

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Оренбургский государственный университет»

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

Н.А. Соловьев, Н.А. Тишина, Л.А. Юркевская

# **ЦИФРОВАЯ ОБРАБОТКА ИНФОРМАЦИИ В ЗАДАЧАХ И ПРИМЕРАХ**

Рекомендовано ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» в качестве учебного пособия для студентов, обучающихся по программам высшего образования по направлениям подготовки 09.04.01 Информатика и вычислительная техника, 09.04.04 Программная инженерия, 09.04.02 Информационные системы и технологии

Оренбург  
2016

УДК 004.4 : 378.245 (076.5)

ББК 32.973-018я7+74.48я7

С 60

Рецензент – доктор технических наук, профессор А.М. Пищухин

**Соловьев, Н.А.**

С 60 Цифровая обработка информации в задачах и примерах: учебное пособие / Н.А. Соловьев, Н.А.Тишина, Л.А.Юркевская; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2016. – 122 с.

**ISBN 978-5-7410-1614-5**

В учебном пособии содержатся материалы, необходимые студентам для изучения теории и формирования умений и навыков выполнения практических задач по дисциплине «Системы цифровой обработки сигналов». В разделах содержатся теоретические вопросы, связанные с реализацией практических задач, примеры, методические материалы для выполнения лабораторных работ: цель, задачи, порядок выполнения работы, представлены варианты заданий.

Учебное пособие предназначено для преподавателей и студентов, обучающихся по направлениям подготовки 09.04.01 Информатика и вычислительная техника, 09.04.04 Программная инженерия и 09.04.02 Информационные системы и технологии, изучающих дисциплину «Системы цифровой обработки сигналов».

С 1602120000

УДК 004.4: 378.245 (076.5)

ББК 32.973-018Я7+74.48Я7

ISBN 978-5-7410-1614-5

© Соловьев Н.А.,  
Тишина Н.А.,  
Юркевская Л.А., 2016  
© ОГУ, 2016

## Содержание

Введение .....	5
1 Генерация данных .....	7
1.1 Понятие цифровой обработки .....	7
1.2 Цель и задачи выполнения лабораторной работы.....	17
1.3 Порядок выполнения лабораторной работы .....	18
2 Сглаживание экспериментальных данных .....	21
2.1 Фильтрация на основе метода наименьших квадратов .....	21
2.1.1 Фильтры МНК 1-го порядка.....	21
2.1.2 Фильтры МНК 2-го порядка.....	23
2.2 Цель и задачи выполнения лабораторной работы.....	26
2.3 Порядок выполнения работы.....	26
3 Гармонический анализ.....	28
3.1 Дискретное преобразование Фурье .....	28
3.2 Цель и задачи выполнения лабораторной работы.....	32
3.3 Порядок выполнения работы.....	32
4 Вейвлет анализ .....	34
4.1 Быстрое вейвлет-преобразование .....	34
4.2 Цель и задачи выполнения лабораторной работы.....	38
4.3 Порядок выполнения работы.....	39
5 Сжатие данных .....	41
5.1 Сжатие изображений на основе дискретных преобразований.....	41
5.2 Цель и задачи выполнения лабораторной работы.....	43
5.3 Порядок выполнения работы.....	44
6 Сжатие звука.....	46
6.1 Основы сжатия звука.....	46
6.2 Перцепционное сжатие звука с использованием вейвлет-пакетов .....	48
6.3 Цель и задачи выполнения лабораторной работы.....	52
6.4 Порядок выполнения работы.....	53
7 Фильтрация изображений .....	54
7.1 Пространственная фильтрация изображений .....	54
7.2 Частотная фильтрация изображений .....	58
7.3 Цель и задачи выполнения работы .....	67
7.4 Порядок выполнения работы.....	68
8 Статистическая регуляризация данных с очисткой от шумов .....	69
8.1 Модели шума.....	69
8.2 Очистка изображений от шумов .....	73
8.3 Цель и задачи выполнения работы .....	75
8.4 Порядок выполнения работы.....	76
9 Аппроксимация сигналов и функций.....	77
9.1 Интерполяция.....	77
9.2 Аппроксимация эмпирических данных.....	79

9.3	Цель и задачи выполнения работы .....	82
9.4	Порядок выполнения работы.....	82
10	Обнаружение и распознавание объектов изображений.....	87
10.1	Постановка задачи распознавания. Основные определения и понятия.....	87
10.2	Математические основы обнаружения и распознавания сигналов.....	89
10.3	Цель и методика выполнения работы.....	94
10.4	Порядок выполнения работы .....	96
11	Спектральный анализ в задаче обнаружения вторжений.....	97
11.1	Методы выявления аномалий трафика корпоративной сети .....	97
11.2	Модель сетевого трафика .....	99
11.3	Прогнозирование текущего состояния трафика корпоративной сети .....	106
11.4	Методика идентификации уровня аномальности трафика корпоративной сети .....	110
11.5	Алгоритмы мониторинга информационных процессов .....	113
	Список использованных источников.....	120

## Введение

Цифровая обработка информации является важным направлением в области информационных технологий, являясь основой множества видов цифровых разработок и различных приложений, таких как медицинские системы обследования организма, цифровая мобильная связь, цифровые видеокамеры, телевидение и системы звукозаписи. Специалист в области разработки программного обеспечения должен иметь представление о современных методах цифровой обработки данных, т.е. ориентироваться в современных алгоритмах программирования дискретных преобразований информации.

Целью учебного пособия является введение студентов в теорию и практику цифровой обработки. Хотя теоретические основы и изложены в необходимом объеме, основной акцент делается на практические вопросы применения методов цифровой обработки, такие как цифровая обработка звука, изображений и обработка числовых рядов на примере сетевого трафика. В основе изложения материалов по решению задачи обнаружения вторжений на основе спектрального анализа лежат собственные исследовательские наработки. В учебном пособии приведены примеры, выполненные в программной системе Mathcad и в среде программирования MS Visual Studio.

Для более полного усвоения излагаемого материала в учебном пособии также представлены методические материалы к лабораторным работам, направленные на практическое закрепление теоретического материала, изучаемого в дисциплине «Системы цифровой обработки сигналов» (СЦОС). Они охватывают все основные разделы ее учебной программы, включающие методы и алгоритмы СЦОС.

Методические материалы к лабораторным работам содержат: формулировки цели и задач, необходимых для достижения цели; задания и порядок выполнения работы; контрольные вопросы.

Студенты выполняют программную реализацию и моделирование исследуемых алгоритмов обработки цифровых сигналов в инструментальных средах программирования и математическом пакете Mathcad и в среде программирования MS Visual Studio.

Лабораторные задания по каждой лабораторной работе подлежат выполнению группой в целом или по вариантам. Число выполняемых лабораторных заданий может варьироваться в зависимости от направления подготовки обучаемых и формы обучения. По результатам выполнения работ оформляется отчет. В отчете должны быть приведены:

- 1 Исходные данные к лабораторной работе, ее цели и решаемые задачи.
- 2 Расчетные результаты по пунктам лабораторной работы – числовые данные, таблицы, графики и др.
- 3 Результаты анализа полученных расчетных данных.
- 4 Выводы по лабораторной работе.

На титульном листе отчета указывается номер индивидуального варианта, в соответствии с которым выполнена работа.

# 1 Генерация данных

## 1.1 Понятие цифровой обработки

Цифровая обработка представляет собой процесс изменения цифровых данных в любой форме.

Цифровая обработка охватывает множество предметных областей, таких как обработка изображений и биомедицинских данных, обработка звука и речи, обработка сигналов с сонаров, радаров и сенсоров, спектральный анализ. Наиболее распространенные ситуации, в которых участвует цифровая обработка являются компьютерная графика и цифровая обработка звука.

Для характеристики *формы представления* данных, при которой данные рассматриваются как результат некоторых измерений объекта исследований в виде последовательности значений скалярных величин (аналоговых, числовых, графических и пр.) в зависимости от изменения каких-либо переменных значений (времени, энергии, температуры, пространственных координат, и пр.) применяется термин «*сигнал*».

Термин “сигнал” очень часто отождествляют с понятиями “*данные*” (data) и “*информация*” (information). Действительно, эти понятия взаимосвязаны и не существуют одно без другого, но относятся к разным категориям.

Понятие *информации* имеет много определений, от наиболее широкого (информация есть формализованное отражение реального мира) до практического (сведения и данные, являющиеся объектом хранения, передачи, преобразования, восприятия и управления). “*Данные*” (от латинского datum – факт), то это совокупность фактов, результатов наблюдений, измерения каких-либо физических свойств объектов, явлений или процессах материального мира, представленных в формализованном виде, количественном или качественном. Информация получается путем соответствующей обработки и интерпретации данных. Данные могут быть получены путем регистрации сигнала.

### 1.1.1 Представление изображений

Изображение в математическом представлении – двумерный сигнал, несущий огромное количество информации. Растровая графика оперирует с изображениями в виде растров. Под растром (raster) понимается представление изображения на плоскости путем разбиения (дискретизации) его на одинаковые элементы (пиксели, pixels)) и присвоение каждому элементу своего цветового и других атрибутов [4,6].

Стандартно пиксель описывается в виде функции:

$$f(i, j) = (A(i, j), C(i, j))$$

где  $A(i, j) \subset R^2$  - область пикселя,

$C(i, j) \in C$  - атрибут пикселя.

Используются два вида атрибутов:

$C(i, j) = I(i, j)$  - интенсивность (яркость) пикселя;

$C(i, j) = \{R(i, j), G(i, j), B(i, j)\}$  - атрибуты в цветовой модели RGB.

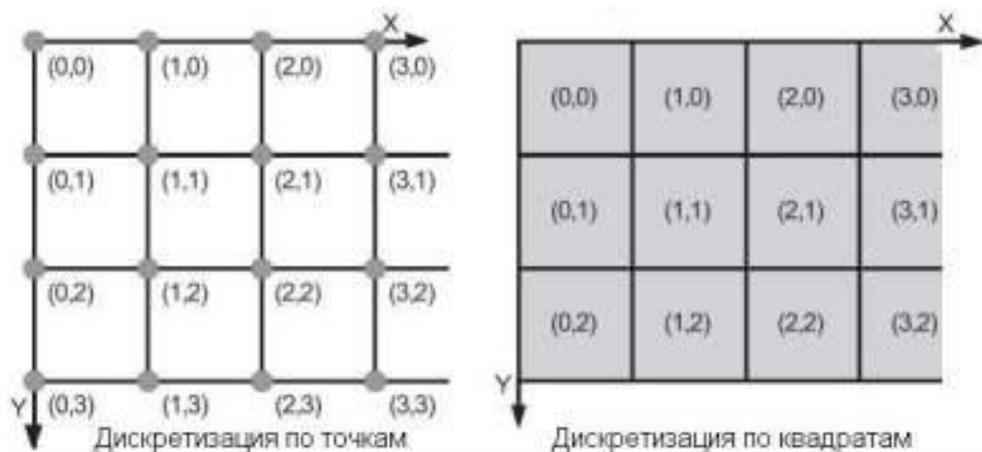


Рисунок 1.1 –Представление изображений

*Функции чтения и записи изображений в Mathcad [14].*

**READ\_IMAGE("file").** чтение изображения в оттенках серого, возвращает целочисленный массив

**READBMP("file")** чтение изображения в оттенках серого, возвращает целочисленный массив

**READRGB("file")** возвращает массив, представляющий изображение в основных цветах red-green-blue (Цветовая модель RGB).

**READ\_RED("file"), READ\_GREEN("file"), READ\_BLUE("file")**  
Возвращает массив содержащий только red, green, or blue компоненты из цветного изображения.

**WRITEBMP("file", [M])** Создает файл изображения в оттенках серого.

**WRITERGB("file", [M])** Создает цветной файл изображения (16-million-color bitmap image file).

**READ\_HLS("file")** возвращает массив, представляющий изображение в модели: яркость, насыщенность, оттенок

**READ\_HLS\_HUE("file"), READ\_HLS\_LIGHT("file"), READ\_HLS\_SAT("file")** Извлекает только яркость, насыщенность, оттенок из цветного изображения

Аргументы:

- **"file"** – имя, либо путь к файлу. Форматы рисунков BMP, GIF, JPG, PCX, TGA.

- **M** матрица. Независимо от файлового формата оригинального образа, образ в Mathcad – матрицы с величинами между 0 (черный) и 255 (белый), который может быть отображен оператором изображения. Каждый матричный элемент представляет интенсивность одного пикселя. Образы цвета RGB загружены как три

упакованные матрицы  $m \times n$ , следующим образом:

$$M = \begin{bmatrix} \text{Red} & \text{Green} & \text{Blue} \\ \text{Red} & \text{Green} & \text{Blue} \end{bmatrix}$$

Первые  $n$  столбцов представляют красный компонент  $m$  образа. Второй и третий комплекты  $n$  столбцов представляют зеленый и синий компоненты образа. HSV или образы HLS аналогично.

Загрузить рисунок для просмотра в системе Mathcad можно командой меню: Insert->Picture, затем в левом углу появившегося окна набрать в кавычках имя файла либо полный путь, если файл находится в другом каталоге.

Пример 1.1 – Чтение двух рисунков в оттенках серого (формате BMP) и в цветовой модели RGB, просмотр полученных матриц значений пикселей, уменьшение значений пикселей в два раза и запись в файл изображения.



"2.bmp"



"3.jpeg"

`x := READBMP("d:\2.bmp")`

`Y := READRGB("d:\3.jpeg")`

`WRITEBMP("c:\temp\1.jpeg") :=  $\frac{x}{2}$`

`WRITERGB("c:\temp\4.jpeg") :=  $\frac{Y}{2}$`

Рисунок 1.2 – Чтение, запись графического файла

	0	1	2	3	4
0	217	217	217	217	217
1	217	217	217	217	217
2	217	217	217	217	217
3	217	217	217	217	217
4	217	217	217	217	217
5	217	217	217	217	217
6	217	217	217	217	217
7	217	217	217	217	217
8	217	217	217	217	217
9	217	217	217	217	217
10	217	217	217	217	217
11	217	217	217	217	217
12	217	217	217	217	217
13	217	217	217	217	217
14	217	217	217	217	217
15	217	217	217	217	217

x =

`rows(x) = 222`    `cols(x) = 268`

	0	1	2	3	4
0	40	36	34	39	52
1	43	41	40	44	56
2	49	46	45	51	61
3	54	49	48	55	61
4	55	48	47	54	60
5	56	48	48	56	61
6	54	49	51	59	64
7	54	49	53	63	68
8	58	61	65	66	63
9	59	62	64	62	60
10	61	63	62	59	57
11	62	61	61	58	56
12	61	61	58	56	55
13	60	58	57	56	55
14	58	57	55	54	54
15	58	57	55	53	53

Y =

`rows(Y) = 480`    `cols(Y) = 1.92 × 103`

Рисунок 1.3 – Массив x (значения яркости пикселей) и массив Y (значения цветов red-green-blue пикселей)

Основные объекты и методы MS VS языка C# для обработки изображений:

```
Bitmap bmp = new Bitmap(@"getimage2.jpg"); // объект Bitmap для работы с
// изображениями
for (int x = 0; x < bmp.Width; x++)
    for (int y = 0; y < bmp.Height; y++) {
        Color c = bmp.GetPixel(x, y); // чтение цвета пикселя
        //с координатами x и y
        bmp.SetPixel(x, y, Color.FromArgb(255 - c.R, 255 - c.G, 255 - c.B));
// запись нового цвета пикселя с координатами x и y
    }
bmp.Save("out.bmp"); // сохранение рисунка в файле
pictureBox1.Image = bmp; // просмотр рисунка на форме в компоненте
// pictureBox1
```

### 1.1.2 Основы генерации компьютерного звука

Звук – физическое явление, представляющее собой распространение в виде упругих волн механических колебаний в твёрдой, жидкой или газообразной среде. В узком смысле под звуком имеют в виду эти колебания, рассматриваемые по отношению к тому, как они воспринимаются органами чувств животных и человека [16, 20, 24].

Основные понятия [16, 20, 24].

Выборка (sample) – измеренный фрагмент звуковой волны за очень малый промежуток времени. 44 100 таких фрагментов подряд образуют один аудиоканал с качеством CD.

Амплитуда (amplitude) – громкость, значение выборки. Максимальное и минимальное значения зависят от разрядности.

Разрядность (bit depth) – количество битов, используемых для представления выборки: 16, 32 и т. д. Максимальная амплитуда равна  $(2^{\text{разрядность}}) / 2 - 1$ .

Частота дискретизации (sample rate, sampling rate, bit rate) – число выборок в секунду. Например, при частоте 44100Hz количество периодов равно 44100 [27].

Современные приложения, как правило, генерируют звук «качества CD». Частота дискретизации при этом постоянна и равна 44 100 выборок в секунду. Теорема Найквиста утверждает, что частота дискретизации должна быть минимум в два раза больше самой высокой воспроизводимой частоты. Человек воспринимает механические колебания частотой 20 Гц - 20 КГц (дети - до 30 КГц) как звуковые, поэтому частота дискретизации, равная 44 100, вполне адекватна. Колебания с частотой менее 20 Гц называются инфразвуком, колебания с частотой более 20 КГц – ультразвуком. Для передачи разборчивой речи достаточен диапазон частот от 300 до 3000 Гц. Каждая выборка (sample) является знаковым 16-битным целым значением, размер которого предполагает отношение уровня сигнала к шуму в 96 децибел.

Чистый звуковой тон представляет собой звуковую волну, подчиняющуюся синусоидальному закону:

$$y = A * \sin(\omega t) = A * \sin(2\pi f t),$$

где  $A$  – максимальная амплитуда синусоиды;

$\omega$  – частота ( $\omega = 2\pi f$ );

$f$  – количество колебаний упругой среды в секунду ( $f = 1/T$ );

$T$  – период;

$t$  – время (параметрическая переменная).

Звук характеризуется частотой ( $f$ ), обычно измеряемой в герцах, т.е. количеством колебаний в секунду, и амплитудой ( $A$ ). Амплитуда звуковых колебаний определяет громкость звука.

Для монотонного звука (меандр) характерно постоянство амплитуды во времени.

Затухающие звуковые колебания характеризуются уменьшением амплитуды с течением времени.

Если несколько чистых синусоидальных колебаний смешать, то вид колебания изменится – колебания станут несинусоидальными.

Особый случай, когда смешиваются не любые синусоидальные колебания, а строго определенные, частота которых отличается в два раза (гармоники).

Основная гармоника имеет частоту  $f_1$ , и амплитуду  $a_1$ ; вторая гармоника - частоту  $f_2$  и амплитуду  $a_2$ ; третья гармоника соответственно  $f_3$  и  $a_3$ .

Причем  $f_1 < f_2 < f_3$ ,  $a_1 > a_2 > a_3$ . При бесконечном количестве таких гармоник образуется периодический сигнал, состоящий из прямоугольных импульсов.

На слух всякое отклонение от синусоиды приводит к изменению звучания.

Звук можно представить в виде бесконечного количества волн различной частоты и амплитуды. Волны, в свою очередь, могут иметь практически любую форму.

Из самых распространенных и чаще всего используемых форм можно назвать: синусоидальная (sine), квадратная (square), пилообразная (saw), треугольная (triangle), и шум (noise).



Рисунок 1.4 – Формы волн звука

*Алгоритмы генерации волн.*

1) Sine волна

```
float samplerate; // частота сэмпла
```

```
float wavefrequency; // частота волны
```

```
float wavevolume; // громкость волны
```

```
float period=samplerate/wavefrequency/2; //вычисляем период волны
```

```
float pi=3.14; //число pi
```

```
int n;
```

```

for(int a=0;a<samplelength;a++) //устанавливаем цикл на длину сэмпла{
    n=wavevolume*sin(a*pi/period); //вычисление sine-волны
    buffer[a]=n; //заносим вычисленное значение в буфер
}

```

Например, надо получить 16-килобайтный синусоидальный звук частотой 1000Hz и при этом качество сэмпла должно быть 44100Hz, тогда параметры будут иметь следующие значения: samplerate=44100, wavefrequency=1000, samplelength=16384.

Особенного пояснения требует параметр wavevolume. Издавна известно, что качество звука пропорционально зависит от его битности (8-bit, 16-bit, 24-bit и т.д.). Для 8-bit — значение от 0..255, для 16-ти – 0..65536, для 24-х – 0..16777216. Какой выбрать? Смотря какая у вас задача, но меньше 16-ти не рекомендуется, хотя бывают и исключения – когда надо сократить объем сэмпла взамен качеству.

## 2) пилообразная волна (saw)

```

float sr_2m=samplerate/wavefrequency; //Вычисляем период
int c=0; //Специальная переменная для проверки
    //на окончание периода
for(int a=0;a<samplelength;a++) //цикл на длину сэмпла{
    if(c>=sr_2m) c=0; //если период закончился, то начать следующий
    buffer[a]=wavevolume*(c/period)-wavevolume; //вычисление волны
    c++;
}

```

## 3) Треугольная (triangle)

```

float period=samplerate/wavefrequency/2; //вычисляем период волны
int c=period*2;
int c2=-1;
float sr_2m=period;
float sr=samplerate/wavefrequency/4;
for(int a=0;a<samplelength;a++){

```

```

if(c>sr_2m) c=sr_2m,c2=-1;
if(c<0) c=0,c2=1;
buffer[a]=wavevolume*(c/sr)-wavevolume;
c+=c2;
}

```

#### 4) Шум (noise)

```

srand(wavevolume);
s1=mv.samplerate/wor,psc=s1+1,wov>>=7;
if(tabcnt==0){
    for(ps=0;ps<mv.samplelength;ps++){
        if(psc>s1) {
            psc-=s1;
            n=256*((rand()%(wov+1))-wov/2);
        }
        buffer[ps]=n;
        psc++;
    }
}

```

Сложение двух волн разной частоты и амплитуды.

...

```

char buffer[16384];
float tone1=65,tone2=131;
float samplerate=44100/2; // Частота дискретизации
int a,b,amp1=128,amp2=64;
for(a=0;a<16384;a++){
    b=amp1*sin(a*pi/samplerate/tone1)+amp2*sin(a*pi/samplerate/tone2);
    if(b>128) b=128;
    if(b<-128) b=-128;
    buffer[a]=b;
}

```

}

...

С таким сэмплом уже можно писать мелодии. В принципе, это основа wave synthesis при помощи которой можно создать любой звук.

Советы:

1) Для получения более насыщенного звука складывайте волны с различными формами: sine+square, triangle+saw, или вот такой вот монстр: saw+square+triangle+saw.

2) Перкуссия (hat) лучше всего получается путем сложения noise+sine.

*Как представляется звук.*

Секунда аудиоданных представляется массивом некоего типа целочисленных данных с размером в 44 100 элементов. Однако, если нужно проигрывать аудиоданные со звуковой платы компьютера, эти данные должны быть дополнены целым букетом сведений о формате. Самый простой в обращении формат – WAV.

Более подробную информацию по этой тематике см. в статье «Часть 2: срываем завесу тайны с формата WAV» (EN). Как формировать WAV-файл в старом и двоичном стиле, см. в статье «Часть 3: синтез простого звука в формате WAV с помощью C#» (EN) [27].

Также можно использовать готовую библиотеку DirectSound. DirectSound дает множество удобных классов для любых форматов, абстрагируя их и позволяя просто закачивать поток данных в DirectSound-объект и воспроизводить их [27].

*Функции чтения и записи wav-файлов в системе Mathcad.*

**READWAV("file")** Считывает wav-файл и создает матрицу, содержащую амплитуды звукового сигнала. Каждый столбец представляет отдельный канал данных, каждая строка – временной образ

**WRITEWAV("file", s, b, [M])** Создает WAV- файл из матрицы M.

**GETWAVINFO("file")** Создает вектор с информацией о файле: количество каналов, норма отбора, количество битов за образец, и среднее количество байт в секунду.

Аргументы:

- file – имя, либо путь к файлу Microsoft WAV.
- s – скорость следования сэмплов, задаваемых матрицей.
- b – разрешение звука в битах.
- **M** – вещественная матрица, из которой каждый столбец

переписывается в отдельный канал, а каждая строка – во временной образ.

Пример чтения звукового файла в системе MathCad представлен на рисунке 1.5.

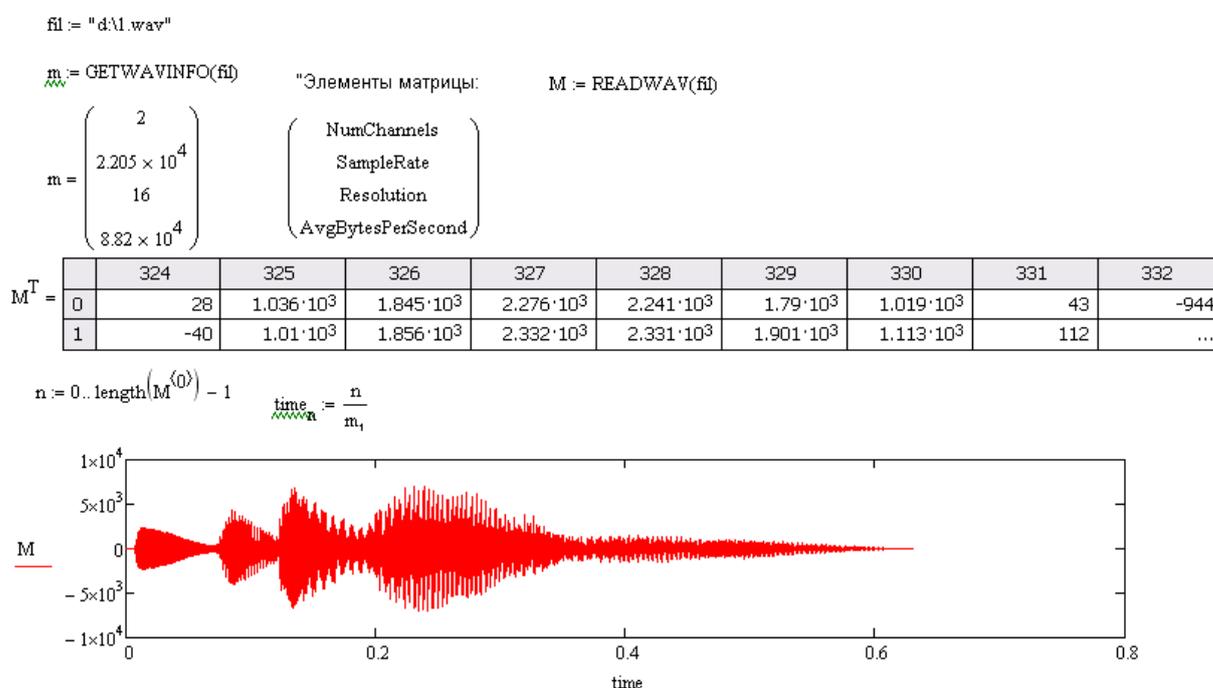


Рисунок 1.5 – Чтение звукового файла

Таким образом, цифровые изображения и звуки дают возможность простой и наглядной генерации данных.

## 1.2 Цель и задачи выполнения лабораторной работы

*Цель:* Формирование умений подготовки цифровых данных для анализа методами цифровой обработки данных в математическом пакете и среде программирования на примере изображения и звука.

*Задачи:*

- 1 Формирование цифрового массива данных и генерация шумов для анализа.
- 2 Формирование цифрового массива данных в математическом пакете Mathcad.
- 3 Предварительная обработка цифровых данных.

### **1.3 Порядок выполнения лабораторной работы**

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Изучить по справочному руководству средства работы с изображениями в системе Mathcad и в среде MS Visual Studio: типы данных, компоненты, встроенные функции для чтения, записи, просмотра изображений, представление звука; примеры загрузки изображений и звука в Mathcad.
- 3 Подготовить исходные данные к работе.
- 4 В системе Mathcad и в MS Visual Studio выполнить практические задания.
- 5 Сделать выводы.

*Задания*

- 1 Разработать программное средство для реализации следующих задач:
  - a) формирование двумерного цифрового массива двумя способами: на основе значений яркости пикселей цветного изображения и генерацией случайных чисел с помощью встроенного датчика случайных чисел (должна быть возможность отображения на форме двух введенных массивов);
  - b) сложение и поэлементное умножение двух введенных массивов (в программе должна быть возможность выбора);
  - c) расчет для массива математического ожидания, дисперсии и нормы;
  - d) центрирование и нормирование введенного массива;
  - e) сохранение обработанного массива в файле .txt и формата bmp (jpg и др.);

2 На основе разработанного программного средства и в системе Mathcad (рисунок 1.2-1.4) осуществить генерацию зашумленных данных и осуществить предварительную обработку данных, выполнив шаги:

a) задать двумерный массив (путем задания цифрового массива данных по изображению – задание 1a));

b) добавить аддитивный или мультипликативный шум в полученный массив (используя сложение или поэлементное умножение на массив сгенерированных случайных чисел – задание 1b));

c) рассчитать характеристики полученного массива (задание 1c)-e))

d) сохранить массив в файл и просмотреть результат.

3 Разработать программное средство для реализации задач:

a) ввод одномерного цифрового массива тремя способами: ввод из файла, генерацией случайных чисел, табулированием функции, заданной аналитически (согласно варианту) на  $[a;b]$  с шагом  $\Delta t$  (должна быть возможность отображения на форме двух введенных массивов и их хранение в оперативной памяти);

b) сложение двух введенных массивов и поэлементное умножение (по выбору);

c) расчет для массива математического ожидания, дисперсии и нормы;

d) центрирование и нормирование введенного массива;

e) построение графиков заданных функций (согласно варианту) и массива точек в одной системе координат;

f) сохранение обработанного массива в файле .txt и формата WAV (за основу взять проект syntezWAV, прикрепленный к лабораторной работе);

4 На основе разработанного программного средства и в системе Mathcad осуществить генерацию звука и шума.

a) смоделируйте звуковой сигнал (путем задания цифрового массива данных путем чтения из файла или по формуле – задание 3a));

b) добавьте аддитивный или мультипликативный шум в полученный сигнал (используя сложение или поэлементное умножение на массив сгенерированных случайных чисел – задание 1b));

- c) рассчитайте характеристики полученного сигнала (задание 3c)-e))  
 d) сохраните сигнал в файл, прослушайте.

### Варианты заданий к №3

Таблица 1.1 – Аналоговые сигналы

Вариант	Сигнал	Вариант	Сигнал
1	$s(x) = e^{-0.005(x-10)^2}$	6	$s(t) = \sum_{k=0}^N A_k \cdot \cos(2 \cdot \pi \cdot f_k \cdot t + \varphi_k)$
2	$s(t) = A \cdot \sin(\omega_0 t + \phi)$	7	$s(t) = \exp(-a \cdot t) - \exp(-b \cdot t)$
3	$s(t) = A \cdot \cos(\omega_0 t + \varphi)$	8	$s(t) = u(t) \cos(2\pi f_0 t + \varphi_0)$
4	$s(t) = \sum_{n=0}^N A_n \sin(2\pi f_n t + \varphi_n)$	9	$s(t) = 2A/T(t - kT),$ $(k-1/2)T < t <= (k+1/2)T$
5	$s(t) = A \sin(\pi t/T) / (\pi t/T)$	10	$s(t) = A(1 - 4 t - kT /T),$ $(k-1/2)T <= t < (k+1/2)T$

### *Отчет и контрольные вопросы*

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные изображения и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 Что такое цифровой сигнал?
- 2 Перечислите типы сигналов.
- 3 Чем отличается дискретный сигнал от цифрового?
- 4 Для чего нужна математическая модель сигнала?
- 5 Где применяются цифровые методы обработки данных?
- 6 Перечислите основные операции цифровой обработки сигналов.
- 7 Что такое растр, элемент растра?

## 2 Сглаживание экспериментальных данных

Сглаживание данных, как искаженных помехами, так и статистических по своей природе, можно считать частным случаем регрессии *без определения символьной формы* ее функции [6,16]. Сглаживание производится с помощью фильтров.

Предмет цифровой фильтрации данных (сигналов) является естественным введением в широкую и фундаментальную область цифровой обработки информации. Под фильтрацией будем понимать любое преобразование информации (сигналов, результатов наблюдений), при котором во входной последовательности обрабатываемых данных целенаправленно изменяются определенные соотношения (динамические или частотные) между различными компонентами этих данных [6].

В общем случае термином «Цифровой фильтр» называют аппаратную или программную реализацию математического алгоритма, входом которого является цифровой сигнал, а выходом – другой цифровой сигнал с определенным образом модифицированной формой и/или амплитудной и фазовой характеристикой [6].

К основным операциям фильтрации информации относят операции сглаживания, прогнозирования, дифференцирования, интегрирования и разделения сигналов, а также выделение информационных (полезных) сигналов и подавление шумов (помех) [6].

Одним из методов получения сглаживающих фильтров является метод наименьших квадратов.

### 2.1 Фильтрация на основе метода наименьших квадратов

#### 2.1.1 Фильтры МНК 1-го порядка

*Расчет коэффициентов фильтра.* Простейшим способом аппроксимации по МНК произвольной функции  $s(t)$  является аппроксимация полиномом первой степени, т.е. функцией вида  $y(t) = A + Bt$  (метод скользящих средних).

Для симметричного фильтра МНК на  $(2N+1)$  точек с окном от  $-N$  до  $N$  коэффициенты полинома, определенные путём поиска минимума функции остаточных ошибок приближения имеют вид:

$$A = \frac{1}{2N+1} \sum_{n=-N}^N s_n, \quad B = \frac{\sum_{n=-N}^N n \cdot s_n}{\sum_{n=-N}^N n^2}.$$

Подставив значения коэффициентов в уравнение аппроксимирующего полинома, переходят в систему координат по точкам  $k$  массива  $y(k+\tau) = A+B \cdot \tau$ , где отсчет  $\tau$  производится от точки  $k$  массива, против которой находится точка  $n = 0$  фильтра, и получают в общей форме уравнение фильтра аппроксимации [1,6]:

$$y(k+\tau) = \frac{1}{2N+1} \sum_{n=-N}^N s_{k-n} + \tau \frac{\sum_{n=-N}^N n \cdot s_{k-n}}{\sum_{n=-N}^N n^2}.$$

Для сглаживающего нерекурсивного цифрового фильтра (НЦФ) вычисления производятся непосредственно для точки  $k$  в центре окна фильтра ( $\tau = 0$ ), при этом фильтр принимает вид:

$$y_k = \frac{1}{2N+1} \sum_{n=-N}^N s_{k-n}. \quad (2.1)$$

*Характеристики фильтров.* Сумма квадратов значений нормированного импульсного отклика системы представляет собой *коэффициент усиления аддитивных шумов* во входном сигнале [6].

1 Сумма коэффициентов импульсного отклика является коэффициентом  $K_{nc}$  усиления постоянной составляющей входного сигнала.

2 Частотная характеристика фильтра представляет собой Фурье-образ его импульсной реакции, и наоборот. При  $\Delta t = 1$ :

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n) \exp(-j\omega n), \quad (2.2)$$

На рисунке 2.1 приведен пример экранной формы программного средства для расчета коэффициентов фильтров МНК 1-го порядка.

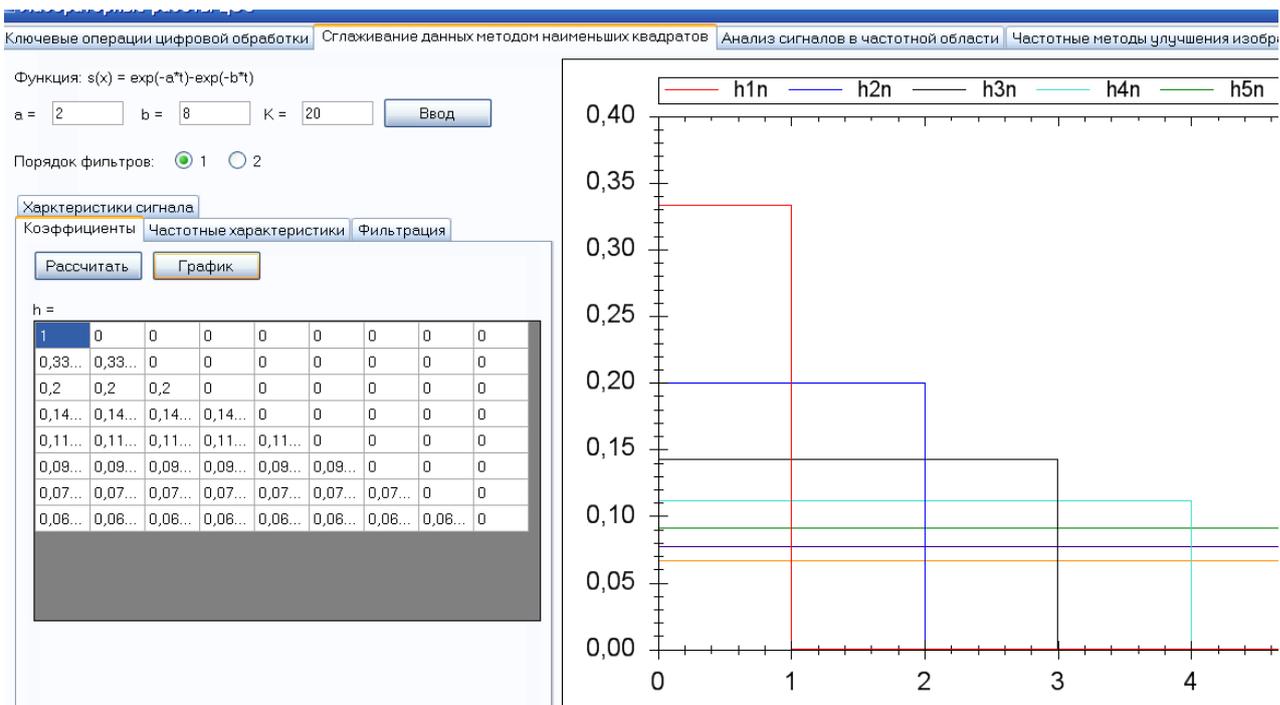


Рисунок 2.1 – Коэффициенты фильтров МНК 1-го порядка

### 2.1.2 Фильтры МНК 2-го порядка

*Расчет фильтров.* Фильтры МНК 2-го порядка (МНК-2) рассчитываются и анализируются аналогично 1-му порядку, коэффициенты полинома определяются путём поиска минимума функции остаточных ошибок приближения.

На основе квадратного многочлена вида  $y(t) = A + B \cdot t + C \cdot t^2$  можно построить симметричный сглаживающий НЦФ при  $\Delta t = 1$ , для которого потребуются только коэффициенты  $A$ :

$$A = \left\{ \sum_{n=-N}^N n^4 \sum_{n=-N}^N s_n - \sum_{n=-N}^N n^2 \sum_{n=-N}^N n^2 s_n \right\} / \left\{ \sum_{n=-N}^N 1 \sum_{n=-N}^N n^4 - \left[ \sum_{n=-N}^N n^2 \right]^2 \right\}. \quad (2.3)$$

На рисунках 2.2-2.4 приведены примеры экранных форм программного средства для расчета характеристик фильтров МНК 1-го порядка.

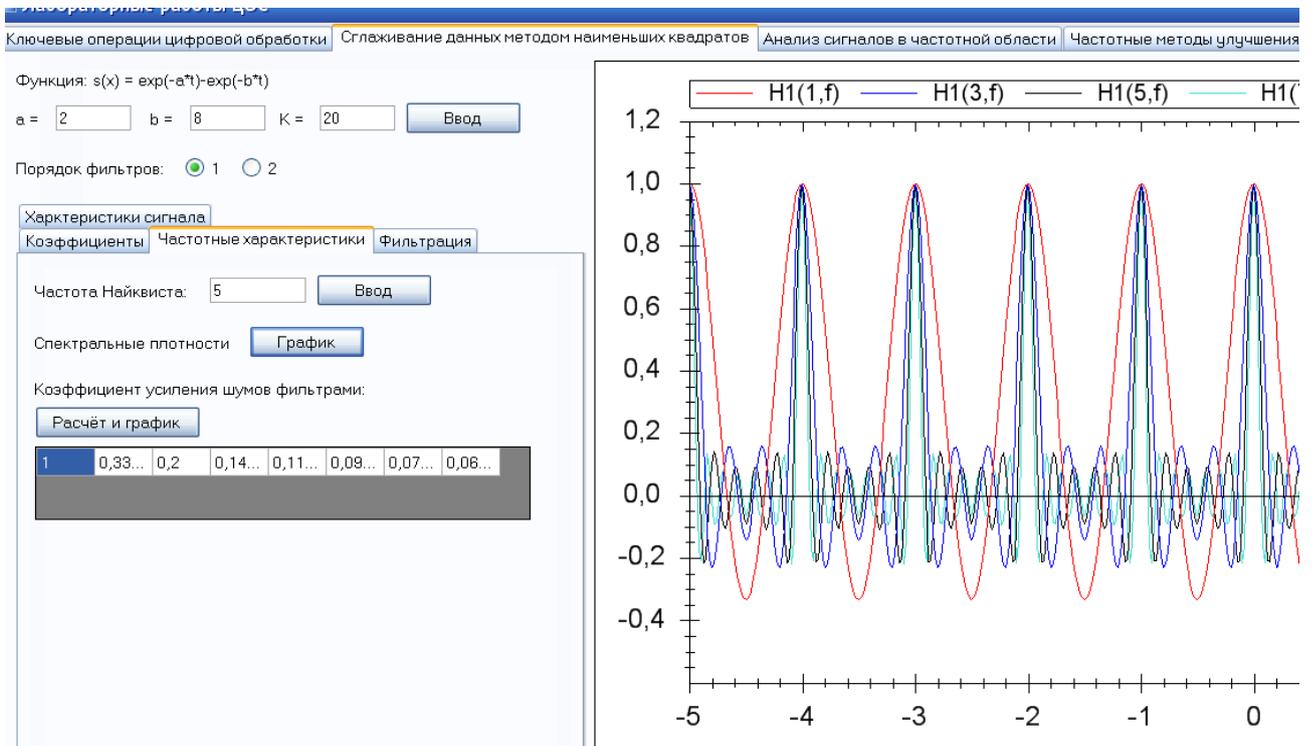


Рисунок 2.2 – Частотные характеристики фильтров МНК 1-го порядка

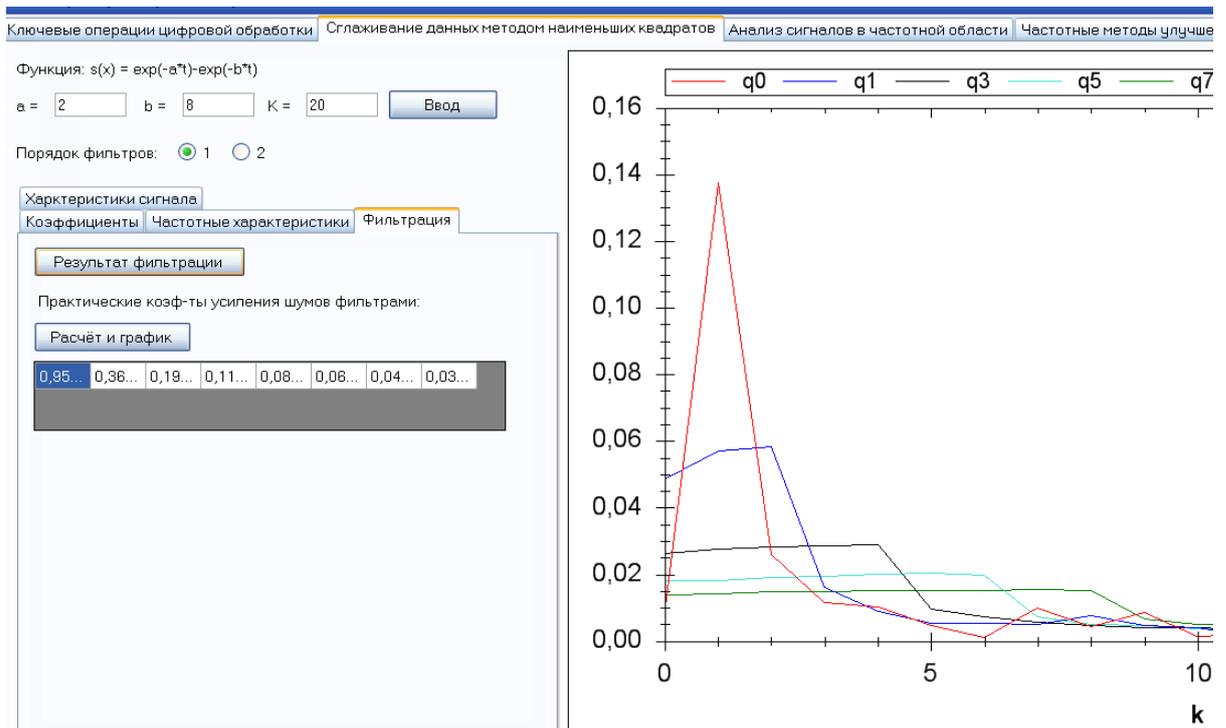


Рисунок 2.3 – Отфильтрованный сигнал фильтрами МНК 1-го порядка

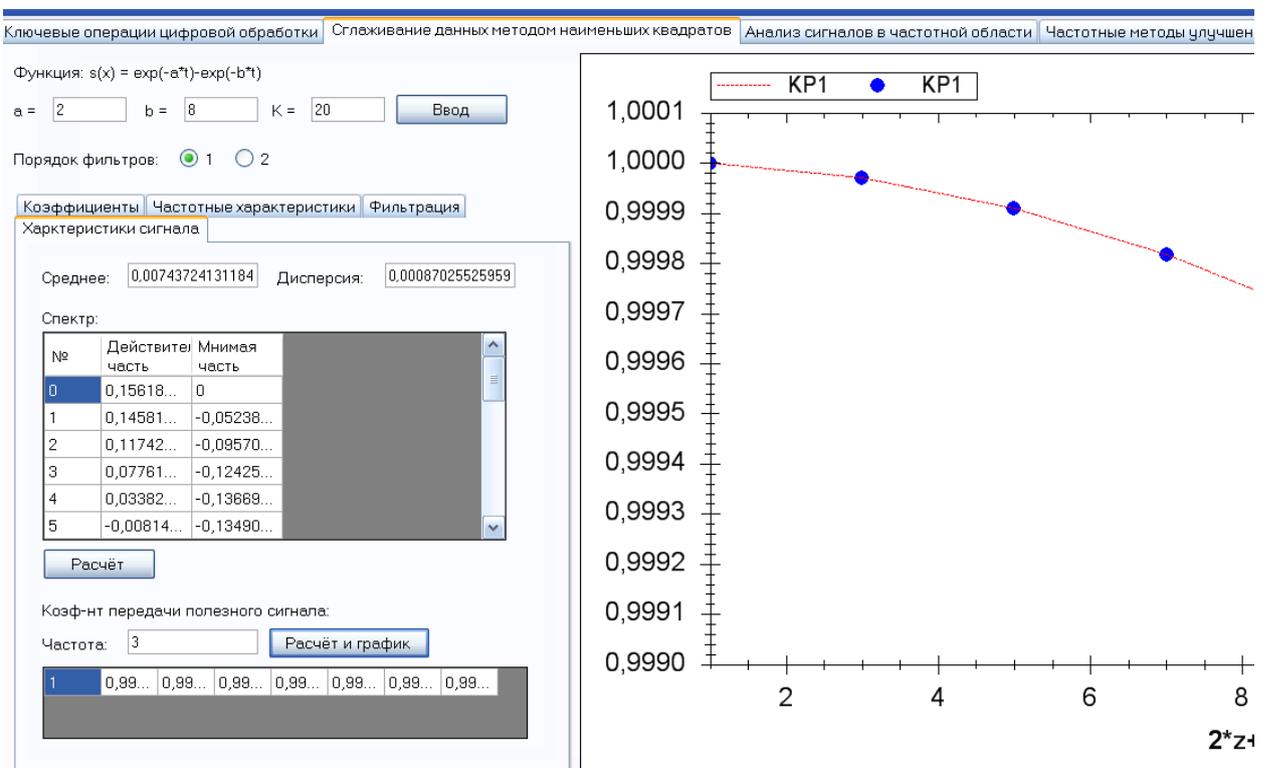


Рисунок 2.4 – Коэффициент передачи полезного сигнала фильтрами МНК 1-го порядка

Таким образом, фильтрация является одной из основных операций цифровой обработки, позволяет решать различные задачи, в том числе сглаживание данных.

## **2.2 Цель и задачи выполнения лабораторной работы**

*Цель:* Формирование умений сглаживания цифровых данных методом наименьших квадратов (МНК) (фильтры МНК)

*Задачи:*

- 1 Сглаживание цифровых данных фильтрами МНК 1-го порядка и 2-го порядка.
- 2 Расчет характеристик фильтра и цифровых данных.

## **2.3 Порядок выполнения работы**

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Прочитать теоретические предпосылки.
- 3 Подготовить исходные данные к работе.
- 4 Выполнить практические задания.
- 5 Сделать выводы.

*Задания*

Разработать модуль к программному средству из первой работы, выполняющий функции:

- 1 Добавить шум в полученный цифровой звук (см. раздел 1), используя датчик случайных чисел.
- 2 Рассчитать коэффициенты фильтра (2.1) для ширины окна  $N=0..7$ . Построить графики фильтров.
- 3 Рассчитать частотные характеристики фильтров: спектральные плотности фильтров, коэффициент усиления дисперсии шумов фильтрами.

- 4 Произвести фильтрацию шумов полученными фильтрами, рассчитать коэффициент  $K_{nc}$  усиления постоянной составляющей входного сигнала.
- 5 Рассчитать коэффициент передачи полезного сигнала.
- 6 Повторить действия по пунктам 3-6 для фильтров МНК 2-го порядка.

### *Отчет и контрольные вопросы*

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные графики и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 Что такое сглаживание данных?
- 2 Как определяются коэффициенты симметричного фильтра МНК?
- 3 Какими свойствами обладает симметричный фильтр МНК?
- 4 Перечислите характеристики фильтров.
- 5 Что показывает коэффициент передачи полезного сигнала?
- 6 Как влияет ширина окна на коэффициент передачи полезного сигнала?

### 3 Гармонический анализ

В основе гармонического анализа лежит предположение – временной ряд является суммой, или спектром многих волнообразных изменений, которые можно описать с помощью тригонометрических функций. Целью **гармонического** анализа является отыскание скрытых периодичностей и оценка их интенсивности, что позволяет строить усредненный прогноз [14].

Математической основой гармонического анализа является интегральное преобразование и ряды Фурье.

#### 3.1 Дискретное преобразование Фурье

Преобразование Фурье позволяет сопоставить сигналу, заданному во временной области, его эквивалентное представление в частотной области [6].

При дискретном представлении сигналов аргумент  $t_k$  обычно проставляется номерами отсчетов  $k$  (по умолчанию  $\Delta t = 1$ ,  $k = 0, 1, \dots, N-1$ ), а преобразования Фурье выполняются по аргументу  $n$  (номер шага по частоте) на главных периодах. При значениях  $N$ , кратных 2 [1,6, 18,19]:

$$S_n = \sum_{k=0}^{N-1} s_k \exp(-j2\pi kn/N), \quad n = -N/2, \dots, 0, \dots, N/2. \quad (3.1)$$

$$s_k = (1/N) \sum_{n=-N/2}^{N/2-1} S_n \exp(j2\pi kn/N), \quad k = 0, 1, \dots, N-1. \quad (3.2)$$

Главный период спектра в (3.1) для циклических частот от  $-0.5$  до  $0.5$ , для угловых частот от  $-\pi$  до  $\pi$ . При нечетном значении  $N$  границы главного периода по частоте (значения  $\pm f_N$ ) находятся на половину шага по частоте за отсчетами  $\pm(N/2)$  и, соответственно, верхний предел суммирования в (3.2) устанавливается равным  $N/2$ .

Прямое и обратное преобразование Фурье изображения, заданного функцией

$s(n,m)$  размером  $N$  на  $M$  определяются выражениями:

$$S(k,l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} s(n,m) \exp(-jn2\pi k/N - jm2\pi l/M), \quad (3.3)$$

$$S(k,l) = \sum_{n=0}^{N-1} \exp(-jn2\pi k/N) \sum_{m=0}^{M-1} s(n,m) \exp(-jm2\pi l/M), \quad (3.4)$$

$$s(n,m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} S(k,l) \exp(jn2\pi k/N + jm2\pi l/M). \quad (3.5)$$

$$s(n,m) = \frac{1}{NM} \sum_{k=0}^{N-1} \exp(jn2\pi k/N) \sum_{l=0}^{M-1} S(k,l) \exp(jm2\pi l/M). \quad (3.6)$$

$$k=0..N-1; l=0..M-1.$$

Преобразования (3.1-3.6) называют дискретными преобразованиями Фурье (ДПФ). Для ДПФ справедливы все свойства интегральных преобразований Фурье, однако при этом следует учитывать периодичность дискретных функций и спектров.

Алгоритм быстрого преобразования Фурье (БПФ) – это оптимизированный по скорости способ вычисления ДПФ. Основная идея заключается в двух пунктах [2].

Необходимо разделить сумму (3.1) из  $N$  слагаемых на две суммы по  $N/2$  слагаемых, и вычислить их по отдельности. Для вычисления каждой из подсумм, надо их тоже разделить на две и т.д.

Необходимо повторно использовать уже вычисленные слагаемые.

Применяют либо "прореживание по времени" (когда в первую сумму попадают слагаемые с четными номерами, а во вторую - с нечетными), либо "прореживание по частоте" (когда в первую сумму попадают первые  $N/2$  слагаемых, а во вторую - остальные). Оба варианта равноценны. В силу специфики алгоритма приходится применять только  $N$ , являющиеся степенями 2 [2,18].

В основе алгоритма БПФ в случае прореживания по времени лежат следующие формулы:

$$x_{[even]n} = x_{2n}, \quad x_{[odd]n} = x_{2n+1}, \quad n = 0, 1, \dots, N/2-1$$

$$X_{\{N\}k} = X_{\{N/2\}[even]k} + W_N^k X_{\{N/2\}[odd]k}$$

$$X_{\{N\}N/2+k} = X_{\{N/2\}[even]k} - W_N^k X_{\{N/2\}[odd]k}, k=0,1,\dots,N/2-1$$

$$W_N = e^{\frac{-j2\pi}{N}} = W_{2^m}^{2^m}; X_{\{1\}0} = x_0.$$

Стандартные средства дискретных преобразований Фурье в системе *Mathcad*. *Mathcad* содержит функции для выполнения быстрого дискретного преобразования Фурье (БПФ) и его обращения в вещественной и комплексной области. В *Mathcad* имеется также одномерное дискретное волновое преобразование и его обращение [15,18].

Ниже приведены два типа функций для дискретного преобразования Фурье: *fft/ifft* и *cfft/icfft*.

**fft (v)** Возвращает дискретное преобразование Фурье  $2^m$ -мерного вещественнозначного вектора. Аргумент можно интерпретировать как результат измерений через равные промежутки времени некоторого сигнала.

Вектор **v** должен иметь  $2^m$  элементов. Результат — комплекснозначный вектор размерности  $1+2^{m-1}$ .

**ifft (v)** Возвращает обратное дискретное преобразование Фурье; результат — вещественнозначный.

Вектор **v** должен иметь  $1+2^m$  элементов, где  $m$  — целое. Комплексное преобразование Фурье требует следующих функций:

**cfft (A)** Возвращает дискретное преобразование Фурье комплекснозначных вектора или матрицы. Возвращаемый массив имеет тот же самый размер, что и массив, используемый как аргумент.

**icfft (A)** Возвращается обращение дискретного преобразования Фурье вектора или матрицы данных. Функция *icfft* — обратная к функции *cfft*. Подобно *cfft*, эта функция возвращает массив того же самого размера, что и аргумент.

На рисунках 3.1 и 3.2 представлены примерные экранные формы программной реализации фильтрации спектра.

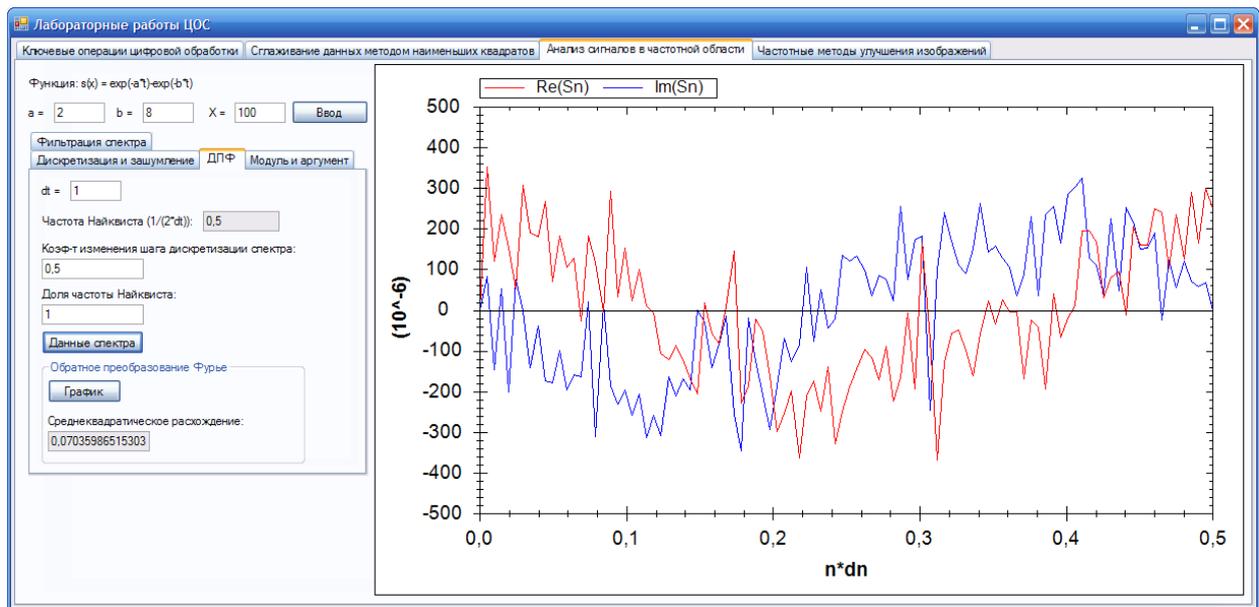


Рисунок 3.1 – Спектр функции

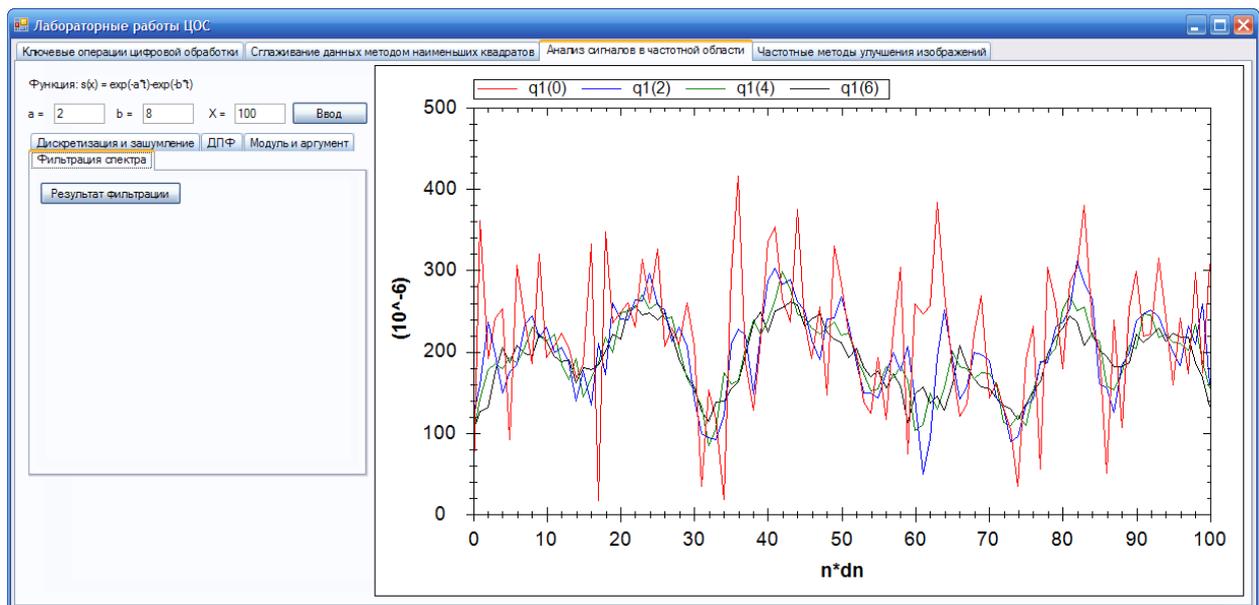


Рисунок 3.2 – Результат фильтрации спектра фильтрами МНК первого порядка

Таким образом, анализ и обработка сигналов в частотной области на основе преобразования Фурье позволяет использовать дополнительные к временным частотные характеристики и выявлять скрытые периодические закономерности.

### **3.2 Цель и задачи выполнения лабораторной работы**

*Цель:* Формирование умений выполнения дискретного преобразования Фурье, расчета характеристик спектра, очистки сигнала от шумов в частотной области.

*Задачи:*

- 1 Программная реализация прямого и обратного преобразования Фурье для заданного цифрового массива.
- 2 Расчет частотных характеристик: модуля спектра и аргумента спектра.
- 3 Сглаживание спектра.

### **3.3 Порядок выполнения работы**

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Прочитать теоретические предпосылки.
- 3 Подготовить исходные данные к работе.
- 4 Выполнить практические задания.
- 5 Сделать выводы.

*Задания*

- 1 Сгенерировать цифровой звук (см. раздел 1, задание №3).
- 2 Добавить шум в полученный сигнал, используя датчик случайных чисел (отсчёты с равномерным распределением). Построить графики.
- 3 Для полученного зашумлённого сигнала выполнить:
  - а) Разработать программный модуль, реализующий дискретное преобразование Фурье.
  - б) Построить графики модуля спектра и аргумента спектра.
- 4 После прямого преобразования произвести фильтрацию спектра по МНК.
- 5 Реализовать обратное преобразование Фурье.
- 6 Проверить работу программы с помощью системы Mathcad (файл “furtr-преобразование Фурье.mcd”).

## *Отчет и контрольные вопросы*

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные графики и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 Запишите формулы ДПФ.
- 2 Что такое поворачивающий множитель?
- 3 Чему равно разрешение по частоте при вычислении ДПФ?
- 4 С чем связаны трудности прямого вычисления ДПФ?
- 5 Что такое БПФ?
- 6 Назовите основные свойства ДПФ.
- 7 Поясните смысл ДПФ для периодической последовательности.
- 8 Как с помощью ДПФ рассчитывается амплитудный и фазовый спектры периодической последовательности?

## 4 Вейвлет анализ

### 4.1 Быстрое вейвлет-преобразование

Вейвлетное преобразование сигналов является обобщением спектрального анализа. В отличие от преобразования Фурье, базисными функциями которого являются гармонические функции, вейвлет-преобразования основаны на разложении по различным базисам – вейвлетам («маленьким волнам»), изменяющейся частоты и ограниченными во времени (в пространстве), что позволяет реализовать различные варианты соотношения неопределенности [4].

По сравнению с разложением сигналов на ряды Фурье вейвлеты способны с гораздо более высокой точностью представлять локальные особенности сигналов, вплоть до разрывов 1-го рода (скачков) [6].

Вейвлетный спектр, в отличие от спектра Фурье, является двумерным и определяет двумерную поверхность в пространстве переменных  $t$  - масштаб и  $k$  - сдвиг. При графическом представлении параметр растяжения/сжатия спектра  $t$  откладывается по оси абсцисс, параметр локализации  $k$  по оси ординат – оси независимой переменной сигнала, что дает возможность анализа сигналов сразу в двух пространствах [6].

В основе вейвлет-преобразований лежит использование двух взаимозависимых функций:

- вейвлет-функции  $\psi(t)$ , как функции времени с нулевым средним, которая используется для выделения локальных особенностей сигнала;

- масштабирующей функции  $\varphi(t)$ , как временной скейлинг-функции с единичным средним, с помощью которой выполняется грубое приближение (аппроксимация) сигнала.

*Принцип преобразования* [1,2,3,6,7,14]. Любую функцию  $s(t)$  можно рассматривать на любом  $m'$  - уровне разрешения. Для разделения функции на этом уровне между ее усредненными значениями и флюктуациями вокруг средних значений имеет место формула [1,7,23]:

$$s(t) = \sum_{k=-\infty}^{\infty} C_{m',k} \varphi_{m',k}(t) + \sum_{m=m'}^{\infty} \sum_{k=-\infty}^{\infty} D_{m,k} \psi_{m,k}(t). \quad (4.1)$$

На практике обычно имеют дело с цифровыми данными в виде конечного набора отсчетов, а, соответственно, наилучший уровень разрешения определен интервалом, содержащим один отсчет, и суммирование выполняется в конечных пределах. Значение  $m=0$  обычно принимается для этого наилучшего уровня разрешения. Для принятой формы вейвлетов  $\varphi_{m,k} = 2^{m/2} \varphi(2^m t - k)$  усреднение отсчетов (расширение размеров вейвлетов) происходит при уменьшении значений  $m$ , т.е. при  $m = 0, -1, -2, \dots$ . Для исключения использования отрицательных индексов масштабирования знак "минус" обычно вводится непосредственно в функции вейвлетов, т.е.  $\varphi_{m,k} = 2^{-m/2} \varphi(2^{-m} t - k)$ , при этом вейвлет-коэффициенты вычисляются для значений уровней масштаба  $m > 0$  [6].

*Алгоритм Малла.* Кратномасштабный анализ при последовательном увеличении значений  $m$  приводит к естественной форме быстрых итерационных вычислений [7, 23]:

$$C_{m+1,k} = \sum_n h_n C_{m,2k+n}, \quad (4.2)$$

$$D_{m+1,k} = \sum_n g_n C_{m,2k+n}, \quad (4.3)$$

$$C_{0,k} = \int_{k\Delta t}^{(k+1)\Delta t} s(t) \varphi(t-k) dt. \quad (4.4)$$

Уравнения обеспечивают пирамидальный алгоритм вычисления вейвлет-коэффициентов (алгоритм Малла), приведенный на рисунке 4.1. Для цифровых данных в качестве значений  $C_{0,k}$  принимаются исходные значения данных, т.е.  $C_{0,k} = s(k)$ . Явный вид вейвлета требуется только для расчета коэффициентов  $h_n$  и  $g_n$ , а при собственно быстром вейвлет-преобразовании он не используется – используются полученные значения коэффициентов  $h_n$  и  $g_n$  на соответствующем уровне детализации.

*Коэффициенты вейвлета.* Значения коэффициентов  $h_k$  и  $g_k$  определяются на основании общих свойств скейлинг-функций и вейвлетов, которые приводят к решению следующих уравнений при различных  $M$

$$\sum_k h_k h_{k+2x} = \delta_x, \quad (4.5)$$

$$\sum_k h_k = \sqrt{2}, \quad (4.6)$$

$$\sum_k h_k g_{k+2x} = 0, \quad (4.7)$$

$$g_k = (-1)^k h_{2M-1-k}, \quad (4.8)$$

$$m = 0, 1, \dots, M-1, \quad \sum_k k^m g_k = 0, \quad (4.9)$$

$$\sum_k (-1)^k k^m h_k = 0. \quad (4.10)$$

Например, при  $M=2$  коэффициенты равны

$$h_0 = 2^{-5/2} (1 + \sqrt{3}) = 0.483, \quad h_1 = 2^{-5/2} (3 + \sqrt{3}) = 0.837,$$

$$h_2 = 2^{-5/2} (3 - \sqrt{3}) = 0.224, \quad h_3 = 2^{-5/2} (1 - \sqrt{3}) = -0.129.$$

Соответственно, значения коэффициентов  $g_k$ , вычисленные по (4.8):

$$g_0 = -0.129, \quad g_1 = -0.224, \quad g_2 = 0.837, \quad g_3 = -0.483.$$

Простейший и самый короткий вейвлет соответствует  $M=1$  (вейвлет Наара), значения коэффициентов равны  $h_0 = h_1 = 1/\sqrt{2}$ ,  $g_0 = 1/\sqrt{2}$ ,  $g_1 = -1/\sqrt{2}$ .

*Реконструкция сигналов.* Поскольку в формулах (4.2, 4.3) вместо базисных функций используются фильтры, то обратные преобразования, т.е. последовательную сборку сигнала от больших  $m$  к малым и реконструкцию сигналов по значениям его вейвлет-коэффициентов с любого уровня разрешения, также можно произвести через фильтры реконструкции [6,7]:

$$C_{m-1} = \sum_{n \in I} C_{m,n} h^r_{k-2n} + \sum_{n \in I} D_{m,n} g^r_{k-2n}, \quad (4.11)$$

где  $h^r(k) = \text{reverse}(h(k))$ ,  $g^r(k) = \text{reverse}(g(k))$ .

Алгоритм вычислений по (4.11) обратен алгоритму декомпозиции, т.е. представляет собой аппроксимацию коэффициентов  $C_m$  и  $D_m$  на новый, в 2 раза меньший, шаг дискретизации с двукратным увеличением частоты Найквиста и восстановлением спектра коэффициентов  $C_m$  в низкочастотную часть нового главного диапазона спектра  $C_{m-1}$ , а спектра коэффициентов  $D_m$  в высокочастотную часть спектра  $C_{m-1}$ . Это выполняется расстановкой нулевых значений между коэффициентами  $C_m$  и  $D_m$  (увеличение в 2 раза числа отсчетов), фильтрацией полученных массивов низкочастотным  $h^r(k)$  и высокочастотным  $g^r(k)$  фильтрами реконструкции, и сложением результатов фильтрации.

*Пакетные вейвлеты.* Основная информация обычно заключена в низкочастотной части сигнала, разложение которой может быть продолжено до нулевого уровня. Но аналогичная операция может применяться и к любой высокочастотной части разложения: для коэффициентов 'D' любого уровня. Разложение представляется в виде дерева (рисунок 4.1). Такое разложение является адаптивным и легко приспособляется к индивидуальным особенностям сигналов. Функции адаптивного преобразования называют вейвлет-пакетом [1,6,26].

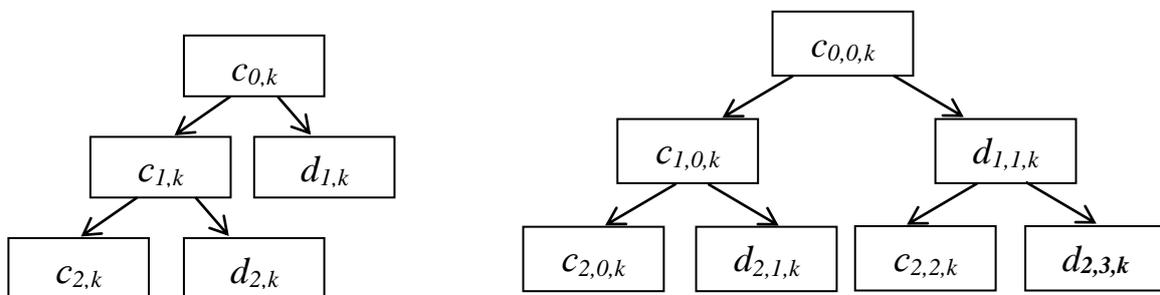


Рисунок 4.1 – Схема алгоритма Малла и пакетного БВП

Вейвлет-преобразование двумерного цифрового сигнала можно выполнить путем последовательного применения одномерного вейвлет преобразования к строкам и столбцам матрицы, описывающей двумерный сигнал.

Пусть задан двумерный сигнал  $s$  – матрица конечного размера. Применяя к каждой строке матрицы один шаг одномерного вейвлет-преобразования получатся

две матрицы, строки которых содержат аппроксимированную и детализированную составляющие строк исходной матрицы. К каждому столбцу обеих матриц также применяется шаг одномерного преобразования. В результате получается четыре матрицы: низкочастотная и три высокочастотные: вертикальная, горизонтальная и диагональная детализирующая информация [3,4].

Таким образом, шаг двумерного преобразования сводится к композиции одномерных преобразований и не требует дополнительных операций.

*Стандартные средства дискретных вейвлет-преобразований системы Mathcad.* В Mathcad включены две функции волновых преобразований: для выполнения прямого одномерного дискретного волнового преобразования и его обращения. Преобразование выполняется с использованием четырех коэффициентного волнового базиса Добеши [10,15].

wave ( $\mathbf{v}$ ) Возвращает дискретное волновое преобразование  $\mathbf{v}$ ,  $2^m$ -мерного вещественнозначного вектора. Возвращается вектор того же самого размера, что и  $\mathbf{v}$ .

iwave ( $\mathbf{v}$ ) Возвращает обращение дискретного волнового преобразования  $\mathbf{v}$ ,  $2^m$ -мерного вещественнозначного вектора. Возвращается вектор того же самого размера, что и  $\mathbf{v}$ .

Таким образом, вейвлет-преобразование имеет ряд преимуществ по сравнению с широко применяемым преобразованием Фурье и может обеспечить большую эффективность.

## **4.2 Цель и задачи выполнения лабораторной работы**

*Цель:* Формирование умений вейвлетного представления и обработки сигналов на различных уровнях декомпозиции.

*Задачи:*

- 1 Выполнение прямого и обратного быстрого вейвлет-преобразования.
- 2 Расчет энергии по уровням вейвлет-коэффициентов.

3 Фильтрация высокочастотных составляющих вейвлетного спектра на заданном уровне.

### 4.3 Порядок выполнения работы

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Прочитать теоретические предпосылки.
- 3 Подготовить исходные данные к работе.
- 4 Выполнить практические задания.
- 5 Сделать выводы.

#### *Задания*

1 Сформировать двумерный массив на основе значений яркости изображения (см. раздел 1, задание 2).

2 Добавить шум в полученный сигнал, используя датчик случайных чисел (отсчёты с равномерным распределением). Вывести изображения на форму.

3 Для полученного зашумлённого сигнала выполнить (вид преобразования взять согласно варианту (таблица 4.1)):

а написать программный модуль, реализующий быстрое вейвлет-преобразование двумерного массива, вывести на форму изображения коэффициентов по уровням разложения;

б рассчитать энергии спектра (вейвлет-коэффициентов) по уровням разложения.

4 После прямого преобразования произвести фильтрацию спектра (полученных вейвлет-коэффициентов) на заданном уровне (предусмотреть возможность варьирования количества уровней).

5 Восстановить сигнал по отфильтрованным вейвлет-коэффициентам.

Таблица 4.1 – Варианты базисных функций

Вариант	Преобразование	Вариант	Сигнал и преобразование
1	Вейвлет-пакеты (Добеши, $M=2$ )	11	Алгоритм Мала (Добеши, $M=2$ )
2	Алгоритм Мала (Добеши, $M=2$ )	12	Вейвлет-пакеты (Добеши, $M=2$ )
3	(Добеши, $M=2$ ) Алгоритм Малла	13	Вейвлет-пакеты (Добеши, $M=1$ )

4	Вейвлет-пакеты (Добеши, $M=2$ )	14	Алгоритм Мала (Добеши, $M=1$ )
5	Алгоритм Мала (Добеши, $M=2$ )	15	(Добеши, $M=2$ ) Алгоритм Малла

### *Отчет и контрольные вопросы*

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные графики и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 В чем заключается главная идея вейвлетного представления сигналов на различных уровнях декомпозиции?
- 2 В чем преимущество вейвлет-преобразования перед преобразованием Фурье?
- 3 Что характеризуют низкие и высокие частоты?
- 4 Как математически представляется вейвлет?
- 5 Какую роль выполняют в вейвлет-преобразовании вейвлет-функция и масштабирующая функция?
- 6 Перечислите свойства вейвлетов.
- 7 В чем заключается самоподобие базиса вейвлет-преобразования?

## 5 Сжатие данных

### 5.1 Сжатие изображений на основе дискретных преобразований

Изображения требуют много памяти для хранения. Поэтому изображения сжимают, т.е. производят алгоритмическое преобразование изображений, уменьшающее их объём. А во время просмотра записанные в память данные декодируются и восстанавливается исходный кадр [3,22]. В основе сжатия лежит устранение некоторой избыточности, содержащейся в исходных данных. Различают сжатие с потерями и без потерь.

При использовании сжатия без потерь возможно полное восстановление исходных данных, сжатие с потерями позволяет восстановить данные с искажениями несущественными с точки зрения дальнейшего использования восстановленных данных.

При работе с аудио- и видеоданными и цифровыми фотографиями, когда полное соответствие исходных и восстановленных данных не требуется, используется сжатие с потерями, обеспечивающее сокращение объема с большой эффективностью.

*Дискретные преобразования изображений.* Одним из эффективных подходов к сжатию данных является использование дискретных преобразований. Основная идея подхода состоит в представлении изображений коэффициентами их дискретного преобразования (трансформантами) [3,20,22]. Коэффициенты преобразования упорядочиваются в соответствии с их важностью, так что трансформанты с небольшим информационным содержанием можно отбросить. Оставшиеся трансформанты квантуются, кодируются и передаются или запоминаются. [26].

Широко известны дискретные преобразования Фурье, Адамара, Уолша, косинус- и синус-преобразования (ДКП), имеющие быстрые схемы реализации с вычислительной сложностью порядка  $O(MN \log_2(MN))$ .

Если изображение сжимается применением дискретного преобразования, то оно обычно разбивается на блоки размером 8x8 или 16x16 пикселей, тогда каждый блок преобразуется и обрабатывается отдельно для ускорения вычислений. Здесь не учитывается корреляция между блоками изображения, которая может оказаться серьезным источником информационной избыточности итогового сжатия. Подходом, позволяющим снизить межблочную избыточность и фрагментарность преобразования, является *рекурсивное блочное кодирование*, а также *различные виды дискретного вейвлет-преобразования* (ДВП) [22,26].

Разбиение на блоки с последующим ДКП-сжатием приводит к снижению качества восстановленного изображения при больших коэффициентах сжатия, в отличие от сжатия изображений посредством ДВП, которое может быть эффективно применено к целому изображению, и при нем не будут присутствовать артефакты блочного разбиения [26].

Вейвлет-сжатие сводится к тем же этапам обработки, что и ДКП-преобразование, квантование и кодирование ДВП-трансформант.

*Сжатие изображений с использованием вейвлетов.* Вейвлетное сжатие изображений использует двумерное вейвлет-разложение кодируемого изображения или его частей.

Алгоритмы вейвлетной компрессии используют представления результатов вейвлет-разложения в виде нуль-дерева (zero-tree) (рисунок 4.1).

Упорядоченные в нуль-дереве битовые плоскости коэффициентов вейвлет-разложения *огрубляются* (квантование коэффициентов) и кодируются далее с использованием статистических методов сжатия [22].

Для работы с дискретными изображениями используется быстрый алгоритм вейвлет-преобразования, известный как алгоритм Малла. Исходное изображение раскладывается на две составляющие, вдвое меньше исходного изображения – высокочастотные детали (состоящие в основном из резких перепадов яркости), и сглаженную уменьшенную версию оригинала. В алгоритме используются фильтры с конечным импульсным откликом, в которых пиксели, попавшие в небольшое «окно», умножаются на заданный набор коэффициентов, полученные значения

суммируются, и окно сдвигается для расчёта следующего значения на выходе (см. формулы 4.2 -4.4, 4.11). В качестве базисных функций в задаче сжатия данных предпочтительнее использовать вейвлеты с нулевыми моментами скейлинг-функции, такие как вейвлеты Добеши, койфлеты, обеспечивающих лучшую сжимаемость за счет формирования большего количества малозначимых коэффициентов разложения.

Сжатие основано на особенностях восприятия человеком изображения: наибольшей чувствительности в определенном диапазоне волн цвета, способности воспринимать изображение как единое целое, не замечая мелких искажений (деталей) [6]. Изображения «деталей» состоят обычно из набора резких границ, характеризуются высокочастотными компонентами и содержат обширные участки, где интенсивность близка к нулю. Учитывая тот факт, что человеческий глаз практически не замечает изменения в высокочастотных составляющих допустимо пренебречь некоторым количеством мелких деталей, обнулив эти значения, если они не превышают заданного порога. В результате получается версия исходного изображения, хорошо поддающаяся сжатию. Для восстановления оригинала снова применяется алгоритм Малла, но с парой фильтров, обратной к исходным.

Таким образом, в задаче сжатия данных использование вейвлетов позволит добиться большего коэффициента сжатия за счет выбора базисных функций, дающих большее количество малозначимых коэффициентов разложения.

## **5.2 Цель и задачи выполнения лабораторной работы**

*Цель:* Формирование умений сжатия изображений на основе дискретного преобразования.

*Задачи:*

- 1 Подготовка изображения для сжатия.
- 2 Реализация алгоритма сжатия изображения, использующего вейвлетное преобразование.

### 5.3 Порядок выполнения работы

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Прочитать теоретические предпосылки.
- 3 Подготовить исходные данные к работе.
- 4 Выполнить практические задания.
- 5 Сделать выводы.

#### *Задания*

1. Разработать программные модули для реализации задач:
  - a) выбор максимального уровня разложения;
  - b) вейвлет-преобразование двумерного цифрового массива (прямое и обратное);
  - c) извлечение информации из значений вейвлетных коэффициентов, посредством расчета суммарной энергии в каждой частотной полосе  $k$ ;
  - d) нормировка вейвлет-коэффициентов в каждой частотной полосе  $k$ ;
  - e) выбор порога квантования ( $\epsilon=1; 0,1; 0,01;0,001;0,0001$ );
  - f) квантование (округление) вейвлет-коэффициентов с выбранным порогом;
  - g) кодирование коэффициентов по методу Хаффмена;
  - h) восстановление изображения;
  - i) расчет среднеквадратичного отклонения, коэффициента сжатия.
2. На основе разработанного программного средства осуществить сжатие изображения.
  - a) сформировать цифровые данные на основе изображения (см. раздел 1, задание 2);
  - b) применить алгоритм сжатия изображения, использующий вейвлетное преобразование (задание 1a)-i));
  - c) сохранить полученное изображение в файл, просмотреть.

### *Отчет и контрольные вопросы*

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные графики и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 В чем заключается главная идея сжатия на основе дискретных преобразований?
- 2 Что лежит в основе сжатия изображений?
- 3 В каких случаях используется сжатие с потерями?
- 4 На каких особенностях восприятия человеком изображения основано сжатие изображений?
- 5 Как осуществляется вейвлетное сжатие изображений?

## **6 Сжатие звука**

### **6.1 Основы сжатия звука**

С развитием средств коммуникации возникла необходимость в передаче больших объёмов данных, что привело к разработке различных способов их сжатия. Сжатие (компрессия) аудиоданных представляет собой процесс уменьшения скорости цифрового потока за счет сокращения статистической и психоакустической избыточности цифрового звукового сигнала [20, 21, 24].

Методы сокращения статистической избыточности аудиоданных также называют сжатием без потерь, а, соответственно, методы сокращения психоакустической избыточности - сжатием с потерями [20].

Алгоритмы сжатия звуковых файлов реализуются в компьютерных программах, называемых *аудиокодеками*. Изобретение специальных алгоритмов сжатия звуковых данных мотивировано тем, что общие алгоритмы сжатия неэффективны для работы со звуком и делают невозможным работу в реальном времени [21].

*Сжатие без потерь.* Сокращение статистической избыточности основано на учете свойств самих звуковых сигналов. Она определяется наличием корреляционной связи между соседними отсчетами цифрового звукового сигнала, устранение которой позволяет сокращать объем передаваемых данных на 15...25% по сравнению с их исходной величиной. Сложность сжатия звука без потерь заключается в том, что записи звука являются чрезвычайно сложными в своей структуре. Одним из методов сжатия является поиск образцов и их повторений, однако этот метод не эффективен для более хаотических данных, которыми являются, например оцифрованный звук или фотографии.

Другая сложность заключается в том, что звучание обычно меняется очень быстро и это также является причиной того, что упорядоченные последовательности байтов появляются очень редко.

Для передачи сигнала необходимо получить более компактное его представление, что возможно осуществить с помощью ортогонального преобразования.

Важными условиями применения такого метода преобразования являются:

- возможность восстанавливать исходный сигнал без искажений;
- способность обеспечивать наибольшую концентрацию энергии в небольшом числе коэффициентов преобразования;
- быстрый вычислительный алгоритмом.

Уменьшить скорость цифрового потока позволяют методы кодирования, учитывающие статистику звуковых сигналов, например, вероятности появления уровней разной величины. Одним из таких методов является код Хаффмана, где наиболее вероятным значениям сигнала приписываются более короткие кодовые слова, а значения отсчетов, вероятность появления которых мала, кодируются кодовыми словами большей длины. Именно в силу этих двух причин в наиболее эффективных алгоритмах компрессии цифровых аудиоданных кодированию подвергаются не сами отсчеты звукового сигнала, а коэффициенты модифицированного *дискретно-косинусного преобразования*.

Наиболее распространенными форматами сжатия без потерь являются: Free Lossless Audio Codec (FLAC), Apple Lossless , MPEG-4 ALS , Monkey's Audio , и TTA [20].

*Сжатие с потерями.* Сжатие с потерями имеет чрезвычайно широкое применение. Кроме компьютерных программ, сжатие с потерями используется в потоковом аудио в DVD, цифровом телевидении и радио и потоковому медиа в интернете.

Сжатие с потерями основано на сокращении психоакустической избыточности. Психоакустика - это наука, изучающая восприятие звука человеком. При составлении алгоритмов сокращения психоакустической избыточности (т.е. для исключения из звукового сигнала "лишней" информации) может быть взят за основу *эффект маскирования*. Кодек вычисляет, какие частоты в какой момент времени не будут восприниматься вашей слуховой системой и просто удаляет их из сигнала.

Этот эффект напрямую зависит от частот тихого и громкого звуков: низкочастотные звуки маскируют высокочастотные, но высокочастотные маскировать низкочастотные не могут. Отсюда и характерные срезы верхних частот на спектрах при lossy-кодировании.

Эффекты слухового маскирования зависят от спектральных и временных характеристик маскируемого и маскирующего сигналов и могут быть разделены на две основные группы:

- частотное (одновременное) маскирование
- временное (неодновременное) маскирование

Эффект маскирования в частотной области связан с тем, что в присутствии больших звуковых амплитуд человеческое ухо нечувствительно к малым амплитудам близких частот. То есть, когда два сигнала одновременно находятся в ограниченной частотной области, то более слабый сигнал становится неслышимым на фоне более сильного.

Маскирование во временной области характеризует динамические свойства слуха, показывая изменение во времени относительного порога слышимости (порог слышимости одного сигнала в присутствии другого), когда маскирующий и маскируемый сигналы звучат не одновременно [20].

Наиболее распространенными форматами сжатия с потерями являются: AAC, ADPCM, ATRAC, Dolby AC-3, MP2, MP3, Musepack Ogg Vorbis, WMA и другие.

## **6.2 Перцепционное сжатие звука с использованием вейвлет-пакетов**

Перцепционное кодирование подразумевает способ преобразования потока звуковых данных, при котором кодированию подвергается только та часть информации, которую способен воспринять слуховой анализатор человека [17]. Часть информации, которая не может быть воспринята, устраняется из потока данных. Другая часть информации, параметры которой могут восприниматься только частично, кодируется более грубо, с использованием меньшего числа

информационных разрядов. Таким образом, часть исходной информации утрачивается.

На рисунке 6.1 приведена общая структура кодера звука, использующего перцепционный принцип сжатия.

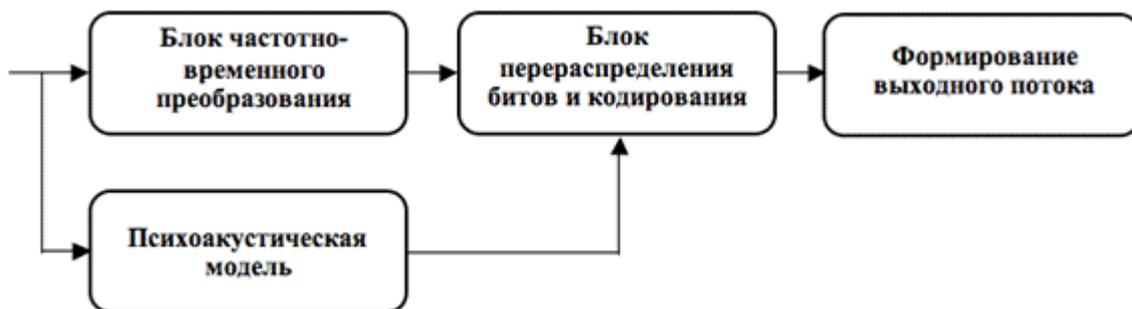


Рисунок 6.1 – Структура стандартного перцепционного кодера

Входной сигнал делится на кадры определенной длины, затем сигнал внутри каждого кадра подвергается частотно-временному преобразованию.

Параллельно происходит вычисление энергетического спектра сигнала в блоке психоакустического анализа. На основании данных о спектре сигнала вычисляется *глобальный порог маскирования*, позволяющий *определить количество битов*, до которого коэффициенты преобразования в той или иной частотной полосе могут быть сокращены, так, чтобы при этом шумы квантования оставались неслышимыми, то есть *замаскированными*.

Далее происходит переквантование коэффициентов преобразования на основе данных психоакустической модели. Кодеры достигают большей эффективности за счет устранения не только психоакустической избыточности, но и статистической избыточности, которая устраняется с помощью энтропийного кодирования.

Идея кодеков, использующих вейвлетное преобразование, заключается в *переквантовании коэффициентов вейвлетного преобразования на основе порога маскирования, полученного из психоакустической информации*.

Психоакустическая информация может быть извлечена непосредственно из значений вейвлетных коэффициентов путем расчета *суммарной энергии* в каждой частотной полосе  $k$  [17]:

$$S(k) = \sum_{i=1}^{L(k)} (X(i, k))^2, \quad k \in x[1, 28] \quad (6.1)$$

выполнении нормировки

$$S'(k) = \frac{S(k)}{N \frac{\sum_{i=1}^N \omega(i)^2}{N}} = \frac{S(k)}{\sum_{i=1}^N \omega(i)^2}, \quad (6.2)$$

где  $N$  – размер кадра;

$w(i)$  – оконная функция.

Уровень звукового давления в полосе

$$P(k) = 90,302 + 10 \cdot \lg S'(k). \text{ дБ} \quad (6.3)$$

Уровень порога маскирования зависит от уровня маскира и его типа. При этом тональные и шумовые маскиры имеют определенные отличия. Для определения типа маскира простым, но эффективным способом является гипотеза о том, что маскиры, расположенные в низкочастотной области спектра, ближе по своим маскирующим свойствам к тональным, в то время как маскиры из высокочастотной области, наоборот, ближе к шумовым. Придерживаясь этой гипотезы, следует выбрать ту условную частоту или критическую полосу  $k_M$ , выше которой маскиры принимаются как шумовые, а ниже – как тональные.

Уровень относительного порога маскирования  $O(k)$  в  $k$ -ой частотной полосе

$$O(k) = \begin{cases} P(k) - 14,5 - k, & k \in [1, k_M] \\ P(k) - 5,5, & k \in [k_M + 1, 28] \end{cases}, \text{ дБ} \quad (6.4)$$

Путем взятия свертки  $O(k)$  и функции маскирования получаются индивидуальные пороги маскирования

$$T_{ISM}(i, k) = 10 \cdot \lg \left( \sum_{i=1}^{28} 10^{\frac{O(i)}{10}} \cdot 10^{\frac{B(k-i)}{10}} \right) \quad (6.5)$$

Функция маскирования определяется как

$$B(n) = 15,81 + 7,5(n + 0,474) - 17,5\sqrt{1 + (n + 0,474)^2}, \text{ дБ}, \quad (6.6)$$

где  $n$  – значение центральной частоты полосы в барках.

От шкалы частот к шкале барков можно перейти по формуле

$$z(f) = 13 \arctg(0,00076f) + 3,5 \arctg[(f / 7500)^2] \quad (6.7)$$

Суммарный порог маскирования складывается из индивидуальных

$$T_{SM}(k) = 10 \cdot \lg \left( \sum_{i=1}^{28} 10^{0,1T_{ISM}(k, j)} \right), \text{ дБ} \quad (6.8)$$

Как правило, при моделировании критических полос слуха используется дерево вейвлет-пакетов представленное на рисунке 6.2, отражающее семейство полосовых фильтров, полосы пропускания которых растут с увеличением центральной частоты. Цифры, указанные на рисунке, означают граничные частоты групп полосовых фильтров.

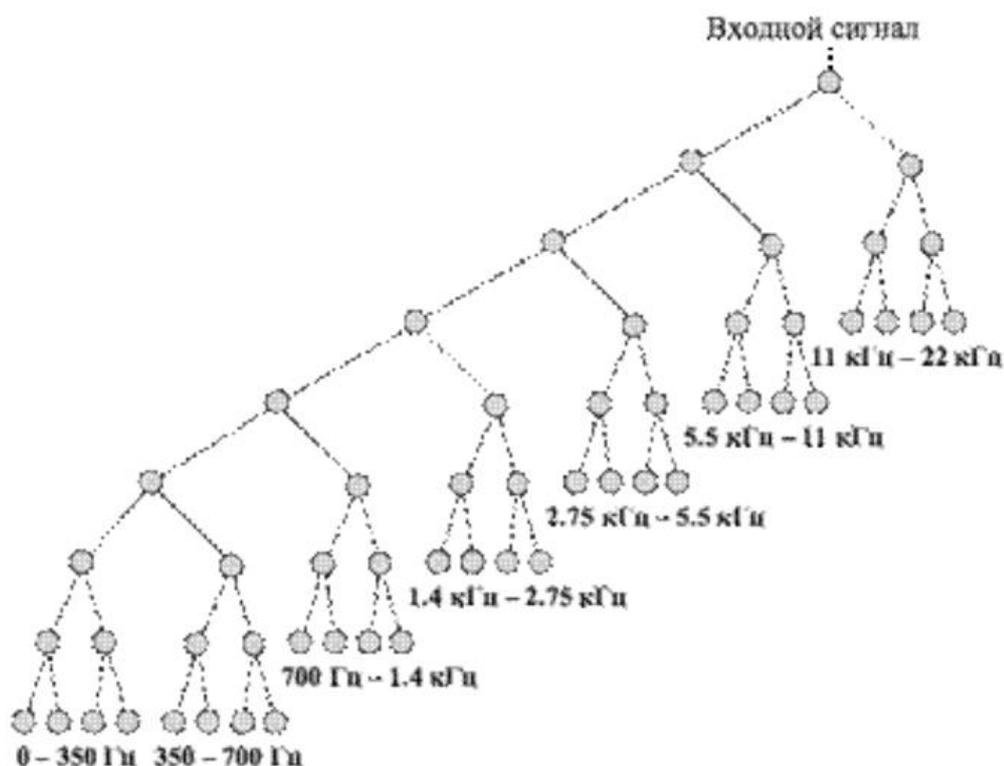


Рисунок 6.2 – Дерево вейвлетной декомпозиции

Структура дерева вейвлетных коэффициентов подбирается таким образом, чтобы подобрать оптимальную ширину критических полос пропускания.

Таким образом, *переквантование коэффициентов вейвлет-преобразования на основе порога маскирования, полученного из психоакустической информации* позволяет обеспечить большую гибкость сжатия звука.

### 6.3 Цель и задачи выполнения работы

*Цель:* Приобретение навыков применения алгоритмов спектрального анализа для сжатия звука.

*Задачи:*

- 1 Подготовка цифрового звука для сжатия.
- 2 Реализация алгоритма перцепционного кодирования звука, использующего вейвлетное преобразование.

## 6.4 Порядок выполнения работы

- 1 Прочитать теоретические предпосылки.
- 2 Выполнить практические задания.
- 3 Сделать выводы.

### *Задания*

1. Разработать программные модули для реализации задач:
  - a) вейвлет-пакетное преобразование одномерного цифрового массива (прямое и обратное);
    - b) извлечение психоакустической информации из значений вейвлетных коэффициентов, посредством расчета суммарной энергии в каждой частотной полосе  $k$ ;
    - c) нормировка вейвлет-коэффициентов в каждой частотной полосе  $k$ ;
    - d) расчет порога маскирования;
    - e) маскирование и переквантовка вейвлет-коэффициентов.
2. На основе разработанного программного средства и в системе Mathcad осуществить генерацию зашумленного звука и его сжатие:
  - a) смоделируйте звуковой сигнал (путем задания цифрового массива данных путем чтения из файла или по формуле согласно варианту, см. раздел 1, задание 3)
  - b) добавьте аддитивный или мультипликативный шум в полученный сигнал (используя сложение или поэлементное умножение на массив сгенерированных случайных чисел);
  - c) примените алгоритм перцепционного кодирования звука, использующего вейвлетное преобразование (задание 1a)-e));
  - d) сохраните сжатый звуковой сигнал в файл, прослушайте.

## 7 Фильтрация изображений

### 7.1 Пространственная фильтрация изображений

Под фильтрацией изображений в более узком смысле понимают помеховую фильтрацию, или фильтрацию изображений от шума и искажений. Искажения изображения, в частности, получаемого путем видеосъемки реального трехмерного объекта в природной обстановке, могут носить весьма сложный характер и здесь уже говорят уже не о фильтрации от шума, а о реставрации или реконструкции изображений, или методах улучшения изображений.

Множество подходов к улучшению изображений распадается на две большие категории: методы обработки в пространственной области (пространственные методы) и методы обработки в частотной области (частотные методы).

Пространственные методы основаны на прямом манипулировании пикселями изображения. Методы обработки в частотной области основываются на модификации спектра сигнала [4].

Процессы пространственной обработки описываются уравнением

$$g(x,y)=T[f(x,y)] \quad (7.1)$$

где  $f(x,y)$  – входное изображение;

$g(x,y)$  – обработанное изображение;

$T$  – оператор над  $f$ , определенный в некоторой окрестности точки  $(x,y)$ .

Окрестность вокруг точки  $(x,y)$  может быть определена в виде квадратной или прямоугольной области – подмножества изображения, centered в точке  $(x,y)$ .

Простейший случай оператора  $T$ , когда окрестность вырождается в один пиксель,  $T$  становится функцией градационного преобразования.

*Градационные преобразования.* Преобразование изображения в негатив с яркостями в диапазоне  $[0, L-1]$  осуществляется с использованием негативного преобразования, определяемого выражением

$$s=L-1-r, \quad (7.2)$$

где  $r$  – исходная яркость текущего пикселя;  
 $s$  – преобразованная яркость пикселя.

Логарифмическое преобразование выражается формулой

$$s=c \cdot \log(1+r) \quad (7.3)$$

где  $c$  – константа,  $r \geq 0$ .

Степенные преобразования имеют вид

$$s=c \cdot r^\gamma \quad (7.4)$$

Вычитание изображений

$$g(x,y)=f(x,y)-h(x,y) \quad (7.5)$$

*Пространственная фильтрация.* Для полутонового изображения  $A$  с интенсивностями его пикселей  $A(x,y)$  фильтрация производится при помощи операции дискретной свертки (взвешенного суммирования) матрицы интенсивностей пикселей с линейным фильтром [4,6]:

$$B(x, y) = h(i, j) \otimes \otimes A(x, y) = \sum_i \sum_j h(i, j) A(x-i, y-j), \quad (7.6)$$

где  $h$  – линейный фильтр (ядро фильтра), заданный вещественнозначной функцией, определенной на растре.

Обычно ядро фильтра  $h$  отлично от нуля только в некоторой окрестности  $N$  точки  $(0,0)$ . За пределами этой окрестности  $h(i,j)$  принимает нулевые или близкие к ним значения, которыми можно пренебречь. Суммирование производится по  $(i,j) \in N$ , и значение каждого пикселя  $V(x,y)$  определяется пикселями изображения  $A$ , которые лежат в окне  $N$ , центрированном в точке  $(x,y)$ . Ядро фильтра, заданное на прямоугольной окрестности  $N$ , может рассматриваться как матрица  $m$  на  $n$ , где длины сторон являются нечетными числами [4,6,26].

*Сглаживающие фильтры.* Применяются для шумоподавления и расфокусировки. Сглаживающие фильтры снижают локальную контрастность изображения, размывая его.

Примеры маски фильтра  $3 \times 3$ :

– однородный усредняющий фильтр

$$h_1^{low} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}; \quad (7.7)$$

– фильтр с увеличением веса центральной точки

$$h_1^{low} = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}; \quad (7.8)$$

– гауссовский фильтр с ядром:  $h(i, j) = (1/2\pi\sigma^2) \exp(-(i^2+j^2)/2\sigma^2)$ .

*Контрастоповышающие фильтры.* Контрастоповышающие фильтры производят эффект обратный сглаживающим фильтрам, являются фильтрами высоких пространственных частот. Ядро контрастоповышающего фильтра в точке  $(0, 0)$  имеет значение, большее 1, при общей сумме значений, равной 1. Пример контрастоповышающих фильтров [6]:

$$h_1^{contrw} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}; \quad h_2^{contr} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \quad (7.9)$$

*Разностные фильтры* – это линейные фильтры, задаваемые дискретными аппроксимациями по методу конечных разностей дифференциальных операторов. Данные фильтры являются фильтрами высоких пространственных частот, позволяют выделить перепады яркостей, например, границы объектов на изображении.

Примером дифференциального оператора является производная по  $x$ -координате  $d/dx$ , которая определена для непрерывных функций. Дискретными аналогами данного оператора являются фильтры Прюита (Prewitt) и Собеля (Sobel), позволяющие выделить вертикальные линии:

$$h_1^{Prew} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}; \quad h_2^{Sob} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (7.10)$$

Аналогично для выделения горизонтальных и косых линий:

$$h_3^{Prew} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}; \quad h_4^{Sob} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (7.11)$$

$$h_5^{Prew} = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}; \quad h_6^{Prew} = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}; \quad h_7^{Sob} = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}; \quad h_8^{Sob} = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad (7.12)$$

На рисунке 7.1 приведен пример применения пространственного фильтра Собеля в математическом пакете Mathcad.

```

X := READ_IMAGE("C:\Documents and Settings\Admin\Мои документы\222.jpg")

n := rows(X)    m := cols(X)
x := 2..n-3    y := 2..m-4
i := 0..2      j := 0..2
h := 1/4 * [ [-1 0 1]
             [-2 0 2]
             [-1 0 1] ]
R_{x,y} := sum_j [ sum_i (h_{i,j} * X_{x-i+2,y-j+2}) ]

WRITEBMP("C:\Documents and Settings\Admin\Мои документы\221.bmp") := B

```




"C:\Documents and Settings\Admin\Мои документы\222.jpg"
"C:\Documents and Settings\Admin\Мои документы\221.bmp"

Рисунок 7.1 –Пространственный фильтр Собеля в Mathcad

## 7.2 Частотная фильтрация изображений

Множество подходов к улучшению изображений распадается на две большие категории: методы обработки в пространственной области (пространственные методы) и методы обработки в частотной области (частотные методы).

Методы обработки в частотной области основываются на модификации сигнала, формируемого путем применения к изображению преобразования Фурье.

Прямое и обратное преобразование Фурье изображения, заданного функцией  $s(n,m)$  размером  $N$  на  $M$  определяются выражениями [4, 6]:

$$S(k,l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} s(n,m) \exp(-jn2\pi k/N-jm2\pi l/M), \quad (7.13)$$

$$S(k,l) = \sum_{n=0}^{N-1} \exp(-jn2\pi k/N) \sum_{m=0}^{M-1} s(n,m) \exp(-jm2\pi l/M), \quad (7.14)$$

$$s(n,m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} S(k,l) \exp(-jn2\pi k/N - jm2\pi l/M). \quad (7.15)$$

$$s(n,m) = \frac{1}{NM} \sum_{k=0}^{N-1} \exp(-jn2\pi k/N) \sum_{l=0}^{M-1} S(k,l) \exp(-jm2\pi l/M). \quad (7.16)$$

$$k=0..N-1; l=0..M-1.$$

Процедура фильтрации в частотной области состоит из следующих шагов:

- 1 Исходное изображение умножается на  $(-1)^{x+y}$ , чтобы центрировать его фурье-образ;
- 2 Вычисляется прямое ДПФ  $F(u,v)$  изображения, полученного после шага 1;
- 3 Функция умножается на функцию фильтра  $H(u,v)$ ;
- 4 Вычисляется обратное ДПФ от результата шага 3;
- 5 Выделяется вещественная часть результата шага 4;
- 6 Результат шага 5 умножается на  $(-1)^{x+y}$ .

*Некоторые основные фильтры. Фильтр – пробка.* Обнуляет среднее значение функции, т.е. величину  $F(u,v)$ .

$$H(u,v) = \begin{cases} 0, & \text{при } (u,v) = (M/2, N/2); \\ 1, & \text{в другом случае} \end{cases} \quad (7.17)$$

В результате применения – падение общей яркости и выделение контуров.

*Низкочастотные фильтры.* Ослабляет высокие частоты, одновременно пропуская низкие. Фильтр, обладающий противоположными свойствами называется высокочастотным.

*Виды фильтров* низких частот: идеальный фильтр, фильтр Баттерворта и Гауссов фильтр.

*Идеальный фильтр низких частот (ИФНЧ).* Обрезает все высокочастотные составляющие фурье-образа, находящиеся на большем расстоянии от начала координат (центрированного) преобразования, чем некоторое заданное расстояние  $D_0$ , имеет передаточную функцию:

$$H(u, v) = \begin{cases} 1, & \text{при } D(u, v) \leq D_0; \\ 0, & \text{при } D(u, v) > D_0 \end{cases}, \quad (7.18)$$

где  $D_0$  – заданная неотрицательная величина;

$D(u, v)$  обозначает расстояние от точки  $(u, v)$  до начала координат (центра частотного прямоугольника), задается формулой  $D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$ .

ИФНЧ имеют эффекты размывания и появления ложных контуров не имеют большого практического значения.

*Низкочастотный фильтр Баттерворта (БФНЧ).* Передаточная функция

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}, \quad (7.19)$$

где  $n$  – порядок фильтра.

В отличие от ИФНЧ не имеет разрыва, который точно устанавливает границу между пропускаемыми и обрезаемыми частотами.

*Гауссовы фильтры низких частот (ГФНЧ).* Передаточная функция

$$H(u, v) = e^{-D^2(u, v) / 2D_0} \quad (7.20)$$

*Высокочастотные фильтры.* Передаточная функция может быть получена при помощи следующего соотношения:

$$H_{hp}(u, v) = 1 - H_{lp}(u, v),$$

где  $H_{lp}(u, v)$  обозначает передаточную функцию соответствующего низкочастотного фильтра.

Идеальный фильтр высоких частот ИФВЧ имеет вид

$$H(u, v) = \begin{cases} 1, & \text{при } D(u, v) \leq D_0; \\ 0, & \text{при } D(u, v) > D_0 \end{cases}. \quad (7.21)$$

Высокочастотный фильтр Баттерворта (БФВЧ)

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}. \quad (7.22)$$

Гауссовы фильтры высоких частот (ГФВЧ)

$$H(u, v) = e^{-D^2(u, v) / 2D_0}. \quad (7.23)$$

Частотный фильтр Лапласа

$$H(u, v) = 1 + \left[ (u - M/2)^2 + (v - N/2)^2 \right]. \quad (7.24)$$

*Нерезкое маскирование.* Состоит в формировании резкого изображения путём вычитания из оригинала его сглаженной копии. В частотных терминах это означает, что имеет место высокочастотная фильтрация, которая достигается вычитанием из изображения результата его низкочастотной фильтрации, т.е.

$$f_{hp}(x, y) = f(x, y) - f_{lp}(x, y).$$

*Высокочастотная фильтрация с подъемом частотной характеристики*

$$f_{hp}(x, y) = Af(x, y) - f_{lp}(x, y).$$

Или тоже самое через высокочастотную фильтрацию

$$f_{hp}(x, y) = (A - 1)f(x, y) - f_{lp}(x, y).$$

Аналогично в частотной области

$$H_{hp}(v, u) = (A - 1) - H_{lp}(v, u). \quad (7.25)$$

*Фильтрация с усилением высоких частот*

$$.H_{hfc}(v, u) = a + bH_{lp}(v, u), \quad (7.26)$$

где  $a \geq 0$ ,  $b > a$ ,  $a = 0,25..0,5$ ;  $b = 1,5..2,5$ .

*Гомоморфная фильтрация.* Направлена на улучшение изображения путем одновременного сжатия яркостного диапазона и усиления контраста.

Передаточная функция  $H(u,v)$  задается так, чтобы фильтр по-разному воздействовал на низкочастотные и высокочастотные составляющие фурье-преобразования. Пример фильтра данного вида:

$$H(u,v) = (\gamma_H - \gamma_L) \left[ 1 - e^{-c(D^2(u,v)/D_0^2)} \right] + \gamma_L . \quad (7.27)$$

На рисунках 7.2-7.11 приведены результаты применения частотной фильтрации изображения, реализованной в программном средстве, позволяющие сравнить работу разных фильтров.

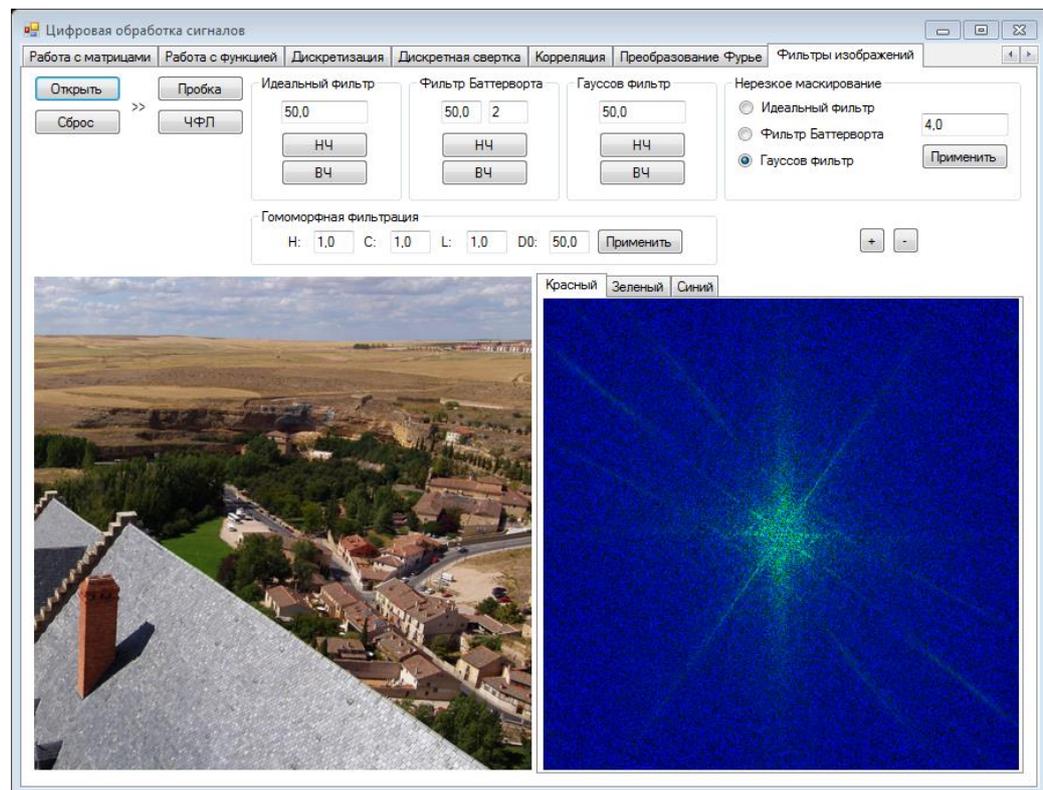


Рисунок 7.2 – Представление исходного изображения  
Первый фильтр – «Пробка» (рисунок 7.3).

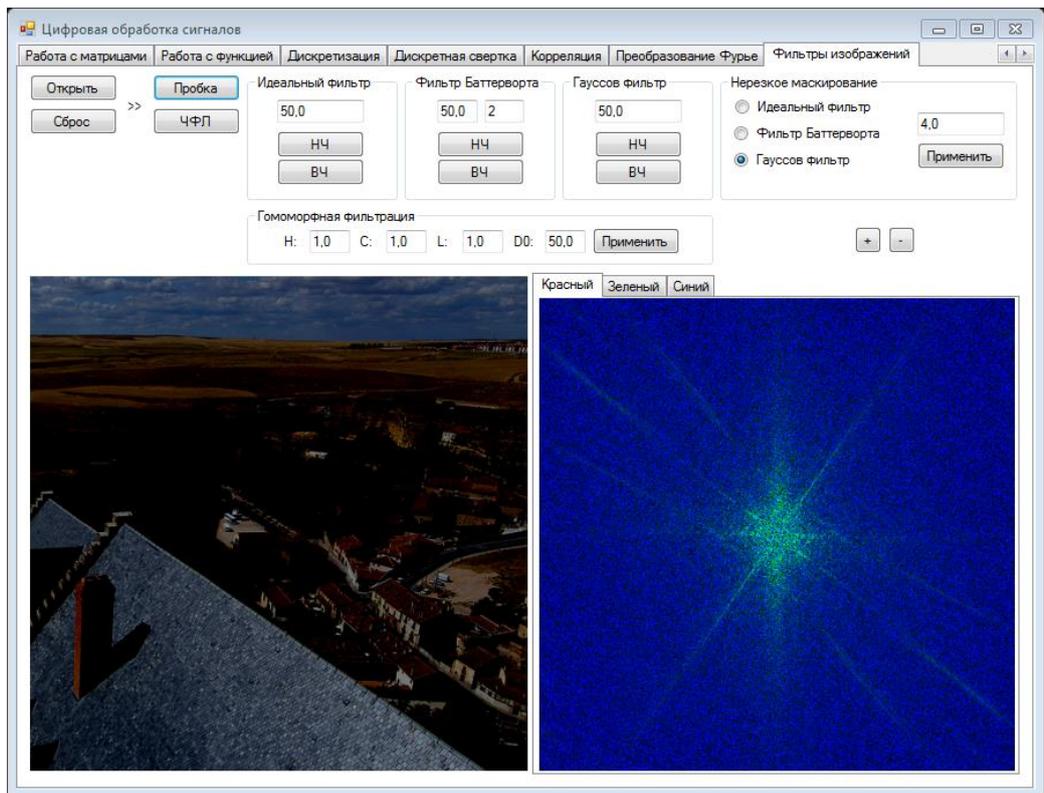


Рисунок 7.3 – Фильтр «Пробка»

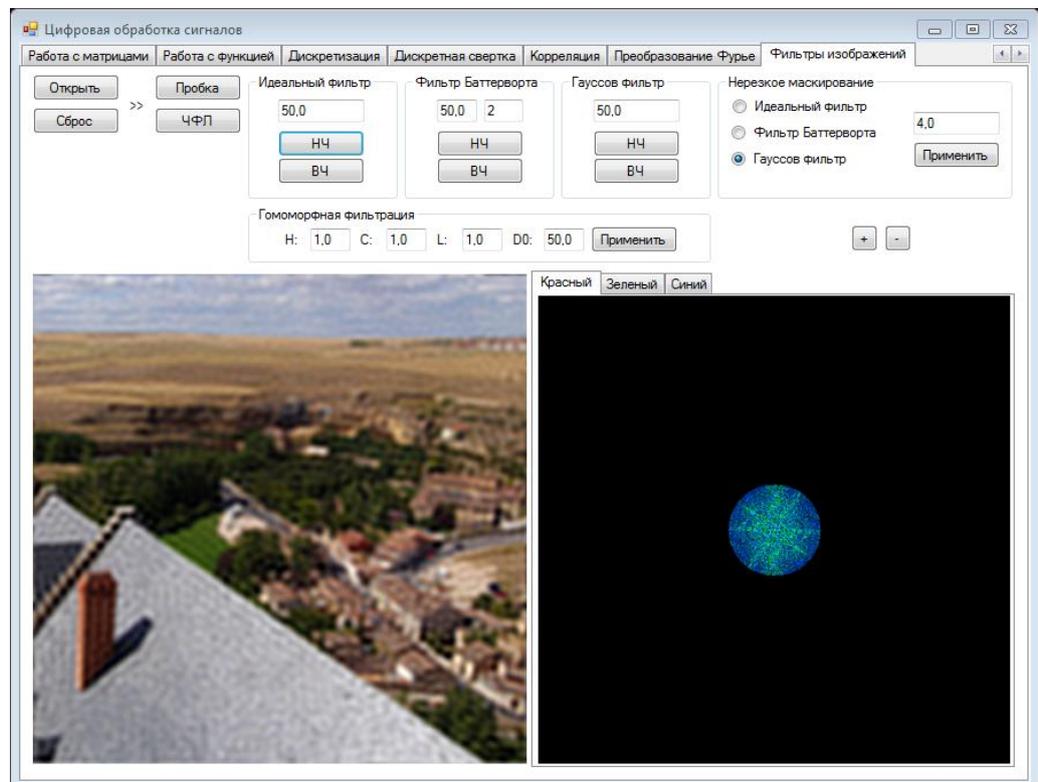


Рисунок 7.4 – Низкочастотный идеальный фильтр

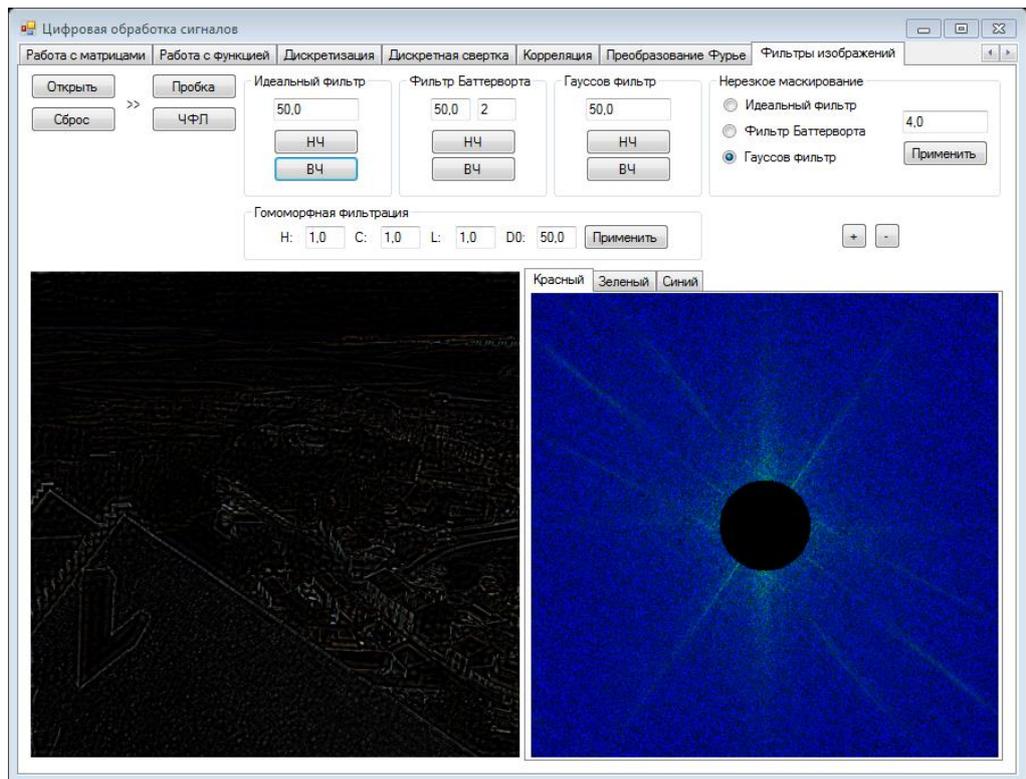


Рисунок 7.5 – Высокочастотный идеальный фильтр

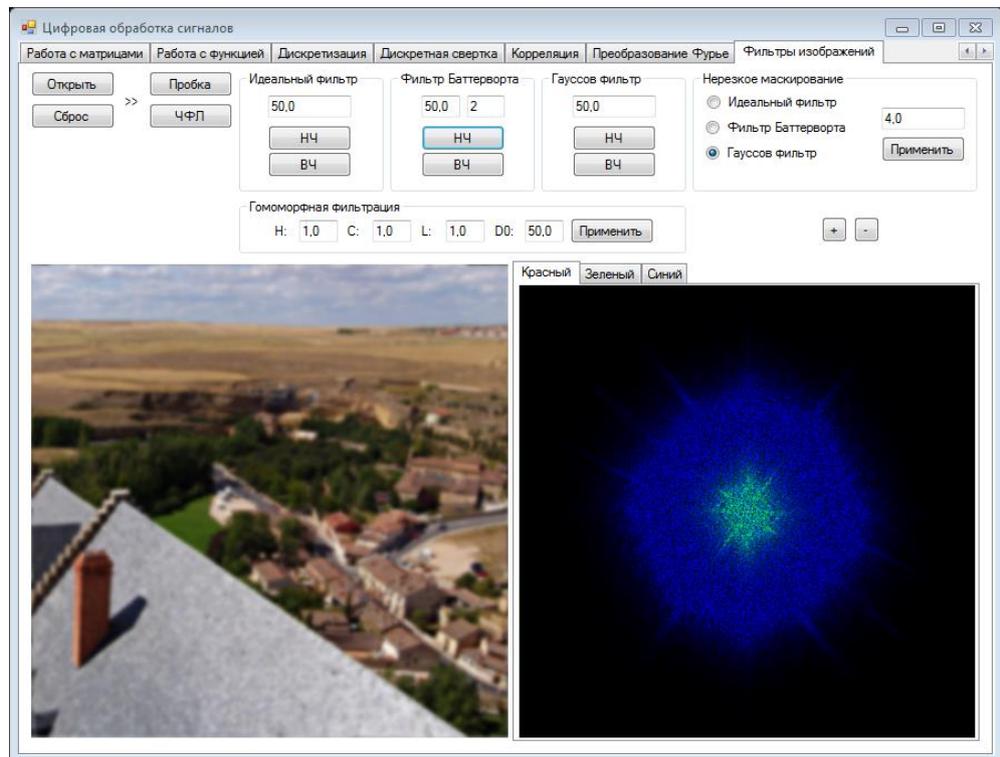


Рисунок 7.6 – Низкочастотный фильтр Баттерворта

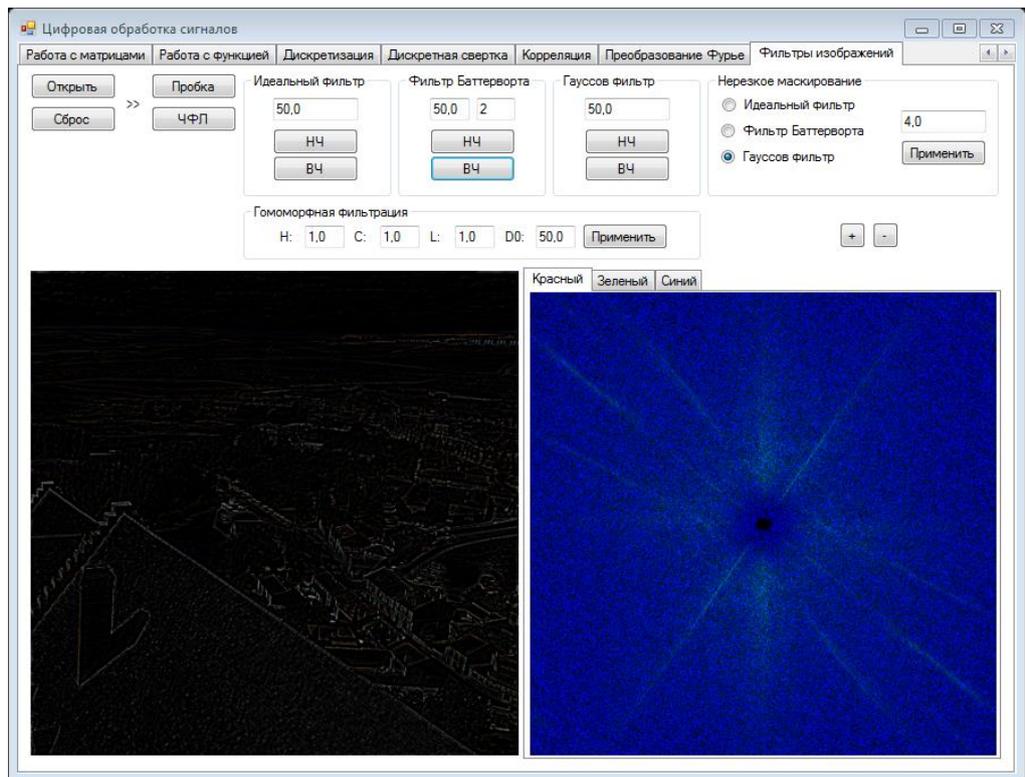


Рисунок 7.7 – Высокочастотный фильтр Баттлерорта

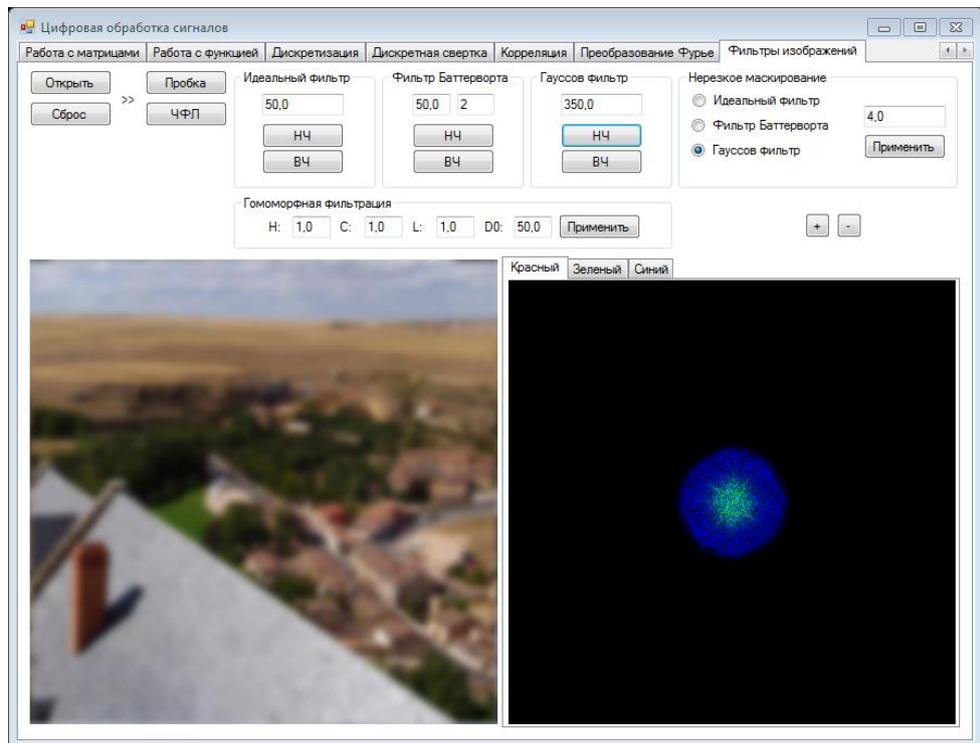


Рисунок 7.8 – Низкочастотный фильтр Гаусса



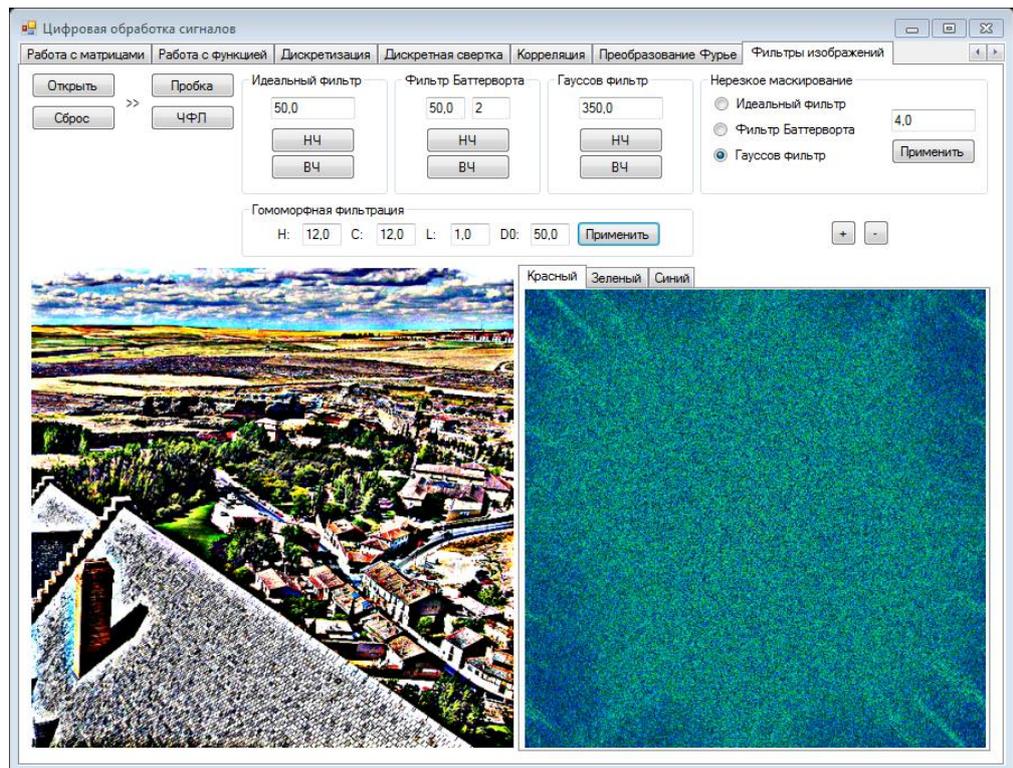


Рисунок 7.11 – Гомоморфная фильтрация

Таким образом, рассмотренные методы улучшения изображений, основанные на фильтрации в пространственной и частотной области, позволяют осуществлять предварительную обработку изображений и являются базовыми при решении сложных задач реставрации и конструкции изображений.

### 7.3 Цель и задачи выполнения работы

*Цель:* Формирование умений реализации пространственных методов улучшения изображений.

*Задачи:*

- 1 Изучить методы обработки изображений в пространственной области.
- 2 Применить градационные операции.
- 3 Реализовать примеры пространственной фильтрации изображений.
- 4 Рассмотреть методы обработки изображений в частотной области.
- 5 Реализовать примеры частотной фильтрации изображений.

## 7.4 Порядок выполнения работы

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Прочитать теоретические предпосылки.
- 3 Подготовить исходные данные к работе.
- 4 Выполнить практические задания.
- 5 Сделать выводы.

### *Задания*

- 1 Выполните операции (формулы 7.2-7.5) для рисунков 01.jpg, 31.jpg, 32.jpg.
- 2 Выполните операцию фильтрации по формулам 7.7-7.12 для рисунков 01.jpg, 31.jpg, 32.jpg, 33.jpg, 34.jpg.
- 3 Операцию 7.5 произвести над изображениями 33.jpg, 34.jpg.
- 4 Выполните процедуру фильтрации (формулы 7.13-7.16) для рисунков 01.jpg, 32.jpg, 33.jpg

### *Отчет и контрольные вопросы*

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные графики и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 На чем основаны методы обработки в пространственной и частотной области?
- 2 При помощи какой операции производится фильтрация?
- 3 Для чего применяются сглаживающие фильтры?
- 4 Фильтрами каких частот являются контрастоповышающие фильтры?
- 5 Какие фильтры задаются дискретными аппроксимациями по методу конечных разностей дифференциальных операторов?
- 6 Какие фильтры являются фильтрами высоких пространственных частот?
- 7 Что позволяют выделить перепады яркостей?

## 8 Статистическая регуляризация данных с очисткой от шумов

### 8.1 Модели шума

Основные источники шума на цифровом изображении – это процесс его получения (оцифровки) и процесс передачи. Искажения изображения, получаемого путем видеосъемки реального трехмерного объекта в природной обстановке, могут носить весьма сложный характер, зависящий от условий съемки (освещенность, туман, блики, тени, дождь, снег и т. п.), характеристик оптической системы (дисторсии, расфокусировки, замутненность линз и зеркал и т. п.), характеристик электронной регистрирующей аппаратуры, характеристик канала передачи, характеристик устройств оцифровки и еще многих других факторов [4,6, 25, 26].

Исходя из задачи восстановления исходных значений яркости незашумленного изображения, а также из того, что шумовая компонента каждого пикселя является заранее не известной случайной величиной, следует, что для решения данной задачи необходимо использовать ту или иную процедуру *статистического оценивания*. В связи с этим и основная идея фильтрации помех в изображениях заключается в том, что для оценки исходного значения каждого пикселя изображения используется не только значение самого данного пикселя, но и значения еще нескольких близких к нему пикселей, попадающих в так называемое "окно" или апертуру фильтра. При этом "близость" пикселей к оцениваемому понимается в геометрическом смысле.

Наиболее простыми для вычислительной реализации являются прямоугольные окна (апертуры) фильтрации. Типовая процедура оконной фильтрации предполагает, что окно фильтрации последовательно движется по входному изображению (например, алгоритм может обходить изображение: сверху вниз по строкам, слева направо в каждой строке), при этом в каждом положении окна происходит анализ всех пикселей, принадлежащих в данный момент окну, и на основе такого анализа центральному пикселю окна на выходном изображении присваивается результирующее значение.

Конструируя и исследуя оконные процедуры фильтрации изображений, следует оценивать наблюдаемое качество фильтрации по двум следующим основным позициям:

- способность фильтра удалять (отфильтровывать) с изображения шум;
- способность фильтра сохранять на изображении мелкоразмерные детали и форму контуров.

Данные требования противоречивы, поэтому в различных методах фильтрации приходится иметь дело лишь с различными вариантами компромисса между ними. Выбор конкретного фильтра шумов для реализации определяется тем, какое из требований является более важным в данной конкретной задаче, а также ограничениями, связанными с архитектурой и скоростью имеющихся вычислительных средств.

Описание поведения шума в пространственной области основано на статистических свойствах значений яркости компоненты шума в модели. Эти значения яркости могут рассматриваться как случайные величины, характеризующиеся функцией плотности распределения вероятностей. Наиболее часто встречаются при обработке изображений следующие функции плотности распределения вероятностей.

*Гауссов шум* (нормальный шум):

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}, \quad (8.1)$$

где  $x$  – случайная величина, представляющая значение яркости;

$\mu$  – среднее значение случайной величины;

$\sigma$  – среднеквадратичное отклонение.

*Шум Релея*:

$$p(x) = \begin{cases} \frac{2}{b}(x-a)e^{-(x-a)^2/b}, & x \geq a; \\ 0, & x < a \end{cases}. \quad (8.2)$$

Среднее и дисперсия имеют вид:

$$\mu = a + \sqrt{\pi b/4}, \sigma^2 = \frac{b(4-\pi)}{4}. \quad (8.3)$$

Шум Эрланга (гамма шум):

$$p(x) = \begin{cases} \frac{a^b x^{b-1}}{(b-1)!} e^{-ax}, & x \geq 0; \\ 0, & x < 0 \end{cases}. \quad (8.4)$$

Среднее и дисперсия имеют вид:

$$\mu = \frac{b}{a}, \sigma^2 = \frac{b}{a^2}. \quad (8.5)$$

Экспоненциальный шум:

$$p(x) = \begin{cases} ae^{-ax}, & x \geq 0; \\ 0, & x < 0 \end{cases} \quad a > 0. \quad (8.6)$$

Среднее и дисперсия имеют вид:

$$\mu = \frac{1}{a}, \sigma^2 = \frac{1}{a^2}. \quad (8.7)$$

Равномерный шум:

$$p(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b; \\ 0, & x < a, x > b \end{cases}. \quad (8.8)$$

Среднее и дисперсия имеют вид:

$$\mu = \frac{a+b}{2}, \sigma^2 = \frac{(b-a)^2}{12}. \quad (8.9)$$

Импульсный (биполярный) шум:

$$p(x) = \begin{cases} P_a, & x = a; \\ P_b, & x = b; \\ 0, & \text{в остальных случаях} \end{cases}. \quad (8.10)$$

Среднее и дисперсия имеют вид:

$$\mu = \frac{1}{a}, \sigma^2 = \frac{1}{a^2}. \quad (8.11)$$

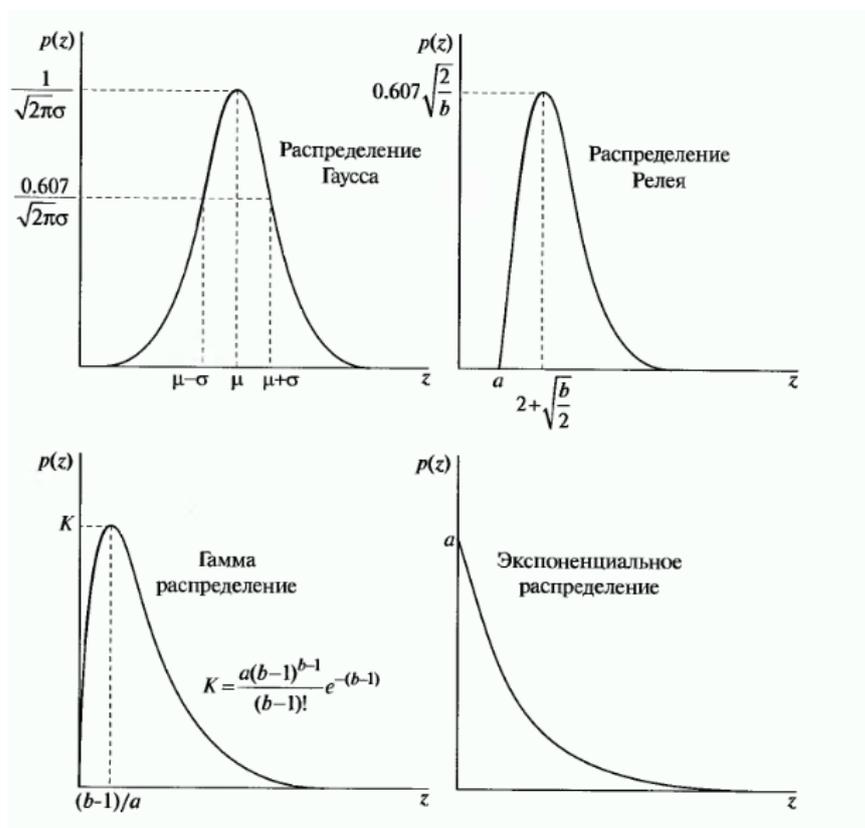


Рисунок 8.1 – Графики плотностей распределений вероятностей

Рассмотрение небольших участков изображения примерно постоянной яркости часто дает возможность оценить параметры функции плотности распределения вероятностей шума.

Пусть  $S$  – фрагмент изображения, например – полоса. Оценка может быть осуществлена по следующим простым статистическим формулам:

$$\mu = \sum_{x_i \in S} x_i p(x_i); \quad (8.12)$$

$$\sigma^2 = \sum_{x_i \in S} (x_i - \mu)^2 p(x_i), \quad (8.13)$$

где  $x_i$  – значения яркости элементов части изображения  $S$ ,

$p(x_i)$  – соответствующие нормализованные значения гистограммы.

Вид гистограммы определяет, какая из функций плотности распределения вероятностей является наиболее подходящей. Параметры  $a$  и  $b$  распределений можно найти из выражений для среднего и дисперсии, рассматривая их как уравнения для параметров  $a$  и  $b$ .

## 8.2 Очистка изображений от шумов

В тех случаях, когда на изображении присутствует только аддитивный шум, пространственная фильтрация является лучшим из возможных методов восстановления [4].

Существует множество пространственных фильтров, эффективных в разных ситуациях. Наибольшие возможности фильтрации представляют адаптивные фильтры, при этом увеличивается их сложность.

Поведение адаптивных фильтров изменяется в зависимости от статистических свойств изображения внутри области действия фильтра, которая определяется прямоугольной  $m \times n$  окрестностью  $S_{xy}$ .

*Адаптивные локальные фильтры подавления шума.* За основу при создании фильтра взяты среднее значение и дисперсия, поскольку их величины тесно связаны

с внешним видом изображения. Среднее значение дает меру средней яркости той области, по которой оно вычисляется, а дисперсия дает меру среднего отклонения в этой области.

Поведение фильтра определяется следующими условиями:

- Если дисперсия шума  $\sigma_{\eta}^2$  равна нулю, то отклик фильтра должен быть равен значению  $g(x,y)$ . Это отвечает тривиальному случаю нулевого шума.
- Если локальная дисперсия шума  $\sigma_L^2$  много больше  $\sigma_{\eta}^2$ , то значение отклика фильтра должно быть порядка  $g(x,y)$ . Большое значение локальной дисперсии обычно связано с наличием контуров, которые должны быть сохранены.
- Если обе дисперсии принимают значения одного порядка, то отклик фильтра должен быть равен среднему арифметическому значений в окрестности  $S_{xy}$ . Условие выполнено в том случае, когда статистические характеристики данной локальной области и изображения в целом совпадают, и локальный шум должен быть уменьшен, для чего используется простое усреднение.

Такой фильтр задается формулой:

$$f(x, y) = g(x, y) - \frac{\sigma_{\eta}^2}{\sigma_L^2} [g(x, y) - m_L]. \quad (8.14)$$

При реализации формулы нужно проверять условие  $\sigma_{\eta}^2 \leq \sigma_L^2$ , если оно не выполнено использовать 1 вместо их отношения.

*Адаптивные медианные фильтры.* Адаптивная медианная фильтрация, в отличие от медианной фильтрации помогает справиться с импульсным шумом, вероятности которого  $P_a$  и  $P_b$  превышают 0,2, а также лучше сохраняет детали в областях не искаженных шумом.

Пусть

$S_{xy}$  – прямоугольная окрестность точки  $(x,y)$ ;

$Z_{\min}$  – минимальное значение яркости в  $S_{xy}$ ;

$Z_{\max}$  – максимальное значение яркости в  $S_{xy}$ ;

$Z_{\text{med}}$  – медиана значений яркости в  $S_{xy}$ ;

$z_{xy}$  – значение яркости в точке  $(x,y)$ ;

$S_{max}$  – максимальный допустимый размер в  $S_{xy}$ .

Алгоритм адаптивной медианной фильтрации состоит из двух ветвей А и Б:

*Ветвь А:*  $A1 = z_{med} - z_{min}$ ;

$A2 = z_{med} - z_{max}$ ;

если  $A1 > 0$  и  $A2 < 0$ , перейти к ветви Б;

иначе увеличить размер окрестности;

если размер окрестности  $\leq S_{max}$  повторить ветвь А;

иначе результат равен  $z_{xy}$ .

*Ветвь Б:*  $B1 = z_{xy} - z_{min}$ ;

$B2 = z_{xy} - z_{max}$ ;

если  $B1 > 0$  и  $B2 < 0$ , результат равен  $z_{xy}$ ;

иначе результат равен  $z_{med}$ .

Последняя операция соответствует действию обычного медианного фильтра.

После получения значения обрабатываемого элемента изображения, центр окрестности смещается в позицию следующего элемента и алгоритм повторяется.

Применение алгоритма обеспечивает удаление биполярного импульсного шума, сглаживание шумов других типов, сведение к минимуму таких искажений как чрезмерное утолщение или утончение границ объектов.

### **8.3 Цель и задачи выполнения работы**

*Цель:* Формирование умения применять адаптивные алгоритмы фильтрации изображения.

*Задачи:*

- 1 Добавление шумов в изображения.
- 2 Реализация алгоритма восстановления изображения.

## 8.4 Порядок выполнения работы

- 1 Прочитать теоретические предпосылки.
- 2 Выполнить практические задания.
- 3 Сделать выводы.

### *Задания*

- 1 Разработать программные модули для реализации задач:
  - a) зашумление данных;
  - b) адаптивная локальная фильтрация подавления шума;
  - c) адаптивная медианная фильтрация.
- 2 На основе разработанного программного средства осуществить фильтрацию изображения.
  - a) загрузить изображение, добавьте искажения;
  - b) применить алгоритмы восстановления изображения (задание 1b)-с));
  - c) сохранить восстановленное изображение в файл, просмотреть.

## 9 Аппроксимация сигналов и функций

### 9.1 Интерполяция

При проведении анализа различных физических явлений, технологических процессов результаты эксперимента обычно представляются в виде табличной зависимости функции  $y(x)$  [6].

Для аппроксимации табличной зависимости функции используются многочлены, ряды Фурье, экспоненциальные, логарифмические, степенные и другие элементарные функции. При обработке сигналов чаще применяется преобразование Фурье или вейвлет-преобразование.

Простейший способ *интерполяции* — *кусочно - линейная*, требующая минимальных требований на гладкость функции  $f(x)$ . При таком способе интерполяции соседние точки  $(x_i; f_i)$  и  $(x_{i+1}; f_{i+1})$  соединяют отрезками прямых [12]:

$$\varphi(x) = \frac{f_{i+1}(x - x_i) + f_i(x_{i+1} - x)}{x_{i+1} - x_i}, \quad x \in [x_i; x_{i+1}]. \quad (9.1)$$

Часто используется *кусочно - полиномиальная интерполяция*. Например, эрмитов кубический интерполянт, кубический сплайн.

В системе Mathcad для линейной интерполяции может использоваться функция  $\text{linterp}(X, Y, x)$ , где  $X$  и  $Y$  – вектора узловых точек. Функция  $\text{linterp}(X, Y, x)$  возвращает значение функции при её линейной интерполяции по заданным аргументам  $x$  [15].

Аппроксимация функции  $y(x)$  интерполяционным многочленом Лагранжа  $n$ -й степени  $Y(x)$  может быть проведена по формуле:

$$L_n(x) = \sum_{i=0}^n y_i \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}. \quad (9.2)$$

Сплайновая интерполяция [6,10,13,15]. Сплайн - кусочный многочлен степени  $K$  с непрерывной производной степени  $K - 1$  в точках соединения сегментов.

В системе Mathcad при выполнении сплайновой интерполяции по узловым точкам функции  $(X_i; Y_i)$  сначала вычисляется вектор  $S$  вторых производных входной функции  $y(x)$  по одной из подпрограмм:

$S := \text{cspline}(X, Y)$  – возвращает вектор  $S$  вторых производных при приближении в узловых точках к кубическому полиному;

$S := \text{pspline}(X, Y)$  – возвращает вектор  $S$  при приближении в узловых точках к параболической кривой;

$S := \text{lspline}(X, Y)$  – возвращает вектор  $S$  при приближении в узловых точках к прямой.

По значениям вектора  $S$  функцией  $\text{interp}(S, X, Y, x)$  вычисляются значения аппроксимирующей функции по аргументам  $x$ .

Пример интерполяции в системе Mathcad приведен на рисунке 9.1.

$$f(z) = \sum_i y_i \cdot \prod_j \text{if} \left( i = j, 1, \frac{z - x_j}{x_i - x_j} \right) \quad \leftarrow \text{Формула Лагранжа}$$

$$S = \text{cspline}(x, y) \quad f_s(z) = \text{interp}(S, x, y, z) \quad \leftarrow \text{Кубическая сплайновая интерполяция}$$

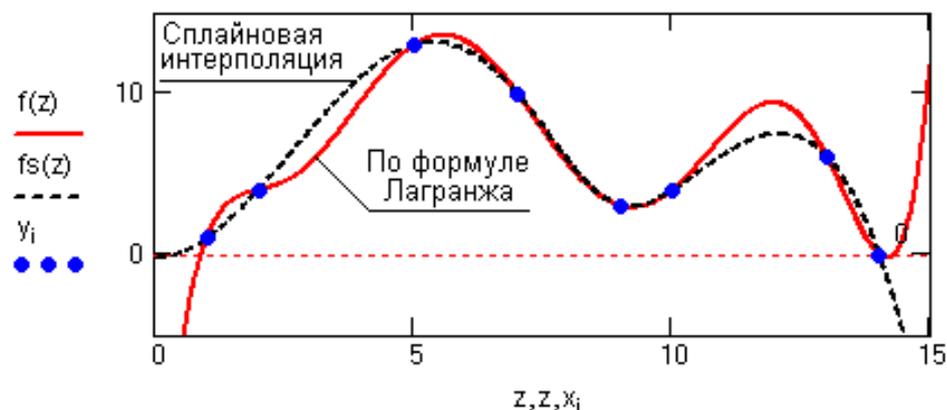


Рисунок 9.1 – Сплайновая интерполяция и интерполяция по Лагранжу

Наиболее точную интерполяцию финитных сигналов при дискретизации данных с равномерным шагом по аргументу обеспечивает *спектральный метод*. При условии, что в спектре сигнала не содержится частотных составляющих,

превышающих частоту Найквиста:  $f_N = 1/2\Delta t = F/2$  (для центрального периода от  $-f_N$  до  $f_N$ ). Частота дискретизации сигнала должна быть минимум в два раза выше максимальной частотной составляющей в спектре сигнала ( $F = 1/\Delta t \geq 2f_{\max}$ ). Обратное преобразование Фурье такого спектра, должно восстанавливать непрерывный сигнал, равный исходному аналоговому сигналу  $s(t)$ .

## 9.2 Аппроксимация эмпирических данных

Аппроксимация эмпирических данных предполагает существование детерминированной связи  $y(x)$  между регулярными составляющими двух числовых рядов  $x$  и  $y$  на статистически значимом уровне, достаточном для ее выявления на уровне случайных составляющих [6,10,11,15]. Определение закономерности зависит от:

- выбора меры близости и метода построения приближения (параметров математической модели);
- выбора подходящего класса функции аппроксимации;
- оптимизации порядка модельной функции или числа членов ряда аппроксимирующего выражения.

*Мера приближения.* Наиболее распространен критерий наилучшего приближения в виде минимума **квадратичной меры**, несмещенной оценкой является дисперсия остатков:

$$D = \{ \sum_k [y_k - \varphi(x_k)]^2 \} / (k-m), \quad (9.3)$$

где  $m$  – количество параметров в функции приближения;

$(k-m)$  – число степеней свободы.

*Аппроксимирующая функция* может быть математической функцией любого типа, линейной комбинацией различных функций или функциональным рядом из степенных, тригонометрических и любых других функций. Свойства функции

должны соответствовать сущности изучаемого явления, например такие свойства, как область определения переменных и производных, асимптоты, минимумы и максимумы.

*Порядок модели.* Для функциональных рядов порядок модели (или степень ряда) определяется значением меры приближения. Наилучшим является приближение, у которого мера приближения близка к дисперсии шума.

$$\{ \sum_k [y_k - \varphi(x_k)]^2 \} / (k-m) \approx \sum_k \sigma_k^2 / k. \quad (9.4)$$

При отсутствии информации о дисперсии шумов оптимальный порядок модели может определяться сравнением по критерию Фишера значимости различия дисперсии остатков каждого нового порядка с предыдущим. При увеличении порядка модели (начиная с 1-го) значимость различия дисперсий сначала является довольно высокой, постепенно уменьшается, и в области оптимальных порядков становится малозначимой, затем снова начинает увеличиваться.

Оптимальный порядок модели при нормальном распределении шума может устанавливаться и по минимуму дисперсии остатков (рисунок 9.2) [6].

Одномерная полиномиальная аппроксимация данных в векторе  $Y$  полиномом с произвольной степенью  $n$  и с произвольными координатами отсчетов в векторе  $X$  в Mathcad выполняется функциями:

- `regress(X,Y,n)` – вычисляет вектор  $S$  для функции `interp(...)`, в составе которого находятся коэффициенты  $c_i$  полинома  $n$ -й степени;
- `interp(S,X,Y,x)` – возвращает значения функции аппроксимации по координатам  $x$ .

Функция `interp(...)` реализует вычисления по формуле:

$$f(x) = c_0 + c_1 \cdot x^1 + c_2 \cdot x^2 + \dots + c_n \cdot x^n \equiv \sum_i c_i \cdot x^i.$$

Значения коэффициентов  $c_i$  могут быть извлечены из вектора  $S$  функцией

- `submatrix(S, 3, length(S), 0, 0)`.

*Оценка качества приближения.* Для оценки качества математической модели

эмпирической зависимости используется коэффициент детерминации (Adjusted R<sup>2</sup>):

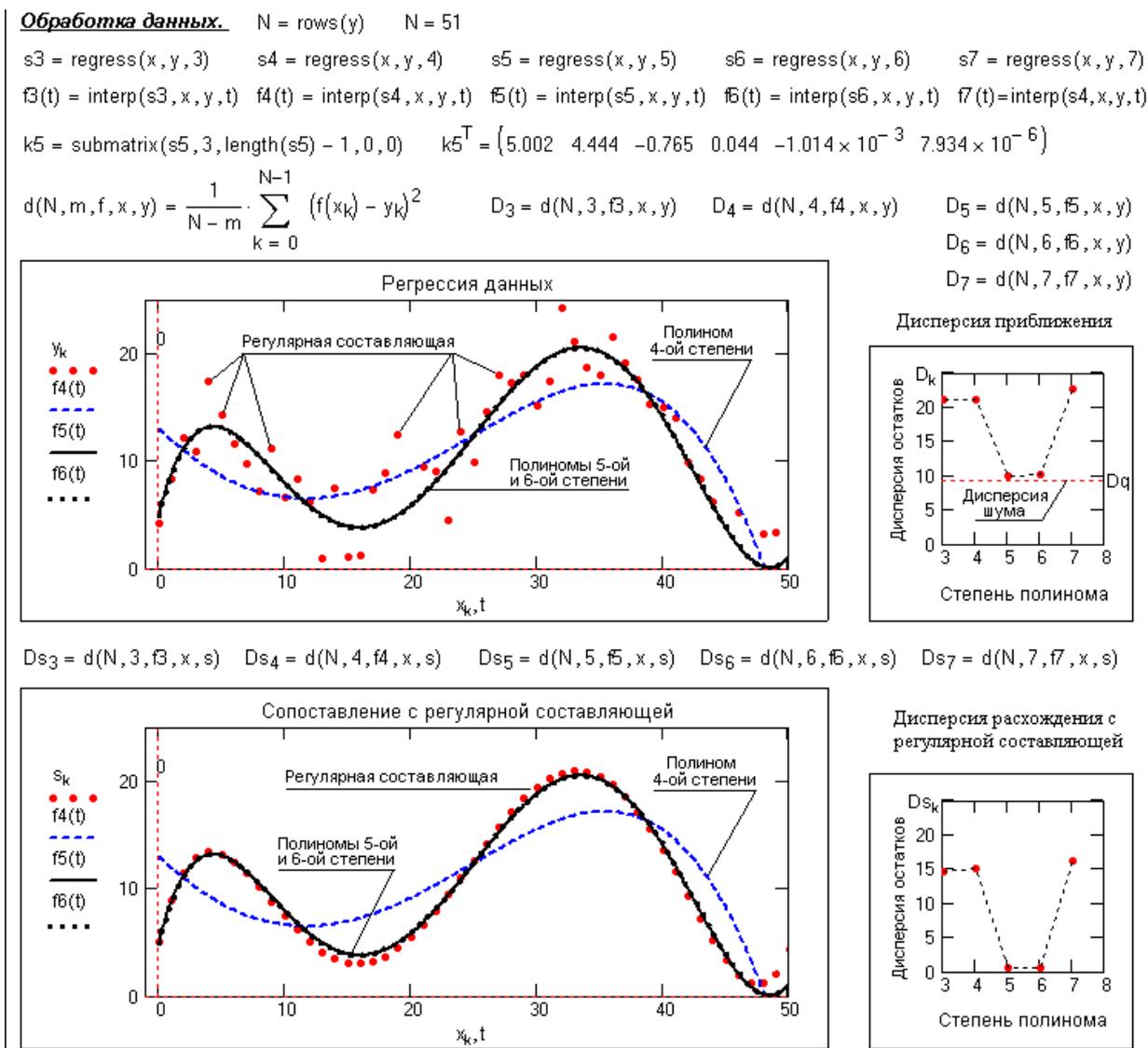


Рисунок 9.2 – Оптимальная полиномиальная аппроксимация

$$\text{Adjusted } R^2 = D_\varphi / D_y = 1 - D_o / D_y, \quad (9.5)$$

где  $D_\varphi$  – дисперсия функции приближения,

$D_y$  – дисперсия данных,

$D_o$  – дисперсия остатков.

Чем ближе к 1 значение коэффициента детерминации, тем выше качество аппроксимации.

Таким образом, математический пакет Mathcad предоставляет широкие возможности для интерполяции и аппроксимации эмпирических данных на основе различных моделях функций.

### **9.3 Цель и задачи выполнения работы**

*Цель:* Формирование умений аппроксимации данных в динамической и спектральной области

*Задачи:*

- 1 Интерполяция одномерного цифрового сигнала.
- 2 Регрессионный анализ одномерного цифрового сигнала.

### **9.4 Порядок выполнения работы**

- 1 Ознакомиться с целью, задачами и содержанием лабораторной работы.
- 2 Прочитать теоретические предпосылки.
- 3 Подготовить исходные данные к работе.
- 4 Выполнить практические задания.

*Задания*

1 Внести погрешности в исходные данные Вашего варианта задания. Диапазон задаваемой погрешности (0.1-5)%.

2 Выполнить интерполяцию табличных данных (исходных и с погрешностью) в системе Mathcad используя кубические сплайны, спектральную интерполяцию с шагом дискретизации: 1,2,3 с контролем частотного спектра сигналов.

3 Обосновать оптимальный метод полиномиальной интерполяции с учетом оценки качества приближения.

4 Получить аппроксимирующую модель с помощью регрессионного анализа в системе Mathcad для табличных данных (исходных и с погрешностью), определить оптимальную степень многочлена с учетом дисперсии приближения.

- 5 Сделать выводы.

### Отчет и контрольные вопросы

Отчет составляется в текстовом редакторе, содержит исходные данные и результаты выполнения заданий, созданные графики и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

- 1 Для чего применяется децимация и интерполяция в современных системах обработки данных?
- 2 Какими методами решается задача аппроксимации функции?
- 3 Чем отличается понятие аппроксимации функции от интерполяции функции?
- 4 Какие требования накладываются на интерполирующую функцию?
- 5 В каком случае более эффективным является спектральный метод интерполяции?
- 6 Как определить оптимальную степень аппроксимирующего многочлена?
- 7 Какие факторы влияют на определение функциональной зависимости эмпирических данных?

### Варианты заданий

Значения  $x_i = i \times 0,1$ ;  $i = 1, 2, \dots, 15$  одинаковые для всех вариантов

Таблица 9.1– Варианты функций

$i$	Значения $Y_i = y(X_i)$					
	Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5	Вариант 6
1	2	3	4	5	6	7
1	2,05	2,09	2,02	1,99	2,23	2,07
2	1,94	2,05	1,98	2,03	2,29	2,17
3	1,92	2,19	1,67	2,20	2,27	2,21
4	1,87	2,18	1,65	2,39	2,62	2,31
5	1,77	2,17	1,57	2,19	2,72	2,10

Продолжение таблицы 9.1

1	2	3	4	5	6	7
6	1,88	2,27	1,42	2,61	2,82	2,09
7	1,71	2,58	1,37	2,35	3,13	2,12
8	1,60	2,73	1,07	2,60	3,49	1,63
9	1,56	2,82	0,85	2,55	3,82	1,78
1	2	3	4	5	6	7
10	1,40	3,04	0,48	2,49	3,95	1,52
11	1,50	3,03	0,35	2,50	4,22	1,16
12	1,26	3,45	-0,30	2,52	4,48	1,07
13	0,99	3,62	-0,61	2,44	5,06	0,85
14	0,97	3,85	-1,20	2,35	5,50	0,56
15	0,91	4,19	-1,39	2,26	5,68	0,10
<i>i</i>	Вариант 7	Вариант 8	Вариант 9	Вариант 10	Вариант 11	Вариант 12
1	2,18	-0,10	-0,16	2,09	2,15	0,10
2	2,43	-0,21	0,01	2,31	2,41	-0,01
3	2,40	0,01	0,10	2,72	2,58	-0,19
4	2,43	0,05	0,16	2,77	2,84	-0,11
5	2,65	-0,13	0,05	2,78	3,28	-0,31
6	2,75	-0,23	0,35	2,97	3,46	-0,78
7	2,67	-0,21	0,19	3,00	4,02	-0,64
8	2,66	-0,43	0,50	3,51	4,11	-0,85
9	2,63	-0,57	0,74	3,43	4,61	-1,18
10	2,75	-0,44	1,03	3,58	5,03	-1,39
11	2,41	-0,44	1,06	3,58	5,34	-1,79
12	2,24	-0,83	1,49	3,54	5,86	-2,02
13	2,12	-0,78	1,79	3,82	6,33	-2,48
14	1,74	-0,81	2,03	3,90	6,81	-2,93
15	1,57	-1,06	2,22	3,77	7,21	-3,26

Продолжение таблицы 9.1

1	2	3	4	5	6	7
<i>i</i>	Вариант 7	Вариант 8	Вариант 9	Вариант 10	Вариант 11	Вариант 12
1	0,17	0,80	0,04	0,08	-0,02	0,14
2	0,07	0,29	0,47	0,14	0,44	0,23
3	0,17	0,52	0,78	0,37	0,51	0,44
4	0,05	0,77	1,01	0,36	0,67	0,54
5	0,12	0,93	1,19	0,44	0,69	0,72
6	0,00	1,20	1,60	0,48	1,04	3,76
7	0,01	1,20	1,93	0,27	1,1	0,37
8	-0,05	1,35	2,22	0,39	1,3	0,64
9	-0,21	1,39	2,50	0,50	1,7	357
10	-0,50	1,48	3,01	0,48	2,0	3,44
11	-0,50	1,52	3,22	0,69	2,1	3,41
12	-0,86	1,71	3,71	0,50	2,4	0,30
13	-1,24	1,72	4,23	0,31	2,90	-0,0
14	-1,47	1,87	4,78	0,37	3,50	-0,03
15	-1,79	1,86	5,27	0,43	3,99	-0,47
<i>i</i>	Вариант 19	Вариант 20	Вариант 21	Вариант 22	Вариант 23	Вариант 24
1	-1,86	- 1,65	-1,89	-1,84	-1,92	-1,90
2	-1,95	- 2,00	-2,07	-1,98	-1,60	-1,80
3	-2,12	- 1,87	-2,30	-1,72	-1,57	-1,82
4	-2,06	- 1,89	-2,26	- 1,58	-1,41	-1,86
5	-2,15	- 1,75	-2,34	- 1,69	-1,36	-1,83
6	-2,00	- 1,59	-2,66	- 1,59	-0,97	-2,00
7	-2,12	-1,44	-2,88	-1,58	-0,59	-2,01
8	-2,31	-1 ,61	-2,85	-1,64	-0,71	-2,05
9	-2,29	- 1,00	-3,16	- 1,55	-0,15	-2,46

Продолжение таблицы 9.1

1	2	3	4	5	6	7
10	-2,57	- 1,17	-3,49	- 1,35	0,01	-2,68
11	-2,56	-0,87	-3,88	- 1,33	0,22	-2,85
12	-2,86	-0,47	-4,22	- 1,47	0,63	-2,98
13	-2,85	-0,33	-4,45	- 1,50	1,07	3,30
14	-3,03	-0,00	-4,99	- 2,65	1,42	-3,40
15	-3,25	0,34	-5,36	- 1,65	1,68	-3,90
<i>i</i>	Вариант 25	Вариант26	Вариант27	Вариант 28	Вариант 29	Вариант 30
1	-1,80	-1,65	-1,88	- ,01	-4,13	-3,97
2	-1,66	-1,64	-1,69	- ,06	-4,11	-4,07
3	-1,36	-1,41	-1,52	- ,88	-3,87	-4,04
4	-1,41	-0,91	-1,55	-3,98	-3,74	-4,30
5	-1,13	-0,63	-1,16	-4,36	-3,85	-4,27
6	-0,82	-0,34	-1,27	-4,18	-3,71	-4,54
7	-0,74	-0,12	-1,23	-4,16	-3,53	-4,79
8	-0,76	0,25	-1,36	-4,51	-3,56	-5,07
9	-0,64	0,64	-1,26	-4,53	-3,19	-5,30
10	-0,46	0,96	-1,47	-4,38	-3,04	-5,51
11	-0,30	1,50	-1,72	-4,76	-2,83	-5,83
12	-0,27	1,77	-1,76	-4,66	-2,54	-6,06
13	-0,22	2,24	-2,00	-4,82	-2,41	-6,40
14	-0,11	2,93	-2,03	-4,77	-1,97	-6,83
15	-0,02	3,17	-2,35	-5,12	-1,78	-7,54

## 10 Обнаружение и распознавание объектов изображений

### 10.1 Постановка задачи распознавания. Основные определения и понятия

*Распознавание образов* – раздел кибернетики, развивающий теоретические основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т.п. объектов, которые характеризуются конечным набором некоторых свойств и признаков [5,8,25].

*Образ, класс* – классификационная группировка в системе классификации, объединяющая (выделяющая) определенную группу объектов по некоторому признаку.

В целом проблема распознавания образов состоит из двух частей: обучения и распознавания.

*Обучение* – это процесс, в результате которого система постепенно приобретает способность отвечать нужными реакциями на определенные совокупности внешних воздействий, а адаптация – это подстройка параметров и структуры системы с целью достижения требуемого качества управления в условиях непрерывных изменений внешних условий.

*Распознавание образов* — это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы несущественных данных.

Классически *постановка задачи распознавания образов* формулируется в виде [8]: дано множество объектов, относительно которых необходимо провести классификацию. Множество представлено подмножествами, которые называются классами. Заданы: информация о классах, описание всего множества и описание информации об объекте, принадлежность которого к определенному классу неизвестна. Требуется по имеющейся информации о классах и описании объекта установить – к какому классу относится этот объект.

Под *объектом* понимается некоторое его приближенное описание, в виде набора характерных свойств (признаков). Основное назначение описаний (образов

объектов) – это их использование в процессе установления соответствия объектов, осуществляемого путем сравнения (сопоставления). Задачей распознавания является определение «скрытой» принадлежности объекта к тому или иному классу путем анализа вектора значений наблюдаемых признаков.

*Классификация методов обнаружения и распознавания образов.*

Различные авторы дают различную *типологию методов распознавания образов*. Одни авторы различают параметрические, непараметрические и эвристические методы, другие – выделяют группы методов, исходя из исторически сложившихся школ и направлений в данной области. Одна из типологий методов распознавания образов:

- 1 методы, основанные на принципе разделения;
- 2 статистические методы;
- 3 методы, построенные на основе “потенциальных функций”;
- 4 методы вычисления оценок (голосования);
- 5 методы, основанные на исчислении высказываний, в частности на аппарате алгебры логики.

Исторически выделяют два подхода к распознаванию: детерминистский и статистический, хотя чаще всего строго различить их не удастся. *Детерминистский* подход включает различные методы: эмпирические, эвристические, в основе которых лежат здравый смысл, моделирование действий, осуществляемых мозгом человека; математически формализованные, например, основанные на модели порождения объектов (реализаций) того или иного образа. При этом используется различный математический аппарат (математическая логика, теория графов, топология, математическая лингвистика, математическое программирование и др.).

*Статистический* подход опирается на фундаментальные результаты математической статистики (теория оценок, последовательный анализ, стохастическая аппроксимация, теория информации).

Многие методы распознавания, появившиеся как детерминистские, получили в дальнейшем статистическое обоснование.

Одна из классификаций методов распознавания образов основана на видах операций [8]:

*Интенциональные* методы распознавания образов – методы, основанные на операциях с признаками. *Экстенциональные* методы распознавания образов – методы, основанные на операциях с объектами.

*Интенциональные* методы.

- Методы, основанные на оценках плотностей распределения значений признаков (или сходства и различия объектов).
- Методы, основанные на предположениях о классе решающих функций.
- Логические методы.
- Лингвистические (структурные) методы.

*Экстенциональные* методы.

- Метод сравнения с прототипом.
- Метод k-ближайших соседей.
- Алгоритмы вычисления оценок (голосования) АВО.
- Коллективы решающих правил.

## **10.2 Математические основы обнаружения и распознавания сигналов**

*Обнаружение объектов.* Обнаружение (выявление) образов объектов в искажённом шумами и помехами сигнале может быть определено в виде процедуры проверки определенного условия сравнения двух чисел – результата преобразования анализируемого изображения и определенного порогового значения [6]:

$$L[A(x, y)] \geq \Pi[A(x, y)]. \quad (10.1)$$

Здесь  $L[.]$  – оператор преобразования исходного сигнала,  $\Pi[.]$  – оператор формирования порогового значения. Решение о наличии объекта принимается в случае выполнения условия. *Качество обнаружения* характеризуется *вероятностью выполнения условия* при наличии объекта в анализируемом сигнале [5].

Конкретный вид операторов  $L[.]$ ,  $\Pi[.]$  и качество обнаружения зависят от наличия априорных сведений об ожидаемых объектах, шумах, помехах и искажениях. Основой для определения оптимальных параметров операторов является *теория статистических решений*.

Классической считается процедура обнаружения сигнала известной формы на фоне аддитивных нормальных шумов с нулевым средним значением. В этом случае в качестве *операторов преобразования* выступают *корреляционные интегралы*, вычисляемые с использованием заданного описания известного изображения объекта и типового шума соответственно. Идентификация заключается в *сравнении сигнала с эталонами заданного класса*. Решение об объекте выносится по наилучшему совпадению.

*Способ прямого сравнения объекта с эталонным изображением*. Пусть  $S(x, y)$  – исходный сигнал,  $F(x, y)$  – эталонный сигнал. Алгоритм прямого сравнения имеет вид:

$$T = \sum_x \sum_y (S(x, y) - F(x, y))^2 \leq D, \quad (10.2)$$

где  $D$  – порог различия.

Способ прост и может быть легко реализован. Однако, при наличии в реальных условиях дестабилизирующих факторов надёжность способа невелика. При большем значении порога  $D$  условию (9.2), могут удовлетворять различные объекты и могут возникнуть ошибки, связанные с неправильной идентификацией объекта (ошибки первого рода). При уменьшении  $D$  могут возникнуть ошибки типа пропуска объекта (ошибки второго рода). Регулируя величину  $D$ , можно лишь менять соотношение между вероятностями возникновения ошибок первого и второго рода в соответствии с заданным критерием оптимальности.

*Корреляционный метод* основан на вычислении взаимной корреляции между объектами и эталонами. Из множества  $k$  альтернативных вариантов выбирается тот объект (или эталон), при котором получается максимальная сумма взаимной корреляции:

$$K(k) = \sum_x \sum_y S_k(x, y) F_k(x, y). \quad (10.3)$$

При идентификации объектов удобно пользоваться коэффициентами корреляции, которые в первом приближении дают и оценку вероятности отнесения объекта к данному эталону:

$$R(k) = K(k)/K_{\max}(k), \quad K_{\max}(k) = \sum_x \sum_y F_k^2(x, y), \quad (10.4)$$

где  $K_{\max}(k)$  – значения автокорреляции эталонов.

Корреляционный метод более надёжен, получил довольно широкое распространение. Однако при распознавании реальных объектов корреляционный метод характеризуется большой вычислительной сложностью. Связано это с масштабированием и поворотами распознаваемого изображения.

*Методы распознавания на основе системы признаков* также используют эталоны объектов, но в качестве элементов сравнения используются признаки объекта и эталона, что позволяет сократить объём эталонных данных и время обработки информации. Однако следует иметь в виду, что на практике выделение признаков объектов всегда осуществляется с некоторой погрешностью, а, следовательно, необходимо выявлять и учитывать характер и степень возможного рассеяния оценок используемых признаков для каждого из ожидаемых объектов, т.е. использовать *гистограммы распределения значений признаков*.

*Перцептивные хэш-алгоритмы.* Используют различные свойства изображения для построения индивидуального «отпечатка» для дальнейшего сравнения изображений [1,4,25]. Формирование перцептивного хеша изображения выполняется в 3 этапа [9]:

- *Предварительная обработка.* На этой стадии изображение приводится к виду, в котором его легче обрабатывать для построения хеша. Это может быть применение различных фильтров.

– *Основные вычисления.* Из полученного на 1 стадии изображения строится матрица (или вектор). Матрица (вектор) может представлять из себя матрицу частот, гистограмму яркостей, или еще более упрощенное изображение.

– *Построение хеша.* Из матрицы (вектора), полученной на 2 стадии берутся некоторые коэффициенты и преобразуются в хеш. Обычно хеш получается размером от 8 до ~100 байт. Полученные хеши сравниваются путем оценки «расстояния» между ними.

Наиболее распространены 4 различных хеш алгоритма: Simple Hash, DCT Based Hash, Radial Variance Based Hash, Marr- Hildreth Operator Based Hash [9].

*Simple Hash.* Суть данного алгоритма заключается в отображении среднего значения низких частот. Принцип работы:

– Уменьшить размер для того, чтобы избавиться от высоких частот. Изображение уменьшается до размера в диапазоне 32x32 и 8x8.

– Убрать цвет, переведя в градации серого, хеш уменьшается втрое.

– Вычислить среднее значение цвета для всех пикселей.

– Построить цепочку битов. Для каждого пикселя делается замена цвета на 1 или 0 в зависимости от того, больше он или меньше среднего.

– Построить хеш. Перевод 1024 битов в одно значение. Порядок не имеет значения, но обычно биты записываются слева направо, сверху вниз.

Полученный хеш устойчив к масштабированию, сжатию или растягиванию изображения, изменению яркости, контраста, манипуляциями с цветами, имеет высокую скорость работы. Для сравнения хешей используется функция нормированного расстояния Хэмминга.

*Discrete Cosine Transform Based Hash.* Дискретное косинусное преобразование (ДКП) – ортогональное преобразование, использующее только косинусные функции, в отличие от дискретного преобразования Фурье, использующего и косинусные, и синусные функции.

Пусть  $x[m]$ , где  $m = 0, \dots, N-1$  — последовательность сигнала длины  $N$  и ДКП определяется по формулам 10.5 или 10.6.

$$X[n] = \sqrt{\frac{2}{N}} \times \sum_{m=0}^{N-1} x[m] \times \cos\left(\frac{(2m+1) \times n\pi}{2N}\right), (n = 0, \dots, N-1) \quad (10.5)$$

$$X[n] = \sum_{m=0}^{N-1} c[n, m] \times x[m], (n = 0, \dots, N-1) \quad (10.6)$$

где  $c[n, m]$  – элемент матрицы ДКП на пересечении строки с  $n$  и столбца с  $m$ .  
ДКП матрица определяется как:

$$c[n, m] = \sqrt{\frac{2}{N}} \times \sum_{m=0}^{N-1} \cos\frac{(2m+1) \times n\pi}{2N}, (n = 0, \dots, N-1) \quad (10.7)$$

ДКП может быть представлена в виде формулы 10.8:

$$\text{ДКП} = M \times I \times M' \quad (10.8)$$

где  $M$  – ДКП матрица,  $I$  – изображение квадратного размера,  $M'$  – обратная матрица.

В качестве матрицы  $I$  берется изображение, сжатое до размера 32x32 и упрощенное с помощью различных фильтров, например, обесцвечивания. В итоге получается матрица ДКП ( $I$ ), в левом верхнем углу которой находятся низкочастотные коэффициенты. Чтобы построить хеш берется левый верхний блок частот 8x8. Затем из этого блока строится хеш размером 8 байт с помощью нахождения среднего и построения цепочки бит (также, как в Simple Based Hash).

Основные шаги построения хеша с помощью данного алгоритма:

- Убрать цвет для подавления ненужных высоких частот.
- Применить медианный фильтр [4] для уменьшения уровня шума.
- Уменьшить изображение до размера 32x32.
- Применить ДКП к изображению.
- Построить хеш.

Главные преимущества такого хеша: устойчивость к малым поворотам, размытию и сжатию изображения, а также скорость сравнения хешей, благодаря их маленькому размеру. Для сравнения хешей этого типа используется функция расстояния Хэмминга

### 10.3 Цель и методика выполнения работы

*Цель:* Получить опыт решения одной из задач компьютерного зрения – классификации изображений в процессе индивидуального решения поставленной задачи и коллективного обсуждения ее решения.

*Форма занятия:* лабораторная работа.

*Метод обучения:* Case-study.

*Метод кейс-стади* – это метод активного обучения на основе описания реальных ситуаций. Метод направлен как на освоение конкретных знаний, или умений, так и на развитие общего интеллектуального и коммуникативного потенциала студентов и преподавателей. Обучающиеся должны проанализировать ситуацию, разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них. Разбор ситуации позволяет демонстрировать академическую теорию с точки зрения реальных ситуаций.

*Сценарий работы:*

Представленный кейс выдается в начале семестра, задача актуализируется на занятии, предшествующем данной лабораторной работе, за 10-20 минут до его окончания.

- 1 Первая лекция. Выдан кейс (не более 20 минут).
- 2 Лабораторная работа №8 или предшествующая лекция. Актуализация кейса.
- 3 Самостоятельная работа студента дома (4 часа) – подготовительная работа: анализ задачи, поиск информации, ответы на вопросы, проект решения.
- 4 Лабораторная работа №9. Проверка, консультация (от 45 минут).  
Разработка программы студентами.
- 5 Самостоятельная работа дома (6 часов) – Разработка программы студентами.
- 6 Лабораторная работа №9. Консультация, доработка. Обсуждение и анализ (не более 45 минут).

*Тип кейса:* Научно-исследовательский

*Кейс:*

Требуется решить задачу распознавания номеров автомобилей, т.е. написать программу для распознавания некоторой комбинации чисел на статическом изображении, изображение может иметь различную высоту и ширину. Для данной задачи характерны следующие условия: белый фон, черные цифры; шрифт Arial (обычный), цифры не имеют общей линии по вертикали и не могут накладываться друг на друга.

*Что требуется сделать:*

1 Выяснить в каких случаях не возможно обнаружение объектов изображений, обосновать теоретически.

2 Обосновать выбор метода обнаружения и распознавания объектов изображений.

3 Реализовать программно выбранный метод обнаружения и распознавания объектов изображений.

4 Исследовать эффективность реализованного метода.

*Что нужно знать:*

1 В чем заключается спектральный анализ?

2 Суть преобразования Фурье.

3 Суть вейвлет-преобразования.

4 Цветовые модели изображений.

5 Понятие образ.

6 В каких реальных задачах могут встречаться гармонические сигналы?

7 Что понимают под обнаружением и распознаванием образов?

8 Какие существуют методы обнаружения и распознавания образов, какие можно применить к распознаванию изображений?

9 Какой метод распознавания является классическим, можно ли его применить для обнаружения изображений?

10 Чем отличается классификация и кластеризация?

11 Какие классы можно выделить в решении задачи распознавания изображений?

12 Что известно и что требуется определить в задаче обнаружения и распознавания изображений?

13 На какие этапы следует разбить решение задачи распознавания изображений?

14 Какие признаки (характеристики) можно выделить для изображений?

15 Какие знания и умения необходимы для решения задачи распознавания образов?

16 Какие критерии можно принять для оценки качества распознавания?

17 Какие факторы будут влиять на качество распознавания изображений?

#### **10.4 Порядок выполнения работы**

1 Внимательно изучить кейс (постановку задачи). Прочитать краткие теоретические сведения.

2 Ответить на вопросы, используя дополнительные источники информации.

3 Выполнить задания кейса.

4 Сделать выводы, написать отчет о проделанной работе.

## 11 Спектральный анализ в задаче обнаружения вторжений

### 11.1 Методы выявления аномалий трафика корпоративной сети

Для решения задачи обнаружения вторжений на основе выявления аномалий трафика корпоративных компьютерных сетей (ККС) широко используются методы обработки трафика, характеризующего загрузку каналов передачи данных ККС при отслеживающих и блокирующих работу сети атаках.

Для проведения анализа трафика необходимо построение математической модели нормального функционирования сети и определение допустимых границ разброса её параметров. Идентификация математической модели сетевого трафика наиболее эффективна с помощью известных методов гармонического анализа (преобразование Фурье). Однако в ККС информационные процессы нестационарны и неоднородны в пространстве [14,23], что затрудняет адекватное описание их математических моделей указанными методами.

Перспективным для математического моделирования сетевого трафика и выявления аномалий трафика сети является использование теории мультиразрешающего анализа (МРА), основой которой являются методы вейвлет – преобразований и теории идентификации динамических систем, одной из широко применяемых параметрических моделей которой является прогнозирующая авторегрессионная модель (ARX-модель) (рисунок 11.1).

Теоретической основой МРА является теория вейвлет – анализа [7,14]. Вейвлеты (wavelet – короткая волна) – это обобщенное название функций определенной формы, локализованных по оси аргументов (независимых переменных), инвариантных к сдвигу и линейных к операции масштабирования (сжатия/растяжения), имеющих вид коротких волновых пакетов с нулевым интегральным значением. Они создаются с помощью специальных базовых функций, которые определяют их вид и свойства. По локализации во временном и частотном представлении вейвлеты занимают промежуточное положение между гармоническими функциями [7], локализованными по частоте, и функцией Дирака,

локализованной во времени.



Рисунок 11.1 – Модели информационных процессов

Основная область применения вейвлетных преобразований – анализ и обработка сигналов и функций, в том числе и дискретных в виде массивов цифровых данных, нестационарных во времени или неоднородных в пространстве, когда результаты анализа должны содержать не только общую частотную характеристику сигнала (распределение энергии сигнала по частотным составляющим), но и сведения об определенных локальных координатах, на которых проявляются те или иные группы частотных составляющих, или, на которых происходят быстрые изменения частотных составляющих функций. По сравнению с разложением сигналов на ряды Фурье [14], вейвлеты способны с гораздо более высокой точностью представлять локальные особенности функций, вплоть до разрывов 1-го рода (скачков). Кроме того, в отличие от преобразований Фурье, вейвлет-преобразование одномерных массивов цифровых данных обеспечивает двумерную развертку, при этом частота и координата рассматриваются как независимые переменные, что дает возможность анализа массивов сразу в двух пространствах.

Одной из целей построения модели является оперативное управление трафиком сети, представляющим собой динамическую систему, в режиме реального времени. Прогнозирование состояния трафика сети повышает оперативность принятия решения. Одним из часто применяемых видов моделей прогнозирования состояния динамических систем является класс ARIMAX-моделей, которые позволяют устанавливать зависимость состояния системы в момент времени  $t+s$  от предыдущих состояний в моменты времени  $t+s \leq t-1$  и позволяют прогнозировать состояние системы, в частности состояние трафика сети [14].

Таким образом, для математического моделирования сетевого трафика и выявления аномалий трафика ККС целесообразно использовать теорию мультиразрешающего анализа и теорию динамических систем.

## 11.2 Модель сетевого трафика

На рисунке 11.2 представлена технология выявления и блокирования аномалий трафика ККС.

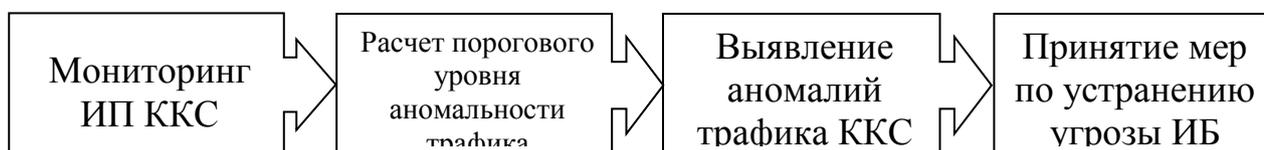


Рисунок 11.2 – Технология выявления и блокирования аномалий трафика

Под мониторингом информационных процессов (ИП) ККС понимается анализ сетевого трафика (объем переданной информации в байтах, количество переданных пакетов, количество потоков) за определенный интервал времени. Трафик сети рассматривается в виде совокупности одномерных числовых рядов  $f(t_i)$  – отсчетов характеристик трафика, заданных в дискретные моменты времени  $t_i = i\Delta$ , где  $i = 0, 1, \dots, N-1$ ,  $\Delta$  – интервал между отдельными наблюдениями,  $N$  – количество наблюдений.

В качестве характеристик трафика берутся его показатели, имеющие потенциал для различения нормального состояния сети и аномального:

а) количество потоков в трафике за период времени;

б) среднее количество пакетов в потоке на интервале времени. Большинство атак происходит с увеличением количества пакетов;

в) среднее количество байт в потоке на интервале времени. Некоторые атаки используют большой размер пакета для истощения вычислительных ресурсов;

г) среднее количество байт на пакет в потоке за период времени. Описывает размер пакета более подробно, чем в предыдущем случае;

д) отношение количества потоков к среднему количеству пакетов в потоке. Характеризует начало большого количества соединений с небольшими пакетами с целью достижения максимальной производительности сканирования.

Каждый полученный ряд обрабатывается независимо от остальных. Такое описание трафика позволяет учитывать его различные характеристики, не прибегая к анализу многомерных сигналов и обрабатывать их в параллельном режиме.

Произвольная последовательность  $f(t_i)$  в теории цифровых временных рядов обычно рассматривается в виде суммы разнотипных составляющих [14]:

– функции тренда  $q_T(t_i)$  – средних значений по большим интервалам усреднения (медленно меняющаяся во времени функция, описывающая изменения среднесуточных загрузок ККС за интервалы времени большие, чем суточная периодичность);

– циклических компонент с определенным периодом повторения  $q_C(t_i)$  как правило, достаточно гладких по форме (периодическая составляющая, описывающая изменения среднесуточных загрузок ККС);

– локальных особенностей (аномалий) разного порядка  $\varepsilon_a(t_i)$ , вплоть до вторжений – резких изменений в определенные редкие моменты;

– флуктуаций – значений более высокого порядка (шумов)  $\varepsilon_\Phi(t_i)$  вокруг всех вышеперечисленных составляющих функции.

Таким образом, обобщенная модель сетевого трафика представляется в следующем виде:

$$f(t_i) = q_{\tau}(t_i) + q_{\Pi}(t_i) + \varepsilon_a(t_i) + \varepsilon_{\Phi}(t_i). \quad (11.1)$$

Построение модели сетевого трафика сводится к определению аналитического представления каждого слагаемого (11.1). В ранее проведенных исследованиях относительно независимой случайной последовательности  $\varepsilon_{\Phi}(t_i)$  принималось допущение, что в случайные моменты времени  $t_i$  математическое ожидание  $M[\varepsilon_{\Phi}(t_i)] = 0$  с дисперсией  $\sigma_{\varepsilon^2}(t_i)$ . Однако использование сетей общего пользования привело к существенному росту шумовой составляющей  $\varepsilon_{\Phi}(t_i)$  ИП ККС и допущение стало неприемлемым.

Следовательно, используя методологию МРА модель сетевого трафика  $f(t_i)$  (11.1) можно рассматривать на любом  $m$  – уровне разрешения между ее усредненными значениями и флуктуациями вокруг средних значений в виде:

$$f(t_i) = \sum_{k=-\infty}^{\infty} c_{m,k} \varphi_{m,k}(t) + \sum_{m=m'}^{\infty} \sum_{k=-\infty}^{\infty} d_{m,k} \psi_{m,k}(t), \quad (11.2)$$

$$m, k \in I,$$

где  $\varphi_{m,k}(t)$  – масштабирующая функция, с помощью которой выполняется аппроксимация сетевого трафика;

$\psi_{m,k}(t)$  – вейвлет-функция, выделяющая детали сетевого трафика и его локальные особенности;

$c_{m,k}, d_{m,k}$  – аппроксимирующие и детализирующие коэффициенты;

$m$  – параметр масштаба;

$k$  – параметр сдвига;

$I$  – пространство целых чисел  $\{-\infty, \infty\}$ .

Значения коэффициентов  $c_{m,k}$  и  $d_{m,k}$ , которые называются масштабными коэффициентами (приближения) и вейвлет-коэффициентами (детали), определяются зависимостями [14]:

$$c_{m,k} = \int_t f(t) \varphi_{m,k}(t) dt, \quad (11.3)$$

$$d_{m,k} = \int_t f(t) \psi_{m,k}(t) dt, \quad (11.4)$$

Первая сумма в (11.2) содержит усредненные (с весовыми функциями  $\varphi_{m,k}$ ) значения функции  $f(t_i)$  по диадным интервалам  $[k \cdot 2^{-m}, (k+1) \cdot 2^{-m}]$ , характеризует тренд и циклические составляющие трафика (суточные и недельные), т.е.

$$q_m(t_i) + q_u(t_i) = \sum_{k=-\infty}^{\infty} c_{m,k} \varphi_{m,k}(t), \quad (11.5)$$

а вторая – значения флюктуаций на данных интервалах, характеризующих аномалии трафика  $\varepsilon_a(t_i)$ , с учетом случайной шумовой помехи  $\varepsilon_\phi(t_i)$ :

$$\varepsilon_a(t_i) + \varepsilon_\phi(t_i) = \sum_{m=m'}^{\infty} \sum_{k=-\infty}^{\infty} d_{m,k} \psi_{m,k}(t). \quad (11.6)$$

Таким образом, выражение (11.2) показывает возможность аппроксимации любой произвольной функции  $f(t_i)$  набором простых локальных функций  $\varphi_{m,k}(t)$  и  $\psi_{m,k}(t)$ , ортогональных на разных уровнях значений  $m$  и полностью покрывающих пространство  $L^2(R)$  за счет смещений  $k$ . Переход от  $m$  к  $m+1$  эквивалентен замене  $t$  на  $2t$ , т.е. перемасштабированию функций  $\varphi_{m,k}(t)$  и  $\psi_{m,k}(t)$ .

При оценке сетевого трафика в виде конечного набора отсчетов, наилучший уровень разрешения определен интервалом, содержащим один отсчет, и суммирование выполняется в конечных пределах. Значение  $m = 0$  принимается для этого наилучшего уровня разрешения. При принятой форме вейвлетов коэффициенты  $c_{m,k}$  и  $d_{m,k}$  вычисляются для  $m > 0$ .

МРА при последовательном увеличении значений  $m$  реализуется в форме быстрых итерационных вычислений вейвлет-коэффициентов по формулам 4.2-4.4, приведенным в разделе 4. Для массивов цифровых данных сетевого трафика в качестве значений  $c_{0,k}$  принимаются исходные значения одномерного числового ряда, т.е.  $c_{0,k} = f(k) = f(t_0)$ .

Реализация быстрого вейвлет-преобразования одномерного числового ряда может быть выполнена на основе пирамидального алгоритма вычисления вейвлет-коэффициентов (алгоритм Малла), приведенного на рисунке 4.1 слева или алгоритма вейвлет-пакетов (ВП), приведенного на рисунке 4.1 справа. Сущность операций алгоритма Малла, выполняемых на основе (4.2) и (4.3), заключается в следующем. С учетом спектров коэффициентов  $h_n$  и  $g_n$ , на первом этапе преобразования первый цифровой фильтр  $h_n$  из числового ряда  $f_k = c_{0,k}$  выделяет низкие частоты  $|\omega| \leq \pi/2$ , а другой (октавный) фильтр  $g_n$  выделяет верхние частоты  $\pi/2 \leq |\omega| \leq \pi$  (рисунок 11.3). Поскольку на выходе фильтра  $h_n$  отсутствует верхняя половина частот, то частота дискретизации выходного сигнала может быть уменьшена в 2 раза, т.е. выполнена децимация выходного массива, что и производится в формуле (4.2) сдвигами  $(2k+n)$  через 2 отсчета по входному массиву. Соответственно, на выходе фильтра  $g_n$  освобождается место в области низких частот, и аналогичное прореживание выходного числового массива приводит к транспонированию верхних частот на освободившееся место. Таким образом, каждый из выходных числовых массивов несет информацию о своей половине частот, при этом выходная информация представлена таким же количеством отсчетов, что и входная.

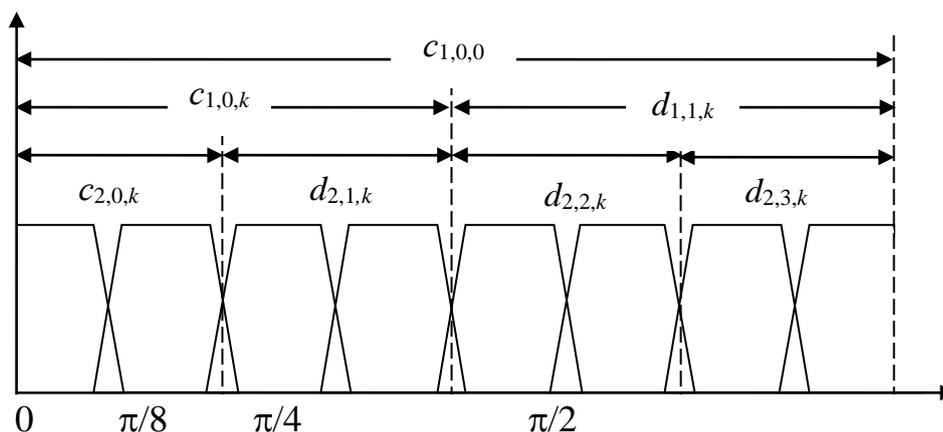


Рисунок 11.3 – Спектральные характеристики ВП

ВП позволяют наиболее полно выделить локальные особенности трафика – всплески и аномалии за счёт дополнительной декомпозиции высокочастотных

составляющих спектра сигнала, представляющего трафик (рисунок 4.1 справа). Реализация алгоритма ВП аналогична пирамидальному свёртке с фильтрами  $h_n$  и  $g_n$ . В алгоритме ВП при переходе с масштабного уровня  $m$  на уровень  $m+1$  как низкочастотные (аппроксимирующие) коэффициенты  $c_{m,k}$ , так и высокочастотные (детализирующие) коэффициенты  $d_{m,k}$  разделяются на низкочастотные ( $c_{m+1,k}$ ) и высокочастотные ( $d_{m+1,k}$ ) части спектрального диапазона по формулам:

$$c_{m+1,2p,k} = \sum_n h_n c_{m,p,2k+n}, \quad d_{m+1,2p,k} = \sum_n g_n c_{m,p,2k+n}, \quad c_{0,k} = f(k), \quad (11.7)$$

$$c_{m+1,2p+1,k} = \sum_n h_n d_{m,p,2k+n}, \quad d_{m+1,2p+1,k} = \sum_n g_n d_{m,p,2k+n}, \quad k = 0..N/2^m. \quad (11.8)$$

Итак, правая сумма формулы (2.7), задающей временной ряд характеризует аномалии сетевого трафика ККС на фоне флуктуаций и, следовательно, анализ данной части можно положить в основу методики определения текущего уровня отклонения от нормального поведения сетевого трафика.

Из множества возможностей определения базиса для разложения ВП – от «минимального» вейвлет-разложения (алгоритм Малла) до полного пакетного разложения на всех уровнях выбирается оптимальный, адаптированный к анализируемому трафику. При обычном вейвлет-разложении (алгоритм Малла) вейвлет-коэффициентов достаточно для восстановления трафика, и можно раскладывать высокочастотные коэффициенты деталей или отказаться от этого. Таким образом, появляется гораздо больше возможностей выбора базиса для разложения – от «минимального» вейвлет-разложения до полного пакетного разложения на всех уровнях. Из всех представителей нужно выбрать то, которое представляет трафик наиболее эффективно. Под «эффективным» подразумевается то, что трафик может быть представлен небольшим количеством коэффициентов разложения, т.е. базис для разложения должен быть таким, что большие коэффициенты сконцентрированы на небольшом количестве элементов вейвлет-пакетного базиса и большое количество коэффициентов близко к нулю.

В качестве критерия выбора самого эффективного или лучшего базиса для трафика можно использовать критерий минимальности энтропии. Энтропия характеризует усреднённость трафика. Лучший базис – это тот базис, который даёт наименьшее количество энтропии. Интуитивно энтропия Шеннона даёт критерий того, сколько эффективных компонент необходимо, чтобы представить трафик в определенном базисе. Чем меньше энтропия, тем меньше существенных коэффициентов нужно для представления трафика.

Критерий энтропии используется для выбора оптимального базиса следующим образом. Если при разложении коэффициентов некоторого узла сумма энтропий, полученных при разложении компонент, меньше, чем энтропия коэффициентов в исходном узле, то разложение применяется, в противном случае коэффициенты (вместе с базисными функциями) остаются без изменения.

На рисунке 11.4 приведен пример трафика и соответствующее ему оптимальное вейвлет-пакетное разложение. Другие примеры приведены в приложении Г.

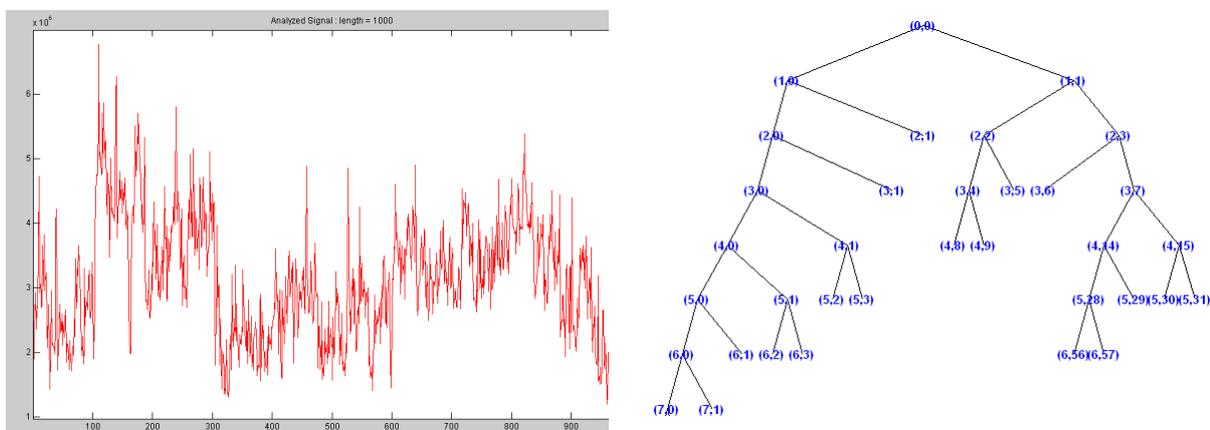


Рисунок 11.4 – Оптимальное вейвлет-пакетное разложение

Пусть значения номеров уровней и номеров узлов определены по результатам выбора оптимального базиса вейвлет-пакетов, и среди них номера уровней  $m_{o1}..m_{ol}$  и номера узлов  $p_{o1}..p_{oz}$  характеризуют аномалии, а номера уровней  $m_{\phi 1}..m_{\phi l}$  и номера узлов  $p_{\phi 1}..p_{\phi z}$  характеризуют шум. Тогда формула (11.6), описывающая аномалии сетевого трафика со случайной шумовой составляющей трафика принимает вид:

$$f_d(t_i) = \varepsilon_a(t_i) + \varepsilon_\phi(t_i) = \sum_{m=m_{o1}}^{m_{o1}} \sum_{p=p_{o1}}^{p_{o1}} \sum_{k=0}^{N/2^m} d_{m,p,k} \Psi_{m,p,k}(t_i) + \sum_{m=m_{\phi 1}}^{m_{\phi 1}} \sum_{p=p_{\phi 1}}^{p_{\phi 1}} \sum_{k=0}^{N/2^m} d_{m,p,k} \Psi_{m,p,k}(t_i) \quad (11.9)$$

Тогда разность между эталонным  $f_{d^3}(t_i)$  уровнем аномального поведения сетевого трафика с учетом случайной шумовой помехи, определяемым в нормальном режиме, и регистрируемым уровнем  $f_{d^p}(t_i)$  с предположением равенства флуктуаций  $\varepsilon_\phi^3(t)$  и  $\varepsilon_\phi^p(t)$ , определяет текущий уровень отклонения от нормального поведения трафика ККС:

$$\tilde{\varepsilon}_a(t_i) = f_{d^3}(t_i) - f_{d^p}(t_i) = \sum_{m=m_{o1}}^{m_{o1}} \sum_{p=p_{o1}}^{p_{o1}} \sum_{k=0}^{N/2^m} d_{m,p,k}^3 \Psi_{m,p,k}(t_i) - \sum_{m=m_{o1}}^{m_{o1}} \sum_{p=p_{o1}}^{p_{o1}} \sum_{k=0}^{N/2^m} d_{m,p,k}^p \Psi_{m,p,k}(t_i). \quad (11.10)$$

Предложенный метод вейвлет – преобразований позволяет повысить адекватность описания модели сетевого трафика, учитывая неопределенность информационных процессов ККС (сетевой трафик в виде массива цифровых данных), в частности стохастическую его часть, характеризующую аномалии трафика ККС  $\varepsilon_a(t_i)$ , с учетом случайной шумовой помехи  $\varepsilon_\phi(t_i)$ . Таким образом, получил развитие метод кратноразрешающего анализа в задаче выявления аномального поведения трафика путём анализа высокочастотных компонент с использованием быстрого вейвлет-преобразования гибкой конструкции вейвлет-пакетов, позволяющий повысить эффективность и оперативность выявления сетевых аномалий.

### 11.3 Прогнозирование текущего состояния трафика корпоративной сети

Модель (11.10) не обеспечивает обнаружение аномалий в режиме реального времени в силу необходимости необходимости пересчета нормального состояния трафика сети  $f_{d^3}(t)$  на каждом интервале мониторинга сети.

Для решения этой проблемы, предлагается прогнозировать величину  $f_{d^3}(t)$ , используя модель «черного ящика», на выходе которого генерируются стохастические

процессы в зависимости от управляющего сигнала  $D^3(t)$ , зашумленного некоторым неконтролируемым сигналом  $\varepsilon_\phi(t)$  (рисунок 11.5).

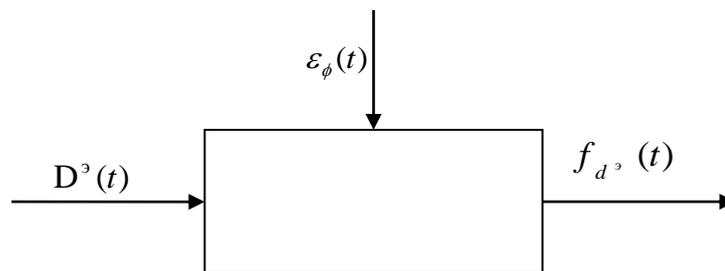


Рисунок 11.5 – Модель «черного ящика»

Из множества структур моделей для рассматриваемой совокупности наблюдений наиболее предпочтительной является класс ARIMAX-моделей линейной разностной динамической системы, компоненты которого могут самостоятельно использоваться как авторегрессионные, стационарные, нестационарные, нелинейные и робастные к различным проявлениям среды модели.

Из всех представителей класса ARIMAX-моделей широко применяется при обработке различного рода сигналов ARX-модель.

Одна из форм записи ARX-модели имеет вид

$$y(t+s) = \sum_{i=1}^p a_i y(t-i) + \sum_{i=1}^q b_i x(t-i) + e(t), \quad (11.11)$$

где  $a_i$  и  $b_i$  настраиваемые параметры модели,

$y(t-i)$  – предыдущие значения выхода (образцы);

$x(t-i)$  – входные значения (регрессор);

$e(t)$  – белый шум.

Тогда предсказатель для ARX-модели случайной составляющей сетевого трафика с регрессором в форме вейвлет-коэффициентов примет вид

$$\tilde{f}_{d^3}(t+s) = \sum_{i=1}^r a_i \tilde{f}_{d^3}(t-i) + \sum_{i=1}^q b_i D^3(t-i), \quad (11.12)$$

где  $a_i$  и  $b_i$  – параметры модели;

$f_{d^s}(t)$  – значения случайной составляющей, определяющие предыдущие значения выхода (образцы);

$D^s(t) = \{d_{m,p,k}^s\}$ ,  $m = m_{o1}, \dots, m_{ol}$ ;  $p = p_{j1}, \dots, p_{jz}$ ;  $k = 0..(N+1)/2^m$  – вектор коэффициентов, определяющий входные значения (регрессор);

$r, q$  – показатели глубины истории.

Ошибка прогноза в момент времени  $t$  определяется формулой:

$$\xi(t) = \tilde{f}_{d^s}(t) - f_{d^s}(t), \quad (11.13)$$

где  $f_{d^s}(t)$  – эталонные значения случайной составляющей.

Значения параметров  $\theta = \{a_i, b_i\}$  определяются из условия минимума ошибки прогноза (11.13) по методу наименьших квадратов.

Тогда прогнозируемый уровень отклонения от нормального состояния трафика определяется как разность прогнозируемого и текущего значения случайной составляющей:

$$\tilde{\varepsilon}_a(t) = \tilde{f}_{d^s}(t) - f_{d^p}(t), \quad (11.14)$$

где  $f_{d^p}(t)$  – текущие значения случайной составляющей.

Для линейных регрессий формула (11.12) может быть записана в виде

$$\tilde{f}_{d^s}(t+s) = \varphi^T(t)\theta, \quad (11.15)$$

где  $\varphi(t) = (-f_{d^s}(t-1), \dots, -f_{d^s}(t-r), d^s(t-1), \dots, d^s(t-q))^T$  – вектор регрессий.

Тогда согласно методу наименьших квадратов вектор  $\theta$  оптимальных параметров можно вычислить путём решения линейной системы с симметричной матрицей

$$R(N)\theta = f(N), \quad (11.16)$$

где

$$R(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) \cdot \varphi^T(t), \quad f(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) \cdot y(t).$$

Вектор регрессий  $\varphi(t)$  составляется из значений трафика и вейвлет-коэффициентов.

Значения для  $r$  и  $q$  определены экспериментально и равны соответственно 7 и 8. На рисунке 11.6 изображены графики высокочастотной составляющей текущего трафика с аномалией (точечная кривая) и её прогнозируемые значения с помощью построенной модели. Аномалия в данном случае есть результат сканирования сети. Рисунок 11.6 показывает, что отклонения прогнозируемых значений от текущих в месте аномалии сильно отличаются от остальных.

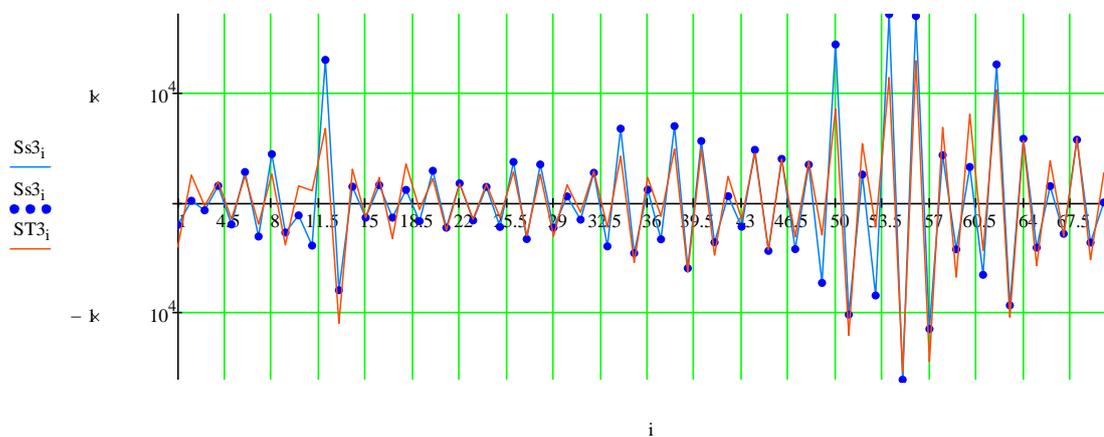


Рисунок 11.6 –Прогнозирование случайной составляющей трафика

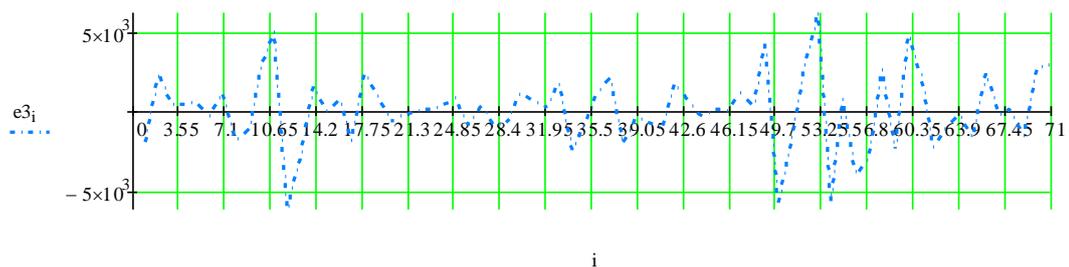


Рисунок 11.7 – Текущая аномальная активность

Таким образом, прогнозирующая ARX-модель (11.12) устанавливает зависимость состояния трафика в момент времени  $t+s$  от предыдущих состояний в моменты времени  $t-1, t-2, \dots, t-r$  и прогнозирует текущие значения случайной составляющей  $f_d(t)$  трафика сети на величину глубины прогноза, тем самым

повышая оперативность принятия решения при выявлении сетевых аномалий, не требуя постоянного пересчета текущих значений.

#### 11.4 Методика идентификации уровня аномальности трафика корпоративной сети

В основу идентификации уровня аномальности трафика ККС положена математическая модель сетевого трафика (11.9) и прогнозирующая авторегрессионная модель (11.12), используемая в двух режимах: обучение и анализ [14].

В режиме обучения проводится моделирование нормального трафика сети, которое состоит из этапов, представленных на рисунке 11.8 [14].

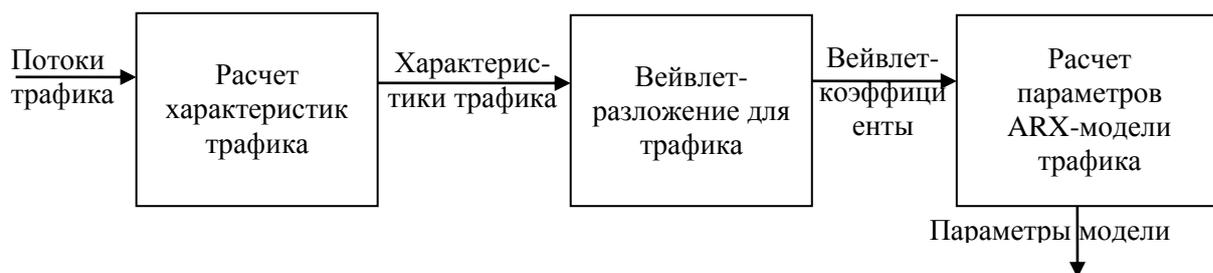


Рисунок 11.8 – Моделирование нормального трафика

СППР собирает информацию о трафике, воспринимая его как нормальный режим работы сети. Длительность обучения и временные интервалы регистрации трафика составляют отдельную научную задачу и обосновываются ниже.

При этом СППР на основе пакетов, прошедших через средство разграничения доступа (маршрутизатор) фиксирует отсчеты сигналов, и запоминает их в базе данных в текстовом формате цифровых данных в виде конечного набора отсчетов.

В ходе вейвлет-разложения полученные цифровые массивы отсчетов, представляющие трафик сети преобразуются в наборы коэффициентов с помощью одного из быстрых алгоритмов кратноразрешающего анализа – вейвлет-пакетов (ВП).

Далее на основе полученных коэффициентов с помощью модели (11.12) прогнозируется нормальный уровень трафика  $\tilde{f}_{d^p}(t)$ . Входной вектор  $D^p(t)$  модели формируется из высокочастотных коэффициентов  $d_{m,p,k}^p$ . Вектор предыдущих значений выхода  $f_{d^p}(t)$  составляется из значений характеристик трафика, также восстановленных из высокочастотных компонент. В режиме анализа по очередным отсчётам трафика, рассчитывается  $f_{d^p}(t)$  и текущий уровень отклонения от нормального поведения сетевого трафика по формуле (11.14).

В настоящее время в МРА разработано большое количество вейвлетов. Выбор анализирующего вейвлета во многом определяется тем, какую информацию необходимо извлечь из массива цифровых данных. С учетом характерных особенностей различных вейвлетов во временном и в частотном пространстве, можно выявлять в анализируемых массивах цифровых данных те или иные свойства и особенности, которые незаметны на гистограммах, особенно в присутствии шумов  $\varepsilon_{\phi}(t_i)$ . Для выявления аномалий трафика ККС необходимо проводить извлечение детализирующей информации из массива цифровых данных. Поэтому для выбора системы базисных вейвлетов были проанализированы вейвлеты с компактным носителем: Хаара, Добеши – 2, вейвлеты Койфмана – койфлеты – 2 (рисунок 11.9), которые наиболее качественно выделяют локальные особенности сигналов.

Для мониторинга сетевого трафика в качестве базиса целесообразно использовать систему вейвлетов – койфлеты-2, которые имеют близкую к симметричной форму, обеспечивают большее количество малозначимых, близких к нулевым, коэффициентов разложения и имеют более высокую крутизну среза полосы пропускания, а, соответственно, обеспечивают лучшее качество разложения сигналов и их реконструкции.

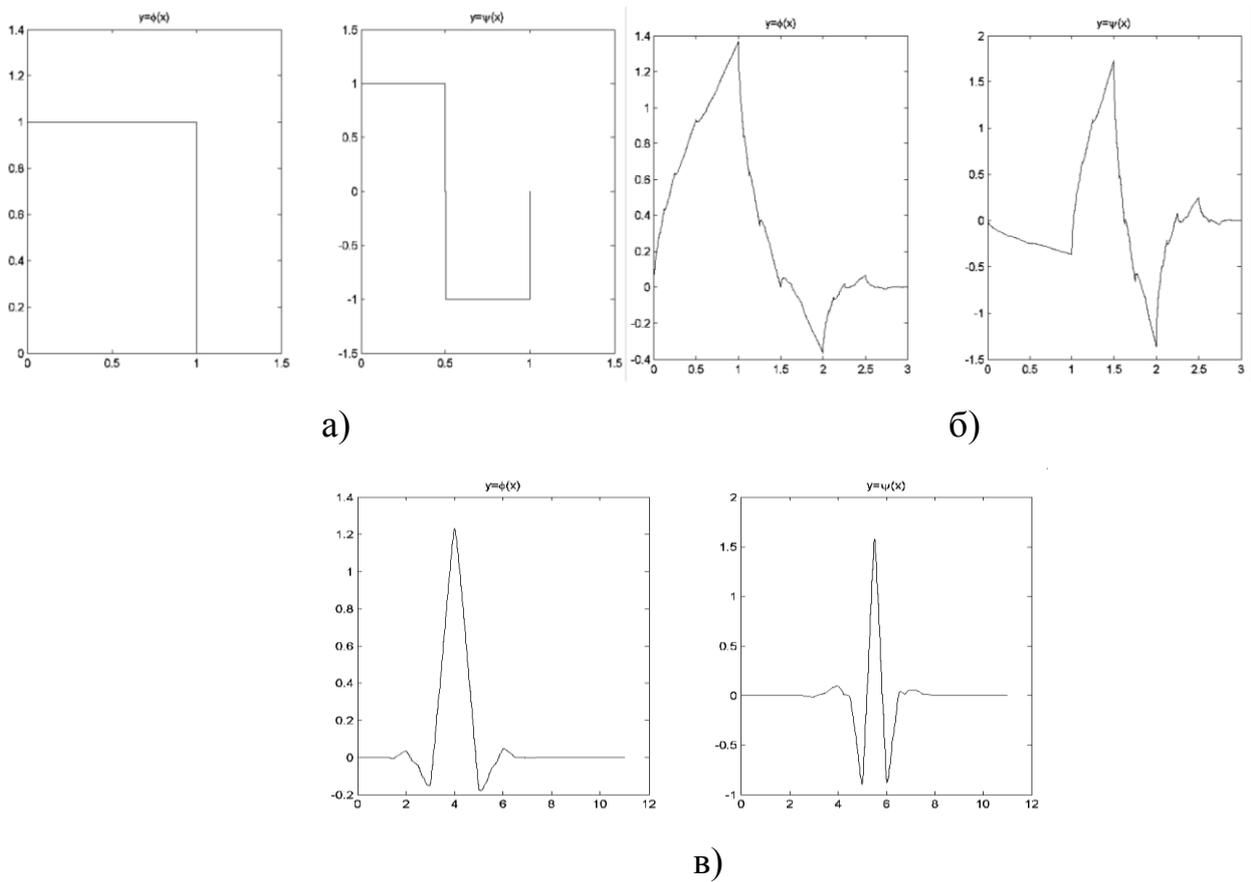


Рисунок 11.9 – Вейвлеты: а) Хаара; б) Добеши – 2, в) койфлеты – 2

Вид массива данных сетевого трафика во временном пространстве с наложением графика койфлетов – 2 представлен на рисунке 11.10 .

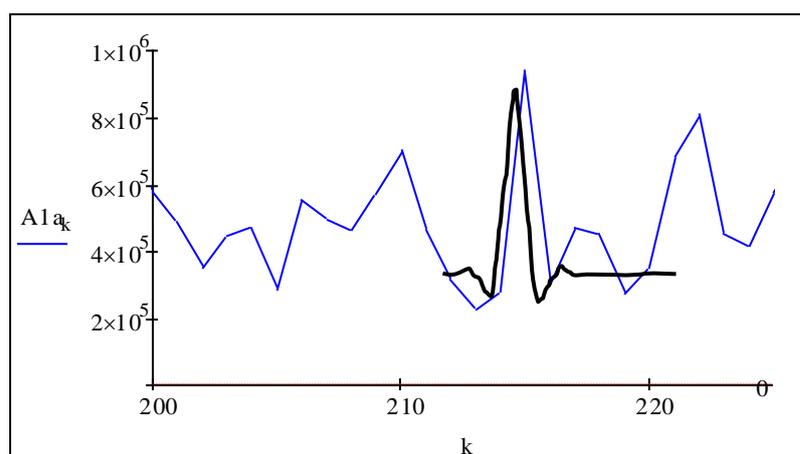


Рисунок 11.10 – Трафик сети с наложением графика койфлета

Таким образом, предложенную методику на основе интеграции вейвлет-

пакетной модели сетевого трафика с базисными функциями койфлетами – 2 и прогнозирующей авторегрессионной модели можно использовать для выявления аномальности трафика ККС.

### 11.5 Алгоритмы мониторинга информационных процессов

На этапе алгоритмизации модель вида (11.9, 11.12) воплощается в конкретную машинную модель. В соответствии с рассмотренным ранее математическим аппаратом выявления аномалий сетевого трафика разработан соответствующий алгоритм функционирования СОА. На рисунке 11.11 представлена укрупненная схема алгоритма мониторинга информационных процессов ККС – мониторинга сетевого трафика.

На последующих рисунках 11.12 – 11.14 представлены схемы алгоритмов функций, запускаемых при мониторинге сетевого трафика [14].

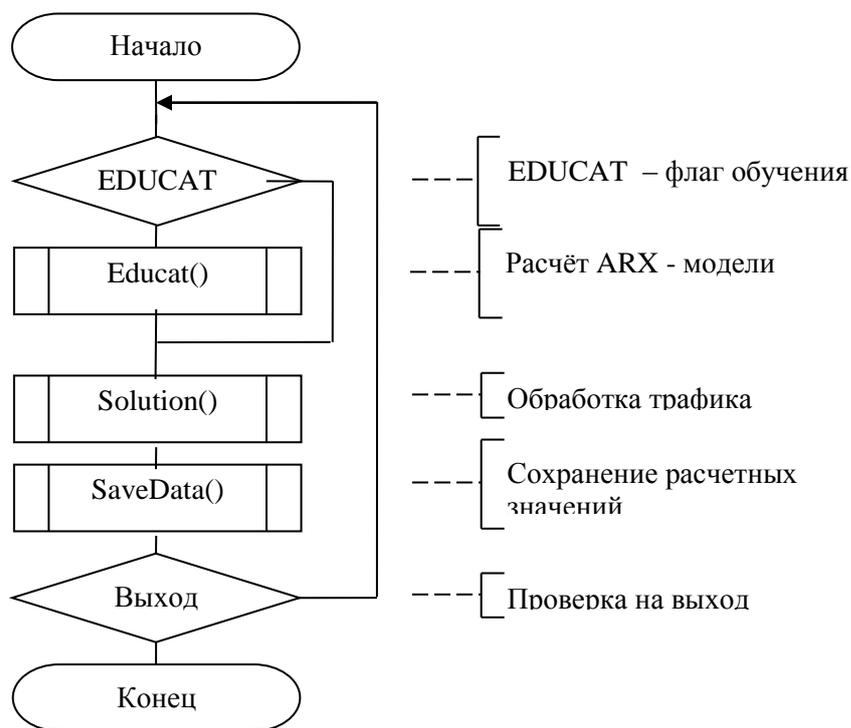


Рисунок 11.11 – Укрупненная схема алгоритма мониторинга информационных процессов ККС

Мониторинг трафика начинается с проверки необходимости обучения системы, по результатам которого формируется модель прогнозирования эталонных значений.

В функции Solution() происходит считывание текущих значений трафика функцией SensAr() массив S. Далее полученные данные (массив S) передаются в функцию WavPack(), где происходит расчет всех аппроксимирующих и детализирующих коэффициентов по алгоритму вейвлет-пакетов, полученные в результате вейвлет-коэффициенты передаются в функцию ARXforecast(), где прогнозируется высокочастотная составляющая трафика, затем определяется уровень отклонения от нормального состояния трафика сети функцией Fanomalius()



Рисунок 11.12 – Схема алгоритма функции определения уровня угрозы безопасности Solution()

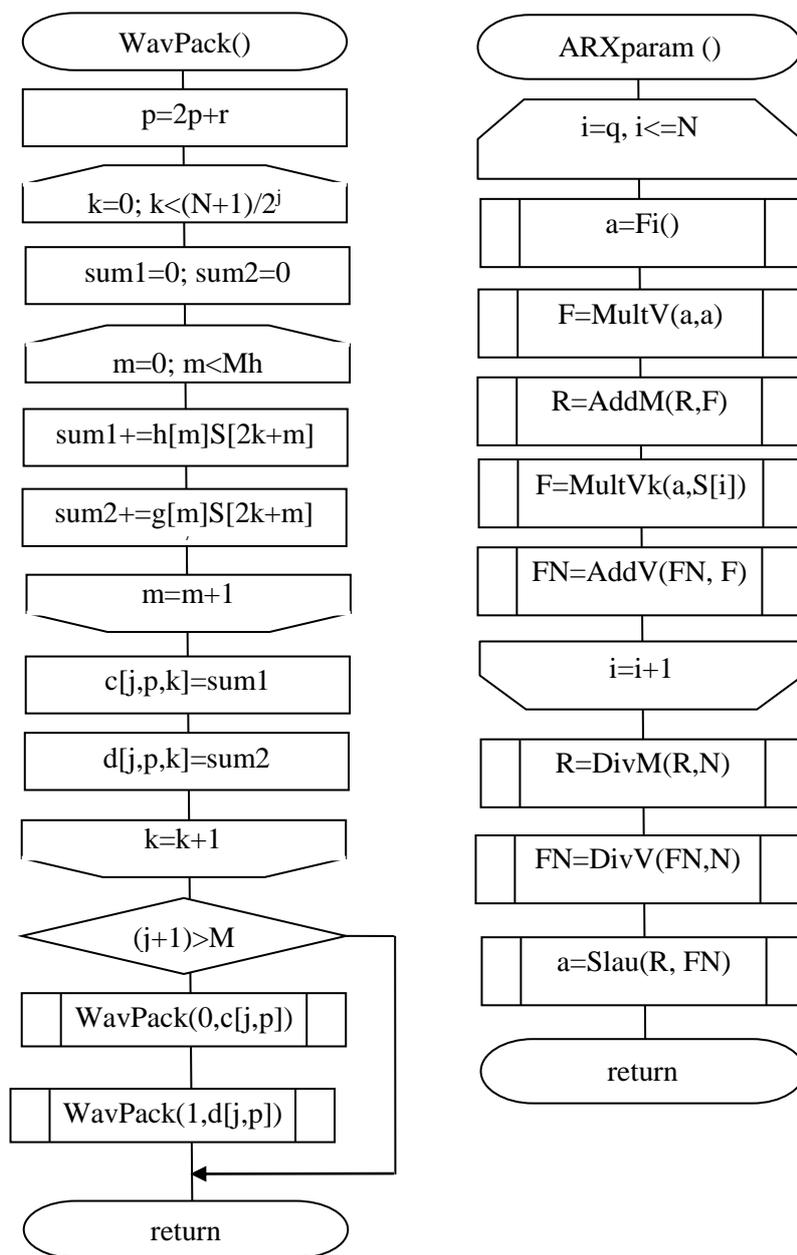


Рисунок 11.13 – Схема алгоритма функции нахождения коэффициентов вейвлет разложения сигнала на определенном уровне WavPack()

По значению аномальной составляющей трафика происходит определение уровня угрозы безопасности с помощью функции OpredAlarm.

Затем полученные результаты сохраняются в базе данных (функция SaveData()), после чего цикл повторяется снова либо происходит выход из мониторинга.



Рисунок 11.14 – Алгоритмы расчёта параметров ARX – модели и прогнозирования

На рисунках 11.15-11.21 представлены экранные формы программного средства, реализующего рассмотренную методику.

Статус состояния сети

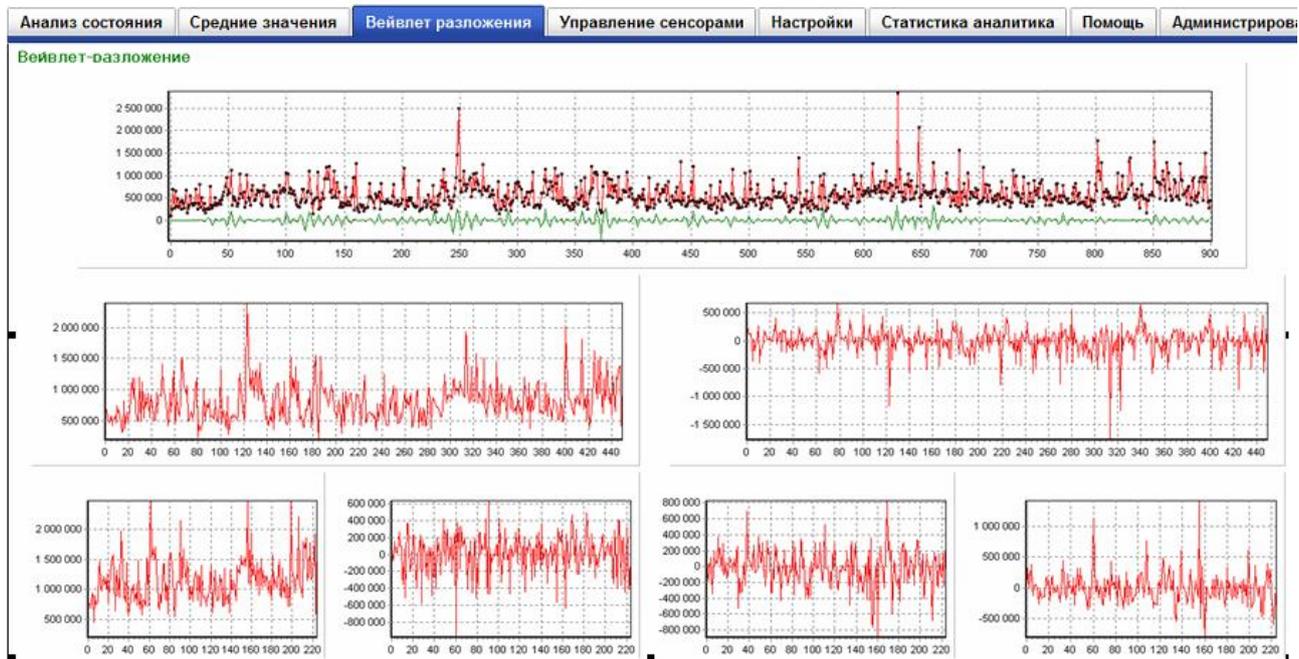


Рисунок 11.15 – Экранная форма вейвлет-разложения

Статус состояния сети

Анализ состояния | Средние значения | Вейвлет разложения | Управление сенсорами | **Настройки** | Статистика аналитика | Помощь | Администрирование пользователями

**Оборудование**

- Сетевые устройства
- Производитель оборудования
- Тип оборудования
- Списки фильтров

**Сенсоры**

- Тип сенсора
- Сенсоры

**События**

- Команды реагирования списки
- Тип правила
- События
- Тип события

**Справочник математической модели**

- Мат.модель
- Чета непонял зачем это (TresHold)
- Чета непонял зачем это (Feature)
- Тип события

**Результаты работы системы**

- Посчитанные результаты
- Статистика (traffic)
- Режимы
- Запуск

Рисунок 11.16 – Экранные формы настроек программного средства

Статус состояния сети

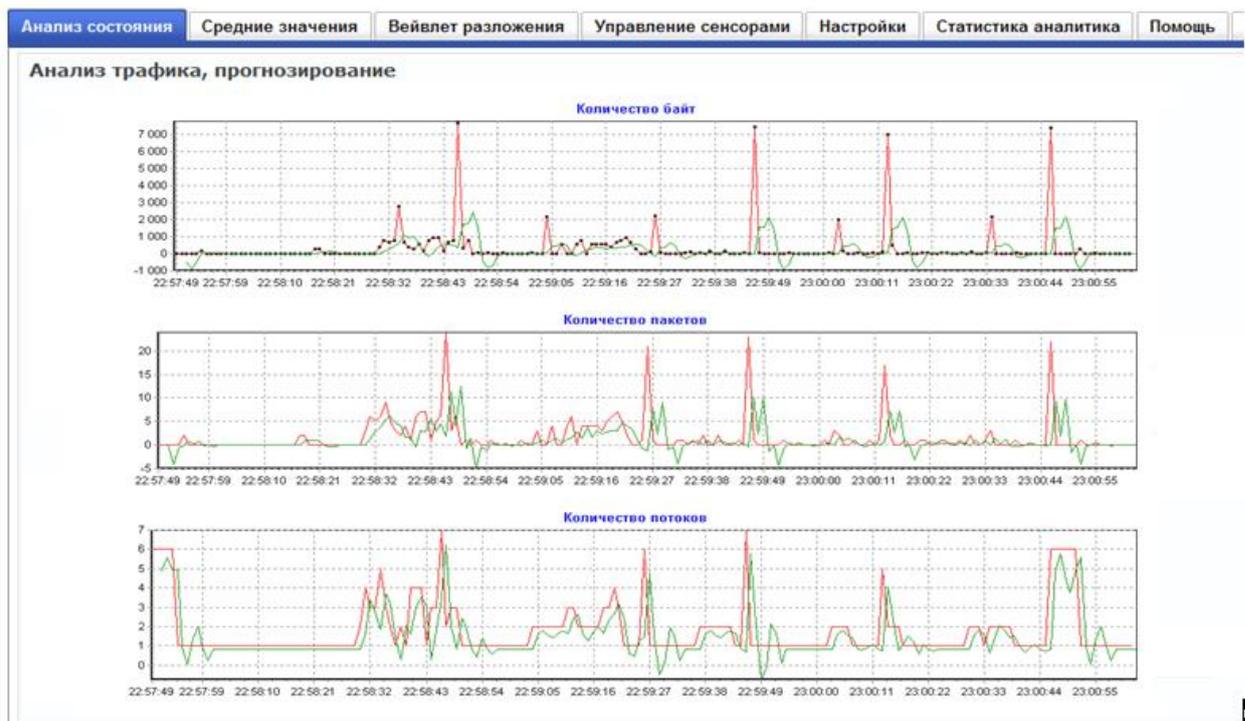


Рисунок 11.17 – Экранная форма анализа состояния

Статус состояния сети

Анализ состояния Средние значения Вейвлет разложения **Управление сенсорами** Настройки Статистика аналитика Помощь Адми

### Список сенсоров

Состояние	Название	Ip	Port	Тип	Управление
В работе	Sensor 2	192.168.1.1	8889	Float 1	<input type="button" value="Запустить"/> <input type="button" value="Посмотреть состояние"/>
В работе	Sensor 1	192.168.1.2	8884	Float 1	<input type="button" value="Запустить"/> <input type="button" value="Посмотреть состояние"/>

Рисунок 11.18 – Управление сенсорами

### Filter Список

Id	Xml Rules	Device
1	<root> <network>192.168.1.3</network> <port>8081</port> <protocol>8</protocol> </root>	c9850

Рисунок 11.19 – Справочник фильтр Список фильтров

### Создать Filter

Xml Rules

```
<root>
<network>192.168.1.1</network>
<port>8080</port>
<protocol>6</protocol>
</root>
```

Device:

Рисунок 11.20 – Справочник фильтр Добавить фильтров

### Показать Filter

Id	1
Xml Rules	<root> <network>192.168.1.3</network> <port>8081</port> <protocol>8</protocol> </root>
Device	c9850

Рисунок 11.21 – Справочник фильтр Отражение фильтров

Таким образом, рассмотренные алгоритмы позволяют в реальном масштабе

времени проводить мониторинг нестационарных во времени или неоднородных в пространстве информационных процессов с целью выявления аномалий сетевого трафика.

Представленная методика может быть реализована в качестве курсовой работы или быть основой при разработке дипломной работы.

## Список использованных источников

- 1 Айфичер, Э. Цифровая обработка сигналов. Практический подход / Э. Айфичер, Б. Джервис . – М.: Издательский дом "Вильямс", 2004. – 992 с . – ISBN 5-8459-0710-1
- 2 Быстрое преобразование Фурье. Теория, программа, разъяснение [электронный ресурс]. – Режим доступа: <http://psi-logic.narod.ru/fft/fft2.htm>
- 3 Вейвлет-сжатие «на пальцах» [электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/168517/>
- 4 Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вуд. – М.: Техносфера, 2005 . – 1072 с. – ISBN 5-94836-028-8
- 5 Гонсалес Р. Принципы распознавания образов / Р. Гонсалес, Дж. Ту. – М.: Изд. «Мир», 1978. – 411 с.
- 6 Давыдов, А.В. Цифровая обработка сигналов: тематические лекции / А.В. Давыдов. – Екатеринбург: УГГУ, 2007 [электронный ресурс]. – Режим доступа: <http://www.prodav.narod.ru/dsp/index.html>.
- 7 Дьяконов, В.П. Вейвлеты. От теории к практике / В.П. Дьяконов. – М.: СОЛОН-Р, 2002. – 448 с.
- 8 Журавлев, Ю.И. Об алгебраическом подходе к решению задач распознавания и классификации / Ю.И. Журавлев // Проблемы кибернетики. – М.: Наука. – 1978, вып. 33. – С. 5 – 68.
- 9 Как бороться с репостами или пара слов о перцептивных хешах [электронный ресурс]. – Режим доступа: <http://special.habrahabr.ru/kyocera/p/237307/>
- 10 Кирьянов, Д. В. Самоучитель Mathcad / Д.В. Кирьянов. – СПб.: БХВ-Петербург, 2003. – 560 с: ил. ISBN 5-94157-348.0 с. 452-456
- 11 Компьютерный анализ и интерпретация эмпирических зависимостей: учебник / С.В. Поршневу, Е.В. Овечкина, М.В. Мащенко, А.В. Каплан и др. – М.: ООО «Бином-Пресс», 2009. – 336 с.
- 12 Корн, Г. Справочник по математике для научных работников и инженеров /

Г. Корн, Е. Корн. – М.: Наука, 1984.

13 Макс, Ж. Методы и техника обработки сигналов при физических измерениях / Ж. Макс . – В 2-х томах. – М.: Мир, 1983.

14 Методы спектрального анализа в задаче обнаружения аномалий информационных процессов телекоммуникационных сетей: монография / Н.А. Соловьев, А.С. Цыганков, Н.А. Тишина, Л.А. Юркевская, Е.Н. Чернопрудова . – Оренбург: ООО ИПК «Университет», 2013. – 171 с. - ISBN 978-5-4417-0330-7

15 Очков, В. Ф. Mathcad 14 для студентов и инженеров: русская версия / В. Ф. Очков . – СПб. : БХВ-Петербург, 2009 . – 498 с. : ил . – Прил.: с. 451-492 . – Предм. указ.: с. 495-498 . – Библиогр.: с. 493 . – ISBN 978-5-9775-0403-4.

16 Программная генерация звуков [электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/126835/>

17 Рогозинский Г. Г. Перцепционное кодирование звука на основе вейвлетной компрессии / Г.Г. Рогозинский // Материалы научно-технических конференций студентов и аспирантов институтов и факультетов СПбГУКиТ. – СПб.: ГУКиТ, 2009. – С. 163 – 164

18 Сергиенко, А.Б. Цифровая обработка сигналов / А.Б. Сергиенко. – СПб.: БХВ-Петербург, 2011. – 608 с.

19 Солонина, А.И. Основы цифровой обработки сигналов: учебное пособие / А.И. Солонина. – СПб.: БХВ Петербург, 2005. – 768 с.

20 Сжатие аудиоданных [электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Сжатие\\_аудиоданных](http://ru.wikipedia.org/wiki/Сжатие_аудиоданных)

21 Сжатие звука [электронный ресурс]. – Режим доступа: <http://life-prog.ru/vadim.php?id=34&page=2>

22 Сжатие с использованием вейвлет [электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Сжатие\\_с\\_использованием\\_вейвлет](http://ru.wikipedia.org/wiki/Сжатие_с_использованием_вейвлет)

23 Соловьев, Н.А. Обнаружение вторжений на основе вейвлет-анализа сетевого трафика / Н.А. Соловьев, Н.А. Тишина, И.Г. Дворовой // Вестник УГАТУ: научно-практический журнал.– Уфа. – 2010. – Т .14, №5(40) . – С. 188-194. – ISSN 1992-6502

24 Стретт, Дж.В. (лорд Рэлей) "Теория звука": пер. с англ. / В.Дж. Стретт. – 2 изд., М.: Государственное издательство технико-теоретической литературы, 1955. – 476 с.

25 Файн, В.С. Оpozнaвание изображений. Основы непрерывно-групповой теории и ее приложения / В.С. Файн. – М: Изд-во «Наука», 1970. – 299 с.

26 Цифровая обработка изображений в информационных системах: учебное пособие / И.С.Грузман, В.С. Киричук, В.П. Косых, Г.И. Перетягин, А.А. Спектор. – Новосибирск: Изд-во НГТУ, 2002. – 352 с.

27 Уотерс, Дэн Генерация звуковых волн с помощью волнового осциллятора на C# [электронный ресурс] / Ден Уотерс. – Режим доступа: <http://blogs.msdn.com/b/rucoding4fun/archive/2010/05/27/c.aspx>

Учебное пособие

Николай Алексеевич Соловьев  
Наталья Александровна Тишина  
Любовь Аркадьевна Юркевская

# ЦИФРОВАЯ ОБРАБОТКА ИНФОРМАЦИИ В ЗАДАЧАХ И ПРИМЕРАХ

ISBN 978-5-7410-1614-5



9 785741 016145