type of speaker several dozen times, otherwise the resulting probabilistic automaton will be statistically unreliable.

Finally, for the successful application of syntactic and semantic dependencies between the words of sentences, it is necessary to build some grammar that reflects the structure of the sentences of the language to some extent. In connection with the above, there is a need to transfer computer speech recognition systems, currently working mainly on models of languages of the Germanic group (English, German, French, Italian, etc.), to other groups of languages, such as Slavic or Asian. In fact, the developers of recognition systems face the task of building such systems anew, almost from scratch, since the lion's share of time and money in the development of a new system falls on the process of building a reliable acoustic model, word standards and language grammar. Russian recognition systems will not only have to build a new acoustic model and train a dictionary of the most commonly used Russian words, but also to build models of Russian grammar, which, as it is easy to assume, will be an order of magnitude more complex than the trigram model, which is now used to set the grammar of English. It is because of this that there is no decent dictation system for Russian texts on the market and it is unlikely to appear in the near future. Working on its construction requires too much financial investment [19].

3.3 Extraction of phoneme information features using fast continuous wavelet transform

To study the speech signal, we use wavelets based on the Gaussian derivative. To record, display, play, edit samples and save them to a text file, use the audio editor: the sampling rate of the acoustic signal is 8000 Hz, the resolution is 16 bits, and the recording mode is mono. The stored data is used to convert the studied signal $S(t)$ to the wavelet spectrum $W(a, b)$. The duration of the speech signal is four seconds.

To calculate the wavelet spectrum of a speech signal, the formula of the continuous wavelet transform is used

$$W(a, b) = \frac{1}{\sqrt{a}} \sum_{-\infty}^{\infty} S(t)\, \psi\left(\frac{t-b}{a}\right) dt$$

The Fourier transform is used to calculate the Fourier spectrum of the wavelet spectrum segments

$$F(\nu) = \int_{-\infty}^{\infty} f(t)\, e^{-i2\pi\nu} t dt$$

The following algorithm is used to form phoneme standards. The wavelet coefficients of $W(1, b)$, $W(2, b)$, $W(4, b)$, $W(6, b)$, $W(8, b)$, $W(20, b)$, and $W(50, b)$ words are calculated, where b varies from 1 to 32768. The resulting wavelet coefficients (functions) $W(1, b)$, $W(2, b)$, $W(4, b)$, $W(6, b)$, $W(8, b)$, $W(20, b)$, and $W\ (50, b)$ are divided into segments of fixed duration ($n = 128$), which corresponds to 16 ms. The number of segments is 256. The duration of the segment is not less than the duration of the pronunciation of phonemes, but exceeds the maximum possible period of the main tone of the phonemes. The wavelet coefficients are calculated for a sampled and level-quantized speech signal, so a discrete version of the continuous wavelet transform is used to calculate the wavelet spectrum [31].

Figures 3.3.1 and 3.3.2 show two segments of the function $W(3, b)$ of the phoneme "a" and "э" ["e"].

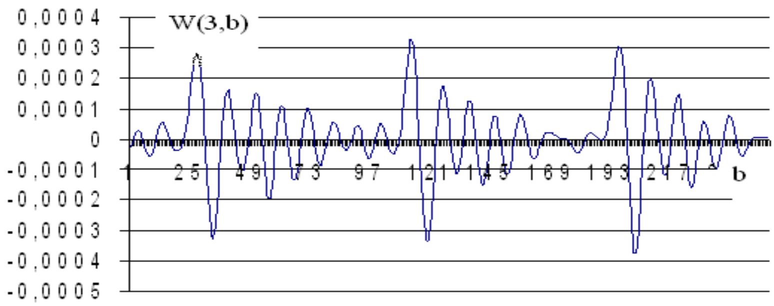


Figure 3.3.1. Two segments of the function $W(3, b)$ of phoneme "a"

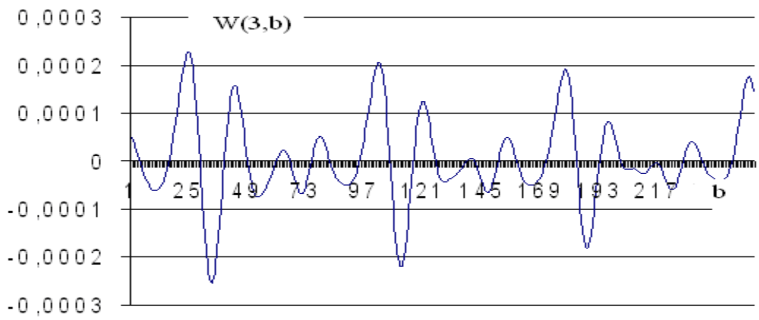


Figure 3.3.2. Two segments of the function $W(3, b)$ of the phoneme "э"

The wavelet spectra $W(a, b)$ of the speech signal were tested by ear. To listen to the wavelet spectra $W(a, b)$, an algorithm is used that includes the following steps:

1. The maximum of the function $W(a, b)$ is found in the observation window.

2. The function $W(a, b)$ is normalized.
3. The normalized function is multiplied by a number that is acceptable in terms of volume.
4. The resulting function is rounded to integers and submitted for listening.

Listening to the wavelet spectra $W(1, b)$, $W(2, b)$, $W(4, b)$, $W(6, b)$, $W(8, b)$, $W(20, b)$ and $W(50, b)$ shows that they sound almost identical to the acoustic signal $S(t)$ under study. For example, the function $W(3, b)$ of phoneme a sounds like phoneme a with a different timbre. The wavelet spectrum of words and sentences, sounds similar to the original word or sentence. Here we see that even with a very strong change in the form of the speech signal, we can hear and distinguish phonemes and words. Apparently, the hearing does not respond to the shape of the signal, but to the number of vibrations over a certain period. Therefore, compression of the acoustic signal by tens of times is quite possible. When the scale factor is increased to more than 20 units for a sample of 32,768 samples, each phoneme loses its identity and sounds like the phoneme "y". In each segment, the Fourier coefficients $d(i)$, $e(i)$ of the functions $W(1, b)$ and $W(2, b)$ are calculated using the FFT. The simplest weight function is used (window) Dirichlet. The effect on the spectrum of other weight functions (Hamming, Bartlett, Hanna, etc.) was not considered. Thus, an adequate mathematical model of the speech signal in the segment is:

$$d(n) = \frac{1}{M}\sum_{k=0}^{M-1} W\,(a, k)\cos\left(\frac{2\pi nk}{M}\right)$$

$$e(n) = \frac{1}{M}\sum_{k=0}^{M-1} W\,(a, k)\sin\left(\frac{2\pi nk}{M}\right)$$

Substituting the result of the inverse Fourier transform of the complex conjugate spectrum of the speech signal and the wavelet instead of the wavelet spectrum, we obtain

$$d(n) = \frac{1}{M}\sum_{k=0}^{M-1}\left(\sum_{j=0}^{N-1}\left(c_1(j) + ic_2(j)\exp\left(i\frac{2\pi kj}{N}\right)\right)\cos\left(\frac{2\pi kj}{N}\right)\right)$$

$$d(n) = \frac{1}{M}\sum_{k=0}^{M-1}\left(\sum_{j=0}^{N-1}\left(c_1(j) + ic_2(j)\exp\left(i\frac{2\pi kj}{N}\right)\right)\sin\left(\frac{2\pi kj}{N}\right)\right)$$

By the formula

$$F(i) = d^2(i) + e^2(i)$$

the Fourier spectrum of the functions $W(1,b)$, $W(2,b)$ of the phonemes of the Russian alphabet is calculated. Figures 3.3.3, 3.3.4 show the Fourier spectra of the segments of the function $W(1,b)$ of phonemes "a" and "э".

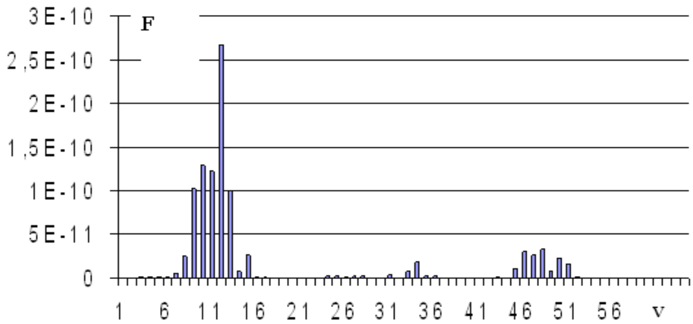


Figure 3.3.3. Fourier spectrum of the function $W(1,b)$ of phoneme "a"

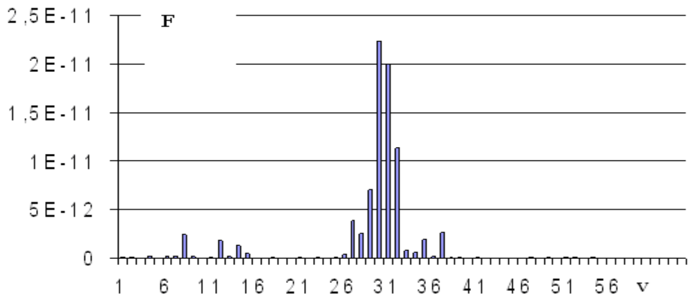


Figure 3.3.4. Fourier spectrum of the function $W(1,b)$ of the phoneme "э"

The sounds (phonemes) selected from the words are examined separately. Usually, if the spectrum of an audio signal is found without using WT, the resulting spectrum undergoes various transformations, for example, a logarithmic change in scale (both in the amplitude space and in the frequency space). This allows us to take into account some features of the speech signal – a decrease in the information content of high-frequency sections of the spectrum, the logarithmic sensitivity of the human ear, etc. Since the low-frequency region contains more information than the high-frequency region, this reduces the number of parameters that receive information from the high-frequency region. Or it compresses the high-frequency region of the spectrum in the frequency space. The most common method is logarithmic compression

$$m = f_{\max} \log(kf + 1)$$

where f is the frequency in the spectrum, Hz, k is the compression ratio, and m is the frequency in the newly compressed frequency space [31].

In this monograph, the spectrum was not compressed, because at WT of the speech signal for small scale coefficients, the information content is also large for the high-frequency region of the spectrum. A flowchart of the algorithm for forming a phoneme database is shown in Figure 3.4.7.3. For the phonemes of the Russian alphabet, a database is created with a set of characteristic frequencies (range) of the function segments $W(1, b)$, $W(2, b)$, $W(4, b)$, $W(6, b)$, $W(8, b)$, $W(20, b)$ and $W(50, b)$. Also, as a characteristic feature, the fractal dimension of phonemes is used, which is accepted as certain frequencies. The lower and upper limits of the range of characteristic frequencies obtained by repeated pronunciation of Russian words are used as phoneme standards for speech recognition. It is possible to update (expand the frequency range) the database with phoneme standards so that speech sounds are distinguished when new words are added to the dictionary (a database of individual words), i.e. a system-training algorithm is developed. To identify the elements of speech, the following algorithms for comparing the acoustic signal with phoneme standards are studied.

The central frequencies, the average energies of the normalized spectrum of the segments of the functions $W(1, b)$ and $W(2, b)$ are calculated for the intervals 0-20, 21-64, 0-64 Hz (in conventional units) using the formulas

$$\nu = \frac{\sum_{i=1}^{n} i\,F(i)}{\sum_{i=1}^{n} F(i)}$$
$$E = \frac{\sum_{i=1}^{n} F(i)}{n}$$

In the sliding mode, the number of local maxima of the functions $W(3,b), W(7,b)$ in the segment and the average number of local maxima in the segment are calculated.

Due to the fact that the number of local maxima of the functions $W(3,b), W(7,b)$ in the segment does not depend on the shift for the stationary signal, but on which phoneme is currently present in the acoustic signal, the calculation of local maxima is similar to dynamic spectral analysis. The only difference is that the execution time of this algorithm is many times less than the spectral processing with an observation step equal to one. The block diagram of the phoneme identification device is shown in Figure 3.4.7.4. The algorithm for identifying phonemes is basically similar to the algorithm for forming a database of phonemes, they differ only in the last block, where the phonemes of the studied speech signal are compared with the standards of the phonemes of the database.

Phoneme "a" is identified in segments using the following algorithm:

```
for i = 1 to n
b = 11
if (x(i,1) ≥ u(b,1)) and (x(i,1) ≤ u(b,2)) and
(x(i,2) ≥ u(b,3)) and (x(i,2) ≤ u(b,4)) and
(x(i,3) ≥ u(b,5)) and (x(i,3) ≤ u(b,6)) and
(x(i,4) ≥ u(b,7)) and (x(i,4) ≤ u(b,8)) and
(x(i,5) ≥ u(b,9)) and (x(i,5) ≤ u(b,10)) and
(x(i,6) ≥ u(b,11)) and (x(i,6) ≤ u(b,12)) and
(x(i,7) ≥ u(b,13)) and (x(i,7) ≤ u(b,14)) and
(x(i,8) ≥ u(b,15)) and (x(i,8) ≤ u(b,16)) then
  x10(i,1) = chrW(1072) '/ A
end if
```

```
. . . . .
next i
```

where $x(i,1)$, $x(i,2)$, …, $x(i,8)$ – characteristic features of phonemes in the i-th segment, $u(b,1)$, $u(b,3)$, …, $u(b,15)$ – lower limit of the frequency range of phoneme standards, $u(b, 2)$, $u(b,4)$, …, $u(b,16)$ – upper limit of the frequency range of phoneme standards, $x10(i, 1)$ – phoneme a in the i-th segment, n – number of segments, b – phoneme sequence number.

Similarly, the remaining phonemes are allocated in segments and stored as a table in the array $x10(i,b)$.

The averaged, smoothed spectrum of segments of the function $W(1,b)$ is used as a phoneme reference. The measure of similarity (difference) is the Euclidean distance between the reference spectrum of phonemes and the spectra of the acoustic signal segments

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2}$$

The calculated distance values for each segment and each phoneme are stored as a table in a two-dimensional array for further processing.

The measure of the difference is a measure of the type of correlation between the reference spectrum of phonemes and the spectra of speech signal segments – the Pearson correlation coefficient.

In each segment, the Fourier spectrum of the function $W(1,b)$ is normalized and compared with the reference phoneme spectra, and the result is stored in a two-dimensional array.

The mutual correlation function of the wavelet spectrum of the acoustic signal and the standards of vowel sounds of speech is calculated. The cross-correlation function is defined by the expression

$$R_{xy}(\tau) = \frac{1}{T}\int_{0}^{T} x(t)\, y(t+\tau)dt$$

The maximum values of the correlation function are used to determine the vowel phonemes in the speech signal.

One of the main factors that negatively affect the recognition of phonemes is the variability of speech, which manifests itself in differences in the utterance of the same word or sentence. Most phoneme recognition errors are caused by random nonlinear deformations of the phoneme spectrum shape and temporary non-stationarity. Therefore, it is difficult to form phoneme standards for a large database of words (a dictionary). One of the ways to ensure invariance to the utterance feature is the WT of the speech signal. WT allows us to separate the low-frequency features of the signal from the high-frequency ones, and as a result, the range of spectral uncertainty is reduced. Experimental studies show that phoneme standards based on the WT speech signal have better recognition qualities.

The algorithm for forming a standard and recognizing phonemes using the correlation coefficient is better than the algorithm for finding the Euclidean distance. It should be noted that the second algorithm is less flexible than the first, because it uses 60 parameters for recognition and it is difficult to update the phoneme reference. The fourth algorithm is only useful for working with vowel phonemes. Therefore, the algorithm of word formation for the first algorithm of standard formation and phoneme recognition is considered below.

3.4 Highlighting the border between vowel and consonant phonemes in speech and recognizing isolated words

One of the main difficulties in recognition is the indefinite temporal organization of the speech signal. Obviously, the accuracy of word recognition significantly depends on the accuracy of determining the boundaries of phonemes. Determining the boundaries of phonemes means the operation of expedient division of speech into fragments, i.e. the segmentation of speech, which, in accordance with phonetic transcription, is a fundamental task of the voice control system. All further speech processing depends fundamentally on the accuracy of determining the boundaries of speech. The Parseval formula is used to calculate the energy of phoneme segments

$$\int_{-\infty}^{\infty} f^2(t)\, dt = \int_{-\infty}^{\infty} |F(\nu)|^2\, d\nu$$

The continuous wavelet transform formula is used to calculate the segment energy wavelet spectrum. To determine the boundaries between the vowel and consonant letters of a word, the energy of the segments of the functions $W(2, b)$, $W(20, b)$ and the studied word $S(t)$ is calculated. We obtain the Fourier spectrum of the segments of the functions $W\,(2, b)$, $W(20, b)$, and $S(t)$. The energy of the segments is calculated by the formula

$$E = \sum_{i=1}^{n} F(i)$$

Calculating the energy of the segments by the formula is almost the same as finding the variance of the wavelet coefficients by the standard formula

$$\sigma(a) = \sum_{i=1}^{n} (W(a, i) - \langle W(a, b) \rangle)^2$$

where

$$\langle W(a, b) \rangle \geq \sum_{i=1}^{n} \frac{W(a, i)}{n}$$

is the average value of the wavelet coefficients in the segment.

Using the energy of the segments when summing all the frequencies is equivalent to using the variance of the wavelet coefficients since the average value of the wavelet coefficients in the segment is close to zero. The positive and negative values of the wavelet coefficients are almost the same, and therefore the average value of the wavelet coefficients in the segment is close to zero (Fig. 3.4.1, 3.4.2). However, the energy can be calculated for different frequency ranges and more information can be obtained. As a result, the number of calculations is reduced.

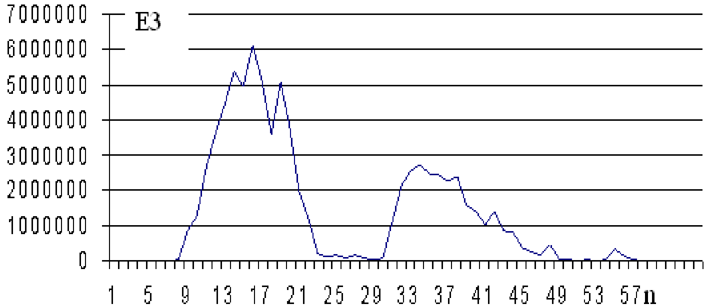


Figure 3.4.1. Energy of the $E3(n)$ segments of the word "осень" ["autumn"]

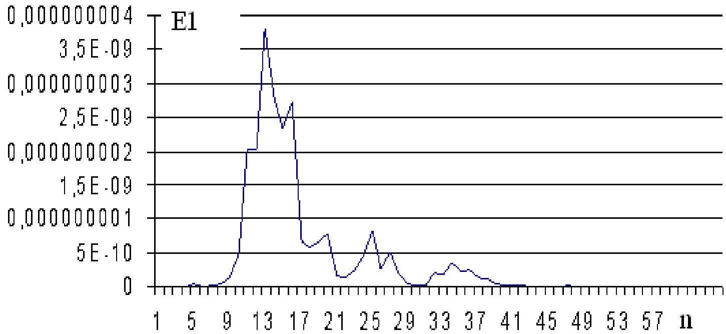


Figure 3.4.2. Energy of segments $E1(n)$ of the function $W(1,b)$ of the word "осень"

We denote the energy of the segments *WT* $W(2,b)$, $W(20,b)$ and the word $S(t)$ under study by functions $E1(n)$, *E2(n)* and $E3(n)$, respectively, where n varies from 1 to 256. Figure 3.4.1 shows the energy of the $E3(n)$ segments of the word "осень". Figure 3.4.2 shows the energy of the segments $E1(n)$ of the function $W(1,b)$ of the word "осень". When comparing the figures, we can immediately see the difference between the functions $E1(n)$ and $E3(n)$. The results of the analysis show that the energy of the segments of vowel letters in $W(1,b)$, $W(2,b)$ is allocated as maximum peaks, and the energy of consonant letters is always lower than the energy of vowels. The energy of the segments of hissing letters in $E1(n)$ is allocated as maximum peaks, in $E2(n)$ and $E3(n)$ as minima.

In order to determine the location of phonemes in a word, WT is calculated by the function $E1(n)$, $E2(n)$, and $E3(n)$ with a scale factor of $a = 4$.

Thus, the mathematical model of the speech signal when finding the boundaries between vowels and consonants of speech sounds is the wavelet spectrum of the energy of the segments of the wavelet spectrum of the speech signal. The coefficient a can vary from 3 to 8. We denote them by the functions $W1(4, b)$, $W2(4, b)$ and $W3(4, b)$, respectively, where b varies from 1 to 256. The block diagram of the algorithm for determining the boundary between vowel and consonant phonemes in a speech signal is shown in Figure 3.4.7.5.

Figure 3.4.3 shows the result WT of the function $E2(n)$ of the word "сигнал" ["signal"], where the positive values of the function $W2(4, b)$ correspond to vowels, and the negative values correspond to consonants.

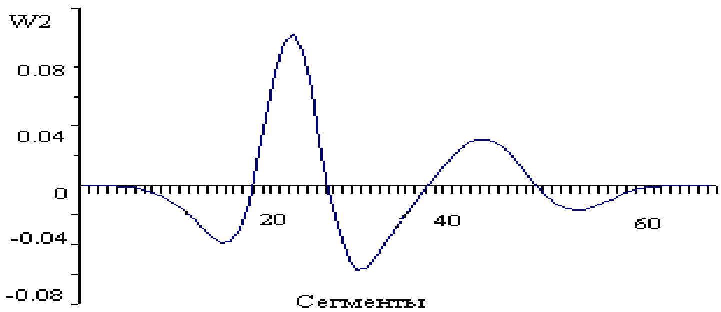


Figure 3.4.3. The wavelet spectrum $W2(4, b)$ of the function $E2$ (n) of the word "сигнал"

According to the results of the WT, it was found that vowel letters always have a positive value in $W1(4, b)$, $W2(4, b)$ and $W3(4, b)$. Sibilant consonants have a negative value in the function $W2(4, b)$ and $W3(4, b)$. Some sibilant letters have a positive value in $W1(4, b)$. Therefore, to find the location of vowel letters, the energies $E2(n)$, $E3(n)$ are normalized, their sum is found, and *WT* $W4(4, b)$ is performed. Thus, if a word contains one vowel letter, then one positive maximum is allocated, if two vowel letters, two positive maxima, etc. Each word has a certain structure. The boundary between vowels and consonants, or between vowels and sibilants, is determined with an accuracy of 2-3 segments [37, 39, 40, 41, 44, 45, 47, 48].

To form a word, the number of recognized phonemes a is counted in the interval where the vowel letters are highlighted. Similarly, for other vowel letters, the number of recognized letters is found separately.

The three vowel sounds for which these numbers are the largest are determined and written in descending order in the string array $x(1, i)$, $x(2, i)$, $x(3, i)$, in order to later use them for comparison with the letters of words from the dictionary. In the same way, three consonants of non-hissing or hissing sounds are counted and selected in the interval where the consonants of non-hissing or hissing sounds are highlighted.

Figure 3.4.4 shows a fragment of the section where the "u" phoneme is and stands out more than the others.

	е	и				
		и	о	у	ю	
а	е	и			ю	э
	е	и			ю	
	е	и				
		и		у	ю	
	е	и		у		э
	е	и	о			э
	е	и				
а		и			ю	
		и			ю	

Figure 3.4.4. Fragment of the vowel letters section

Phonemes are written to *Excel* electronic cells from the string array $x10(i, b)$, so that we can visually observe how the recognition process occurs at different stages and also to train the system when entering new words into the dictionary. The block diagram of the word formation algorithm is shown in Figure 3.4.7.5.

Depending on the number of positive maxima in the function $W4(4, b)$, different algorithms are selected for comparing the studied word with the words in the database. If one positive maximum is selected, an algorithm is used to find the boundary between vowel and consonant phonemes for three phonemes. If two positive maxima are distinguished, then an algorithm is used to find the boundary between vowel and consonant phonemes for five phonemes, etc.

The interval for counting consonants is chosen wider to take into account the error of determining the boundaries of these letters, words are made up of these letters. Letters such as "м", "н", "л" ["m", "n", "l"] have almost identical features, so a dictionary is used to identify words to check for the presence of composed words in the word database. Words consisting of three letters are compared with words in the dictionary using the following algorithm:

```
for i = 1 to n
if (x(1,1) = x10(i,1) or x(2,1) = x10(i,1) or x(3,1) = x10(i,1))
and (x(1,2) = x10(i,2) or x(2,2) = x10(i,2) or x(3,2) = x10(i,2))
and (x(1,3) = x10(i,3) or x(2,3) = x10(i,3) or x(3,3) = x10(i,3))
then
for j = 1 to 7
if (y10(i,j) = 0) then exit for
.cells(153, 55 + j).value = y10(i,j)
.cells(155, 55).value = i
next j,
else
.cells(154, 55).value = 0
end if
next i
```

where $x(1,1)$, $x(2,1)$, $x(3,1)$ – the first, second, and third vowel letters, respectively, $x(1,2)$, $x(2,2)$, $x(3,3)$ –the first, second, third consonant is not a sibilant or sibilant letter, respectively, $x(1,3)$, $x(2,3)$, $x(3,3)$ –the first, second, and third vowel letters, respectively, $x10(i,1)$, $x10(i,2)$, $x10(i,3)$ – the first, second, and third letters of the i-th word in the dictionary.

Since the pronunciation of words is very much contextually dependent on the spelling, the word is written in the dictionary as pronounced, and the output as correctly spelled. Moreover, in the database of words, we can store different versions of the pronunciation of one word.

For example, the word "яма" ["hole"] can be written in the dictionary in four variants: "яма", "амя", "яла", "ена" in the string array $x10(i, j)$. For all these combinations of letters, the letters "я", "м", "а" are stored in the string array $y10(i, j)$, i.e. for i – $i + 1$, $i + 2$, $i + 3$, because there are no Russian words "амя", "яла", "ена". Phonemes "а" and "я" have almost the same features while phonemes "н", "л", "м" differ little from each other. Encoding a single "яма" word with multiple variants increases the probability of recognizing that word. Other words in the dictionary are also stored in several variants. The recognized word can also be output in another language.

A similar algorithm is used to recognize words consisting of four, five, six or more letters. If a word has two vowel letters, i.e. the wavelet spectrum $W4(4, b)$ consists of two positive maxima, the six-letter algorithm is first used to identify the word. There are several options for the arrangement of letters. For example, two consonants side by side, two through a vowel. Then for five, four, and three letters. The database of individual words can be used for all people, because the structure of the word does not depend on the different pronunciation, timbre and emotional state of the person, and the boundaries between vowels and consonants are determined for all people in the same way.

In order to distinguish some words, additional features are used. The number of segments that fall on vowels and sibilant letters is usually greater than the number of segments that fall on the rest. The Fourier spectrum of the WT segments of hissing letters is very different from other letters. Figure 3.4.5 shows the Fourier spectrum of the WT segments of the word "щебень" ["rubble"].

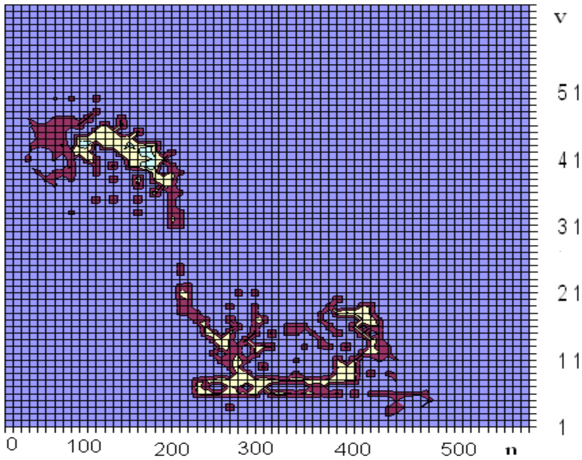
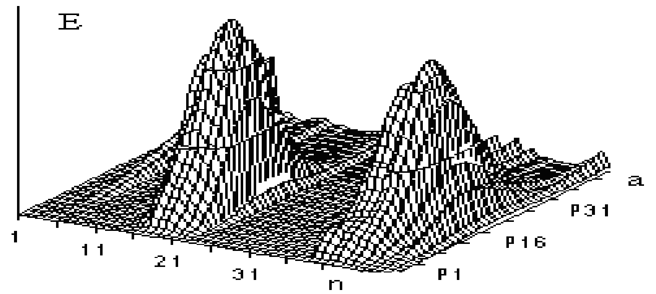


Figure 3.4.5. Fourier spectrum of the WT segments of the word "щебень"

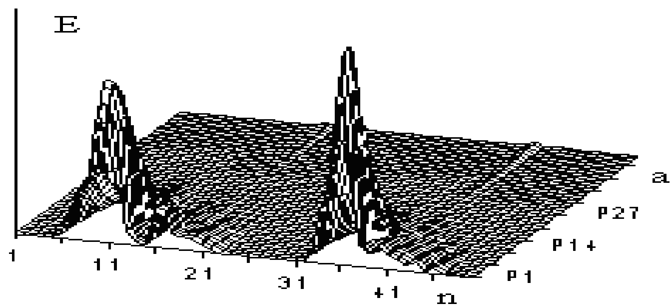
The figure shows that the phoneme "щ" has a different spectrum from the others. Hissing phonemes when spoken contain more high-frequency components in the spectrum than other phonemes.

3.4.1 Selection of phonemes by the energy of the segments of the wavelet coefficients $W(a, b)$ at different scales

A detailed picture of the location of phonemes in a word or sentence can be established by studying the dependence of the energy of the segments of the wavelet spectrum on the scale factor a. The *MHAT* wavelet is used for the study. The continuous wavelet transform formula is used to calculate the wavelet spectrum of a speech signal. The Fourier transform is used to calculate the Fourier spectrum of the wavelet spectrum segments. The Parseval formula is used to calculate the energy of phoneme segments. To study the dependence of the energy of the segments of the wavelet spectrum on the scale factor a, the energy of the segments of the functions $W(a, b)$ is calculated. Figure 3.4.1.1 shows graphs of the dependence of the energy of the segments E on the scale factor a of the wavelet transform $W(a, b)$ of the word "часы" ["clock"]. In Figure 3.4.1.1a the scale factor a changes from 1 to 50 in increments of 1. When calculating the energy, the first half of the Fourier coefficients is summed up [79]. In Figure 3.4.1.1b the scale factor a changes from 0.4 to 2.9 in increments of 0.05. When calculating the energy, the second half of the Fourier coefficients is summed up.



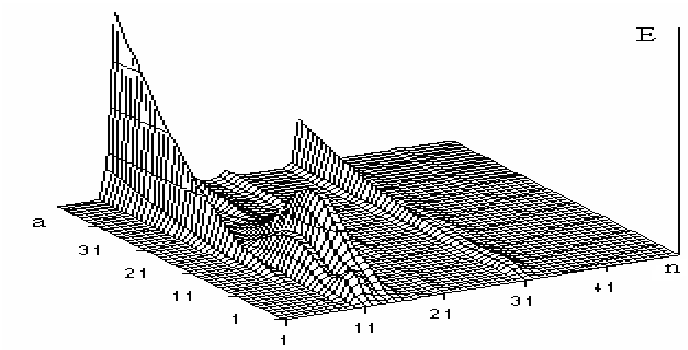
a



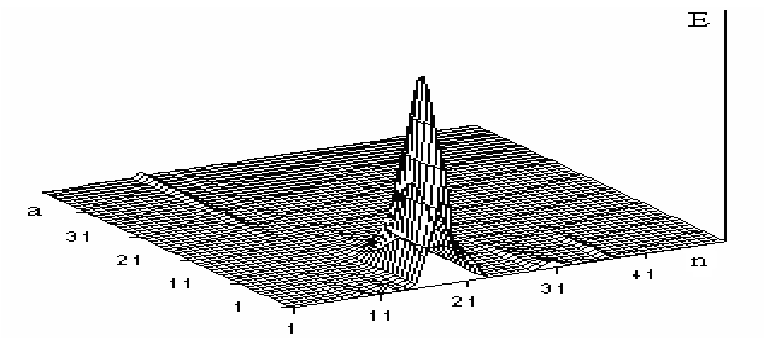
b

Figure 3.4.1.1 Energy of the segments WT $W(a, b)$ of the word "часы"

A similar picture is obtained when calculating the variance of the wavelet coefficients. However, when calculating the variance, all the Fourier coefficients are taken into account, and it is impossible to present graphs for different spectral ranges. The first graph (Fig. 3.4.1.1, a) shows in which segments the vowel sounds "а", "ы" are distinguished. The phonemes "ч" and "с" are not distinguished. In the second graph (Fig. 3.4.1.1, b), the phonemes "ч" and "с" (segments 4-13 and 32-38) are clearly visible and have more energy than the vowel phonemes. Figure 3.4.1.2 shows graphs of the dependence of the energy of the segments E on the scale factor a WT $W(a, b)$ of the word "пуск" ["start"].



a



b

Figure 3.4.1.2. Energy of the segments WT $W(a, b)$ of the word "пуск"

The conversion parameters are the same as for the word "часы". The phoneme "п" is distinguished in segments 7-9. Phoneme "с" is allocated (segments 15-21) in the same way as phoneme "с" is allocated in the word "часы" for small values of the scale factor a. The graph clearly shows the pause before the "к" phoneme. The wavelet analysis of the speech signal shows that the vowel phonemes and phonemes "н", "м", "л" have maximum energies at the average values of a. The energy of the phonemes "н", "м", "л" is much less than the energy of the vowel sounds of speech, but much higher than the energy of noise. The phonemes "к", "т", "п", "д" are distinguished at large values of a.

There is a pause before the phonemes "к" and "т". This pattern is observed with repeated repetition and does not depend on random factors. Hissing and whistling phonemes with small values of the scale factor a have an energy $W(a,b)$, comparable to the energy of vowel phonemes. At average values of a, they have energy at the noise level.

It should be noted that phonemes have a different dependence of the energy $W(a,b)$ on the scale factor a when using a different frequency range. This allows us to distinguish two or three adjacent vowel or consonant phonemes. For example, all digits from 0 to 9, arithmetic operations $(+,-,/,\times)$, and commands "пуск", "стоп" ["stop"], "выключить" ["turn off"], and others can be recognized by examining the dependence of the energy of segments E on the scale factor a WT $W(a,b)$, without using other parameters since for each digit and command when pronounced, the dependence of the energy of segments $W(a,b)$ on the scale factor a is very different. Figures 3.4.1.3 and 3.4.1.4 show graphs of the dependence of the energy of the segments E on the scale factor a WT $W(a,b)$ of the word "стоп".

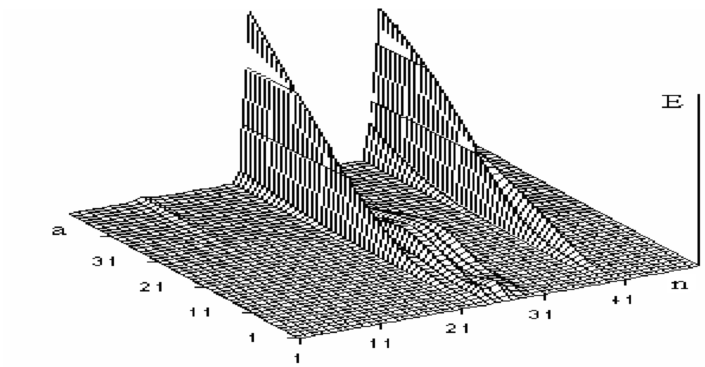


Figure 3.4.1.3. The energy of the segments WT $W(a,b)$ of the word "стоп" for large and medium values of the scale factor a

In Figure 3.4.1.3, the scale factor a changes from 1 to 50 in increments of 1. In Figure 3.4.1.4, the scale factor a changes from 0.4 to 2.9 in increments of 0.05.

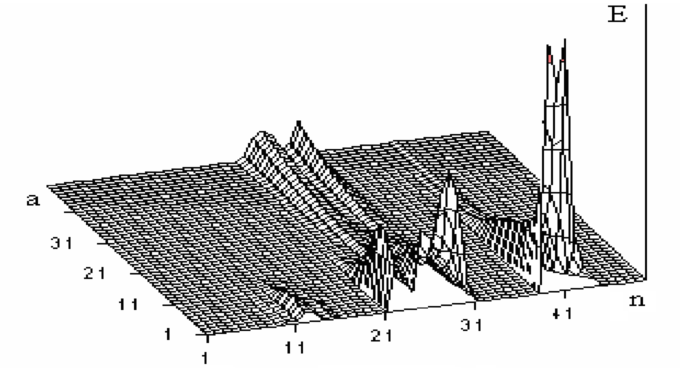


Figure 3.4.1.4. The energy of the segments WT $W(a, b)$ of the word "стоп" for small values of the scale factor a

In Fig. 3.4.1.2 – 3.4.1.4, the location of the phonemes "п", "т" and "к" for large values of the scale factor a is different for the words "пуск" and "стоп", as well as for vowel phonemes at medium values of the scale factor a. For example, the words "выключить" and "отключить" have the same functions $W4(4, b)$. The first word begins with a consonant, and the second with a vowel phoneme. However, the ratio of the energy of the segments WT $W(9, b)$ to the energy of the segments WT $W(2, b)$ of the first word is greater than for the second at the beginning of the word, since this pattern is observed for all consonant phonemes. For the word "выключить", the peak appears at the beginning of the word, and for the word "отключить", it is absent.

Multiscale speech signal filtering allows us to represent words as a matrix of three rows and n columns. The first line corresponds to a large, the second to an average, and the third to a small value of the scale factor a. The scale factor a in the range from 0.2 to 1 corresponds to a small scale, from 1 to 40 – to an average scale, from 41 to 50 – to a large scale. The number of columns depends on the length of the word and the order of the vowels or hissing, muffled, explosive sounds of speech in the word. For example, the word "лента" ["tape"] can be represented by a matrix (Figure 3.4.1.5).

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 3.4.1.5. The matrix of the word "лента"

The units in the first row of the matrix A correspond to the phoneme "т", which is distinguished at large scales when pronouncing the word "лента". The zero elements of the fourth column of the same matrix correspond to the pause between the syllables "лен" and "та", which appears when pronounced. Figure 3.4.1.6 shows the dependence of the energy of the WT segments $W(a,b)$ for the scale coefficients $a = 23$ and $a = 47$.

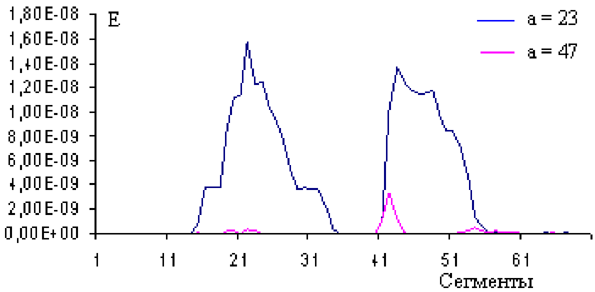


Figure 3.4.1.6. The energy of the segments of the word "лента" for the scale coefficients $a = 23$ and $a = 47$

The graph clearly shows the phoneme "н" with a large scale factor $a = 47$, and the phonemes "а", "е" have insignificant energy. The graph also shows a pause between the syllables "лен" and "та", the segments of which have an energy that is insignificant for the scale coefficients $a = 23$ and $a = 47$. The energy of the segments WT $W(a,b)$ changes during the transition from the phoneme "л" to the phoneme "е" and from the phoneme "е" to the phoneme "н".

The wavelet spectrum $W4(4,b)$ clearly distinguishes the boundaries between the phonemes "л", "е", "н" and is the basis for constructing the matrix of the word "лента". Depending on the number of maxima of this function, the rectangular surface (segments, scale factor) is divided into several areas. Figure 3.4.1.7 shows the projection of the energy of the segments WT $W(a,b)$ of the word "умножить" ["multiply"] by this surface.

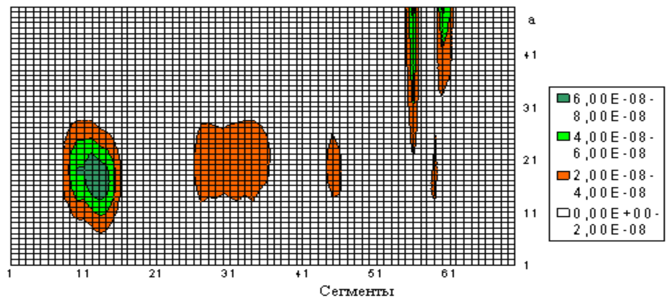


Figure 3.4.1.7. Projection of the energy of the segments WT $W(a,b)$ words "умножить" by a rectangular surface (segments, scale factor)

For the word "умножить", the function $W4(4,b)$ has 4 maxima corresponding to the phonemes "у", "о" and "и".The surface is divided into 5 regions. The first region is from the first segment to the first maximum of the function $W4(4,b)$, the second is from the first maximum of the function $W4(4,b)$ to the second maximum, and so on. As we know, a word consists of several syllables, so therefore, a word matrix can be made up of several smaller matrices by joining them together. Thus, word recognition is facilitated by sequential, synchronous string shifting and a matrix comparison [38, 39, 44].

Multi-scale analysis based on WT allows us to combine words into different groups. As a result, the recognition time is reduced and the recognition accuracy is increased since the database of words can be divided into subgroups and presented as a search tree.

3.4.2 Dependence of the energy of the segments WT $W(a,b)$ on the value of the scale factor a

The dependence of the energy of the segments WT $W(a,b)$ on the value of the scale factor a for "п" and "а" phonemes is shown in Figure 3.17. It can be seen that the phoneme "п" has many times more segment energy at a large scale factor than the phoneme "a", and less energy at an average value of a.

The dependence of the energy of the segments WT $W(a,b)$ on the value of the scale factor a for the phonemes "п" and "a" is shown in Figure 3.4.2.1 It can be seen that the phoneme "п" has many times more segment energy at a large scale factor than the phoneme "a", and less energy at an average value of a.

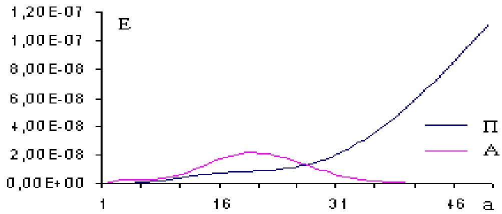


Figure 3.4.2.1. Dependence of the energy of the WT $W(a,b)$ segments on the value of the scale factor a of the phonemes "п" and "a"

For the phonemes "т" and "к", the dependence of the energy of the segments WT $W(a,b)$ on the value of the scale factor a is the same as for the phoneme "п". In the word "лента", at large scale coefficients, the signal level is comparable to the noise level for all phonemes except for the phoneme "т", so the elements of the first row of the matrix are zero for the phonemes "л", "е", "н", "а". The dependence of the energy of the segments WT $W(a,b)$ on the value of the scale factor a for the phonemes "н", "м", "л" differs only in that these phonemes always have less energy than the vowel phonemes. Figure 3.4.2.2 shows this difference.

Whistling phonemes, which are distinguished at a small scale factor on par with vowel phonemes, have the same dependence of the energy of the segments WT $W(a,b)$ on the value of the scale factor a as the phoneme "ш".

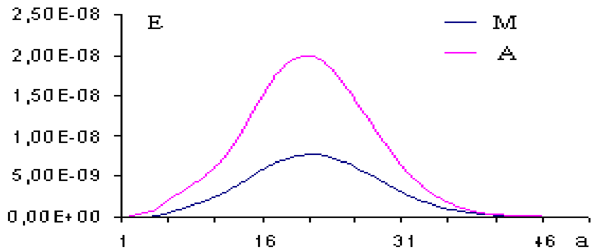


Figure 3.4.2.2. Dependence of the energy of the WT segments $W(a, b)$ on the value of the scale factor a of the phonemes "м" and "a"

Figure 3.4.2.3 shows the dependence of the energy of the segments $W(a, b)$ on the scale factor a of the phonemes "a" and "ш".

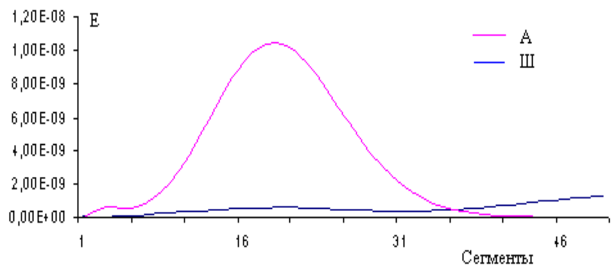


Figure 3.4.2.3. Dependence of the energy of the WT segments $W(a, b)$ on the value of the scale factor a of the phonemes "a" and "ш"

These examples show that the multiscale representation allows us to visualize the dynamics of changes in the speech signal along the "scale axis". These changes in the "scale variable" provide important information about the speech signal.

3.4.3 Speech fusion recognition

Unlike printed text or artificial signals, natural speech does not allow simple and unambiguous division into elements (phonemes, words, phrases), since these elements do not have explicit physical boundaries. They are isolated in the mind of the listener – a native speaker of a given language – as a result of a complex multi-level process of speech recognition and understanding. The *MHAT* wavelet is used for the study.

The boundaries can only be determined during the recognition process by selecting the optimal sequence of words that best matches the input stream of speech according to acoustic, linguistic, and pragmatic criteria.

Figure 3.4.3.1 shows the energy of the segments WT $W(a, b)$ of the sentence "лента остановилась" ["tape spopped"] for the scale coefficients $a = 23$ and $a = 47$. The energy of the segments of the word "лента" in the sentence exactly repeats the dependence of the energy of the segments of the individual word "лента". There is no pause between the words "лента" and "остановилась". The phoneme "с" has large energy at a large scale factor a.

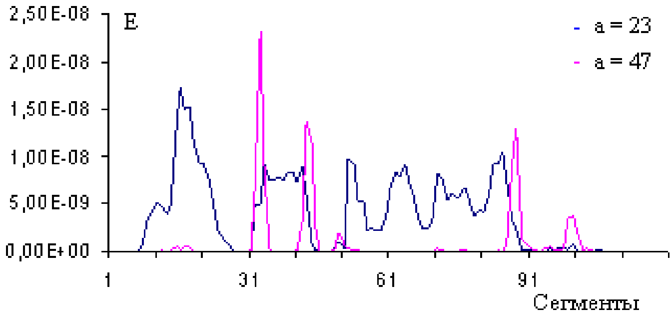
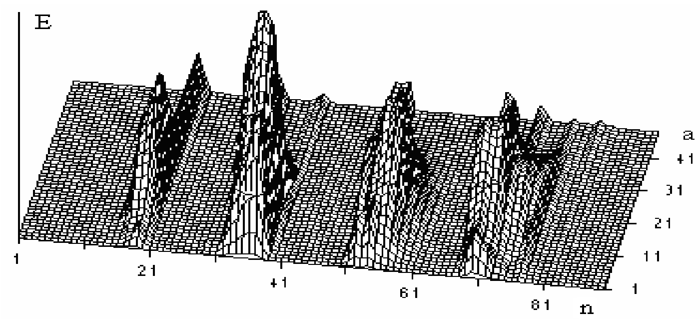


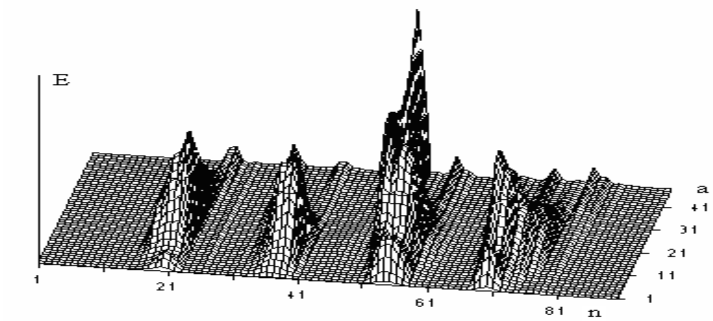
Figure 3.4.3.1. The energy of the segments of the sentence "лента остановилась" for the scale coefficients $a = 23$ and $a = 47$

Figure 3.4.3.2 shows a graph of the dependence of the energy of the segments E on the scale coefficient a of the wavelet transformation $W(a, b)$ of the phrase "открыть бункер" ["open the bunker"].

The scale factor a varies from 1 to 50 in increments of 1. When calculating the energy, the first half of the Fourier coefficients is summed up.



a



b

Figure 3.4.3.2. The energy of the segments WT $W(a, b)$ of the phrase "открыть бункер"

When the same sentence is repeated many times, the energy dependence of the segments WT $W(a, b)$ remains the same. The positions of the vowels and consonants remain unchanged relative to each other, only the durations between the peaks and their heights change. This arrangement of peaks for different scale coefficients for the same sentence does not depend on who pronounces the given sentence [38, 44].

3.4.4 Speech fusion recognition based on image processing techniques

The energy of the WT segments $W(a, b)$ is a two-dimensional object, so we can use two-dimensional WT methods for it. In [57, 71], strategies for searching an image database are considered. In them, the query is expressed either as a low-resolution image obtained using a scanner or video camera, or as a rough sketch of the desired image drawn by the user himself. This approach to the formation of an image query has received various names, including "query by content" [71], "query by sample" [71], "search by similarity method" [68] and "search by sketch" [69, 70]. Using the wavelet decomposition of the query image and the image from the database can quickly and efficiently satisfy the content query. The algorithm for generating an image query with variable resolution, which uses the most significant wavelet coefficients, significantly increases the speed and reliability of recognition of two-dimensional objects. Figure 3.4.4.1–3.4.4.4 shows the results of a two-dimensional WT of the energy of the segments WT $W(a,b)$ of the phrases "мелкий гравий" ["fine gravel"] and "крупный песок" ["coarse sand"].

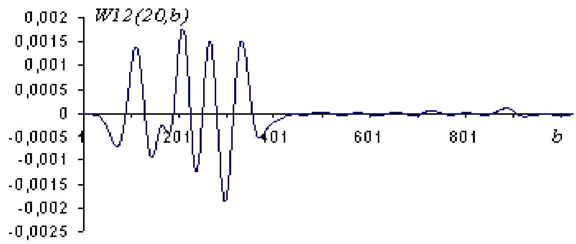


Figure 3.4.4.1. Wavelet transformation of the energy of the segments WT $W(a,b)$ of the phrase "мелкий гравий" by columns

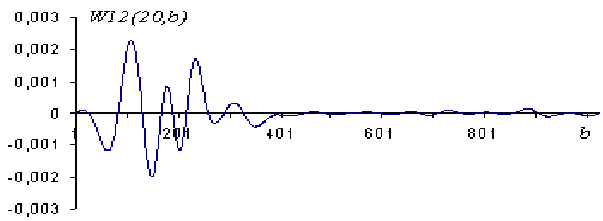


Figure 3.4.4.2. Wavelet transformation of the energy of the segments WT $W(a,b)$ of the phrase "крупный песок" by columns

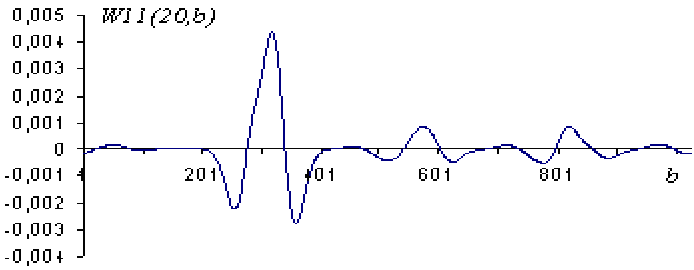


Figure 3.4.4.3. Wavelet transformation of the energy of the segments WT $W(a, b)$ of the phrase "мелкий гравий" along the lines

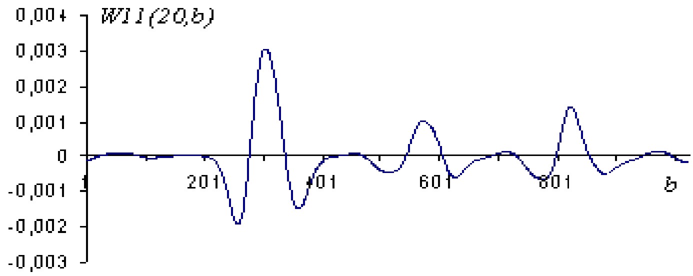


Figure 3.4.4.4. Wavelet transformation of the energy of the segments WT $W(a, b)$ of the phrase "крупный песок" along the lines

The coefficients of a two-dimensional WT have different values depending on the direction of decomposition, i.e. on the one-dimensional expansion over the rows or columns of a two-dimensional object. In this regard, we can use both the anisotropy of the two-dimensional WT and the dependence of the energy of the WT segments $W(a, b)$ on the scale factor a for the recognition of merged speech. The energies of the segments WT $W(a, b)$ are pre-calculated for the scale coefficients a, equal to 1, 21, 41, 61, to reduce the conversion time. Next, the wavelet coefficients are calculated, expanded by columns and rows for the scale factor $a = 20$, and are indicated in Figures 3.4.4.1–3.4.4.4 $W12(20, b)$ and $W11(20, b)$. The total number of segments is 1024.

We see that even though the word combinations have the same number of letters, the low-resolution wavelet transforms for these sentences are different. For example, the ratio between the first and second maxima in Figures 3.4.4.3 and 3.4.4.4 always remains the same, because the word "песок" has more energy for large scale factors a than the word "гравий". If we normalize the maxima by one and calculate the correlation function for these WT, we can easily recognize these phrases.

3.4.5 Algorithm for numerical calculation of fractal dimension

WT is well suited for analyzing fractal sets that have a hierarchical nature. One of the areas of application of wavelet analysis is the study of the properties of fractal objects of various nature and, in particular, the determination of the fractal dimension. The skeleton of the wavelet transform shows the presence of hidden self-similarity in a continuous display or a discrete data set in the form of a developed tree-like structure with forks that depend on the scale according to the power law.

The term "fractal" (Latin *fractus* – broken, fractional) was introduced into use by the American mathematician B. Mandelbrot. A fractal is a structure consisting of parts that are in some sense similar to the whole. These structures are characterized by a parameter called fractal dimension. Central to the definition of fractal dimension is the concept of the distance between points in the space ξ. As a test function for measuring the length of curves, surface area, or volume, a test function of the form is selected

$$h(\delta) = \gamma(d)\delta^d$$

where δ is the distance between points in space, $\gamma(d) = 1$, $\gamma(d) = \frac{\pi}{4}$, $\gamma(d) = \frac{\pi}{6}$ for a segment, square, cube, respectively. The measure of the set of points ξ in space is

$$M_d = \sum_{n=1}^{N(\delta)} h(\delta)$$

In general, for $\delta \to 0$, the measure M_d is zero or infinite, depending on the choice of the d – dimension of the measure.

The Hausdorff-Bezikovich dimension D of a set ξ is the critical dimension at which the measure M_d changes its value from zero to infinity

$$M_d = \sum_{n=1}^{N(\delta)} h(\delta) = \gamma(d) N(\delta) \delta^d \to 0 \text{ for } d > D \text{ with } \delta \to 0$$
$$M_d = \sum_{n=1}^{N(\delta)} h(\delta) = \gamma(d) N(\delta) \delta^d \to \infty \text{ for } d < D \text{ with } \delta \to 0$$

When $d = D$, the M_d measure changes its value abruptly. The Hausdorff-Bezikovich dimension $D = 1$ is for a line, for planes $D = 2$ and $D = 3$ – for balls and other bodies of finite volume. For fractal sets, the Hausdorff-Bezikovich dimension D is not an integer and is called the fractal dimension, which is strictly greater than the topological dimension, which is always equal to an integer. For example, for a line, the topological dimension is one. Thus, a fractal is a set whose Hausdorff-Bezikovich dimension is strictly greater than the topological dimension [2, 12, 60]. Many fractals are self-similar, i.e. a part of a set is similar to an entire set. This property is called scale invariance, or scaling. Scaling is the transformation of parallel transfer and scaling. For a self-similar set with a similarity coefficient r, the similarity dimension D_s is calculated by the formula

$$D_s = \frac{\ln N}{\ln r(N)}$$

where N is an integer for each self-similar fractal, N has its own value for such fractals whose similarity dimension coincides with the Hausdorff-Bezikovich dimension, $D_s = D$.

A fractal measure can be represented by interconnected fractal subsets varying in power law with different exponents. Such systems are called multifractal. The Hungarian mathematician A. Renyi proposed a family of dimensions that generalizes the Hausdorff-Bezikovich dimension. By definition, q-the Renyi dimension is defined by the formula

$$D_q = \lim_{\delta \to 0} \frac{1}{q-1} \frac{\log \sum_{i=1}^{N} p_i^q}{\log \delta}$$

where p_i is the probability of hitting the i-th component of the fractal.

For $q = 0$, $D_0 = D$, the Renyi dimension coincides with the Hausdorff-Bezikovich dimension. The dimension of the self-similar fractal coincides with the Hausdorff-Bezikovich dimension [12].

The fractal dimension is closely related to the Helder exponent α, also called the singularity exponent. The features of the M_d measure are characterized by the Helder exponent, i.e. the measure has singularities with the Helder exponent on fractal sets with a fractal dimension depending on α. Multifractal signals can be decomposed into "sub-signals", each of which is characterized by its own local dimension, given by a certain weight function [60].

Examples of classical fractal sets are: "Sierpinski carpet", "Sierpinski napkin", Koch curve, Cantor sets, Riemann, Weierstrass, Hankel, Bezikovich functions, etc. It is known that all the wavelets of this family are similar to their basic wavelet and are obtained from it by means of compressions and shifts. Since wavelet analysis involves the study of the behavior of signals at different scales, it is most suitable for the study of fractal behavior. By examining the sum of the higher moments of the wavelet coefficients at different scales, it is possible to determine whether a given signal is mono-or multifractal [60]. The wavelet analysis allows us to determine the fractal dimension of the set of points at which the function is singular. It is shown in [59] that the dimension of a monofractal can be calculated by the formula

$$D = \frac{\ln N(a)}{\ln a}$$

where $N(a)$ is the number of local maxima of the wavelet coefficients when the scale factor a tends to zero.

Consider a triad Cantor set. The construction of a Cantor set begins with a segment of unit length. Then the segment of unit length is divided into three parts, the middle part is discarded and two segments remain. Then each of the remaining segments is again divided into three parts and the middle parts are discarded, etc. After an infinite number of generations, the remaining infinite set is scattered over a single segment. This set is called Cantor dust. Figure 3.4.5.1 shows the 9th generation Cantor function. The experimental determination of the fractal dimension may introduce an error because the scale factor a is closer to zero or further from zero, depending on the signal sample. For example, with a sample of 8192 samples, the scale factor $a = 0.5$ is further from zero than $a = 1$ with a sample of 32768 samples. Also, the number of local maxima for the same interval depends on the number of samples.

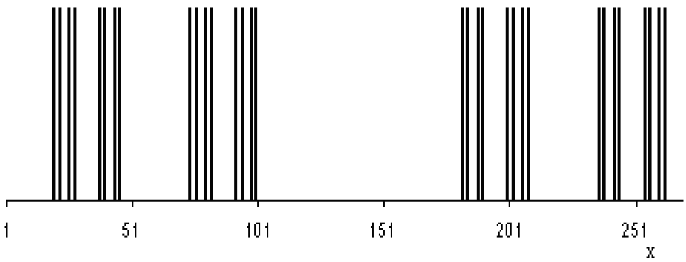


Figure 3.4.5.1. Cantor dust

To determine the fractal dimension, we have developed the following algorithm, which allows us to eliminate these shortcomings. The algorithm for calculating the fractal dimension includes the following steps:

1. The wavelet coefficients with scale factors from $a = 1$ to $a = 2000$ are calculated on a logarithmic scale.
2. The number of local maxima N for different scale coefficients a is calculated.
3. A graph of the dependence of the logarithm N on the logarithm a is plotted on a twice-logarithmic scale.
4. The least squares method calculates the slope of the curve, which corresponds to the fractal dimension.

Figure 3.4.5.2 shows the skeleton of the wavelet coefficients on the logarithmic scale for the 5th generation prefractal, i.e. the local maxima of the wavelet coefficients.

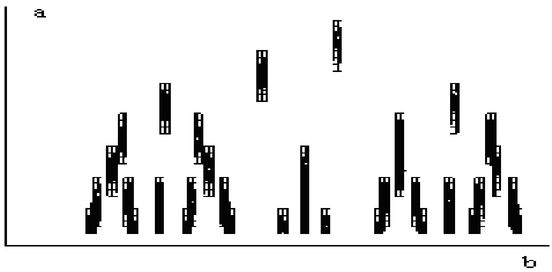


Figure 3.4.5.2. Skeleton of Cantor dust

It can be seen how the skeleton reveals not only the hierarchical structure of the analyzing set, but also the way of constructing the fractal measure on which it is formed. For the constructed 9th generation prefractal, the calculated dimension is $D = 0.60$. The theoretical fractal dimension $D = 0.63$ for a triad Cantor set. The higher the generation order of the Cantor series is used, the more precisely its dimension is determined [12, 31, 59].

3.4.6 Speech fusion recognition using fractal dimension

To recognize merged speech, we can use a picture of the local maxima of phonemes. The picture of the local maxima of phonemes in a sentence coincides with a similar picture in a word. Figures 3.4.6.1 and 3.4.6.2 show the skeletons of the phonemes "a" and "и". The scale factor a in these figures varies from 1 to 27.

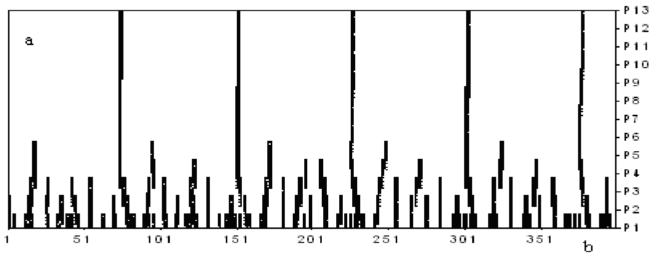


Figure 3.4.6.1. Skeleton of phoneme "a"

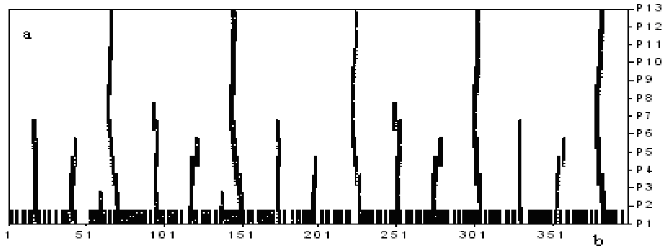


Figure 3.4.6.2. The skeleton of the phoneme "и"

As the scale factor a increases, the number of local maxima decreases, similar to fractal behavior. If we understand something different from an abstract triad set by a Cantor set, the speech signal can be represented as a set whose rod heights are different, in contrast to the triad set. When calculating the dimension according to the developed algorithm, fractional numbers are obtained for phonemes. Since the topological dimension is zero for a Cantor set, and the calculated dimensions are greater than zero and not integers, the speech signal can be represented as a fractal set. The fractal dimensions for different phonemes differ from each other, and therefore they can be used as information features of phonemes in speech recognition [31]. Figures 3.4.6.1 and 3.4.6.2 clearly show the difference between the phonemes from each other. For vowel phonemes, when the scale factor a is increased, the line corresponding to the main tone is clearly visible. To find the pitch frequency, it is enough to determine the time between these lines. Skeletons of phonemes show that regardless of whether the words are pronounced separately or together in a sentence, they have the same picture.

The representation of a sentence in the form of a wide matrix allows us to select words from a sentence by sequentially comparing a section of the sentence matrix with the word matrix. In addition, using the matrix allows us to select the entire sentence from the speech stream, if such a sentence is in this stream.

To recognize a sentence, we can also use the grammatical rules of a particular language. By comparing the roots and endings of the words of a sentence with the roots and endings of individual words, we can recognize the sentence.

3.4.7 Automatic speech command recognition device

The block diagram of the speech command recognition device is shown in Figure 3.4.7.1. The principle of operation of the device is as follows.

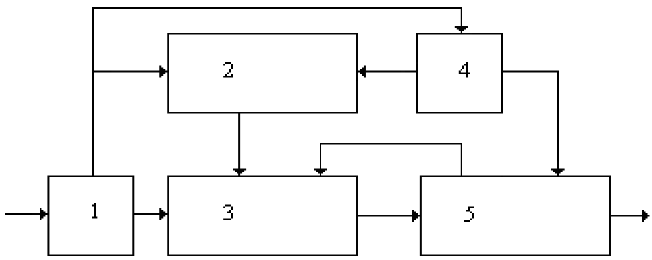


Figure 3.4.7.1. Block diagram of the speech command recognition device:
1 – audio signal preprocessing unit
2 – block for highlighting information features of phonemes
3 – phoneme identification block
4 – block for determining the boundaries between vowel and consonant phonemes
5 – word formation block

The analyzed audio signal $S(t)$ is sent to the audio signal preprocessing unit (block 1). In block 1, the audio signal is digitized using an ADC, cleared of extraneous noise, and stored in a RAM. From the output of block 1, the digitized audio signal, with the parametric representation of the audio signal, simultaneously enters the block for selecting information features of phonemes (block 2) and the block for determining the boundaries between vowels and consonant phonemes (block 4).

In the parametric representation mode of the audio signal, the selected phoneme information features from block 2 are sent to block 3 and stored so that they can be used for phoneme recognition. In speech recognition mode, the digitized audio signal simultaneously enters the phoneme identification block (block 3) and the vowel-consonant phoneme boundary detection block (block 4). When speech recognition is performed from the outputs of blocks 3 and 4, the processed audio signal is sent to the word formation block. The block diagram of the algorithm for forming the phoneme database is shown in Figure 3.4.7.2.

The following algorithm is used to form phoneme standards. The wavelet spectrum $W(a, b)$ is calculated. The resulting wavelet coefficients (functions) $W(a, b)$ are divided into segments of fixed duration. In each segment, the Fourier coefficients $a(i)$, $b(i)$ of the functions $W(1, b)$ and $W(2, b)$ are calculated using the FFT. The Fourier spectrum of the functions $W(1, b)$ and $W(2, b)$ of each letter of the Russian alphabet is calculated. The number of local maxima $W(a, b)$ is calculated as the background for the scale factor a, greater than 2. For each letter of the Russian alphabet, a database is created with a set of characteristic frequencies (range) of function segments $W(a, b)$.

The lower and upper limits of the range of characteristic frequencies obtained by the repeated pronunciation of Russian words are used as phoneme standards for speech recognition. It is possible to update (expand the frequency range) the database with phoneme standards so that speech sounds stand out when new words are added to the dictionary (a database of individual words), i.e. a system-training algorithm has been developed. A block diagram of the phoneme identification algorithm is shown in Figure 3.4.7.3. To identify phonemes in the phoneme identification block, basically the same actions are performed as in the phoneme database formation block, only in the last point, the phonemes of the studied speech signal are compared with the phoneme standards of the database. The selected phonemes are stored in RAM.

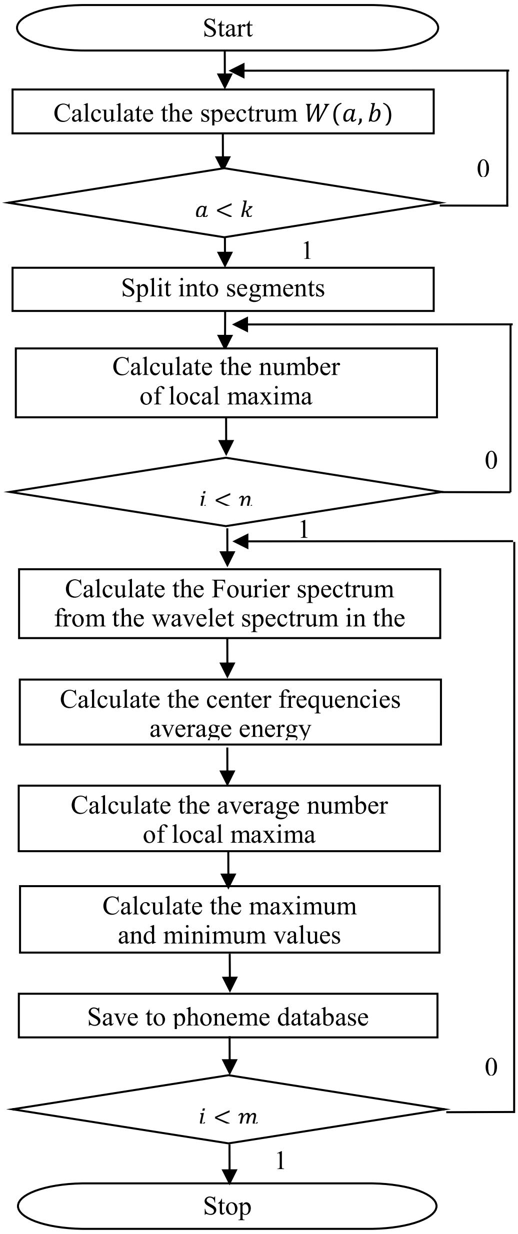


Figure 3.4.7.2. Block diagram of the phoneme database generation algorithm

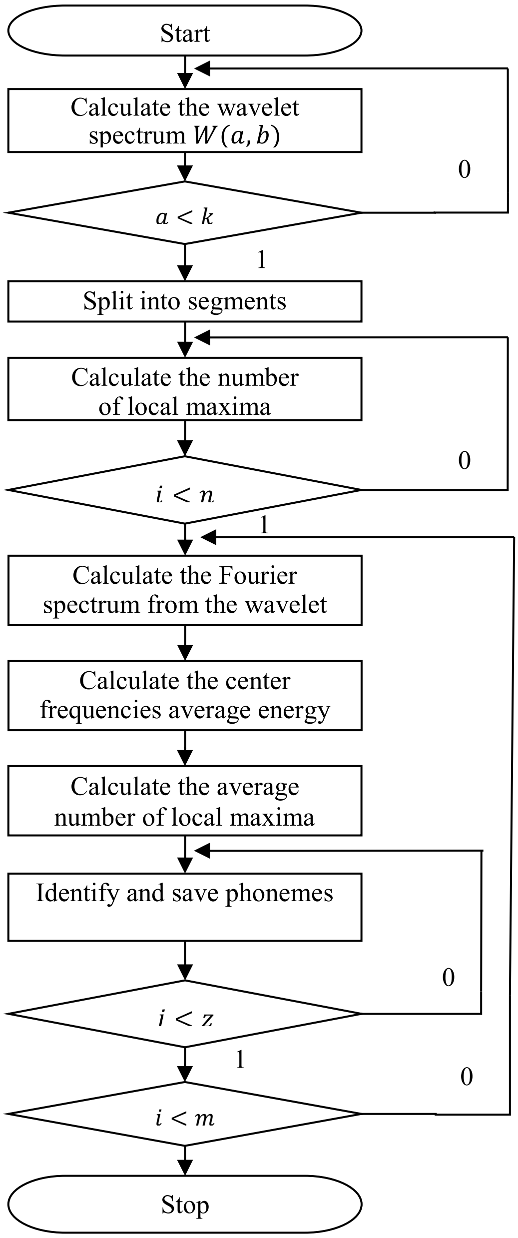


Figure 3.4.7.3. Block diagram of the phoneme identification algorithm

A block diagram of the algorithm for determining the boundary between vowel and consonant phonemes is shown in Figure 3.4.7.4.

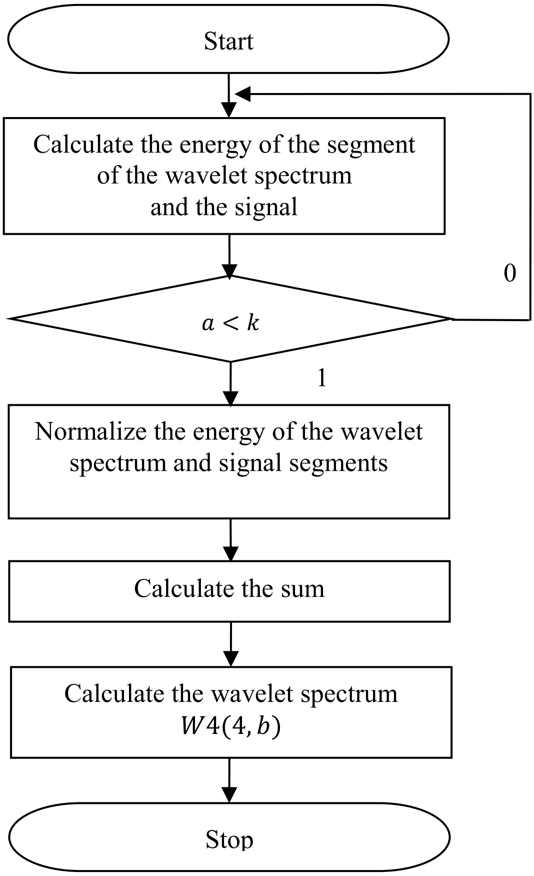


Figure 3.4.7.4. Block diagram of the algorithm for determining the boundary between vowel and consonant phonemes

The algorithm for determining the boundary between vowel and consonant phonemes includes the following steps. The energy of the segments of the wavelet spectrum is calculated for the scale factor a, equal to two, and the energy of the segments of the audio signal, the sum of the normalized energies is determined. The result of summation is subjected to WT. The wavelet spectrum $W4(4, b)$ determines the boundaries between vowel and consonant phonemes in the audio signal, which are used in block

2 for visual observation when forming the phoneme database and block 5 for word formation. The block diagram of the word formation algorithm is shown in Figure 3.4.7.5.

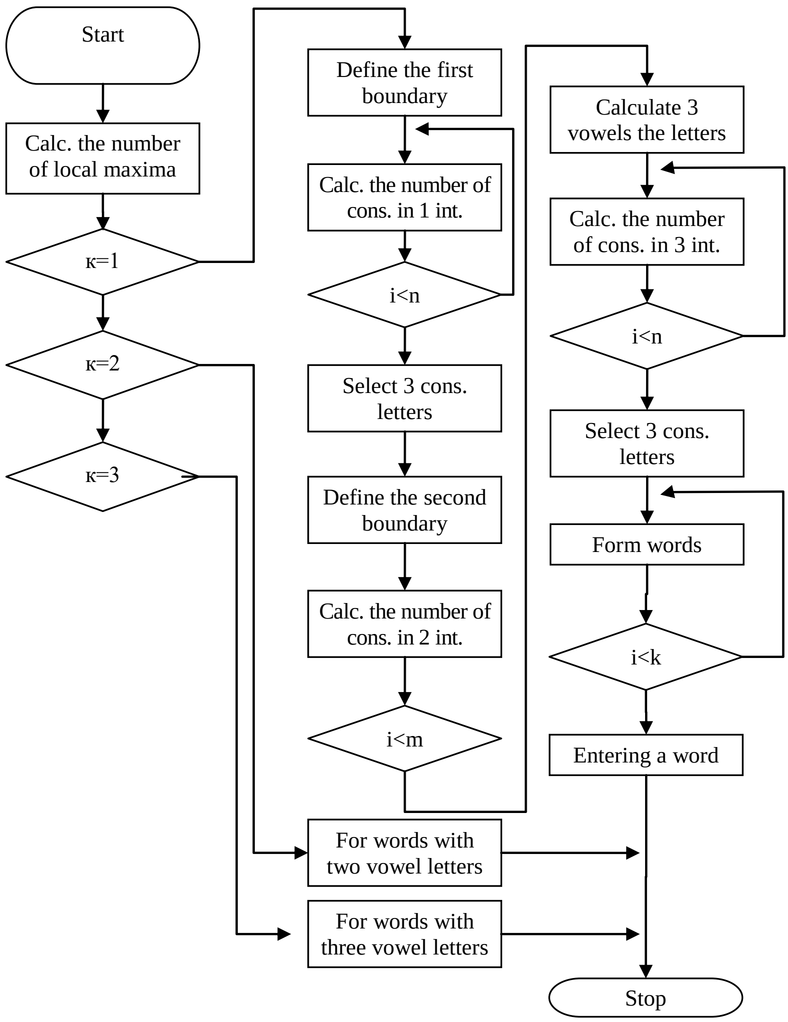


Figure 3.4.7.5. Block diagram of the word formation algorithm

Depending on the number of positive maxima in the function $W4(4, b)$, different algorithms are selected for comparing the studied word with the words in the word database. If one positive maximum is selected, an algorithm is used to find the boundary between vowel and consonant phonemes for three phonemes. If two positive maxima are distinguished, then an algorithm is used to find the boundary between vowel and consonant phonemes for five phonemes, etc.

To form a word, the number of recognized letters is counted in the interval where the vowel letters are highlighted. The three vowel letters for which these numbers are the largest are determined and written in descending order in a string array in order to later use them for comparison with the letters of words from the dictionary. In the same way, three sibilants or sibilant letters are counted and selected in the interval where the consonant non-sibilant or sibilant letters are highlighted. To identify words, a dictionary is used to check for the presence of composed words in the word database.

3.5 Conclusions

1. Phoneme standards based on the Fourier spectra of the segments of the wavelet coefficients of the speech signal and on the Fourier spectra of the segments of the speech signal are studied. The former were less variable than the latter. A phoneme database and algorithms for identifying speech signal phonemes have been developed.

2. The WT of the speech signal allows us to determine the temporal organization of speech, i.e. to determine the position of the vowels and consonants of the phonemes of a word or sentence. An algorithm for determining the boundaries between vowels and consonants of speech sounds has been developed. An algorithm for forming words based on recognized phonemes has been developed.

3. Multiscale processing of the speech signal produces muffled explosive sounds at a large scale factor, and muffled slits and affricates at a small scale factor. Vowel phonemes have the highest values of the wavelet coefficients at the average values of the scale factor and a longer duration compared to other speech sounds.

4. Multiscale speech signal processing shows that the words in a sentence have the same structure as when pronounced separately. A two-dimensional WT-based speech fusion recognition algorithm has been developed.

5. A software package has been developed using *Visual C++* and *Visual Basic for Applications*, which implements algorithms for multiscale speech signal processing.

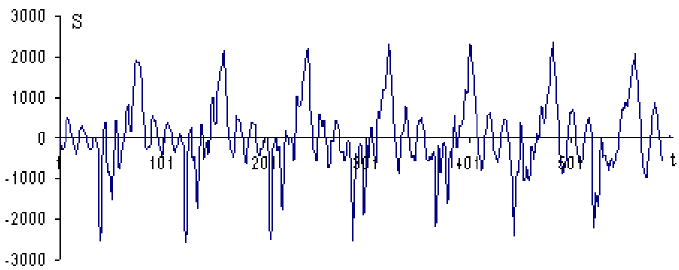
4 APPLICATION OF ALGORITHMS FOR NUMERICAL CALCULATION OF FAST CONTINUOUS WAVELET TRANSFORM FOR SIGNAL PROCESSING

4.1 Compression of a one-dimensional and two-dimensional signal

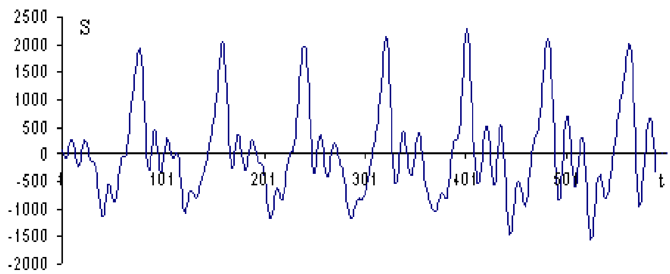
Discrete and continuous WT is redundant, since the number of wavelet coefficients exceeds the number of samples of the original signal. In order for the number of wavelet coefficients to be equal to the number of samples of the original signal, a discrete WT uses a sub-band encoding (filtering) algorithm. This algorithm is closely related to MSA. Sub-band filtering is not just used for decomposition and recovery, but the purpose, according to I. Daubechies, is compression or processing between the stages of decomposition and recovery. Compression after sub-band filtering is more feasible than in the absence of filtering in many applications. Compression after sub-band filtering is performed by discarding the wavelet coefficients with a small value, without noticeable signal distortion. In this way, data reduction is achieved. In this regard, discrete WT is widely used for information compression, since there is no similar algorithm for continuous WT, and continuous WT is not used for compression.

The algorithm developed by the author for signal reconstruction also allows the use of sub-band encoding for continuous WT. Using the *Sinc*-wavelet and multiplying it by the cosine, we can create low-pass and high-pass filters. By sequentially passing the signal through low-pass and high-pass filters and using decimation, we can repeat the Mall algorithm for continuous WT.

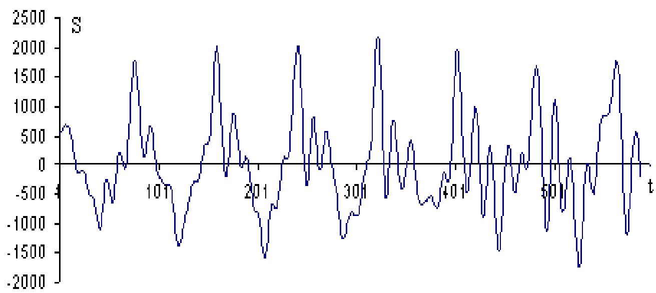
Consider an example of signal compression and applying fast continuous WT without using decimation and interpolation. If we examine the wavelet coefficients or complex conjugate Fourier coefficients for different levels of decomposition m, we see that for large scale coefficients a, the wavelet coefficients are almost the same throughout the signal. The Fourier coefficients have very few coefficients with large amplitudes. Also, the energy that falls on the average values of m for the acoustic signal is much greater than on the large and small levels of decomposition. There are differences for other types of signals. Information compression for a continuous WT can be performed either in the region of the wavelet coefficients or in the region of the Fourier coefficients by removing the coefficients with a small value. Removing coefficients with a small value of the wavelet coefficients is called threshold signal processing. Figure 4.1.1 shows the graphs of the speech signal $S(t)$ and its variants, compressed 3 and 6 times.



a



b



c

Figure 4.1.1. Signal compression

The acoustic signal $S(t)$ is divided into 12 levels of decomposition. Figure 4.1.1 shows the $\frac{1}{4}$ part of the signal. The compression was performed in the region of Fourier coefficients, since when finding the complex conjugate spectrum of an acoustic signal and a wavelet, the wavelet spectrum at each level leaves the Fourier coefficients of the signal large in a certain frequency range, and outside this range with very small values. In this regard, at each level of the decomposition, it is sufficient to leave the coefficients in a narrow spectral range for large scale coefficients. The smaller the scale factor value, the wider the range, and the center of this range is shifted towards high frequencies. By ear, the difference between the original signal and its compressed version is almost imperceptible. The Pearson correlation coefficient for a signal compressed by a factor of 3 is 0.890. For a signal compressed 6 times, – 0.817.

If we try to compress the same signal using FT, the results will be much worse. Even when half of the Fourier coefficients are removed, the restored signal is severely distorted because the time of occurrence of frequencies in the signal is lost with FT. Regardless of whether a certain frequency appears at the beginning or end of a signal with a certain duration, the position and amplitude of this harmonic will be the same in the Fourier spectrum. If the same frequency appears at the beginning and end of the signal, the amplitude of this harmonic will increase according to the total duration. In other words, removing any harmonics can change the waveform very much. With MSA, the signal is decomposed into several levels, and the appearance of a certain frequency of any duration is manifested on the Fourier spectrum of the m-th level. At other levels, the amplitude of this harmonic will be very small.

There are many methods of compressing information. Just like for discrete WT, it is possible to compress the signal using wavelet coefficients. Sub-band coding specialists prefer symmetry, since less asymmetry leads to more compressibility of the signal. Symmetric filters are called linear phase filters, i.e. they have a linear phase-frequency response. Unlike discrete wavelets, continuous wavelets are symmetric and smooth functions, so they are more suitable for information compression.

4.1.1 Two-dimensional continuous fast WT

An example of a two-dimensional signal is an image. Image processing in one way or another is carried out by specialists in almost all fields of knowledge. It is difficult to name a field of science and technology where image processing is not used. Currently, using digital image processing, such complex mathematical problems as the identification of an individual by fingerprints, photographs, image restoration, projections in tomography (used in medical diagnostics to obtain images of internal organs), etc. are solved. The transition from the one-dimensional case to the two-dimensional case is not only quantitative, but also qualitative. Many of the problems encountered in the processing of multidimensional signals are absent in the processing of one-dimensional signals, and conversely, many of the difficulties of digital signal processing are absent or easily removed when switching from one-dimensional signals to multidimensional ones [63].

The developed algorithms for forward and inverse fast WT in the frequency domain can be used for two-dimensional continuous WT. Here, a continuous two-dimensional WT means that the function $z = f(x, y)$ depends on two variables x and y. The function z is a three-dimensional object. If the object is an image, the z function is generated by encoding each pixel of the M-row and N-column with grayscale so that the darker areas correspond to smaller values and the lighter areas correspond to larger pixel values. WT methods are also suitable for color images with three color components.

To do this, we need to perform WT independently on each of the three color components of the image and present the results as an array of vector-valued wavelet coefficients. For a discrete two-dimensional wavelet transform for many applications, a construction is used in which the wavelet bases are obtained by the tensor product of two one-dimensional multiple-scale analyses of columns and rows. There is a standard and non-standard construction of a two-dimensional basis. The standard construction of a two-dimensional wavelet basis involves taking all possible tensor products of functions of a one-dimensional basis. For a standard image decomposition, we need to perform a one-dimensional transformation of all rows, and then all columns. A non-standard construction of a two-dimensional basis produces one scaling function and three wavelets, called horizontal, vertical, and diagonal. In this design, low-and high-frequency filtering is repeated across rows and columns by applying all four possible combinations. The discrete wavelet transform for a non-standard two-dimensional basis is given by the scheme

$$z \to \{H_r H_c z, H_r G_c z, G_r H_c z, G_r G_c z\}$$

where H is low-pass filtering, G high-frequency filtering.

The r index indicates that the filter is applied to rows, and the c index is applied to columns. If the signal (image) is given by an array of $N \times N$, then each array of approximating and detailing (for horizontal, vertical, and diagonal wavelet) coefficients of the first level consists of $\frac{N}{2} \times \frac{N}{2}$ elements

$$\begin{pmatrix} z^1 & d^{1h} \\ d^{1v} & d^{1d} \end{pmatrix}$$

where z^1 – approximating coefficients of the scaling function, d^1 with indices h, v, d – detailing coefficients of the horizontal, vertical, and diagonal wavelet.

For the second level $\frac{N}{4} \times \frac{N}{4}$ elements, instead of z^1, the approximating and detailing coefficients of the second level are also formed, for the third level $\frac{N}{8} \times \frac{N}{8}$ elements, approximating and detailing coefficients of the third level, etc. The decomposition of the signal into wavelet series at a given resolution level m is performed using these coefficients.

In order to reconstruct a two-dimensional signal using the developed forward and inverse WT algorithm in the frequency domain, we first convert this signal into a one-dimensional one. In this case, the transformation of a two-dimensional signal into a one-dimensional one can be carried out by a sawtooth, a triangular scan in rows and columns, and a zigzag scan diagonally. After decomposition, the one-dimensional m-level signal is converted to a two-dimensional m-level signal of the decomposition. Figure 4.1.1.1 shows a block diagram of the device for direct and inverse two-dimensional fast continuous WT.

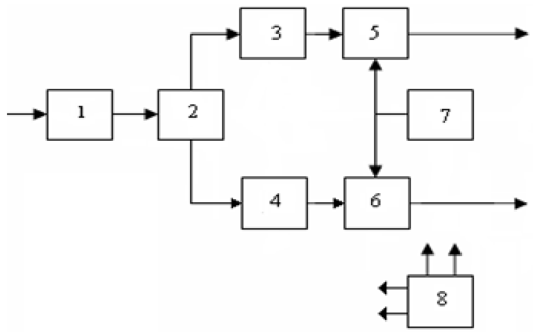


Figure 4.1.1.1. Block diagram of the device of two-dimensional direct fast WT: 1 – Analog-to-Digital Converter (ADC); 2 – random access memory; 3 – scanner row; 4 – scanner columns; 5, 6 – calculators fast continuous wavelet transform; 7 – persistent storage device; 8 – control device

The analyzed signal $S(x, y)$ is sent to the ADC (block 1), from the output of which a discrete sample $S(n, n)$ is sent to the input of the RAM (block 2). From the output of block 2, a two-dimensional sample of the signal simultaneously enters the inputs of the two-dimensional signal scanners in one-dimensional rows and columns (blocks 3, 4). From the scan blocks, one-dimensional signals with the number of samples $n \times n$ are sent to the inputs of the continuous fast forward WT calculators (blocks 5, 6), from the outputs of which the results of the WT signal are taken as an array of values of the wavelet coefficients with the size M of scales on N shifts $W(m, n)$. The control device (block 8) synchronizes the operation of ADC units (block 1), RAM (block 2), scanners (blocks 3, 4), and continuous fast direct WT calculators (blocks 5, 6).

This device allows us to select various types of wavelet functions with an arbitrary sampling step of scale coefficients stored in ROM (block 7) for analyzing a two-dimensional input signal. The calculation of the continuous fast forward WT in blocks 5, 6 is similar to the one-dimensional WT algorithms considered.

Since the inputs of these blocks receive a digitized signal, there is no need to have an ADC. Also, the ROMs and control device are taken out of these blocks and combined. A block diagram of a two-dimensional inverse fast continuous WT device is shown in Figure 4.1.1.2.

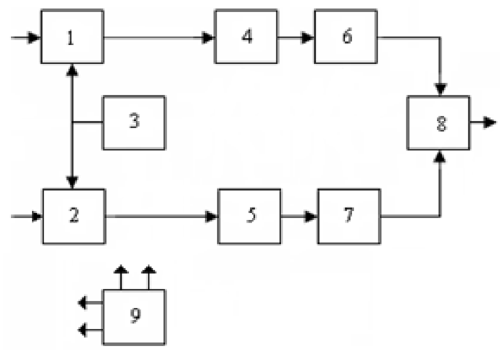


Figure 4.1.1.2. Block diagram of a two-dimensional inverse fast WT device: 1, 2 – continuous inverse fast WT calculators; 3 – permanent memory; 4, 5 – random access memory; 6, 7 – blocks for converting a one-dimensional array into a two-dimensional one; 8 – adder; 9 – control device

Two wavelet spectra $W(m,n)$ are received at the inputs of the blocks of continuous fast inverse WT (blocks 1, 2), from the output of which the reconstructed signals $s1(n)$, $s2(n)$ are received at the inputs of the RAM (blocks 4, 5). From the outputs of blocks 4, 5, signals $s1(n)$, $s2(n)$ are received at the inputs of the blocks for converting a one-dimensional signal into a two-dimensional one (blocks 6, 7). From blocks 6, 7, two-dimensional signals are received at the input of the adder (block 8). In block 8, the sum of two two-dimensional arrays is calculated, the elements of which are previously divided into two. From the output of block 8, the results of the inverse fast continuous WT two-dimensional signal are taken in the form of an array of values $S(n,n)$. The control device (block 9) synchronizes the operation of the continuous fast inverse WT blocks (blocks 1, 2), RAM (blocks 4, 5), blocks for converting a one-dimensional signal into a two-dimensional one (blocks 6, 7), and the adder (block 8).

This device allows us to select various types of wavelet functions with an arbitrary sampling step of scale coefficients stored in ROM (block 3) for the synthesis of a two-dimensional signal. The calculation of the continuous inverse of the fast continuous WT in blocks 1, 2 is similar to the considered algorithms of the one-dimensional inverse WT. Also, the ROM and control device blocks are moved outside of these blocks and combined.

4.1.2 Compression of a two-dimensional signal

Until the 1990s, the most common compression methods used in the *JPEG* and *MPEG* standards, based on the FT of the signal (discrete cosine transformation). *JPEG* (*JointPhotographicExpertsGroup*) – is a group of photography experts, a standard for encoding and compressing still images. *MPEG* (*MovingPictureExpertsGroup*) – is a group of experts on moving images, a standard for encoding and compressing moving objects, and video images. In the early 1990s, a new standard called wavelet compression was developed.

The *JPEG*-2000 standard uses discrete WT for image compression. The practical implementation of video compression is carried out by means of a two-band filter bank, known as sub-band encoding. Video compression is usually understood as a reduction in the amount of memory required to store digital video data and transmit it over communication channels. The goal of video compression is a more compact representation of images. The development of effective video compression algorithms is important for the creators of digital video surveillance systems, graphics, videos, etc. There are two main types of redundancy that we can use.

The first of these is intra-frame or spatial redundancy. It can be identified by one current video frame without referring to any other video frame. The second type is inter-frame, or temporary, redundancy, which requires both the current and the next video frame to be identified. In every real image, there is a spatial redundancy. If the image contains an object of a sufficiently large size, then all the elements representing this object have very close values. Large objects create low spatial frequencies, while small objects create high spatial frequencies [15].

Video compression takes place in several stages, and one of them is compressed using discrete WT. At other stages, time redundancy is removed, i.e. there is the removal of identical regions in frames, quantization with variable length (entropy coding, Huffman), and data archiving. Compression with WT, like compression with discrete cosine transformation (DCT), is based on the position that most of the energy is concentrated in a small number of coefficients, which are quantized according to their value. The concentration of energy in a few significant coefficients is called energy localization, it is the main prerequisite for data compression.

When using DCT, the image is transformed block by block. A typical block contains 8×8 image samples. As a result of the DCT of such a block, a block of 64 coefficients is formed. The coefficient is a number that expresses the degree to which a particular spatial frequency is present in the image. DCT itself does not give any compression. Moreover, the codeword lengths for the coefficients are longer than for the original image samples. However, the DCT provides a conversion of the image samples into a form that allows the redundancy to be identified.

Since not all spatial frequencies are present at the same time, the DCT forms a set of coefficients, some of which have relatively large values, but many coefficients are small or equal to zero. Obviously, if the coefficient is zero, it does not matter whether it is transmitted or not. If the coefficient has a small, almost zero value, then its exclusion has the same effect as adding a weak interference with the same spatial frequency, but with the opposite sign. The decision to exclude a certain coefficient is made based on the visibility of such a small-amplitude signal. If the coefficient is very large and cannot be excluded, compression can still be achieved by reducing the number of bits used to transmit this coefficient. This operation is also accompanied by a certain negative effect, namely, a small level of noise is added to the image.

The visibility of the spatial frequencies observed by humans is not the same, and much higher noise levels are acceptable at high frequencies. In this regard, the coding uses a weighting operation, which ensures the concentration of any resulting noise in the high frequency region. Each coefficient is divided by the weight, which is a function of its position in the block. The constant component is not weighed at all, and the weight of the coefficient (the divisor) increases as it approaches the lower-right corner. In the decoder, we need to perform an operation that is the inverse of the weighting. In this case, the coefficients for higher frequencies are multiplied by larger numbers, which leads to increased noise at these high frequencies. After weighing, some small coefficients become even smaller. In a television image, the coefficients with large values are usually located in the upper-left corner, and the remaining coefficients are often negligible or equal to zero.

In this regard, it is advantageous to transmit the coefficients in a zigzag sequence, starting from the upper-left corner. As a result of this process, non-zero coefficients are usually transmitted first, and zero coefficients are transmitted at the very end. At some point, starting from which all subsequent coefficients are equal to zero, it is advisable to interrupt the transmission by using a simple symbol that tells the receiver that there will be no non-zero coefficients [15].

A reconstruction algorithm that uses multiple-scale analysis is a way to concentrate all the available information in a signal in a few significant coefficients. There is a distinction between lossless and lossy image compression. The first is characterized by insignificant compression ratios. When compressing an image with acceptable losses, the compression ratio may be large. WT can be used for both lossless and lossy image compression.

It is known that the operation of multiscale image representation is performed in the human eye. The eye is more sensitive to distortion in the low-frequency region. Any image contains redundant information that is not perceived by the human eye. Therefore, when compressing a two-dimensional image, it is necessary to take into account the human vision system. It is possible to improve the visual quality of the reconstructed image by applying algorithms that take into account the sensitivity of the eye in different frequency ranges. The discrete WT is not spatially invariant due to the presence of decimation and interpolation. This variability in space interferes with the compact representation of video signals. The developed algorithm for two-dimensional continuous WT does not use decimation and interpolation, and there is also the possibility of compression in the frequency domain, so it is invariant to the shift.

Therefore, it is possible to achieve good compression results by transmitting the same information only once, from frame to frame, as is done using the discrete cosine transform in the *MPEG* standard. Using the developed reconstruction algorithm allows us to transform a two-dimensional object both in rows and columns individually, and the entire object as a whole. As a result, such objects can be preserved using few coefficients, because for periodic objects, the complex conjugate spectrum has large values for few coefficients.

For a family of discrete Daubechies wavelets, dbn, the concentrating mechanism works more efficiently with increasing n. In other words, the energy localization effect for the $db2$ wavelet is better than for the $db1$ wavelet, because the $db2$ wavelet has two zero moments. As n increases, the number of zero moments increases proportionally to n, and the wavelet symmetry also improves.

The use of multiple iterations improves the smoothness of the wavelet, and it becomes really smooth only when the number of iterations tends to infinity. We can say that the use of continuous wavelets based on derivatives of the Gaussian function of large orders, for example, of the 10^{th} order, should increase the compression ratio without noticeable distortion of the signal. The number of zero moments for continuous wavelets based on the derivatives of the Gaussian function increases in proportion to the order of the derivative, and symmetry and smoothness are inherent a priori. Figure 4.1.2.1 shows an image of a girl without compression. Figures 4.1.2.2 – 4.1.2.4 show images of the girl compressed 10, 22, and 52 times using a wavelet based on the second-order derivative of the Gauss function (*MHAT*-wavelet).

Figure 4.1.2.1. Image of a girl

Figure 4.1.2.2.10x compression

Figure 4.1.2.3. 22x compression

Figure 4.1.2.4. 52x compression

When compressing, the conversion from the RGB color model to the YCbCr model is not performed, as in JPEG, statistical encoders are not used, and the WinRAR archiver was used.

4.2 Investigation of the dependence of the average grain size of ceramics on temperature

To calculate the statistical parameters of the sample, the wavelet transform of the image obtained using an electron microscope is used. The proposed method provides a significant increase in the calculation speed due to the implementation of multiple-scale image analysis. When studying the microstructure of a material, statistical parameters are often used, such as the average size of the microparticles, the dispersion, the standard deviation of the size, and the coefficient of variation.

To calculate these parameters, the diameters of a very large number of microparticles are measured, which is a long and time-consuming process [29]. To calculate these parameters, a continuous fast WT image obtained by an electron microscope is used. To determine the average size of ceramic grains, the intensity of each pixel of the image with a size of 512×512 in bmp format is read by horizontal and vertical progressive scanning, and the WT of the image with different scale coefficients is calculated. The image is decomposed into 100 levels and a histogram of the distribution of the total intensity J, which falls on each level of decomposition, is constructed. The average grain size of ceramics D is calculated by the formula

$$D = \frac{\sum_{i=1}^{50} J_i i}{\sum_{i=1}^{50} J_i} \tag{4.1}$$

where i is the decomposition number, J_i is the total intensity of the i-th decomposition. To determine the dispersion of the ceramic grain size C, the formula below is used

$$C = \frac{\sum_{i=1}^{50} J_i (i-D)^2}{\sum_{i=1}^{50} J_i} \tag{4.2}$$

where i is the decomposition number, J_i is the total intensity of the i-th decomposition. The standard deviation of the ceramic grain size S is calculated by the formula

$$S = C^{\frac{1}{2}} \tag{4.3}$$

By formulas (4.1), (4.2), and (4.3) calculate the average size D of the ceramic grains, the variance, and the standard deviation on a logarithmic scale. To determine these values at the image scale, the base of the logarithm is found from the ratio

$$x^{100} = 2^{18}$$

The average size of ceramic grains in the image scale is calculated by the formula

$$D_{cp} = x^D \tag{4.4}$$

The standard deviation of the ceramic grain size in the image scale is calculated by the formula

$$S_{cp} = x^S \tag{4.5}$$

The measurement method [10, 52] has been tested on various objects. In [54], it is shown that the relative error in measuring the size of objects is the same at all levels of decomposition, i.e. the measurement quality is the same for both small particles and large particles. The average size of ceramic grains is calculated by formula (4.4) and the standard deviation of the size by formula (4.5) in the range from 1000 to 1250 degrees Celsius. Figure 4.2.1 shows the temperature dependence of the average ceramic grain size. The actual average grain size of ceramics, taking into account the image scale at a temperature of 1000 degrees, is 420 nm, and the standard deviation is 20 nm.

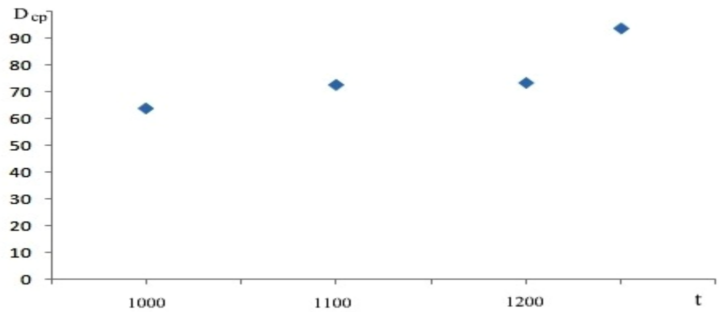


Figure 4.2.1. Dependence of the average grain size of ceramics on temperature

The size is somewhat overstated because the images are obtained for the fracture of the ceramic and not for the slot. For the fracture of ceramics at large scales, the intensity of objects is high, which is typical for fractals. This is due to the fact that the images of ceramic grains look like clouds. To assess the degree of uniformity of the size, a value called the coefficient of variation is also used. The coefficient of variation is calculated by the formula

$$\nu = \frac{S_{cp}}{D_{cp}}$$

Figure 4.2.2 shows the temperature dependence of the coefficient of variation.

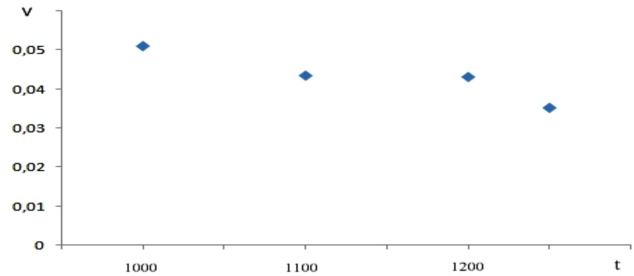


Figure 4.2.2. Dependence of the coefficient of variation on temperature

A decrease in the coefficient of variation with increasing temperature indicates that the degree of uniformity increases with increasing temperature. Calculations show that as the temperature increases, the average size of ceramic grains increases, which coincides with traditional measurement methods, and the time spent obtaining the dependence is many times less and there is no need to have any device or tool for measuring the size of ceramic grains [21]. This method of measuring the statistical characteristics of micro-objects allows one to save time and costs.

4.3 Comparison of the algorithm for multiple-scale image analysis in the frequency domain with the algorithm presented in the MatLab computer mathematics system

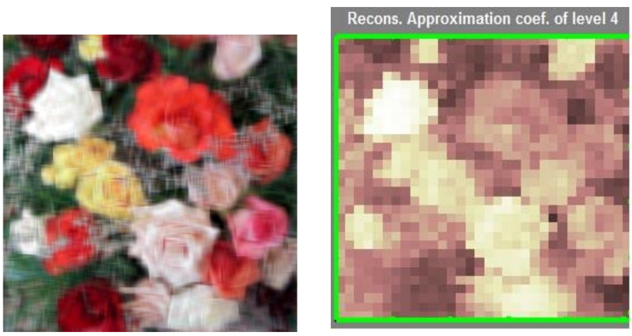
The MSA algorithm uses frequency domain wavelets based on the derivatives of the Gaussian function [30, 33, 35, 49]. The main advantage of these wavelets is that they are smooth and symmetric functions with N-order derivatives. Just such functions are necessary for WT. However, in scientific literature and on the Internet, the following is noted for the continuous wavelet transform:

- the analysis is not orthogonal
- the wavelet does not have a compact carrier
- there is a missing scaling function
- the possibility of reconstruction is not guaranteed
- there is an excessive number of wavelet coefficients, far exceeding the number of samples in the original signal

- fast continuous wavelet transform algorithms and accurate reconstruction are not possible

For these reasons, MSA does not use wavelets based on the derivatives of the Gaussian function. In fact, all these points are not true. The use of continuous WT gives better results and additional signal processing capabilities that are not possible with discrete WT. In this case, WT is performed for the entire image by a sawtooth scan in rows and columns. Unlike the Mall algorithm, this algorithm allows us to get many more levels of decomposition, thereby allowing us to study the image in more detail.

Figure 4.3.1a shows a color image with a size of 512×512 pixels, reconstructed with approximating coefficients of the fourth level for the algorithm in the frequency domain using the *MHAT* wavelet. Also, when using this algorithm, there is no mosaic when the image is approximated by high-level coefficients. In fact, for the algorithm developed by the author, there are no approximating and detailing coefficients, but there are decomposition levels corresponding to the decomposition levels in the Mall algorithm, and we can compare the results of the decompositions. Figure 4.3.1b presents a reconstructed image with approximating coefficients of the fourth level for the Mall algorithm using the Daubechies wavelet (db2). The calculation was carried out in the MatLab computer mathematics system.

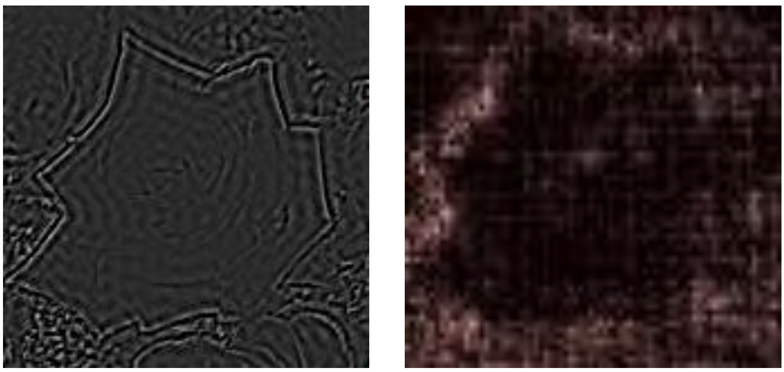


a **b**

Figure 4.3.1. Reconstruction of the image with approximating coefficients of the fourth level

In Figure 4.3.1b, the mosaic is clearly visible. Despite the fact that the conversion time in the Mall algorithm is almost the same as the algorithm in the frequency domain, the quality is much worse. With an increase in the order of the wavelet (dbN) or with the use of other listed wavelets, the mosaic decreases, but it still turns out worse than for wavelets based on the derivatives of the Gaussian function. Reconstruction of the image with detail coefficients when also using the algorithm in the frequency domain gives a clearer image than in the MatLab computer mathematics system.

Figure 4.3.2*a* shows the reconstructed image with the first-level detail coefficients for the frequency domain algorithm using the *MHAT* wavelet. Figure 4.3.2*b* shows a reconstructed image with the first-level detail coefficients for the Mall algorithm using the Daubechies wavelet (db2). The presented images are enlarged for a better view. In Figure 4.3.2*a*, the borders of the flower petals are clearly visible, and in Figure 4.3.2*b*, everything is blurred and there is no clear border of the petals. This is due to the fact that the phase-frequency characteristics of wavelets based on the derivatives of the Gaussian function have linear characteristics, and the phase-frequency characteristics of Rowdy wavelets are nonlinear since they are non-symmetric functions.



a **b**

Figure 4.3.2. Reconstruction of the image with the detailing coefficients of the first level

Figures 4.3.3*a* and 4.3.3*b* show reconstructed images using all levels for the algorithm in the frequency domain and the MatLab computer mathematics system.

a **b**

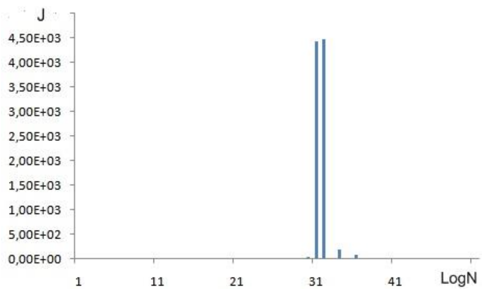
Figure 4.3.3. Reconstructed images in the frequency domain in MatLab

When reconstructing an image using all levels in MatLab, many colors are lost, and when using the algorithm in the frequency domain, all colors are preserved and the reconstructed image does not differ from the original (Figure 4.3.3). In contrast to the discrete WT, based on the developed algorithm in the frequency domain, it is possible to perform multiple-scale analysis of images with a multiplicity of less than 2. This multiple-scale analysis allows us to view the image from the largest scale to the smallest, decomposing the image into dozens of levels. When moving from one level to another, the changes in the image are barely noticeable to the eye. Based on this algorithm, a microfilm is created, in which different images smoothly replace each other. Multiscale analysis of images with a multiplicity of less than 2 also allows one to determine the statistical parameters of the image, such as the average size, the standard deviation of the size and the coefficient of variation of micro-and macro-objects [15, 19, 36-38, 46, 68, 76].

Satellite images are used to determine the average size of buildings in a particular area of a city or natural objects. A horizontal and vertical progressive scan reads the intensity of each pixel of the image size 512×512 in bmp format, and it calculates the wavelet transform of the image with different scale coefficients. The image is decomposed into 100 levels and a histogram of the distribution of the total intensity J, which falls on each level of decomposition, is constructed. In order to get the result, a few tens of seconds are enough. The technique was tested on images of simple objects with the same size and on images of fractals.

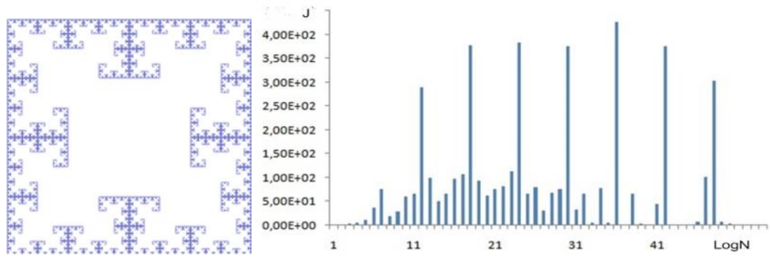
Figure 4.3.4*a* shows an image of objects that have almost the same size. Figure 4.3.4*b* shows the intensity distribution from the level of decomposition of these objects. In Figure 4.3.4*b*, the peaks that correspond to the intensity of objects of a given size are clearly distinguished. Figure 4.3.5*a* shows an image of the T-fractal. Figure 4.3.5*b* shows the intensity distribution from the decomposition level for this fractal. For fractals, the peaks corresponding to the n-th generation prefractal are well distinguished, i.e. several peaks located at the same distance from each other are distinguished. The average size of objects D is calculated using the formula (4.1). The average size of objects in the image scale is calculated using the formula (4.4). Figure 4.3.6*a* shows a satellite image of the center of Sofia, the capital of Bulgaria. Figure 4.3.6*b* shows the intensity distribution from the decomposition level for this image. The average size of the structures for this area is 68 meters.

The average size is somewhat overstated, since all levels of decomposition are taken into account. Decomposition levels above 32 take into account the fractality of cities, so they do not need to be taken into account. In the same way, the average size of all the capitals of the republics and regions of Russia is calculated, but without taking into account the levels above 32. Figure 4.3.7 shows a graph of the average size of all the capitals of Russia.



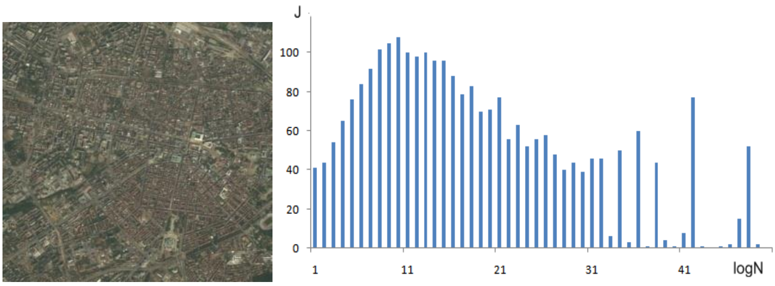
a **b**

Figure 4.3.4. The image of objects and the distribution of intensity from the level of decomposition



a **b**

Figure 4.3.5. T-fractal and intensity distribution from the decomposition level



a **b**

Figure 4.3.6. The center of Sofia and the distribution of intensity from the level of decomposition

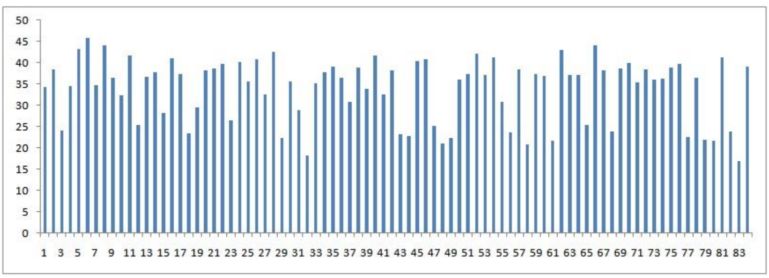


Figure 4.3.7. Average size of the capital's

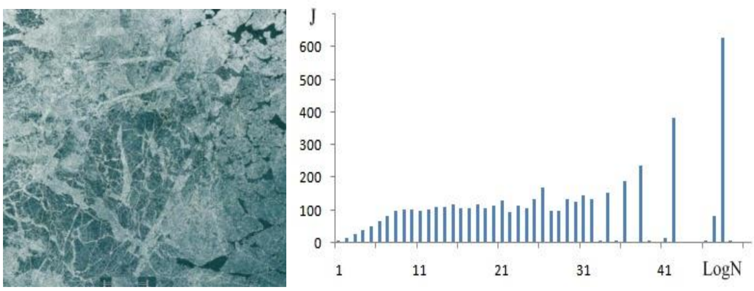
The cities are arranged in the same order as in maps online. Number 6 on the chart shows the average size of the city of Belgorod, while number 83 shows the average size of the city of Salekhard. Based on the calculated values of the average size, we can say that in Belgorod the size of the structures is the largest for this area, and in Salekhard the smallest. The remaining capitals have an average size of structures between 18 and 46 meters. Figures 4.3.8*a* and 4.3.8*b* show satellite images of Novosibirsk and Grozny.

a **b**

Figure 4.3.8. Satellite images of Novosibirsk and Grozny

If one compare the images from space of Novosibirsk and Grozny, we can immediately see that in Novosibirsk there are more large structures, and in Grozny there are more small structures. The calculated average size of the structures in Novosibirsk is 37 meters, and the average size of the structures in Grozny is 22 meters. In this way, we can compare the other capitals.

Figure 4.3.9*a* shows a satellite image of the ice floes of the Arctic Ocean. Figure 4.3.9*b* shows the intensity distribution from the decomposition level for this image.



a **b**

Figure 4.3.9. Image of ice floes and intensity distribution from the decomposition level

In Figure 4.3.9, the average size of the ice floes, calculated using the formula (4.4), is 30.83 units in the image scale. The real average size of the ice floes, taking into account the image scale, is 1200 meters. Thus, by calculating the average size of ice floes in different places in real time, we can quantify the ice situation or trace the dependence of the average size of ice floes on the time of year for a particular place [10, 21, 35, 49, 50, 52, 54, 55, 74]. Such measurements allow us to plot the route of the movement of sea vessels along optimal trajectories, thereby saving time, fuel and ultimately financial costs.

4.4 Calculation of the anisotropy measure using the continuous fast wavelet transform

In crystals, the mechanical, electrical, magnetic, and optical properties depend on direction, i.e. the crystals have anisotropy. This is due to the fact that in crystals, the atoms, molecules, or ions are arranged in the correct order. With the correct arrangement of the atoms, they are placed along different directions with different densities. The atoms are located at the nodes of the spatial lattice. If we draw planes through the lattice nodes in different directions, we can see that the density of the arrangement of atoms on these planes is different. Consequently, there are planes in crystals in which the atoms are more strongly bound to each other.

There are planes in which the atoms are more loosely bound to each other, so the mechanical and other properties along these planes are different. Also, houses and structures are arranged in the correct order, as cities are built by people. A consequence of this order is that cities have anisotropy. The anisotropy of cities is related, for example, to the direction in which the wind speed is greatest under given weather conditions, or how fast it is possible to get from one point of a city to another point. If we look at satellite images of cities on a large scale, it is difficult to see the measure of anisotropy of cities, i.e. to what extent structural objects (houses, factories, or stadiums) are oriented in one direction or another. The anisotropy measure is calculated from the results of the wavelet transform of satellite images. The image is decomposed into different levels with multiplicity less than 2. The horizontal and vertical progressive scan reads the intensity of each pixel of the 512×512 image in bmp format and calculates the wavelet transform of the image with different scale factors. The image is decomposed into 100 levels, and then a histogram of the distribution of the total intensity J, which falls on each level of decomposition, is constructed.

The average size of city objects is calculated with horizontal and separate vertical image scans. The measure of anisotropy is calculated by the formula

$$a = \frac{D_x}{D_y}$$

where D_x is the average size of objects with a horizontal scan; D_y is the average size of objects with a vertical scan.

The anisotropy measure shows how the structures are oriented: from north to south $a < 1$ or from west to east. Figure 4.4.1 shows an image of one of the districts of New York City. Figure 4.4.2 shows the distribution of intensity from the level of decomposition in the horizontal scan. Figure 4.4.3 shows the distribution of the intensity from the level of decomposition in the vertical scan. For the New York City area represented, the anisotropy measure is 1.127 units. Figure 4.4.4 shows the dependence of the anisotropy measure on the angle of rotation of the satellite image of one of the districts of New York city.

Figure 4.4.1. Image of one of the districts of New York city

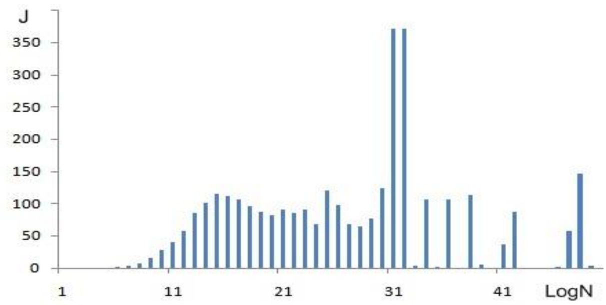


Figure 4.4.2. The distribution of intensity from the level of decomposition in the horizontal scan

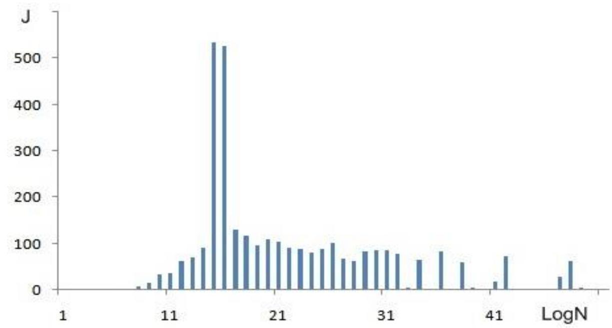


Figure 4.4.3. The distribution of intensity from the level of decomposition in the vertical scan

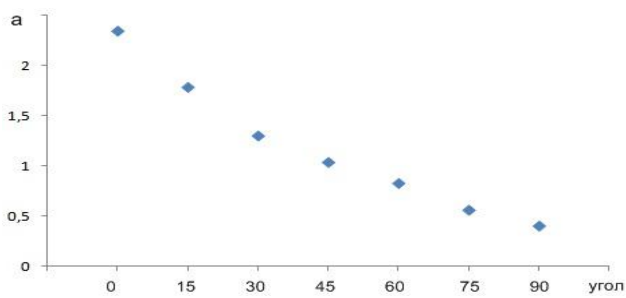


Figure 4.4.4. Dependence of the anisotropy measure on the angle of rotation

When the angle of rotation is 45°, the measure of anisotropy is 1, i.e. the structures are oriented mainly in the north-east direction. When the angle of rotation is 90°, the structures are oriented mainly in the direction from north to south. Calculating the measure of anisotropy, it is possible to determine the angle between the north-south direction and the direction under which the buildings of cities are mainly oriented [51, 53].

The design of wavelets with a large number of zero moments is relevant since they concentrate information more efficiently and also allow analyzing a more subtle (high-frequency) structure of the signal, suppressing its slowly-changing components. For the effective study of non-stationary signals, just such functions are needed. Wavelets for different scale coefficients a in the frequency domain are constructed taking into account the property of scale-frequency locality, so that they have more zero moments at all levels of decomposition.

In scientific literature, it is written that for practice it would be advisable to have orthogonal symmetric and antisymmetric wavelets, but such ideal wavelets do not exist. Due to the fact that orthogonal wavelets are constructed in the frequency domain, the resulting wavelets have almost perfect frequency characteristics, both amplitude and phase. They are almost perfect in the sense that they differ from the theoretical characteristics only in the error of calculation. Figure 4.4.5 shows a symmetric orthogonal time-domain wavelet constructed in the frequency domain.

Figures 4.4.5 and 4.4.6 show that the wavelets have many maxima and minima. There are even more of them, since only a fifth of the wavelets are represented in the figures. To obtain such wavelets in the time domain, it would be necessary to solve equations with the same number for each level of decomposition. Orthogonal symmetric and antisymmetric wavelets constructed in the frequency domain make it possible to reconstruct one - and two-dimensional signals with greater accuracy, since they have a linear phase-frequency characteristic. The conversion time is no longer than for discrete wavelets.

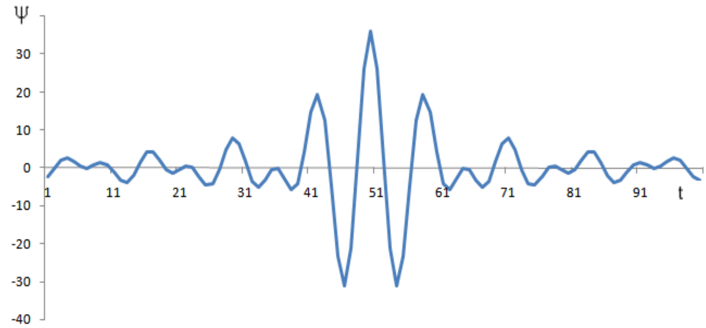


Figure 4.4.5. Symmetric orthogonal wavelet

Figure 4.4.6 shows an antisymmetric orthogonal time-domain wavelet constructed in the frequency domain.

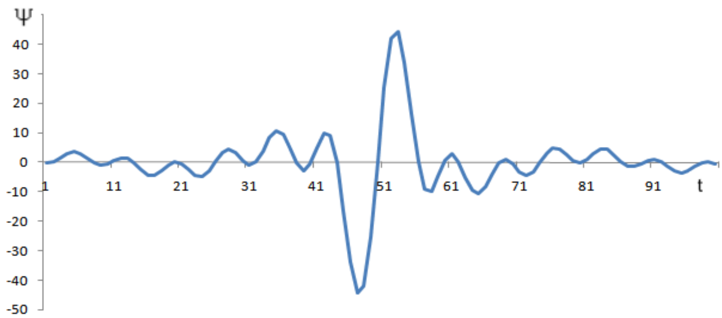


Figure 4.4.6. Antisymmetric orthogonal wavelet

If we consider the amplitude-frequency characteristic of such wavelets, we see that these wavelets have an ideal characteristic. Figure 4.4.7 shows the amplitude-frequency response of an antisymmetric orthogonal wavelet. It can be seen that there is no unevenness in either the passband or the delay band, and there is no transition band.

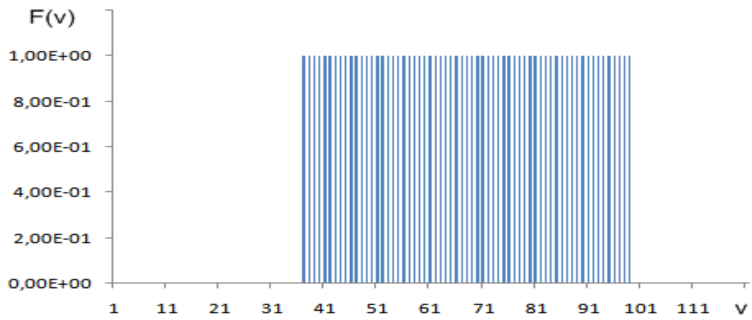


Figure 4.4.7. The amplitude-frequency response of an orthogonal wavelet

Figure 4.4.8 shows the amplitude-frequency response of a wavelet constructed in the time domain. The wavelet equation has the form:

$$\psi(t,a,b) = \frac{\sin\left(\frac{t-b}{a}\right)}{\left(\frac{t-b}{a}\right)}\cos(\omega t)$$

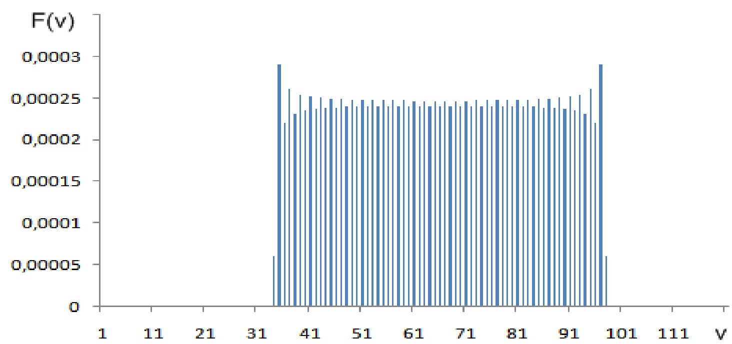


Figure 4.4.8. The amplitude-frequency response of a wavelet constructed in the time domain

Figure 4.4.8 shows that the amplitude-frequency response is uneven in the passband. This effect is called the Gibbs phenomenon. It was first studied in connection with the truncation of the Fourier series used for the harmonic decomposition of periodic signals. Figure 4.4.9 shows the frequency response of a wavelet constructed in the time domain and the frequency response of an orthogonal wavelet constructed in the frequency domain on the same graph in decibels. Up to a frequency of 260 units, the frequency response of a wavelet constructed in the time domain is represented, and from 260 to 512, the frequency response of a wavelet constructed in the frequency domain.

Figure 4.4.9 clearly shows how much the frequency characteristics of these wavelets differ when compared at the same scale. The steepness of the decay (attenuation) of a wavelet constructed in the frequency domain is much higher, i.e. the transition band is very narrow. If the wavelet has a higher slope, then the resolution is higher. Resolution refers to the ability to separately measure (isolate) the spectral responses of two sinusoidal signals of equal amplitude and differing in frequency. A condition for the resolution of two spectral lines in optics is the Rayleigh criterion for the case when the instrumental contour has a diffraction shape. In some cases, two spectral peaks are considered resolved if there is a point between these peaks where the second derivative is greater than zero (the line has a bulge looking down).

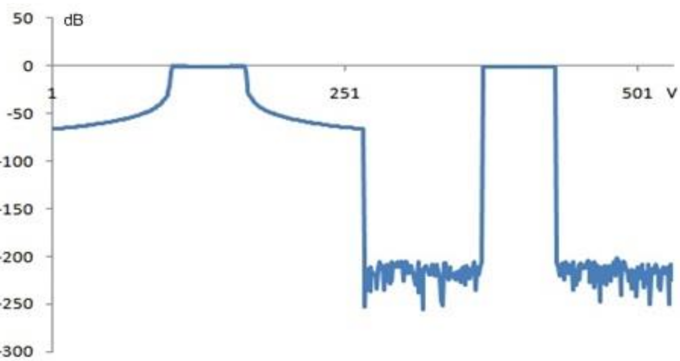


Figure 4.4.9. Amplitude-frequency response of a wavelet constructed in the time and frequency domain

In radio engineering, two spectral peaks are considered allowed if the gap between them has a value of at least 3 decibels, i.e. the gap is 70% of the maximum peak value. Indeed, it can be noted that the wavelets constructed in the frequency domain have an ideal frequency response because in the delay band the signal is attenuated 10^{21} times, i.e. a billion trillion times. All wavelets constructed in the time domain have an uneven amplitude-frequency response in both the passband and the delay band. There is no way to fix this, but one can only reduce it by a certain amount by applying various weighted functions (windows), such as Dirichlet, Hemming, Bartlett, Hanna, Blackman, and Kaiser windows. Currently, more than 50 types of windows are used.

In many textbooks on digital signal processing, methods for reducing the unevenness of the amplitude-frequency response are considered. It was noted above that the bandpass filtering of the signal occurs during the wavelet transform. Currently, digital filtering is so widespread that the volume of literature devoted to it exceeds the volume of literature on any other field of digital signal processing. If the algorithms of the wavelet transform in the frequency domain allow us to obtain the ideal impulse characteristics of wavelets, then we can also obtain the ideal impulse characteristics of low-pass, high-pass, notch, and blocking filters. Figure 4.4.10 shows the pulse response of the high-pass filter.

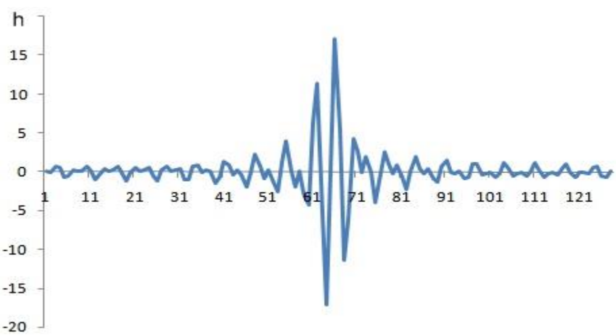


Figure 4.4.10. Antisymmetric pulse response of the high-pass filter

In the digital filter literature, such filters are called finite impulse response (FIR) filters. In filters with infinite impulse response (IIR), there is feedback, as in analog filters with feedback. IIR filters, in contrast to FIR filters, have a nonlinear phase-frequency response and are potentially unstable. They are used due to the fact that IIR filters can be implemented with a smaller number of calculations than FIR filters.

Figure 4.4.11 shows the pulse response of the band-pass filter.

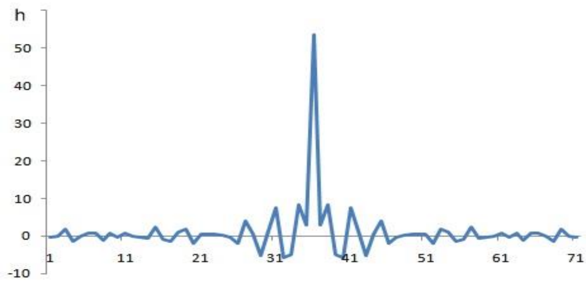


Fig. 4.4.11. Impulse response of the blocking filter

Digital filters allow us to filter signals by narrowing the frequency range of the signal. Consider how the signal is cleared of noise, i.e. frequency filtering of the signal with a narrowing of the frequency range. In addition to the frequency filtering method, the accumulation method, the correlation method (time filtering), and the matched filtering are also used. In any communication channel, during transmission, noise $u(t)$ is superimposed on the signal $x(t)$, resulting in a distorted signal

$$y(t) = x(t) + u(t)$$

Signal power to noise power ratio

$$r = \frac{P_x}{P_u}$$

In order to detect a signal with a high probability, it is necessary to increase the ratio of the useful component of the signal to the noise. This transformation is called filtering. After leaving the filter, a signal is obtained

$$z(t) = s(t) + g(t)$$

The ratio of signal power to noise power after leaving the filter is

$$m = \frac{P_s}{P_g}$$

The filtering task is to increase m over r. One of these methods is frequency filtering of the signal.

Let the signal be a sinusoid of a certain duration and frequency

$$x(t) = A\sin(w_0 t)$$

and the interference is "white noise". To isolate a useful signal of this type, bandpass filters are used that are tuned to the signal frequency. Real "white noise" has a uniform spectrum over a wide frequency range. At a finite frequency range, the noise power will be finite and the ratio of signal power to noise power r will be finite. If the average noise power per unit frequency is P_0 and the filter bandwidth is Δw, then the noise power at the filter output is

$$P_g = P_0 \Delta w$$

The power of the useful component of the signal will be the same as before passing the filter, since the filter is tuned to the frequency of the signal. The ratio of signal power to noise power at the filter output is

$$m = \frac{P_s}{P_0 \Delta w}$$

It follows from this formula that the smaller the passband of the filter Δw, the greater the ratio of signal power to noise power m. The same principle is used to filter signals using wavelets, since the wavelet transform, as shown above, is the transmission of a signal through a bandpass filter. We will demonstrate this by using an example where the noise and signal levels are almost the same [31]. Figure 4.4.12 shows a graph of the dependence of $H(t)$ on the time of the pipeline noise and the word "*cmon*". The word "*cmon*" was pronounced at a distance of 10 meters from the microphone against the background of the noise of the conveyor.

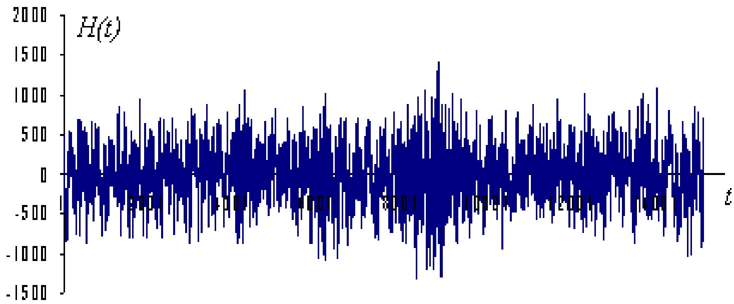


Figure 4.4.12. The word "*cmon*" against the background of conveyor noise

Before filtering, the graph shows that the noise level and the words are almost the same. After the wavelet transform and calculating the energy of the segments of the coefficients of the wavelet spectrum of this section, the energy of the word "$cmon$" is much higher, i.e. after filtering, the ratio of signal power to noise power has become much higher [31]. Figure 4.4.13 shows a graph of the energy of the segments of the wavelet coefficients $W(2, b)$ of the word "$cmon$". Thus, the wavelet transform can be used on a par with digital filters.

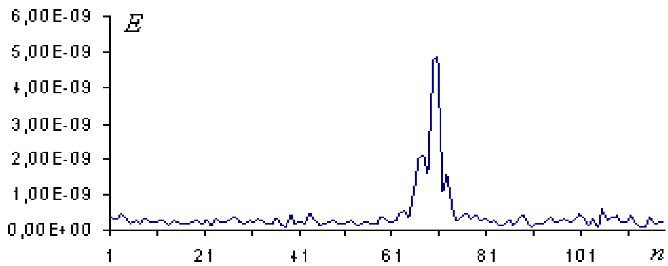
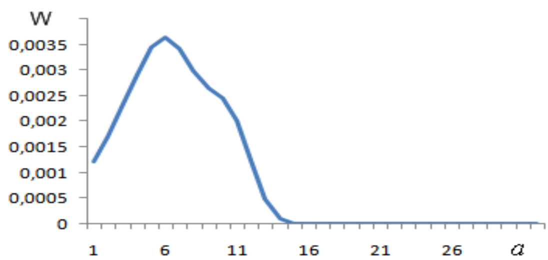


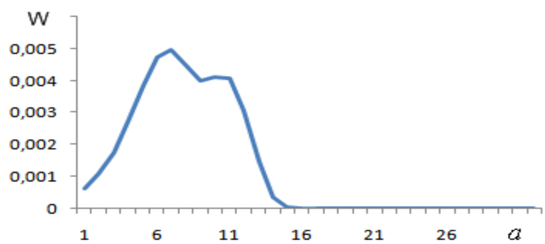
Figure 4.4.13. Energies of the segments of the wavelet coefficients of the word "*cmon*"

Orthogonal symmetric and antisymmetric wavelets can also increase the spectral resolution. Resolution refers to the ability to separately measure (isolate) the spectral responses of two sinusoidal signals of equal amplitude and differing in frequency. A condition for the resolution of two spectral lines in optics is the Rayleigh criterion for the case when the instrumental contour has a diffraction shape. In some cases, two spectral peaks are considered resolved if there is a point between these peaks where the second derivative is greater than zero (the line has a bulge looking down).

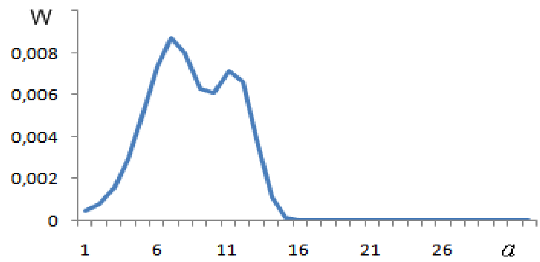
In radio engineering, two spectral peaks are considered allowed if the gap between them has a value of at least 3 decibels, i.e. the gap is equal to 70% of the maximum peak value. In optics, since light passes through spectral devices and diffraction occurs at the input slit, the Rayleigh criterion, according to which the minimum of one spectral line must coincide with the maximum of the other, leads to almost the same failure as in radio engineering. Let's compare the resolution of wavelets based on the derivatives of the Gaussian function and wavelets constructed in the frequency domain. The higher the order of the derivative of the Gaussian function, the narrower the spectrum at the same scale factor. The narrower the spectrum and the steeper the slope, the higher the resolution of the wavelet. Figure 4.4.14 shows the results of the wavelet transform of the sum of two sinusoids with frequencies slightly different from each other.



a



b



c

Figure 4.4.14. A slice of the wavelet transform with derivatives of the Gaussian function

Figure 4.4.14*a* uses the *MHAT*-wavelet, i.e. the second derivative of the Gaussian function, Figure 4.4.14*b* is the third derivative of the Gaussian function, and Figure 4.4.14*c* is the fourth derivative of the Gaussian function. For the fourth-order derivative of the Gaussian function, the resolution is better. Figure 4.4.15 shows the result of a wavelet transform of the same sum of sinusoids using an orthogonal symmetric wavelet constructed in the frequency domain. The resolution of this wavelet is much higher than for wavelets based on derived Gaussian functions.

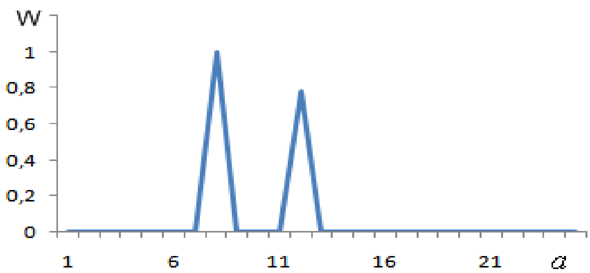


Figure 4.4.15. A slice of a wavelet transform based on an orthogonal symmetric wavelet

Figure 4.4.16 shows a cross-section of the wavelet transform for the sum of sinusoids with closer frequencies. For wavelets based on the derivatives of the Gaussian function, these sinusoids are not resolved at all, i.e. they are distinguished as a single sinusoid.

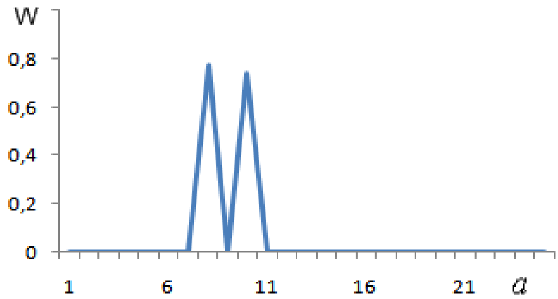


Figure 4.4.16. A slice of the wavelet transform based on an orthogonal symmetric wavelet

Figure 4.4.17 shows one of the orthogonal symmetric wavelets used to study the sum of sinusoids with close frequencies.

Figure 4.4.17. Orthogonal symmetric wavelet

Figure 4.4.17 shows the fifth part of the wavelet. It can be seen that the number of zero moments of this wavelet is very large, and its spectrum is the same as in Figure 4.4.7.

In the theory of the generalized Fourier transform, it is proved that the wavelets are orthogonal if the scalar transformation of these wavelets is zero. In the frequency domain, this statement corresponds to the fact that the product of the spectra of these vectors is zero, i.e. the spectra should not be superimposed on each other, as shown in Figure 4.4.18*a*. The spectra should be arranged as in Figure 4.4.18*b*.

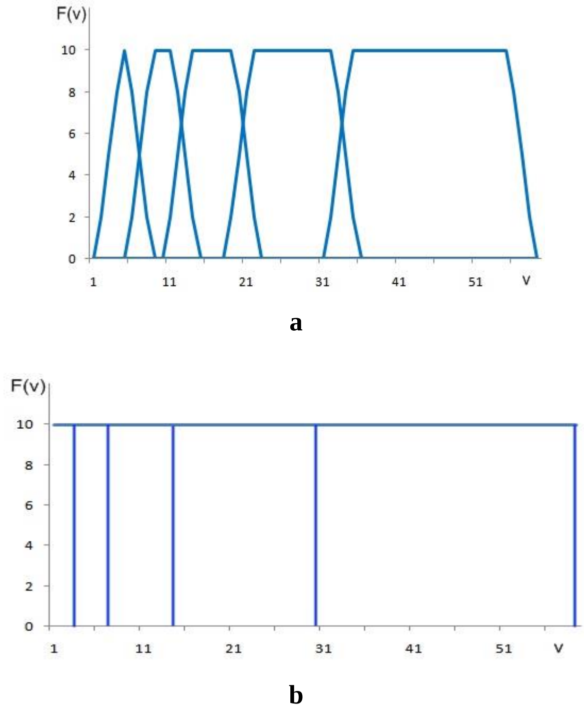


Figure 4.4.18. Wavelet spectra of different scale coefficients

If we call the wavelet spectra "windows", then these "windows" should be rectangular, having the same height but different widths. All other "windows" of a different shape form an uneven amplitude-frequency response in the set (sum). Figure 2.2.2.3 (Chapter 2) shows the frequency response of a set of *MHAT* wavelets, i.e. the wavelets obtained on the basis of the second derivative of the Gaussian function. These wavelets cannot be called orthogonal due to the fact that the set (sum) of wavelets is a block of filters with an uneven amplitude-frequency response. This is due to the fact that it is impossible to obtain a uniform amplitude-frequency response using the spectra shown in Fig. 1.4.3 and 1.4.4 (Chapter 1). Only the spectra similar to those shown in Figure 4.4.7 allow us to obtain a uniform amplitude-frequency response for the filter block. From this point of view, even orthogonal wavelets for discrete WT are not truly orthogonal, regardless of the fact that they are called orthogonal in the scientific literature. Their spectra have overlaps because they do not have ideal frequency characteristics (Fig. 4.4.18a). They are also not symmetric, or antisymmetric functions. Accordingly, in the Proni transformation, the functions described by equation (1.10) do not give ideal amplitude-frequency characteristics. Their spectra have overlaps and therefore the recovery error is greater than when using orthogonal wavelets constructed in the frequency domain. By constructing a wavelet or any function in the time domain, it is impossible to obtain an ideal amplitude-frequency response for such a function.

In the course of signal studies, it was found that the forward and inverse wavelet transform in the frequency domain using orthogonal symmetric and antisymmetric wavelets can increase the calculation speed many times. Moreover, the speed of calculating the inverse WT increases even in comparison with using the FFT multiple, depending on which wavelet is used. Here are some ways to increase the speed of calculating the continuous wavelet transform.

The first method is to use the parity or odd property of the wavelets, i.e. symmetry and antisymmetry. This method is presented in the forward and inverse WT algorithms. Most continuous wavelets are either even or odd functions. For even-numbered wavelets, the series consists of one cosine, and for odd-numbered ones of one sine. For even wavelets

$$b_2(n) = 0$$
$$c_1(n) = a_1(n)a_2(n)$$
$$c_2(n) = b_1(n)a_2(n)$$

For odd wavelets

$$a_2(n) = 0$$
$$c_1(n) = b_1(n)b_2(n)$$
$$c_2(n) = -a_1(n)b_2(n)$$

The second method is the construction of wavelets in the frequency domain, i.e. there is no need to calculate the wavelets and the coefficients of the trigonometric series $a_2(n)$, $b_2(n)$ in the second step.

The third method is to calculate the M inverse Fourier transforms of the complex conjugate spectrum by the formula

$$W(a,b) = \sum_{k=0}^{N-1} \left(c_1(k) + ic_2(k)\right) \exp\left(i\frac{2\pi nk}{N}\right)$$

only at certain intervals, i.e. for small scale coefficients at a smaller interval and for large scale coefficients at a larger interval. In the FFT algorithm, this is possible. Thus, the number of coefficients of the wavelet spectrum is equal to the number of samples of the original signal, i.e. the non-redundant WT. These wavelet coefficients can be used to compress the signal by first equating to zero values that are less than a certain threshold. By linear or nonlinear interpolation of these wavelet coefficients, it is easy to reconstruct the signal with the inverse WT. By nonlinear interpolation, the signal is reconstructed more accurately. Nonlinear interpolation uses the same wavelets as decomposition. The WT calculation time is reduced by a factor of 2.5 for a sample of 32768 samples. When processing images, the sample becomes even larger and, accordingly, the calculation time will decrease by an even greater value. Since the processing of color images requires obtaining a wavelet spectrum for three colors (RGB) along the horizontal and vertical axes, significant time savings are achieved.

The fourth method is to use such a property for WT color images as color independence for small scale coefficients, i.e. the wavelet coefficients of red, blue and green have the same value. It is enough to calculate the wavelet spectrum for a single color. This property can be used for compression, since for small wavelet coefficients, triple compression is obtained without encoding. The fact that for small scale coefficients all three colors have the same wavelet coefficients, the human eye uses, in all probability, very well. There are 130 million rods that are responsible for dark vision in the human eye, and 7 million cones that are responsible for color vision. The cones are further apart than the rods, as the colors only appear at large scales. Since the intensity for small scale coefficients is less than for large ones, the radiation receivers should be more sensitive for small scales, i.e. the sensitivity of the rods should be higher than that of the cones. In fact, the rods feel a fainter light than the cones.

The fifth method is to construct the wavelets in such a way that it is not necessary to calculate the normalizing coefficient during reconstruction.

The sixth method is the application of the fourth method in the reconstruction, so that the wavelet coefficients of one color can be used for all colors in 1, 2, 5, 6, 7, 8 steps in the inverse WT algorithm and in multiscale analysis.

The seventh method is the construction of wavelets and preprocessing of the wavelet coefficients in such a way that it is possible to reconstruct a signal with the order of computational operations $O(N)$. This method allows one to reduce the signal reconstruction time by 1000 times compared to the algorithm using the FFT for sampling the signal of 32768 samples. Compared to direct numerical integration, more than 1,000,000 times.

The construction of wavelets in the time domain does not allow to simultaneously improve the amplitude-frequency characteristics and reduce the conversion time. As a rule, improving the frequency characteristics of wavelets leads to an increase in the conversion time, since more equations need to be solved. The use of symmetric and antisymmetric orthogonal wavelets constructed in the frequency domain allows one to reduce the conversion time, and it also allows one to improve the phase and amplitude-frequency characteristics.

In addition to designing digital filters, using algorithms for the forward and inverse fast continuous wavelet transform, we can calculate derivatives of functions, integrals, and perform interpolation, extrapolation of functions.

4.5 Conclusions

1. Algorithms for compressing one- and two-dimensional signals in the frequency domain have been developed.

2. An algorithm for numerical calculation of the fractal dimension is developed.

3. An algorithm for calculating the average size of macro - and micro-objects in the image obtained from a satellite or a microscope has been developed.

4. The results of multiple-scale image analysis in the frequency domain are compared with the results presented in the MatLab computer mathematics system.

5. Orthogonal symmetric and antisymmetric wavelets with a scale factor of less than two are constructed. The WT calculation time is reduced many times.

MAIN RESULTS

The following main results are obtained:

An algorithm for numerical calculation of direct continuous fast WT with an arbitrary choice of scaling coefficients is developed, which allows real-time signal conversion.

An algorithm for numerical calculation of the inverse fast WT is developed, which allows us to reconstruct a signal with a sample of 2 to the power of m in $m + 1$ steps. For signals with a sample of more than 1024 samples, the Pearson correlation coefficient r is 0.99.

A comparison of the results of multiple-scale image analysis in the frequency domain with the result presented in the MatLab computer mathematics system shows that the developed algorithms allow for better signal analysis.

Algorithms for numerical calculation of the forward and inverse fast WT with a scaling factor of less than two are developed, which allows the signal to be decomposed into a larger number of levels than for a scaling factor of two.

In the time domain, it is impossible to construct wavelets with ideal amplitude-frequency characteristics due to the Gibbs phenomenon, and in the frequency domain, they can be presented in the work. Algorithms for designing digital filters of low, high frequencies, notch, and blocking filters have been developed.

Algorithms for calculating the average size, standard deviation, and anisotropy of macro- and micro-objects in the image obtained from a satellite or a microscope are developed on the basis of wavelets constructed in the frequency domain.

A set of computer programs has been developed that implements the proposed models and algorithms in real time.

REFERENCES

[1] Astaf'eva N. M. (1996). *Vejvlet-analiz: osnovy teorii i principy primeneniya* [Wavelet analysis: fundamentals of theory and application principles], In: Advances in Physical Sciences, vol. 166, no. 11, pp. 1145–1170.

[2] Ashihmin V. N., Gitman M. B. (2004). *Vvedenie v matematicheskoe modelirovanie* [Introduction to mathematical modeling], Moscow, Logos Publ., 440 p.

[3] Bermant A. F., Aramanovich I. G. (1967). *Kratkij kurs matematicheskogo analiza* [Short course of mathematical analysis], Moscow: Nauka Publ.

[4] Vasil'ev V. N., Gurov I. P. (2004). *Komp'yuternaya obrabotka signalov v prilozhenii k interferometricheskim signalam* [Computer signal processing in the application to interferometric signals], St. Petersburg: BHV-Petersburg Publ., 237 p.

[5] Vityazev V. V. (2001). *Vejvlet-analiz vremennyh ryadov: ucheb. posobie* [Wavelet analysis of time series: tutorial], St. Petersburg: Publishing House of St. Petersburg University, 58 p.

[6] Voevodin V. V., Kuznecov Yu. A. (1984). *Matricy i vychisleniya* [Matrixes and computations]. Moscow: Nauka Publ., 320 p.

[7] Gajdyshev I. P. (2004). *Reshenie nauchnyh inzhenernyh zadach sredstvami Excel, VBA u C/C++* [Solving scientific engineering problems by means of Excel, VBA and C / C++], St. Petersburg: BHV-Petersburg Publ., 512 p.

[8] Galyagin D. K. (2000). *Vejvlet-analiz vremennoj struktury kosmicheskih magnitnyh polej.* Cand. Diss. [Wavelet analysis of the time structure of cosmic magnetic fields. Cand. Diss.]. Perm.

[9] Dremin I. L., Ivanov O. V., Nechitajlo V. A. (1996). *Vejvlety i ih ispol'zovanie* [Wavelets and their use], In: Advances in Physical Sciences, vol. 171, no. 5, pp. 465–501.

[10] Kazakov S. M., Semenov V. I., Sorokin G. M., Shurbin A. K. (2015). *Opredelenie srednih razmerov nazemnyh ob''ektov na osnove vejvlet-*

analiza kosmicheskih snimkov mestnosti [Determination of the average size of ground objects based on the wavelet analysis of satellite images of the terrain], *Aktual'nye zadachi matematicheskogo modelirovaniya i informacionnyh tekhnologij: materialy Mezhdunar. nauch.-prakt. konf.* [Actual problems of mathematical modeling and information technologies: materials of the International Scientific and Practical Conference]. Sochi, May 15-24, 2015, 162 p.

[11] Kiedzi A. (1993). *Prikladnye nechyotkie sistemy* [Applied fuzzy systems], Moscow: Mir Publ.

[12] Kravchenko V. F., Masyuk V. M. (2002). *Novyj klass fraktal'nyh funkcij v zadachah analiza i sinteza antenn.* [A new class of fractal functions in problems of antenna analysis and synthesis]. Moscow: IPRZHR Publ., 72 p.

[13] Krivosheev V. I. (2006). *Sovremennye metody cifrovoj obrabotki signalov (cifrovoj spektral'nyj analiz)* [Modern methods of digital signal processing (digital spectral analysis)], Publishing House of Lobachevsky State University of Nizhny Novgorod, 117 p.

[14] Kurat-Gilbert. (1951). *Metody matematicheskoj fiziki* [Methods of mathematical physics], Vol. 1, Moscow: Gostekhizdat Publ.

[15] Kuryachij M. I. (2009). *Cifrovaya obrabotka signalov: ucheb.posobie dlya vuzov* [Digital signal processing: textbook for universities], Publishing House of Tomsk State Un-t of Systems of Control and Radioelectronics, 190 p.

[16] Kuhapenko B. G. (2008). *Tekhnologiya spektpal'nogo analiza na osnove bystpogo ppeobpazovaniya Pponi* [The technology of spectral analysis based on the fast transformation of Proni], In: *Infopmacionnye tekhnologii* [Information Technologies], No. 4, pp. 38-42.

[17] Leonovich A. A. (2013). *Sovremennye tekhnologii raspoznavaniya rechi* [Modern speech recognition technologies], online: https://www.km.ru/referats/334470-sovremennye-tekhnologii-raspoznavaniya-rechi

[18] Marpl-jr. S. L. (1990). *Cifrovoj spektral'nyj analiz i ego primeneniya.* [*Digital spectral analysis and its applications*], Moscow: Mir Publ., 584 p.

[19] U. Li. M. (1983). *Metody avtomaticheskogo raspoznavaniya rechi: v 2 kn.; per. s angl. pod red. U. Li. M.* [Methods of automatic speech recognition: in 2 books; translated from English. ed. by U. Li. M], Mir Publ., 1983, Book 1, 328 p.

[20] Mitrofanov G., Prijmenko V. (2011). *Osnovy i prilozheniya metoda Proni-fil'tracii* [Fundamentals and applications of the Proni filtration method], In: *Tekhnologii sejsmorazvedki* [Technologies of seismic exploration], No. 3, pp. 93-108.

[21] Miherkina N. V., Semenov V. I. (2017). *Issledovanie zavisimosti srednego razmera zyoren keramiki ot temperatury* [Investigation of the dependence of the average size of ceramic grains on temperature]. *Sovremennoe sostoyanie nauki i tekhniki: Mezhdunar. mul'tidisciplinarnaya nauch.-prakt. konf. Mezhdunar. mul'tidisciplinarnyj molodezhnyj forum "Molodezh': nauka i tekhnika".* [Modern state of science and Technology: International multidisciplinary scientific and practical conference. International. multidisciplinary youth forum "Youth: Science and Technology"], 22-31 May 2017, Sochi, 96 p.

[22] Myasnikova N. V., Dudkin V. A. (2009). *Ispol'zovanie metoda Proni dlya analiza sejsmicheskih signalov idushchego cheloveka* [The use of the Proni method for the analysis of seismic signals of a walking person], In: *Tekhnicheskie nauki. Elektronika, izmeritel'naya i radiotekhnika* [Technical Sciences. Electronics, measuring and radio engineering], Vol. 4, No. 12, pp. 111-117.

[23] Novikov I. YA., Protasov V. YU., Skopina M. A. (2005). *Teoriya vspleskov* [Theory of bursts], Moscow: Fizmatlit Publ., 616 p.

[24] Novikov L. V. (1999). *Osnovy vejvlet-analiza signalov: ucheb.posobie IAnP RAN* [Fundamentals of wavelet analysis of signals], textbook manual, Publishing House of the Institute for Analytical Instrumentation, 152 p.

[25] Nussbaumer G. (1985). *Bystroe preobrazovanie Fur'e i algoritmy vychisleniya svertok* [Fast Fourier transform and algorithms for calculating convolutions], Moscow: Radio i svyaz' Publ.

[26] Popov R. A. (2001). *Sozdanie sistemy raspoznavaniya nachal'nogo urovnya* [Creating an entry-level recognition system].

[27] Privalov I. I. (1930). *Ryady Fur'e.* [Fourier series]. Moscow: Gosizdat Publ.

[28] Rabiner L. R. (1989). *Skrytye markovskie modeli i ih primenenie v izbrannyh prilozheniyah pri raspoznavanii rechi* [Hidden Markov models and their application in selected applications in speech recognition], In: Proceeding of the IEEE, Vol. 77, No. 2.

[29] Saltykov S. A. (1976). *Stereometricheskaya metallografiya* [Stereometric metallography], Moscow: Metallurgiya Publ., 270 p.

[30] Napalkov V. V. (1982). *Convolution equations in multidimensional spaces*. Moscow: Nauka Publ., 240 p.

[31] Semenov V. I. (2012). *Razrabotka i modelirovanie algoritmov bystrogo nepreryvnogo vejvlet-preobrazovaniya s primeneniem k obrabotke rechevyh signalov. Cand. Diss.* [Development and modeling of algorithms for fast continuous wavelet transformation with application to speech signal processing. Cand. Diss.], 168 p.

[32] Semenov V. I. (2010). *Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2010616103* [Certificate of state registration of the computer program No. 2010616103]. *Nepreryvnoe sverhbystroe vejvlet-preobrazovanie* [Continuous ultrafast wavelet transform], September 16, 2010.

[33] Semenov V. I. (2011). *Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2011610159.* [Certificate of state registration of the computer program No. 2011610159]. *Nepreryvnoe bystroe ne izbytochnoe vejvlet-preobrazovanie* [Continuous fast non-redundant wavelet transform], January 11, 2011.

[34] Semenov V. I. (2011). *Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2011615827.* [Certificate of state registration of the computer program No. 2011615827]. *Ortogonal'noe bystroe*

vejvlet-preobrazovanie [Orthogonal fast wavelet transform], July 26, 2011.

[35] Semenov V. I. (2010). *Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2010616103.* [Certificate of state registration of the computer program No. 2010616103]. *Nepreryvnoe sverhbystroe vejvlet-preobrazovanie*. [Continuous ultrafast wavelet transform], September 16, 2010.

[36] Semenov V. I. (2007). *Svidetel'stvo ob oficial'noj registracii programmy dlya EVM № 2007615024.* [Certificate of official registration of the computer program No. 2007615024]. *Nepreryvnoe bystroe vejvlet-preobrazovanie* . [Continuous fast wavelet transform], December 4, 2007.

[37] Semenov V. I., Zheltov P. V. (2008). *Vejvlet-analiz akusticheskogo signala* [Wavelet analysis of the acoustic signal], In: Vestnik of Kazan State Technological University, Issue 4, Vol. 4.

[38] Semenov V. I., Zheltov P. V. (2008). *Vejvletnye funkcii* [Wavelet functions]. In: *Komp'yuternye tekhnologii i modelirovanie: sb. nauch. tr*. [Proc. of "Computer technologies and modeling"]. Cheboksary, Issue 3, pp. 60-65.

[39] Semenov V. I., Zheltov P. V. (2008). *Vejvlet-preobrazovanie akusticheskogo signala.* [Wavelet transform of an acoustic signal] / Publishing House of Kazan State Technological University, 102 p.

[40] Semenov V. I., Zheltov P. V. (2009). *Vejvlet-preobrazovanie rechevyh signalov* [Wavelet transformation of speech signals]. In: *Matematicheskie modeli i ih prilozheniya: sb. nauch. tr.* [Proc. of "Mathematical models and their applications"]. Cheboksary, Issue 11, pp. 185-191.

[41] Semenov V. I., Zheltov P. V. (2009). *Vejvlety i fraktaly* [Wavelets and fractals]. In: *Dinamika nelinejnyh diskretnyh elektrotekhnicheskih i elektronnyh sistem: materialy VIII Vseros. nauch.-tekhn. konf.* [Proc. of the VIII All-Russian Scientific-Technical Conf. "Dynamics of nonlinear discrete electrotechnical and electronic systems"], Cheboksary, pp. 131-133.

[42] Semenov V. I., Zheltov P. V. (2011). *Metodika opredeleniya granic mezhdu glasnymi i soglasnymi zvukami rechi s primeneniem bystrogo nepreryvnogo vejvlet-preobrazovaniya* [A technique for determining the boundaries between vowels and consonants of speech using a fast continuous wavelet transform]. In: *Dinamika nauchnyh issledovanij 2011: materialy VII mezhdunar. nauch.-prakt. konf.* [Proc. of the VII International Scientific and Practical Conference "Dynamics of scientific Research 2011"], Przemysl, pp. 12-17.

[43] Semenov V. I., Zheltov P. V. (2010). *Sposob raspoznavaniya klyuchevyh slov v slitnoj rechi*. [A method for recognizing keywords in merged speech]. Patent RF, no. 2403628, IPC G10L 15/10.

[44] Semenov V. I., Zheltov P. V. (2008). *Primenenie vejvlet-preobrazovaniya dlya rekonstrukcii akusticheskogo signala* [Application of the wavelet transform for the reconstruction of an acoustic signal]. In: *Komp'yuternye tekhnologii i modelirovanie: sb. nauch. tr.* [Proc. of "Computer technologies and modeling"], Issue 4, pp. 67-70.

[45] Semenov V. I., Zheltov P. V. (2008). *Primenenie vejvlet-preobrazovaniya k model'nym signalam* [Application of the wavelet transform to model signals]. In: *Komp'yuternye tekhnologii i modelirovanie: sb. nauch. tr.* [Proc. of "Computer technologies and modeling"], Issue 3. pp. 65-70.

[46] Semenov V. I., Zheltov P. V. (2008). *Raspoznavanie rechi na osnove vejvlet-preobrazovaniya* [Speech recognition based on the wavelet transform], 16 p., Dep. of All-Russian Scientific and Technical Information Institute of Russian Academy of Sciences 29.02.08, No. 174 - B2008.

[47] Semenov V. I., Zheltov P. V. (2009). *Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2009616896* [Certificate of state registration of the computer program No. 2009616896]. *Nepreryvnoe bystroe* $m + 1$ *shagovoe vejvlet-preobrazovanie* [Continuous fast $m + 1$ step wavelet transform], December 11, 2009.

[48] Semenov V. I., Zheltov P. V. (2010). *Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2010610456* [Certificate of state

registration of the computer program No. 2010610456]. *Nepreryvnoe bystroe dvuhshagovoe vejvlet-preobrazovanie* [Continuous fast two-step wavelet transform], January 11, 2010.

[49] Semenov V. I., Zheltov P. V., Pavlova N. V., Grigor'ev V. G. (2014). *Algoritm dekompozicionnogo vejvlet-preobrazovaniya izobrazhenij* [Algorithm of image decomposition wavelet transformation]. In: *Sovremennye problemy nauki i obrazovaniya* [Modern problems of science and education], vol. 6.

[50] Semenov V. I., Kazakov S. M., Sorokin G. M., Shurbin A. K. (2015). *Primenenie vejvlet-preobrazovaniya dlya vychisleniya srednih razmerov ob"ektov na izobrazhenii* [Application of the wavelet transform for calculating the average size of objects in the image]. In: *Sovremennye tendencii razvitiya nauki i tekhniki: materialy III nauch.-tekhn. konf*. [Proc. of the III scientific-Technical Conf. "Modern trends in the development of science and technology"], Belgorod, pp. 10-12.

[51] Semenov V. I., Kazakov S. M., Shurbin A. K. (2017). *Primenenie nepreryvnogo bystrogo vejvlet-preobrazovaniya dlya vychisleniya mery anizotropii* [Application of the continuous fast wavelet transform for calculating the anisotropy measure]. In: *Dinamika nelinejnyh diskretnyh elektrotekhnicheskih i elektronnyh sistem: materialy XI Vseros. nauch.-prakt. konf*. [Proc. of the XI All-Russian Scientific-Technical Conf. "Dynamics of nonlinear discrete electrotechnical and electronic systems"], Cheboksary, pp. 93-96.

[52] Semenov V. I., Miheev K. G., Shurbin A. K., Miheev G. M. (2014). *Fil'traciya izobrazhenij, poluchennyh s pomoshch'yu opticheskogo mikroskopa, s primeneniem kratnomasshtabnogo analiza* [Filtering of images obtained with an optical microscope using multiple-scale analysis]. In: *Himicheskaya fizika i mezoskopiya* [Chemical physics and mesoscopy], Vol. 16, no. 3, pp. 399-404.

[53] Semenov V. I., Sorokin G. M., Shurbin A. K., Petrov N. I. (2017). *Opredelenie srednekvadratichnogo otkloneniya razmera ob"ektov na izobrazhenii* [Determination of the standard deviation of the size of objects in the image]. In: *Dinamika nelinejnyh diskretnyh elektrotekhnicheskih i elektronnyh sistem: materialy XI Vseros. nauch.-*

prakt. konf. [Proc. of the XI All-Russian Scientific-Technical Conf. "Dynamics of nonlinear discrete electrotechnical and electronic systems"], Cheboksary, pp. 96-99.

[54] Semenov V. I., Hristoforov O. V., Chuchkalov S. I., Shurbin A. K. (2017). *Vychislenie srednekvadratichnogo otkloneniya razmera ob"ektov na osnove vejvlet-analiza izobrazhenij* [Calculating the standard deviation of the object size based on image wavelet analysis]. In: *Uspekhi sovremennoj nauki* [Achievements of modern science], vol. 4. no. 2. pp. 109-112.

[55] Semenov V. I., Shurbin A. K., Hristoforov O. V. (2016). *Poisk podobiya v baze dannyh izobrazhenij s ispol'zovaniem vejvlet-preobrazovaniya* [Search for similarity in the database of images using the wavelet transform]. In: *Informacionnye tekhnologii v elektrotekhnike i elektroenergetike: materialy H Vseros. nauch.-tekhn. konf* [Proc. of X All-Russian Scientific-technical Conf. "Information Technologies in electrical engineering and electric power engineering"], Cheboksary, pp. 147-149.

[56] Smirnov V. I. (1965). *Kurs vysshej matematiki* [Course of higher mathematics]. Vol. 1, 2. Moscow: Nauka Publ.

[57] Stolnic E., De Rouz E., Salezin D. (2002). *Vejvlety v komp'yuternoj grafike.* [Wavelets in computer graphics]. Moscow-Izhevsk, 271 p.

[58] Umnyashkin S. V. (2008). *Teoreticheskie osnovy cifrovoj obrabotki i predstavleniya signalov* [Theoretical foundations of digital signal processing and representation]. M.: FORUM: Infra-M Publ., 304 p.

[59] Uelstid S. (2003). *Fraktaly i vejvlety dlya szhatiya izobrazhenij* [Fractals and wavelets for image compression]. Moscow: Triumph Publ., 319 p.

[60] Feder E. (1991). *Fraktaly.* [Fractals]. Moscow: Mir Publ., 254 p.

[61] Fihtengol'c G. M. (1969). *Kurs differencial'nogo i integral'nogo ischisleniya.* [Course of differential and integral calculus]. Vol. 1-3. M.: Nauka Publ.

[62] Chekmarev A. (1997). *Rechevye tekhnologii – problemy i perspektivy* [Speech technologies-problems and prospects], In: *Komp'yuterra* [Computer], no. 49. pp. 26-43.

[63] Shvarc E. (2001). *Avtorskie prava na puti Voice XML* [Copyright on the Voice XML path], In: Computerworld, No. 36.

[64] Shirman Ya. D., Manzhos V. N. (1981). *Teoriya i tekhnika obrabotki radiolokacionnoj informacii na fone pomekh* [Theory and technique of processing radar information against the background of interference]. Moscow: Radio and Communications Publ.

[65] Shtark G. G. (2007). *Primenenie vejvletov dlya COS* [Application of wavelets for DSP]. Moscow: Technosphere Publ., 192 p.

[66] Yakovlev A. N. (2003). *Osnovy vejvlet-preobrazovaniya* [Fundamentals of the wavelet transform]. Moscow: Science Press Publ., 79 p.

[67] Daubechies I. (1992). *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM Publications, Philadelphia, 377 p., ISBN: 9780898712742.

[68] Hirataand K., Kato T. (1992). *Query by visual example-content based image retrieval*. Pirotte A., Delobel C. and Gottlob G., ed. In: Advances in Database Technology (EDBT'92), Springer-Verlag, Berlin, pp. 56–71.

[69] Kato T., Kurita T., Otsu N., Hirata K. A (1992). *Sketch retrieval method for full color image database – query by visual example*. In: Proceedings of the 11th IAPR International Conference on Pattern Recognition, IEEE Computer Soceity Press, Los Alamitos, CA, pp. 530–533.

[70] Kitamoto A., Zhou C., Takagi M. (1993). *Similarity retrieval of NOAA satellite imagery by graph matching*. In: Storage and Retrieval for Image and Video Databases, vol. 1908 of Proceedings of the SPIE, SPIE, Bellingham, WA, pp. 60–73.

[71] Niblck W., Barber R., Eguitz W., Glasman E., Petkovic D., Yanker P., Faloutsos C., Taubin G. (1993). *The QBIC project: Querying images by content using color, texture, and shape*. In: Storage and Retrieval for

Image and Video Databases, vol. 1908 of Proceedings of the SPIE, SPIE, Bellingham, WA, pp. 673-681.

[72] Polikar R. (1999). *Vvedenie v vejvlet-preobrazovanie* [Introduction to the Wavelet transform], online: http://www.autex.spb.ru.

[73] Prony R. (1796). *Essai experimental et analytique*. In: Journal de l'Ecole Polytechnique. vol. 1, no. 22, pp. 24–76.

[74] Semenov V. I., Khristoforov O. V., Chuchkalov S. I. (2017). *Calculating the standard deviation of the size of objects in an image*. In: Journal of Advanced Research in Technical Science. North Charleston, USA, pp. 62–64.

[75] https://www.adriver.ru/

[76] http://intsys.msu.ru

[77] http://www.cmu.edu

[78] http://www.comptek.ru

[79] http://www.dfki.de/verbmobil/

[80] http://www.dragonsys.com

[81] https://www.ibm.com/cloud/watson-text-to-speech

[82] http://www.intel.com

[83] http://www.ipu.ru

[84] http://www.isa.ru

[85] http://www.istrasoft.ru

[86] https://azure.microsoft.com/en-us/services/cognitive-services/

[87] http://www.mstechnology.ru

[88] http://www.opera.com

[89] https://www.sensory.com/

[90] http://www.speechpro.ru

[91] http://www.spiritdsp.com

[92] http://www.stel.ru/speech/frame.html

[105] http://www.w3.org/TR/voicexml20/

Authors

Viliam Ďuriš (1981) – Dr., PhD.

Viliam Ďuriš studied mathematics and computer science and has been an assistant professor at the Department of Mathematics of the Faculty of Natural Sciences of Constantine the Philosopher University in Nitra, Slovakia, since 2009, focusing mainly on object programming, applied mathematics and numerical mathematics He is an author of many major projects and training systems and has also been involved in major national and international projects or projects of the European Social Fund. He has been awarded several times during his work at the university for publishing and scientific work, and for 7 years was a member of the program and organizing committee of international conferences „Mathematical Conference in Nitra“ and „Statistical Days in Nitra“ at the Faculty of Natural Sciences, Constantine the Philosopher University in Nitra. He is also one of the founding members of the GeoGebra Institute in Nitra, which is part of a network of local GeoGebra institutes accredited by the International GeoGebra Institute International GeoGebra Institute – IGI and the author of the local GeoGebra portal. As expert in information technology and professional database programmer runs his projects worldwide.

Department of Mathematics
Faculty of Natural Sciences
Constantine the Philosopher University in Nitra
Tr. A. Hlinku 1
949 74 Nitra, Slovakia

Vladimir I. Semenov (1960) - Assoc. Prof. Dr.-Ing., PhD.

Vladimir I. Semenov received the higher education in physics from Chuvash State University, Cheboksary, Russia, in 1985 and the Ph.D. degree in optics from Ulyanovsk State Technical University, Ulyanovsk, Russia, in 2012. Since 1999, he has been with the Department of General Physics, Chuvash state University, where he is currently an Associate Professor of the Department. His current research interests include digital signal and image processing and continuous fast wavelet transform.

Department of General Physics
I. N. Ulyanov Chuvash State University
428015 Cheboksary, Russia

Sergey G. Chumarov (1977) - Assoc. Prof. Dr.-Ing., PhD.

Sergey G. Chumarov received the higher education in radio engineering from Mari State Technical University, Yoshkar-Ola, Russia, in 1999 and the Ph.D. degree in radio engineering from Mari State Technical University in 2002. Since 1999, he has been with the Department of Radio Engineering and the Department of Telecommunications, Chuvash State University, where he is currently a Head of Department of Radio Engineering. His current research interests include digital processing of radio signals, wavelet transform, engineering education, computer-aided design systems, software-defined radio.

Department of Radio Engineering
I. N. Ulyanov Chuvash State University,
428015 Cheboksary, Russia

Authors:	Viliam Ďuriš (3 AH) Vladimir I. Semenov (3 AH) Sergey G. Chumarov (3 AH)
Title:	APPLICATION OF CONTINUOUS FAST WAVELET TRANSFORM FOR SIGNAL PROCESSING
Publisher:	SCIEMCEE PUBLISHING, London
Year:	2021
Edition:	first
Number of pages:	181 (9 AH)
Number of prints:	300
Type of publication:	Scientific monograph

Reviewers:

Sergey M. Kazakov, prof., DrSc.
Department of General Physics, I.N. Ulyanov Chuvash State University

Aleksander I. Kitaev, doc., CSc.
Department of General and Theoretical Physics, Yakovlev Chuvash State Pedagogical University

Science editor:

Grigory P. Okhotkin, prof., DrSc.
Faculty of Radioelectronics and Automation, I.N. Ulyanov Chuvash State University

ISBN: 978-1-9993071-9-6

ISBN 978-1-9993071-9-6

SCIEMCEE
PUBLISHING LONDON