

Министерство образования Республики Беларусь

Учреждение образования  
«Полоцкий государственный университет»

Р. П. Богуш

## **ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ**

Учебно-методический комплекс  
для студентов специальности 1-40 02 01  
«Вычислительные машины, системы и сети»

Новополоцк  
ПГУ  
2009

УДК 004(075.8)  
ББК 32.973я73  
Б73

Рекомендовано к изданию методической комиссией  
радиотехнического факультета в качестве учебно-методического  
комплекса (протокол № 6 от 20.06.2008)

РЕЦЕНЗЕНТЫ:  
ведущий инженер Новополоцкого филиала РУП «Белтелеком»  
В. П. ЕРМАШКЕВИЧ;  
зав. каф. радиоэлектроники, канд. техн. наук С. В. МАЛЬЦЕВ

**Богуш, Р. П.**

Б73 Цифровая обработка сигналов и изображений : учеб.-метод. комплекс для  
студентов спец. 1-40 02 01 «Вычислительные машины, системы и сети» /  
Р. П. Богуш. – Новополоцк : ПГУ, 2009. – 320 с.  
ISBN 978-985-418-812-6.

Представлен курс лекций по цифровой обработке сигналов и изображений: рассмотрены методы и алгоритмы предварительной обработки сигналов и изображений в пространственной и частотной областях; методы и алгоритмы сжатия изображений; технологии построения классификаторов для распознавания образов и основные методы распознавания образов. Представлены методические указания к выполнению лабораторных работ. Предлагается система оценки знаний студентов.

Предназначен для студентов и преподавателей факультета информационных технологий.

**УДК 004(075.8)  
ББК 32.973я73**

**ISBN 978-985-418-812-6**

© Богуш Р. П., 2009  
© УО «Полоцкий государственный университет», 2009

## **ВВЕДЕНИЕ**

Цифровая обработка сигналов (ЦОС) – это область науки и техники, в которой изучаются общие для различных технических приложений принципы, методы и алгоритмы обработки сигналов и изображений средствами цифровой вычислительной техники [2].

Под сигналом понимают физическую величину, изменяющуюся во времени, содержащую некоторую информацию, и пригодную для обработки.

Известно, что главная цель обработки физических сигналов заключается в необходимости получения содержащейся в них информации, которая обычно присутствует в амплитуде сигнала, в частоте или в спектральном составе, в фазе или в относительных временных зависимостях нескольких сигналов [2].

Цифровая информация, представляющая индивидуальные звуковые каналы, мультиплексируется во времени (многоканальный доступ с временным разделением, TDMA) или использует технологию доступа с кодовым разделением каналов (CDMA).

Другим примером необходимости обработки сигналов является сжатие полосы частот сигнала (без существенной потери информации) с последующим форматированием и передачей информации на пониженных скоростях, что позволяет сузить требуемую полосу пропускания канала. В качестве примеров можно привести высокоскоростные модемы, которые широко используют алгоритмы устранения избыточности данных, цифровые системы мобильной связи, алгоритмы сжатия статических изображений JPEG и JPEG 2000, системы записи звука и видео MPEG, цифровое телевидение и телевидение высокой четкости (HDTV).

На практике в сигнале, содержащем информацию, присутствует шум, поэтому используется предварительная обработка, основной целью которой является улучшение отношения сигнал/шум. Для решения данной задачи применяются такие методы, как фильтрация, автокорреляция, свертка и т. д., часто используются для выполнения этой задачи и в аналоговой, и в цифровой областях.

Известно, что сигналы могут быть обработаны с использованием аналоговых методов (ASP), цифровых методов (DSP) или комбинации аналоговых и цифровых методов (MSP).

Преимущества ЦОС [4]:

- высокая гибкость, т. е. системы ЦОС можно перепрограммировать на выполнение различных функций без изменения оборудования;

- гарантированная точность, которая определяется числом используемых битов;
- совершенная воспроизводимость, поскольку отсутствуют отклонения, обусловленные устойчивостью отдельных составляющих;
- высокая надежность, малые размеры, низкое энергопотребление.

К основному недостатку следует отнести высокие временные затраты при обработке широкополосных сигналов.

В настоящее время ЦОС широко используется для обработки речи, например, распознавание и синтез речи, озвучивание текста; обработка изображений, например, улучшение и синтез изображений, машинное зрение и распознавание образов, обработка динамических изображений; в инструментальных средствах, например, спектральный анализ, контроль положения и скорости, сжатие информации и др.

*Цель и задачи дисциплины.*

Основная цель курса «Цифровая обработка сигналов и изображений»: изучение студентами методов и алгоритмов обработки сигналов и изображений, а также получение навыков по их программной реализации для решения прикладных задач обработки изображений и распознавания образов.

В результате изучения дисциплины студенты *должны знать*:

- основные положения теории цифровой обработки сигналов и изображений;
- методы и алгоритмы предварительной обработки сигналов и изображений;
- методы и алгоритмы сжатия сигналов и изображений;
- технологии построения классификаторов для распознавания образов.

Студенты *должны уметь*:

- использовать специальные алгоритмы цифровой обработки сигналов и изображений для корреляционной обработки, цифровой фильтрации и спектрального анализа;
- оценивать вычислительную и пространственную сложность алгоритмов цифровой обработки сигналов;
- разрабатывать программное обеспечение для обработки изображений в приложении к решению различных прикладных задач;
- применять основные методы и алгоритмы цифровой обработки сигналов и изображений для построения цифровых распознающих комплексов;

Студенты *должны иметь представление*:

- о преимуществах и областях использования цифровой обработки сигналов и изображений;

- об актуальных прикладных задачах, включающих или имеющих этапы распознавания объектов;
- о направлениях, перспективах и проблемах развития цифровой обработки сигналов и изображений.

*Структура дисциплины.*

Согласно учебному плану специальности 1-40 02 01 «Вычислительные машины, системы и сети», курс «Цифровая обработка сигналов и изображений» изучается студентами на 4 курсе, рассчитан на 64 часа лекций и 32 часа лабораторных работ. Ниже представлено распределение курса по видам аудиторных занятий, по разделам и темам.

## **РАБОЧАЯ ПРОГРАММА**

### **Лекционный курс**

Наименование разделов и тем лекций и их содержание	Кол-во часов
Введение	2
Раздел 1. Основные понятия и определения	
1.1. Обобщенная схема цифровой обработки сигналов и основные типы сигналов	2
1.2. Типовые дискретные сигналы и их математическое описание. Основная полоса частот и нормирование частоты	2
Раздел 2. Дискретные преобразования и алгоритмы ЦОС	
2.1. Дискретное преобразование Фурье и его свойства	2
2.2. Ключевые операции ЦОС: свертка и корреляция	2
2.3. Алгоритмы быстрого преобразования Фурье	2
2.4. Преобразование Уолша-Адамара	2
2.5. Преобразование Хаара	2
2.6. Элементы вейвлет-преобразования сигналов	2
Раздел 3. Цифровые фильтры	
3.1. Методы математического описания и типы ЦФ	2
3.2. Структуры ЦФ	2
3.3. Частотные характеристики и элементы синтеза ЦФ	2
3.4. Реализация алгоритмов ЦОС на процессорах	2
Раздел 4. Спектральный анализ сигналов	
4.1. Анализ спектра на основе ДПФ	2
4.2. Весовые функции. Связь ДПФ с цифровой фильтрацией при анализе спектра	2
4.3. Корреляционные характеристики дискретных случайных сигналов и их статистические оценки	2

Раздел 5. Введение в цифровую обработку изображений. Обработка бинарных изображений	
5.1. Направления, развитие и области применения цифровой обработки изображений	2
5.2. Элементы зрительного восприятия и цифрового представления	2
5.3. Обработка бинарных изображений	2
Раздел 6. Методы пространственной обработки полутоновых изображений	
6.1. Градационные преобразования	2
6.2. Фильтрация изображений	2
6.3. Улучшение изображений на основе арифметико-логических операций и изменения гистограмм	2
6.4. Сегментация изображений посредством выделения линий и контуров	2
6.5. Связывание контуров. Представление и описание изображений	2
6.6. Обработка изображений с помощью математической морфологии	2
Раздел 7. Обработка и сжатие изображений с использованием дискретных преобразований	
7.1. Двумерное преобразование Фурье. Фильтрация изображений в частотной области	2
7.2. Сжатие изображений. Спектральные методы сжатия	2
7.3. Стандарты сжатия JPEG и JPEG 2000	2
Раздел 8. Сравнение цифровых изображений и обнаружение объектов на изображениях	
8.1. Сравнение изображений	2
8.2. Поиск объектов на изображениях	2
Модуль 9. Распознавание образов	
9.1. Основы распознавания	2
9.2. Основные методы распознавания	2

## Лабораторные работы

Наименование лабораторной работы	Кол-во часов
1. Дискретное преобразование Фурье и ключевые операции ЦОС	4
2. Алгоритмы быстрого преобразования Фурье	4
3. Преобразование Уолша-Адамара	4
4. Реализация цифровых фильтров на TMS 320VC5510	4

5. Градационные преобразования изображений [9]	4
6. Методы фильтрации и улучшения контраста изображений [9]	4
7. Линейные и нелинейные методы контрастирования изображений [9]	4
8. Корреляционные методы сравнения, поиска и распознавания объектов [9]	4

### **Оценка знаний студентов**

Для оценки работы и знаний студентов в рамках курса «Цифровая обработка сигналов и изображений» используется накопительная система. Результирующая оценка выставляется по сумме баллов, которые студент набирает в течение всего учебного года, а также в результате выходного итогового контроля – зачета (в осеннем семестре) и экзамена (в весеннем семестре).

Для получения аттестации студент должен успешно написать контрольную работу – два теоретических вопроса (максимальное количество баллов за коллоквиум – 50, для аттестации – 30 и более) и выполнить все предыдущие лабораторные работы.

### **Распределение баллов по видам занятий**

<i>Осенний семестр</i>			
Вид занятий	Форма оценки учебной активности студента	Максимальное количество баллов по каждой форме оценки	Максимальное количество баллов по каждому виду занятий
Лабораторные работы	Успешное выполнение лабораторной работы на текущем занятии	20	$50 \times 4 = 200$
	Полное оформление отчета	10	
	Защита лабораторной работы	20	
Зачет	Ответы на вопросы в устной либо письменной форме	120	120

Весенний семестр			
Вид занятий	Форма оценки учебной активности студента	Максимальное количество баллов по каждой форме оценки	Максимальное количество баллов по каждому виду занятий
Лабораторные работы	Успешное выполнение лабораторной работы на текущем занятии	20	$50 \times 4 = 200$
	Полное оформление отчета	10	
	Защита лабораторной работы	20	
Экзамен	Ответы на экзаменационные вопросы	220	220

*Примечание.* Аттестации – 100 баллов максимум.

Итоговый зачет по данной дисциплине выставляется в случае, если студент набрал 220 баллов.

Для этого необходимо, например:

- получить две аттестации – минимум 60 баллов;
- по результатам выполнения лабораторных работ набрать не менее 120 баллов;
- получить 40 баллов при сдаче зачета.

Дополнительные баллы предусматриваются за:

- выполнение научно-исследовательской работы по разработке и исследованию методов и алгоритмов ЦОСиИ (до 800 баллов);
- изучение, программную реализацию и подготовку доклада о современных алгоритмах ЦОСиИ, представленных в научных тематических изданиях (до 200 баллов);
- подготовку докладов по методам и алгоритмам ЦОСиИ, не включенным в рабочую программу курса «Цифровая обработка сигналов и изображений» (до 200 баллов).

Итоговая оценка выставляется по шкале:

Оценка	1	2	3	4	5	6	7	8	9	10
Сумма баллов	0 – 289	290 – 369	370 – 419	420 – 469	470 – 419	520 – 59	560 – 599	600 – 639	740 – 779	780 и более

Очевидно, что для получения минимальной положительной оценки – 4 балла, студент должен набрать 420 баллов, а для этого необходимо:

- выполнить все лабораторные работы и представить отчеты – 240 баллов;
- получить все аттестации – минимум 120 баллов;
- получить 60 баллов при сдаче экзамена.

Для получения высшей оценки – 10 баллов – студент должен: проявлять самостоятельную творческую работу на лабораторных занятиях; выступать с самостоятельно подготовленными докладами по тематике учебного курса; иметь необходимые знания и способность самостоятельно и творчески решать задачи цифровой обработки сигналов и изображений в нестандартной ситуации.

Перечень дисциплин с указанием разделов (тем), усвоение которых необходимо для изучения данной дисциплины:

№	Название дисциплины	Раздел, тема
1	Высшая математика	Операции над матрицами. Ряды Фурье. Элементы теории вероятности. Статистический анализ
2	Конструирование программ и языки программирования	Язык программирования С
3	Объектно-ориентированное программирование	Язык программирования C++

## **МОДУЛЬ 1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ**

Цель модуля – изучение студентами основных понятий цифровой обработки сигналов, включая модель системы ЦОС, типовые дискретные сигналы и их математическое описание.

В результате изучения модуля студенты *должны знать*:

- принцип построения схемы цифровой обработки сигналов;
- основные этапы ЦОС и их особенности;
- основные типы сигналов.

Студенты *должны уметь*:

- математически описать дискретный сигнал;
- применять теорему Котельникова для определения частоты дискретизации сигналов;
- а также иметь представление о нормировании частоты сигналов.

### **1.1. Обобщенная схема цифровой обработки сигналов и основные типы сигналов**

#### **1.1.1. Обобщенная схема цифровой обработки сигналов**

Системы ЦОС непосредственно оперируют цифровыми сигналами, которые обрабатываются процессором ЦОС. Поэтому, алгоритмическая обработка аналоговых сигналов цифровыми средствами предполагает их предварительное преобразование в цифровую форму, а в системах с аналоговым выходом – и из цифровой формы в аналоговую. Общей структурной схеме системы цифровой обработки аналоговых сигналов (рис. 1.1) отвечает цепочка функциональных преобразований сигнала вида [11]:  $A / A \Rightarrow A / \mathcal{I} \Rightarrow \mathcal{I} / \mathcal{I} \Rightarrow \mathcal{I} / A \Rightarrow A / A$  («аналог/аналог», «аналог/цифра», «цифра/цифра», «цифра/аналог», «аналог/аналог»), реализуемых соответственно аналоговым фильтром низких частот ФНЧ1, аналого-цифровым преобразователем АЦП, процессором ЦОС, цифроаналоговым преобразователем ЦАП и аналоговым фильтром низких частот ФНЧ2.

Совокупность элементов ФНЧ1, АЦП, ЦАП и ФНЧ2 системы цифровой обработки аналоговых сигналов, выполняющих преобразования сигналов вида  $A / A$ ,  $A / \mathcal{I}$  и  $\mathcal{I} / A$ , образуют подсистему ее аналогового ввода-вывода или аналого-цифровой интерфейс.

С преобразованиями сигналов при цифровой обработке связаны искажения и погрешности, которые зависят от параметров соответствующих преобразователей. Оценка и выбор их основываются на математических моделях и описаниях сигналов систем ЦОС.

Обобщенная схема и временные диаграммы поэтапного процесса ЦОС представлены на рис. 1.1.

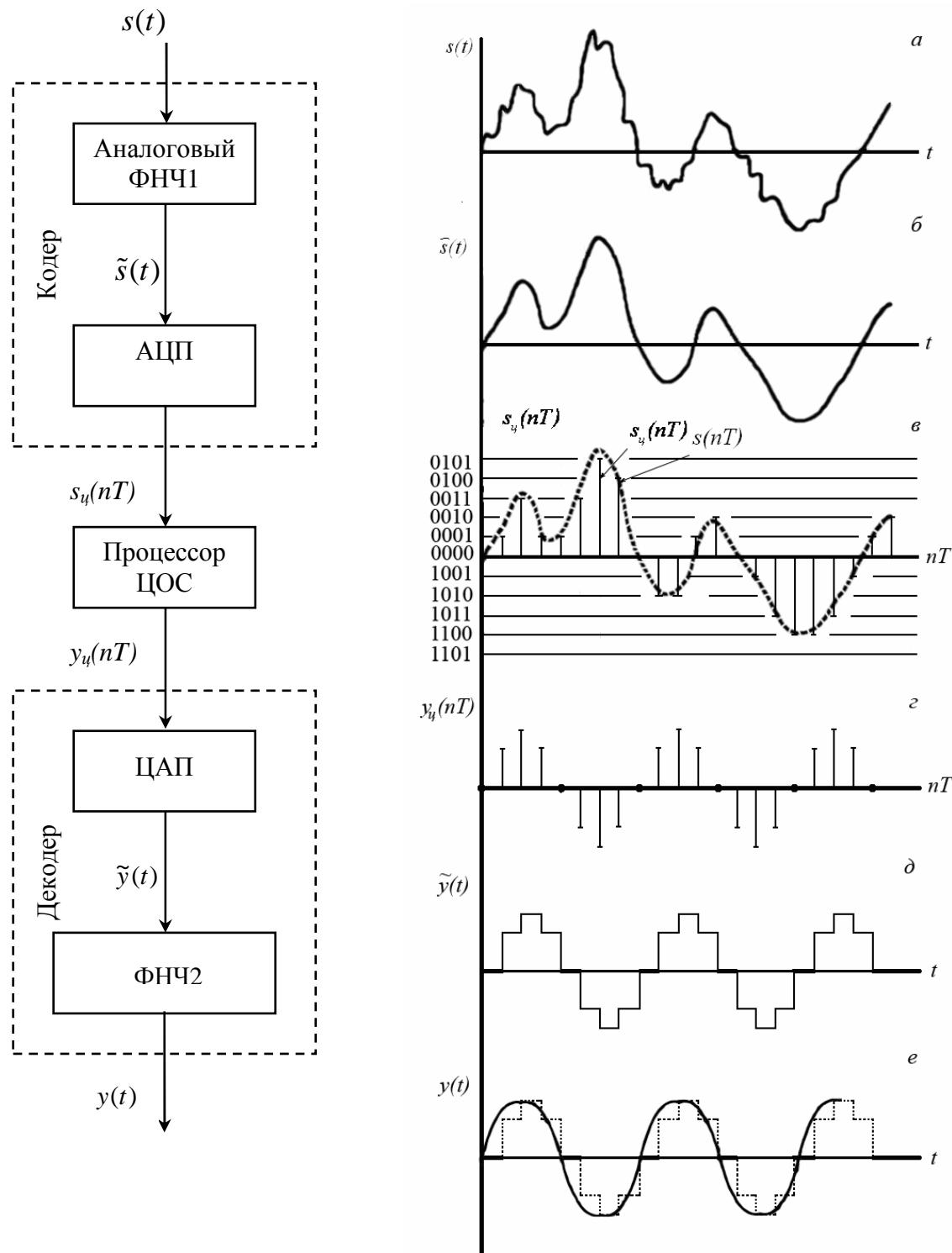


Рис. 1.1. Обобщенная схема цифровой обработки сигнала

На первом этапе кодер, включающий аналоговый фильтр нижних частот (ФНЧ) и аналогово-цифровой преобразователь (АЦП), формирует из исходного аналогового сигнала  $s(t)$  (рис. 1.1, *a*) цифровой сигнал  $s_u(nT)$  (рис. 1.1, *б*). Аналоговый фильтр нижних частот предназначен для ограничения спектра  $F(j\omega)$  исходного аналогового сигнала  $s(t)$ . Необходимость ограничения спектра вытекает из теоремы Котельникова, в соответствии с которой частота дискретизации  $f_D$  выбирается из условия:  $f_D \geq 2f_e$ , где  $f_e$  – верхняя частота спектра сигнала. Возможность ограничения спектра связана с особенностями частотного распределения энергии сигнала: основная часть его энергии сосредоточена в области  $f_e \leq f$ , т. е. амплитуды спектральных составляющих, начиная с некоторой частоты  $f > f_e$ , существенно снижаются. Выбор значения  $f_e$  определяется конкретным типом сигнала и решаемой задачей. При обработке аудио- и видеосигналов выбор  $f_e$  зависит от особенностей психофизического восприятия этих сигналов. Например, для стандартного телефонного сигнала верхняя частота  $f_e$  равна 3,4 кГц, а минимальная стандартная частота дискретизации  $f_D = 8$  кГц.

Аналого-цифровой преобразователь формирует цифровой сигнал  $s_u(nT)$  посредством дискретизации и квантования сигнала  $\tilde{s}(t)$  с интервалом времени, равным периоду дискретизации  $T$ . Значения отсчетов  $s(nT)$  совпадают со значениями сигнала  $\tilde{s}(t)$  в моменты времени  $t = nT$ :

$$s(nT) = \tilde{s}(t)|_{t=nT}.$$

Совокупность отсчетов  $s(nT)$ ,  $n = 0, 1, \dots$  называют дискретным сигналом.

Квантование по уровню (квантование) производится с целью представления точных значений отсчетов  $s(nT)$  в виде двоичных чисел конечной разрядности – квантованных отсчетов  $s_u(nT)$ . Для этого динамический диапазон дискретного сигнала  $s(nT)$  разбивается на конечное число дискретных уровней – уровней квантования – и каждому отсчету по определенному правилу присваивается значения одного из ближайших уровней, между которыми он оказывается. Уровни квантования кодируются двоичными числами разрядности  $b$ , зависящей от числа уровней квантования  $R$ :

$$R \leq 2^b,$$

где  $b = \text{int}(\log_2 R)$ .

На временной диаграмме (рис. 1.1, *в*) для примера показано 5 уровней квантования (без учета знака), поэтому  $b = 3$  и отсчеты  $s_u(nT)$  кодируются четырехразрядными двоичными числами: один разряд знаковый, три значащих.

Совокупность квантованных кодированных отсчетов  $s_u(nT)$ ,  $n = 0, 1, \dots$  называют цифровым сигналом.

На втором этапе устройство ЦОС преобразует цифровой сигнал  $s_u(nT)$  (рис. 1.1, в) в цифровой сигнал  $y_u(nT)$  (рис. 1.1, г) по заданному алгоритму.

Устройство ЦОС может быть реализовано аппаратно или программно. В первом случае – в виде специализированного цифрового устройства, во втором – в виде программы на компьютере или цифровом процессоре обработки сигналов (ЦПОС). В настоящее время программная реализация ЦОС значительно преобладает над аппаратной.

Устройства ЦОС могут работать в реальном или нереальном времени.

В реальном времени обработка сигналов должна выполняться в темпе поступления отсчетов входного сигнала  $s_u(nT)$ ,  $n = 0, 1, \dots$  и отвечать следующим требованиям:

- время цикла  $\Delta t_n$  при вычислении  $y_u(nT)$  не должно превышать интервала между двумя соседними отсчетами  $s_u(nT)$ , т. е. периода дискретизации  $T$  ( $\Delta t_n \leq T$ );
- тактовая частота  $f_m$  процессора должна быть много выше частоты дискретизации  $f_d$  сигнала  $s_u(nT)$  ( $f_m \gg f_d$ ).

Второе требование вызвано тем, что в алгоритмах ЦОС количество операций в цикле, необходимое для вычисления одного отсчета  $y_u(nT)$ , весьма велико. Например, для стандартного телефонного сигнала с частотой дискретизации 8 кГц тактовая частота должна быть не менее 6 мГц.

В реальном времени выполняется обработка сигналов, связанная с их передачей по каналам связи, в том числе, по сети Internet. К типовым задачам ЦОС в реальном времени относятся: обнаружение, фильтрация, сжатие, распознавание сигналов и др.

В нереальном времени выполняется обработка сигналов, связанная, прежде всего, с их исследованием. К типовым задачам ЦОС в нереальном времени относятся: студийная обработка аудио- и видеосигналов; обработка данных различной физической природы, полученная от датчиков, и др.

На третьем этапе декодер формирует результирующий аналоговый сигнал  $y(t)$  из цифрового сигнала  $y_u(nT)$ . В состав декодера входят цифроаналоговый преобразователь (ЦАП) и сглаживающий фильтр. Цифроаналоговый преобразователь формирует из цифрового сигнала  $y_u(nT)$  (рис. 1.1, г) ступенчатый аналоговый сигнал  $\tilde{y}(t)$  (рис. 1.1, д). Сглаживающий фильтр (низкочастотный) устраняет ступенчатый эффект (скачки) в выходном сигнале ЦАП  $\tilde{y}(t)$ . На выходе сглаживающего фильтра получаем аналоговый сигнал  $y(t)$  (рис. 1.1, е) – результат преобразования исходного сигнала  $s(t)$ .

### **1.1.2. Основные типы сигналов и их математическое описание**

Для аналитического описания сигналов и процессов их преобразований используют математические модели. Если сигналы рассматривают как функции, заданные в физических координатах, то говорят об одномерных сигналах (например, зависящих от времени), двумерных, заданных на плоскости (например, изображениях), трехмерных (характеризующих, например, трехмерные пространственные объекты). В теории сигналов в качестве математических моделей сигналов используются: скалярные, комплексные и векторные функции. Например, для описания электромагнитного поля как сигнала удобно использовать комплексные функции, для описания цветных изображений – трехкомпонентные векторные функции, для описания данных многоспектральной съемки – 4 – 6-компонентные векторные функции.

Важными общими характеристиками сигналов как математических функций являются множества значений, которые могут принимать они сами и их аргументы. С этой точки зрения целесообразно различать финитные и инфинитные, ограниченные и неограниченные, непрерывные, дискретные, квантованные и цифровые сигналы.

Финитными называются сигналы, область определения которых ограничена [16]. Например, финитным является сигнал, характеризующий кадр фотоснимка, значение электромагнитного поля в раскрыве антенны и т. п. Инфинитные сигналы имеют неограниченную область определения. Финитный сигнал можно превратить в инфинитный, если доопределить его значения за пределами заданного интервала. Однако любое доопределение должно выполняться так, чтобы при обработке сигнала не изменились его свойства внутри области определения.

Если ограничена область значений сигнала, то сигнал называется ограниченным. Так, например, ограниченным является сигнал, характеризующий степень почернения фотонегатива или распределение яркости свечения экрана телевизионного монитора. Фактически все встречающиеся в природе сигналы являются ограниченными. Неограниченный сигнал – идеализация, принимаемая в тех случаях, когда ограничение области значений сигнала несущественно для данной задачи или просто неизвестно.

С точки зрения математического описания сигналов различают также детерминированное и вероятностное описания. При детерминированном описании сигналы рассматриваются индивидуально, независимо друг от друга, и считается, что значение сигнала может быть задано в каждой точке, где он определен. Однако иногда индивидуальное рассмотрение характеристик физических объектов невозможно, а можно измерить и учесть только некоторое число «макропараметров», характеризующих объекты в среднем. В этих случаях ис-

пользуется вероятностное описание, т. е. сигналы рассматриваются как выборочные функции, или реализации из некоторого ансамбля сигналов, и строится математическое описание не каждого отдельного сигнала, а ансамбля в целом.

Сигналы также подразделяют на дискретные, непрерывные и дискретно-непрерывные. Сигнал считают дискретным по данному параметру, если число значений, которое может принимать этот параметр, конечно (или счетно). Непрерывными называются сигналы, область определения и область значений которых непрерывны, т. е. для каждой точки области определения и области значения можно найти точку, удаленную от нее на бесконечно малое расстояние.

Сигнал, дискретный по одному параметру и непрерывный по другому, называют дискретно-непрерывным.

Как математическая модель используются:

- непрерывная функция непрерывного аргумента (рис. 1.2);
- непрерывная функция дискретного аргумента, например функция, значения которой отсчитывают только в определенные моменты времени (рис. 1.3). Временной интервал между соседними отсчетами называется шагом дискретизации;
- дискретная функция непрерывного аргумента, например функция времени, квантованная по уровню (рис. 1.4);
- дискретная функция дискретного аргумента, принимающая одно из конечного множества возможных значений в дискретные моменты времени (рис. 1.5).

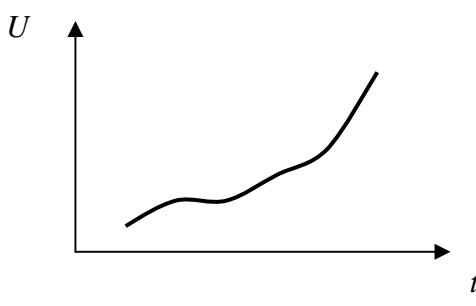


Рис. 1.2. Модель непрерывного сигнала

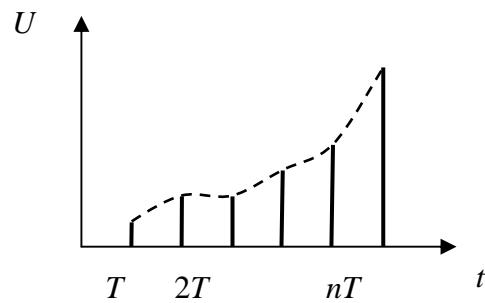


Рис. 1.3. Модель дискретного по времени сигнала

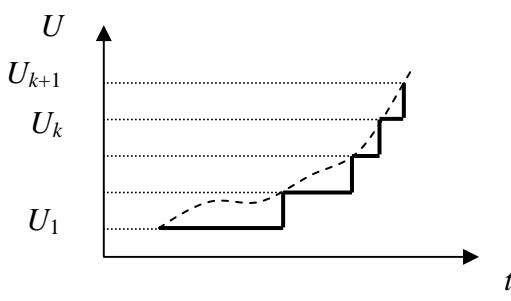


Рис. 1.4. Модель квантованного сигнала

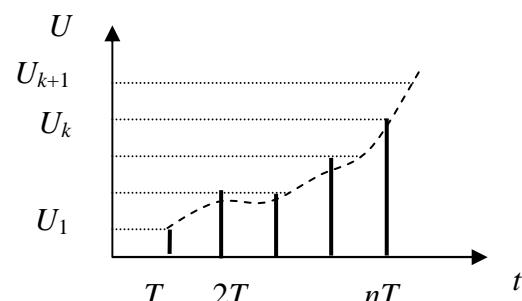


Рис. 1.5. Модель дискретного сигнала

Дискретный сигнал может быть как вещественным, так и комплексным. В последнем случае его вещественная и мнимая части описываются вещественными последовательностями:

$$s(nT) = s_1(nT) + js_2(nT).$$

При описании дискретных и цифровых сигналов используют такое понятие, как нормированное время  $\hat{t}$ :

$$\hat{t} = \frac{t}{T},$$

которое при  $t = nT$  имеет смысл номера отсчета  $n$ :

$$\hat{t} = \frac{t}{T} = \frac{nT}{T} = n, \quad (1.2)$$

и означает, что отсчет взят в момент  $nT$ .

Это позволяет описывать дискретный сигнал функцией целочисленной переменной  $s(n)$  и считать тождественными обозначения дискретного сигнала  $s(n)$  и  $s(nT)$ :

$$s(nT) = s(n).$$

## **1.2. Типовые дискретные сигналы и их математическое описание. Основная полоса частот и нормирование частоты**

### **1.2.1. Типовые дискретные сигналы [11]**

Цифровой единичный импульс (дельта-импульс), описываемый последовательностью:

$$u_0(n) = \begin{cases} 1, & n = 0; \\ 0, & n \neq 0, \end{cases}$$

т. е. этот сигнал равен единице при  $n = 0$  и нулю при всех остальных значениях  $n$  (рис. 1.6, а).

Задержанный цифровой единичный импульс описывается последовательностью:

$$u_0(n - m) = \begin{cases} 1, & n = m; \\ 0, & n \neq m, \end{cases}$$

т. е. этот сигнал, в отличие от незадержанного, равен единице при  $n = m$  и нулю при всех остальных значениях  $n$  (рис. 1.4, б).

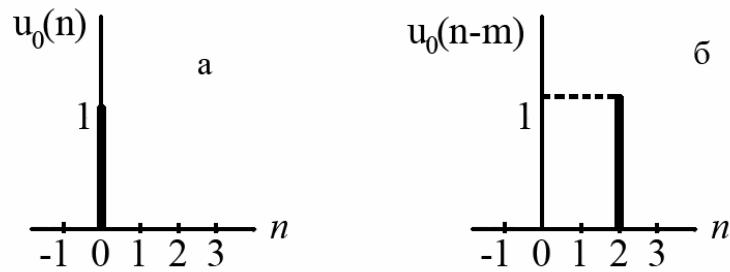


Рис. 1.6. Цифровой единичный (а) и задержанный цифровой единичный (б) импульсы

Цифровой единичный скачок, описываемый последовательностью:

$$u_1(n) = \begin{cases} 1, & n \geq 0; \\ 0, & n < 0, \end{cases}$$

т. е. этот сигнал равен единице при всех неотрицательных значениях  $n$  (рис. 1.7, а).

Задержанный цифровой единичный скачок (рис. 1.7, б) описывается последовательностью:

$$u_1(n-m) = \begin{cases} 1, & n \geq m; \\ 0, & n < m, \end{cases}$$

т. е. этот сигнал, в отличие от незадержанного, равен единице при всех значениях  $n \geq m$  и нулю при остальных значениях  $n$ .

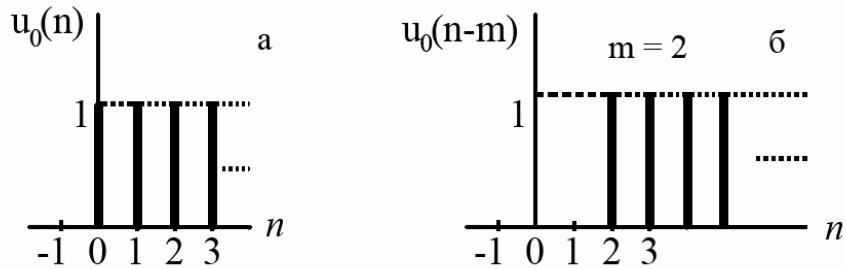


Рис. 1.7. Цифровой единичный (а) и задержанный цифровой единичный (б) скачки

Дискретная экспонента, описываемая последовательностью:

$$s(n) = \begin{cases} a^n, & n \geq 0; \\ 0, & n < 0, \end{cases}$$

где  $a$  – вещественная константа.

В зависимости от величины и знака  $a$  дискретная экспонента будет:

- $|a|<1$  и  $a>0$  – убывающей знакопостоянной (рис. 1.8,  $\alpha$ );
- $|a|<1$  и  $a<0$  – убывающей знакопеременной (рис. 1.8,  $\beta$ );
- $|a|>1$  – возрастающей;
- $|a|=1$  и  $a>0$  – цифровым единичным скачком;
- $|a|=1$  и  $a<0$  – знакопеременной последовательностью единиц.

Дискретный гармонический сигнал, например, дискретная косинусоида, описываемая последовательностью:

$$s(nT) = s(n) = A \cos(2\pi f n T) = A \cos(\omega n T),$$

где  $T$  – период дискретизации;

$A$  – амплитуда;

$\omega$  – круговая частота, связанная с частотой  $f$  коэффициентом пропорциональности  $2\pi$  ( $\omega = 2\pi f$ ).

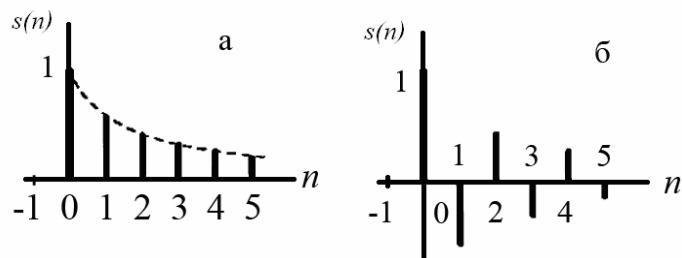


Рис. 1.8. Знакопостоянная ( $\alpha$ ) и знакопеременная ( $\beta$ ) дискретные экспоненты

Дискретная косинусоида получается из аналоговой в результате замены непрерывного времени дискретным (рис. 1.9):

$$s(t) = A \cos(2\pi f t) = A \cos(\omega t),$$

$$s(nT) = s(n) = A \cos(\omega t) |_{t=nT} = A \cos(\omega T n).$$

Дискретная синусоида описывается аналогично.

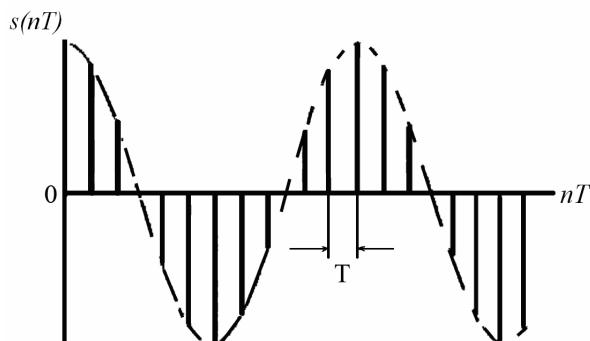


Рис. 1.9. Дискретная косинусоида

Дискретный комплексный гармонический сигнал, описываемый комплексной последовательностью:

$$s(n) = A e^{j\omega T n},$$

или двумя вещественными последовательностями: косинусоидой (вещественная часть) и синусоидой (мнимая часть):

$$s(nT) = A \cos(\omega T n) + j A \sin(\omega T n).$$

### 1.2.2. Основная полоса частот. Нормирование частоты

Согласно теореме Котельникова, верхняя частота  $f_e$  аналогового сигнала не должна превышать половины частоты дискретизации  $f_D$  этого сигнала. Следовательно, дискретные сигналы целесообразно рассматривать в области  $\left[0; \frac{f_D}{2}\right]$ , которая называется основной полосой частот или основным диапазоном частот [11].

Это позволяет ввести понятие нормированных частот:

- 1)  $\hat{f}$  – отношение текущей частоты  $f$  к частоте дискретизации  $f_D$ :

$$\hat{f} = \frac{f}{f_D} = fT;$$

- 2)  $\hat{\omega}$  – отношение текущей частоты  $\omega$  к частоте дискретизации  $f_D$ :

$$\hat{\omega} = \frac{\omega}{f_D} = \omega T;$$

- 3) нормирование частоты  $\hat{f}$  и  $\hat{\omega}$  связаны отношением:

$$\hat{\omega} = \frac{\omega}{f_D} = \frac{2\pi f}{f_D} = 2\pi \hat{f}.$$

В зависимости от выбранной шкалы частот основная полоса соответствует областям:

$$\begin{aligned} f &\rightarrow \left[0; \frac{f_D}{2}\right]; \\ \hat{f} &\rightarrow [0; 0.5]; \\ \omega &\rightarrow \left[0; \frac{\omega_D}{2}\right] = \left[0; \frac{\pi}{T}\right]; \\ \hat{\omega} &\rightarrow [0; \pi]. \end{aligned}$$

Используются абсолютная частота  $f$  и нормированная частота  $\hat{\omega}$ .

Для косинусоиды в области нормированных частот выражение имеет вид:

$$s(n) = A \cos(2\pi \hat{f}n) = A \cos(\hat{\omega}n).$$

Введение нормированной частоты указывает на то, что в ЦОС важны не абсолютные значения частот сигнала и дискретизации, а их отношение.

Пример двух дискретных косинусоид [11]:

$$s_1(n) = \cos(2\pi f_1 T_1 n) = \cos\left(2\pi \frac{f_1}{f_{D1}} n\right) \text{ при } f_1 = 4 \text{ Гц}, f_{D1} = 16 \text{ Гц};$$

$$s_2(n) = \cos(2\pi f_2 T_2 n) = \cos\left(2\pi \frac{f_2}{f_{D2}} n\right) \text{ при } f_2 = 10 \text{ кГц}, f_{D2} = 40 \text{ кГц}.$$

Подставив значения  $f_1$  и  $f_D$ , получим в области нормированных частот одинаковые дискретные сигналы:

$$s_1(n) = \cos(2\pi \frac{4}{16} n) = \cos\left(\frac{\pi}{2} n\right);$$

$$s_2(n) = \cos(2\pi \frac{10000}{40000} n) = \cos\left(\frac{\pi}{2} n\right).$$

### 1.3. Вопросы и задания для самопроверки

1. Назовите и обоснуйте основные преимущества и недостатки ЦОС по сравнению с системами аналоговой обработки сигналов.
2. Что такое сигнал и что понимают под цифровой обработкой сигналов?
3. Представьте структурную схему ЦОС и поясните принцип ее работы.
4. Сформулируйте теорему Котельникова и ее назначение.
5. Какие сигналы называют одномерными, двухмерными и трехмерными? Приведите примеры.
6. Дайте определение дискретному, непрерывному и дискретно-непрерывному сигналу, приведите примеры.
7. Назовите типовые дискретные сигналы и опишите их математически.
8. Для каких целей в ЦОС используется нормирование частоты?
9. В каких областях используется цифровая обработка сигналов?

## **МОДУЛЬ 2. ДИСКРЕТНЫЕ ПРЕОБРАЗОВАНИЯ И АЛГОРИТМЫ ЦОС**

Цель модуля – изучение студентами дискретных преобразований применяемых при разработке алгоритмов цифровой обработки сигналов.

В результате изучения модуля студенты *должны знать*:

- дискретные преобразования: Фурье, Адамара, Хаара, вейвлет-преобразование;
- быстрые алгоритмы дискретных преобразований;
- ключевые операции ЦОС.

Студенты *должны уметь*:

- разрабатывать эффективные алгоритмы цифровой обработки сигналов на основе дискретных преобразований;
- оценивать вычислительную и пространственную сложность алгоритмов цифровой обработки сигналов;
- а также иметь представление о видах вейвлетов и возможности их применения для решения конкретных задач ЦОС.

### **2.1. Дискретное преобразование Фурье и его свойства**

#### **2.1.1. Основы преобразования Фурье**

Выражение регулярного сигнала определенной функцией времени называют временным представлением сигнала. В частности, при некоторых ограничениях, функция времени, заданная на некотором отрезке времени, может быть представлена в виде тригонометрического ряда, каждый член которого является простейшей гармонической функцией времени (косинус, синус). Эти функции называются гармониками и каждой из них принадлежат определенные амплитуда, частота и фаза.

Множество амплитуд, частот и фаз называют спектром рассматриваемого сигнала. Функция времени находится в однозначном соответствии с принадлежащим ей спектром. На этом основании временное представление сигнала может быть заменено так называемым частотным представлением. Оба представления адекватны.

Представление периодического сигнала суммой гармонических составляющих осуществляется с помощью его разложения в ряд Фурье. Если функция (сигнал)  $f(t)$  задана на интервале времени  $t_1 \leq t \leq t_2$  и повторяется с периодом  $T_p = 2\pi/\omega$ , то тригонометрическая форма ряда Фурье для нее может быть записана следующим образом:

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

Амплитуды косинусоидальных и синусоидальных членов определяются выражениями:

$$a_n = \frac{2}{T_p} \int_0^{T_p} f(t) \cos(n\omega t) dt ;$$

$$b_n = \frac{2}{T_p} \int_0^{T_p} f(t) \sin(n\omega t) dt .$$

Слагаемое  $\frac{a_0}{2} = \frac{A_0}{2} = \frac{1}{T_p} \int_0^{T_p} f(t) dt$  является постоянной составляющей

сигнала, которая равна среднему значению функции  $f(t)$  за период.

Частоты  $n\omega$  называют  $n$ -ми гармониками частоты  $\omega$ . Следовательно, бесконечный ряд содержит зависящие от частоты синусоидальные и косинусоидальные члены с различными амплитудами на положительных частотах гармоник.

Амплитуда  $D_n$  и фаза  $\psi_n$   $n$ -ой гармоники связаны с величинами  $a_n$  и  $b_n$  соотношениями:

$$D_n = \sqrt{a_n^2 + b_n^2}, \quad \psi_n = \arctg(b_n/a_n).$$

Весьма удобной является комплексная форма записи ряда Фурье, к которой легко перейти, если выразить тригонометрические функции через показательные, воспользовавшись известными формулами Эйлера:

$$\cos n\omega t = \frac{1}{2}(e^{jn\omega t} + e^{-jn\omega t});$$

$$\sin n\omega t = \frac{1}{2j}(e^{jn\omega t} - e^{-jn\omega t}).$$

В результате получим:

$$f(t) = \frac{a_0}{2} + \frac{1}{2} \sum_{n=1}^{\infty} (A_n e^{jn\omega t} + A_n^* e^{-jn\omega t}),$$

где  $A_n$  и  $A_n^*$  – комплексные амплитуды, связанные с  $a_n$  и  $b_n$  соотношениями

$$A_n e^{-j\psi_n} = a_n - jb_n,$$

$$A_n^* e^{j\psi_n} = a_n + jb_n.$$

Таким образом, комплексные амплитуды  $A_n$  и  $A_n^*$  являются комплексно-сопряженными величинами. Действительно каждое слагаемое ряда можно представить как вектор на комплексной плоскости (рис. 2.1), вращающийся с частотой  $n\omega$  (т. е. в положительном направлении отсчета углов – против направления движения часовой стрелки). Каждое слагаемое второго ряда – вектор, вращающийся в обратном направлении.

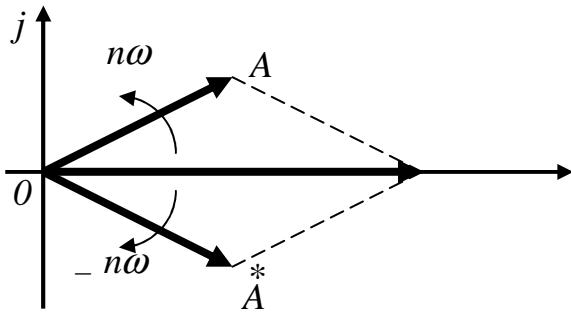


Рис. 2.1. Векторная диаграмма комплексно-сопряженных величин

Так как  $A_n$  и  $A_n^*$  – комплексно-сопряженные величины, то сумма векторов в любой момент времени дает вектор, направленный по вещественной оси, т. е.  $n$ -ю гармоническую составляющую вещественной функции времени  $f(t)$ . Отрицательная частота ( $-n\omega$ ) только указывает направление вращения вектора.

Таким образом, комплексная амплитуда  $D_n$  может быть определена согласно выражению

$$D_n = \frac{1}{T_p} \int_0^T f(t) e^{-jn\omega t} dt.$$

Следовательно, в экспоненциальной форме ряд Фурье имеет вид:

$$f(t) = \sum_{n=1}^{\infty} D_n e^{jn\omega t}.$$

При такой записи ряда Фурье периодический сигнал заменяется суммой простых гармонических колебаний как с положительными частотами ( $n > 0$ ), так и с отрицательными ( $n < 0$ ). Конечно, отрицательные частоты не имеют здесь физического смысла, а являются формальным следствием произведенного математического преобразования.

## 2.1.2. Дискретные экспоненциальные функции

Дискретный сигнал  $s(n)$ , заданный на интервале  $[0, \infty)$  можно записать в виде [8]:

$$s(n) = \sum_{k=0}^{\infty} c(k) \eta_k(n),$$

$$\text{где } c(k) = E_k^{-1} \sum_{n=0}^{N-1} s(n) \eta_k(n).$$

Здесь  $\{\eta_k(n)\}$  – совокупность базисных функций, определенных на том же интервале;  $\{c(k)\}$  – коэффициенты разложения сигнала по базису  $\{\eta_k(n)\}$ , называемые спектром сигнала;  $E_k$  – энергия  $k$ -й базисной функции.

Из выражений следует, что сигнал можно задать либо его отсчетами в дискретных точках, либо набором спектральных коэффициентов, причем спектральное представление неоднозначно и зависит от выбранной системы базисных функций. Выбор системы  $\{\eta_k(n)\}$  определяется соображениями практического или математического удобства.

Математические преобразования применяются к сигналу для того, чтобы получить о нем какую-то дополнительную информацию, недоступную в исходном виде. Среди ряда известных преобразований сигналов наиболее часто используется на практике преобразование Фурье.

Дискретное преобразование Фурье оперирует дискретной по времени выборкой периодического сигнала во временной области  $\{s(n)\} = \{s(0), s(1), \dots, s(N-1)\}$ . Для того, чтобы быть представленным в виде суммы синусоид, сигнал должен быть периодическим. Но в качестве набора входных данных для ДПФ доступно только конечное число отсчетов ( $N$ ).

В дискретном преобразовании Фурье используется система дискретных экспоненциальных функций (ДЭФ), которая определяется следующим выражением:

$$def(k, n) = \exp\left(-j \frac{2\pi}{N} kn\right) = \cos\left(\frac{2\pi}{N} kn\right) - j \sin\left(\frac{2\pi}{N} kn\right),$$

$$\text{где } j = \sqrt{-1};$$

$k$  и  $n$  – принимают целочисленные значения  $(0, 1, 2, \dots, N-1)$ .

Переменную  $k$  отождествляют с номером функции, а переменную  $n$  – с номером отсчета.

Если обозначить  $W = \exp(-j\frac{2\pi}{N})$ , то  $\text{def}(k, n) = W^{kn} = W_N^{k \cdot n}$ .

Всю систему ДЭФ можно записать в виде матрицы  $V$ , строки которой нумеруются переменной  $k$ , столбцы переменной  $n$ , а в пересечении  $k$ -й строки и  $n$ -го столбца записана величина  $W^{kn}$ . Такое представление называется матричным представлением ДЭФ.

$$V = \begin{bmatrix} 0 & & & & & | & & & \\ 1 & & & & & | & & & \\ \vdots & & & & & | & & & \\ k & \cdots & \cdots & \cdots & W^{kn} & \cdots & \cdots & \\ \vdots & & & & | & & & \\ N-1 & & & & | & & & \end{bmatrix}$$

0    1    .     $n$     ..     $N-1$

Например, для  $N = 8$  матрица  $V$  имеет следующий вид:

$$V = \begin{bmatrix} W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^8 & W^{10} & W^{12} & W^{14} \\ W^0 & W^3 & W^6 & W^9 & W^{12} & W^{15} & W^{18} & W^{21} \\ W^0 & W^4 & W^8 & W^{12} & W^{16} & W^{20} & W^{24} & W^{28} \\ W^0 & W^5 & W^{10} & W^{15} & W^{20} & W^{25} & W^{30} & W^{35} \\ W^0 & W^6 & W^{12} & W^{18} & W^{24} & W^{30} & W^{36} & W^{42} \\ W^0 & W^7 & W^{14} & W^{21} & W^{28} & W^{35} & W^{42} & W^{49} \end{bmatrix}.$$

Основные свойства ДЭФ [8]:

1. Ортогональность:

$$\sum_{n=0}^{N-1} W^{kn} W^{-ln} = \begin{cases} N, & \text{если } (k - l) \equiv 0 \pmod{N}; \\ 0, & \text{если } (k - l) \neq 0 \pmod{N}. \end{cases}$$

Свойство ортогональности показывает, что скалярное произведение любых двух строк матрицы  $V$ , одна из которых взята с комплексно сопряженными элементами, равно нулю, если строки различны, и равно  $N$ , если они совпадают. Матричная запись этого свойства имеет следующий вид:

$$V(V^T)^* = NI,$$

где знак  $*$  – означает взятие комплексного сопряжения для всех элементов матрицы ( $I$  – единичная матрица).

2. *Периодичность.* Если  $kn = Nl + r$ , то  $W^{kn} = W^{Nl}W^r = W^r$ , что позволяет записать элементы матрицы  $V$  с минимальными степенями (фазами). Например, для  $N = 8$ :

$$V = \begin{bmatrix} W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^1 & W^4 & W^7 & W^2 & W^5 \\ W^0 & W^4 & W^0 & W^4 & W^0 & W^4 & W^0 & W^4 \\ W^0 & W^5 & W^2 & W^7 & W^4 & W^1 & W^6 & W^3 \\ W^0 & W^6 & W^4 & W^2 & W^0 & W^6 & W^4 & W^2 \\ W^0 & W^7 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix}.$$

3. *Симметричность:*

$$V = V^T.$$

Свойство симметричности позволяет легко найти обратную матрицу для матрицы  $V$ . Матричное выражение для ортогональности запишется:

$$VV^* = NI.$$

Умножив обе части этого равенства справа на  $V^{-1}$ , получим:

$$V^{-1} = N^{-1}V^*.$$

4. *Мультипликативность:*

– по номеру функции:

$$\text{def}(k_1, n)\text{def}(k_2, n) = \text{def}(k_1 + k_2, n);$$

– по номеру отсчета:

$$\text{def}(k, n_1)\text{def}(k, n_2) = \text{def}(k, n_1 + n_2).$$

Свойство означает, что при умножении любых двух строк (столбцов) матрицы  $V$  получается соответственно строка (столбец) той же матрицы, причем номер строки (столбца) равен сумме номеров сомножителей.

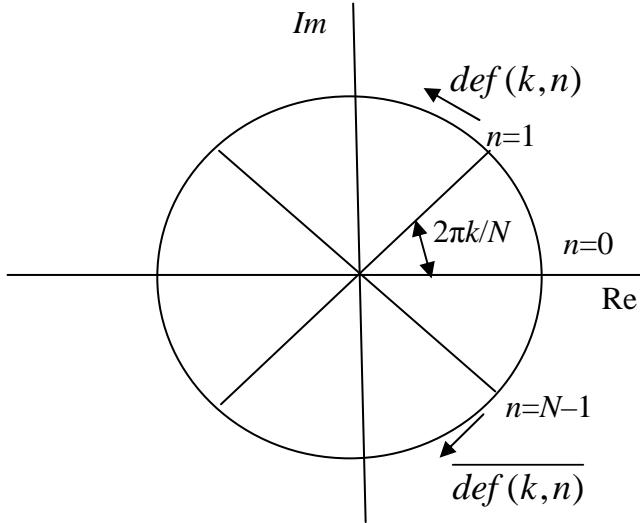


Рис. 2.2. Представление ДЭФ в виде вектора

Подобно обычным комплексным экспоненциальным функциям, ДЭФ можно изобразить на плоскости в виде вращающегося вектора единичной длины, проекции которого на оси абсцисс и ординат дают действительную и мнимую части функции (рис. 2.2). Разница будет заключаться в том, что если у обычных функций этот вектор вращается непрерывно, то в случае ДЭФ он вращается скачкообразно, проходя при изменении  $n$  на единицу угол  $2\pi k/N$  рад. В общей сложности на интервале  $N$  вектор проходит угол  $2\pi k$  рад, т. е. совершает ровно  $k$  оборотов. Движение комплексно-сопряженного вектора  $\overline{def(k,n)} = def(N - k, n)$  при изменении  $n$  на единицу можно представить как поворот на один оборот и возврат обратно на угол  $2\pi k/N$  рад, так, что создается иллюзия вращения в противоположную сторону (стробоскопический эффект). Такому представлению соответствует запись системы ДЭФ в виде матрицы с полными фазами, а матрица с минимальными фазами не дает представления об истинной скорости вращения вектора. Поэтому величину  $W^{kn}$  называют *поворачивающим множителем*.

По формуле Эйлера можно выразить значения вещественной и мнимой частей поворачивающих множителей через косинус и синус угла поворота  $\theta$  соответственно:

$$e^{-j\theta} = \cos(\theta) - j \sin(\theta).$$

Более того, при значениях  $k \geq \frac{N}{2}$  значения поворачивающих множителей можно найти из соображения симметрии:

$$e^{-j\frac{2\pi(\frac{N}{2}+k)}{N}} = e^{-j\pi} e^{-j\frac{2\pi k}{N}} = -e^{-j\frac{2\pi k}{N}}, \text{ при } 0 \leq k \leq \frac{N}{2}.$$

Вычисление поворачивающегося множителя для 8-ми точечной последовательности показано на рис. 2.3, а значения поворачивающегося множителя приведены в табл. 2.1.

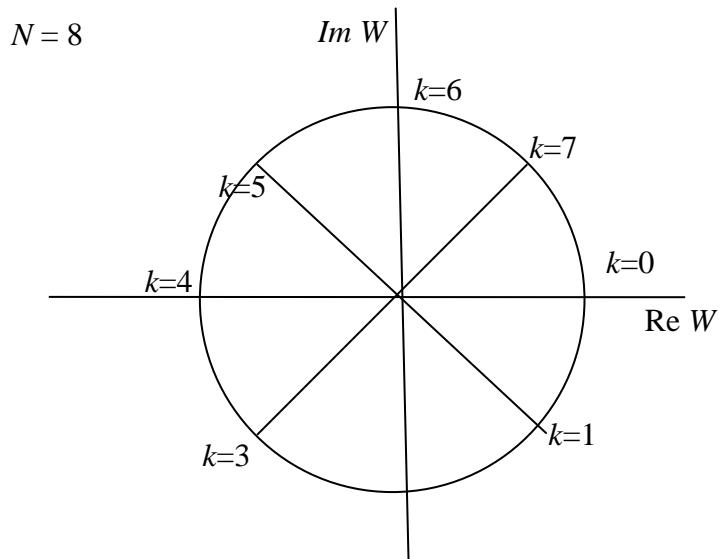


Рис. 2.3. Представление поворачивающегося множителя для восьмиточечной последовательности

Таблица 2.1

**Значения поворачивающегося множителя  
для восьмиточечной последовательности**

Значения $k$	Поворачивающий множитель
0	1
1	$\cos\left(\frac{\pi}{4}\right) - j \sin\left(\frac{\pi}{4}\right)$
2	$-j$
3	$-\cos\left(\frac{\pi}{4}\right) - j \sin\left(\frac{\pi}{4}\right)$
4	-1
5	$-\cos\left(\frac{\pi}{4}\right) + j \sin\left(\frac{\pi}{4}\right)$
6	$j$
7	$\cos\left(\frac{\pi}{4}\right) + j \sin\left(\frac{\pi}{4}\right)$

### 2.1.3. Дискретное преобразование Фурье и его свойства

Пара дискретного преобразования Фурье последовательности  $\{s(n)\} = \{s(0), s(1), \dots, s(N-1)\}$  определяется следующими равенствами [8]:

$$f(k) = \sum_{n=0}^{N-1} s(n)W^{kn}, \quad k = 0, 1, 2, \dots, N-1; \quad (2.1)$$

$$s(n) = N^{-1} \sum_{k=0}^{N-1} f(k)W^{-kn}, \quad n = 0, 1, 2, \dots, N-1; \quad (2.2)$$

где  $\{f(k)\}$  – дискретный спектр.

Выражение (2.1) называется прямым преобразованием, а выражение (2.2) – обратным. Равенства (2.1) и (2.2) представляют собой экспоненциальную форму записи ДПФ. Как уже было отмечено, ДПФ устанавливает связь между временными и частотным представлениями сигнала при разложении в базисе гармонических функций.

В матричной форме ДПФ имеет вид:

$$F = VS,$$

$$S = N^{-1}V^*F.$$

Матрицы прямого и обратного преобразований  $V$  и  $V^*$  соответственно состоят из комплексно-сопряженных элементов  $W^{kn}$  и  $W^{-kn}$ . Для  $k = n = 0$  строки обеих матриц совпадают и состоят только из единиц. Поэтому  $s(0) = N^{-1}f(0)$ . Для  $k \neq 0$  можно записать следующее равенство:

$$W^{(N-k)n} = W^{Nn}W^{-kn} = W^{-kn}.$$

Из него следует, что строки матрицы  $V^*$  совпадают со строками матрицы  $V$ , но записаны в обратном порядке. Поэтому для вычисления обратного ДПФ вместо  $V^*$  можно использовать матрицу прямого преобразования  $V$ , но результат умножения на эту матрицу следует прочесть в обратном порядке и уменьшить в  $N$  раз.

Основные области применения ДПФ:

1) цифровой спектральный анализ:

- анализаторы спектра;
- обработка речи;
- обработка изображений;
- распознавание образов;
- проектирование фильтров;
- вычисление импульсной характеристики по частотной;
- вычисление частотной характеристики по импульсной;

2) быстрое преобразование Фурье (БПФ) – простой алгоритм для эффективного вычисления ДПФ.

Пример ДПФ для  $\{s(n)\} = \{1, 0, 0, 1\}$ .

Матрица ДЭФ для выполнения ДПФ входного сигнала с количеством отсчетов  $N = 4$  имеет вид:

$$V = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^0 & W^2 \\ W^0 & W^3 & W^2 & W^1 \end{bmatrix}.$$

Поворачивающийся множитель  $W^{kn}$  для  $N = 4$  на комплексной плоскости представлен на рис. 2.4.

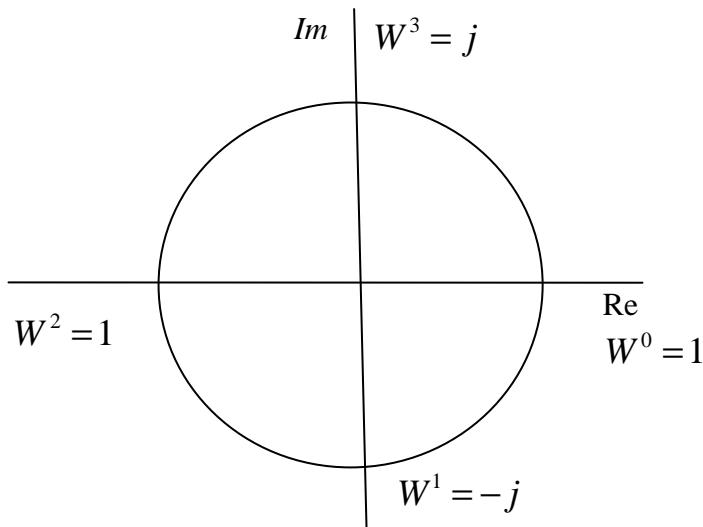


Рис. 2.4. Представление поворачивающегося множителя для восьмиточечной последовательности

Следовательно:

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T = \begin{bmatrix} 2 \\ 1+j \\ 0 \\ 1-j \end{bmatrix}^T.$$

Таким образом, временному ряду  $\{1,0,0,1\}$  соответствует его ДПФ образ вида  $\{2,1+j,0,1-j\}$ .

ДПФ принято представлять в виде графиков зависимости  $|f(k)|$  от номера гармоники  $k$  или от частоты  $\omega$ . Чтобы определить  $\omega$ , необходимо знать значение интервала дискретизации. Если предположить, что представленная выше последовательность была дискретизована с частотой 8 кГц, то  $T = 1 / (8 \cdot 10^3) = 125$  мкс. Тогда  $\omega = 2\pi / (NT) = 2\pi / (4 \cdot 125 \cdot 10^{-6}) = 12,57 \cdot 10^3$  рад/с. Следовательно,  $2\omega = 25,14 \cdot 10^3$  рад/с,  $3\omega = 37,71 \cdot 10^3$  рад/с.

Свойства ДПФ[8]:

*1. Периодичность.* В силу периодичности ДЭФ функции  $f(k)$  и  $s(n)$  также периодичны, т. е.:

$$f(Nl \pm k) = f(\pm k); s(Nl \pm n) = s(\pm n).$$

*2. Связь с коэффициентами ряда Фурье.* Если частота дискретизации выбрана в соответствии с теоремой Котельникова, то при дискретизации периодической аналоговой функции  $s(t)$  ДПФ позволяет по выборкам  $s(n)$  найти спектр  $f(k)$ , который на интервале  $0 \leq k \leq N - 1$  равен спектру исходной функции  $s(t)$ . При этом первые  $\left(\frac{N}{2} - 1\right)$  точек функции  $f(k)$  соответствуют спектральным линиям на положительных частотах, а последние  $\left(\frac{N}{2} - 1\right)$  точек  $f(k)$  – спектральным линиям на отрицательных частотах. В обратном преобразовании первые  $\left(\frac{N}{2} - 1\right)$  линий функции  $s(n)$  соответствуют области положительных времен, а последние  $\left(\frac{N}{2} - 1\right)$  линий – области отрицательных времен.

*3. Линейность.* Пусть даны последовательности  $x(n)$  и  $y(n)$ , для которых ДПФ равны соответственно  $f_x(k)$  и  $f_y(k)$ . Рассмотрим взвешенную сумму этих последовательностей  $z(n) = ax(n) + by(n)$ . Спектр последовательности  $z(n)$  равен аналогичной взвешенной сумме спектров последовательностей  $x(n)$  и  $y(n)$ , т. е.:

$$f_z(k) = af_x(k) + bf_y(k).$$

*4. Инвариантность относительно сдвига по времени и частоте.*

Пусть последовательность  $z(n)$  образована сдвигом по времени последовательности  $s(n)$ , т. е.  $z(n) = s(n \pm h)$ . Тогда:

$$f_z(k) = W^{\mp kh} f_s(k).$$

Данное свойство показывает, что при сдвиге по времени амплитудный спектр (величина амплитуд отдельных гармоник) не меняется. Изменениям подвергаются только фазы гармонических составляющих (фазовый спектр).

Аналогичное свойство справедливо и для обратного преобразования, а именно: если  $l(k) = f(k \pm h)$ , то:

$$S_F(n) = W^{\pm nh} s_l(n).$$

*5. Теорема о свертке.* Пусть  $\{f_s(k)\}$  и  $\{f_h(k)\}$  – спектры последовательностей  $\{s(n)\}$  и  $\{h(n)\}$  соответственно. Теорема о свертке утверждает, что спектр свертки равен произведению спектров сворачиваемых последовательностей, т. е.:

$$f_y(k) = f_s(k) f_h(k). \quad (2.3)$$

Матричное представление свертки имеет вид:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(1) & \dots & \dots & \dots & h(N-1) \\ h(1) & h(2) & \dots & \dots & \dots & h(0) \\ h(2) & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ h(N-1) & h(0) & \dots & \dots & \dots & h(N-2) \end{bmatrix} \begin{bmatrix} s(0) \\ s(N-1) \\ s(N-2) \\ \vdots \\ s(1) \end{bmatrix}.$$

Теорема позволяет вычислить свертку при помощи ДПФ по формуле:

$$\{y(n)\} = \mathcal{D}\Phi^{-1}(\mathcal{D}\Phi\{s(n)\} \cdot \mathcal{D}\Phi\{h(n)\}).$$

*6. Теорема о корреляции.* Спектр корреляционной функции последовательностей  $\{s(n)\}$  и  $\{h(n)\}$  равен произведению их спектров, причем один из спектров берется в комплексном сопряжении, т. е.:

$$f_r(k) = f_h^*(k) f_s(k) = f_h(k) f_s^*(k).$$

Матричное представление корреляции:

$$\begin{bmatrix} r(0) \\ r(1) \\ \cdot \\ \cdot \\ r(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(1) & \dots & h(N-1) \\ h(N-1) & h(0) & \dots & h(N-2) \\ \dots & \dots & \dots & \dots \\ h(1) & h(2) & \dots & h(0) \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ \cdot \\ \cdot \\ s(N-1) \end{bmatrix}.$$

7. *ДПФ вещественных последовательностей*. Определяется согласно выражению:

$$f(k) = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{2\pi}{N} kn\right) - j \sum_{n=0}^{N-1} s(n) \sin\left(\frac{2\pi}{N} kn\right).$$

Отсюда особенности:

а) спектральные коэффициенты комплексно сопряжены относительно счета  $N/2$ , т. е.:

$$f\left(\frac{N}{2} + l\right) = f^*\left(\frac{N}{2} - l\right), \quad \ell = 0, 1, \dots, N/2,$$

а коэффициент  $f(N/2)$  для четных  $N$  всегда действителен;

б) если последовательность  $\{s(n)\}$  четная, т. е.  $\{s(n)\} = \{s(-n)\}$ , то ее ДПФ  $\{f(k)\}$  – вещественная последовательность. Аналогично, если  $\{s(n)\}$  нечетная, т. е.  $\{s(n)\} = -\{s(-n)\}$ , то ее ДПФ  $f(k)$  есть чисто мнимая последовательность.

8. *Равенство Парсеваля*:

$$\sum_{n=0}^{N-1} s^2(n) = N^{-1} \sum_{k=0}^{N-1} |f(k)|^2.$$

Данное соотношение показывает, что энергия сигнала равна суммарной энергии спектральных компонент, т. е. энергия сигнала больше никуда не расходуется.

## 2.2. Ключевые операции ЦОС: свертка и корреляция

### 2.2.1. Импульсная и переходная характеристики

Дискретная система преобразует входной дискретный сигнал  $s(nT)$  в выходной дискретный сигнал  $y(nT)$ :

$$y(nT) = F[s(nT)],$$

где  $F$  – оператор, определяющий математическое преобразование (линейное или нелинейное алгебраическое, дифференциальное и т. д.).

Дискретная система называется стационарной, если ее реакция инвариантна по отношению к началу отсчета времени (свойство инвариантности во времени), т. е. для реакции  $y(nT)$  и  $y_1(nT) = y[(n-m)T]$  при любом целом  $m$  справедливо равенство [11]:

$$y(nT) = y_1[(n+m)T].$$

Параметры стационарной системы неизменны во времени.

В стационарной системе задержка воздействия па время  $mT$  ( $m > 0$ ) приводит к задержке реакции на то же время:

$$s[(n-m)T] \Rightarrow y[(n-m)T].$$

В дискретной системе начальные условия могут быть нулевыми или ненулевыми.

Для нулевых начальных условий является отсутствие реакции  $y(nT) = 0$  при отсутствии воздействия  $s(nT) = 0$ . Для ненулевых начальных условий является наличие значений реакции (свободных колебаний) при отсутствии воздействия.

Во временной области основной характеристикой линейной дискретной системы является импульсная характеристика (ИХ).

Импульсной характеристикой  $h(nT)$  дискретной системы называется ее реакция цифровой единичный импульс  $u_0(nT)$  при нулевых начальных условиях [11].

Импульсную характеристику  $h(nT)$  считают основной характеристикой системы, потому что, зная ее, можно определить реакцию на любое (произвольное) воздействие.

Переходной характеристикой  $g(nT)$  линейной дискретной системы называется ее реакция на цифровой единичный скачок  $u_1(nT)$  при нулевых начальных условиях.

Переходная характеристика линейной дискретной системы связана с ее импульсной характеристикой соотношением [11]:

$$g(nT) = \sum_{m=0}^n h(mT).$$

Например, если импульсная характеристика имеет вид убывающей дискретной экспоненты:

$$h(n) = a^n, |a| < 1,$$

то переходная характеристика определяется как сумма конечной убывающей геометрической прогрессии и имеет следующий вид:

$$g(nT) = \sum_{m=0}^n h(mT) = \sum_{m=0}^n a^m = \frac{1-a^n}{1-a}.$$

Зная переходную характеристику  $g(nT)$ , также можно определить реакцию на произвольное взаимодействие.

Во временной области соотношение вход/выход линейной дискретной системы описывается линейными уравнениями:

- формулой свертки (сверткой), если используется импульсная характеристика;
- разностными уравнениями, если используются параметры дискретной системы.

### 2.2.2. Линейная свертка

Свертка – это математический способ комбинирования двух сигналов для формирования третьего сигнала [11]. Это один из самых важных методов ЦОС. Свертка связывает три сигнала: входной сигнал, выходной сигнал и импульсную характеристику системы.

Формулой свертки (сверткой) называют линейное уравнение вида:

$$y(nT) = \sum_{m=0}^{\infty} h[(n-m)T]s(mT), \quad (2.4)$$

где  $h[(n-m)T]$  – импульсная характеристика, задержанная на  $m$  периодов дискретизации.

Таким образом, реакция системы  $y(nT)$  вычисляется как дискретная свертка воздействия  $s(nT)$  и импульсной характеристики  $h(nT)$ .

Выполнив в (2.4) замену переменных, можно записать другой вариант записи формулы свертки:

$$y(nT) = \sum_{m=0}^{\infty} h(mT)s[(n-m)T]. \quad (2.5)$$

Для нормированного времени формулы (2.4) и (2.5) принимают вид соответственно:

$$y(n) = \sum_{m=0}^{\infty} h(n-m)s(m),$$

$$y(n) = \sum_{m=0}^{\infty} h(m)s(n-m).$$

При стандартном обозначении операции свертки, формулы (2.4) и (2.5) записываются в комплексном виде

$$y(n) = s(n) * h(n).$$

Рассмотрим две апериодические последовательности  $s(n)$  длины  $L$ ,  $s(n)=0$  при  $L-1 < n < 0$  и последовательность  $h(n)$  длины  $M$ ,  $h(n)=0$  при  $M-1 < n < 0$ . Апериодическая или линейная свертка этих последовательностей имеет длину  $L+M-1$  и определяется как:

$$y(n) = \sum_{m=0}^{L+M-2} h(m)s(n-m), n=0,1,\dots,L+M-2.$$

В матричной форме свертка при  $L = M = N$  получим

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ \vdots \\ \vdots \\ y(2N-2) \end{bmatrix} = \begin{bmatrix} h(0) & & & & & & & s(0) \\ h(1) & h(0) & & & & & & s(1) \\ h(2) & h(1) & h(0) & & & & & s(2) \\ \cdots & \cdots \\ h(N-1) & h(N-2) & h(N-3) & & & & h(0) & \cdots \\ & h(N-1) & h(N-2) & & & & h(1) & \cdots \\ \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & h(N-1) & h(N-2) \\ y(2N-2) & & & & & & h(N-1) & s(N-1) \end{bmatrix}$$

В большинстве алгоритмов вычисления свертки входная последовательность  $\{h(l)\}$  делится на последовательные блоки по  $N$  – отсчетов и  $y(n)$  вычисляется как сумма линейных сверток каждого из этих блоков с  $L$  – точечной последовательностью  $s(m)$ .

Пример [11]. Вычислить реакцию линейной дискретной системы по формуле свертки при нулевых начальных условиях. Импульсная характеристика и воздействие заданы графически на рис. 2.5 и рис. 2.6. Требуется определить 7 отсчетов реакции.

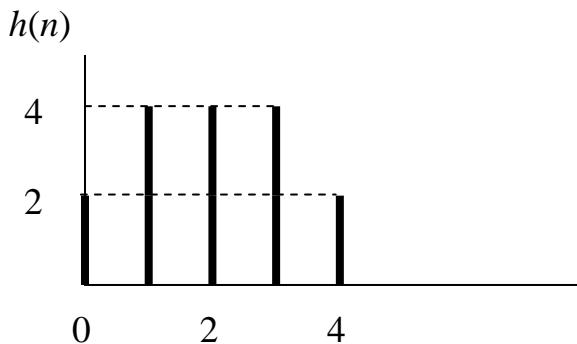


Рис. 2.5. Импульсная характеристика системы

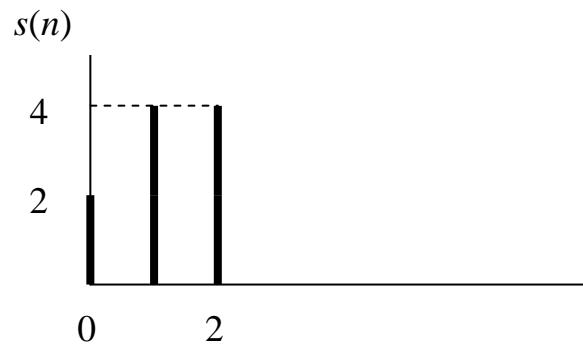


Рис. 2.6. Входной сигнал

Линейная свертка этих последовательностей имеет длину  $L + M - 1 = 5 + 3 - 1 = 7$ . Вычисление свертки представлено в табл. 2.2.

Таблица 2.2

#### Вычисление свертки

$n$	Реакции системы
0	$y(0) = h(0)s(0) + h(1)s(-1) + h(2)s(-2) + \dots = h(0)s(0) = 2 \cdot 2 = 4$
1	$y(1) = h(0)s(1) + h(1)s(0) + h(2)s(-1) + \dots = h(0)s(1) + h(1)s(0) = 2 \cdot 4 + 4 \cdot 2 = 16$
2	$y(2) = h(0)s(2) + h(1)s(1) + h(2)s(0) + h(3)s(-1) + \dots =$ $= h(0)s(2) + h(1)s(1) + h(2)s(0) = 2 \cdot 4 + 4 \cdot 4 + 4 \cdot 2 = 32$
3	$y(3) = h(0)s(3) + h(1)s(2) + h(2)s(1) + h(3)s(0) + h(4)s(-1) + \dots =$ $= h(0)s(3) + h(1)s(2) + h(2)s(1) + h(3)s(0) = 2 \cdot 0 + 4 \cdot 4 + 4 \cdot 4 + 4 \cdot 2 = 40$
4	$y(4) = h(0)s(4) + h(1)s(3) + h(2)s(2) + h(3)s(1) + h(4)s(0) + h(5)s(-1) + \dots =$ $= h(0)s(4) + h(1)s(3) + h(2)s(2) + h(3)s(1) + h(4)s(0) =$ $= 2 \cdot 0 + 4 \cdot 0 + 4 \cdot 4 + 4 \cdot 4 + 2 \cdot 2 = 36$
5	$y(5) = h(0)s(5) + h(1)s(4) + h(2)s(3) + h(3)s(2) + h(4)s(1) + h(5)s(0) +$ $+ h(6)s(-1) + \dots = h(0)s(5) + h(1)s(4) + h(2)s(3) + h(3)s(2) + h(4)s(1) +$ $+ h(5)s(0) = 2 \cdot 0 + 4 \cdot 0 + 4 \cdot 0 + 4 \cdot 4 + 2 \cdot 4 + 0 \cdot 2 = 24$
6	$y(6) = h(0)s(6) + h(1)s(5) + h(2)s(4) + h(3)s(3) + h(4)s(2) + h(5)s(1) +$ $+ h(6)s(0) = 2 \cdot 0 + 4 \cdot 0 + 4 \cdot 0 + 4 \cdot 0 + 2 \cdot 4 + 0 \cdot 4 + 0 \cdot 2 = 8$

Вычисление отсчетов реакции  $y(n), n = 0, 1, 2, \dots$  по формуле свертки можно представить как вычисление сумм локальных произведений двух последовательностей – импульсной характеристики и воздействия. При этом одна из последовательностей фиксирована, а другая зеркально отображается относительно оси ординат и затем скользит слева направо по оси времени.

### 2.2.3. Циклическая свертка

Циклическая свертка определяется для периодических последовательностей длины  $N$  выражением [8]:

$$y(n) = \sum_{l=0}^{N-1} h(l)s(n-l), \quad n = 0, 1, \dots, N-1.$$

В силу периодичности последовательностей номера отсчетов берутся по модулю  $N$ , поэтому  $s(-n) = s(N-n)$ ,  $h(-n) = h(N-n)$ . Матричная запись циклической свертки имеет вид:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & h(1) & \dots & \dots & \dots & h(N-1) \\ h(1) & h(2) & \dots & \dots & \dots & h(0) \\ h(2) & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ h(N-1) & h(0) & \dots & \dots & \dots & h(N-2) \end{bmatrix} \begin{bmatrix} s(0) \\ s(N-1) \\ s(N-2) \\ \vdots \\ s(1) \end{bmatrix}.$$

Пример. Вычисление трехточечной циклической свертки:

$$\begin{aligned} y(0) &= h(0)s(0) + h(1)s(2) + h(2)s(1); \\ y(1) &= h(1)s(0) + h(0)s(2) + h(2)s(1); \\ y(2) &= h(2)s(0) + h(1)s(2) + h(0)s(1). \end{aligned}$$

### 2.2.4. Вычисление сверток при помощи дискретных преобразований

Согласно теореме о свертке для преобразования Фурье, спектр свертки равен произведению спектров сворачиваемых последовательностей. Поскольку все рассмотренные преобразования обладают аналогичными свойствами, то их можно объединить понятием обобщенного преобразования Фурье. Тогда свертка двух векторов [8]:

$$\begin{aligned} S &= [s(0), s(1), \dots, s(N-1)]^T, \quad H = [h(0), h(1), \dots, h(N-1)]^T, \\ y(n) &= \mathcal{D}\mathcal{P}\Phi^{-1}(\mathcal{D}\mathcal{P}\Phi\{s(n)\} \cdot \mathcal{D}\mathcal{P}\Phi\{h(n)\}). \end{aligned}$$

Таким образом, для получения свертки следует выполнить следующие действия:

- 1) найти преобразования Фурье (обобщенные спектры) входных последовательностей;
- 2) вычислить поточечное произведение этих последовательностей;
- 3) вычислить обратное преобразование Фурье от произведения спектров.

Может показаться, что такой метод вычисления свертки довольно сложен, он тем не менее позволяет во многих случаях сократить объем вычислений. Это происходит вследствие того, что для выполнения БПФ существуют быстрые алгоритмы с числом операций, пропорциональным  $N \log_2 N$ . Для многих приложений один из векторов (например,  $H$ ) известен заранее, что позволяет предварительно вычислить спектр  $H$ . В этом случае вычисление свертки заключается в выполнении двух быстрых преобразований и перемножении  $N$  чисел. Для БПФ это циклические свертки, для преобразования Адамара вычисляется диадная свертка.

Для вычисления линейной свертки двух последовательностей длины  $N_1$  и  $N_2$  исходные данные последовательности следует дополнить нулевыми отсчетами так, чтобы их длина стала равной  $N_1 + N_2 - 1$ , и рассматривать как периодические.

Пример. Вычислить линейную свертку последовательностей  $S = [2, -2, 1]^T$  и  $H = [1, 2]^T$ .

Так как  $N_1 = 3$  и  $N_2 = 2$ , то  $N_1 + N_2 - 1 = 4$ , следует вычислить циклическую свертку последовательностей  $S' = [2, -2, 1, 0]^T$  и  $H = [1, 2, 0, 0]^T$ .

Прямые преобразования Фурье последовательностей равны:

$$F_s = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1+2j \\ 5 \\ 1-2j \end{bmatrix},$$

$$F_h = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1-2j \\ -1 \\ 1+2j \end{bmatrix}.$$

Поточечное произведение спектров:

$$F_h \cdot F_s = \begin{bmatrix} 3 \\ 1-2j \\ -1 \\ 1+2j \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1+2j \\ 5 \\ 1-2j \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 \\ (1-2j)(1+2j) \\ -1 \cdot 5 \\ (1+2j)(1-2j) \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ -5 \\ 5 \end{bmatrix}.$$

Обратное преобразование Фурье дает значения свертки:

$$Y = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 5 \\ -5 \\ 5 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 8 \\ 8 \\ -12 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ -3 \\ 2 \end{bmatrix}.$$

В тех случаях, когда одна последовательность намного длиннее другой, используют разбиение длинной последовательности на короткие секции. Затем вычисляются короткие свертки и из них формируется конечный результат. Такая ситуация встречается в цифровой фильтрации, когда фильтруемые последовательности намного длиннее по сравнению с короткой импульсной характеристикой.

Существует два метода секционирования – метод перекрытия с суммированием и метод перекрытия с накоплением. Предположим, что более длинной является последовательность  $s(n)$ . Она разбивается на блоки по  $N$  отсчетов. Последовательность  $h(n)$  имеет длину  $L$ . Линейная свертка каждого из блоков последовательности  $s(n)$  с последовательностью  $h(n)$  имеет размер  $N+L-1$  и перекрываетя со сверткой следующего блока в  $L-1$  отсчетах. Поэтому на участке перекрытия их отсчеты следует сложить. Таким образом, на каждые  $L$  входных отсчетов вычисляется  $N+L-1$ -точечная циклическая свертка и выполняется  $L$  – сложений.

В методе перекрытия с накоплением длинная последовательность  $s(n)$  разбивается на секции по  $N$  отсчетов так, что соседние секции перекрываются в  $L-1$  отсчетах. Последовательность  $h(n)$  дополняется нулевыми значениями до длины  $N$ , и вычисляются циклические свертки каждой секции с дополненной последовательностью  $h(n)$ . Первые  $L-1$  отсчетов каждой секционной свертки отбрасываются, а остальные присоединяются к оставшимся отсчетам предыдущей секции. Алгоритм перекрытия с накоплением дает  $N-l+1$  отсчетов свертки без дополнительного суммирования, поэтому его реализация проще.

Если длина блока увеличивается, то общее количество блоков уменьшается. При этом число преобразований становится меньше, но время выполнения каждого преобразования возрастает. С другой стороны, сокращение длины блока потребует большего числа коротких преобразований, поэтому существует оптимум между длиной блока  $N$  и длиной короткой последовательности  $L$ .

## 2.2.5. Корреляция

При решении ряда практических задач необходимо определять степень сходства сигналов, например, при обнаружении одного сигнала в другом. Для надежного обнаружения перекрывающихся сигналов и сигналов смеси с шумом применяется метод корреляции.

Пусть датчик генерирует сигнал  $s(n)$ , и мы хотим обнаружить в нем последовательность  $g(n)$  некоторой конечной длины. Для поиска этой последовательности вычисляются скалярные произведения сигналов  $s(t+n)$  и  $g(n)$  для различных  $t$ . Таким образом, необходимо «приложить» искомый сигнал во всех возможных положениях к сигналу с датчика и найти их «степень похожести» (скалярное произведение) для каждого положения. Таким образом, на выходе формируется сигнал  $r(t)$ , показывающий, насколько сигнал с датчика  $s(n)$  в позиции  $t$  похож на искомый сигнал  $g(n)$ .

Смысл сигнала  $r(t)$  в том, что его величины для каждого  $n$  показывают, насколько входной сигнал в позиции  $k$  похож на искомый сигнал. Если во входном сигнале присутствует только шум, то и значения корреляции будут шумом небольшой амплитуды. Но как только в шуме входного сигнала появится форма, похожая на искомый сигнал, так значение корреляции в этой точке станет высоким.

Корреляцию можно вычислять как свертку, положив в качестве ядра свертки искомый сигнал, развернутый относительно нулевой точки. Иногда корреляцию называют кросс-корреляцией или перекрестной корреляцией. Термин «автокорреляция» применяется, когда находится корреляция сигнала с самим собой. Смысл этой операции в том, чтобы найти наиболее вероятные периоды повторения формы исходного сигнала.

Взаимная корреляционная функция (ВКФ) разных сигналов описывает как степень сходства формы двух сигналов, так и их взаимное расположение друг относительно друга по координате (независимой переменной).

Автокорреляционные функции (АКФ) сигналов. АКФ сигнала  $s(n)$ , является количественной характеристикой формы сигнала, и определяется суммой от произведения двух копий сигнала  $s(n)$ , сдвинутых относительно друг друга. АКФ является скалярным произведением сигнала и его копии в функциональной зависимости от переменной величины значения сдвига  $t$ .

Для вычисления корреляции в основном используют два подхода.

*1. Прямой метод.* Вычисление корреляции двух сигналов выполняется во временной области и в данном случае выполняется векторно-матричное произведение входного вектора на матрицу, строки которой представляют собой всевозможные сдвиги исходного сигнала. Соответствующее матричное представление имеет вид [8]:

$$\begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(N-1) \end{bmatrix} = \begin{bmatrix} g(0) & g(1) & \dots & g(N-1) \\ g(N-1) & g(0) & \dots & g(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ r(N-1) & g(1) & g(2) & \dots & g(0) \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ \vdots \\ s(N-1) \end{bmatrix}.$$

*2. Вычисление корреляции на основе двукратных преобразований.* Процедура отыскания корреляции двух сигналов выполняется в частотной области, используя свойство:

$$\begin{aligned} r(n) &= \mathcal{D}\Phi^{-1}(\mathcal{D}\Phi\{s(n)\} \cdot \mathcal{D}\Phi^*\{g(n)\}) = \\ &= \mathcal{D}\Phi^{-1}(\mathcal{D}\Phi^*\{s(n)\} \cdot \mathcal{D}\Phi\{g(n)\}). \end{aligned}$$

Таким образом, для получения корреляции следует выполнить следующие действия:

- 1) найти преобразования Фурье (обобщенные спектры) исходных последовательностей;
- 2) вычислить поточечное произведение этих последовательностей, причем одна из них должна быть взята в комплексно-сопряженной форме;
- 3) вычислить обратное преобразование Фурье от произведения спектров.

При вычислении корреляции можно добиться значительного сокращения вычислительных затрат, применяя быстрые алгоритмы дискретных преобразований.

### 2.3. Алгоритмы быстрого преобразования Фурье

Вычисление дискретного преобразования Фурье требует выполнения  $N^2$  умножения и  $N(N-1)$  операций сложения для сигнала длины  $N$ . Существует способ снизить вычислительную сложность до величины порядка  $N \log_2 N$ . Этот способ называется быстрым преобразованием Фурье (БПФ, FFT(fast Fourier transform)). Он основан на том, что среди множителей есть

много повторяющихся значений (в силу, например, своей периодичности). Алгоритм БПФ группирует слагаемые с одинаковыми множителями, значительно сокращая число умножений. В результате быстродействие БПФ может в зависимости от  $N$  в сотни раз превосходить быстродействие стандартного алгоритма. При этом алгоритм БПФ является точным. Он даже точнее стандартного, т. к. сокращая число операций, он приводит к меньшим ошибкам округления.

Однако у большинства алгоритмов БПФ есть особенность [11]: они способны работать лишь тогда, когда длина анализируемого сигнала  $N$  является степенью двойки. Обычно это не представляет большой проблемы, так как анализируемый сигнал всегда можно дополнить нулями до необходимого размера. Число  $N$  называется размером или длиной БПФ (FFT size).

Наибольшее распространение получил алгоритм БПФ с основанием 2, известный как алгоритм БПФ Кули-Тьюки (по имени разработчиков) и впервые опубликованный в 1965 году в США. Существуют два эквивалентных по эффективности алгоритма БПФ с основанием 2: с прореживанием по времени и с прореживанием по частоте.

### **2.3.1. Алгоритм БПФ с прореживанием по времени**

Основная идея алгоритма БПФ с прореживанием по времени заключается в поэтапном вычислении  $N$ -точечного ДПФ ( $N = 2^v$ ) на  $v$  этапах, на каждом из которых текущее ДПФ определяется как комбинация ДПФ вдвое меньшей размерности.

Алгоритм БПФ с прореживанием по времени в общем виде представляется [11]:

- задание начальных условий: отсчеты  $N$ -точечной последовательности расставляются по определенному правилу;
- на первом этапе определяется 2-точечное ДПФ каждой пары отсчетов последовательности;
- на втором этапе определяются 4-точечные ДПФ как комбинация 2-точечных ДПФ;
- на  $i$ -ом этапе определяются  $2^i$ -точечные ДПФ как комбинация  $2^{i-1}$ -точечных ДПФ;
- на  $(v - 1)$ -ом этапе определяются  $N/2$ -точечные ДПФ как комбинация  $N/4$ -точечных ДПФ;
- на  $v$ -ом (последнем) этапе определяется искомое  $N$ -точечное ДПФ как комбинация  $N/2$ -точечных ДПФ, отсчеты ДПФ следуют в естественном порядке  $k = 0, 1, \dots, (N - 1)$ .

Таким образом, для реализации алгоритма БПФ необходимо знать правило начальной расстановки отсчетов  $N$ -точечной последовательности и общую формулу расчета ДПФ для произвольного  $i$ -го этапа.

Получить их можно следующим образом [11].

Начальные условия одноэтапного алгоритма БПФ формируются в результате однократного разбиения исходной  $N$ -точечной последовательности на две  $N/2$ -точечные, выделив отдельно четные и нечетные отсчеты:

$$\begin{aligned}\{s(2n)\} &= \{s(0), s(2), \dots, s(N-2)\}; \\ \{s(2n+1)\} &= \{s(1), s(3), \dots, s(N-1)\}.\end{aligned}$$

Таким образом, начальная расстановка отсчетов производится по правилу –  $N/2$  четных отсчетов,  $N/2$  нечетных отсчетов:

$$\{s(0), s(2), \dots, s(N-2); s(1), s(3), \dots, s(N-1)\}.$$

Это позволяет разбить сумму в ДПФ на две:

$$\begin{aligned}f(k) &= \sum_{n=0}^{\frac{N}{2}-1} s(2n)W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} s(2n+1)W_N^{(2n+1)k} = \\ &= \sum_{n=0}^{\frac{N}{2}-1} s(2n)W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} s(2n+1)W_N^{2nk}, k = 0, 1, \dots, N-1.\end{aligned}$$

Представим поворачивающий множитель  $W_N^{2nk}$  в виде:

$$W_N^{2nk} = e^{-j\frac{2\pi}{N}2nk} = e^{-j\frac{2\pi}{N/2}nk} = W_{N/2}^{nk}.$$

ДПФ можно записать как:

$$f(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} s(2n+1)W_{N/2}^{nk}, k = 0, 1, \dots, N-1.$$

Анализ выражения показывает, что каждая из сумм представляет собой  $N/2$ -точечное ДПФ: первая сумма – ДПФ последовательности четных отсчетов, а вторая – нечетных. Причем каждое из  $N/2$ -точечных ДПФ определяется при  $k = 0, 1, \dots, N-1$ .

Используем обозначения:

$$f_0^v(k) = f(k),$$

$$f_0^{v-1}(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n) W_{N/2}^{nk},$$

$$f_1^{v-1}(k) = \sum_{n=0}^{\frac{N}{2}-1} s(2n+1) W_{N/2}^{nk},$$

где *верхний* индекс – указывает номер этапа алгоритма БПФ;

*нижний* – номер ДПФ, четный (четное ДПФ) и нечетный (нечетное ДПФ).

С учетом введенных обозначений ДПФ, можно рассчитать  $N$ -точечное ДПФ, оно определяется как комбинация двух  $N/2$ -точечных ДПФ – четного  $f_0^{v-1}(k)$  и нечетного  $f_1^{v-1}(k)$ :

$$f^v(k) = f_0^{v-1}(k) + W_N^k f_1^{v-1}(k), \quad k = 0, 1, \dots, N-1.$$

$N/2$ -точечные ДПФ  $f_0^{v-1}(k)$  и  $f_1^{v-1}(k)$  – периодические функции переменной  $k$  с периодом  $N/2$ , нет необходимости определять их при  $k = 0, 1, \dots, N-1$ , достаточно только определить их при  $k = 0, 1, \dots, \frac{N}{2}-1$ , а затем

повторить при  $k = \frac{N}{2}, \frac{N}{2}+1, \dots, N-1$ , т. е.:

$$f_0^{v-1}(k) = f_0^{v-1}\left(k + \frac{N}{2}\right), \quad k = 0, 1, \dots, \frac{N}{2}-1;$$

$$f_1^{v-1}(k) = f_1^{v-1}\left(k + \frac{N}{2}\right), \quad k = 0, 1, \dots, \frac{N}{2}-1.$$

Поворачивающий множитель при  $k = \frac{N}{2}, \frac{N}{2}+1, \dots, N-1$  равен

$$W_N^{k+\frac{N}{2}} = W_N^k W_N^{N/2} = W_N^k e^{-j \frac{2\pi N}{N/2}} = -W_N^k, \quad k = 0, 1, \dots, \frac{N}{2}-1.$$

Следовательно, и его достаточно определить при  $k = 0, 1, \dots, \frac{N}{2}-1$ , а

затем, изменив знак, повторить при  $k = \frac{N}{2}, \frac{N}{2}+1, \dots, N-1$ .

Рассмотренные свойства позволяют представить ДПФ в виде:

$$\begin{cases} f_0^v(k) = f_0^{v-1}(k) + W_N^k f_1^{v-1}(k); \\ f_0^v(k + \frac{N}{2}) = f_0^{v-1}(k) - W_N^k f_1^{v-1}(k); \\ k = 0, 1, \dots, \frac{N}{2} - 1. \end{cases} \quad (2.6)$$

Анализ полученного выражения показывает, что расчет ДПФ упрощается за счет распараллеливания вычислений при  $k = 0, 1, \dots, \frac{N}{2} - 1$  и  $k = \frac{N}{2}, \frac{N}{2} + 1, \dots, N - 1$  по верхней и нижней формулам соответственно.

Если алгоритм БПФ состоит из двух,  $v$ -го и  $(v-1)$ -го этапов, то начальные условия алгоритма формируются в результате последовательного двукратного разбиения  $N$ -точечной последовательности: сначала  $N$ -точечная последовательность разбивается на две  $N/2$ -точечные, а затем каждая из этих последовательностей вновь разбивается на две  $N/4$ -точечные – четных и нечетных отсчетов па порядку их следования, считая от нуля:

- $N/2$ -точечная последовательность четных отсчетов, полученная для БПФ, состоящего из одного этапа, разбивается на две  $N/4$ -точечные последовательности четных и нечетных отсчетов:

$$\begin{aligned} & \{s(0), s(4), \dots, s(N-4)\}, \\ & \{s(2), s(6), \dots, s(N-2)\}; \end{aligned}$$

- $N/2$ -точечная последовательность нечетных отсчетов, полученная для БПФ, состоящего из одного этапа, разбивается на две  $N/4$ -точечные последовательности четных и нечетных отсчетов:

$$\begin{aligned} & \{s(1), s(5), \dots, s(N-3)\}, \\ & \{s(3), s(7), \dots, s(N-1)\}. \end{aligned}$$

Таким образом, начальная расстановка отсчетов производится по правилу:

$$\begin{aligned} & \{s(0), s(4), \dots, s(N-4); s(2), s(6), \dots, s(N-2)\}, \\ & \{s(1), s(5), \dots, s(N-3); s(3), s(7), \dots, s(N-1)\}. \end{aligned}$$

На  $(v - 1)$ -м этапе определяются два  $N/2$ -точечных ДПФ, причем каждое из них – как комбинация двух  $N/4$ -точечных ДПФ:

- $N/2$ -точечное ДПФ  $f_0^{v-1}(k)$  – как комбинация  $N/4$ -точечных ДПФ – четного  $f_0^{v-2}(k)$  и нечетного  $f_1^{v-2}(k)$ ;
- $N/2$ -точечное ДПФ  $f_1^{v-1}(k)$  – как комбинация  $N/4$ -точечных ДПФ – четного  $f_2^{v-2}(k)$  и нечетного  $f_3^{v-2}(k)$ .

$N/2$ -точечные ДПФ  $f_0^{v-1}(k)$  и  $f_1^{v-1}(k)$  определяются по формуле (2.6), в которой индекс  $v$  уменьшается на единицу, а размерности ДПФ и поворачивающего множителя поникаются вдвое, следовательно два  $N/2$ -точечных ДПФ  $f_0^{v-1}(k)$  и  $f_1^{v-1}(k)$  определяются:

$$\begin{cases} f_0^{v-1}(k) = f_0^{v-2}(k) + W_{N/2}^k f_1^{v-2}(k), \\ f_0^{v-1}\left(k + \frac{N}{4}\right) = f_0^{v-2}(k) - W_{N/2}^k f_1^{v-2}(k), \\ k = 0, 1, \dots, \frac{N}{4} - 1; \\ \\ f_1^{v-1}(k) = f_2^{v-2}(k) + W_{N/2}^k f_3^{v-2}(k), \\ f_1^{v-1}\left(k + \frac{N}{4}\right) = f_2^{v-2}(k) - W_{N/2}^k f_3^{v-2}(k), \\ k = 0, 1, \dots, \frac{N}{4} - 1. \end{cases}$$

На  $v$ -м этапе  $N$ -точечное ДПФ  $f_0^v(k)$  определяется как комбинация  $N/2$ -точечных ДПФ.

Если алгоритм БПФ состоит из трех,  $v$ -го,  $(v - 1)$ -го и  $(v - 2)$ -го этапов, то аналогично начальные условия формируются а результате трехкратного разбиения  $N$ -точечной последовательности, на  $(v - 1)$ -м этапе каждое из четырех  $N/4$ -точечных ДПФ определяется по формуле (2.6), в которой индекс  $v$  уменьшается на два, а размерности ДПФ и поворачивающего множителя поникаются в 4 раза на  $(v - 1)$ -м и  $v$ -м этапах два  $N/2$ -точечных и  $N$ -точечное ДПФ определяются согласно двухэтапному алгоритму. Подобный подход можно применить и для  $v$ -этапного алгоритма БПФ с прореживанием по времени.

Начальные условия формируются в результате  $v$ -кратного разбиения  $N$ -точечной последовательности, а сформированная последовательность называется прореженной по времени.

Процедура многократного прореживания приводит к тому, что исходные данные располагаются не в естественном, а в бит-реверсивном порядке.

Например, для  $N = 8$  естественный и бит-реверсивный порядок расположения отчетов запишется следующим образом:

Естественный порядок	Бит-реверсивный порядок
000(0)	000(0)
001(1)	100(4)
010(2)	010(2)
011(3)	110(6)
100(4)	001(1)
101(5)	101(5)
110(6)	011(3)
111(7)	111(7)

Общая формула расчета ДПФ на произвольной  $i$ -м этапе, полученная на основе (2.6), имеет вид:

$$\begin{cases} f_m^i(k) = f_{2m}^{i-1}(k) + W_L^k f_{2m+1}^{i-1}(k); \\ f_m^i\left(k + \frac{L}{2}\right) = f_{2m}^{i-1}(k) - W_L^k f_{2m+1}^{i-1}(k); \\ i = 1, 2, \dots, v; \\ m = 0, 1, \dots, M - 1; \\ k = 0, 1, \dots, \frac{L}{2} - 1; \end{cases} \quad (2.7)$$

где  $i$  – номер этапа;

$m$  – номер ДПФ;

$k$  – номер отсчета ДПФ;

$M$  – количество  $L$ -точечных ДПФ:

$$M = \frac{N}{2^i} = \frac{2^v}{2^i} = 2^{v-i}; \quad (2.8)$$

$L$  – размерность ДПФ:

$$L = 2^i; \quad (2.9)$$

$f_m^i(k)$  –  $L$ -точечное ДПФ;

$f_{2m}^{i-1}(k)$ ,  $f_{2m+1}^{i-1}(k)$  – четное и нечетное  $L/2$ -точечное ДПФ соответственно.

Согласно (2.7),  $L$ -точечное ДПФ определяется параллельно: первая половина отсчетов  $L$ -точечного ДПФ – по верхней формуле; вторая половина отсчетов  $L$ -точечного ДПФ – по нижней формуле.

Для *первого* этапа алгоритма ( $i = 1$ ) по формуле (2.9) размерность  $L$  ДПФ равна:

$$L = 2^1 = 2,$$

а по формуле (2.8) – количество 2-точечных ДПФ:

$$M = 2^{v-i} = 2^{v-1} = \frac{N}{2}.$$

Общая формула для ДПФ:

$$\begin{cases} f_m^1(0) = f_{2m}^0(0) + W_2^0 f_{2m+1}^0(0); \\ f_m^1(1) = f_{2m}^0(0) - W_2^0 f_{2m+1}^0(0); \end{cases} \quad (2.10)$$

где поворачивающий множитель  $W_2^0 = 1$  сохранен для единства с общей формулой (2.7). В правой части (2.10) согласно принятым обозначениям  $f_{2m}^0(0)$  и  $f_{2m+1}^0(0)$  – отсчеты ДПФ нулевого этапа ( $i = 0$ ), однако такого этапа нет, поэтому для вычисления 2-точечных ДПФ по общей формуле (2.7) перед выполнением первого этапа задаются начальные условия: каждой паре отсчетов ДПФ четному и нечетному присваивают значения пары отсчетов прореженной последовательности четного и нечетного отсчетов соответственно:

$$\begin{cases} f_{2m}^0(0) = \tilde{s}_{2m}, \\ f_{2m+1}^0(0) = \tilde{s}_{2m+1}, \end{cases} \quad (2.11)$$

где  $\tilde{s}_{2m}$  и  $\tilde{s}_{2m+1}$  – условные обозначения четного и нечетного отсчетов 2-точечной последовательности, полученные в результате  $v$ -кратного разбиения исходной  $N$ -точечной последовательности.

Выражение (2.7) показывает, что алгоритм БПФ с прореживанием по времени представляет собой  $v$ -этапную процедуру типа «цикл в цикле», где [11]:

- внешний цикл организуется по переменной  $i$ :  $i = 1, 2, \dots, v$ ;
- первый внутренний цикл (при фиксированном  $i$ ) – по переменной  $m$ :  $m = 0, 1, \dots, M - 1$ ;
- второй внутренний цикл (при фиксированных  $i$  и  $m$ ) – по переменной  $k$ :  $k = 0, 1, \dots, (L/2 - 1)$ . В цикле вычисляется  $k$ -й отсчет  $L$ -точечного ДПФ по формуле (2.7).

На выходе алгоритма БПФ с прореживанием по времени определяется  $N$ -точечное ДПФ, отсчеты которого следуют в естественном порядке:  $k = 0, 1, \dots, (N - 1)$ .

Основной операцией алгоритма БПФ, называемой базовой, является одновременное (параллельное) вычисление двух отсчетов ДПФ –  $k$ -го и  $(k + L/2)$ -го по двум верхним формулам (2.7). На рис. 2.7 приведен направленный граф базовой операции, получивший название «бабочка» по ассоциации с изображением графа. Кружок обозначает арифметическую операцию сложения/вычитания, верхний выход соответствует сумме, а нижний – разности; стрелка обозначает операцию умножения на поворачивающий множитель, стоящий над ней.

Алгоритм БПФ обычно представляют в виде направленного графа – совокупности «бабочек»:

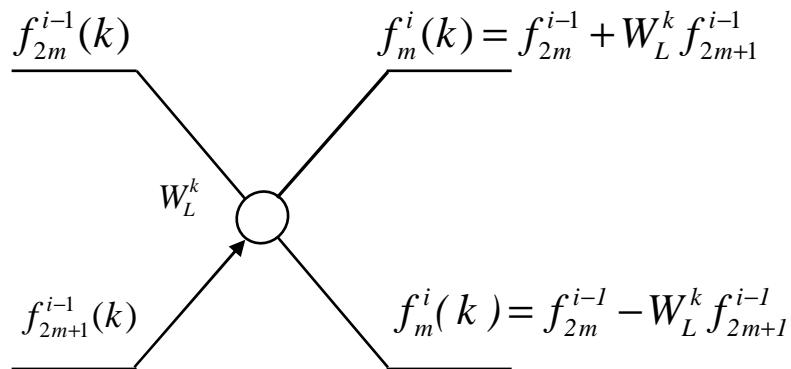


Рис. 2.7. Граф базовой операции алгоритма БПФ  
с прореживанием по времени

Количество «бабочек» т. е. базовых операций алгоритма, на любом  $i$ -м этапе одинаково: согласно (2.7), на  $i$ -м этапе при фиксированном значении  $m$  и  $k = 0, 1, \dots, L/2$  требуется  $L/2$  «бабочек», следовательно, при  $m = 0, 1, \dots, (M - 1)$  необходимо  $ML/2$  «бабочек». Количество «бабочек» на любом  $i$ -м этапе определяется как:

$$M \frac{L}{2} = 2^{v-i} \frac{2^i}{2} = \frac{N}{2}.$$

Алгоритм БПФ с прореживанием по времени традиционно рассматривается на примере вычисления 8-точечного ДПФ для  $N = 2^v = 2^3 = 8$ ,  $v = 3$ .

Алгоритм БПФ представляет собой трехэтапную процедуру определения ДПФ по общей формуле (2.7) при  $i = 1, 2, 3$ . Начальные условия

алгоритма формируются в результате трехкратного разбиения исходной 8-точечной последовательности на четные и нечетные отсчеты и представляются в двоично-инверсном порядке:

$$\{s(0), s(4), s(2); s(6), s(1), s(5), s(3), s(7)\}.$$

На нулевом этапе каждой паре отсчетов ДПФ – четному  $f_{2m}^0(0)$  и нечетному  $f_{2m+1}^0(0)$  при  $m = 0, 1, 2, 3$  присваиваются значения четного и нечетного отсчетов прореженной 8-точечной последовательности.

На первом этапе, ( $i = 1$ ), определяются четыре 2-точечных ДПФ при  $m = 0, 1, 2, 3$  с учетом начальных условий:

$$\begin{aligned} \begin{cases} f_0^1(0) = s(0) + W_2^0 s(4); \\ f_0^1(1) = s(0) - W_2^0 s(4); \end{cases} & \quad \begin{cases} f_1^1(0) = s(2) + W_2^0 s(6); \\ f_1^1(1) = s(2) - W_2^0 s(6); \end{cases} \\ \begin{cases} f_2^1(0) = s(1) + W_2^0 s(5); \\ f_2^1(1) = s(1) - W_2^0 s(5); \end{cases} & \quad \begin{cases} f_3^1(0) = s(3) + W_2^0 s(7); \\ f_3^1(1) = s(3) - W_2^0 s(7). \end{cases} \end{aligned}$$

Поворачивающий множитель в данном случае  $W_2^0 = 1$ .

На втором этапе ( $i = 2$ ) определяются два 4-точечных ДПФ при  $m = 0, 1$ :

$$\begin{aligned} \begin{cases} f_0^2(k) = f_0^1(k) + W_4^k f_1^1(k); \\ f_0^2(k+2) = f_0^1(k) - W_4^k f_1^1(k); \\ k = 0, 1; \end{cases} & \quad \begin{cases} f_1^2(k) = f_2^1(k) + W_4^k f_3^1(k); \\ f_1^2(k+2) = f_2^1(k) - W_4^k f_3^1(k); \\ k = 0, 1. \end{cases} \end{aligned}$$

На третьем этапе ( $i = 3$ ) определяется 8-точечное ДПФ при  $m = 0$ :

$$\begin{cases} f_0^3(k) = f_0^2(k) + W_8^k f_1^2(k); \\ f_0^3(k+4) = f_0^2(k) - W_8^k f_1^2(k); \\ k = 0, 1, 2, 3. \end{cases}$$

Полученные отсчеты восьмиточечного ДПФ следуют в естественном порядке.

Граф восьмиточечного БПФ с прореживанием по времени показан на рис. 2.8.

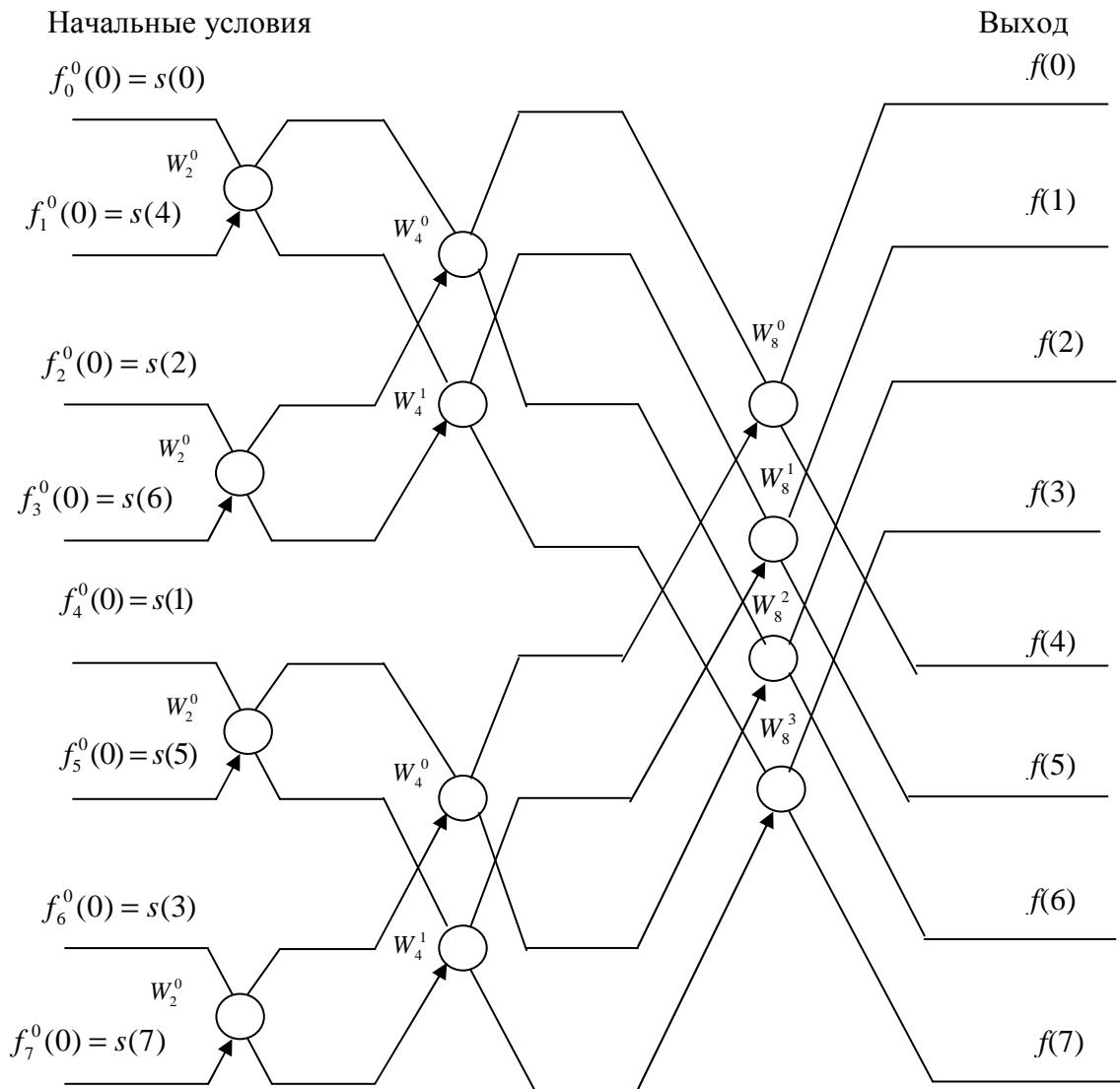


Рис. 2.8. Граф алгоритма БПФ с прореживанием по времени для восьмиточечного ДПФ

### 2.3.2. Алгоритм БПФ с прореживанием по частоте

Основная идея алгоритма БПФ с прореживанием по частоте заключается в поэтапном вычислении  $N$ -точечного ДПФ па  $v$  этапах, на каждом из которых ДПФ определяется через ДПФ вдвое большей размерности [11].

Алгоритм БПФ с прореживанием по частоте в общем случае требует выполнения следующих итераций:

- задание начальных условий – естественный порядок следования отсчетов входных отсчетов;

- на первом этапе определяются  $N/2$ -точечные ДПФ  $N/2$ -точечных последовательностей (двух половин исходной последовательности);
  - на втором этапе определяются  $N/4$ -точечные ДПФ как комбинация  $N/2$ -точечных ДПФ;
- .....
- на  $i$ -ом этапе определяются  $2^{i-1}$ -точечные ДНФ как комбинация  $2^i$ -точечных ДПФ;
- .....
- на  $v$ -ом (последнем) этапе определяются 2-точечные ДПФ как комбинация 4-точечных ДПФ. Выходные отсчеты ДПФ следуют в бит-реверсивном порядке двоичных номеров.

Анализ алгоритмов БПФ с прореживанием по времени и частоте показывает, что общая формула расчета БПФ с прореживанием по частоте на произвольном  $i$ -м этапе может быть получена путем замены входа и выхода, а также обращения стрелки направленного графа – «бабочки» алгоритма БПФ с прореживанием по времени (рис. 2.9).

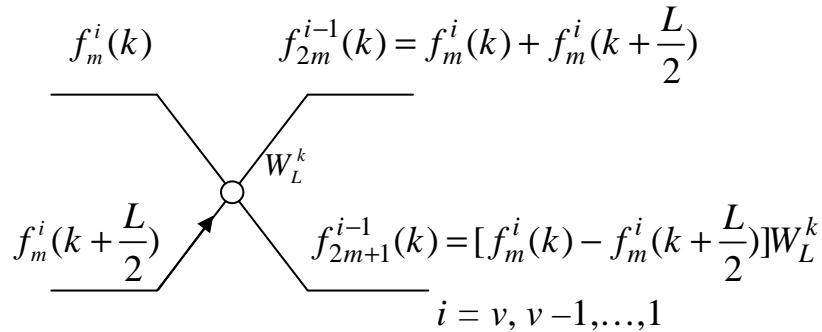


Рис. 2.9. Граф базовой операции алгоритма БПФ с прореживанием по частоте

Общая формула расчета ДПФ для произвольного  $i$ -го этапа [11]:

$$\begin{cases} f_{2m}^{i-1}(k) = f_m^i(k) + f_m^i(k + \frac{L}{2}); \\ f_{2m+1}^{i-1}(k) = [f_m^i(k) - f_m^i(k + \frac{L}{2})]W_L^k; \\ i = v, v-1, \dots, 1; \\ m = 0, 1, \dots, M-1; \\ k = 0, 1, \dots, \frac{L}{2}-1; \end{cases} \quad (2.12)$$

где  $i$  – номер этапа ( $i = v, v - 1, \dots, 1$ );  
 $m$  – номер ДПФ;  
 $k$  – номер отсчета ДПФ;  
 $M$  – количество ДПФ, определяемое по формуле  $M = 2^{v-i}$ ;  
 $L$  – размерное ДПФ, определяемое по формуле  $L = 2^i$ ;  
 $f_m^i(k)$  –  $L$ -точечное ДПФ;  
 $f_m^i(k), f_{2m+1}^{i-1}(k)$  – четное и нечетное  $L/2$ -точечное ДПФ соответственно.

Согласно (2.12),  $L$ -точечное ДПФ определяется параллельно:

- отсчеты *четного*  $L/2$ -точечного ДПФ – по *верхней* формуле;
- отсчеты *нечетного*  $L/2$ -точечного ДПФ – по *нижней* формуле.

Анализ выражения (2.12) описывает алгоритм БПФ с прореживанием по частоте –  $v$ -этапную процедуру типа «цикл в цикле», где:

- внешний цикл организуется по переменной  $i$ :  $i = v \dots 1$ ;
- первый внутренний цикл (при фиксированном  $i$ ) – по переменной  $m$ :  $m = 0, 1, 2, \dots, M - 1$ ;
- второй внутренний цикл (при фиксированных  $i$  и  $m$ ) – по переменной  $k$ :  $k = 0, 1, \dots, (L/2 - 1)$ . В цикле вычисляется  $k$ -й отсчет  $L/2$ -точечного ДПФ по формуле (2.12).

На выходе алгоритма (в результате вычислений при  $i = 1$  и  $L = 2^1 = 2$ ) согласно (2.12) получаем  $N$ -точечное ДПФ, отсчеты которого следуют в бит-реверсивном порядке двоичных номеров.

На практике алгоритм БПФ с прореживанием по частоте применяют реже, чем с прореживанием по времени, т. к. последний обеспечивает естественный порядок следования отсчетов ДПФ на выходе.

Как и в алгоритме с прореживанием по времени, в данном алгоритме вычислительные затраты составляют  $M \log_2 N$  операций сложения и  $0,5M \log_2 N$  операций умножения, а вычисления можно выполнять с замещением. Величина «бабочки» по мере продвижения к концу вычислений уменьшается.

Если  $N = 8$ , алгоритм БПФ представляет собой трехэтапную процедуру вычисления ДПФ по общей формуле (2.12) при  $i = 3, 2, 1$ .

Начальные условия алгоритма задаются как:

$$\begin{cases} f_0^3(k) = s(k); \\ k = 0, 1, \dots, 7. \end{cases}$$

*Первый этап:*  $i = v = 3$ .

Определяются два 4-точечных ДПФ (по формуле 2.12) при  $m = 0$  с учетом начальных условий:

$$\begin{cases} f_0^2(k) = s(k) + s(k+4); \\ f_1^2(k) = [s(k) - s(k+4)]W_8^k; \\ k = 0, 1, \dots, 3. \end{cases}$$

*Второй этап:*  $i = v - 1 = 2$ .

Определяются четыре 2-точечных ДПФ при  $m = 0, 1$ :

$$\begin{cases} f_{2m}^1(k) = f_m^2(k) + f_m^2(k+2); \\ f_{2m+1}^1(k) = [f_m^2(k) - f_m^2(k+2)]W_4^k; \\ k = 0, 1. \end{cases}$$

*Третий этап:*  $i = v - 2 = 1$ .

Определяются 8-точечное ДПФ при  $m = 0, 1, 2, 3$ :

$$\begin{cases} f_{2m}^0(0) = f_m^1(0) + f_m^1(1); \\ f_{2m+1}^0(0) = [f_m^1(0) - f_m^1(1)]W_2^0. \end{cases}$$

Значения  $f_{2m}^0(0)$  и  $f_{2m+1}^0(0)$  при  $m = 0, 1, 2, 3$  есть искомое восьмиточечное ДПФ, отсчеты которого следуют в бит-реверсивном порядке двоичных номеров:

$$\begin{cases} f(0) = f_0^0(0); \\ f(4) = f_1^0(0); \end{cases}$$

$$\begin{cases} f(2) = f_2^0(0); \\ f(6) = f_3^0(0); \end{cases}$$

$$\begin{cases} f(1) = f_4^0(0); \\ f(5) = f_5^0(0); \end{cases}$$

$$\begin{cases} f(3) = f_6^0(0); \\ f(7) = f_7^0(0). \end{cases}$$

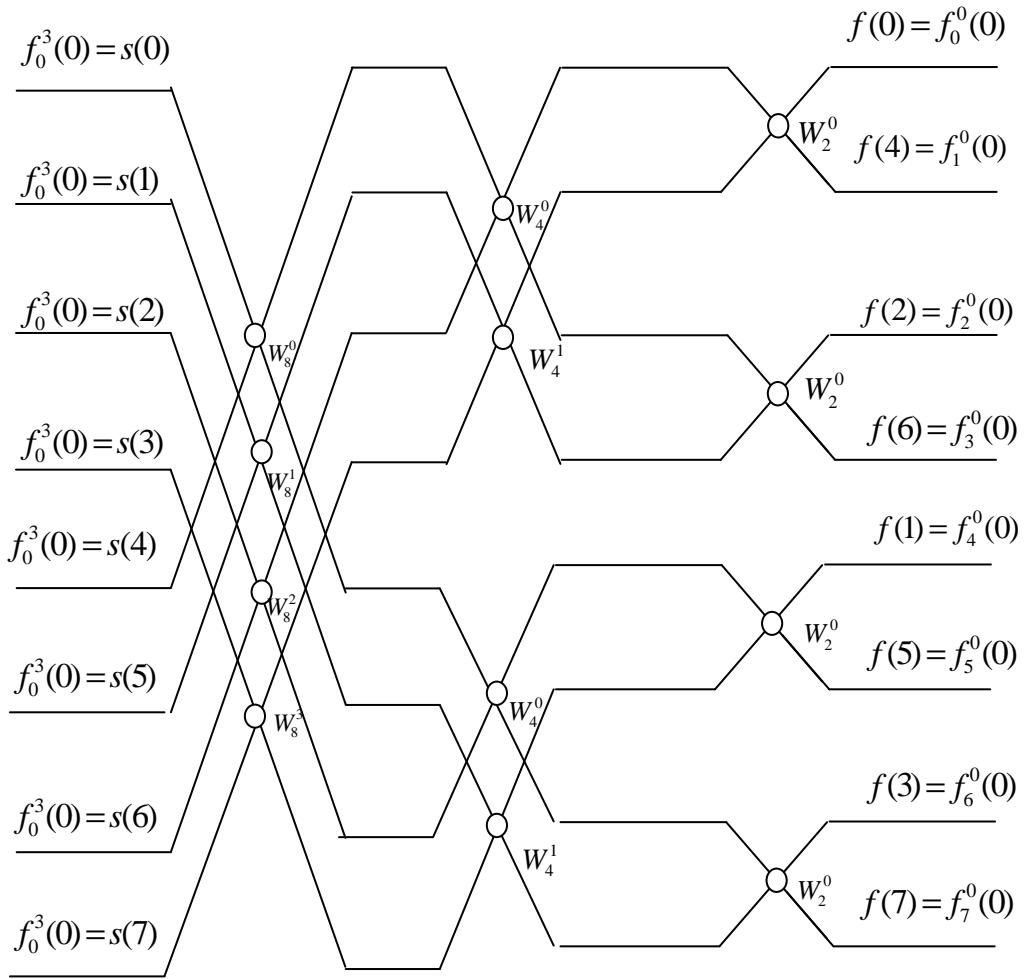


Рис. 2.10. Граф алгоритма БПФ с прореживанием по частоте для восьмиточечного ДПФ

### 2.3.3. Алгоритм действительного БПФ (АДБПФ)

Быстрый алгоритм для расчета ДПФ действительной  $N$ -точечной последовательности  $\{s(n)\}$  представлен в [7]. С целью снижения вычислительной сложности алгоритма БПФ с прореживанием по времени  $N$ -точечной действительной последовательности используют свойство эрмитовой симметрии спектральных коэффициентов ДПФ и весовых множителей:

$$\begin{aligned}
 f_k &= f_{N-k}^* \\
 f_k &= f_{N/2-k}^* \\
 w_N^k &= -\left(w_N^{N/2-k}\right)^*, \\
 k &= 1, \dots, N/4-1.
 \end{aligned} \tag{2.13}$$

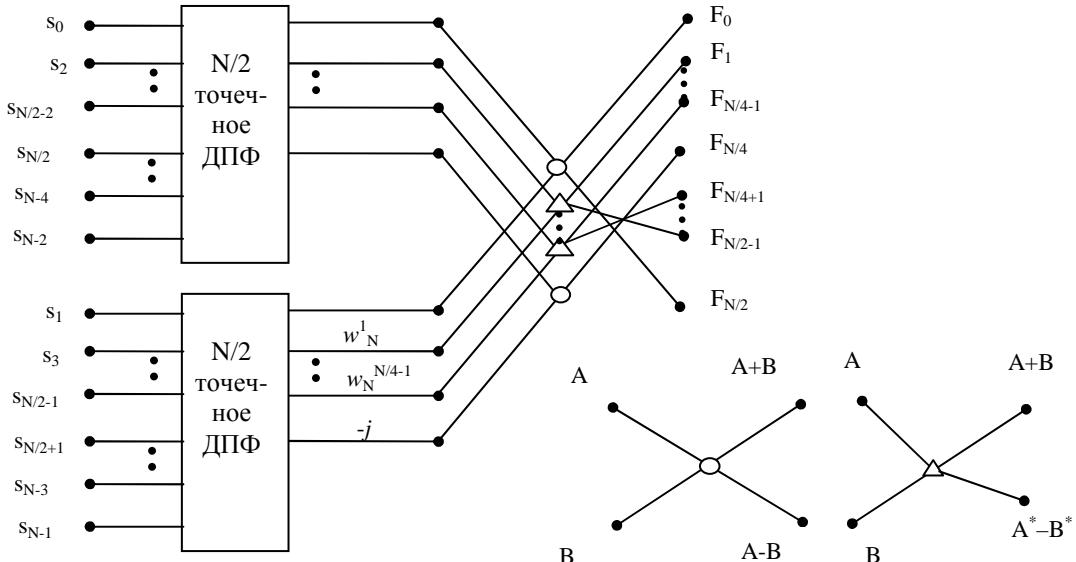


Рис. 2.11. Первый этап разложения в АДБПФ

Из соотношений (2.13) следует, что на этапе взвешивания достаточно выполнить только  $N/4 - 1$  комплексных умножений  $f_k = s_k W_N^k$  ( $k=1,..,N/4-1$ ), поскольку  $f_{N/2-k} = -f_k^*$ . Учитывая свойство (2.13), традиционная операция «бабочка» преобразуется в «модифицированную бабочку», которая помимо суммирования содержит операцию разности комплексно-сопряженных чисел. На рис. 2.11 представлен график, иллюстрирующий разложение  $N$ -точечного ДПФ действительной последовательности на два  $N/2$ -точечных ДПФ с помощью алгоритма действительного быстрого преобразования Фурье (АДБПФ). Исходная последовательность  $\{s(n)\}$  ( $n=0,1,...,N-1$ ) разделена по четным и нечетным отсчетам, а последовательность коэффициентов ДПФ  $f_k$  ( $k=0,1,...,N/2$ ) расположена в прямом порядке. Каждый этап АДБПФ состоит из следующих операций (рис. 2.11):

1. Выполнение двух  $N/2$ -точечных ДПФ действительных последовательностей.
2. Выполнение  $N/4 - 1$  комплексных взвешиваний.
3. Вычисление  $N/2+1$  коэффициентов  $N$ -точечного ДПФ на основе  $N/4 - 1$  попарных сложений и вычитаний комплексных чисел и одного сложения и вычитания пары действительных чисел при выполнении двух базовых операций.

Для АДБПФ характерно:

1. Наличие обратной двоичной перестановки входной последовательности  $\{s(n)\}$  ( $n=0,1,...,N-1$ ;  $N=2^\nu$ ).
2. Прямой порядок коэффициентов Фурье  $f_k$  ( $k=0,1,...,N/2$ ) на выходе.
3. Рекурсивная структура, содержащая  $\nu$  этапов.

4. Два вида базовых операций: «бабочка» и «модифицированная бабочка», состоящая из операции суммирования двух комплексных чисел и разности двух комплексно-сопряженных чисел.

5. Выполнение ( $N/4 - 1$ ) комплексных взвешиваний на  $W_N^k$  ( $k=1,..,N/4-1$ ) для каждого этапа АДБПФ.

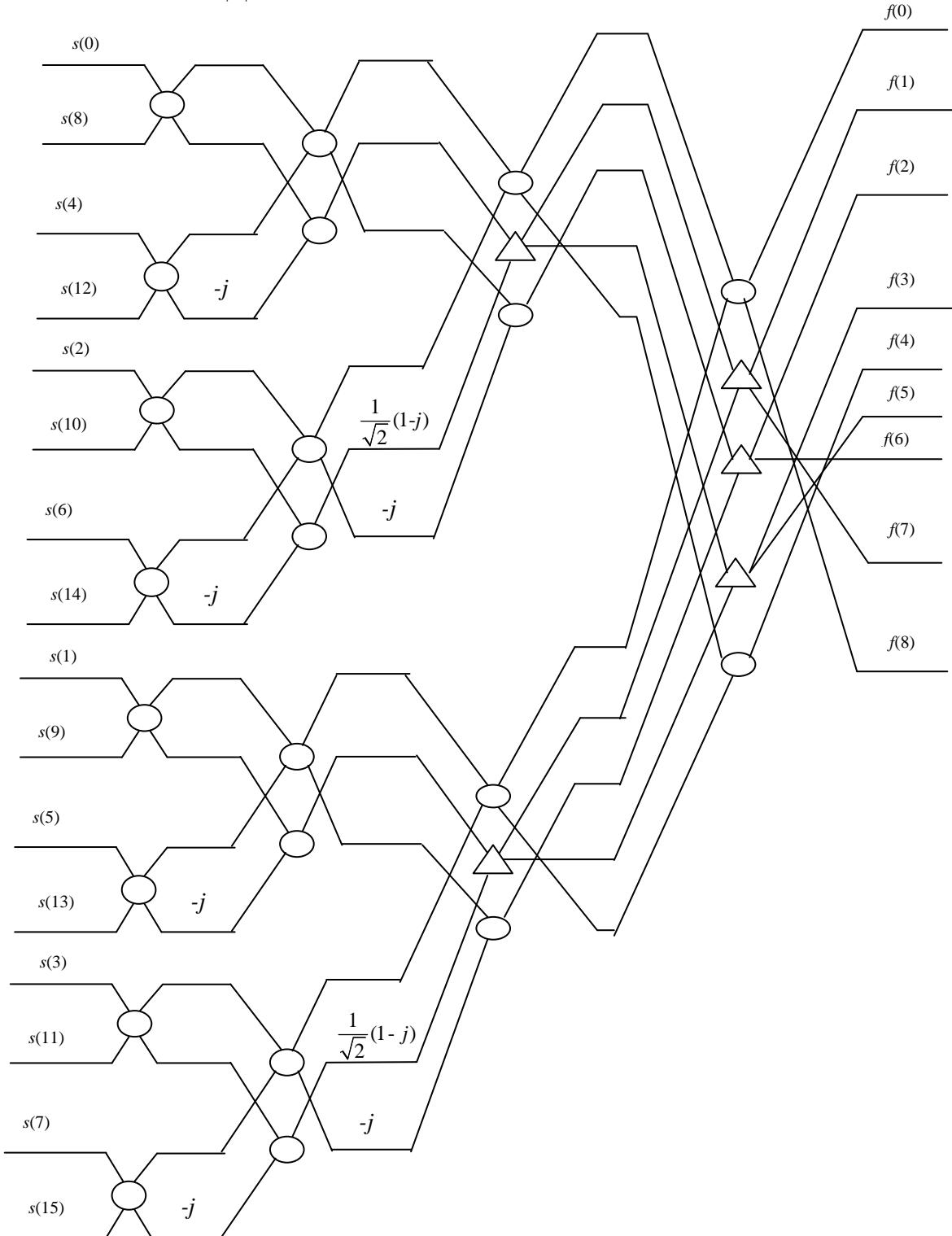


Рис. 2.12. Граф АДБПФ для  $N = 16$

## 2.4. Преобразование Уолша-Адамара

### 2.4.1. Функции Уолша

Функции Уолша образуют полную ортонормированную систему с количеством функций  $2^l$ ,  $l = 1, 2, 3 \dots$ , и значениями  $\pm 1$  [8]. Число функций, включаемых в систему, обычно равно числу отсчетов каждой функции, так как при дискретном спектральном анализе сигналов с  $N$  отсчетами число спектральных составляющих также должно быть равно  $N$ . Поскольку нумерация (упорядочение) функций Уолша может быть произведена разными способами, то возможны различные системы функций Уолша. Наиболее широко используется представление таких систем в виде квадратных матриц, в которых каждая строка-это функция Уолша.

Адамар показал, что эти матрицы принадлежат классу матриц с максимально возможным определителем. Поэтому они получили название матриц Адамара [8].

Матрицей Адамара порядка  $N$  называется ортогональная матрица размером  $N \times N$ , элементами которой являются действительные числа « $\pm 1$ » [8]. Матрица Адамара порядка  $N$  обозначается  $H_N$ .

Простейшей матрицей Адамара является матрица второго порядка вида:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Если  $H_N$  – матрица Адамара порядка  $N$ , то матрица

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

является матрицей Адамара порядка  $2N$ .

Например, матрицы четвертого и восьмого порядка имеют соответственно следующий вид:

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix};$$

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \left[ \begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array} \right].$$

В более компактном виде матрицу Адамара можно представить:

$$H_8 = H_2 \otimes H_4,$$

где  $\otimes$  – знак прямого или кронекерова умножения матриц, которое вычисляется путем умножения каждого элемента матрицы  $H_4$  на матрицу  $H_2$ .

Для функций, принадлежащих системе Уолша-Адамара, используется обозначение  $had(k,n)$ , где  $k$ -номер строки, а  $n$  – номер столбца матрицы Адамара. Пронумеруем строки и столбцы матрицы Адамара двоичными числами от 0 до  $N - 1$ . Тогда элемент матрицы, стоящий на пересечении строки с номером  $k$  и столбца с номером  $n$ , вычисляется по формуле:

$$had(k,n) = (-1)^{(k \cdot n)},$$

где  $(k \cdot n) = \sum_{i=0}^{l-1} k_i n_i$  – скалярное произведение векторов  $k$  и  $n$ ;

$k$  – двоичный номер строки;

$n$  – двоичный номер столбца.

Пусть  $k$  – двоичный номер строки, а  $n$  – двоичный номер столбца ( $k, n \in 0, \dots, N - 1$ ). Тогда элемент матрицы, стоящий на пересечении строки с номером  $k$  и столбца с номером  $n$ , вычисляется по формуле

$$had(k,n) = (-1)^{(kn)}, \quad (2.14)$$

где  $(kn) = \sum_{i=0}^{l-1} k_i n_i$  – скалярное произведение векторов  $k$  и  $n$ .

Например, для матрицы  $H_8$  с учетом (2.14):

$$\begin{array}{ccccccccc} 0 & & 0 & & & & 1 & & \\ 0 & .. & .. & 1 & .. & .. & .. & .. & .....1 \\ 0 & & & 1 & & & & & 1 \end{array} \begin{pmatrix} n_2 \\ n_1 \\ n_0 \end{pmatrix} = n$$

$$k = (k_2 \ k_1 \ k_0)$$

$$\begin{array}{ccc} 0 & 0 & 0 \\ \vdots & & \\ 0 & 1 & 0 \\ \vdots & & \\ \vdots & & \\ 1 & 1 & 1 \end{array} \left| \begin{array}{c} | \\ | \\ - - - (-1)^{kn} - - - - - - - - \\ | \\ | \\ | \end{array} \right| = \begin{bmatrix} had(000,n) \\ had(001,n) \\ had(010,n) \\ had(011,n) \\ had(100,n) \\ had(101,n) \\ had(110,n) \\ had(111,n) \end{bmatrix}.$$

Строки матриц Адамара  $had(000,n), had(001,n), \dots, had(k,n)$  можно рассматривать как функции дискретных переменных  $n$  и  $k$ , определенные в целочисленных точках  $0, 1, 2, \dots, N - 1$  интервала  $[0, N)$ . Эти функции называются дискретными функциями Уолша [8]. Первая переменная  $k$  обозначает номер функции, вторая переменная  $n$  – дискретное время. В качестве номера функции берется номер соответствующей строки матрицы Адамара. При этом говорят, что функции упорядочены по Адамару.

Важнейшими свойствами функций Уолша являются [8]:

1) ортогональность:

$$\sum_{n=0}^{N-1} had(k_1, n) had(k_2, n) = \begin{cases} N, & k_1 = k_2; \\ 0, & k_1 \neq k_2; \end{cases}$$

или в матричной записи:

$$H_N H_N^T = NI,$$

где  $I$  – единичная матрица;

2) симметричность:

$$H_N = H_N^T;$$

используя свойства ортогональности и симметричности, легко получить  $H^{-1} = N^{-1}H$ ;

3) мультипликативность по номеру функции:

$$had(k_1, n) had(k_2, n) = had(k_1 \oplus k_2, n)$$

и номеру отсчета:

$$had(k, n_1) had(k, n_2) = had(k, n_1 \oplus n_2).$$

Постоянная функция и функции меандрового типа с одной ненулевой позицией в двоичной записи номера называются функциями Радемахера. Для  $N = 8$  они равны:

$$\begin{aligned} R_0 &= had(000, n); & R_2 &= had(010, n); \\ R_1 &= had(100, n); & R_3 &= had(001, n). \end{aligned}$$

Из свойства мультипликативности следует, что любая функция Уолша может быть получена перемножением функций Радемахера. Например:

$$had(011, n) = R_2 R_3;$$

$$had(110, n) = R_1 R_2.$$

В практических приложениях используется и другая нумерация функций, что соответствует другому способу ранжирования обобщенных гармоник. Наиболее употребительными из них являются упорядочения по Пэли и по Уолшу [8].

В некоторых практических приложениях, например в аналого-цифровых преобразованиях, желательно использовать коды, у которых все следующие друг за другом кодовые слова различаются только одной цифрой в некотором разряде. Коды, обладающие таким свойством, называются циклическими.

Очень важным циклическим кодом является код Грея. Двоичное представление числа может быть легко преобразовано в код Грея с помощью полусумматоров.

Пусть последовательность  $g_{n-1}, g_{n-2}, \dots, g_0$  является кодовым словом в  $n$  разрядном двоичном коде Грея, соответствующее двоичному числу  $b_{n-1}, b_{n-2}, \dots, b_0$ .

Тогда  $g_i$  может быть определена согласно:

$$g_i = b_i \oplus b_{i+1}, \quad g_{n-1} = b_{n-1}, \quad 0 \leq i \leq n-2.$$

Таблица 2.3

**Трехразрядный код Грея**

Десятичное число	Двоичный код			Код Грея		
	$b_2$	$b_1$	$b_0$	$g_2$	$g_1$	$g_0$
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Система Пэли образуется из системы Адамара двоичной инверсией номеров функций, т. е. путем записи разрядов двоичного представления номера  $k$  в обратном порядке. Система Уолша располагает функции в порядке возрастания числа смен знака на интервале (по «частотам» следования) (рис. 2.13), что соответствует расположению их номеров в двоично-инверсном коде Грея.

Для  $N = 4$  в рассмотренных способах упорядочения строки расположены следующим образом:

по Адамару	по Пэли	по Уолшу
00	00	00
01	10	10
10	01	11
11	11	01

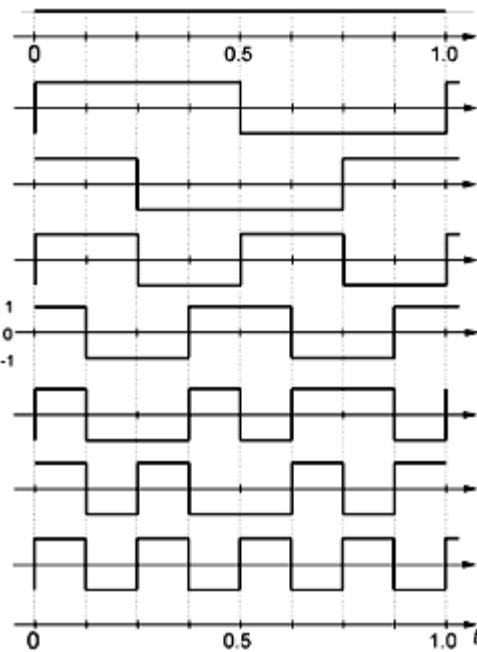


Рис. 2.13. Функции Уолша

Соответствующие матрицы при упорядочении по Пэли и по Уолшу равны:

$$P = \{pal(k, n)\} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \quad W = \{wal(k, n)\} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

#### 2.4.2. Преобразование Уолша-Адамара и его свойства

Пусть  $\{s(n)\} = \{s(0), s(1), \dots, s(N-1)\}$  – совокупность равноотстоящих отсчетов сигнала. Пара дискретного преобразования Уолша-Адамара в показательной форме представляется в виде [8]:

$$b(k) = \sum_{n=0}^{N-1} s(n)(-1)^{(kn)}, \quad k = 0, 1, \dots, N-1; \quad (2.15)$$

$$s(n) = N^{-1} \sum_{k=0}^{N-1} b(k)(-1)^{(kn)}, \quad n = 0, 1, \dots, N-1. \quad (2.16)$$

Равенство (2.15) называется *прямым преобразованием* и дает спектр сигнала в базисе Уолша. Равенство (2.16) называют *обратным преобразованием*.

Используя матрицу Адамара порядка  $N$ , можно записать преобразование в матричной форме:

$$\begin{aligned} B &= HS; \\ s &= N^{-1}HB, \end{aligned}$$

где  $s = [s(0), s(1), \dots, s(N-1)]^T$ ,  $B = [b(0), b(1), \dots, b(N-1)]^T$  – векторы-столбцы отсчетов сигнала и спектральных коэффициентов соответственно.

Основными свойствами преобразования являются [8]:

1. *Линейность.* Если  $\{x(n)\}$  и  $\{y(n)\}$  – две последовательности со спектрами  $\{b_x(k)\}$  и  $\{b_y(k)\}$  соответственно, то спектр их взвешенной суммы  $\{z(n)\} = \{A_x(n)\} + \{B_y(n)\}$  равен:

$$\{b_z(k)\} = \{Ab_x(k)\} + \{Bb_y(k)\}.$$

2. *Инвариантность к диадному сдвигу.* Рассмотрим функцию  $s(n)$  дискретной переменной  $n$ . Функция  $s(n \oplus \tau)$  называется диадным сдвигом функции  $s(n)$ . Сущность диадного сдвига заключается в перестановке отсчетов исходной функции. В частности, на место отсчета с номером  $n$  ставится отсчет с номером  $(n \oplus \tau)$ . Например, пусть  $\{s(n)\} = \{0011\}$  и  $\tau = 2$ .

Значения  $(n \oplus \tau)$  для  $n = 0, 1, 2, 3$  равны:

$n$	$n \oplus \tau$
00	$\oplus$ 10 = 10
01	$\oplus$ 10 = 11
10	$\oplus$ 10 = 00
11	$\oplus$ 10 = 01

Из этой записи следует, что отсчет с номером ноль ставится на второй отсчет, а на место отсчета с номером один – третий и т. д. В результате получим:

$$\begin{aligned} \{s(n)\} &= \{0011\}; \\ \{s(n \oplus \tau)\} &= \{1100\}. \end{aligned}$$

Из свойств мультипликативности следует, что при фиксированном  $\tau$

$$had(k, n \oplus \tau) = had(k, n) had(k, \tau) = \pm had(k, n),$$

т. е. спектральные составляющие исходного и диадно-сдвинутого сигналов могут отличаться только знаком. Амплитудный же спектр при диадном сдвиге не меняется.

*3. Теорема о свертке и корреляции.* Понятие диадного сдвига позволяет обобщить понятия свертки и корреляционной функции. Так как суммирование и вычитание по модулю два совпадают, то диадная свертка совпадает с диадной корреляцией и определяется выражением

$$y(n) = r(n) = \sum_{l=0}^{N-1} h(l)s(n \oplus l),$$

где  $n = 0, 1, \dots, (N - 1)$ .

Теорема о свертке утверждает, что спектр свертки равен произведению спектров сворачиваемых последовательностей:

$$b_y(k) = b_h(k)b_s(k).$$

Это позволяет для вычисления диадной свертки и корреляционной функции использовать преобразование Адамара:

$$R = H^{-1}(B_h B_s) = N^{-1}H(B_h B_s).$$

### 2.4.3. Быстрое преобразование Уолша-Адамара

Вычисление преобразования прямым способом требует выполнения  $N(N - 1)$  операций сложения. Существуют быстрые алгоритмы, которые требуют только  $M \log_2 N$  операций.

Для построения быстрого алгоритма рассмотрим матричное равенство  $B = HS$ . Запишем в виде:

$$\begin{bmatrix} b(0) \\ b(1) \\ \vdots \\ b\left(\frac{N}{2} - 1\right) \\ b(N/2) \\ \vdots \\ b(N - 1) \end{bmatrix} = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ \vdots \\ s\left(\frac{N}{2} - 1\right) \\ s(N/2) \\ \vdots \\ s(N - 1) \end{bmatrix},$$

что позволит отдельно вычислить первую и вторую половины спектра.

В результате:

$$\begin{bmatrix} b(0) \\ b(1) \\ \vdots \\ \vdots \\ b\left(\frac{N}{2}-1\right) \end{bmatrix} = H_{N/2} \begin{bmatrix} s(0) + s(N/2) \\ s(1) + s(N/2+1) \\ \vdots \\ \vdots \\ s\left(\frac{N}{2}-1\right) + s(N-1) \end{bmatrix} \quad \begin{bmatrix} b(N/2) \\ b\left(\frac{N}{2}-1\right) \\ \vdots \\ \vdots \\ b(N-1) \end{bmatrix} = H_{N/2} \begin{bmatrix} s(0) - s(N/2) \\ s(1) - s(N/2) \\ \vdots \\ \vdots \\ s\left(\frac{N}{2}-1\right) - s(N-1) \end{bmatrix}$$

Из этих выражений следует, что вычисление  $N$ -точечного преобразования сводится к предварительному суммированию (вычитанию) входных данных и последующему вычислению двух  $N/2$ -точечных преобразований (рис. 2.14).

Процедуру снижения размерности преобразований можно продолжить до получения двухточечного преобразования. Для этого потребуется  $\log_2 N$  шагов. На каждом шаге производится  $N$  сложений, поэтому общее количество операций сложения равно  $M \log_2 N$ . Рассмотренный алгоритм называется быстрым преобразованием Адамара (БПА). На рис. 2.15 приведен график вычислительного процесса для  $N = 8$ . БПУА принято также трактовать как разложение (факторизацию) матрицы Адамара в произведение слабозаполненных сомножителей.

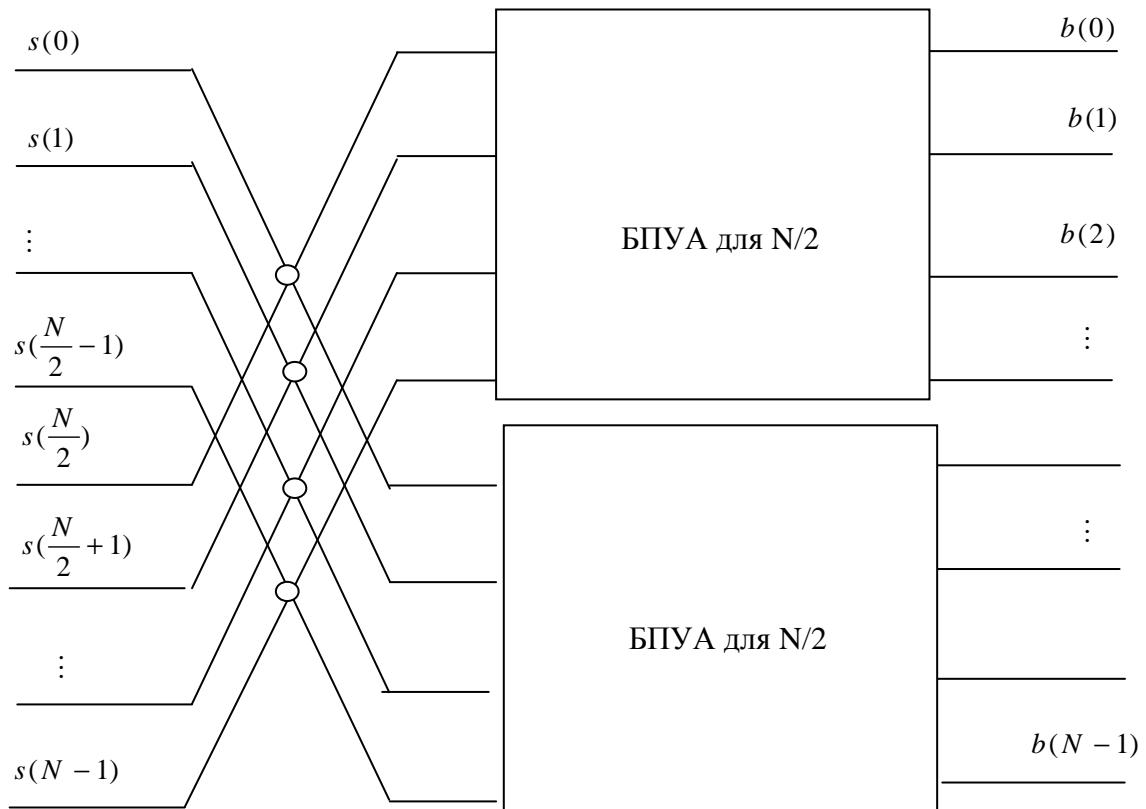


Рис. 2.14. Принцип построения алгоритма БПУА

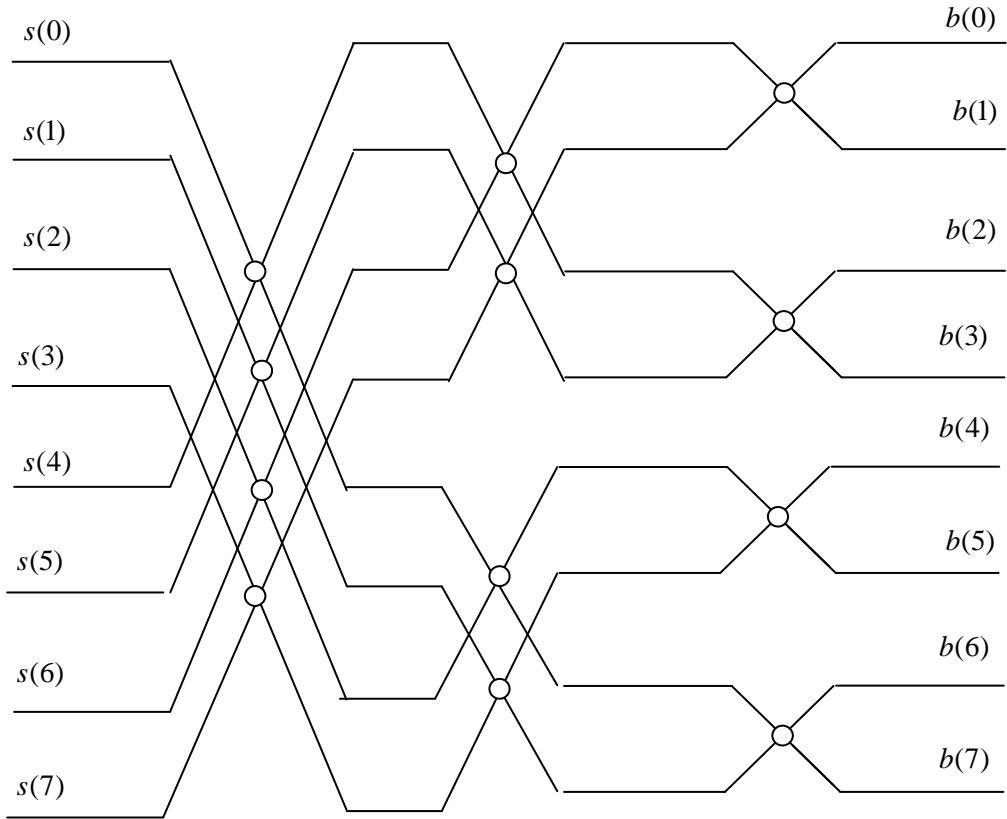


Рис. 2.15. Граф алгоритма БПУА для  $N = 8$

Возможность представления матриц Адамара в виде произведения слабозаполненных матриц показал Гуд. Факторизация методом Гуда для матриц, представляющих собой кронекеровское произведение нескольких одинаковых матриц, основана на следующих теоремах.

1. Если  $E_m^{[n]}$  произвольная матрица размером  $m \times m$ , то

$$E_m^{[n]} = \underbrace{E_m \otimes E_m \otimes \dots \otimes E_m}_{n \text{ раз}} = B_{m^n}^n.$$

Другими словами,  $n$ -я кронекеровская степень матрицы  $E_m$  размером  $m \times m$  равна обычной  $n$ -й степени матрицы  $B_m^n$  размером  $m^n \times m^n$ .

2. Если  $E_m^{[n]}$  произвольная матрица размером  $m \times m$ , то

$$E_m^{[n]} = \underbrace{E_m \otimes E_m \otimes \dots \otimes E_m}_{n \text{ раз}} = C_{m^n}^{(1)} \cdot C_{m^n}^{(2)} \dots C_{m^n}^{(n)},$$

где  $C_{m^n}^{(1)} = E_m \otimes 1_m \otimes \dots \otimes 1_m$ ;  $C_{m^n}^{(2)} = 1_m \otimes E_m \otimes \dots \otimes 1_m$ ;  $C_{m^n}^{(n)} = 1_m \otimes 1_m \otimes \dots \otimes E_m$ .

Здесь  $1_m$ -единичная матрица размером  $m \times m$ , а матрицы  $C_{m^n}^{(1)}, C_{m^n}^{(2)}, \dots, C_{m^n}^{(n)}$  – все размером  $m^n \times m^n$ .

Особенностью матриц  $B_{m^n}$  и матриц  $C_{m^n}^{(1)}, C_{m^n}^{(2)}, \dots, C_{m^n}^{(n)}$  является то, что они имеют не больше  $m^{n+1}$  отличных от нуля элементов, поэтому общее число операций типа сложение-вычитание не превышает величины  $n \times m^n$ .

Для матриц Адамара:

$$H_{2^n} = E_2^{[n]},$$

где  $E_2^{[n]} = E_2^{[1]} \otimes E_2^{[2]} \otimes \dots \otimes E_2^{[i]} \otimes \dots \otimes E_2^{[n]}$ ,  $E_2^{[l]} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ ,  $l = 1, 2, \dots, n$ .

Согласно теореме 1, матрица Адамара размером  $8 \times 8$  раскладывается на произведение трех слабозаполненных сомножителей:

$$H_8 = H_2 \otimes H_2 \otimes H_2 = B^3,$$

$$\text{где } B = \begin{bmatrix} 1 & 1 & & & & & & \\ & 1 & 1 & & & & & \\ & & 1 & 1 & & & & \\ & & & 1 & 1 & & & \\ 1 & -1 & & & & 1 & 1 & \\ & 1 & -1 & & & & 1 & \\ & & 1 & -1 & & & & 1 \\ & & & 1 & -1 & & & \end{bmatrix}.$$

Согласно теореме 2, матрица Адамара может быть факторизована следующим образом:

$$H_8 = C_8^{(1)} \cdot C_8^{(2)} \cdot C_8^{(3)},$$

$$\text{где } C_8^{(1)} = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & 1 & 1 & & & & & \\ & 1 & -1 & & & & & \\ & & 1 & 1 & & & & \\ & & 1 & -1 & & & & \\ & & & 1 & 1 & & & \\ & & & & 1 & -1 & & \end{bmatrix},$$

$$C_8^{(2)} = \begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & 1 \\ 1 & & -1 & \\ & 1 & -1 & \\ & & 1 & 1 \\ & & & 1 & 1 \\ & & & & 1 & 1 \\ & & & & & -1 \end{bmatrix}, C_8^{(3)} = \begin{bmatrix} 1 & & & & 1 & & \\ & 1 & & & & 1 & \\ & & 1 & & & & 1 \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & -1 & \\ & & & & & & -1 \\ & & & & & & & 1 \end{bmatrix}.$$

Вычисление векторно-матричного произведения при использовании рассмотренных методов факторизации матриц Адамара требует выполнения 24 операций типа сложение-вычитание. Прямой метод умножения вектора на матрицу  $H_8$  предполагает 56 вычислений.

## 2.5. Преобразование Хаара

Функции Хаара  $\{ har(r, k, t) \}$  были построены в 1910 г. Они образуют полную систему ортонормированных функций, определенных на интервале  $[0, N)$ ,  $N = 2^l$ ,  $l = 1, 2, \dots$  и описываются следующими рекуррентными соотношениями [8]:

$$har(0, 0, t) = 1, \quad t \in x[0, 1];$$

$$har(r, k, t) = \begin{cases} 2^{r/2}, & (k-1)2^{-r} \leq t < (k-0,5)2^{-r}; \\ -2^{r/2}, & (k-0,5)2^{-r} \leq t < k2^{-r}; \\ 0, & \text{при остальных } t \in x[0, 1]; \end{cases}$$

$$0 \leq r < \log_2 N \quad 1 \leq k \leq 2^r.$$

Для  $N = 4$   $r \in 0, 1$ ,  $k \in 1, 2$

$$har(0, 0, t) = 1, \quad t \in x[0, 1]; \quad har(0, 1, t) = \begin{cases} 1, & 0 \leq t < 0,5; \\ -1, & 0,5 \leq t < 1. \end{cases}$$

$$har(1, 1, t) = \begin{cases} 2^{1/2}, & 0 \leq t < 0,25; \\ -2^{1/2}, & 0,25 \leq t < 0,5; \\ 0, & \text{для других } t; \end{cases} \quad har(1, 2, t) = \begin{cases} 2^{1/2}, & 0,5 \leq t < 0,75; \\ -2^{1/2}, & 0,75 \leq t < 1; \\ 0, & \text{для других } t. \end{cases}$$

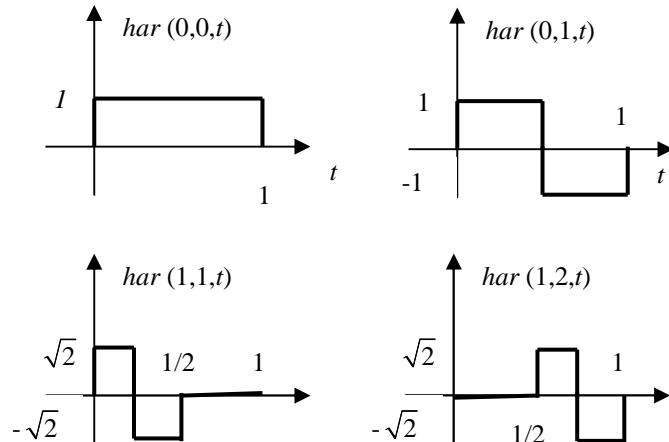


Рис. 2.16. Функции Хаара для  $N = 4$

При дискретизации системы функций Хаара, представленной выше, получаются дискретные функции Хаара  $har(r,k,t)$ , определенные в целочисленных точках  $0, 1, 2, \dots, N - 1$ .

Для  $N = 4$  матрица функций Хаара имеет вид:

$$Hr_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}.$$

Для  $N = 8$  матрица функций Хаара:

$$Hr_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}.$$

Так как функции Хаара ортогональны, то  $Hr_N Hr_N^T = NI$ . Множество функций Хаара, в отличие от функций Уолша, не является мультиплексивным, т. е. произведение функций Хаара не приводит вновь к функции Хаара. С увеличением переменной  $r$  уменьшается интервал, на котором

функция Хаара отлична от нуля. Это обстоятельство позволяет использовать разложение Хаара для получения спектральных коэффициентов, учитывающих локальное поведение функции.

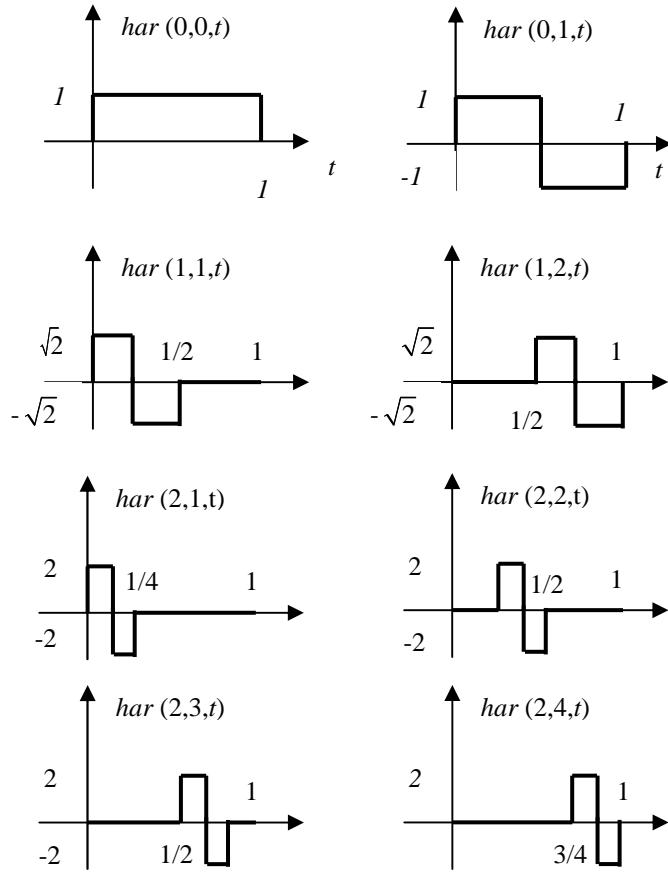


Рис. 2.17. Функции Хаара для  $N = 8$

Пара дискретного преобразования Хаара сигнала  $S$  определяется следующими соотношениями [8]:

$$Y = HrS,$$

$$S = N^{-1}Hr^TY.$$

Первое выражение называется прямым преобразованием Хаара, а второе выражение – обратным преобразованием Хаара.

Например, для сигнала вида  $S = (2, 4, 1, 4)$  спектр сигнала в базисе Хаара определяется:

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ -2\sqrt{2} \\ -3\sqrt{2} \end{bmatrix}.$$

В результате выполнения обратного преобразования Хаара получаем исходный сигнал:

$$S = \frac{1}{4} \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 11 \\ 1 \\ -2\sqrt{2} \\ -3\sqrt{2} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 8 \\ 16 \\ 4 \\ 16 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 1 \\ 4 \end{bmatrix}.$$

Для выполнения преобразования Хаара существует ряд быстрых алгоритмов, на рисунках, представленных ниже, показаны графы быстрого преобразования Хаара для  $N = 4$  (рис. 2.18) и  $N = 8$  (рис. 2.19).

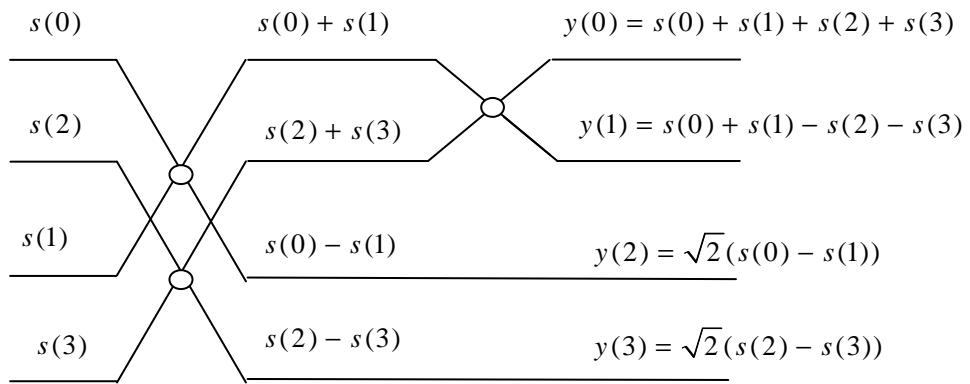


Рис. 2.18. Граф быстрого преобразования Хаара для  $N = 4$

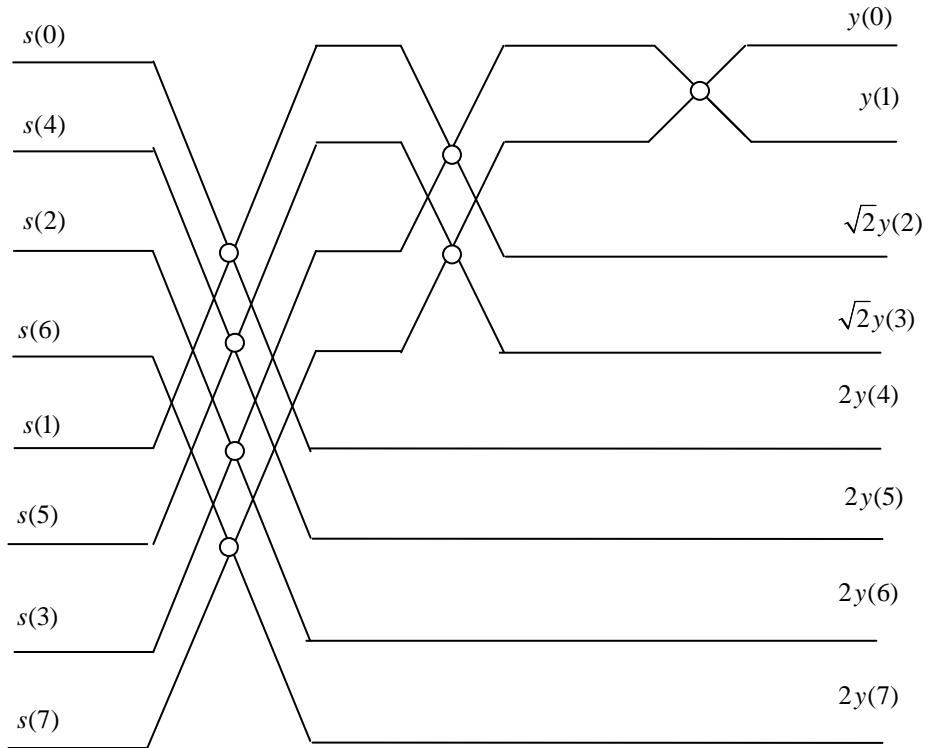


Рис. 2.19. Граф быстрого преобразования Хаара  $N = 8$

## **2.6. Элементы вейвлет-преобразования сигналов**

### **2.6.1. Основы вейвлет-преобразования**

Вейвлеты (wavelet – короткая волна, иногда переводится как всплеск) – функции определенной формы, локализованные по оси аргументов (независимых переменных), инвариантные к сдвигу и линейные к операции масштабирования (сжатия/растяжения). Они создаются с помощью специальных базисных функций, которые определяют их вид и свойства. Впервые этот термин использовали Гроссман и Морле (A. Grossmann, J. Morlet) при анализе свойств сейсмических и акустических сигналов [10].

Основная область применения вейвлетных преобразований – анализ и обработка сигналов и функций, 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 себя проявляют те или иные группы частотных составляющих, или на которых происходят быстрые изменения частотных составляющих сигнала. 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). В отличие от преобразований Фурье, вейвлет-преобразование одномерных сигналов обеспечивает двумерную развертку, при этом частота и координата рассматриваются как независимые переменные, что дает возможность анализа сигналов сразу в двух пространствах.

Одна из главных и особенно плодотворных идей вейвлетного представления сигналов на различных уровнях декомпозиции (разложения) заключается в разделении функций приближения к сигналу на две группы: аппроксимирующую – грубую, с достаточно медленной временной динамикой изменений, и детализирующую – с локальной и быстрой динамикой изменений на фоне плавной динамики, с последующим их дроблением и детализацией на других уровнях декомпозиции сигналов. Это возможно как во временной, так и в частотной областях представления сигналов вейвлетами.

В настоящее время вейвлет-преобразования и вейвлетный анализ используются во многих областях науки и техники для самых различных задач: для распознавания образов, для численного моделирования динамики сложных нелинейных процессов, для анализа аппаратной информации и изображений в медицине, космической технике, астрономии, геофизике, для эффективного сжатия сигналов и передачи информации по каналам с ограниченной пропускной способностью и т. п.

Возможности вейвлетов еще не раскрыты полностью. Однако это не означает, что их развитие приведет к полной замене традиционных средств обработки и анализа информации, хорошо отработанных и проверенных временем. Но оно может существенно расширить инструментальную базу информационных технологий обработки данных.

Непрерывное вейвлет-преобразование (CTWT) есть скалярное произведение сигнала  $s(t)$  и базисных вейвлет-функций  $\Psi_{a,b}$  [10]:

$$CTWT_s(a,b) = \int_{-\infty}^{+\infty} \Psi_{a,b} \cdot s(t) dt,$$

где  $a$  – значение масштаба вейвлет функции;

$b$  – значение сдвига вейвлет функции относительно сигнала;

$$\Psi_{a,b}(n) = \frac{1}{\sqrt{a}} \cdot \Psi\left(\frac{x-b}{a}\right), \quad a \in R^+, \quad b \in R, \quad R – множество вещественных чисел.$$

Базисные функции являются вещественными и колеблются вокруг оси абсцисс. Они определены на некотором интервале.

Параметр  $b$  показывает расположение во времени вейвлета, а  $a$  – параметр масштаба. Большие значения  $a$  соответствуют низким частотам, малые – высоким. Операция умножения на окно как бы содержится в самой базисной функции, которая позволяет сужать и расширять это окно. Отсюда появляется возможность адаптивного к сигналу выбора параметров окна.

Результатом вейвлет-преобразования одномерного числового ряда (сигнала) является двумерный массив амплитуд – значений коэффициентов. Распределение этих значений в пространстве  $(a,b) = (\text{временной масштаб}, \text{временная локализация})$  дает информацию об изменении относительного вклада вейвлетных компонент разного масштаба во времени и называется спектром коэффициентов вейвлет-преобразования, масштабно-временным (частотно-временным) спектром или просто вейвлет-спектром (wavelet spectrum).

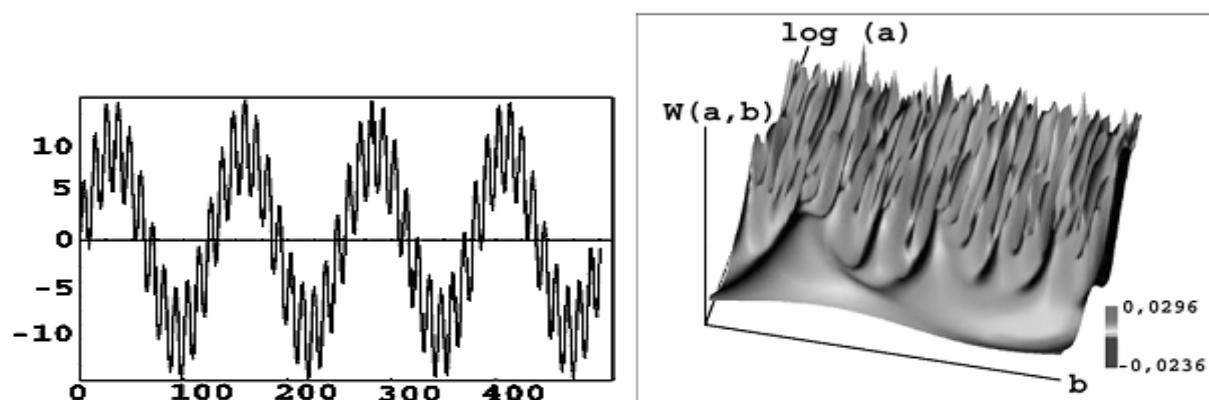


Рис. 2.20. Пример вейвлет преобразования сигнала

Выбор конкретного материнского вейвлета зависит от характера поставленной задачи и от конкретного анализируемого сигнала. Разные сигналы удается анализировать тем или иным способом и критерием успеха может служить, например, простота получаемого разложения.

При вейвлет-преобразовании нет частотного параметра, как это было при ДПФ. Вместо него используется параметр масштаба, который можно определить как величину, обратную частоте. Параметр масштаба в вейвлет-анализе имеет аналогию с масштабом географических карт. Большое значение масштаба соответствуют малому количеству деталей, глобальному представлению сигнала, а низкие значения масштаба позволяют различить детали. Аналогично, в терминах частоты, низкие частоты соответствуют глобальной информации о сигнале (которая содержится на всей его протяженности), а высокие частоты – детальной информации, скрытым особенностям, которые имеют обычно малую протяженность. Масштабирование, как математическая операция, расширяет или сжимает сигнал. Поэтому, при  $a > 1$  расширяет сигнал, а при  $a < 1$  сжимает его.

Наиболее простой пример дискретного вейвлета – это HAAR-вейвлет. Рассмотрим вейвлет-преобразование  $N$ -точечной последовательности отсчетов цифрового сигнала. В качестве примера возьмем 8-точечную последовательность:

$$s(n) = \{20, 16, 14, 10, 8, 6, 4, 2\}.$$

На первом этапе для каждой пары соседних отсчетов вычисляются их средние значения (полусуммы) и полуразности:

$$\begin{aligned} (20 + 16)/2 &= 18; (14 + 10)/2 = 12; (8 + 6)/2 = 7; (4 + 2)/2 = 3; \\ (20 - 16)/2 &= 2; (14 - 10)/2 = 2; (8 - 6)/2 = 1; (4 - 2)/2 = 1. \end{aligned}$$

Полученную последовательность, состоящую из четырех полусумм и четырех полуразностей, можно использовать для восстановления отсчетов исходного сигнала:

$$s^3(n) = \{18, 12, 7, 3, 2, 2, 1, 1\}.$$

Описанная процедура применяется к четырем первым (крупным) элементам последовательности:

$$(18 + 12)/2 = 15; (7 + 3)/2 = 5; (18 - 12)/2 = 3; (7 - 3)/2 = 2.$$

Остальные элементы последовательности  $s^3(n)$  оставим без изменения. Получилась последовательность следующего вида:

$$s^2(n) = \{15, 5, 3, 2, 2, 2, 1, 1\}.$$

Последняя итерация преобразует первые два элемента последовательности  $s^2(n)$  в одно среднее и одну полуразность:

$$(15+5)/2 = 10; (15-5)/2 = 5.$$

Очевидно, что последняя полусумма является средним значением всех восьми отсчетов исходного цифрового сигнала. В итоге получилась преобразованная 8-точечная последовательность:

$$s^1(n) = \{10, 5, 3, 2, 2, 2, 1, 1\},$$

которая состоит из общего среднего и семи полуразностей, выражающих различные уровни детализации исходного сигнала.

Рассмотренная процедура представляет собой вейвлет-преобразование, а результирующая последовательность является вейвлет-образом исходного сигнала.

### 2.6.2. Матричное представление

Рассмотрим 4-точечную последовательность отсчетов цифрового сигнала:

$$s(n) = \begin{bmatrix} s(0) \\ s(1) \\ s(2) \\ s(3) \end{bmatrix}.$$

Согласно рассмотренной выше процедуре вейвлет-преобразования для восьмиточечной последовательности, в данном случае на первом этапе преобразование, состоящее из вычисления двух полусумм и двух полуразностей в матричной форме, определяется следующим образом:

$$s^2(n) = H_W^2 s = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ s(2) \\ s(3) \end{bmatrix}.$$

Верхние две строки матрицы  $H_W^2$  усредняют два соседних отсчета сигнала, а нижние две – детализируют их. Если каждую строку (или каждый

столбец) матрицы  $H_W^2$  считать четырехмерным вектором, то можно заметить, что скалярное произведение любой пары векторов, как сумма произведений соответствующих компонентов, равно нулю, т. е. они образуют ортогональную систему. Норма (длина) каждого вектора равна  $1/\sqrt{2}$ . Для удобства преобразований нормы векторов приводят к единице, умножая все компоненты на  $\sqrt{2}$ . В этом случае формируется ортогональная матрица:

$$H_W^2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

На последнем этапе необходимо сохранить полуразности и преобразовать две полусуммы. Исходя из аналогичных рассуждений, можно записать матрицу вида:

$$H_W^1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Первая строка этой матрицы усредняет отсчеты, вторая – детализирует, третья и четвертая, образуя единичную матрицу, сохраняют предыдущие полуразности. Произведение матриц  $H_W^1$  и  $H_W^2$  дает искомое дискретное вейвлет-преобразование исходной последовательности:

$$H_W = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Матрица  $H_W$  также является ортогональной и представляет собой нормированную матрицу Хаара.

В базисе функций Хаара возможно разложение непрерывного во времени сигнала, заданного также на единичном интервале. При увеличении порядка матрицы  $N$  будет происходить сжатие функций во времени (масштабирование), а сами они станут напоминать короткие всплески, перемещаемые по оси абсцисс.

Подобный процесс составляет сущность непрерывного вейвлет-анализа, основанного на масштабировании и сдвигах. Ограниченные по длительности и перемещаемые во времени вейвлеты в результате свертки с отрезками сигнала позволяют выявить их локальные особенности, чего не предполагают другие методы анализа.

Дискретным аналогом описанного процесса является воздействие матрицы  $H_W$  на вектор отсчетов входного сигнала  $s(n)$ .

Обобщим матричный алгоритм усреднения и детализации на произвольную  $N$ -точечную последовательность, когда  $N = 2^v$  при поэтапном процессе преобразования. На первом шаге используется ортогональная квадратная матрица порядка  $N = 2^v$ :

$$H_W^v = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & & & & \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & & \\ 0 & & \cdots & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Анализ  $H_W^v$  позволяет выделить две новые матрицы, входящие в состав  $H_W^v$ :

$$A_v = \begin{bmatrix} 1 & 1 & 0 & . & . & . & . & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & & & & & \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & . & . & 0 \\ & & & & & & & \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, D_v = \begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & 0 & . & . & . & . & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & & & & & \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & . & . & 0 \\ & & & & & & & \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Таким образом, можно утверждать, что  $H_W^v$  является блочной матрицей, состоящей из двух подматриц  $A_v$  и  $D_v$ , размером  $2^{v-1} \times 2^v$  каждая:

$$H_W^v = \begin{bmatrix} A_v \\ D_v \end{bmatrix}.$$

Подматрицу  $A_v$  можно назвать преобразованием усреднения, а  $D_v$  – преобразованием детализации. Введем векторы-столбцы результатов поэтапного преобразования

$$a_{v-1} = \begin{bmatrix} a_{v-1}^0 \\ . \\ . \\ . \\ a_{v-1}^{2^{v-1}-1} \end{bmatrix} \quad \text{и} \quad d_{v-1} = \begin{bmatrix} d_{v-1}^0 \\ . \\ . \\ . \\ d_{v-1}^{2^{v-1}-1} \end{bmatrix},$$

которые содержат  $2^{v-1}$  элементов и соответственно являются векторами коэффициентов усреднения (низкочастотные коэффициенты) и детализации (высокочастотные коэффициенты). С учетом принятых обозначений вейвлет - преобразование на первом этапе примет вид:

$$H_W^v s = \begin{bmatrix} A_v \\ D_v \end{bmatrix} s = \begin{bmatrix} a_{v-1} \\ d_{v-1} \end{bmatrix}.$$

На каждом шаге вейвлет-преобразования сохраняются высокочастотные коэффициенты  $d$ , а обрабатываются низкочастотные коэффициенты  $a$ . Таким образом, очередной шаг вейвлет-преобразования состоит в

применении к вектору  $a$  операций усреднения и детализации. Чтобы построить часть вейвлет-преобразования, вектор  $d_{v-2}$  сохраняется вместе с  $d_{v-1}$ . Этот процесс продолжается до заключительного шага, на котором получаются скалярные величины  $a$  и  $d$ . Итоговый результат вейвлет-преобразования можно представить вектором-столбцом, содержащим  $N = 2^v$  элементов:

$$w = \begin{bmatrix} a_1 \\ d_1 \\ d_2 \\ d_3 \\ \vdots \\ \vdots \\ d_v \end{bmatrix}.$$

В практических приложениях выдвигается традиционное требование обратимости вейвлет-преобразования, как и любого преобразования. Для обратного вейвлет-преобразования необходимо применить выражение:

$$S = H_W^T w.$$

Очевидно, что результат вейвлет-преобразования сильно зависит от базисных функций, временных сдвигов и от типа вейвлета. На практике масштабирующие множители, сжимающие базисные функции, принимаются равными отрицательным степеням двойки. Соответственно, временные сдвиги выбираются равными положительным степеням двойки. Основное направление в исследовании типов вейвлетов состоит в поисках таких семейств, которые образуют ортогональный базис (ортогональную матрицу преобразований).

Самый простой способ описания вейвлет-преобразований использует произведение матриц. Усложнение конструкции вейвлетов основано на построении взвешенных средних и взвешенных разностей для  $N$ -точечных последовательностей отсчетов входного сигнала ( $N > 2$ ) и некоторого множества коэффициентов  $c_0, c_1, \dots, c_{N-1}$ . Подобный выбор приводит к следующему свойству:

$$c_0c_1 - c_0c_1 + c_2c_3 - c_2c_3 + \dots = 0.$$

По такому принципу построен один из самых используемых вейвлетов – вейвлет Добеши, который принято обозначать  $D4$ , т. к. он основан на четырех коэффициентах  $c_0, c_1, c_2, c_3$ . Матрица этого преобразования имеет вид:

$$D_W = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & 0 & . & . & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & . & . & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 \\ 0 & . & . & 0 & c_0 & c_1 & c_2 & c_3 \\ 0 & . & . & 0 & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & . & . & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & . & . & 0 & c_3 & -c_2 \end{bmatrix}.$$

Если эту матрицу умножить на вектор-столбец отсчетов входного сигнала, то ее первая строка дает взвешенную сумму первых четырех отсчетов. Третья строка определит такую же сумму, но сдвинутую на две временные позиции. Все строки с нечетными номерами зададут аналогичные взвешенные суммы (усреднения). Таким образом, коэффициенты  $c_3, c_2, c_1$  и  $c_0$  определяют фильтр, выходной сигнал которого подлежит децимации с коэффициентом 2. Аналогично четные строки матрицы порождают взвешенные разности (высокочастотные коэффициенты). При этом задается другой фильтр с коэффициентами  $-c_0, c_1, -c_2$  и  $c_3$  и децимацией выходного сигнала.

Требование ортогональности матрицы для ее легкого обращения (транспонирования) приводит к соотношениям:

$$\begin{aligned} c_0^2 + c_1^2 + c_2^2 + c_3^2 &= 1; \\ c_0c_2 + c_1c_3 &= 0. \end{aligned}$$

Одним из возможных решений, удовлетворяющих уравнениям, и является следующий набор коэффициентов:

$$c_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \approx 0.483; \quad c_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \approx 0.837; \quad c_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \approx 0.224; \quad c_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \approx -0.129.$$

Обратное вейвлет-преобразование получается транспонированием матрицы прямого преобразования:

$$D_W^T = \begin{bmatrix} c_0 & c_3 & 0 & . & . & 0 & c_2 & c_1 \\ c_1 & -c_2 & 0 & . & . & 0 & c_3 & -c_0 \\ c_2 & c_1 & c_0 & c_3 & 0 & . & . & 0 \\ c_3 & -c_0 & c_1 & -c_2 & 0 & . & . & 0 \\ 0 & 0 & c_2 & c_1 & c_0 & c_1 & 0 & 0 \\ 0 & 0 & c_3 & -c_0 & c_1 & -c_2 & 0 & 0 \\ 0 & . & 0 & . & 0 & 0 & c_0 & c_3 \\ 0 & . & 0 & . & 0 & 0 & c_1 & -c_2 \end{bmatrix}.$$

Таким образом, в отличие от преобразования Фурье, вейвлет-преобразование определено неоднозначно: каждому вейвлету соответствует свое преобразование. В качестве анализирующих вейвлетов обычно выбираются функции, хорошо локализованные также и в «пространственной области» (т. е. по времени). На рис. 2.21 изображен типичный график вейвлета и его Фурье-образа.

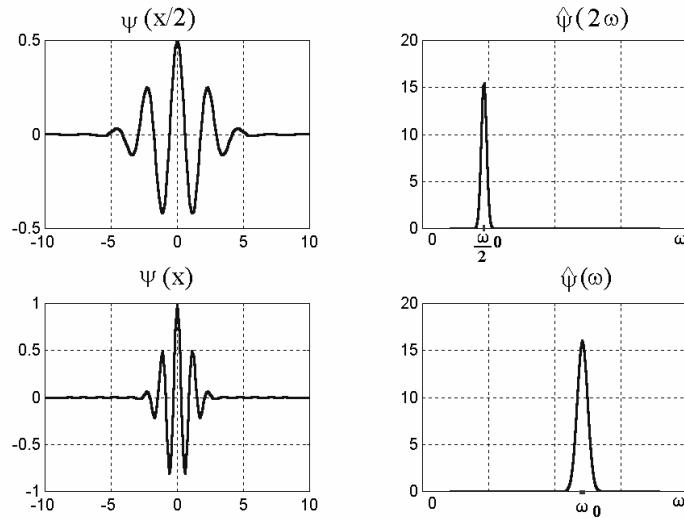


Рис. 2.21. Вейвлет в пространственной и частотной областях

Вейвлет-преобразование широко используется для обработки растровых изображений. Поскольку растровое изображение является двумерным дискретным сигналом, то к нему применимы двумерные дискретные вейвлет-преобразования. Один шаг двумерного вейвлет-преобразования выделяет одну низкочастотную и три высокочастотных компоненты исходного сигнала-изображения. Если не производить никаких дополнительных действий с этими компонентами, то по ним с помощью шага обратно-

го вейвлет-преобразования можно полностью восстановить исходное изображение. Специальным образом обрабатывая низко- и высокочастотные компоненты такого разложения и осуществляя (если нужно) обратные преобразования, можно добиваться различных результатов обработки.

Для вейвлет-преобразования цифровых изображений применяют пирамиду Маллата: для каждой строки цифрового изображения  $f(i, j)$  выполняются операция фильтрации с помощью одномерных низкочастотного фильтра  $L$  (с импульсной характеристикой  $l_k$ ) и высокочастотного фильтра  $H$  (с импульсной характеристикой  $h_k$ ) и операция децимации в два раза. В результате формируются матрица низкочастотных коэффициентов  $f^L(i, j)$  и матрица высокочастотных коэффициентов  $f^H(i, j)$ . Затем для каждого столбца полученных матриц выполняются операции фильтрации и децимации, в итоге формируются четыре матрицы (изображения)  $f^{LL}(i, j)$ ,  $f^{LH}(i, j)$ ,  $f^{HL}(i, j)$  и  $f^{HH}(i, j)$ . Изображение  $f^{LL}(i, j)$  представляет собой аппроксимацию изображения  $f(i, j)$ , а изображения  $f^{LH}(i, j)$ ,  $f^{HL}(i, j)$  и  $f^{HH}(i, j)$  содержат детализирующую информацию.

Для изображения  $f(i, j)$  размером  $N_1 \times N_2$  один шаг вейвлет-преобразования выполняется на основе выражений:

$$\begin{aligned} f^{LL(g+1)}(i, j) &= \sum_{k_1=0}^{L-1} \left( \sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot l_{k_2} \right) \cdot l_{k_1}; \\ f^{HL(g+1)}(i, j) &= \sum_{k_1=0}^{L-1} \left( \sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot h_{k_2} \right) \cdot l_{k_1}; \\ f^{LH(g+1)}(i, j) &= \sum_{k_1=0}^{L-1} \left( \sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot l_{k_2} \right) \cdot h_{k_1}; \\ f^{HH(g+1)}(i, j) &= \sum_{k_1=0}^{L-1} \left( \sum_{k_2=0}^{L-1} f^{LL(g)}(2i+k_1, 2j+k_2) \cdot h_{k_2} \right) \cdot h_{k_1}; \end{aligned}$$

где  $f^{LL(g)}(i, j)$  – аппроксимирующие вейвлет-коэффициенты ( $f^{LL(0)}(i, j) = f(i, j)$ );

$g$ -этап вейвлет-разложения –  $g = \overline{0, n}$  ( $n = \log_2 \lfloor \min(N_1, N_2) \rfloor$ ),

$\lfloor * \rfloor$  – наибольшее ближайшее целое;

$l_{k_1}, l_{k_2}$  и  $h_{k_1}, h_{k_2}$  – коэффициенты низкочастотного и высокочастотного фильтров длиной  $L$  соответственно, причем  $\left( l_{k_1} = l_{k_2} = \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}, \quad h_{k_1} = h_{k_2} = \begin{bmatrix} 1 \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \right)$  и

$$\left( h_{k_1} = h_{k_2} = \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}, \quad -\frac{1}{\sqrt{2}} \end{bmatrix} \right);$$

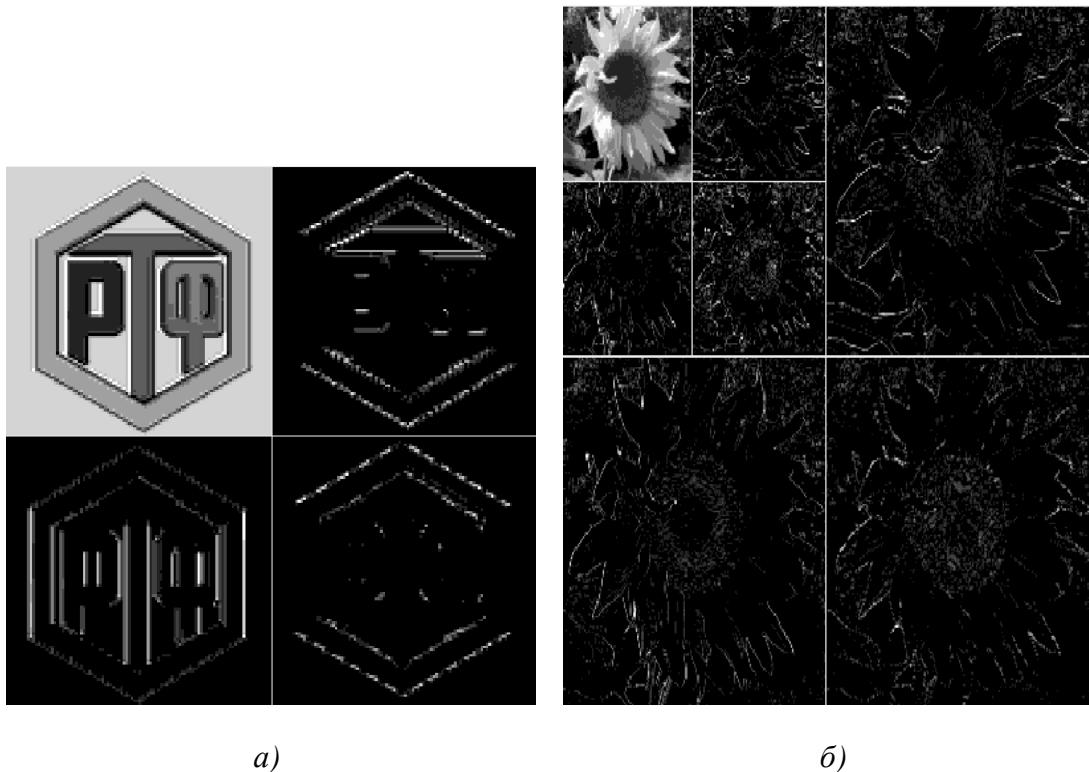
$$i = \overline{0, N_1 / 2^{g+1} - 1}, j = \overline{0, N_1 / 2^{g+1} - 1}.$$


Рис. 2.22. Пример двумерного вейвлет-преобразования:  
 а) один шаг с использованием вейвлет Хаара;  
 б) два шага с использованием вейвлет Добеши

## 2.7. Вопросы и задания для самопроверки

1. Что понимают под спектром сигнала?
2. Запишите выражение для тригонометрической формы ряда Фурье.
3. На основе каких выражений можно определить амплитуду и фазу  $n$ -ой гармоники?
4. Каким образом можно выразить тригонометрические функции через показательные? Запишите формулы.
5. Запишите выражение для системы ДЭФ.
6. Представьте матрицу ДЭФ для  $N = 5$  с минимальными фазами.

7. Сформулируйте и запишите свойства ДЭФ.
8. Графически представьте поворачивающий множитель для  $N = 4$  ( $N = 6$ ,  $N = 8$ ).
9. Запишите пару ДПФ в показательной (матричной) форме.
10. Доказать, что Фурье-базис является ортогональным.
11. Запишите свойства ДПФ.
12. Сформулируйте алгоритм вычисления свертки (корреляции) с использованием ДПФ.
13. Постройте матрицу ДЭФ для выполнения обратного преобразования Фурье для сигнала длиной  $N = 6$ .
14. Назначение и особенности алгоритмов БПФ?
15. Какую операцию алгоритма БПФ называют базовой?
16. Что понимается под числом этапов прореживания БПФ?
17. Как определяется вычислительная сложность алгоритма БПФ с прореживанием по времени (частоте) по основанию 2?
18. Представьте сигнальный граф базовой операции алгоритма БПФ с прореживанием по времени (частоте)?
19. Каков порядок следования входных и выводных отсчетов в алгоритме БПФ с прореживанием по времени (частоте)?
20. Покажите, как с помощью алгоритма БПФ с прореживанием по времени (частоте) получить БПФ восьмиточечной последовательности путем объединения БПФ двух четырехточечных последовательностей?
21. Назовите особенности алгоритма действительного БПФ.
22. С использованием графа алгоритма действительного БПФ вычислить спектр сигнала  $\{s(n)\} = \{1, 2, 2, 2, -2, -2, -2, -1\}$ .
23. Дайте определение импульсной характеристике линейной дискретной системы. Как определить импульсную характеристику системы?
24. Дайте определение переходной характеристике линейной дискретной системы.
25. Что такое свертка и для каких целей она используется в ЦОС? Какие виды свертки можете назвать и в чем их отличие?
26. Вычислить линейную свертку двух сигналов  $s(n) = (10, 10, 8, 6, 8, 10, 10)$  и  $h(n) = (-2, 5, -2)$ .
27. Что такое корреляция и как она может вычисляться?
28. Вычислите ВКФ с помощью БПФ с прореживанием по времени для двух сигналов  $s(n) = (2, 2, -2, -2)$  и  $w(n) = (3, 2, -2, -2)$ . Определите вычислительную сложность.

29. Докажите, что функции Уолша образуют ортонормированную систему.
30. Дайте определение матрице Адамара и запишите способы построения матриц Адамара.
31. Сформулируйте и запишите основные свойства функций Уолша.
32. Дайте определение функциям Радемахера и постройте их для  $N = 16$ .
33. Представьте взаимосвязь различных систем функций Уолша.
34. Запишите пару дискретного преобразования Уолша-Адамара в показательной (в матричной) форме.
35. По какому правилу образуется диадный сдвиг функции?
36. Сформулируйте алгоритм вычисления диадной свертки на основе преобразования Уолша-Адамара.
37. Сколько слабозаполненных матриц образуется при факторизации матрицы Адамара, упорядоченной по Пэли, размером  $128 \times 128$ ?
38. На основе какого выражения можно построить функции Хаара?
39. Запишите пару дискретного преобразования Хаара и сформулируйте его особенности по сравнению с ДПФ?
40. В чем заключается сущность и особенности вейвлет-преобразования сигналов?
41. Представьте матрицы для выполнения прямого и обратного вейвлет-преобразования для сигнала  $N = 8$  с использованием вейвлетов Хаара и Добеши  $D4$ .
42. На основе каких выражений можно выполнить двумерное вейвлет-преобразование?

## Лабораторная работа №1

### Дискретное преобразование Фурье и ключевые операции ЦОС

*Цель работы:* изучение дискретного преобразования Фурье и его основных свойств.

*Задания.* Внимательно изучите дискретное преобразование Фурье и его свойства.

1. Выполнить прямое преобразование Фурье для заданной последовательности, сделать проверку с помощью лабораторного программного обеспечения или прикладных математических пакетов.

2. С использованием лабораторного программного комплекса (закладка «Дискретное преобразование Фурье») выполните следующие пункты задания.

2.1. Получить спектры прямоугольного импульса, шума, дельта-импульса, постоянной составляющей, синусоидального сигнала. Отметить особенности спектров.

2.2. Получить спектр сдвинутой копии сигнала (согласно варианту задания). Оценить взаимосвязь спектров исходного сигнала и его сдвинутой копии.

2.3. Выполнить сдвиг спектральных составляющих сигнала и выполнить обратное преобразование Фурье. Сравнить формы исходного и полученного сигналов.

2.4. Получить спектры действительного сигнала и комплексного сигнала и отметить их отличие.

2.5. Получить спектры четной и нечетной действительной последовательности и отметить их отличие.

2.6. Вычислить корреляционную функцию исходного сигнала и его сдвинутой копии.

2.7. Выполнить свертку сигнала с тестовой последовательностью в соответствии с вариантом задания. Сделать выводы.

2.8. Выполнить свертку дельта функции с тестовой последовательностью. Сделать выводы.

2.9. Вычислить спектры гармонических сигналов с параметрами  $N - \text{const}$ ,  $\tau = T = N/i$ ,  $i = 2,4,8,16$ .

#### Варианты заданий

№ варианта	Тип сигнала	Длина сигнала	Величина сдвига	Вид тестовой последовательности
1	прямоугольный	64	-8	$h = [0.33 \ 0.33 \ 0.33]$
2	гармонический	128	-16	$h = [0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2]$
3	треугольный	256	+32	$h = [-0.25 \ 0.5 \ -0.25]$
4	пилообразный	128	+25	$h = [0.5 \ -0.5]$
5	экспоненциальный	64	-45	$h = [-0.25 \ 0.25 \ -0.25 \ 0.25]$

## **Содержание отчета**

1. Титульный лист: название лабораторной работы, Ф.И.О., группа.
2. Цель работы.
3. Использованные в работе теоретические сведения перед выполнением очередного задания.
4. Результаты работы по п. 1 в рукописном виде, по п. 2.1 – 2.9 в виде графиков. По каждому пункту задания представить вывод.

## **Лабораторная работа №2**

### **Алгоритмы быстрого преобразования Фурье**

*Цель работы:* изучение алгоритмов быстрого преобразования Фурье и сравнение их качественных характеристик.

*Задания.*

1. Внимательно изучите алгоритмы БПФ.
2. Ознакомьтесь с возможностями лабораторного программного обеспечения используя клавишу «F1».
3. Определите быстродействие алгоритмов БПФ и их точность для заданных длин (количество тестов выбирается из расчета 15 – 20 с на выполнение вычислений).
4. Согласно варианту задания постройте граф алгоритма БПФ.
5. Для заданной последовательности выполните прямое преобразование Фурье по построенному графу.

### **Варианты заданий**

<b>№ варианта</b>	<b><math>N</math> для п. 3</b>	<b><math>N</math> и тип алгоритма для п. 4</b>
1	256, 512, 4096, 8192, 16384, 32768	$N = 8$ , АДБПФ
2	512, 1024, 4096, 8192, 16384, 32768	$N = 16$ , БПФ с прореживанием по частоте
3	1024, 2048, 4096, 8192, 16384, 32768	$N = 8$ , БПФ с прореживанием по времени для выполнения обратного преобразования
4	512, 1024, 2048, 8192, 16384, 32768	$N = 8$ , БПФ с прореживанием по частоте для выполнения обратного преобразования
5	128, 512, 1024, 2048, 4096, 8192, 16384, 32768	$N = 16$ , БПФ с прореживанием по времени

## **Содержание отчета**

1. Титульный лист: название лабораторной работы, Ф.И.О., группа.
2. Цель работы.
3. Таблицы, характеризующие быстродействие и точность алгоритмов БПФ.

4. Графики сравнения качественных характеристик алгоритмов для различных длин сигналов (выбирать рациональный масштаб представления графиков).
5. Граф алгоритма БПФ с численными значениями в каждом узле графа.

## Лабораторная работа №3

### Преобразование Уолша-Адамара

*Цель работы:* изучение способов построения матриц Адамара, свойств функций Уолша и способов их упорядочения; изучение преобразования Уолша-Адамара и методики построения алгоритма быстрого преобразования Уолша-Адамара.

#### *Задания.*

1. Внимательно ознакомьтесь с теоретической частью преобразования Уолша-Адамара.
2. Постройте матрицы Адамара, упорядоченные по Пэли и по Уошу размером  $8 \times 8$ .
3. Удостоверьтесь в правильности свойств функций Уолша: проверьте самостоятельно для  $N = 4$ , а для других длин – с помощью лабораторного программного обеспечения (закладка «Функции Уолша»).
4. Изучите преобразование Уолша-Адамара и его свойства: проверьте самостоятельно для  $N = 4$ , а для других длин – с помощью лабораторного программного обеспечения (закладка «Дискретное преобразование Уолша-Адамара»).
5. Изучите методику факторизации матриц методом Гуда.
6. Постройте граф быстрого преобразования Уолша-Адамара для  $N = 8$ .
7. Построить двоично-инверсный и обратный код Грэя для  $n = 4$ .

#### **Содержание отчета**

1. Титульный лист (название лабораторной работы, Ф.И.О., группа).
2. Цель работы.
3. Использованные в работе теоретические сведения при выполнении п. 2 .
4. Результаты работы и вывод по п. 3.
5. Результаты работы и вывод по п. 4.
6. Граф БПУА для  $N = 8$ .

## МОДУЛЬ 3. ЦИФРОВЫЕ ФИЛЬТРЫ

*Цель модуля – изучение студентами методов математического описания, основных типов фильтров и особенностей реализации цифровых фильтров.*

*В результате изучения модуля студенты должны знать:*

- принцип построения структуры цифрового фильтра;
- основные типы ЦФ, их математическое описание и особенности.

*Студенты должны уметь:*

- определять основные параметры фильтров по заданным исходным данным;
- а также иметь представление о процессорах ЦОС и особенностях реализации алгоритмов ЦОС.

### 3.1. Методы математического описания и типы ЦФ

#### 3.1.1. Основы Z-преобразования

В отличие от аналоговых фильтров в цифровых фильтрах используются элементы, которые выполняют функции, отличающиеся от функций резисторов, индуктивностей и емкостей. К таким элементам относятся умножители, сумматоры и элементы задержки на период дискретизации, характеризующиеся зависимостью между входными и выходными параметрами, а не соотношением между током и напряжением. Физически элемент задержки представляет собой регистр, хранящий один предшествующий (задержанный) отсчет сигнала.

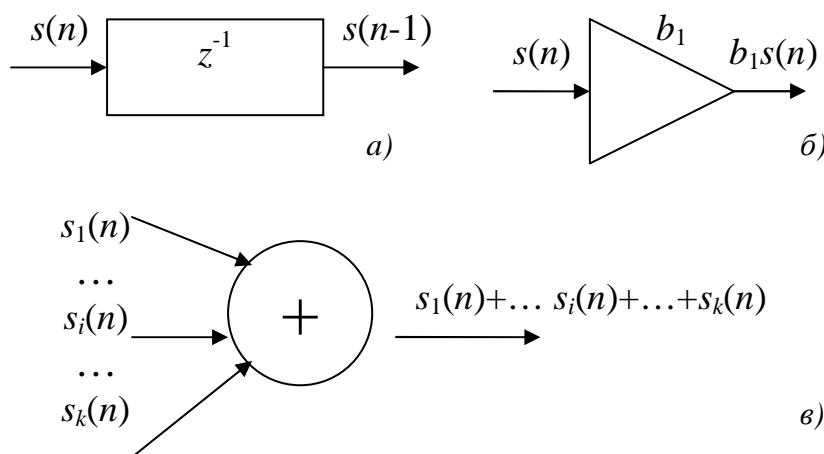


Рис. 3.1. Элементы цифрового фильтра:  
а) элемент задержки; б) умножитель; в) сумматор

Функционирование ЦФ описывается разностным уравнением [11]:

$$y(n) = \sum_{k=0}^{N-1} b(k)s(n-k) - \sum_{k=1}^{M-1} a(k)y(n-k), \quad (3.1)$$

где  $b(k)$ ,  $a(k)$  – постоянные коэффициенты;

$s(n)$ ,  $y(n)$  – отсчеты входного и выходного сигнала соответственно;

$k$  – значение задержки;

$N$ ,  $M$  – константы;

Коэффициенты  $b(k)$  и  $a(k)$  называют внутренними параметрами или просто параметрами ЦФ.

ЦФ, соотношение вход/выход которого описывается в виде разностного уравнения (3.1), отвечает условиям физической реализуемости: при нулевых начальных условиях реакция не может возникнуть раньше воздействия. Значения реакции  $y(n)$  в каждый момент времени  $n$  зависят от текущего и предшествующих значений воздействия, но не зависят от его последующих значений.

Однако поскольку разностные уравнения не позволяют судить о передаточной характеристике ЦФ, для ее определения применяют  $Z$  – преобразование разностного уравнения. Поэтому кратко рассмотрим данное преобразование и его особенности.

Одностороннее  $Z$ -преобразование определяется выражением:

$$X(z) = \sum_{n=0}^{\infty} s(n)z^{-n}, \quad (3.2)$$

где  $z$  – комплексная переменная, которая может быть представлена в двух формах: в алгебраической ( $z = \xi + j\eta$ ) и в показательной ( $z = re^{j\phi}$ ), где радиус  $r$  является модулем, а угол  $\phi$  – аргументом переменной  $z$ :

$$r = |z| = \sqrt{\xi^2 + \eta^2}, \quad \phi = \arg(z) = \arctg\left(\frac{\eta}{\xi}\right).$$

Соответственно положение произвольной точки на комплексной  $Z$ -плоскости может задаваться:

– координатами  $(\xi; \eta)$  – в декартовой системе координат;

– полярными координатами  $(r; \phi)$  – в полярной системе координат.

$Z$ -преобразование однозначно связывает последовательность  $s(n)$  с ее  $Z$ -изображением  $X(z)$  и справедливо только в области абсолютной

сходимости ряда  $\sum_{n=0}^{\infty} |s(n)z^{-n}| < \infty$ , которую называют областью сходимости  $Z$ -изображения.

На комплексной  $Z$ -плоскости это область вне круга радиуса  $R = |a|$ .

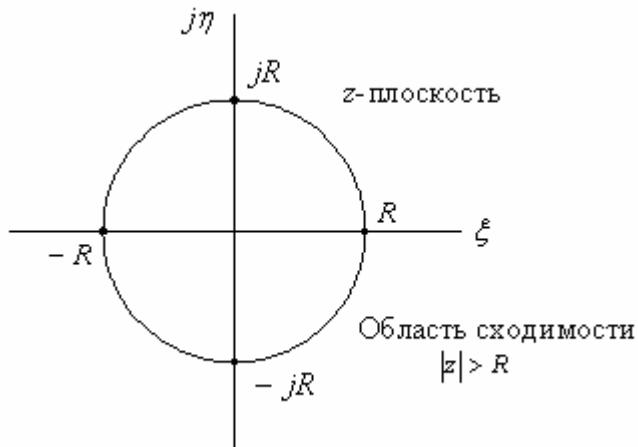


Рис. 3.2. Область сходимости  $Z$ -изображения

Область сходимости определяется свойствами последовательности  $s(n)$ .

Если сравнить выражения для  $Z$ -преобразования и для преобразования Фурье, то можно увидеть, что преобразование Фурье представляет собой частный случай  $Z$ -преобразования, т. е. преобразование Фурье представляет собой  $Z$ -преобразование, вычисленное на единичной окружности ( $R=1$ ) комплексной  $Z$ -плоскости.

*Пример [2]. Найти  $Z$ -преобразование и области сходимости последовательности дискретного времени, приведенной на рис. 3.3.*

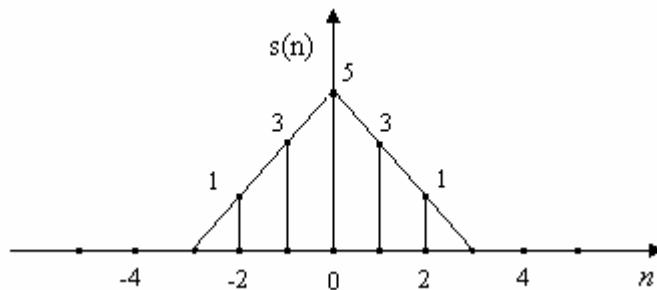


Рис. 3.3. Последовательность дискретного времени

Последовательность имеет конечную длину, значения последовательности  $s(-3) = 0$ ,  $s(-2) = 1$ ,  $s(-1) = 3$ ,  $s(0) = 5$ ,  $s(1) = 3$ ,  $s(2) = 1$ ,  $s(3) = 0$ . Двухстороннее  $z$ -преобразование определяется как:

$$X(z) = \sum_{-\infty}^{\infty} s(n)z^{-n} = z^2 + 3z + 5 + 3z^{-1} + z^{-2}.$$

Обратное  $Z$ -преобразование ( $Z^{-1}$ ) позволяет восстанавливать последовательность дискретного времени  $s(n)$  по ее  $Z$ -образу. В символьической форме обратное  $Z$ -преобразование определяется как:

$$s(n) = Z^{-1}(X(z)).$$

Для определения обратного  $Z$ -преобразования существует ряд способов: с использованием таблицы соответствий; методом разложения на элементарные дроби; методом разложения в степенной ряд; методом вычетов.

Рассмотрим нахождение обратного  $Z$ -преобразования с использованием таблицы соответствий.

*Пример [11]. Найти последовательность  $s(n)$  по ее  $z$ -образу:*

$$X(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}}.$$

Числитель  $X(z)$  – многочлен ненулевой степени, поэтому следует представить в виде суммы дробей:

$$X(z) = \frac{b_0}{1 - a_1 z^{-1}} + \frac{b_1 z^{-1}}{1 - a_1 z^{-1}}.$$

В таблице соответствий находится  $Z$ -изображение с таким же знаменателем и записывается соответствие:

$$X(z) = \frac{1}{1 - a_1 z^{-1}} \rightarrow s(n) = a^n.$$

Таблица 3.1

**Таблица последовательностей и их  $Z$ -преобразования**

Последовательность	$Z$ -изображение
$u_0(n) = \begin{cases} 1, & n = 0; \\ 0, & n \neq 0. \end{cases}$	$U_0(z) = 1$
$u_0(n - m) = \begin{cases} 1, & n = m; \\ 0, & n \neq m. \end{cases}$	$z^{-m} U_0(z) = z^{-m}$
$u_1(n) = \begin{cases} 1, & n \geq 0; \\ 0, & n < 0. \end{cases}$	$U_1(z) = \frac{1}{1 - z^{-1}}$
$u_1(n - m) = \begin{cases} 1, & n \geq m; \\ 0, & n < m. \end{cases}$	$U_1(z) z^{-m} = \frac{z^{-m}}{1 - z^{-1}}$
$s(n) = (\pm a)^n$ $ a  < 1$	$X(z) = \frac{1}{1 \pm az^{-1}}$

Согласно свойству линейности:

$$s(n) = Z^{-1} \left\{ \frac{b_0}{1 - a_1 z^{-1}} \right\} + Z^{-1} \left\{ \frac{b_1 z^{-1}}{1 - a_1 z^{-1}} \right\} = b_0 Z^{-1} \left\{ \frac{1}{1 - a_1 z^{-1}} \right\} + b_1 Z^{-1} \left\{ \frac{z^{-1}}{1 - a_1 z^{-1}} \right\}.$$

На основании теоремы о задержке и соответствия для  $Z$ -преобразования получаем последовательность:

$$s(n) = b_0 a^n + b_1 a^{n-1}.$$

В число областей применения  $z$ -преобразования входят: описание сигналов и систем дискретного времени, позволяющее без труда регулировать степень их устойчивости и визуализировать их частотные характеристики; анализ ошибок квантования цифровых фильтров и вычисление частотных характеристик систем дискретного времени.

### 3.1.2. Рекурсивные и нерекурсивные ЦФ

ЦФ называется рекурсивным, если хотя бы один из коэффициентов  $a(k), k = 1, 2, \dots, M-1$ , разностного уравнения (3.1) не равен нулю [11].

Порядок рекурсивного ЦФ (РЦФ) определяется как:

$$\max\{(M-1), (N-1)\}.$$

Согласно (3.1), реакция  $y(n)$  РЦФ в каждый момент времени  $n$  определяется: текущим отсчетом воздействия  $s(n)$ ; предысторией воздействия  $s(n-k), k = 1, 2, \dots, N-1$ ; предысторией реакции  $y(n-k), k = 1, 2, \dots, M-1$ .

Примеры разностных уравнений РЦФ:

- первого порядка ( $y(n) = b_0 s(n) + b_1 s(n-1) - a_1 y(n-1)$ ),
- второго порядка ( $y(n) = b_0 s(n) + b_1 s(n-1) + b_2 s(n-2) - a_1 y(n-1) - a_2 y(n-2)$ ).

ЦФ называется нерекурсивным (НЦФ), если все коэффициенты  $a(k)$  разностного уравнения (3.1) равны нулю. Для НЦФ разностное уравнение принимают вид:

$$y(n) = \sum_{k=0}^{N-1} b(k) s(n-k). \quad (3.3)$$

Порядок НЦФ определяется как  $(N-1)$ .

Согласно (3.3), реакция  $y(n)$  НЦФ в каждый момент времени  $n$  определяется: текущим отсчетом воздействия  $s(n)$ ; предысторией воздействия  $s(n-k), k = 1, 2, \dots, N-1$ .

Пример разностного уравнения НЦФ второго порядка:

$$y(n) = b_0 s(n) + b_1 s(n-1) + b_2 s(n-2). \quad (3.4)$$

Для РЦФ и НЦФ можно определить импульсную характеристику по разностному уравнению и сравнить результаты на примерах [11].

Для вычисления импульсной характеристики НЦФ второго порядка подадим на вход цифровой единичный импульс, соотношение вход/выход которого описывается разностным уравнением (3.4).

Решим уравнение (3.4) методом подстановки при нулевых начальных условиях:

$$\begin{aligned} y(0) &= b_0 s(0) + b_1 s(-1) + b_2 s(-2) = b_0 \cdot 1 + b_1 \cdot 0 + b_2 \cdot 0 = b_0; \\ y(1) &= b_0 s(1) + b_1 s(0) + b_2 s(-1) = b_0 \cdot 0 + b_1 \cdot 1 + b_2 \cdot 0 = b_1; \\ y(2) &= b_0 s(2) + b_1 s(1) + b_2 s(0) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 1 = b_2; \\ y(3) &= b_0 s(3) + b_1 s(2) + b_2 s(1) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 0 = 0. \end{aligned}$$

Очевидно, что  $y(n) = 0$  при  $n > 2$ .

Таким образом, обобщая полученные результаты на НЦФ произвольного порядка, можно сделать выводы, что импульсная характеристика НЦФ имеет конечную длительность; значения отсчетов импульсной характеристики равны коэффициентам разностного уравнения.

Поэтому НЦФ называют фильтрами с конечной импульсной характеристикой (КИХ-фильтрами).

Теперь вычислим импульсную характеристику РЦФ первого порядка, соотношение вход/выход которой описывается разностным уравнением:

$$y(n) = b_0 s(n) + b_1 s(n-1) - a_1 y(n-1).$$

При  $b_1 = 0$  получим:

$$y(n) = b_0 x(n) - a_1 y(n-1).$$

Для данного уравнения методом прямой подстановки при нулевых начальных условиях имеем:

$$\begin{aligned} y(0) &= b_0 s(0) - a_1 s(-1) = b_0; \\ y(1) &= b_0 s(1) - a_1 y(0) = -a_1 b_0; \\ y(2) &= b_0 s(2) - a_1 y(1) = -a_1 (-a_1 b_0) = a_1^2 b_0; \\ y(3) &= b_0 s(3) - a_1 y(2) = -a_1 (a_1^2 b_0) = -a_1^3 b_0. \end{aligned}$$

Анализируя выражения, можно записать общую формулу вычисления импульсной характеристики для данного случая:

$$y(n) = (-1)^n a_1^n b_0, n = 4, 5 \dots$$

Таким образом, очевидно, что импульсная характеристика РЦФ имеет бесконечную длительность. Поэтому РЦФ называют фильтрами с бесконечной импульсной характеристикой (БИХ-фильтрами). Однако РЦФ могут иметь и конечную импульсную характеристику.

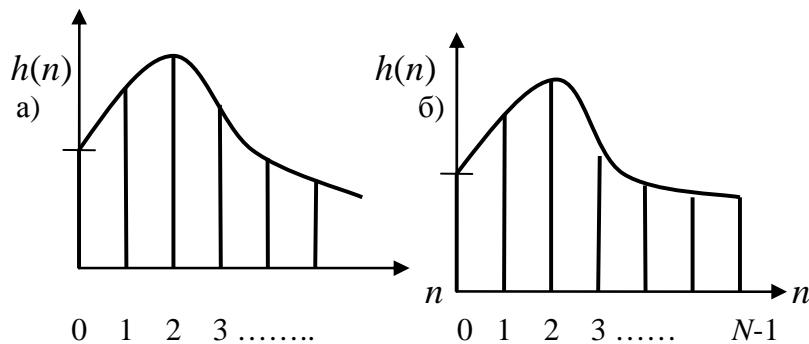


Рис. 3.4. Виды импульсных характеристик: а) БИХ; б) КИХ

По импульсной характеристике можно судить об устойчивости РЦФ. Устойчивому РЦФ отвечает затухающая со временем импульсная характеристика, что математически выражается как:

$$\sum_{n=0}^{\infty} |h(n)| < \infty.$$

### 3.1.3. Передаточная функция

Передаточной функцией  $H(z)$  ЦФ называется отношение  $Z$ -преобразования выходной последовательности к  $Z$ -преобразованию входной последовательности при нулевых начальных условиях [11].

$$H(z) = \frac{Z\{y(n)\}}{Z\{x(n)\}} = \frac{Y(z)}{X(z)}.$$

Таким образом, передаточная функция ЦФ может быть получена путем применения  $Z$ -преобразования к разностным уравнениям. Для РЦФ передаточная функция имеет вид:

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}.$$

Для НРЦ передаточная функция имеет вид:

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k}.$$

Анализ представленных выражений и разностных уравнений показывает, что:

- коэффициенты разностного уравнения являются коэффициентами передаточной функции;
- коэффициенты разностного уравнения  $b(k)$  при  $s(n-k)$  равны коэффициентам числителя передаточной функции при  $z^{-k}$ , при  $k = 0, 1, \dots, N-1$ ;
- коэффициенты разностного уравнения  $-a(k)$  при  $y(n-k)$  равны коэффициентам знаменателя передаточной функции (с обратным знаком) при  $z^{-k}$ , при  $k = 1, \dots, M-1$ .

*Пример.* Известна передаточная функция РЦФ:

$$H(z) = \frac{0,8z^{-1} - 0,3z^{-2}}{1 + 0,3z^{-1} - 0,55z^{-2}}.$$

Разностное уравнение для данного РЦФ имеет вид:

$$y(n) = 0,8s(n) - 0,3s(n-1) - 0,3y(n-1) + 0,55y(n-2).$$

### 3.2. Структуры ЦФ

#### 3.2.1. Основные структуры РЦФ

ЦФ может быть реализован аппаратно или программно. В первом случае – в виде специализированного цифрового устройства на интегральных логических элементах, во втором – в виде программы на компьютере или цифровом процессоре обработки сигналов. В последнее время программная реализация является преобладающей.

Для РЦФ определяют три основные структуры [11]: прямую; каскадную; параллельную.

Прямая структура определяется передаточной функцией  $H(z)$ :

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}},$$

и отображает разностное уравнение:

$$y(n) = \sum_{k=0}^{N-1} b(k)s(n-k) - \sum_{k=1}^{M-1} a(k)y(n-k).$$

На рис. 3.5 приведена прямая структура звена 2-го порядка, описываемого передаточной функцией:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}},$$

и разностным уравнением:

$$y(n) = b_0 s(n) + b_1 s(n-1) + b_2 s(n-2) - a_1 y(n-1) - a_2 y(n-2).$$

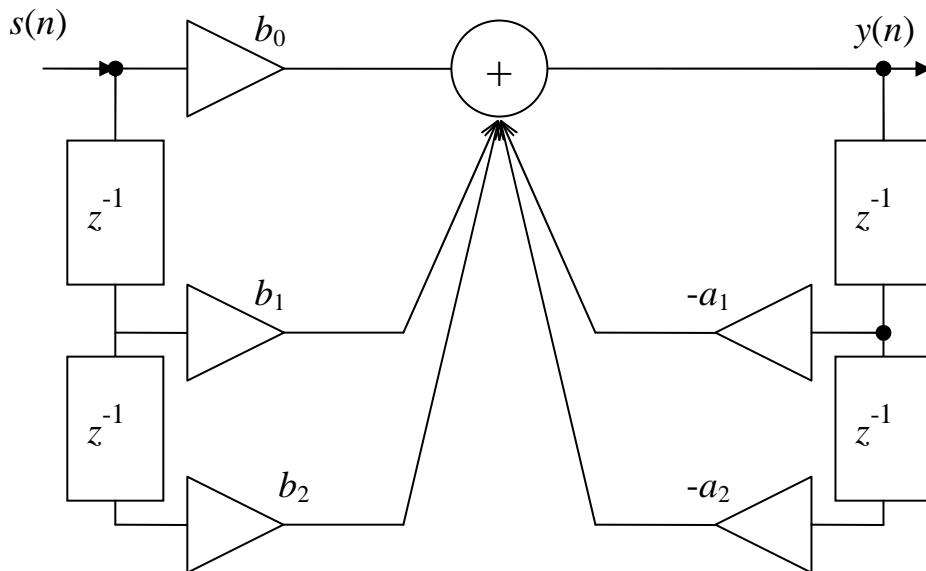


Рис. 3.5. Прямая структура РЦФ 2-го порядка

В общем случае прямая структура РЦФ содержит  $[(N - 1) + (M - 1)]$  элементов задержки, из них:  $(N - 1)$  – для предшествующих отсчетов воздействия и  $(M - 1)$  – для предшествующих отсчетов реакции.

Для уменьшения числа элементов задержки разработана каноническая структура. Структуру называют канонической, если число элементов задержки в ней минимально и равно порядку передаточной функции –  $\max\{(M - 1), (N - 1)\}$ .

Прямая каноническая структура 1 определяется эквивалентным представлением передаточной функции  $H(z)$  в виде произведения двух передаточных функций:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + \sum_{k=1}^{M-1} a_k z^{-k}} \sum_{k=0}^{N-1} b_k z^{-k} = \frac{V(z)}{X(z)} \cdot \frac{Y(z)}{V(z)} = H_1(z)H_2(z),$$

одна из которых описывает рекурсивную часть ЦФ:

$$H_1(z) = \frac{V(z)}{X(z)} = \frac{1}{1 + \sum_{k=1}^{M-1} a_k z^{-k}},$$

а вторая – нерекурсивную:

$$H_2(z) = \frac{Y(z)}{V(z)} = \sum_{k=0}^{N-1} b_k z^{-k}.$$

Передаточным функциям  $H_1(z)$ ,  $H_2(z)$ , согласно их определению, соответствуют разностные уравнения:

$$v(n) = s(n) - \sum_{k=1}^{M-1} a_k v(n-k), \quad y(n) = \sum_{k=0}^{N-1} b_k v(n-k),$$

отображаемые прямыми структурами.

На рис. 3.6 показана прямая структура звена 2-го порядка.

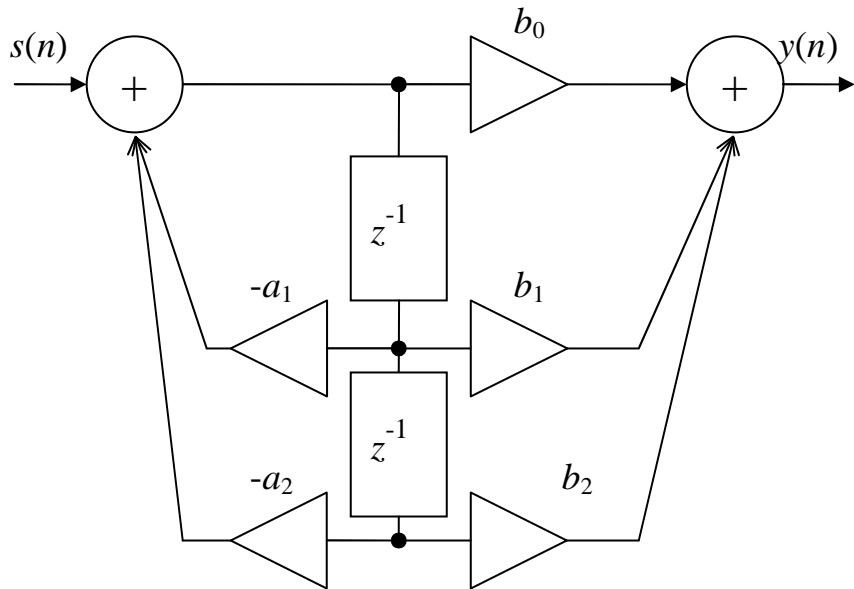


Рис. 3.6. Прямая каноническая структура 1 РЦФ 2-го порядка

Аналогичные канонические структуры 1 типа имеют РЦФ меньшего и большего порядков.

Каноническая структура 2 (рис. 3.7) определяется другим эквивалентным представлением передаточной функции  $H(z)$  и системой разностных уравнений:

$$H(z) = b_0 + \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} \left[ (b_1 - b_0 a_1) z^{-1} + (b_2 - b_0 a_2) z^{-2} \right];$$

$$\begin{cases} y(n) = b_0 s(n) + (b_1 - b_0 a_1) v(n-1) + (b_2 - b_0 a_2) v(n-2); \\ v(n) = s(n) - a_1 v(n-1) - a_2 v(n-2). \end{cases}$$

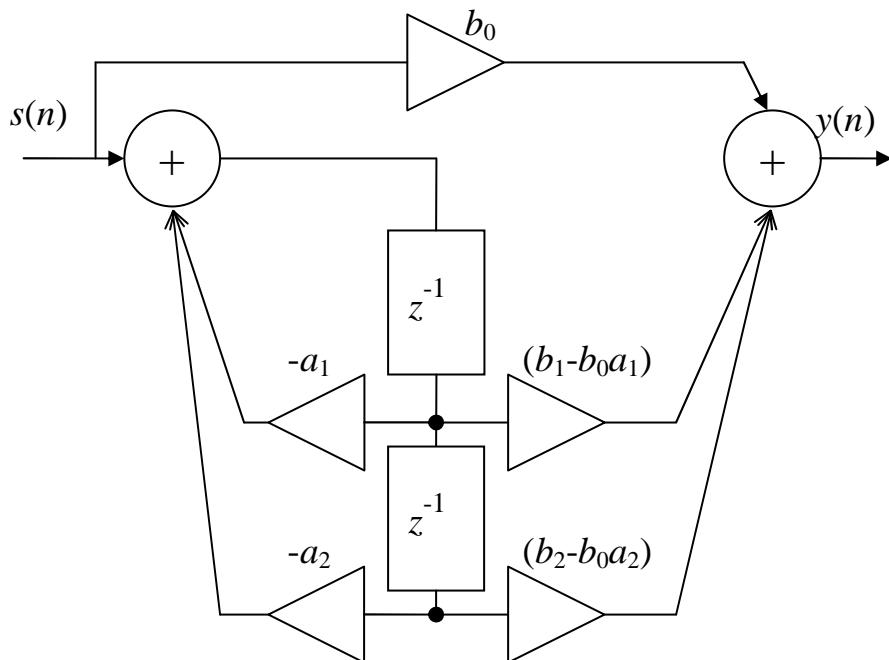


Рис. 3.7. Каноническая структура 2 РЦФ 2-го порядка

На рис. 3.8 приведена каноническая структура 3 РЦФ 2-го порядка. В этом случае передаточная функция и разностное уравнение имеют вид:

$$\begin{cases} H(z) = b_0 + z^{-1} H_1(z); \\ H_1(z) = [b_1 - a_1 H(z)] + z^{-1} H_2(z); \\ H_2(z) = b_2 - a_2 H(z). \end{cases}$$

$$\begin{cases} y(n) = b_0 s(n) + v_1(n-1); \\ v_1(n) = b_1 s(n) - a_1 y(n) + v_2(n-1); \\ v_2(n) = b_2 s(n) - a_2 y(n). \end{cases}$$

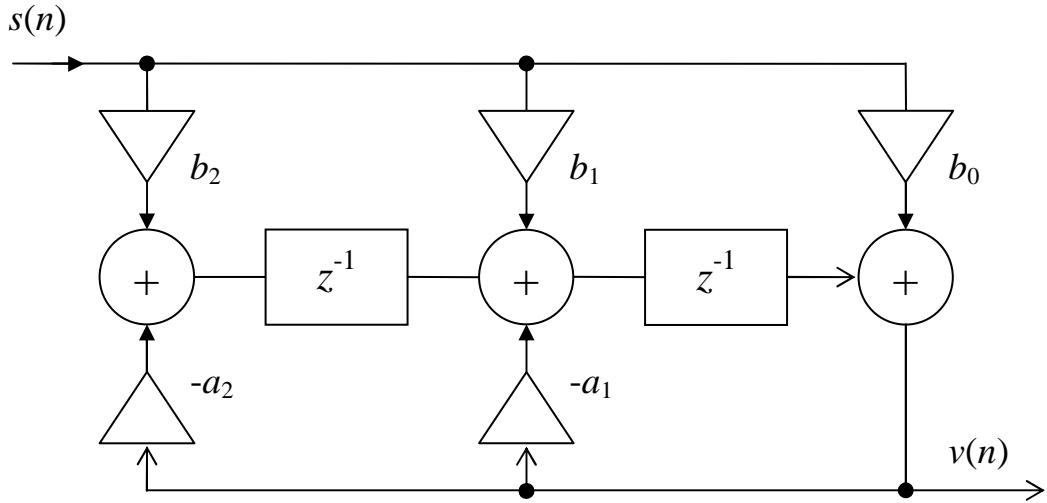


Рис. 3.8. Каноническая структура 3 РЦФ 2-го порядка

### 3.2.2. Каскадная структура

Каскадная структура определяется передаточной функцией  $H(z)$ , представленной в виде произведения множителей второго порядка:

$$H(z) = \prod_{i=1}^K H_k(z) = \prod_{i=1}^K \left( \frac{b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \right),$$

где  $b_{0i}, b_{1i}, b_{2i}, a_{1i}, a_{2i}$  – вещественные коэффициенты;

$K$  – количество РЦФ 2-го порядка.

При прямой структуре всех звеньев данному виду передаточной функции соответствует система разностных уравнений:

$$\begin{cases} v_1(n) = b_{01}s(n) + b_{11}s(n-1) + b_{21}s(n-2) - a_{11}v_1(n-1) - a_{21}v_1(n-2); \\ v_2(n) = b_{02}v_1(n) + b_{12}v_1(n-1) + b_{22}v_1(n-2) - a_{12}v_2(n-1) - a_{22}v_2(n-2); \\ \dots \\ y(n) = b_{0,K-1}v_{K-1}(n) + b_{1,K-1}v_{K-1}(n-1) + b_{2,K-1}v_{K-1}(n-2) - a_{1,K-1}y(n-1) - a_{2,K-1}y(n-2), \end{cases}$$

из которой следует, что реакция  $i$ -го звена –  $I = 1, 2, \dots, (K-1)$ , служит воздействием для  $(i+1)$ -го звена, поэтому данная система отображается каскадным соединением РЦФ 2-го порядка – каскадной (последовательной) структурой (рис. 3.9).

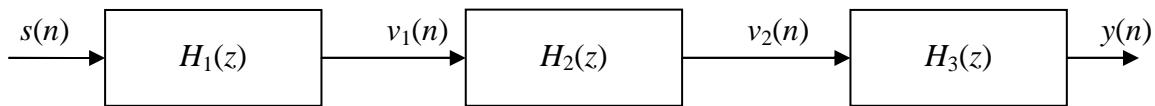


Рис. 3.9. Каскадная структура из трех РЦФ 2-го порядка

### 3.2.3. Параллельная структура

Параллельная структура определяется передаточной функцией  $H(z)$ , представленной в виде суммы дробей второго порядка (в частном случае) [11]:

$$H(z) = \sum_{i=1}^K H_i(z) = \sum_{i=1}^K \left( \frac{b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \right),$$

где  $b_{0k}, b_{1k}, b_{2k}, a_{1k}, a_{2k}$  – вещественные коэффициенты,  
 $K$  – количество звеньев 2-го порядка.

Разностное уравнение определяется:

$$y(n) = \sum_{i=1}^K v_i(n),$$

где  $v_i(n) = b_{0i}x(n) + b_{1i}s(n-1) + b_{2i}s(n-2) - a_{1i}v_i(n-1) - a_{2i}v_i(n-2)$ .

Из разностного уравнения для параллельной структуры следует, что воздействие для всех РЦФ одинаковое, а реакция равна сумме реакций отдельных РЦФ, поэтому данное РУ отображается параллельным соединением рекурсивных звеньев 2-го порядка – параллельной структурой (рис. 3.10).

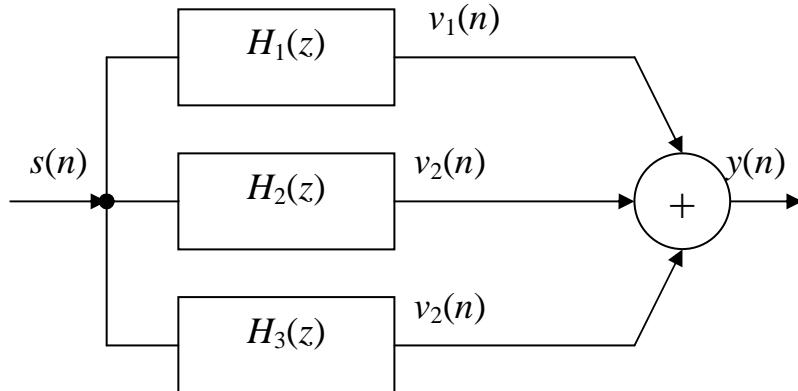


Рис. 3.10. Параллельная структура из трех РЦФ 2-го порядка

### 3.2.4. Структуры НЦФ

Передаточная функция НЦФ может рассматриваться как частный случай  $H(z)$  общего вида при  $a_k = 0, k = 1, 2, \dots, M - 1$ . В этом случае знаменатель  $H(z)$  оказывается равным 1.

НЦФ соответствуют два основных типа структур: прямая и каскадная.

Прямая структура (рис. 3.11) определяется передаточной функцией  $H(z)$ , представленной в виде рациональной функции:

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k},$$

и отображает разностное уравнение:  $y(n) = \sum_{k=0}^{N-1} b_k s(n-k)$ .

Каскадная структура определяется передаточной функцией, представленной в виде произведения множителей второго порядка:

$$H(z) = \prod_{i=1}^K H_i(z) = \prod_{i=1}^K (b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}),$$

где  $b_{0i}, b_{1i}, b_{2i}$  – вещественные коэффициенты;

$K$  – количество НЦФ 2-го порядка.

Система разностных уравнений НЦФ звеньев 2-го порядка описывается:

$$\begin{cases} v_1(n) = b_{01}s(n) + b_{11}s(n-1) + b_{21}s(n-2); \\ v_2(n) = b_{02}v_1(n) + b_{12}v_1(n-1) + b_{22}v_1(n-2); \\ \dots \\ y(n) = b_{0,K-1}v_{K-1}(n) + b_{1,K-1}v_{K-1}(n-1) + b_{2,K-1}v_{K-1}(n-2). \end{cases}$$

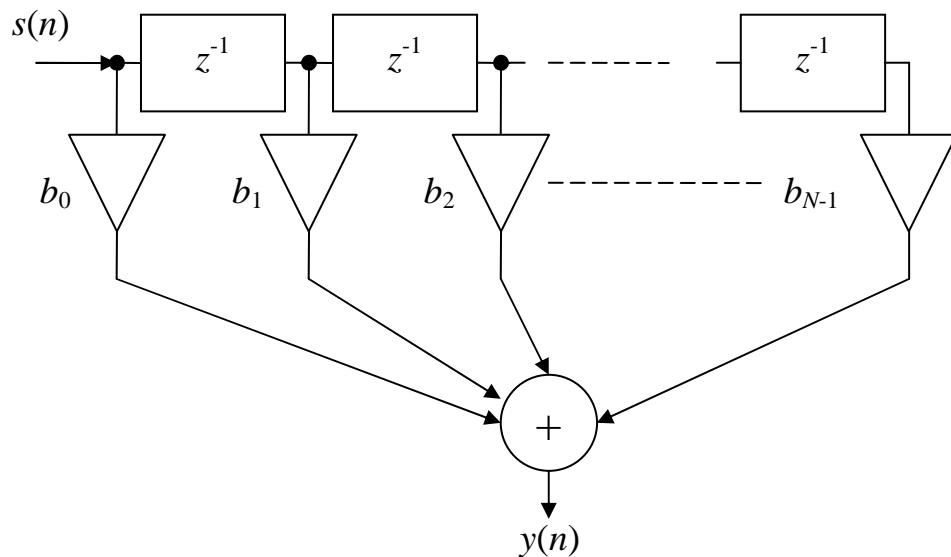


Рис. 3.11. Прямая структура НЦФ

Поскольку одна и та же передаточная функция может быть представлена в разных видах, возникает вопрос о выборе структуры ЦФ, для решения которого прежде всего используются следующие положения:

– для ЦФ вид передаточной функции определяется методом синтеза, тем самым структура по существу оказывается автоматически выбранной: для рекурсивных фильтров обычно она каскадная или параллельная, а для нерекурсивных – прямая приведенная;

– при выборе структуры отдельных ЦФ и последовательности их расположения в каскадном соединении необходимо иметь в виду, что от этого зависит погрешность вычислений – собственные шумы системы.

### 3.3. Частотные характеристики и элементы синтеза ЦФ

#### 3.3.1. Общие сведения

Частотной характеристикой (ЧХ)  $H(e^{j2\pi\hat{f}})$  дискретного фильтра называется отношение преобразования Фурье выходной последовательности к преобразованию Фурье входной последовательности при нулевых начальных условиях [11]:

$$H(e^{j2\pi\hat{f}}) = \frac{F\{y_n\}}{F\{x_n\}} = \frac{Y\{e^{j2\pi\hat{f}}\}}{X\{e^{j2\pi\hat{f}}\}}.$$

Частотная характеристика представляет собой коэффициент передачи фильтра для каждого значения частоты  $\hat{f} = \frac{f}{f_D} = fT$ .

Из сравнения выражений для частотной характеристики и передаточной функции видно, что ЧХ равна передаточной функции фильтра, вычисленной на единичной окружности  $z$ -плоскости (при  $z = e^{j2\pi\hat{f}} = e^{j\varphi}$ ):

$$H(e^{j2\pi\hat{f}}) = H(z) \Big|_{z = e^{j2\pi\hat{f}}}.$$

Можно установить связь между значениями нормированной частоты  $\hat{f}$  и соответствующим углом  $\varphi$  в  $z$ -плоскости. Поскольку:

$$e^{j2\pi\hat{f}} = \cos 2\pi\hat{f} + j \sin 2\pi\hat{f} = e^{j\varphi} = \cos \varphi + j \sin \varphi,$$

то  $\hat{f} = \varphi / 2\pi$ .

На рис. 3.12 показана связь точек единичной окружности и частот  $\hat{f}$ :

- точке  $z = 1 = \cos 0 + j \sin 0$  соответствует  $\hat{f} = 0$ ;
- точкам  $z = \pm j = \cos(\pm\pi/2) + j \sin(\pm\pi/2)$  соответствуют  $\hat{f} = \pm 0.25$ .

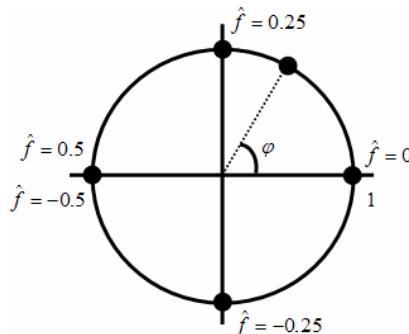


Рис. 3.12. Связь точек единичной окружности и частот  $\hat{f}$

Подставляя значения нормированной частоты, получаем:

$$e^{j2\pi\hat{f}} = \cos 2\pi\left(\pm\frac{1}{4}\right) + j \sin 2\pi\left(\pm\frac{1}{4}\right) = \pm j,$$

Точки  $z = -1 = \cos(\pm\pi) + j \sin(\pm\pi)$  соответствуют  $\hat{f} = \pm 0.5$ .

### 3.3.2. Характеристики КИХ-фильтров и БИХ-фильтров

Используя выражения для передаточных функций для нерекурсивных и рекурсивных фильтров, можно записать общие формулы для частотных характеристик дискретных фильтров [11].

КИХ-фильтр:

$$H(e^{j2\pi\hat{f}}) = \sum_{k=0}^{N-1} b_k e^{-j2\pi\hat{f}k}. \quad (3.5)$$

БИХ-фильтр:

$$H(e^{j2\pi\hat{f}}) = \frac{\sum_{k=0}^{N-1} b_k e^{-j2\pi\hat{f}k}}{\left(1 + \sum_{k=1}^{M-1} a_k e^{-j2\pi\hat{f}k}\right)}. \quad (3.6)$$

Как уже известно, частотная характеристика  $H(e^{j2\pi\hat{f}})$  – комплексная функция. Ее можно представить в виде:

$$H(e^{j2\pi\hat{f}}) = H_{\text{Re}}(e^{j2\pi\hat{f}}) + jH_{\text{Im}}(e^{j2\pi\hat{f}}) = A(\hat{f}) \cdot e^{j\Phi(\hat{f})}.$$

Амплитудно-частотной характеристикой (АЧХ) называется функция

$$A(\hat{f}) = \left| H(e^{j2\pi\hat{f}}) \right| = \sqrt{H_{\text{Re}}^2(e^{j2\pi\hat{f}}) + H_{\text{Im}}^2(e^{j2\pi\hat{f}})}.$$

Фазо-частотной характеристикой (ФЧХ) называется функция

$$\Phi(\hat{f}) = \arg H(e^{j2\pi\hat{f}}) = \tan^{-1}(H_{\text{Im}}(e^{j2\pi\hat{f}})/H_{\text{Re}}(e^{j2\pi\hat{f}})).$$

АЧХ устанавливает связь между амплитудами, а ФЧХ – между фазами выходного  $y(n)$  и входного  $s(n)$  сигналов фильтра.

Если  $s(n) = A_{ex} \cdot \cos(2\pi\hat{f}n + \Phi_{ex})$ , то в установившемся режиме:

$$y(n) = A_{\text{вых}} \cdot \cos(2\pi\hat{f}n + \Phi_{\text{вых}}),$$

где  $A_{\text{вых}} = A_{ex} \cdot A(\hat{f})$ ,

$$\Phi_{\text{вых}} = \Phi_{ex} + \Phi(\hat{f}).$$

Общие формулы АЧХ КИХ-фильтра и БИХ-фильтра имеют вид:

$$A_{KIH}(\hat{f}) = \sqrt{\left( \sum_{k=0}^{N-1} b_k \cos 2\pi \hat{f} k \right)^2 + \left( \sum_{k=0}^{N-1} b_k \sin 2\pi \hat{f} k \right)^2},$$

$$A_{BIH}(\hat{f}) = \sqrt{\frac{\left( \sum_{k=0}^{N-1} b_k \cos 2\pi \hat{f} k \right)^2 + \left( \sum_{k=0}^{N-1} b_k \sin 2\pi \hat{f} k \right)^2}{\left( 1 + \sum_{k=1}^{M-1} a_k \cos 2\pi \hat{f} k \right)^2 + \left( \sum_{k=0}^{N-1} a_k \sin 2\pi \hat{f} k \right)^2}}.$$

Частотную характеристику можно рассматривать как спектр импульсной характеристики:

$$H(e^{j2\pi \hat{f}}) = \sum_{k=0}^{\infty} h_k e^{-j2\pi \hat{f} k}.$$

Следовательно, свойства ЧХ фильтра во многом совпадают со свойствами спектра дискретного сигнала. Ниже приведены основные свойства частотных характеристик дискретных фильтров [11]:

- частотная характеристика является периодической функцией по частоте, с периодом, равным частоте дискретизации;
- АЧХ и ФЧХ также являются периодическими функциями с тем же периодом;
- для фильтров с вещественными коэффициентами АЧХ – четная функция, а ФЧХ – нечетная функция;
- для полного описания частотной характеристики фильтра с вещественными коэффициентами достаточно задать ее в основной полосе частот  $\hat{f} \in [0, 0.5]$ .

Дискретный фильтр называют нормированным, если максимальное значение АЧХ равно 1. Нормировка фильтра достигается путем умножения коэффициентов числителя передаточной функции на нормирующий множитель  $q_n = 1/A_{\max}$ , где  $A_{\max}$  – максимальное значение АЧХ на интервале  $\hat{f} \in [0, 0.5]$ .

Рассмотрим требования к частотно-избирательным ЦФ. На рис. 3.13 представлены идеальные АЧХ фильтров: НЧ, ВЧ, полосового и режекторного, которые, вследствие их физической невозможности, необходимо аппроксимировать при заранее определяемых допусках, зависящих от ряда практических ограничений. Эти ограничения связаны, прежде всего, с назначением синтезируемого фильтра. Все ограничения и допуски составляют требования, предъявляемые к фильтру.

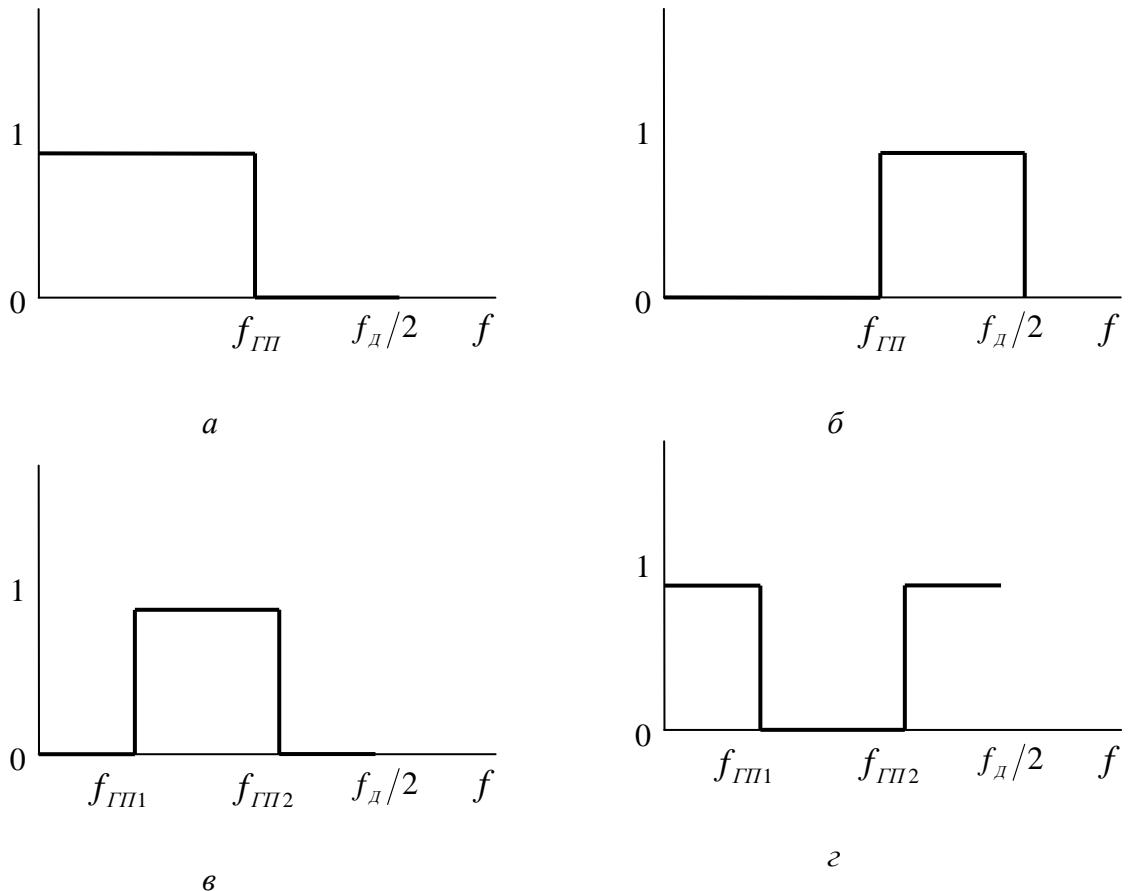


Рис. 3.13. Идеальные АЧХ фильтров:  
а) НЧ, б) ВЧ, в) полосового, г) режекторного

*Основные требования к фильтрам включают в себя [11]:*

- задание частоты дискретизации  $f_d$  [Гц] и типа избирательности фильтра (НЧ, ВЧ и т. д.);
- задание требований к АЧХ  $A(f)$  или к характеристике ослабления (затухания)  $a(f)$ , представляющей собой логарифмическую АЧХ.

При этом необходимо помнить, что:

- задание требований начинается с установки всех граничных частот фильтра только в основной полосе частот  $0 \leq f \leq f_d/2$ ;
- требования в переходных полосах избирательных фильтров не задаются;
- требования формулируются в виде допустимых отклонений от нормированной АЧХ  $\hat{A}(f)$  [или от  $a(f)$ ] в полосах пропускания и задерживания и отображаются на диаграмме допусков;
- АЧХ  $A(f)$  является безразмерной величиной и выражается в абсолютных единицах;

– характеристики ослабления и затухания обозначаются одинаково  $a(f)$ , размерность  $a(f)$  выражается в децибелах, связь между  $A(f)$  и характеристикой ослабления  $a(f)$  определяется формулой:

$$a(f) = 20 \lg \hat{A}(f),$$

а при отображении требований в виде характеристики затухания:

$$a(f) = -20 \lg \hat{A}(f);$$

– выбор метода аппроксимации АЧХ  $\hat{A}(f)$  (или  $a(f)$ ).

Рассмотрим пример реализации полосового КИХ-фильтра [11].

Начальные условия: тип – полосовой; аппроксимация – по Чебышеву; частота дискретизации  $f_d = 48$  кГц; граничные частоты  $f_{z31} = 7,2$  кГц;  $f_{z21} = 9,6$  кГц;  $f_{z22} = 12,96$  кГц;  $f_{z32} = 15,36$  кГц; допуски на отклонение АЧХ от номинального уровня в полосе пропускания  $\epsilon_n = 0,05$  и в полосе задерживания  $\epsilon_3 = 0,01$ ; коэффициенты квантованы до 24 бит,  $N = 35$ .

Важным является то обстоятельство, что в соответствии с методом синтеза КИХ-фильтров отклонение АЧХ задается симметрично, а для БИХ-фильтров отклонение задается только в одну сторону так, чтобы АЧХ не превышала единицы. Переходная полоса располагается между полосами пропускания и задерживания, ее ширина  $\Delta f_{ПЗ} = f_k - f_\chi$ .

*Полосовой фильтр* (ПФ) характеризуется пятью частотными полосами (рис. 3.14), из которых центральная полоса пропускания, две полосы задерживания и две переходных полосы. На рис. 3.14 обозначены [11]:

- $f_{Г31}$  – граничная частота первой полосы задерживания;
- $f_{ГП1}$  – левая частота среза полосы пропускания;
- $f_{ГП2}$  – правая частота среза полосы пропускания, ширина полосы пропускания  $\Delta f_{ПП} = f_{ГП2} - f_{ГП1}$ ;
- $f_{Г32}$  – граничная частота второй полосы задерживания.

Расчет коэффициентов фильтра выполняется на ЭВМ с помощью специальных прикладных пакетов, например MATLAB. Для заданного фильтра коэффициенты представлены в таблице 3.2 [11], АЧХ – на рис. 3.14 [11].

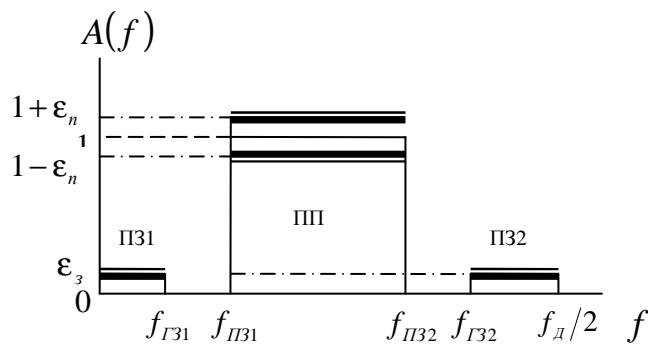


Рис. 3.14. Диаграмма требований к АЧХ ПФ

Таблица 3.2

**Рассчитанные коэффициенты фильтра**

i	$b_k, \quad k = i + j$				
	$j = 0$	$j = 1$	$j = 2$	$j = 3$	$j = 4$
0	-0,0058384	-0,0002581	0,0186806	0,0069064	-0,0268698
5	-0,0114828	0,0221347	0,0109901	-0,0019101	0,0148118
10	-0,0294373	-0,0680606	0,0525404	0,1389393	-0,0501657
15	-0,2012113	0,0208654	0,2256697	0,0208654	-0,2012113
20	-0,0501657	0,1389393	0,0525404	-0,0680606	-0,0294373
25	0,0148118	-0,0019101	0,0109901	0,0221347	-0,0114828
30	-0,0268698	0,0069064	0,0186806	-0,0002581	-0,0058384

Согласно заданию, граничные частоты  $f_{z31} = 7,2$  кГц;  $f_{en1} = 9,6$  кГц;  $f_{en2} = 12,96$  кГц;  $f_{z32} = 15,36$  кГц.

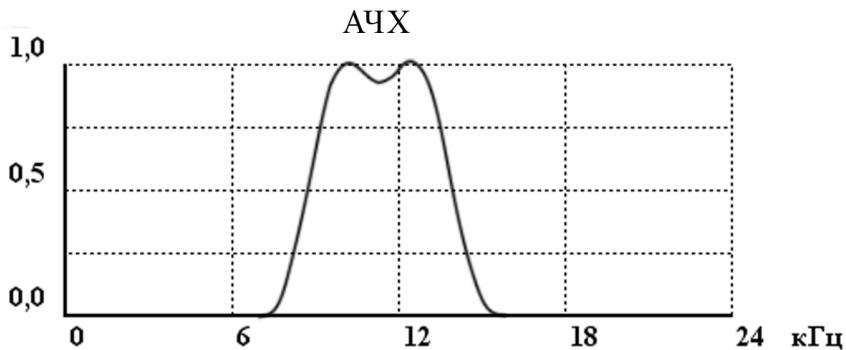


Рис. 3.15. Рассчитанная АЧХ фильтра

**3.4. Реализация алгоритмов ЦОС на процессорах**

Цифровую обработку сигналов в реальном времени можно разбить на две категории [2]: потоковая обработка, например, цифровая фильтрация, где данные обрабатываются по выборке за такт, и блоковая обработка, например, нахождение БПФ и корреляции, где за такт обрабатывается блок данных фиксированной длины. Для реализации алгоритмов ЦОС в реальном времени требуется и программное, и аппаратное обеспечение. В качестве аппаратной поддержки можно использовать матрицы процессоров, стандартные микропроцессоры, чипы ЦОС или микропрограммируемые специализированные устройства. Программной поддержкой служат коды или микрокоды на языке ассемблера низкого уровня, «родном» для аппаратуры ЦОС, и/или коды на эффективном языке высокого уровня, таком как С или С++. В настоящее время обычно используют именно языки высокого уровня, особенно с новейшими процессорами ЦОС, имеющими сложную структуру.

В число распространенных процессоров ЦОС входят такие устройства с фиксированной запятой, как TMS320C55x от Texas Instruments и DSP563x от Motorola, и такие устройства с плавающей запятой, как TMS320C4x (Texas Instruments) и ADSP21xxx SHARC (Analog Devices).

### **3.4.1. Компьютерные архитектуры обработки сигналов**

Большинство доступных сейчас универсальных процессоров основаны на архитектуре фон Неймана, при которой операции выполняются последовательно. Упрощенная архитектура стандартного процессора фон Неймана показана на рис. 3.16 [2]. При обработке инструкции в таком процессоре блоки процессора, не задействованные в каждой фазе инструкции, находятся в холостом состоянии до передачи им управления. Повышение скорости процессора достигается за счет ускорения работы отдельных блоков, хотя существует определенный предел, ограничивающий их возможную скорость работы.

Если устройство должно работать в реальном времени, архитектуру процессора ЦОС нужно оптимизировать под выполнение функций ЦОС. Например, на рис. 3.17 показана общая аппаратная архитектура, подходящая для цифровой обработки сигналов в реальном времени.

Она характеризуется следующими особенностями [2]:

1. Многошинная структура с раздельной памятью для данных и инструкций программы. Обычно память для хранения данных содержит входные данные, промежуточные значения и выходные выборки, а также фиксированные коэффициенты, например, для цифровой фильтрации или БПФ. Команды программы хранятся в специально отведенных ячейках памяти.

2. Порт ввода-вывода позволяет обмениваться данными со внешними устройствами (АЦП, ЦАП) или передавать цифровые данные другим процессорам. Прямой доступ к памяти (Direct Memory Access – DMA) (если он есть) позволяет быстро обмениваться блоками данных с памятью (ОЗУ) для хранения данных, причем обычно это происходит под внешним управлением.

3. Арифметические устройства для логических и арифметических операций, в число которых выходят АЛУ, аппаратные умножители и схемы сдвига (или умножители-накопители).

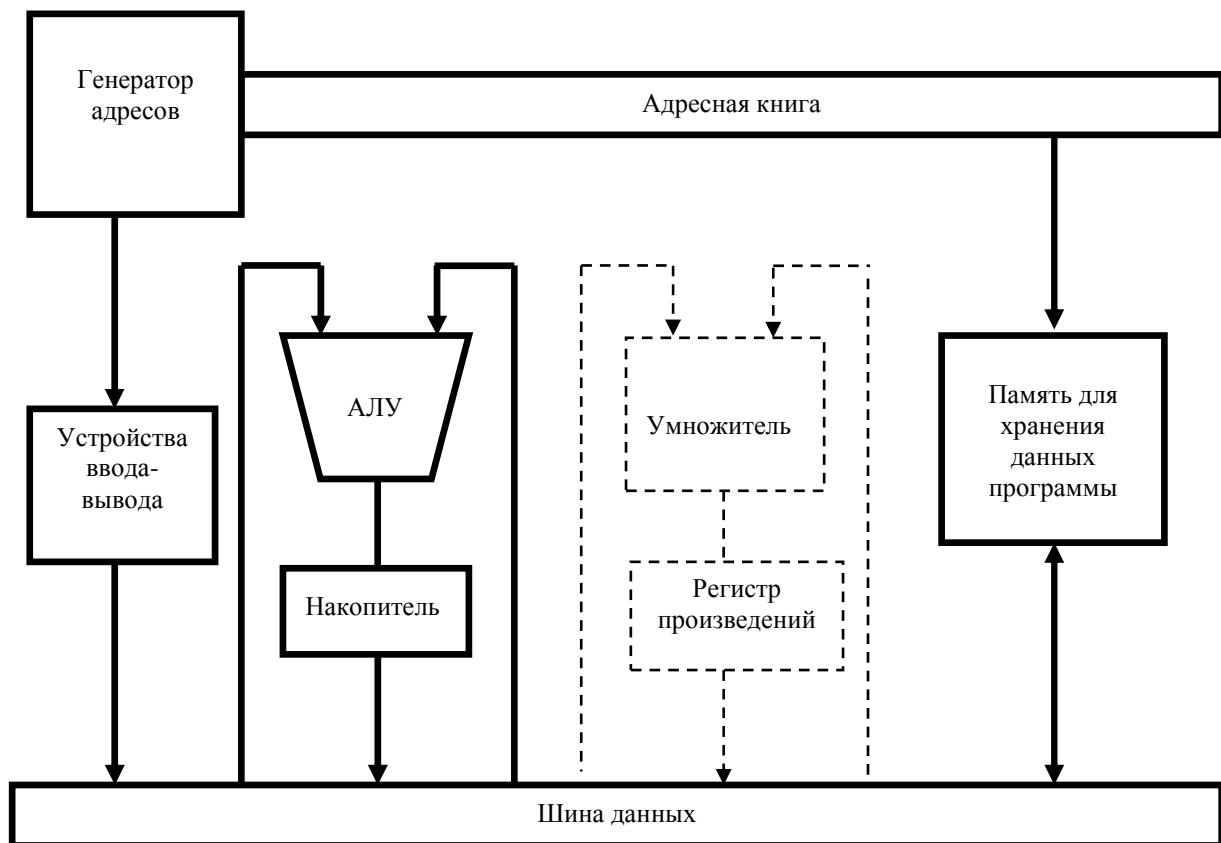


Рис. 3.16. Упрощенная архитектура стандартных процессоров

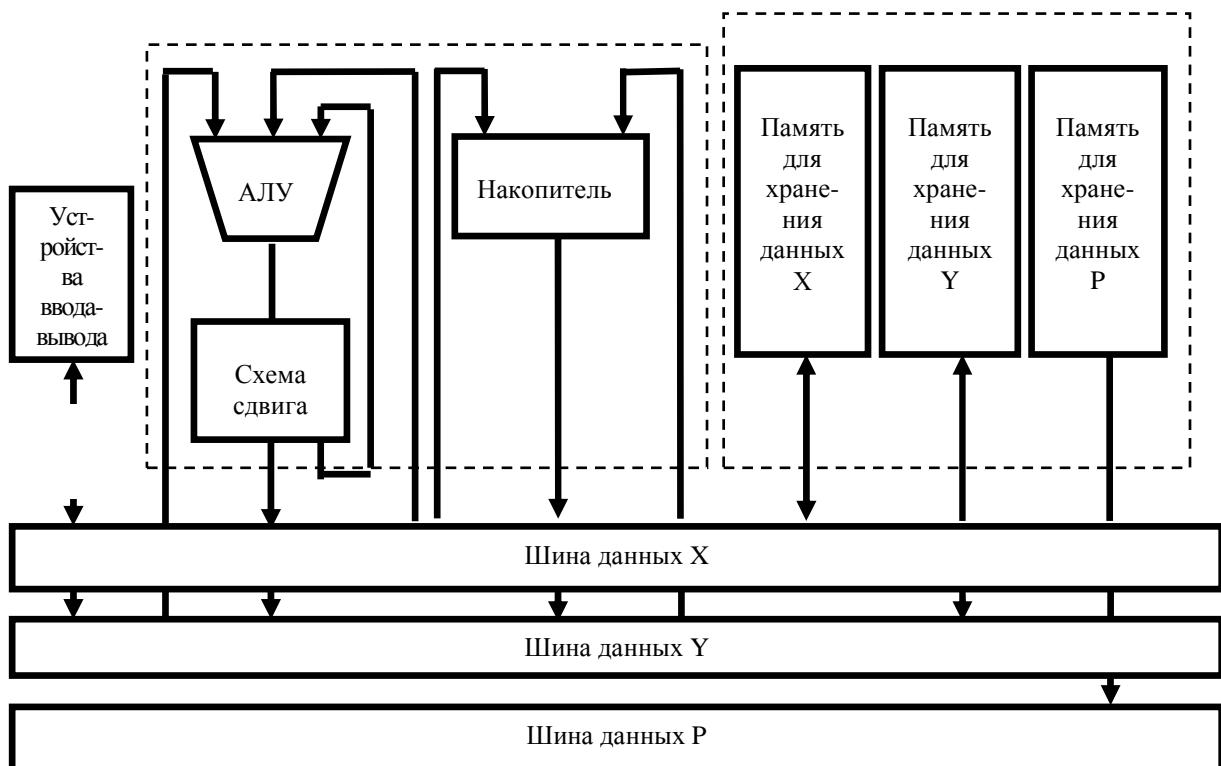


Рис. 3.17. Стандартная универсальная аппаратная архитектура обработки сигналов

Большинство алгоритмов ЦОС (фильтрация, корреляция, преобразование Фурье) включают повторяющиеся арифметические операции, такие как умножение, сложение, обращение к памяти и интенсивная передача данных через центральный процессор. Архитектура стандартных микропроцессоров не предназначена для такого типа деятельности. При разработке аппаратуры ЦОС важно оптимизировать под операции ЦОС и аппаратную архитектуру, и систему команд. В процессорах ЦОС для этого широко используется концепция параллелизма.

В частности, применяются следующие средства [2]:

- гарвардская архитектура;
- конвейерная обработка;
- быстрые специализированные аппаратные умножители-накопители;
- специальные команды, предназначенные для ЦОС;
- копирование;
- встроенная память/кэш;
- расширенный параллелизм – векторная архитектура (SIMD), архитектура с командными словами сверхбольшой длины (VLIW) и статическая суперскалярная обработка.

### 3.4.2. Реализация КИХ-фильтров

В программе 3.1 представлен фрагмент реализации на языке С общего КИХ-фильтра [2]. При фильтрации в реальном времени предполагается, что данные и коэффициенты хранятся в памяти, как показано на рис. 3.18, *a*.

<i>a)</i>		<i>б)</i>	
Коэффици-енты	Данные	Коэффици-енты	Данные
$b(0)$	$s(n)$	$b(N - 1)$	$s(n)$
$b(1)$	$x(n - 1)$	$b(N - 2)$	$x(n - 1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$b(N - 2)$	$x[n - (N - 2)]$	$b(1)$	$x[n - (N - 2)]$
$b(N - 1)$	$x[n - (N - 1)]$	$b(0)$	$x[n - (N - 1)]$

Рис. 3.18. Карты памяти коэффициентов и данных:  
*а)* традиционные; *б)* альтернативные

*Программа 3.1. Псевдокод С для выполнения КИХ-фильтрации.*

$mn1=N-1;$

$yn=0;$

```

for (k=0; k,mn1; ++k) {/* данные смещаются, чтобы освободить место
для новой выборки */
    x[mn1-k]=x[mn1-k-1];
    x[0]=xn;
}
for (k=0; k,N; ++k) {
    yn=yn+b[k]*x[k];      /* данные фильтруются и вычисляется
выходная выборка*/
}
return (yn);                /* выходная выборка фильтра */

```

Предположим, что данные на трехкоэффициентный КИХ-фильтр поступают с АЦП. Первое, что нужно сделать – выделить два набора соседствующих ячеек памяти (в ОЗУ), один для хранения входных данных  $(s(n), s(n-1), s(n-2))$ , а другой для хранения коэффициентов фильтра,  $(b(0), b(1), b(2))$ , как показано ниже:

ОЗУ данных	Память коэффициентов
0	$b(0)$
0	$b(1)$
0	$b(2)$

Во время инициализации значения ячеек ОЗУ, в которых планируется хранить выборки данных, обнуляются, поскольку фильтр всегда начинает работу, не имея данных. Затем выполняются следующие операции [2]:

1. *Первый момент выборки.* Считывается выборка данных с АЦП, ОЗУ данных смещается на одну позицию (чтобы освободить место для новых данных), сохраняется новая входная выборка, вычисляется выходная выборка, а затем выходная выборка посыпается ЦАП:

ОЗУ данных	Память коэффициентов	
$\rightarrow s(1)$	$b(0)$	$y(1) = b(0)s(1) + b(1)s(0) + b(2)s(-1)$
0	$b(1)$	
0	$b(2)$	

*2. Второй момент выборки.* Повторяются описанные выше действия, и новая выходная выборка посыпается ЦАП:

ОЗУ данных	Память коэффициентов	
$\rightarrow s(2)$	$b(0)$	$y(2) = b(0)s(2) + b(1)s(1) + b(2)s(0)$
$s(1)$	$b(1)$	
0	$b(2)$	

*3. Третий момент выборки.* Повторяются описанные выше действия, и новая выходная выборка посыпается ЦАП:

ОЗУ данных	Память коэффициентов	
$\rightarrow s(3)$	$b(0)$	$y(3) = b(0)s(3) + b(1)s(2) + b(2)s(1)$
$s(2)$	$b(1)$	
$s(1)$	$b(2)$	

*4. Четвертый момент выборки.* Повторяются описанные выше действия, и новая выходная выборка посыпается ЦАП:

ОЗУ данных	Память коэффициентов	
$\rightarrow s(4)$	$b(0)$	$y(4) = b(0)s(4) + b(1)s(3) + b(2)s(2)$
$s(3)$	$b(1)$	
$s(2)$	$b(2)$	

Старейшая входная выборка удалена из памяти.

*5. Момент выборки n.* Повторяются описанные выше действия, и новая выходная выборка посыпается ЦАП:

ОЗУ данных	Память коэффициентов	
$\rightarrow s(n)$	$b(0)$	$y(n) = b(0)s(n) + b(1)s(n-1) + b(2)s(n-2)$
$s(n-1)$	$b(1)$	
$s(n-2)$	$b(2)$	

Альтернативным подходом к реализации КИХ-фильтров на процессорах ЦОС является использование круговых буферов. Очевидно, что в КИХ-фильтрации содержимое памяти коэффициентов статично, а вот память данных меняется с поступлением каждой новой входной выборки. С точки

зрения эффективности выгоднее подавать последующие выборки данных в раздвижное окно, удаляя старейшие выборки из памяти. Для обработки изменений в блоке входных данных, используемых при КИХ-фильтрации, можно применять круговой буфер, при этом данные не придется сдвигать, как в линейных буферах [2].

### 3.4.3. Расчет БПФ

На вход бабочки с прореживанием по времени в общем случае подаются два значения  $A$  и  $B$ , по которым вычисляется пара выходов:

$$\begin{aligned} A' &= A + BW_N^k, \\ B' &= A - BW_N^k. \end{aligned}$$

Входные и выходные выборки данных, а также настроочные параметры являются комплексными и в общем виде их можно выразить следующим образом:

$$\begin{aligned} A &= A_{re} + jA_{im}, \\ B &= B_{re} + jB_{im}, \\ W_N^k &= e^{-2\pi jk/N} = \cos(2\pi k / N) - j \sin(2\pi k / N). \end{aligned}$$

где  $re$  – индекс, который обозначает действительную часть величины,  $im$  – индекс, который обозначает мнимую часть величины.

Операция «бабочка», как известно, включает действия с комплексной арифметикой, но на практике она часто выполняется с использованием действительной арифметики. Чтобы выразить эту операцию в форме, подходящей для действительной арифметики, произведение  $B$  и  $W$  имеет следующий вид:

$$BW_N^k = B_{re} \cos(X) + B_{re} \sin(X) + j[B_{im} \cos(X) - B_{re} \sin(X)],$$

где  $X = 2\pi k / N$ .

Используя подстановку, получаем:

$$\begin{aligned} A' &= A_{re} + [B_{re} \cos(X) + B_{im} \sin(X)] + j\{A_{im} + [B_{im} \cos(X) - B_{re} \sin(X)]\}, \\ B' &= B_{re} - [B_{re} \cos(X) + B_{im} \sin(X)] + j\{A_{im} - [B_{im} \cos(X) - B_{re} \sin(X)]\}. \end{aligned}$$

Теперь выходы «бабочки»  $A'$  и  $B'$  представлены в нужной форме. Следовательно, для данной пары комплексных точек  $A$  и  $B$  в можно вычислить выход «бабочки», используя только действительную арифметику.

Вычисление синуса и косинуса в формуле является трудоемкой операцией. В БПФ реального времени эффективнее заранее вычислить действительную и мнимую части настроичного параметра и сохранить данные значения в таблице соответствий. Предварительный расчет значений настроичного параметра иллюстрируется в программе 3.2.

*Программа 3.2. Псевдокод С для предварительного расчета значений настроичного коэффициента [2].*

```
pi=6.28315307179586/N; for(k=0; k<N/2; ++k){  
    X=k*pi;  
    w.real[k]=cos[X];  
    w.imag[k]=sin[X];  
}
```

В программе 3.3 приводится псевдокод С для двоичной «бабочки», значения настроичного параметра которой вычислены заранее и занесены в таблицу соответствий.

*Программа 3.3. Псевдокод С для расчета «бабочки» [2].*

```
t.real=br*w.real[k]+bi*w.imag;  
t.imag=bi*w.real[k]-br*w.imag[k];  
b.real[j]=a.real-t.real ;  
b.imag[ j ]=a.imag-t.imag ;  
a.real[j]=a.real+t.real;  
a.imag[j]=a.imag+t.imag;
```

Таким образом, алгоритмы ЦОС подразумевают трудоемкие арифметические операции, в частности, умножение и сложение с интенсивным потоком данных через центральный процессор. Эффективное выполнение подобных алгоритмов в реальном времени требует аппаратной архитектуры и набора команд, радикально отличающихся от архитектуры и команд стандартных микропроцессоров. В цифровых процессорах сигналов для этого используются гарвардская архитектура, конвейерная обработка и специализированное аппаратное обеспечение, например, быстрые аппаратные умножители-накопители и схемы сдвига, скоростная внешняя память, также в процессоры внедряется множество команд, ориентированных на ЦОС.

Чтобы удовлетворять требованиям таких многоканальных, вычисительно-интенсивных приложений, как современные модемы удаленного доступа к серверу, системы мобильной связи третьего поколения, обработка

мультимедийной информации, вводятся новые архитектуры. В частности, это архитектура VLIW (very long instruction word – с командными словами сверхбольшой длины) и статическая суперскалярная архитектура, которые используются в последнем поколении процессоров ЦОС. В обеих архитектурах применяются множественные тракты обработки данных и арифметические устройства, также для повышения эффективности используется параллелизм на уровне команд.

### **3.4.4. Структура и описание процессора TMS320VC5510/5510A**

Цифровые сигнальные процессоры с фиксированной точкой TMS320VC5510/5510A базируются на ядре TMS320C55x. Данная архитектура обладает высокой производительностью при низком энергопотреблении благодаря повышению параллелизма и снижению потребляемой мощности. ЦПУ обладает внутренней шинной структурой, состоящей из программной шины, трёх шин чтения данных, двух шин записи данных и вспомогательных шин для периферии и контроллера прямого доступа к памяти (DMA). Это позволяет выполнять вплоть до трех операций чтения данных и двух операций записи данных за один цикл. Параллельно этому, контроллер DMA может осуществить до двух операций перемещения данных без задействования ЦПУ.

Ядро ЦСП C55x обладает двумя модулями умножения-накопления (MAC), каждый из которых способен выполнять операции типа «умножение 17-бит  $\times$  17-бит» за один цикл. Центральное 40-битное арифметико-логическое устройство (АЛУ) сопровождается вспомогательным 16-битным АЛУ. Сценарий совместного использования двух АЛУ определяется набором инструкций, обеспечивая оптимальную параллельную работу и снижение энергопотребления. Распределение ресурсов возложено на адресное устройство (АУ) и устройство данных (УД) ядра ЦСП C55x.

Семейством цифровых сигнальных процессоров C55x поддерживаются инструкции с переменным числом байт, что позволяет увеличить плотность кода. Модуль инструкций (МИ) осуществляет 32-битную выборку инструкций из внутренней либо внешней памяти и определяет очередь инструкций для программного модуля (ПМ). В свою очередь, ПМ декодирует инструкции, определяет задачи для АУ и УД и управляет защищенным конвейером. Для предотвращения переполнения конвейера при выполнении условных переходов используется предсказание переходов. Процессоры 5510/5510A также имеют встроенный кэш инструкций размером 24 Кбайт для снижения числа операций доступа к внешней памяти, повышения производительности и снижения энергопотребления.

Набор периферийных устройств процессоров 5510/5510A включает в себя интерфейс внешней памяти (EMIF), обеспечивающий непосредственное подключение различных типов асинхронной памяти, таких, как EEPROM и SRAM, а также высокоскоростных запоминающих устройств высокой плотности, таких, как синхронные DRAM и SRAM с пакетной выборкой. Три полнодуплексных многоканальных буферизованных последовательных порта (McBSP) обеспечивают непосредственное подключение большого ряда устройств со стандартными последовательными интерфейсами и многоканальный обмен (до 128 каналов с индивидуальным запретом). Порт управляющего контроллера (EHPI) представляет собой 16-битный параллельный интерфейс, обеспечивающий внешнему управляющему процессору доступ к встроенной памяти процессоров 5510/5510A. Порт EHPI может быть сконфигурирован как в мультиплексный, так и в немультиплексный режим, что позволяет использовать его совместно с самыми различными управляющими процессорами. Контроллер прямого доступа к памяти (DMA) обеспечивает перемещение данных по шести независимым каналам без вмешательства ЦПУ, его суммарная пропускная способность составляет до двух 16-битных слов за цикл. Кроме этого, набор периферийных устройств включает в себя 2 таймера общего назначения, восемь выводов общего назначения (GPIO) и генератор с ФАПЧ и цифровым управлением (DPLL).

Процессоры 5510/5510A поддерживаются программным обеспечением eXpressDSP™, которое состоит из интегрированной среды разработки (IDE) Code Composer Studio™, PTOC DSP/BIOS™ и документации TMS320™ DSP Algorithm Standard, кроме этого, существует большое количество разработок сторонних производителей. IDE Code Composer Studio состоит из компилятора языка C, линкера Visual Linker, симулятора, поддержки обмена данными в реальном режиме времени Real-Time Data Exchange (RTDX™), драйверов эмулятора XDS510™ и библиотек Chip Support Libraries (CSL). DSP/BIOS представляет собой расширяемое модульное программное обеспечение, бесплатно доступное пользователям цифровых сигнальных процессоров фирмы Texas Instruments и состоящее в основном из планировщика задач и средств поддержки систем реального времени с весьма экономичным расходованием памяти и других ресурсов процессора. TMS320 DSP Algorithm Standard – это набор спецификаций, выполнение которых обеспечивает совместимость кода, созданного различными разработчиками, что значительно облегчает его дальнейшую интеграцию в проекты.

Ядро процессоров TMS320C55x базируется на открытой архитектуре с добавлением специфических модулей, позволяющих значительно ускорить обработку некоторых алгоритмов. Данные модули обеспечивают процессорам

5510/5510A оптимальное соотношение производительности, присущей процессорам с фиксированной точкой, низкого энергопотребления и цены, что является уникальным сочетанием на рынке видеопроцессоров. Дополнительные модули обеспечивают процессорам 5510/5510A высокую производительность видеокодека, в то же время оставляя незадействованными более половины ресурсов ядра, что позволяет реализовывать параллельно такие функции, как преобразование цветовых схем, организация пользовательского интерфейса, функции безопасности, стек TCP/IP, распознавание голоса, преобразование текста в голос и т. д. В результате, цифровой сигнальный процессор 5510/5510A способен в одиночку решить практически все задачи, стоящие перед мобильными видеоустройствами [14].

Отличительные особенности:

1. Высокопроизводительное ядро для операций с фиксированной точкой с низким энергопотреблением TMS320C55™:
  - a) время цикла 6.25/5 нс;
  - b) тактовая частота 160/200 МГц;
  - c) выполнение одной или двух инструкций за такт;
  - d) два умножителя (производительность до 400 миллионов умножений с накоплением в секунду (MMACS));
  - e) два арифметико-логических устройства (АЛУ);
  - f) внутренняя программная шина;
  - g) три внутренних шины чтения данных/операндов;
  - h) две внутренних шины записи данных/операндов.
2. Кэш инструкций (24 КБайта).
3. Встроенное ОЗУ 160К×16 бит, состоящее из:
  - i) 8 блоков по 4К×16 бит двухпортовой ОЗУ (DARAM) (всего 64 КБайт);
  - j) 32 блоков по 4К×16 бит однопортовой ОЗУ (SARAM) (всего 256 КБайт).
4. Встроенное ПЗУ 16К×16 бит (32 КБайт).
5. Общий объем адресуемой памяти 8М×16 бит.
6. 32-битный интерфейс внешней памяти (EMIF) с возможностью подключения памяти типа:
  - k) асинхронное статическое ОЗУ (SRAM);
  - l) асинхронное EEPROM;
  - m) синхронное динамическое ОЗУ (SDRAM);
  - n) синхронное статическое ОЗУ с пакетной выборкой (SBSRAM).
7. Программный контроль энергопотребления шести функциональных блоков из внутренних устройств.

8. Встроенные периферийные устройства:
  - о) два 20-битных таймера;
  - р) шестиканальный контроллер прямого доступа к памяти (DMA);
  - q) три многоканальных буферизованных последовательных порта ( McBSP );
  - r) 16-битный параллельный порт управляющего контроллера (EHPI);
  - s) программируемый тактовый генератор с ФАПЧ;
  - t) портов ввода-вывода общего назначения (GPIO) и выход общего назначения (XF).
9. Встроенный эмулятор.
10. Поддержка периферийного сканирования по стандарту JTAG.
11. Питание ядра 1,6 В.
12. Питание портов ввода-вывода 3,3 В.

Память процессора состоит из адресов регистров и адресов данных программы, содержащей константы и тело программы. Обращение к памяти процессора может осуществляться по адресам. Аппаратно доступ к блокам, описанным в таблице 1, может осуществляться параллельно. Т. е. два доступа за один цикл (два чтения, две записи или чтение и запись). DRAM – *Dual-Access RAM (Память с двойным доступом)* [15].

### **3.5. Вопросы и задания для самопроверки**

1. Представьте разностное уравнение, которое описывает ЦФ.
2. Для каких целей используется Z -преобразование и каким образом оно определяется?
3. Приведите пример прямого и обратного Z -преобразования сигнала.
4. По какому правилу определяется порядок РЦФ?
5. Приведите примеры разностных уравнений РЦФ (НЦФ).
6. Может ли РЦФ иметь конечную импульсную характеристику? Поясните ответ.
7. Каким образом можно определить импульсную характеристику ЦФ?
8. Что понимают под передаточной функцией ЦФ?
9. Известно разностное уравнение ЦФ  $y(n) = 0,5s(n) - 0,3s(n-1) - 0,7y(n-1) + 0,55y(n-2)$ . Запишите передаточную функцию.
10. Приведите классификацию структур ЦФ.
11. Для ЦФ из задания 9 представьте прямую структуру фильтра.
12. В чем преимущество канонической структуры ЦФ по сравнению с прямой?

13. Представьте обобщенную схему канонической структуры второго типа ЦФ.

14. На основе каких правил следует выбирать тип структуры ЦФ при его реализации?

15. Запишите выражения для частотных характеристик КИХ и БИХ фильтров.

16. Представьте идеальные АЧХ для НЧ, ВЧ, полосового и режекторного ЦФ.

17. Сформулируйте основные требования, предъявляемые к ЦФ.

18. Назовите отличительные черты процессоров ЦОС?

19. Сформулируйте основные особенности реализации КИХ фильтров на ЦСП.

20. Сформулируйте основные процедуры реализации БПФ на ЦСП.

## Лабораторная работа №4

### Реализация цифровых фильтров на TMS 320VC5510

Перед началом работы необходимо в программе *Code Composer Studio 3.1* указать какой вид процессора будет использоваться и подключить его библиотеки команд. Для этого, после установки, следует запустить *Code Composer Studio Setup (Setup CCStudio v 3.1)*. На рис 3.19 показан внешний вид окна меню.

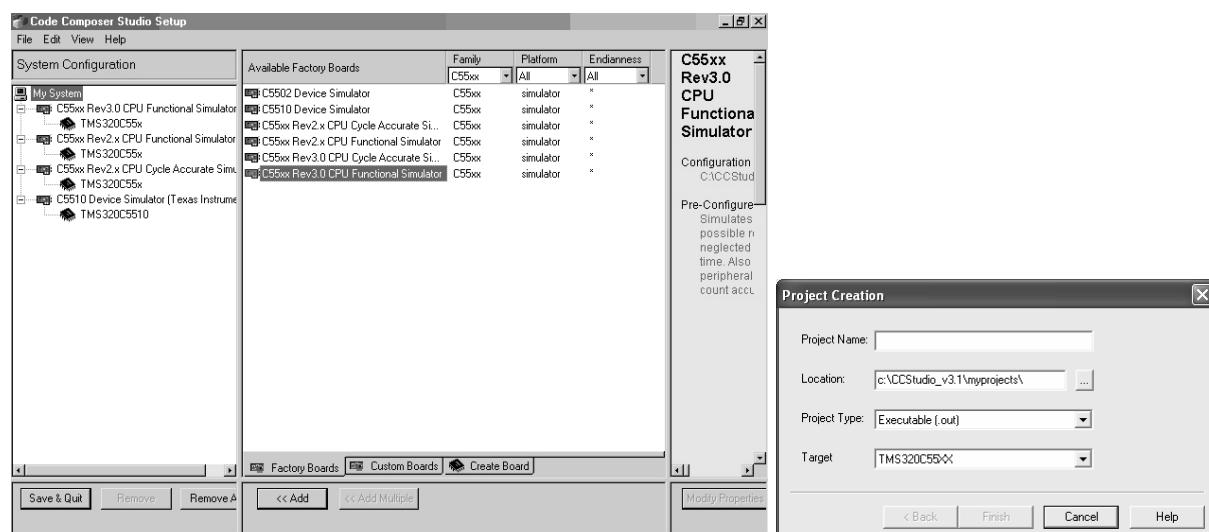


Рис. 3.19. Внешний вид окна меню  
*Code Composer Studio Setup*

Рис. 3.20. Диалоговое окно  
создания нового проекта

Для лабораторной работы необходимо добавить *C5510 Device Simulator*, выделив его и нажав кнопку «Add». Нажимаем «Save & Quit» и на предложение «*Start Code Composer Studio on exit?*» нажимаем «Да». Затем в предложенном меню подтверждается вид микропроцессорного блока двойным щелчком, после чего отображается интерфейс симулятора.

При написания программы используются типы файлов с расширениями: \*.lib – библиотека, которая обеспечивает поддержку во время выполнения целевому чипу цифрового обработчика сигналов; \*.c – файл содержит исходный текст, который обеспечивает главные функциональные возможности проекта; \*.h – файл объявляет буферную С-структуру и определяет любые необходимые константы; \*.pj – файл содержит весь построенный проект и варианты конфигурации; \*.asm – файл содержит инструкции ассемблера; \*.cmd – файл наносит на карту секции в памяти.

Для создания нового проекта следует выбрать из вкладки *Project* пункт *New*, на экране отобразится меню (рис. 3.20). В данном меню пункт *Project type* необходимо выбрать – *executable(.out)* и указать имя проекта.

На рис. 3.21 представлен новый вид менеджера проекта. Следующий шаг состоит в подключении библиотеки – (...CCStudio\_v3.1\C5500\cgtools\lib\rts55.lib), для чего следует нажать на одной из папок правой кнопкой мыши и выбрать вкладку *Add files to project*. Далее создается файл, в котором непосредственно размещается код. Для этого выбирается на панели меню *File/New/Source File*. Для того чтобы он стал главным файлом проекта, его необходимо сохранить под расширением .c, а затем следует его добавить тем же способом что и библиотеку, но из той директории, где находится проект. В папке *Source* появится добавленный файл. В последующем каждый файл добавленный будет появляться в своей папке согласно его назначению автоматически.

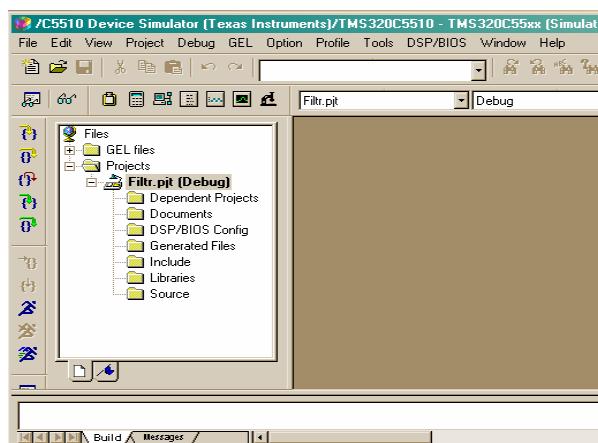


Рис. 3.21. Меню менеджера проекта

Структура написания программ идентична работе в среде программирования С. Поэтому далее синтаксис программы уточняться не будет. Для добавления файлов описаний (.h) необходимо сослаться на них в коде «#include "Fil.h.h"» и, нажав правой кнопкой на название проекта в менеджере выбрать «Scan All File Dependencies», откомпилировать проект и этот файл будет добавлен автоматически. На рис. 3.22 указана кнопка компиляции.

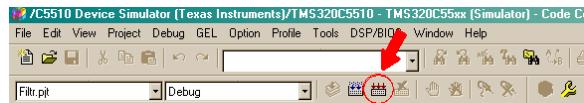


Рис. 3.22. Запуск компилятора

Для того чтобы эмулятор мог работать с переменными внутри программы или применять изменения, необходимо после компиляции в память записать код программы. Для этого необходимо добавить следующий файл из папки *Debug* с помощью пункта *Load Program* из меню *File*. Название файла идентично названию проекта только с расширением *.out*. Особенностью эмулятора является то, что в нем можно подключать внешние файлы данных.

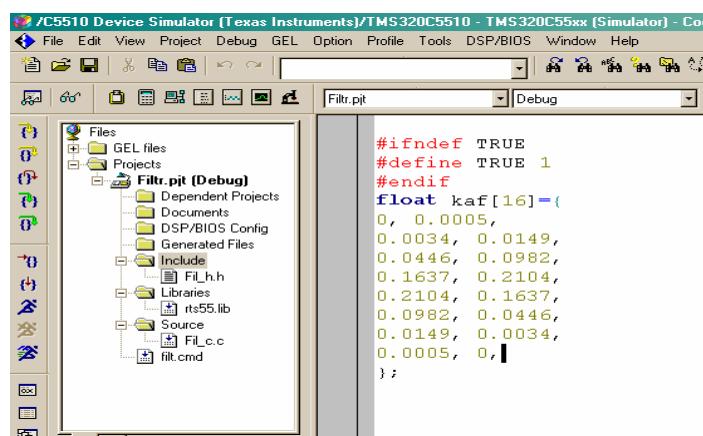


Рис. 3.23. Окно листинга файла *Fil.h.h*

Принцип подключения заключается в том, что указывается часть кода, в момент выполнения которого должно осуществляться чтение из файла в определенный массив. Момент чтения может задаваться и другими событиями, не только переходом указателя выполнения программы через определенный ее участок. Для того чтобы определить место в коде программы, в котором в массив должны записаться данные из внешнего файла, необходимо поставить там

точку прерывания, которую можно увидеть на рис. 3.24. В данном примере точки прерывания поставлены именно напротив той строки, где должно осуществляться присваивание внешней переменной внутреннему массиву. Для подключения файла необходимо во вкладке *File* выбрать пункт *File I/O*. На экране появится меню, представленное на рис. 3.25.

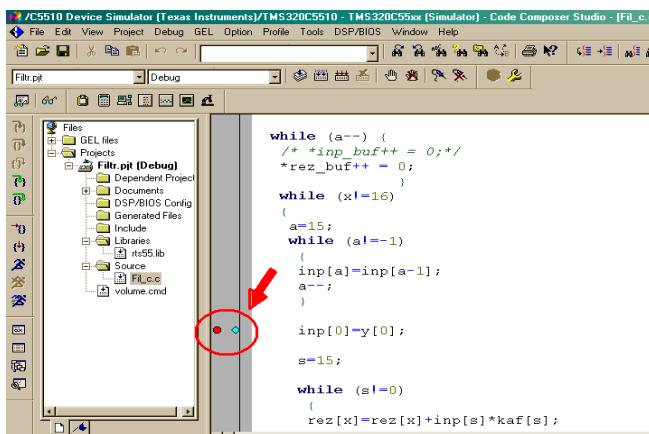


Рис. 3.24. Точки прерывания

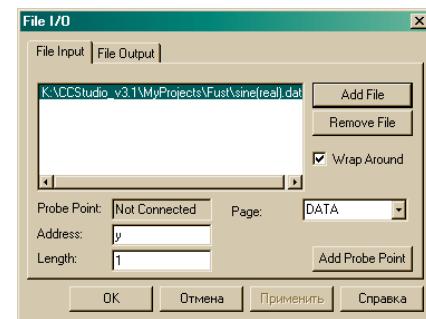


Рис. 3.25. Меню файлов ввода-вывода

Далее нажать *Add File* и выбрать файл данных, например *sine.dat*, в котором указаны числа в шестнадцатеричной форме. *Address* – указатель массива, в который происходит чтение, *Length* – количество читаемых в массив единиц данных (отсчетов). В данном случае чтение происходит по одному отсчету. Затем нажимаем *Add Probe Point* для установки связи между точкой прерывания и файлом из которого происходит чтение. На экране появится меню, указанное на рис. 3.26.

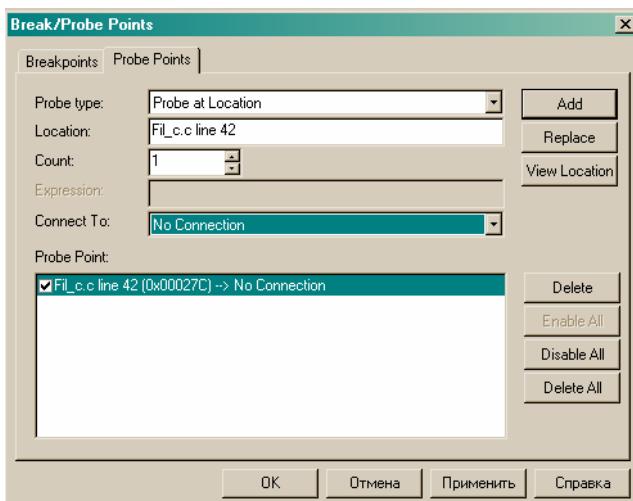


Рис. 3.26. Меню указателя

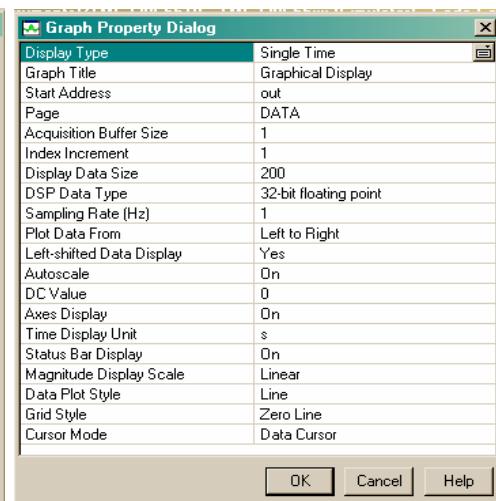


Рис. 3.27. Меню настроек графика, связывающего файл с событием

Выделяем пункт с линией в программе в окне *Probe Point* и выбираем из меню *Connection To* название файла данных, нажимаем *Replace* и подтверждаем.

Рассмотрим возможность визуализации в программе. Для этой цели в эмуляторе предусмотрен осциллограф.

Для подключения из вкладке *View* выбираем *Graph*, затем *Time/Frequency*. На экране появляются настройки изображения, представленные на рис. 3.27, для которого необходимо указать *Start Address* – указатель массива, из которого будут поступать данные в график при переходе точки прерывания, количество переменных, читаемых одновременно (*Acquisition Buffer Size*) и *DSP Data Type* – тип отображаемых данных (необходимо указать тип *float* – с плавающей запятой). На экране появляется график, представленный на рис. 3.28.

Следует не забывать обновлять код в памяти, добавляя следующий файл из папки *Debug* с помощью пункта *Load Program* из меню *File*. Название файла идентично названию проекта, только с расширением *.out*.

Для запуска программы нажимаем на фигурку бегущего человечка «*Run*» (для того чтобы программа проработала до точки прерывания) и «*Animate*» (для многократного повторения (зацикливания)).

Для начала работы с DSK необходимо обеспечить совместимость платы процессора с компьютером с использованием порта USB. При обнаружении компьютером устройства USB операционная оболочка потребует установки драйверов. Для корректной установки необходимо выбрать ручной режим установки и указать путь, по которому находится драйвер: ....\CCStudio\_v3.1\specdig\xds510usb\.

В программе *CCStudio\_v3.1* для совмещения работы симулятора и реального процессорного блока предусмотрено подключение необходимых модулей для работы процессором. Подключение обеспечивается в программе *Setup CCStudio\_v3.1*. Окно подключения модуля и сам модуль представлены на рис. 3.29.

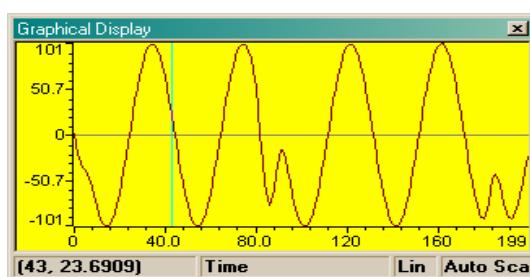


Рис. 3.28. Вид графического окна

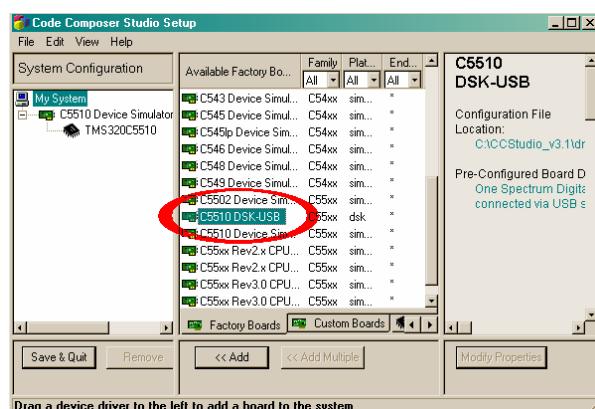


Рис. 3.29. Диалоговое окно модуля процессорного блока

Подсвечиваем его мышью и нажимаем кнопку *Add*, теперь применяем установки кнопкой *Save & Quit* и запускаем *CCStudio\_v 3.1*.

На экране монитора появится меню менеджера параллельной загрузки, представленного на рис. 3.30.

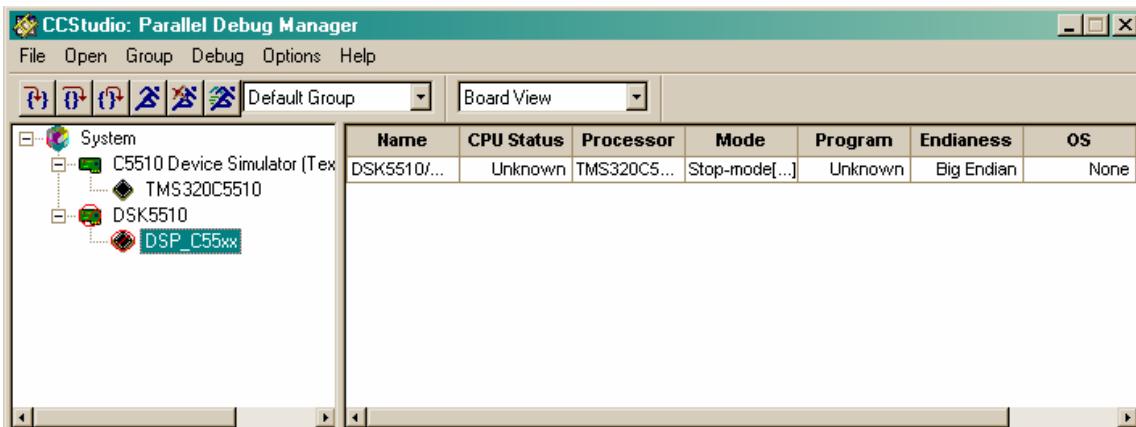


Рис. 3.30. Менеджер параллельной загрузки

На рис. 3.30 видно, что плата процессора не подключена. Однако запускаем окно редактора кода двойным щелчком по иконке процессора *DSP\_C55xx* и на экране появляется окно редактора с предупреждением о том, что плата процессора не подключена. Для подключения процессора необходимо выбрать в главной панели меню *Debug* далее *Connect* или нажать сочетание клавиш Alt + C. Окно редактора изменится (рис. 3.31).

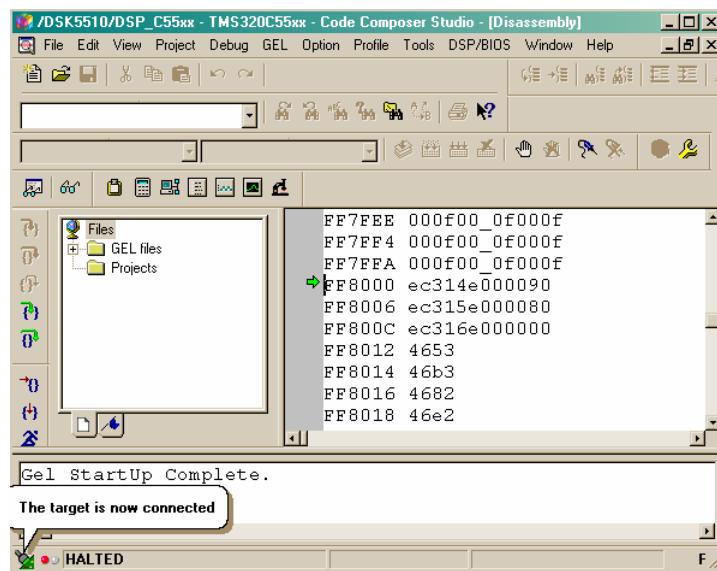


Рис. 3.31. Окно редактора

Можно в окне менеджера параллельной загрузки правой клавишей мыши нажать на иконке *DSP\_C55xx* и выбрать в меню пункт *Connect Device*.

## **Задания**

1. Внимательно изучите порядок работы с Code Composer Studio 3.1.
2. Программно реализуйте цифровой фильтр согласно заданному варианту.
3. Выполните тестирование программной реализации с использованием тестовых сигналов и звукового файла.
4. Внимательно изучите особенности и порядок подключения процессорного блока DSK TMS320VC5510. Подключение DSK допускается только с разрешения преподавателя.
5. Выполните тестирование программной реализации с использованием DSK TMS320VC5510 для обработки звукового файла.

## **Содержание отчета**

1. Титульный лист: название лабораторной работы, Ф.И.О., группа.
2. Цель работы.
3. Алгоритм фильтрации сигналов.
4. Программная реализация цифровой фильтрации и ее особенности.
5. Результаты тестовых испытаний.

## **МОДУЛЬ 4. СПЕКТРАЛЬНЫЙ АНАЛИЗ СИГНАЛОВ**

*Цель модуля – изучение студентами методов и алгоритмов спектрального анализа и оценки параметров сигнала.*

*В результате изучения модуля студенты должны знать:*

- методы спектрального анализа и оценки параметров сигнала;
- основные характеристики весовых функций, применяемых при спектральном анализе.

*Студенты должны уметь:*

- использовать алгоритмы спектрального анализа для обработки сигналов;
- а также иметь представление об областях использования методов спектрального анализа сигналов.

### **4.1. Анализ спектра на основе ДПФ**

#### **4.1.1. Параметры анализаторов спектра**

Спектральный анализ заключается в разложении сигнала на его частотные или спектральные составляющие и оценке или измерении их спектральных характеристик – амплитуды, фазы, мощности, спектральной плотности мощности и др. [4].

К задачам, решаемым методами спектрального анализа, относят [4]: обнаружение, разрешение и оценивание параметров сигналов, сжатие данных, выделение информативных признаков, идентификация объектов, распознавание образов и др. Для случайных сигналов с помощью спектрального анализа решается общая задача выявления скрытых периодичностей и статистических (корреляционных) связей. Спектральный анализ детерминированных периодических сигналов и сигналов конечной длительности называют также гармоническим анализом.

К методам спектрального анализа относят [2,4]: методы полосового анализа на основе цифровой фильтрации, методы на основе ДПФ, параметрические методы (на основе параметрических моделей случайных процессов). Кроме этого рассматривают методы текущего, скользящего и скачущего спектрального анализа, последовательного и параллельного, одноканального и многоканального, в реальном и нереальном режимах обработки.

К параметрам анализаторов спектра относятся [4]:

- число каналов анализа;
- время наблюдения или анализа (ширина окна)  $T_a = NT_D$  и соответствующее ему число отсчетов или длина  $N$  обрабатываемой реализации сигнала;

- полоса анализа  $\pm f_D/2$ , не превышающая для дискретных сигналов основной полосы спектра  $\pm f_D/2$ ;
- разрешение по частоте, обратно пропорциональное времени анализа:  $\Delta f_p \sim 1/T_a$  и соответствующее разности частот двух соседних разрешаемых (разделяемых) частотных составляющих сигнала.

Особенности спектрального анализа на основе ДПФ связаны с оценкой спектра сигнала по его реализациям конечной длины, т. е. на конечном интервале наблюдения. При этом полагается, что за пределами этого интервала сигнал равен нулю или является периодическим продолжением считанной реализации. Широкое применение анализаторов спектра на основе ДПФ обусловлено наличием алгоритмов БПФ.

#### **4.1.2. Базовая структура анализатора спектра на основе ДПФ и БПФ**

В основе анализаторов спектра, использующих ДПФ, лежит базовая структура, приведенная на рис. 4.1 [4], которая реализует операции взвешивания и вычисления ДПФ. Выходом схемы является вектор ДПФ входной, в общем случае не ограниченной по длине, последовательности  $s(n)$ , усеченной весовой функцией  $w(n)$  конечной длины  $N$ :

$$\widetilde{X}(j\omega_k) = \sum_{n=0}^{N-1} s(n)w(n)e^{-j\omega_k nT_D} = \sum_{n=0}^{N-1} \tilde{s}(n)e^{-j(2\pi/N)kn} = \text{ДПФ}_N[\tilde{s}(n)],$$

где  $k = 0, 1, \dots, N-1$ ;

$n = 0, 1, \dots, N-1$ ;

$s(n)$  – входной сигнал;

$w(n)$  – весовая функция;

$\tilde{s}(n) = s(n)w(n)$  – преобразованная входная последовательность ДПФ;

$\omega_k = k\omega_D / N$  или  $f_k = kf_D / N$  – частоты анализа, называемые также бинами ДПФ [2]: 1 бин равен шагу дискретизации сигнала в частотной области  $f_D / N$ .

Анализатор имеет  $N$  разнесенных по частоте на 1 бин ( $f_D / N$ ) каналов анализа с центральными частотами  $\omega_k (f_k)$ , при этом значения  $k = 0, 1, \dots, N-1$  соответствуют номеру канала, номеру бина или номеру частотной выборки ДПФ  $\widetilde{X}(j\omega_k) = \widetilde{X}(jk)$ . Весовая функция представляет окно, через которое наблюдается входной сигнал, длиной ее определяется время анализа  $T_a = NT_D$  или время наблюдения сигнала.

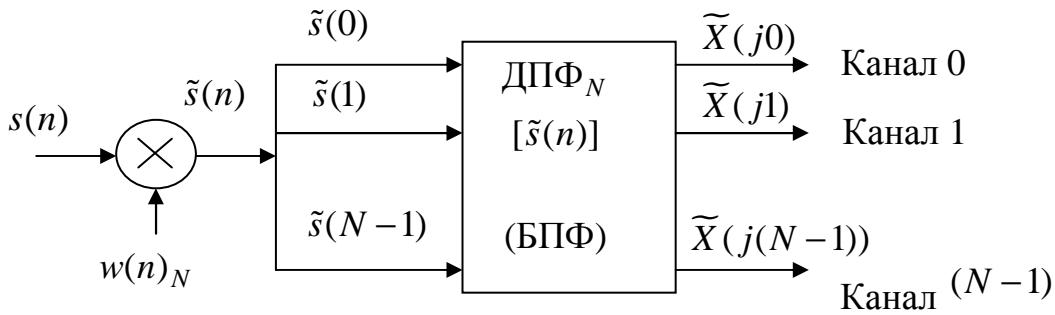


Рис. 4.1. Структурная схема анализатора спектра на основе ДПФ, представленная его базовыми элементами

Умножению или взвешиванию во временной области соответствует свертка в частотной, поэтому вычисляемое ДПФ фактически является дискретизированной сверткой истинного спектра анализируемого сигнала  $X(j\omega)$  с частотной характеристикой (спектром) весовой функции  $W(j\omega)$ :

$$\tilde{X}(j\omega_k) = X(j\omega) * W(j\omega) \mid \omega = \omega_k,$$

где  $*$  – символ свертки.

Вычисляемое ДПФ содержит систематическую (методическую) погрешность анализа, которая является следствием ограничения сигнала по длительности, искажающего результаты спектрального анализа.

Применение специальных весовых функций или окон позволяет сгладить или ослабить вызываемое времененным усечением влияние или эффект разрывов сигнала на краях. Кроме этого, взвешивание позволяет [4]:

- устранять нежелательные эффекты, связанные с явлением Гиббса, вызванного наличием разрывов на границах исследуемых отрезков сигнала; взвешивающее окно плавно сводит на нет функцию вблизи краев анализируемого участка, устранивая разрывы функции;

- добиться значительного снижения вычислительной сложности процесса обработки, так как ДПФ вычисляется не на всей длине сигнала, а на интервалах, на которые разбивается сигнал;

- оконное ДПФ дает возможность получить не просто частотное, а частотно-временное представление сигнала, что делает возможным применения его для анализа нестационарных сигналов с довольно быстрым изменением частотного сигнала.

Однако у оконного ДПФ есть один недостаток, обусловленный принципом неопределенности Гейзенберга, который гласит, что невозможно получить произвольно точное частотно-временное представление

сигнала, то есть нельзя определить для заданного момента времени, какие спектральные компоненты присутствуют в сигнале [4]. Однако можно узнать временные интервалы, в течение которых в сигнале существуют полосы частот. Эта проблема называется проблемой разрешения, и связана она с шириной используемой оконной функции. Узкое окно обеспечивает лучшее временное разрешение, а широкое – лучшее частотное. К тому же для корректного анализа необходимо знать априорную информацию об анализируемом сигнале (в частности, насколько динамичным является сигнал). Если сигнал состоит из далеко отстоящих друг от друга частотных компонент, то можно пожертвовать спектральным разрешением в пользу временного, и наоборот.

Дальнейшая обработка выходных данных ДПФ осуществляется с учетом измеряемых или оцениваемых с помощью ДПФ спектральных характеристик, зависящих от вида анализируемых сигналов.

Для периодических сигналов  $s_p(n)_N$  с периодом  $NT_D$  оценивают амплитуды  $A_m(\omega_i)$  и фазы  $\phi(\omega_i)$  гармоник с частотой  $if_D / N$  или их среднее за период мощности  $[A_m(\omega_i)]^2 / 2$ .

Для детерминированных сигналов конечной длительности  $s(n)_N$  (непериодических) оценивают [4]:

- спектральную плотность  $X(j\omega)$ , определяемую ее модулем  $|X(j\omega)|$  и аргументом  $\phi(\omega)$ , т. е. амплитудным и фазовым спектрами или реальной  $X_{Re}(j\omega)$  и мнимой  $X_{Im}(j\omega)$  частями и вычисляемую на частотах анализа  $\omega = \omega_k$  или бинах ДПФ;

- энергетический спектр или спектральную плотность энергии  $S_x(\omega)$   $(|X(j\omega)|^2)$  размерностью, показывающей распределение энергии сигнала по частоте и также вычисляемую на дискретных частотах  $\omega_k$ .

Для случайных сигналов  $s(n)$  оценивают спектральную плотность мощности  $P_s(\omega)$ , являющуюся статистическим расширением понятия спектральной плотности энергии на случайные сигналы, т. е. сигналы с бесконечной энергией и также вычисляемую на дискретных частотах  $\omega_k$ . Для стохастически связанных случайных сигналов  $s(n)$ ,  $y(n)$  с помощью ДПФ измеряют их взаимную спектральную плотность мощности  $P_{sy}(j\omega)$ .

При реализации конкретных алгоритмов спектрального анализа различных сигналов важное значение имеет правильное масштабирование результатов анализа и учета их размерности.

Так, по вычисленному ДПФ:

$$X(j\omega_k) = \sum_{n=0}^{N-1} s_p(n) e^{-j\omega_k n T_D},$$

вещественного периодического сигнала  $s_p(n)_N$  с периодом  $N T_D$  и частотами гармоник  $i f_D / N$ , совпадающими с бинами ДПФ, амплитуды гармоник  $A_m(\omega_i)$  определяются как:

$$A_m(\omega_i) = (2 / N) |X(j\omega_i)|;$$

фазы  $\phi(\omega_i)$  как:

$$A_m(\omega_i) = \arctg[X_{\text{Im}}(j\omega_i) / X_{\text{Re}}(j\omega_i)];$$

и средние мощности как:

$$P_{cp}(\omega_i) = 2 \left| (1 / N) X(j\omega_i) \right|^2.$$

По ДПФ  $X(j\omega_k)$  детерминированного сигнала конечной длительности  $N T_D$  аналогичным образом находятся амплитуды, фазы и мощности  $k$ -й частотной выборки спектра сигнала.

Другие спектральные характеристики такого сигнала связаны с его ДПФ следующими выражениями [4].

Спектральная плотность энергии на частоте  $\omega_k$ :

$$S_s(k) = \left| T_D X(j\omega_k) \right|^2.$$

Спектральная плотность мощности на частоте  $\omega_k$ :

$$P_s(k) = T_D / N \left| X(j\omega_k) \right|^2.$$

Полная энергия сигнала:

$$S_s = \frac{1}{N T_D} \sum_{k=0}^{N-1} S_x(k).$$

Средняя мощность сигнала:

$$P_s = \frac{1}{NT_D} \sum_{k=0}^{N-1} P_x(k).$$

Пример работы спектроанализатора. На вход спектроанализатора подается сигнал, являющийся суммой синусов с частотами 5 КГц, 10 КГц и 15 КГц (рис. 4.2)

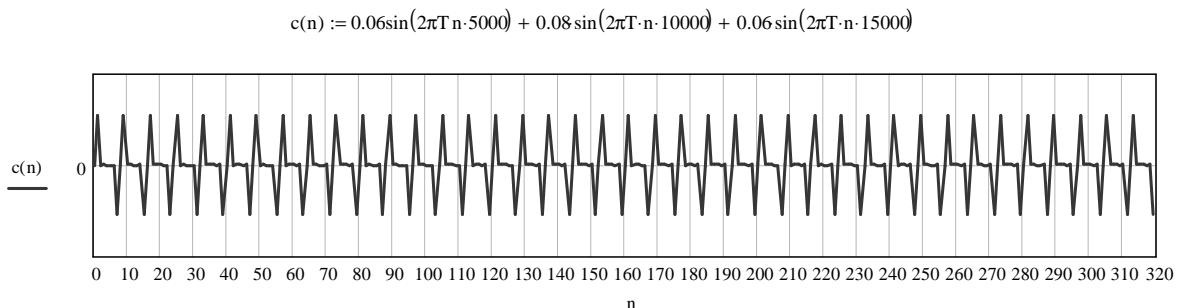


Рис. 4.2. Входной сигнал



Рис. 4.3. Спектральная плотность энергии сигнала на интервале ( $N = 128 - 191$ )

Анализ рис. 4.3 показывает, что энергия сигнала сосредоточена на частотах 5 КГц, 10 КГц и 15 КГц. Боковые составляющие обусловлены влиянием окна Хэмминга ( $T_a = 64$ ).

Так, как выходная последовательность представляет собой спектральную плотность энергии сигнала, то главную роль играет долевое участие соответствующей полосы частот в образовании сигнала. Поэтому на графике значения по оси  $Y$  представлены в нормированном виде.

#### 4.1.3. Частотная характеристика анализатора спектра на основе ДПФ

Существует тесная связь между ДПФ и цифровой фильтрацией. Для ее установления запишем отклик НЦФ с конечной импульсной характеристикой  $h(n)$  длиной  $N$  на анализируемый с помощью ДПФ сигнал  $s(n)$  в момент времени  $nT_D = (N-1)T_D$  (при  $n = N-1$ ) [4]:

$$y(n) = \sum_{m=0}^{N-1} h(m)s(n-m) = \sum_{m=0}^{N-1} s(m)h(n-m).$$

Из сопоставления выше представленных выражений следует, что выход канала ДПФ  $X(j\omega_k)$  совпадает с выходом НФ с импульсной характеристикой, отвечающей условию:

$$h(N-1-m) = w(m)e^{-j\omega_k m T_D},$$

или

$$h(m) = w(N-1-m)e^{-j\omega_k (N-1-m)T_D}.$$

Это означает, что  $k$ -й канал ДПФ эквивалентен полосовому НЦФ с импульсной характеристикой  $h(m)$ , согласованной со взвешенным комплексным гармоническим сигналом  $w_k(m) = w(m)e^{j\omega_k m T_D}$  в соответствии с условием согласования:  $h(n) = w_k^*(N-1-m)$ , ( $*$  – комплексное сопряжение). Такой фильтр имеет частотную характеристику:

$$H(j\omega) = e^{-j\omega(N-1)T_D} W^*[j(\omega - \omega_k)], \quad (3.9)$$

являющуюся комплексно-сопряженной частотной характеристике весовой функции  $W^*(j\omega)$ , смещенной вправо к частоте  $\omega_k > 0$  (центральной частоте канала  $k > 0$ ) или влево к частоте  $\omega_k < 0$  (центральной частоте канала  $k < 0$ ).

Частотную характеристику НФ можно определить также по его отклику  $y(N-1)$  на комплексный гармонический сигнал  $s(n) = e^{j\omega n T_D}$ , относенному к значению этого сигнала при  $n = N-1$  [4]:

$$H(j\omega_k, \omega) = \text{ДПФ}_N \left\{ e^{j\omega n T_D} w(n) \right\} / e^{j\omega(N-1)T_D} = e^{-j\omega(N-1)T_D} W^*[j(\omega - \omega_k)],$$

что совпадает с выражением для частотной характеристики.

Аналогичным образом можно определить и частотную характеристику канала ДПФ, обозначаемую  $H(j\omega_k, \omega)$ , в виде отношения его отклика на комплексный гармонический сигнал  $s(n) = e^{j\omega n T_D}$  к значению сигнала при  $n = N-1$ :

$$H(j\omega_k, \omega) = \text{ДПФ}_N \left\{ e^{j\omega n T_D} w(n) \right\} / e^{j\omega(N-1)T_D} = e^{-j\omega(N-1)T_D} W^*[j(\omega - \omega_k)].$$

Максимум частотной характеристики канала имеет место на частоте  $\omega = \omega_k$ , при этом на частоты  $\omega > 0$ , совпадающие с бинами ДПФ, откликаются каналы анализатора спектра с номерами  $k > 0$  в полосе частот  $(0 \dots \omega_D / 2)$ , а на частоты  $\omega < 0$  (иначе на сигнал  $s(n) = e^{-j\omega n T_D}$ ), совпадающие с бинами ДПФ, откликаются каналы с номерами  $k < 0$  в полосе частот  $(-\omega_D / 2 \dots 0)$ .

Таким образом, анализатор спектра на основе ДПФ обеспечивает разрешение комплексных гармонических сигналов как по частоте, так и по знаку частоты. При вычислении ДПФ в диапазоне частот  $(0...f_D)$ , при  $k = 0,1,\dots(N-1)$ , отрицательным значениям  $k_- = -1,-2,\dots-N/2$  соответствуют значения  $k = k_- + N$ .

В качестве примера можно определить частотную характеристику анализатора спектра с прямоугольной весовой функцией  $w_R(n)$  длиной  $N$ .

Ее преобразование Фурье:

$$W_R(j\omega) = e^{-j\omega(N-1)/2} [\sin(\omega NT_D/2)] / [\sin(\omega T_D/2)].$$

Каналы  $k = 0,1,\dots(N-1)/2$  анализатора спектра, откликающиеся на сигнал  $s(n) = e^{j\omega n T_D}$ , для данной весовой функции описываются частотной характеристикой:

$$H_R(j\omega_k, \omega) = e^{-j\omega(N-1)/2} \frac{\sin[(\omega - \omega_k)NT_D/2]}{\sin[(\omega - \omega_k)T_D/2]}.$$

Выражение для частотных характеристик каналов  $k_- = -1,-2,\dots-(N-1)/2$  или  $k = (N-1)/2,\dots(N-1)$ , откликающихся на сигнал  $s(n) = e^{-j\omega n T_D}$ , имеет вид:

$$H_R(j\omega_k, \omega) = e^{j\omega(N-1)/2} \frac{\sin[(\omega + \omega_k)NT_D/2]}{\sin[(\omega + \omega_k)T_D/2]}.$$

Графики модулей частотных характеристик каналов анализатора спектра представлены на рис. 4.4 [4]. Частотная характеристика канала имеет главный лепесток шириной  $\Delta f_{\text{вл}} = 2f_D/N = 2$  бина и боковые лепестки шириной 1 бин ( $f_D/N$ ) с относительным уровнем максимального бокового лепестка  $\delta_{\text{вл, max}} = -13,6\text{dB}$  (боковые лепестки на рисунке показаны только для одного канала).

Как видно из рис. 4.4 [4], частотные характеристики всех каналов перекрываются. Однозначное разрешение комплексного гармонического сигнала имеет место только на частотах  $\omega = \omega_k$ , совпадающих с частотами анализа ДПФ, когда на интервале анализа укладывается целое число периодов сигнала:

$$T_a = NT_D = kT,$$

(частота  $\omega_l$  на рис. 4.4).

В этом случае сигнал присутствует только на выходе одного канала или говорят – проецируется на 1 бин ДПФ. При этом возможны обнаружение и оценка параметров сигнала по вычисленному значению ДПФ  $X(j\omega_k)$ .

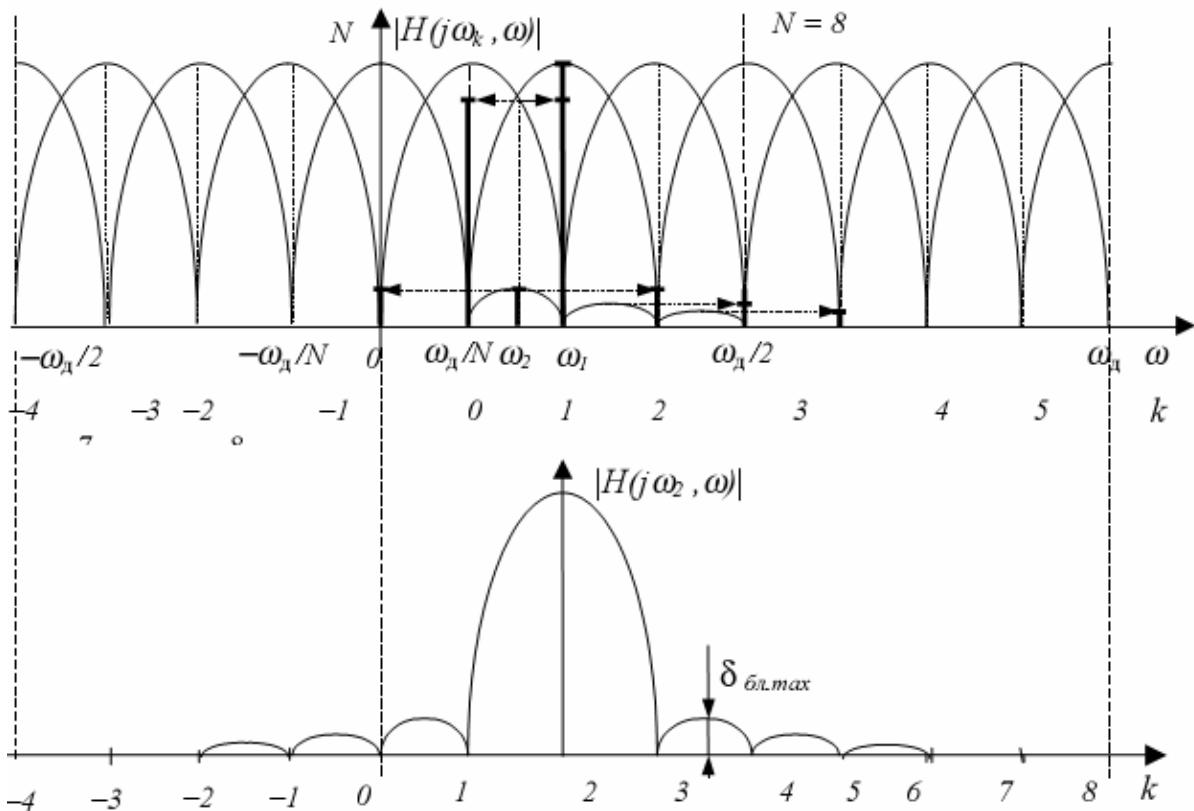


Рис. 4.4. Частотная характеристика  $N$ -канального анализатора спектра с прямоугольной весовой функцией и частотная характеристика одного канала

На сигналы с частотой  $\omega \neq \omega_k$  (частота  $\omega_2$  на рис. 4.4) откликаются два соседних канала на уровне главных лепестков их частотных характеристик, а на уровне боковых лепестков откликаются все каналы ДПФ, т. е. присутствует на выходах всех каналов. Это явление называют размыванием спектра, дроблением или просачиванием спектра между каналами. При этом затруднительны однозначное обнаружение сигнала и оценка его параметров.

Важной для практики является задача обнаружения и оценки параметров вещественных гармонических сигналов  $s(n) = (\cos \omega n T_D)$ . Такой сигнал можно представить в виде суммы двух комплексных гармонических сигналов с частотой  $\omega > 0$  и  $\omega < 0$ :

$$s(n) = \frac{1}{2}[e^{j\omega n T_D} + e^{-j\omega n T_D}] = \frac{1}{2}[s_1(n) + s_2(n)].$$

В соответствии с принципом суперпозиции, ДПФ сигнала  $s(n)$  также можно найти как сумму откликов на его комплексно-сопряженные составляющие, определяемые частотной характеристикой анализатора спектра.

## 4.2. Весовые функции. Связь ДПФ цифровой фильтрацией при анализе спектра

### 4.2.1. Весовые функции для гармонического спектрального анализа

Высокий уровень боковых лепестков и обусловленное им взаимное влияние каналов анализатора спектра является основным недостатком прямоугольной весовой функции. Особенно оно существенно при разрешении нескольких одновременно действующих сигналов или при наличии сосредоточенных помех. Эти недостатки преодолеваются при использовании весовых функций, имеющих меньший уровень боковых лепестков [4].

Для подавления просачивания обычно используют временные окна, сглаживающие исходную реализацию таким образом, чтобы подавить резкие вариации на ее начальном и конечном участках. Однако при этом возрастает ширина главного лепестка  $\Delta f_{\text{gl}} = Df_D / N$ ,  $D > 2$ . Это приводит к увеличению числа каналов с перекрывающимися главными лепестками частотных характеристик и ухудшению разрешающей способности анализатора спектра. Например, у анализатора спектра с весовой функцией Хэмминга, имеющей параметры  $\Delta f_{\text{gl}} = 4f_D / N$ ,  $\delta_{\text{gl max}} = -41 \text{dB}$ , перекрываются главные лепестки частотных характеристик трех соседних каналов (рис. 4.5) [4].

Даже при  $\omega = \omega_k$  (частота  $f_1$  на рис. 4.5) сигнал одновременно присутствует на выходах трех каналов (проецируется на три бина ДПФ), а при  $\omega \neq \omega_k$  (частота  $f_2$  на рис. 4.5) без учета боковых лепестков проецируется на 4 бина ДПФ.

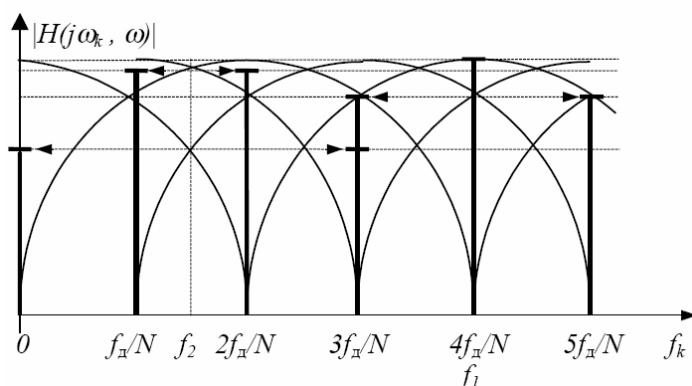


Рис. 4.5. Частотные характеристики каналов анализатора спектра с весовой функцией Хэмминга

Улучшение разрешения анализатора спектра с весовыми функциями, имеющими широкий главный лепесток, обеспечивается путем увеличения числа точек ДПФ  $N$ , т. е. увеличением времени анализа сигнала  $T_a = NT_D$ .

От параметров весовой функции зависят также фильтрующие свойства анализатора спектра при обработке зашумленных гармонических и полигармонических сигналов. Большую роль играют весовые функции и при спектральном анализе случайных сигналов. В то же время умножение на весовую функцию, отличную от прямоугольной, приводит к увеличению общего объема вычислений при спектральном анализе. Для ряда весовых функций умножение во временной области может быть заменено сверткой в частотной области с дискретизированной частотной характеристикой весовой функции. Например, для весовых функций Ханна и Хэмминга такая свертка вычисляется путем скользящего весового усреднения трех соседних частотных выборок с весами соответственно 0,25; 0,5; 0,25 и 0,23; 0,54; 0,23.

#### 4.2.2. Основные параметры весовых функций при спектральном анализе

Эквивалентная шумовая полоса весовой функции  $\Delta F_{uu}$ , выражаемая в бинах. Находится при действии на входе анализатора спектра белого шума, имеющего полосу  $\pm f_D / 2$ , из условия равенства мощности шума  $P_{u\phi\phi}$  на выходе эквивалентного цифрового фильтра нижних частот с частотной характеристикой весовой функции  $W(j\omega)$  и мощности шума  $P_{uuu}$  на выходе идеального фильтра, имеющего полосу пропускания  $\Delta F_{uu}$  и максимум квадрата АЧХ  $|W(j0)|^2$  (рис. 4.6) [4]:

$$P_{u\phi\phi} = \frac{N_0}{2\pi} \int_{-\omega_D/2}^{\omega_D/2} |W(j\omega)|^2 d\omega = \frac{N_0}{T_D} N \sum_{n=0}^{N-1} w^2(n),$$

где  $N_0$  – спектральная плотность мощности белого шума на входе фильтра (или канала).

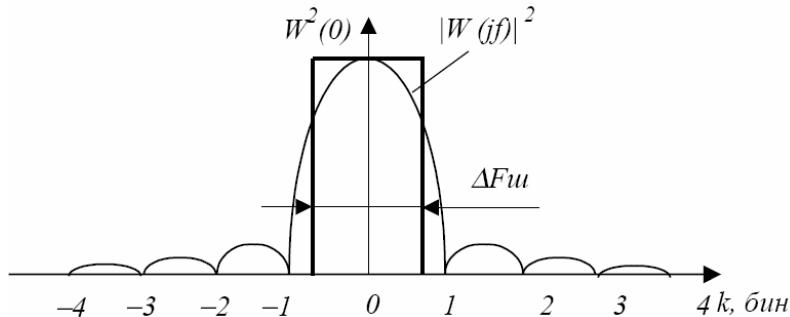


Рис. 4.6. Определение эквивалентной шумовой полосы весовой функции

Максимальное усиление весовой функции на частоте  $\omega=0$  определяется как  $W(j0) = \sum_{n=0}^{N-1} w(n)$  – по уровню, и  $|W(j0)|^2 = \left( \sum_{n=0}^{N-1} w(n) \right)^2$  – по мощности.

$$\text{Мощность шума на выходе идеального фильтра } P_{uu} = \frac{N_0}{T_D} |W(j0)|^2 \Delta F_{uu}.$$

Приравнивая мощности  $P_{uw\phi} = P_{uu}$ , можно найти выражение для эквивалентной шумовой полосы весовой функции и, соответственно, эквивалентной шумовой полосы канала анализатора спектра [4]:

$$\Delta F_{uu} = N \sum_{n=0}^{N-1} w^2(n) / \left( \sum_{n=0}^{N-1} w(n) \right)^2, \text{ бин.}$$

При этом мощность действующего на входе анализатора спектра белого (широкополосного) шума уменьшается на выходе анализатора спектра в число раз, равное  $N / \Delta F_{uu}$ .

Ширина главного лепестка частотной характеристики весовой функции по уровням 3 и 6 дБ, в бинах:  $\Delta F_{el(3/6)}$ . Значение  $\Delta F_{el(6)}$  определяет разрешающую способность анализатора спектра по частоте:

$$\Delta f_p = \Delta F_{el(6)} f_D / N.$$

Если  $\Delta f_p$  задано, то по нему находится необходимое время анализа:

$$NT_D = \Delta F_{el(6)} / \Delta f_p.$$

Совокупность параметров  $\Delta F_{el(3)}$  (полосы по уровню половинной мощности) и  $\Delta F_{uu}$  определяет качество весовой функции  $\delta = (\Delta F_{el(3)} - \Delta F_{uu}) / \Delta F_{el(3)}$  в случае, когда входная последовательность представляет сумму гармонических составляющих с частотами, не кратными  $f_D / N$ , т. е. не совпадающими с бинами ДПФ, и белого шума.

Когерентное усиление весовой функции:  $K_{koe} = |W(j0) / N|$ . Определяет нормированное к  $N$  усиление канала ДПФ с данной весовой функцией на центральной частоте канала  $f_k$ .

Максимальный уровень боковых лепестков  $\delta_{el\max}$ , дБ.

Паразитная амплитудная модуляция спектра. Определяется отношением значения частотной характеристики весовой функции на частоте  $f = f_D / 2N$  к максимальному значению частотной характеристики на частоте:

$$f = 0 : A_{n,\partial B} = 20 \lg \left| W(j\omega) \Big|_{(\omega=\omega_D/2)} / W(j0) \right|.$$

Значение  $A_n$  характеризует искажения или потери оценок ДПФ в случае самых неблагоприятных для ДПФ частот сигнала  $f = f_k \pm f_D / 2N$ .

Значения параметров для некоторых весовых функций представлены на рис. 4.7[4].

Тип ВФ	$\Delta F_{uu}$ , бин	$\Delta F_{\text{эл}(3/6)}$ , бин	$K_{\text{коэ}}$	$\delta_{\text{бл}}, \text{дБ}$	$A_{\Pi}, \text{дБ}$
Прямоугольная	1	0,89 / 1,21	1	-13,6	-3,92
Треугольная	1,33	1,28 / 1,78	0,5	-27	-1,82
Хемминга	1,36	1,30 / 1,81	0,54	-41	-1,78
Блэкмана	1,73	1,68 / 2,36	0,42	-57	-1,1

Рис. 4.7. Значение параметров весовых функций

#### 4.2.3. Элементы анализа спектра с помощью ДПФ и цифровой фильтрации

Наряду с оценкой спектра на заданном интервале наблюдения, существуют задачи анализа изменения спектра во времени или текущего спектрального анализа, относящегося к нестационарным детерминированным и случайному сигналам. Для этой цели применяют скачущие и скользящие ДПФ [4].

Скачущими называют ДПФ с неперекрывающимися (стыкующимися) или частично перекрывающимися интервалами наблюдения с коэффициентами перекрытия 0, 0,25, 0,5, 0,75 и другими.

Скользящее ДПФ вычисляется по текущему  $N$ -точечному набору или вектору данных  $\{s(n - N + 1) - s(n)\}$  при значениях  $s = N - 1 + l$ ,  $l = 0, 1, 2, \dots$ . Очевидно, что скачущие ДПФ являются лишь отсчетами скользящего ДПФ при значениях  $n = N - 1 + lN$  (скачущее ДПФ без перекрытия),  $n = N - 1 + lN/2$  (скачущее ДПФ с перекрытием 0,5) и т. д.

Скачущие ДПФ с коэффициентами перекрытия 0,25, 0,5 и более используются и для усреднения шумовой составляющей погрешности анализа стационарных сигналов при общем ограниченном времени наблюдения сигнала.

Канал скачущего ДПФ эквивалентен согласованному ЦФ, с выхода которого снимается каждый  $N$ -й или каждый  $(N/2)$ -й и т. д. отсчеты сигнала, что соответствует понижению частоты дискретизации выходного сигнала фильтра в  $N$ ,  $N/2$  и т. д. раз. При этом периодически повторяющийся с частотой дискретизации спектр выходного сигнала фильтра периодизируется с мень-

шей в соответствующее число раз частотой  $f'_D = f_D / N$ ,  $f''_D = 2f_D / N$ , в результате чего происходит наложение спектров в основной полосе частот  $0 \pm f'_D$ . В эту полосу и трансформируются или заворачиваются все внеполосные частотные составляющие обрабатываемого сигнала. Наиболее сильно эффект наложения проявляется у ЦФ, эквивалентных скачущему ДПФ без перекрытия (с понижением частоты дискретизации в  $N$  раз). Наглядное представление о нем дает наложение в основной полосе частот  $(0 \pm f_D / 2N)$  разнесенных на  $kf_D / N$  частотных характеристик каналов ДПФ. Трансформация частоты  $f_i = kf'_D + \Delta f_i$   $i$ -частотной составляющей сигнала в частоту  $f'_i$  основной полосы  $(0 \pm f'_D / 2)$  осуществляется в данном случае в соответствии с соотношением [4]:  $f'_i = \Delta f_i$ , если  $k$  – четное и  $f'_i = -\Delta f_i$ , если  $k$  – нечетное, где  $k = [f_i / f'_D]_{\text{ч.окр.}} = 0, \pm 1, \dots, \pm N/2$  – целая часть отношения  $f_i / f'_D$  с округлением;  $\Delta f_i = f_i - kf'_D$ ,  $-f'_D / 2 \leq \Delta f_i \leq f'_D / 2$ . Кратность наложения в данном случае составляет  $N$ .

При коэффициенте перекрытия 0,5 (коэффициенте прореживания  $N/2$ ) частота дискретизации выходного сигнала ЦФ составляет  $f'_D = 2f_D / 2$ , а основная полоса частот  $(0 \pm f_D / N)$ . Она соответствует ширине главного лепестка частотной характеристики ЦФ с прямоугольной весовой (импульсной) характеристикой, для которого периодизированные с периодом  $2f_D / N$  частотные характеристики не перекрываются на уровне главных лепестков, а их перекрытие в основной полосе частот  $(0 \pm f_D / N)$  также характеризует эффект наложения и связанные с ним частотные преобразования. Трансформация частот здесь осуществляется в соответствии с теми же соотношениями, что и в предыдущем случае. Кратность наложения при этом составляет  $N/2$ .

Эффект наложения при спектральном анализе с перекрытием может быть существенно ослаблен использованием весовых функций с малым уровнем боковых лепестков. В то же время перекрытие окон (интервалов наблюдения) ведет к увеличению объема вычислений и в каждом конкретном случае должно быть обоснованным.

Скользящее ДПФ требует очень большого объема вычислений, поэтому в большинстве случаев его заменяют гребенкой цифровых фильтров, состоящей из гребенчатого фильтра и набора комплексных цифровых резонаторов (рис. 4.8) [4].

Такой анализатор спектра с прямоугольной весовой функцией требует  $N$  комплексных операций умножения на один отсчет сигнала. Число каналов его определяется требуемой полосой анализа сигнала. Каждый канал анализатора спектра описывается следующим рекуррентным уравнением:

$$X(j\omega_k, n) = e^{-j\omega_k n T_D} X(j\omega_k, n-1) + s(n) - s(n-N).$$

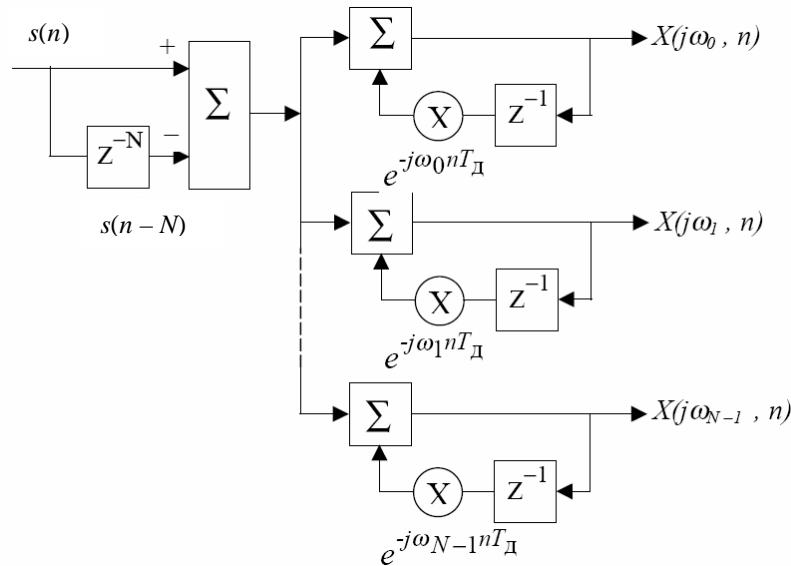


Рис. 4.8. Структура анализатора спектра на основе гребенки фильтров

### 4.3. Корреляционные характеристики дискретных случайных сигналов и их статистические оценки

#### 4.3.1. Корреляционные и спектральные характеристики дискретных случайных сигналов

Дискретные случайные сигналы характеризуются [4]: законом распределения, средним значением, средним квадратом или дисперсией, автокорреляцией или автоковариацией, взаимной корреляцией (кросс-корреляцией) или взаимной ковариацией, спектральной плотностью мощности и взаимной спектральной плотностью мощности. Для рассматриваемых далее случайных процессов стационарных в широком смысле и эргодичных полагается, что их среднее значение является постоянным, не зависящим от времени  $nT_D$ , т. е. от номера выборки  $n$ , а автокорреляция и взаимная корреляция зависят только от разности моментов времени  $mT_D = n_2 T_D - n_1 T_D$ , т. е. от значения дискретной задержки  $m$ . Они определяются как математические ожидания  $E\{\cdot\}$  (средние по ансамблю) или как средние по времени (по одной реализации) соответствующих дискретных функций.

Среднее значение дискретного сигнала  $s(n)$ :

$$\bar{s} = E\{s(n)\} = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N s(n).$$

Автокорреляционная функция (АКФ) дискретного сигнала  $s(n)$ :

$$r_{ss}(m) = E\{s(n+m)s^*(n)\} = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N s(n+m)s^*(n),$$

где символ  $*$  – означает операцию комплексного сопряжения, обобщающую данные статистические характеристики на случай комплексных дискретных сигналов.

Взаимная корреляционная функция (ВКФ) двух дискретных сигналов  $s(n)$  и  $y(n)$ :

$$r_{sy}(m) = E\{s(n+m)y^*(n)\} = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N s(n+m)y^*(n).$$

По максимуму ВКФ находят относительное время задержки обрабатываемых сигналов.

АКФ и ВКФ центрированных случайных сигналов  $[s(n) - \bar{s}]$  и сигналов с нулевым средним значением называют автоковариационной  $c_{ss}(m)$  и взаимной ковариационной  $c_{xy}(m)$  функциями:

$$c_{ss}(m) = E\left\{ [s(n+m) - \bar{s}] [s^*(n) - \bar{s}^*] \right\};$$

$$c_{sy}(m) = E\left\{ [s(n+m) - \bar{s}] [y^*(n) - \bar{y}^*] \right\}.$$

При этом  $c_{ss}(m) = r_{ss}(m) - |\bar{s}|^2$ ,  $c_{sy}(m) = r_{sy}(m) - \bar{s}\bar{y}^*$ .

АКФ  $r_{ss}(m)$  при  $m=0$  соответствует среднему квадрату случайного сигнала:

$$\overline{s^2} = r_{ss}(0) = E\{s(n)s^*(n)\} = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N s(n)s^*(n),$$

а автоковариационная функция  $c_{ss}(m)$  – дисперсия сигнала:

$$c_{ss}(0) = D_x = \sigma_x^2.$$

Они связаны соотношением:  $D_x = \overline{s^2} - \overline{|s|}^2$  и определяют среднюю мощность случайного сигнала. Средний квадрат и дисперсия являются максимальными значениями соответственно автокорреляционной и автоковариационной функций. Термины (средний квадрат – дисперсия), (автокорреляция – автоковариация) используются как синонимы.

Спектральную плотность мощности (СПМ) случайного процесса в соответствии с теоремой Винера-Хинчина определяют преобразованием Фурье АКФ (последовательности  $r_{ss}(m)$ ) [4]:

$$P_{ss}(f) = T_D \sum_{m=-\infty}^{\infty} r_{ss}(m) e^{-j2\pi f m T_D},$$

для значений частоты, заключенных в основной полосе частот  $(0 \pm f_D / 2)$ .

СПМ показывает, как мощность случайного процесса распределена по частоте. Так как  $r_{ss}(-m) = r_{ss}^*(m)$ , то СПМ является строго действительной положительной функцией. Если АКФ – действительна, то  $r_{ss}(-m) = r_{ss}(m)$  и выражение для СПМ приводится к виду:

$$P_{ss}(f) = 2T_D \sum_{m=-\infty}^{\infty} r_{ss}(m) \cos(2\pi f m T_D).$$

При этом  $P_{ss}(f)$  обладает свойством четной симметрии:  $P_{ss}(f) = P_{ss}(-f)$ .

АКФ случайного сигнала может быть определена по его известной СПМ с помощью обратного преобразования Фурье:

$$r_{ss}(m) = \int_{-f_D/2}^{-f_D/2} P_{ss}(f) e^{j2\pi f m T_D} df.$$

$$\text{При } m=0 \quad r_{ss}(0) = \int_{-f_D/2}^{-f_D/2} P_{ss}(f) df \text{ м соответствует средней мощности}$$

сигнала или его дисперсии. Аналогичным образом вводится определение взаимной спектральной плотности мощности (ВСПМ) двух стационарных случайных сигналов  $s(n)$  и  $y(n)$  как преобразование Фурье их ВКФ:

$$P_{sy}(if) = T_D \sum_{m=-\infty}^{\infty} r_{sy}(m) e^{-j2\pi f m T_D}.$$

Даже для действительных последовательностей  $s(n)$  и  $y(n)$  вследствие несимметрии их ВКФ ВСПМ является в общем случае комплексной. В то же время она обладает свойством  $P_{sy}(jf) = P_{ys}^*(jf)$ .

#### **4.3.2. Статистические оценки корреляционных характеристик дискретных случайных сигналов**

При спектральном и корреляционном анализе случайных сигналов вместо истинных АКФ, ВКФ, СПМ, ВСПМ используются их статистические оценки, получаемые по конечному набору данных или по реализациям сигнала конечной длины, а для СПМ и ВСПМ – и конечному числу усредняемых реализаций сигнала. Они определяются следующим образом [4].

Оценка автокорреляции (АКФ):

$$\hat{r}_{ss}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} s(n+m)s^*(n), \quad 0 \leq m \leq N-1,$$

$$\hat{r}_{ss}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} s^*(n+|m|)s(n), \quad -(N-1) \leq m \leq 0.$$

Данная оценка является смещенной, ее математическое ожидание (произведение истинной АКФ и треугольной весовой функцией) не равно  $r_{ss}(m)$ . Однако, она более эффективна, чем несмещенная оценка.

Оценка взаимной корреляции (ВКФ):

$$\hat{r}_{sy}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} s(n+m)y^*(n), \quad 0 \leq m \leq N-1,$$

$$\hat{r}_{sy}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} s(n)y^*(n+|m|), \quad -(N-1) \leq m \leq 0.$$

Она также является смещенной.

При оценке авто и взаимной ковариации используется оценка среднего значения случайного процесса:

$$\hat{s}_N = \frac{1}{N} \sum_{n=0}^{N-1} s(n).$$

### 4.3.3. Статистические оценки спектральной плотности мощности дискретных случайных сигналов

Статистическими оценками СПМ и ВСПМ как преобразований Фурье АКФ и ВКФ случайных процессов являются соответственно преобразования.

Фурье введенных выше статистических оценок АКФ и ВКФ, вычисляемых по реализациям сигнала конечной длины  $N$  [4]:

$$\widehat{P}_{ss}(f) = T_D \sum_{m=-(N-1)}^{N-1} \widehat{r}_{ss}(m) e^{-j2\pi fmT_D} .$$

$$\widehat{P}_{sy}(jf) = T_D \sum_{m=-(N-1)}^{N-1} \widehat{r}_{sy}(m) e^{-j2\pi fmT_D} .$$

Аналогичным образом определяются оценки СПМ (ВСПМ)  $\tilde{P}_{ss}(f)$ ,  $\tilde{P}_{ss}(jf)$  по смещенным оценкам АКФ и ВКФ  $\tilde{r}_{ss}(m)$ ,  $\tilde{r}_{sy}(m)$ . Обе оценки СПМ (ВСПМ) являются смещенными: их математические ожидания представляют собой свертки истинной СПМ (ВСПМ)  $P_{ss}(f)$  ( $P_{sy}(jf)$ ) с частотной характеристикой  $W(jf)$  весовой функции (окна)  $w(m)$ , ограничивающего длину преобразуемой АКФ (ВКФ). При несмещенной оценке АКФ (ВКФ) – это прямоугольное окно  $w_R(m)$  и

$$E\{\widehat{P}_{ss}(f)\} = P_{ss}(f) * W_R(f), \quad E\{\widehat{P}_{sy}(jf)\} = P_{sy}(jf) * W_R(f),$$

при смещенной оценке АКФ (ВКФ) – это треугольное окно  $w_T(m)$  с частотной характеристикой  $W_T(f) = W_R^2(f/2)$  и

$$E\{\tilde{P}_{ss}(f)\} = P_{ss}(f) * (2/N)W_R^2(f),$$

$$E\{\tilde{P}_{ss}(jf)\} = P_{ss}(jf) * (2/N)W_R^2(f),$$

где  $W_R(f) = \sin(\pi f NT_D) / \sin(\pi f T_D)$ .

Дисперсия (погрешность) оценок СПМ (ВСПМ), как и оценок АКФ (ВКФ), возрастает с уменьшением длины реализации  $N_1$  и увеличением максимального вычисляемого индекса  $m_{max}=M-1$ , где  $M \leq N_1$ , что обусловлено уменьшением эффекта усреднения. Максимальная погрешность оценки СПМ (ВСПМ) имеет место при  $M = N_1$ . Она соизмерима со средним значением СПМ (ВСПМ) и является практически неприемлимой. Вычисление СПМ (ВСПМ) случайных сигналов по оценкам их АКФ (ВКФ) называют косвенным методом спектрального анализа случайных сигналов или методом коррелограмм.

#### 4.3.4. Коррелограммный метод вычисления спектральной плотности мощности дискретных случайных сигналов

Вычисляя с помощью ДПФ автокорреляционные и взаимокорреляционные последовательности  $r_{ss}(m)$ ,  $r_{sy}(m)$  для значений задержки  $-(M-1) \leq m \leq (M-1)$ , где  $M \leq N_1$ , взвешивая их корреляционным окном  $\omega(m)$  той же длины и выполняя их ДПФ по числу точек  $N = 2M - 1$ , можно найти коррелограммные оценки СПМ  $P_{ss}(k)(f)$  и ВСПМ  $P_{sy}(k)(jf)$  на дискретных частотах  $f_k = kf_D/N$  [4]:

$$\widehat{P}_{ss(k)}(f_k) = \sum_{m=-(M-1)}^{M-1} r_{ss}(m) e^{-j2\pi f_k m T_D} = 2 \sum_{m=0}^{M-1} r_{ss}(m) \cos(2\pi f_k m T_D);$$

$$\widehat{P}_{sy(k)}(jf_k) = \sum_{m=-(M-1)}^{M-1} r_{sy}(m) e^{-j2\pi f_k m T_D},$$

где  $k = 0, \dots, 2M - 1$ .

Число  $N$  вычисляемых точек ДПФ может быть увеличено добавлением к преобразуемой последовательности  $N_0$  нулевых отсчетов с целью уменьшения шага дискретизации оценок СПМ (ВСПМ) по частоте  $(f_D / N)$  или приведения  $N$  к целой степени двойки:  $N = 2M - 1 + N_0$ , необходимой для применения алгоритма БПФ.

#### 4.3.5. Вычисление спектральной плотности мощности методом периодограмм

Исходными данными для спектрального анализа методом периодограмм являются [4]:

- частота дискретизации сигнала  $f_D$ ;
- требуемое разрешение по частоте  $\Delta f_p$ ;
- общее доступное время наблюдения (регистрации) сигнала  $T_h$ .

Спектральный анализ включает в себя следующие этапы.

Выбирается тип весовой функции  $w(n)$ , исходя из особенностей решаемой задачи, и находится необходимое время анализа или длина реализации (секции) сигнала  $N_1$  из условия некоррелированности соседних секций случайного сигнала и обеспечения требуемого частотного разрешения:

$$\Delta f_p = \Delta F_{\text{ел}}(6) f_D / N_1 : N_1 = \Delta F_{\text{ел}}(6) f_D / \Delta f_p.$$

Выбирается способ разбиения сигнала на реализации (секции) конечной длины  $N_1$ : без перекрытия секций или с перекрытием секций (рис. 4.9) [4]. Коэффициент перекрытия  $K_{nep}$  может быть равным 0,25; 0,5; 0,75. Чаще всего используется значение  $K_{nep} = 0,5$ . Перекрытие уменьшает общее время анализа и влияние разрывов сигнала на краях секций, но несколько ухудшает точность оценок СПМ, которая также зависит от типа весовой функции.

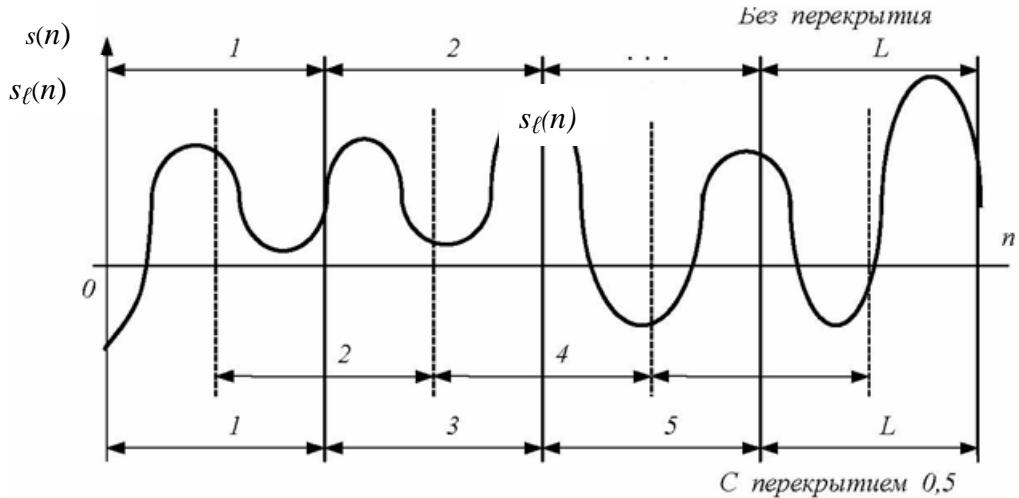


Рис. 4.9. Способы разбиения обрабатываемого сигнала на секции

Выбирается или находится число усредняемых реализаций (секций)  $L$  из следующих возможных условий: из общего доступного времени наблюдения сигнала  $T_h$ :  $L = \{[(T_h/T_d) - K_{nep}N_1]/(N_1 - K_{nep}N_1)\}_{ц.ч.}$ , где (ц.ч.) – символ целой части отношения, заключенного в скобки; из стационарности случайного сигнала на интервале  $T_c = LN_1T_D$ ; из необходимой степени  $q$  уменьшения дисперсии оценки СПМ, т. е. эффективности усреднения. При  $K_{nep} = 0$  (без перекрытия) степень уменьшения дисперсии  $q = L$ , при  $K_{nep} = 0,5$   $q = L/\left[1 + 2c_{(0,5)}^2(L-1)/L\right] < L$ , где  $c_{(0,5)}$  – коэффициент, определяемый видом весовой функции:  $c_{(0,5)} = 0,5; 0,25; 0,235$  соответственно для весовых функций прямоугольной, треугольной и Хэмминга. Например, при использовании весовой функции Хэмминга,  $K_{nep} = 0,5$  и  $L = 10$  степень уменьшения дисперсии составляет  $q \approx 9$ .

Вычисляются  $N_1$ -точечные ДПФ для  $L$  реализаций сигнала  $s_l(n) = s[n + (l-1)(1 - K_{nep})/N_1]$ ,  $n = 0, \dots, N_1 - 1$ :

$$\hat{S}_l(jk) = \sum_{n=0}^{N_1-1} s_l(n)w(n)e^{-j2\pi kn/N}, \quad k = 0, \dots, N_1 - 1, \quad l = 1, 2, \dots, L.$$

Вычисляются значения  $L$  периодограмм:

$$I_{sl}(k) = \frac{1}{U} |\widehat{S}_l(jk)|^2,$$

где  $U = \sum_{n=0}^{N_1-1} w^2(n).$

Находится усредненная периодограмма или оценка СПМ:

$$\widehat{P}_s(k) = \frac{1}{L} \sum_{l=1}^L I_{sl}(k).$$

Номеру частотной выборки  $k$  СПМ соответствуют значения частоты  $f_k = kf_D / N_1$ .

При недостаточном разрешении по частоте реализации дополняются нулевыми отсчетами  $N_0$ , а при большой дисперсии увеличивается число усредняемых реализаций СПМ –  $L$ . Структурная схема анализатора спектра на основе метода периодограмм приведена на рис. 4.10 [4].

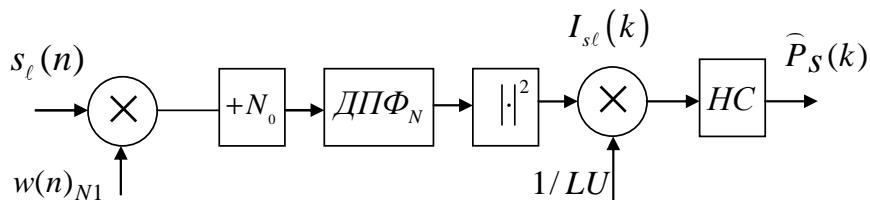


Рис. 4.10. Структурная схема анализатора СПМ случайных сигналов методом периодограмм

На схеме обозначены:  $N_1$  – длина секции (весовой функции);  $N = N_1 + N_0$  – число точек (частотных выборок) ДПФ;  $HC$  – накапливающий сумматор, осуществляющий накопление (усреднение) периодограмм.

При вычислении ВСПМ методом периодограмм (рис. 4.11) [4] секционирование и взвешивание выполняется рассмотренным выше образом для обеих обрабатываемых последовательностей  $s(n)$  и  $y(n)$ . Периодограмма для ВСПМ определяется как:

$$I_{syl}(k) \frac{1}{U} \mathcal{D}\Phi_N [w(n)s_l(n)] \mathcal{D}\Phi_N^* [w(n)y_l(n)] = \frac{1}{U} \widehat{S}_l(jk) \widehat{Y}^*(jk),$$

а усредненная периодограмма или оценка ВСПМ:

$$\widehat{P}_{sy}(k) = \frac{1}{L} \sum_{l=1}^L I_{syl}(k).$$

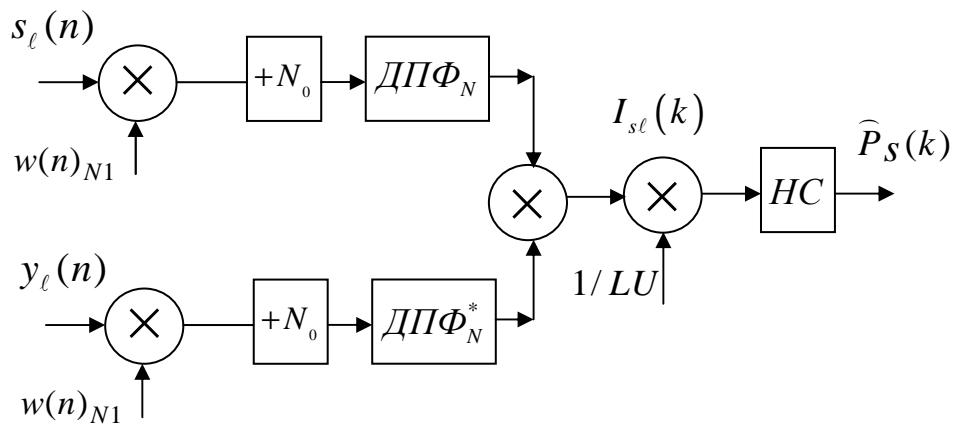


Рис. 4.11. Структурная схема анализатора ВСПМ случайных сигналов методом периодограмм

#### 4.4. Вопросы и задания для самопроверки

1. В каких областях используется спектральный анализ сигналов?
2. Назовите основные методы спектрального анализа.
3. Относится ли частота дискретизации сигнала к основным параметрам его спектра?
4. Назовите основные параметры анализаторов спектра.
5. Представьте структуру анализатора спектра на основе ДПФ.
6. Для решения каких задач в спектроанализаторах применяются весовые функции?
7. Представьте значения основных параметров известных вам весовых функций.
8. В чем отличие между скачущим и скользящим ДПФ?
9. Представьте структуру анализатора спектра на основе гребенки фильтров.
10. Назовите основные характеристики дискретных случайных сигналов.
11. Запишите выражения для определения АКФ и ВКФ.
12. Что характеризуют СПМ и ВСПМ и на основе каких выражений их можно определить?
13. Запишите выражения для статистических оценок АКФ, ВКФ, СПМ, ВСПМ.
14. В чем сущность коррелограммного метода вычисления спектральной плотности мощности дискретных случайных сигналов?
15. Назовите основные шаги алгоритма спектрального анализа сигналов методом периодограмм.

## **МОДУЛЬ 5. ВВЕДЕНИЕ В ЦИФРОВУЮ ОБРАБОТКУ ИЗОБРАЖЕНИЙ. ОБРАБОТКА БИНАРНЫХ ИЗОБРАЖЕНИЙ**

Цель модуля – изучение студентами особенностей обработки изображений, методов и алгоритмов обработки бинарных изображений.

В результате изучения модуля студенты *должны знать*:

- структуру системы обработки изображений;
- основные алгоритмы обработки бинарных изображений.

Студенты *должны уметь*:

- использовать алгоритмы обработки бинарных изображений для удаления шумов, скелетизации, выделения контуров;
- а также иметь представление об особенностях человеческого зрения и зрительных явлениях.

### **5.1. Направления развития и области применения цифровой обработки изображений**

#### **5.1.1. Направления обработки изображений и развитие методов**

Зрительные образы для человека являются основой восприятия окружающего мира, т. к. до 90 % информации человек получает и осмысливает в виде зрительных образов [13]. Изображение несет в себе информацию об объекте и в этом смысле может рассматриваться как многомерный сигнал, описываемый функцией двух или большего числа переменных.

При обработке информации, связанной с изображением на мониторе, выделяют основные направления из области информатики: компьютерное «зрение», распознавание образов, компьютерная (машинная) графика и обработка изображений.

Компьютерное «зрение» (Computer (Machine) Vision) занимается анализом образов. Для некоторой сцены (аудитория, комната и др.) компьютер должен дать описание этой сцены (есть ли в ней предметы, какая освещенность и т. д.). Таким образом, компьютерное «зрение» переводит изображение в описание (рис. 5.1).

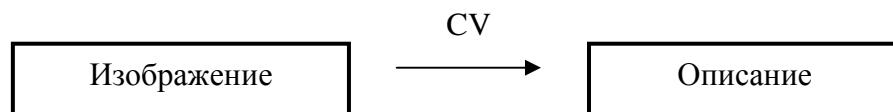


Рис. 5.1. Схема компьютерного «зрения»

Распознавание – это процесс отнесения изображения к некоторому определенному классу. В широком смысле допустимо трактовать распознавание как построение полной модели изображения, которую можно воспроизвести посредством машинной графики.

Компьютерная (машинная) графика (Computer Graphics) – это синтез изображения на основе математического описания предметов, источников света, явлений и т. д., т. е. создание виртуального образа (рис. 5.2).

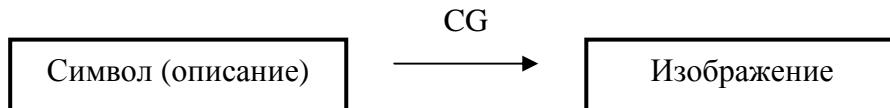


Рис. 5.2. Схема компьютерной (машинной) графики

Обработка изображений (Image Processing). Обработка изображений (в узком смысле) решает задачи преобразования изображений, поэтому исходными и результирующими данными этого этапа являются цифровые изображения. В прикладном плане – это построение некоторой системы (или программного комплекса), которая решает определенный класс задач от получения цифрового изображения до интерпретации его содержания и извлечения некоторой информации. Задачей описания изображений является извлечение набора предварительных признаков, позволяющих охарактеризовать и далее распознать изображение или его отдельные объекты.

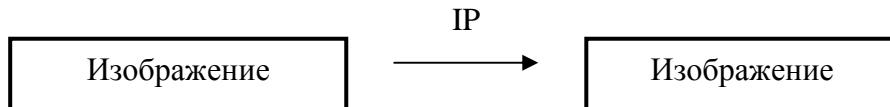


Рис. 5.3. Схема обработки изображения

В общем случае, понятие обработки изображений многопланово и имеет пересечения с другими областями. В тоже время, каждое из перечисленных направлений – это своя, достаточно широкая область науки. Не существует общепринятой точки зрения, где заканчивается обработка изображений и начинаются другие вышерассмотренные области.

Изображения представляют собой специфический вид информации и их специфика еще не полностью изучена ни в исследованиях, посвященных изучению зрительного восприятия человека, ни в информатике. Одной из особенностей является избыточность практически любого изображения. Причем изображение одной и той же сцены реального мира несет разную информацию разным пользователям, что не позволяет значительно уменьшить объем данных, содержащихся, например, в космическом снимке, хранящемся в базе данных изображений [5,13].

*Цифровым изображением* называются данные, организованные в виде числового массива, воспроизводящего свойства некоторой реальной сцены или синтезированного человеком объекта (чертежа, карты) и тесно связанного со способом получения этих данных [5,6]. Для более полного понимания термина «цифровое изображение» важное значение имеет физический механизм получения данных, ориентированных на их последующую визуализацию и обработку. Повседневный опыт связывает понятие изображения с тем, «что видит глаз». При этом формирование изображения подчиняется законам оптики и особенностям сетчатки. Технические средства позволяют получать изображения другими путями. Примерами служат радиолокация, голография, рентгеноскопия, тепловидение, ультразвуковое сканирование и др. Данные могут быть также представлены в графической форме, визуальный анализ которой позволяет делать некоторые заключения (пример – кардиограммы). Термин многомерные видеографические данные, а короче видеоданные или видеоинформация, употребляется все чаще. Если эти данные ориентированы на визуальный анализ, их называют изображениями.

Согласно [5], одним из первых применений цифровых изображений было передача по кабелю изображений между Лондоном и Нью-Йорком в начале 1920-х годов. На рис. 5.4 [5] представлено пятиградационное цифровое изображение, полученное с кодовой ленты, на рис. 5.5 – цифровое изображение после прохождения сигнала через Атлантику дважды, на рис. 5.6 показано изображения, переданное по кабелю с помощью пятнадцатиградационного оборудования (изображения данного пункта заимствованы из [5]).



Рис. 5.4. Цифровое изображение, полученное в 1921 году

Рис. 5.5. Цифровое изображение, полученное в 1922 году

Рис. 5.6. Цифровое изображение, полученное в 1929 году

Цифровой обработкой изображений называется обработка цифровых изображений с помощью цифровых вычислительных машин, т. е. компьютеров. Поэтому, хотя приведенные изображения и являются цифровыми их нельзя рассматривать как результат цифровой обработки, т. к. при этом не использовались цифровые вычислительные машины.

Как уже было отмечено, изображение представляет собой многомерный сигнал, для обработки которого требуются значительные вычислительные затраты. Первые компьютеры с мощностью, достаточной для выполнения реальных задач обработки изображений, появились в начале 1960 гг. В это же время все шире начали распространяться программы изучения космоса. Вот эти два фактора привели к началу развития методов цифровой обработки изображений. Первые космические снимки лунной поверхности были сделаны советской автоматической межпланетной станцией в 1959 г., а в 1964 г. полученные американским космическим аппаратом снимки луны (рис. 5.7) были подвергнуты компьютерной обработке для улучшения качественных характеристик. Параллельно с космическими исследованиями, цифровая обработка изображений начала все шире применяться в медицине, астрономии, ядерной физике, биофизике и т. д. Значительным импульсом в применении цифровой обработки изображений в медицине было изобретение в начале 1970 гг. рентгеновской вычислительной томографии (компьютерной томографии), при которой формируется трехмерное представление сканируемого объекта.

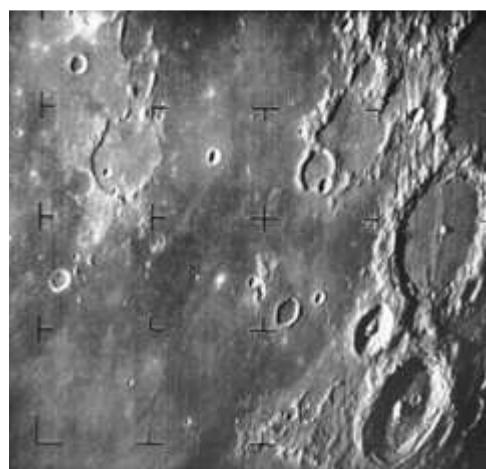


Рис. 5.7. Изображение лунной поверхности, полученное в 1964 г. американским космическим аппаратом

В это же время получены первые эффективные алгоритмы обработки изображений, например, в 1965 г. Кули и Тьюки опубликовали реализованный ими на компьютере алгоритм быстрого преобразования

Фурье, ранее известный только узкому кругу математиков, и это событие также стало мощным толчком в развитие идей и методов цифровой обработки изображений.

### 5.1.2. Применение методов цифровой обработки изображений для решения прикладных задач

*Системы обработки изображений земной поверхности и контроля погодных условий.* Для обработки изображений земной поверхности в ряде стран мира созданы национальные центры сбора и обработки данных дистанционного зондирования земной поверхности, которое обычно охватывает несколько зон в видимом и инфракрасном диапазонах спектра. Например, спутник LANDSAT (США) использует следующие зоны [5]:

- видимый зеленый цвет используется для измерения плотности растительного покрова;
- видимый красный цвет используется для различения формы растительности;
- средний инфракрасный диапазон используется для определения содержания влаги в почве и растительности и т. д.

В США в рамках NASA функционируют несколько таких центров, различающихся территориальной ответственностью и обрабатывающих данные от различных типов космических аппаратов.

На рис. 5.8 приведено изображение в трех вышеперечисленных зонах, содержащее растительность, реку, дома, дороги. На рис. 5.8 четко выделяется река на фоне берегов.

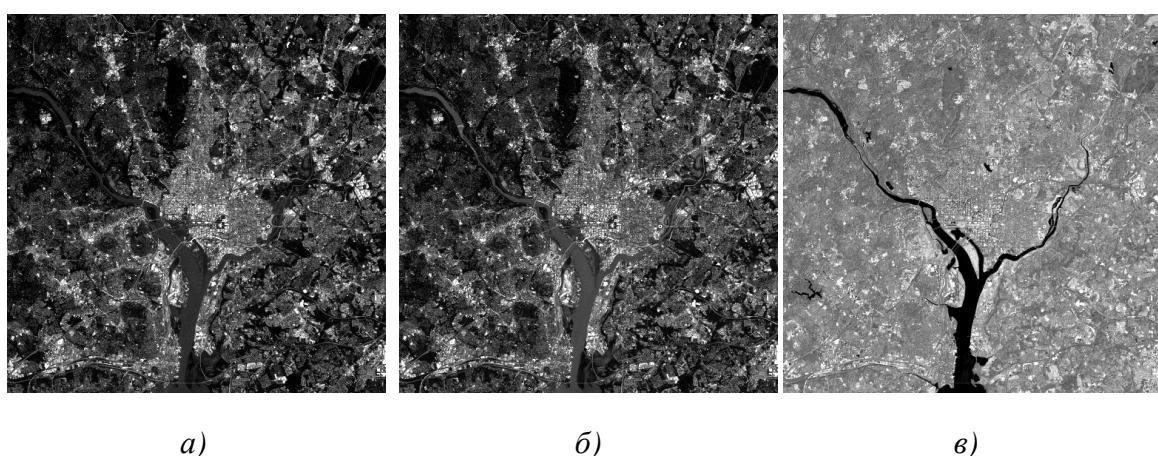


Рис. 5.8. Пример многозональных изображений:  
а) в видимом зеленом цвете; б) в видимом красном цвете;  
в) в среднем инфракрасном диапазоне

Канада в настоящее время является одной из лидирующих в мире стран в области дистанционного зондирования Земли, совершенствования технологии дистанционного зондирования и ее продажи в Западной Европе, США, Австралии, Японии. Ведущее значение в сборе и обработке данных дистанционного зондирования играет созданный в 1971 г. Канадский центр дистанционного зондирования CCRS (Canada Center for Remote Sensing). Основными задачами центра являются:

- сбор, обработка и распространение данных ДЗ;
- координация всех аспектов получения данных с искусственных спутников Земли (ИСЗ);
- прогнозирование получения данных;
- оказание помощи промышленности и другим потребителям в использовании данных ДЗ;
- совершенствование методов и средств получения и обработки данных ДЗ.

Центр обладает возможностями для приема информации с ИСЗ SPOT (Франция) и LANDSAT (США), MOS-I (Япония), ERS-I (Западноевропейское управление космических исследований) и для этой цели имеет наземные станции приема данных [5].

Аналогичного типа службы созданы и в Великобритании. С 1980 г. функционирует национальный центр дистанционного зондирования NRCS (National Remote Sensing Center), который проводит основную работу в Великобритании по сбору, обработке и распространению космической информации как для потребителей в Англии, так и для других стран. В 1969 г. был организован центр данных наблюдений Земли EODC (Earth Observation Data Center). Этот центр создан для обработки, хранения и распространения данных ДЗ с западноевропейского спутника ERS-I японского ИСЗ JERS-I.

Прием данных осуществляется с помощью своих наземных станций, а также от других пунктов приема в виде записи информации на оптических дисках и магнитных лентах с высокой плотностью записи.

В Австралии проблемами использования космической информации в рамках национальной космической программы занимаются центры прикладного использования и центры развития космической промышленности SIDC (Space Industry Development Center).

В странах СНГ также существуют подобные структуры. Так, в Москве и Подмосковье существует и действует в настоящее время целый ряд организаций, осуществляющих сбор, обработку, хранение и коммерческую ре-

лизацию продукции, это и крупные организации типа Госцентр «Природа», МГУ, а также большое количество различных подразделений других организаций. Однако единого центра сбора, обработки, хранения и распределения данных ДЗ не создано. Из других регионов широко известна группа подразделений Сибирского отделения АН в Новосибирске, где на протяжении последних 15 лет создавался и функционирует целый программно-технический комплекс, позволяющий осуществлять непосредственный прием, различные виды обработки, хранение и распространение данных дистанционного зондирования. За это время в Сибирском отделении созданы высокоэффективные технические и программные средства, которые получили достаточно широкую известность в СНГ и других странах.

В Республике Беларусь на базе Объединенного Института Проблем Информатики Национальной Академии Наук Беларуси сформирован и развивается центр по обработке данных дистанционного зондирования Земли.

Наблюдения за погодой и составление прогнозов также является важным применением многозональных спутниковых изображений. Так, на рис. 5.9 демонстрируется изображение урагана, полученное спутником США с помощью датчиков, работающих в видимом и инфракрасном диапазонах, на котором четко виден центр урагана.

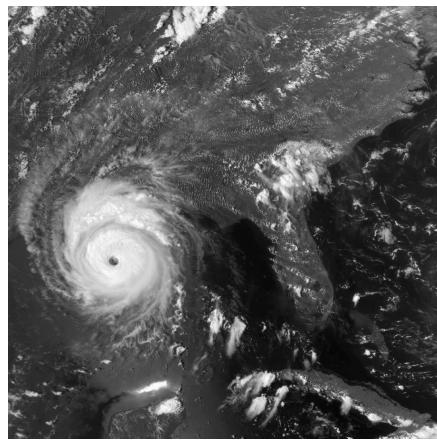


Рис. 5.9. Мультиспектральное изображение урагана, полученное со спутника

*Системы контроля качества выпускаемой продукции.* Одним из примеров является контроль интегральных микросхем [1,5]. Существующие системы контроля дефектов могут быть подразделены на два класса: в одном из них производится проверка правил проектирования путем анализа конфигурации проводников и других объектов интегральной микросхемы, в другом проверка производится путем сравнения с изображением заведомо исправной микросхемы и контрольного изо-

бражения. Кроме этого, выполняется контроль наличия всех компонентов на плате. На рис. 5.10, *а* черный квадрат в правой верхней части изображения демонстрирует отсутствие микросхемы.

Кроме этого, компьютерная обработка изображений используется при изготовлении лекарственных средств (рис. 5.10, *б*), определения уровня заполнения бутылок (рис. 5.10, *в*) и т. д.

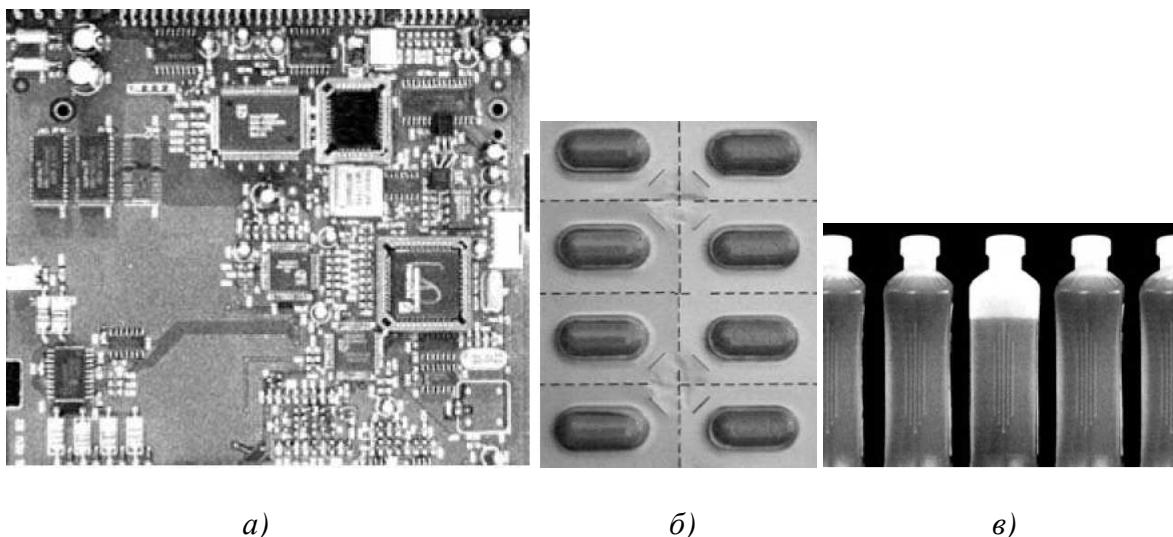


Рис. 5.10. Примеры изображений продукции, контролируемой с помощью методов обработки изображений: *а*) плата контроллера дисковода CD-ROM; *б*) упаковка таблеток; *в*) бутылки с жидкостью

*Анализ дактилоскопических изображений.* В криминалистике одним из часто используемых способов установления личности является идентификация по отпечаткам пальцев [5]. На рис. 5.11 приведен пример изображения отпечатка большого пальца. В связи с этим представляется важной задача автоматизированного сличения изображений отпечатков пальцев.



Рис. 5.11. Пример изображения отпечатка пальца

Сличение отпечатков проводится сравнением их характеристик: глобальных и локальных особенностей. Глобальными особенностями являются такие конфигурации узора, как дельта, спираль, петля. Обнаружение глобальных особенностей проще производится не по исходному изображению, а по полю направлений. Каждому типу особенности соответствует определенная конфигурация поля направлений вблизи особой точки.

Полученное в результате вычислений поле направлений в силу наличия шума на изображении и неидеальности полос папиллярного узора неустойчиво, что проявляется в виде ряби на изображении поля. Поэтому требуется его фильтрация. Использование поля направлений вместо исходных дактилоскопических изображений позволяет в десятки раз повысить эффективность распознавания отпечатков.

Кроме этого, цифровая обработка используется при проверке подлинности денежных по их изображением банковскими структурами, в том числе и в Республике Беларусь применяются специальные приборы, изготовленные НПП «Регула».

*Системы обработки изображений лица человека.* Один из самых первых методов – это анализ геометрических характеристик лица. Изначально применялся в криминалистике и был там детально разработан. Потом появились компьютерные реализации этого метода. Суть его заключается в выделении набора ключевых точек (или областей) лица и последующем выделении набора признаков. Каждый признак является либо расстоянием между ключевыми точками, либо отношением таких расстояний. Наборы наиболее информативных признаков выделяются экспериментально.

Ключевыми точками могут быть уголки глаз, губ, кончик носа, центр глаза и т. п. (рис. 5.12). В качестве ключевых областей могут быть прямоугольные области, включающие в себя: глаза, нос, рот. Так, например, ширину лица для мужчин легче и надежнее измерять на уровне глаз (на большинстве снимков уши открыты, волосы не мешают), а для женщин на уровне линии смыкания губ (женщины не носят бороды и бакенбарды).

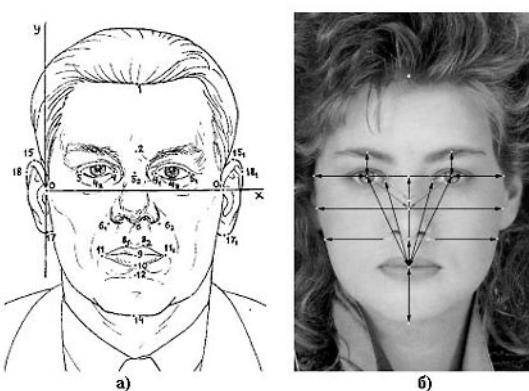


Рис. 5.12. Идентификационные точки и расстояния:  
 а) используемые при криминалистической фотоэкспертизе; б) наиболее часто применяемые при построении автоматизированных систем идентификации

В процессе распознавания сравниваются признаки неизвестного лица, с признаками, хранящимися в базе. Задача нахождения ключевых точек приближается к трудоемкости непосредственно распознавания, и правильное нахождение ключевых точек на изображении во многом определяет успех распознавания [1]:

- между подбородком и линией глаз;
- между серединой линии смыкания губ и подбородком;
- между центром сетчатки глаза и линией волосяного покрова;
- между центром сетчатки глаза и центром носа;
- между кончиком подбородка и серединой лица;
- между внутренним уголком глаза и центром лица;
- между центром сетчатки глаза и центром брови;
- от центра брови до центра лица;
- между центром зрачка и серединой линии смыкания губ;
- между внутренним уголком глаза и уголком рта;
- между серединой линии смыкания губ и нижней точкой носа;
- ширины лица на уровне линии глаз;
- ширины лица на уровне нижней точки носа;
- ширины лица посередине между линией глаз и нижней точкой носа;
- максимальной ширины носа.

Поэтому изображение лица человека должно быть без помех, мешающих процессу поиска ключевых точек. К таким помехам относят очки, бороды, украшения, элементы прически и макияжа. Освещение желательно равномерное и одинаковое для всех изображений. Кроме того, изображение лица должно иметь фронтальный ракурс, возможно с небольшими отклонениями. Выражение лица должно быть нейтральным. Это связано с тем, что в большинстве методов нет модели учета таких изменений. Таким образом, данный метод предъявляет строгие требования к условиям съемки, нуждается в надежном механизме нахождения ключевых точек для общего случая. Кроме того, требуется применение более совершенных методов классификации или построения модели изменений.

В общем случае этот метод не самый оптимальный, однако для некоторых специфических задач перспективен. К таким задачам можно отнести документный контроль, когда требуется сравнить изображение лица, полученного в текущий момент с фотографией в документе.

*Системы получения и контроля медицинских изображений.* Одним из примеров является задача анализа препаратов крови, которую можно представить как совокупность ряда подзадач. В качестве примера рассмотрим одну из них: расчет концентрации эритроцитов.

Основные проблемы, возникающие при анализе указанного класса изображений, – это повышение качества введенного изображения, отделение частиц заданного класса от всей совокупности микрочастиц изображения, подсчет их геометрических характеристик.

Обработка изображения препарата крови, содержащего эритроциты, представляет собой совокупность следующих задач: удаление импульсного и аддитивного шумов, пороговая обработка, медианная фильтрация, построение гистограммы площадей.

Другим примером служит анализ изображения глазного дна. Диагностическое изображение глазного дна представляет собой сеть каналов (вен и артерий), расположенных на фоне с плавно меняющейся яркостью, значение которой существенно больше яркости сосудов. Аномальными проявлениями на изображении глазного дна служат сгустки черных и белых пятен, сильная извилистость сосудов, обильное ветвление капилляров. Наименее заметным и в то же время важным критерием патологии глазного дна является состояние сосудистой системы. К диагностике широкого спектра заболеваний имеют отношение такие характеристики, как ширина вен и артерий, отношение их ширины, изменение ширины по длине сосуда, динамика изменения направления сосудов, углы ответвления сосудов.

Автоматическая трассировка отдельного сосуда производится с помощью задания пользователем начальной и конечной точки сканирования и с учетом найденного направления сосуда в текущей точке. За ширину сосуда принимается число ненулевых отсчетов, лежащих на перпендикуляре к направлению сосуда. После определения ширины исходная точка смещается на некоторый заданный шаг трассировки в направлении, которое выбирается из вычисленных ранее как наиболее близкое к прямому направлению к конечной точке. В результате трассировки формируется последовательность параметров, характеризующих состояние сосудистой системы и используемых при оценке патологии.

Кроме этого используется цифровая обработка изображений, полученных с помощью метода ядерно-магнитного резонанса (рис. 5.13, *a*), ультразвуковых изображений (рис. 5.13, *б*) и т. д.



Рис. 5.13. Примеры медицинских изображений:

- а)* изображение человеческого колена, полученное с помощью ядерно-магнитного резонанса; *б)* ультразвуковое изображение щитовидной железы

### **5.1.3. Структура системы обработки изображений**

Обработка изображений относится к таким задачам обработки данных, в которых объем вычислений прямо пропорционален квадрату размерности изображений и числу анализируемых спектральных каналов этого изображения [1]. В связи с этим для обеспечения необходимой вычислительной мощности в прошлом методы обработки изображений были ориентированы на применение больших ЭВМ или мини-ЭВМ со специализированными системами обработки. Однако в настоящее время в связи с постоянным повышением мощности персональных компьютеров появилась перспектива разработки новых методов и алгоритмов обработки изображений для внедрения их в прикладные системы обработки. Такие системы могут использоваться, например, для обработки полутоновых и цветных аэрофотоснимков, стереоизображений, картографической и символьной информации, изображений Земли в различных спектрах, рентгеновских снимков и т. д. Назначение систем обработки изображений может быть различным – распознавание и локализация объектов определенного класса, ввод и распознавание текстовой информации, содержательная интерпретация карт, местности и т. п.

Архитектура систем обработки изображений зависит от следующих факторов [1]:

- цели разработки системы;
- набора технических средств;
- области применения системы;
- квалификации пользователя;
- класса задач;
- состава решаемых задач;
- базовой операционной системы и языка программирования;
- принципа организации вычислительного процесса;
- структур данных для представления изображений.

Невозможно построить универсальную систему, одинаково эффективно решающую различные классы задач.

В русскоязычной литературе принята следующая классификация систем обработки визуальной информации.

1. Автоматизированные системы обработки изображений (АСОИз), которые обрабатывают полутоновые или многозональные изображения.
2. Системы анализа изображений (САИ), решающие задачи распознавания, классификации и содержательной интерпретации.
3. Системы обработки графических изображений, предназначенные для обработки документов, карт, чертежей.

#### 4. Системы машинной графики, служащие для синтеза изображений.

Однако такое деление достаточно условно. Особенно заметно стирание граней между ними в результате их интеллектуализации. В интеллектуальных системах используются знания специалистов-экспертов по обработке, анализу и интерпретации изображений.

Появление интеллектуальных (экспертных) систем стало новым шагом в развитии АСОИз. Степень интеллектуализации может быть различна в зависимости от уровня представления и использования знаний специалистов по обработке изображений и специалистов предметных областей. Она колеблется от возможности применения набора операций обработки (процедурных знаний) до поддержки специальных баз знаний в экспертных системах.

Основными задачами, решаемыми системами обработки изображений, являются [1]:

- улучшение качества изображений;
- корректировка изображений и реставрация;
- координатная привязка, компоновка и совмещение изображений;
- выделение и измерение признаков;
- поиск объектов на изображении;
- выделение и измерение параметров объектов;
- распознавание и классификация изображений и объектов на них;
- анализ сцен и интерпретация изображений;
- кодирование изображений;
- хранение и поиск изображений в информационных системах.

В рамках обработки изображений рассматриваются следующие технологические элементы:

- транспортирование носителей графической информации в зону восприятия;
- восприятие графической информации (символов, рисунков и т. п.);
- преобразование исходного изображения в цифровую форму;
- предварительная обработка (фильтрация, кадрирование);
- препарирование изображений (выделение контура, выделение скелета, сегментация, аппроксимация);
- выделение признаков;
- анализ изображения;
- идентификация или распознавание двумерных объектов;
- распознавание трехмерных объектов.

Диаграмма обобщенной системы обработки изображений приведена на рис. 5.14 [1].



Рис. 5.14. Диаграмма обобщенной системы обработки изображений

## 5.2. Элементы зрительного восприятия и цифрового представления

### 5.2.1. Строение глаза человека и формирование изображения в глазу

Методы и алгоритмы цифровой обработки изображений основываются на математических и вероятностных формулировок, человеческий анализ до сих пор играет основную роль при выборе того или иного метода обработки, и этот выбор часто совершается на основе субъективного визуального оценивания. Поэтому необходимо понимать хотя бы основные аспекты формирования изображения в глазу.

На рис. 5.15 показан поперечный разрез человеческого глазного яблока [5].

Глаз имеет почти сферическую форму со средним диаметром около 20 мм. Глаз окружен тремя оболочками: *роговица* со *склерой* образуют внешнюю оболочку, под которой последовательно расположены сосудистая оболочка (*хороидия*) и нейроглиальная оболочка (*сетчатка*). Роговица – это плотная прозрачная ткань, закрывающая переднюю поверхность глаза. Продолжением ее является скlera – непрозрачная оболочка, закрывающая остальную часть оптической сферы глаза.

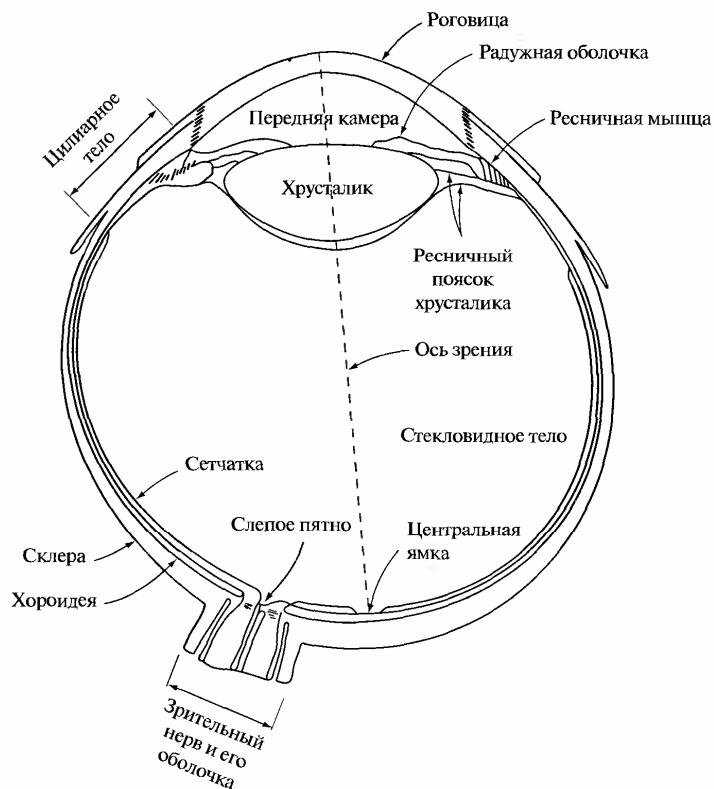


Рис. 5.15. Упрощенная схема глаза человека в разрезе

Хороидия расположена непосредственно под склерой. В этой оболочке проходит сеть кровеносных сосудов, обеспечивающих питание глаза. Даже незначительное повреждение хороидии, часто кажущееся неопасным, может привести к серьезному нарушению зрения из-за воспаления, препятствующего нормальному кровотоку. Поверхностный слой хороидии сильно пигментирован, что снижает интенсивность попадающего через склеру внешнего света, мешающего восприятию из-за его отражения и рассеяния внутри оптической сферы. Передняя область (зубчатая линия) хороидии непосредственно вплетена в *цилиарное тело* и *радужную оболочку* (или *радужку*). Отверстие в центре радужной оболочки (зрачок) может сужаться или расширяться, регулируя тем самым количество попадающего через роговицу света. Диаметр зрачка может

изменяться в пределах от 2 до 8 мм. От цвета пигmenta на передней поверхности радужки зависит цвет глаза человека, а пигмент на задней поверхности радужки имеет черный цвет, что также снижает внутреннее рассеяние света.

*Хрусталик* (*хрусталиковая линза*), состоящий из наружной капсулы и внутрихрусталикового вещества, закреплен внутри глаза с помощью передней и задней порций волокон ресничного пояска хрусталика, которые проходят между отростками цилиарного тела и вплетаются в зубчатую линию хорOIDeи. Капсула и внутрихрусталиковое вещество состоят из коллагеновых волокон и содержат от 60 % до 70 % воды, около 6 % жиров и больше белков, чем любые другие ткани глаза. Внутрихрусталиковое вещество имеет слабо-желтую пигментацию, которая с возрастом усиливается. Ускоренное помутнение вещества хрусталика, связанное с нарушением его питания, приводит к заболеванию, называемому *катарактой*, при котором ухудшается цветовое восприятие и острота зрения. В видимом диапазоне спектра хрусталик поглощает около 8 % света, и практически не пропускает более коротковолновое излучение. Свет инфракрасного и ультрафиолетового диапазонов существенно поглощается белком хрусталика, и при высокой интенсивности может привести к необратимому нарушению зрения.

Самая внутренняя оболочка глаза – *сетчатка* – выстилает изнутри задний отдел глаза. При правильной оптической фокусировке глаза свет от наружного объекта проецируется в виде изображения на сетчатку. Зрительное восприятие образов становится возможным благодаря распределению дискретных светочувствительных клеток (*рецепторов*) по внутренней поверхности сетчатки. Существуют рецепторы двух видов – *колбочки* и *палочки*. В глазу насчитывается от 6 до 7 миллионов колбочек, которые обладают высокой чувствительностью к спектральным составляющим света и располагаются преимущественно в центральной области сетчатки, называемой *желтым пятном*. В центре желтого пятна имеется так называемая *центральная ямка* – область наибольшей остроты зрения. Человек различает мелкие детали изображения в основном благодаря колбочкам, поскольку каждая из них соединена с отдельным нервным окончанием. Наружные мышцы глаза обеспечивают вращение глазного яблока так, чтобы изображение интересующего объекта попадало в область желтого пятна. Колбочки обеспечивают *фотопическое* зрение, или зрение в ярком свете.

Количество палочек в глазу намного больше: по поверхности сетчатки их распределено от 75 до 150 миллионов. Большая, чем у колбочек, область распределения и тот факт, что к одному нервному окончанию присоединено сразу несколько палочек (в среднем около 10), уменьшают возможности раз-

личения деталей с помощью этих рецепторов. Палочки позволяют сформировать общую картину всего поля зрения. Они наиболее чувствительны при низких уровнях освещенности и не участвуют в обеспечении функций цветного зрения. Например, предметы, имеющие яркую окраску при дневном свете, при сумеречном освещении выглядят как лишенные цветов образы, поскольку возбуждаются только палочки. Это явление известно как *скотопическое* (или сумеречное) зрение.

Рисунок 5.16 [5] иллюстрирует зависимость плотности распределения палочек и колбочек по сетчатке в зависимости от величины угла между зрительной осью и линией, проведенной из центра хрусталика до сетчатки. Изображено горизонтальное сечение правого глаза в месте выхода зрительного нерва. Отсутствие рецепторов в этой области приводит к появлению так называемого *слепого пятна* (рис. 5.16). В остальной области сетчатки распределение рецепторов центрально симметрично относительно центра желтого пятна. Из рисунка 5.16 видно, что максимальная плотность колбочек наблюдается в центре сетчатки (в центральной ямке), а плотность палочек возрастает от этой точки приблизительно до угла в 20°, после чего плавно снижается вплоть до периферии сетчатки.

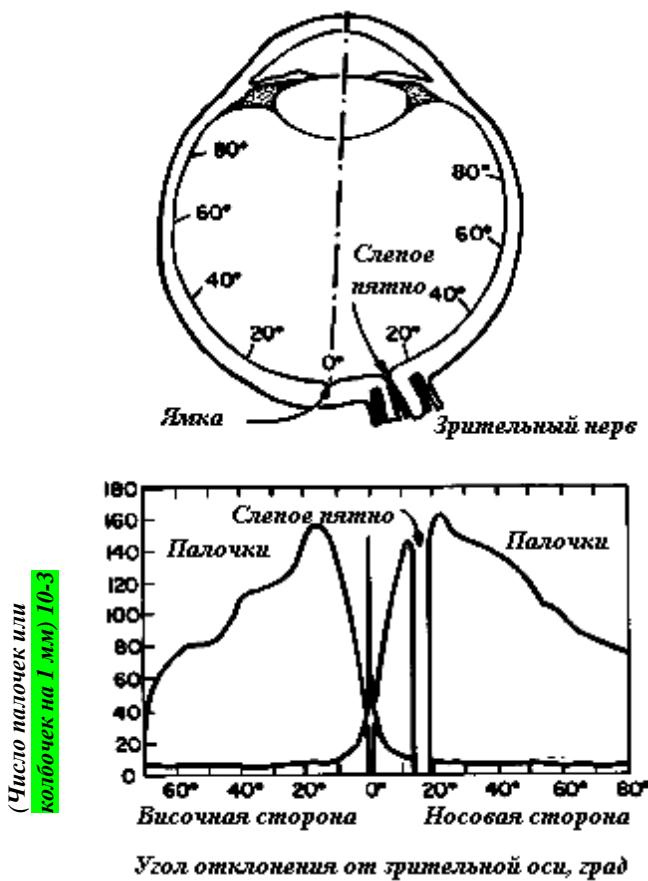


Рис. 5.16. Распределение палочек и колбочек по сетчатке

Центральная ямка представляет собой углубление круглой формы в сетчатке, с диаметром около 1,5 мм. В контексте дальнейшего обсуждения более естественно говорить о прямоугольных массивах чувствительных элементов. В несколько вольной интерпретации можно рассматривать центральную ямку как квадратный массив чувствительных элементов на площади  $1,5 \times 1,5$  мм. Плотность колбочек в этой области сетчатки приблизительно равна 150 тыс. на  $1\text{ mm}^2$ , следовательно, общее количество колбочек в области наибольшей остроты зрения составляет около 337 тыс. элементов. Если рассуждать только в терминах разрешающей способности, то широко применяемые в современной технике светочувствительные матрицы среднего разрешения на основе приборов с зарядовой связью (ПЗС) содержат такое же количество чувствительных элементов при площади кристалла не более  $5 \times 5$  мм.

Известно, что основное отличие хрусталика глаза от жесткой оптической линзы состоит в возможности изменения оптической силы хрусталика глаза за счет некоторой вариации его формы (кривизны). Как видно из рисунка 5.15, радиус кривизны передней поверхности капсулы хрусталика больше, чем задней. Изменение формы хрусталика осуществляется путем натяжения или ослабления передней и задней порций волокон ресничного пояска. Для фокусировки зрения на удаленном предмете ресничная мышца расслабляется, хроидия сжимается, натягивая при этом волокна ресничного пояска, в результате чего хрусталик приобретает уплощенную форму. Наоборот, для фокусировки на близкорасположенном предмете ресничная мышца сокращается, что приводит к ослаблению натяжения волокон ресничного пояска и округлению хрусталика, т. е. к увеличению его преломляющей способности.

При изменении преломляющей способности хрусталика с минимального значения до максимального, его *фокусное расстояние* изменяется соответственно от 17 мм (при фокусировке на дальние предметы) до 14 мм (при фокусировке на близкие предметы). При рассмотрении предметов на удалении более 3 м преломляющая способность хрусталика минимальна, при разглядывании близкорасположенных предметов преломляющая способность хрусталика максимальна. На основе этих данных легко вычислить размеры изображения некоторого объекта на сетчатке. Например, на рис. 5.17 [5] наблюдатель видит дерево высотой 15 м с расстояния 100 м. Обозначая  $H$  высоту изображения дерева на сетчатке (в мм), формируется пропорция  $15/100 = h/17$ , откуда  $h = 2,55$  мм. Проецирующееся на сетчатку изображение воспринимается главным образом областью желтого пятна. Расположенные в ней рецепторы возбуждаются в соответствии с интенсивностью падающего света, что приводит к преобразованию энергии светового излучения в электрические нервные импульсы, которые в конечном счете декодируются в мозге человека.

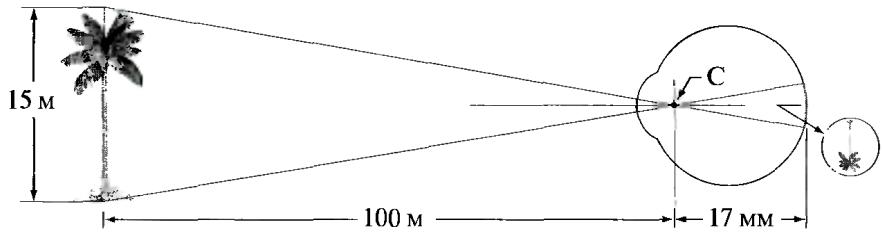


Рис. 5.17. Схематическое изображение глаза, наблюдающего дерево  
(точка C – оптический центр хрусталика)

### 5.2.2. Зрительные явления

Согласно [5], к основным зрительным явлениям относят следующие.

**Контрастная чувствительность.** Поскольку цифровые изображения воспроизводятся как дискретное множество элементов с различной яркостью, способность глаза различать отличающиеся уровни яркости необходимо учитывать при представлении результатов обработки изображений. Зрительная система человека способна адаптироваться к огромному, порядка  $10^{10}$ , диапазону значений яркости – от порога чувствительности скотопического зрения до предела ослепляющего блеска. Известно, что *субъективная яркость* является логарифмической функцией от физической яркости света, попадающего в глаз. На рис. 5.18 показана зависимость субъективной яркости от истинной яркости. Длинная сплошная кривая представляет диапазон яркостей, в котором способна адаптироваться зрительная система. При использовании одного фотопического зрения этот диапазон составляет около  $10^6$ . Постепенный переход от скотопического зрения к фотопическому происходит в диапазоне приблизительно от 0,003 до  $0,3 \text{ кд}/\text{м}^2$  (т. е. от  $-2,5$  до  $-0,5$  по логарифмической шкале), что показано в виде двух ветвей кривой адаптации в этом диапазоне яркостей.

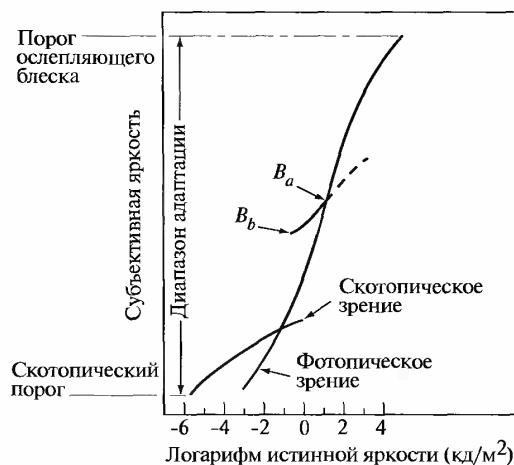


Рис. 5.18. Диапазон субъективно воспринимаемой яркости  
и конкретный уровень адаптации

Однако зрительная система не способна работать во всем этом диапазоне *одновременно*. Вместо этого она охватывает такой большой диапазон за счет изменения общей чувствительности. Это явление известно как *яркостная адаптация*. Общий диапазон одновременно различаемых уровней яркости относительно мал по сравнению со всем диапазоном адаптации. Для любого данного набора внешних условий текущий уровень чувствительности зрительной системы, называемый *уровнем яркостной адаптации*, соответствует некоторой яркости, например, точке  $B_a$  на рис. 5.18. Короткая кривая, пересекающая основной график, представляет диапазон субъективной яркости, которую способен воспринимать глаз при адаптации к указанному уровню. Этот диапазон достаточно ограничен: все уровни яркости ниже  $B_b$  субъективно воспринимаются зрением как черное и, значит, неразличимы. Верхняя (пунктирная) часть этой кривой реально не ограничена, но теряет смысл при большой длине, поскольку при повышении яркости просто повышается уровень адаптации  $B_a$ .

Способность зрения различать *изменения* яркости при данном уровне адаптации также представляет значительный интерес. Классический эксперимент для определения способности зрительной системы человека различать разные уровни яркости состоит в том, что испытуемый смотрит на плоский равномерно освещенный экран достаточно больших размеров, такой, что он занимает все поле зрения. Как правило, это рассеиватель из матового стекла, освещаемый со стороны, противоположной наблюдателю, световым источником, яркость  $I$  которого можно регулировать. На это равномерное поле накладывается добавочная яркость  $\Delta I$  в форме кратковременной вспышки в области круглой формы, расположенной в центре равномерно освещенного экрана.

Если приращение  $\Delta I$  недостаточно велико (неразличимо), испытуемый говорит «нет», указывая тем самым на отсутствие видимых изменений. По мере увеличения  $\Delta I$  в какой-то момент он начнет говорить «да», подтверждая тем самым восприятие изменений яркости.

Наконец, при достаточно большом значении  $\Delta I$  испытуемый станет говорить «да» на каждую вспышку. Величина  $\Delta I_c / I$ , где  $\Delta I_c$  – величина приращения яркости, различимая в 50 % случаев на фоне яркости  $I$ , называется *отношением Вебера* [5]. Малое значение  $\Delta I_c / I$  означает, что различаются очень малые относительные изменения яркости, т. е. имеет место «высокая» контрастная чувствительность. Наоборот, большое значение  $\Delta I_c / I$  означает, что требуется большое относительное изменение яркости, чтобы его заметить; это говорит о «низкой» контрастной чувствительности.

Низкая контрастная чувствительность (т. е. большое отношение Вебера) наблюдается при малых уровнях яркости, и контрастная чувствительность заметно возрастает (т. е. отношение Вебера уменьшается) при увеличении фоновой яркости. При малых уровнях яркости зрение осуществляется благодаря действию палочек, тогда как при больших уровнях яркости (которым соответствует высокая контрастная чувствительность) зрительные функции выполняют колбочки сетчатки.

Если поддерживать фоновую яркость постоянной, а яркость добавочного источника варьировать не вспышками, а ступенчатым изменением яркости от неотличимого до заметного всегда, то типичный наблюдатель способен различить всего 10 – 20 различающихся ступеней яркости. Грубо говоря, этот результат относится к числу различных уровней яркости, которые человек способен различить в произвольной точке монохромного изображения. Это не означает, что изображение может быть представлено таким небольшим числом градаций яркости, так как по мере движения взгляда по изображению меняется среднее значение яркости фона, что позволяет обнаруживать различные множества относительных изменений яркости для каждого нового уровня адаптации. Конечным следствием является способность глаза различать яркости в намного более широком общем диапазоне. В действительности глаз способен обнаруживать нежелательные ложные контуры в монохромных изображениях, общий диапазон яркостей которых представляется значительно большим количеством, чем 20 уровней.

Известны два явления, ясно доказывающие, что воспринимаемая яркость не является простой функцией истинной яркости. Первое основывается на том факте, что вблизи границ соседних областей с отличающимися, но постоянными яркостями зрение человека склонно «подчеркивать» яркостные перепады, как бы добавляя несуществующие выбросы яркости, что убедительно демонстрирует пример на рисунке 5.19. Хотя яркость каждой из полос постоянна, мы, кроме действительно ступенчатого изменения яркости, видим характерные выбросы вбли-

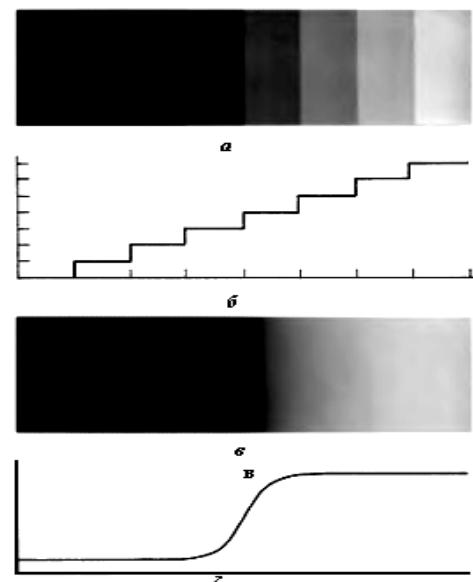


Рис. 5.19. Полосы Маха: *а* – ступенчатый клин; *б* – распределение интенсивности клина; *в* – перепад интенсивности; *г* – распределение интенсивности для картины «*в*»

зи краев полос. Эти полосы с кажущимися изменениями яркости на краях называются *полосами Маха* в честь Эрнста Маха, впервые описавшего этот феномен в 1865 г.

Полосы Маха можно объяснить исходя из пространственно-частотной характеристики глаза. Глаз обладает меньшей чувствительностью в области низких, и высоких пространственных частот по сравнению с чувствительностью на средних частотах. Отсюда можно сделать вывод, что при создании систем обработки изображений можно в некоторой степени пожертвовать верностью воспроизведения контуров, так как глаз не очень чувствителен к высокочастотным перепадам светлоты.

*Одновременный контраст.* Явление одновременного контраста иллюстрируется фотографией, приведенной на рис. 5.20. В действительности все маленькие квадраты имеют одинаковую яркость, но из-за различной яркости фона кажется, что их яркость разная. Цветовой фон пятна света зависит также от цвета окружения. Белое пятно на черном квадрате кажется желтоватым, если вся фигура окружена синим фоном.

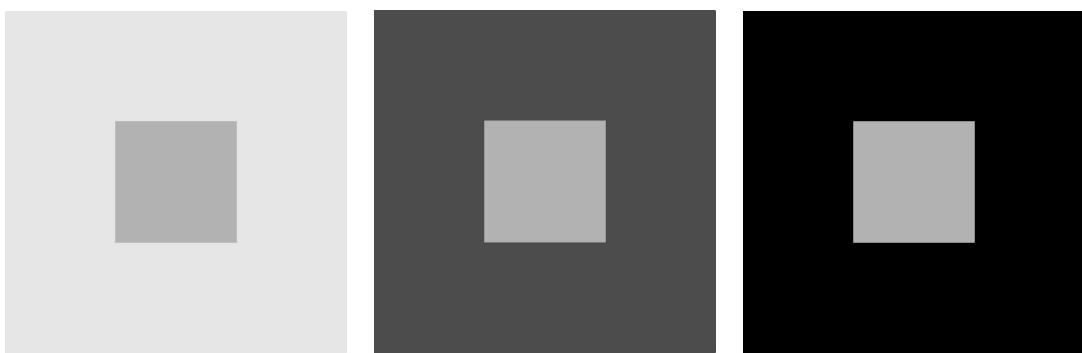


Рис. 5.20. Пример одновременного контраста

Другими примерами феноменов человеческого зрительного восприятия являются оптические иллюзии, в которых глаз восполняет не существующую информацию или ошибочно воспринимает геометрические свойства объектов. Некоторые примеры оптических иллюзий изображены на рис. 5.21 [5]. Так, на рис. 5.21, а ясно видны очертания квадрата, вопреки тому факту, что на изображении отсутствуют линии, определяющие такую фигуру. Аналогичный эффект, на этот раз в виде круга, виден на рис. 5.21, б. Два горизонтальных отрезка на рис. 5.21, в имеют одинаковую длину, но один кажется короче другого. Все проведенные под углом  $45^\circ$  линии на рис. 5.21, г параллельны и расположены на одинаковых расстояниях друг от друга. Однако штриховка создает иллюзию, что эти линии далеки от параллельности.

Оптические иллюзии относятся к числу не вполне понятных характеристик зрительной системы человека.

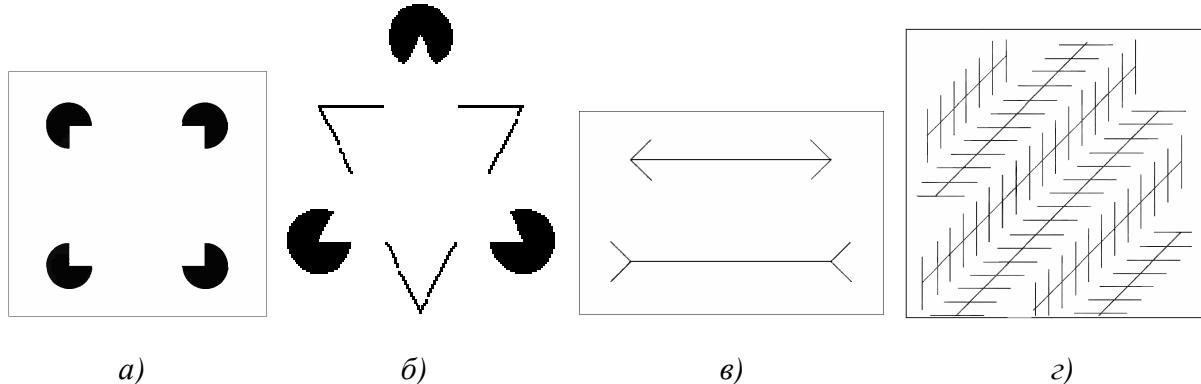


Рис. 5.21. Примеры некоторых оптических иллюзий

### 5.2.3. Считывание и регистрация изображения

Изображение на компьютере это двумерное отображение наблюдаемой сцены (как правило, двух- или трехмерной), возникающее как результат регистрации лучистой энергии, исходящей из наблюдаемой сцены, с помощью некоторого устройства – сенсора (или совокупности сенсоров одновременно).

Используются три основные схемы размещения чувствительных элементов (сенсоров), которые используются для преобразования энергии «освещения» в цифровое изображение (рис. 5.22) [5]: одиночный чувствительный элемент, линейка сенсоров и матрица чувствительных элементов. Идея преобразования заключается в том, что: падающая энергия преобразуется в напряжение благодаря сочетанию материала, обладающего чувствительностью к интересующему виду излучения, и приложенной к нему электрической энергии. В ответ на энергию внешнего излучения такой чувствительный элемент выдает сигнал выходного напряжения, который затем преобразуется в цифровую форму.

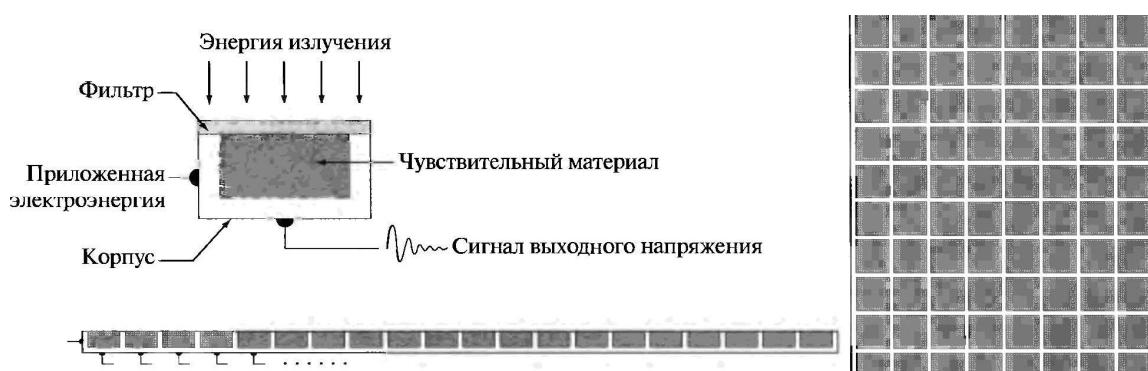


Рис. 5.22. Примеры размещения чувствительных элементов

Наиболее известным одиночным сенсором является фотодиод, напряжение выходного сигнала которого пропорционально освещенности. Установка фильтра перед чувствительным элементом обеспечивает избирательность сенсора. Например, если установить перед сенсором зеленый стеклянный фильтр, то выходной сигнал будет выше для зеленого участка видимого спектра, чем для всех остальных.

Более шире, чем одиночный сенсор, для считывания изображений используется одномерный массив сенсоров (лнейка). Такая линейка обеспечивает одновременную регистрацию элементов изображения в одном направлении, а перемещение всей линейки в перпендикулярном направлении позволяет получить все строки изображения.

Подобная конструкция применяется в большинстве планшетных сканеров. Изготавливаются линейки, состоящие из 4000 и более расположенных в ряд чувствительных элементов. Расположение сенсоров в ряд широко используется при аэрофотосъемке, когда система регистрации устанавливается на самолете, летящем с постоянной скоростью и на неизменной высоте над интересующим районом. Одномерные линейки сенсоров, чувствительных к излучениям в различных участках электромагнитного спектра, располагаются перпендикулярно направлению полета. В каждый момент времени линейка сенсоров регистрирует одну строку изображения, а движение всей системы в перпендикулярном направлении позволяет заполнить второе измерение двумерного изображения. Для проекции сканируемой области на линейку сенсоров применяются линзы или другие фокусирующие устройства.

Многочисленные электромагнитные и некоторые ультразвуковые устройства ввода данных сегодняшних систем обработки изображений используют расположение отдельных сенсоров в форме двумерного массива (матрицы). Такая же конструкция находится внутри подавляющего числа цифровых камер, в которых типичным чувствительным элементом является матрица на основе приборов с зарядовой связью (ПЗС), которые выпускаются в виде монолитной конструкции, объединяющей  $10\ 000 \times 10\ 000$  элементов (и более) с широким диапазоном чувствительных свойств. ПЗС-матрицы широко используются в цифровых фото- и видеокамерах, а также других светочувствительных приборах. Ответная реакция каждого элемента пропорциональна интегралу световой энергии, попадающей на поверхность этого элемента за время экспозиции; это свойство используется в астрономии и других приложениях, где требуется получать изображения с низким уровнем шума. Уменьшение шума достигается за счет того, что чувствительным элементам дают возможность интегрировать принимаемый световой сигнал в течение минут или даже часов.

Главное достоинство такого подхода в том, что можно считать сразу все изображение, если сфокусировать на поверхности матрицы отвечающий ему пространственный поток лучистой энергии.

Изображение представляет собой двумерную функцию, обозначим вида  $F$ . Значение функции в точке с пространственными координатами  $(ij)$  является положительной скалярной величиной, физический смысл которой определяется источником изображения. Если изображение генерируется в результате физического процесса, его значения пропорциональны энергии излучения некоторого физического источника, вследствие чего функция  $F$  должна быть ненулевой и конечной.

Функцию  $F$  можно охарактеризовать величиной светового потока, который падает на наблюдаемую сцену от источника (освещенность) и относительной долей светового потока, отраженного от объектов этой сцены (коэффициент отражения). Произведение этих функций дает функцию изображения.

Коэффициент отражения может меняться в пределах от 0 (полное поглощение) до 1 (полное отражение). Освещенность зависит от источника освещения, тогда как функция отражения определяется свойствами объектов изображаемой сцены. Приведенные правила в равной мере применимы также и к изображениям, сформированным в проходящем освещении (сквозь наблюдаемый объект), как, например, при рентгене грудной клетки. В подобном случае используется коэффициент пропускания вместо коэффициента отражения, но пределы изменения будут те же.

Можно привести примеры, что в ясный день солнце создает на земной поверхности освещенность  $90\,000 \text{ лм}/\text{м}^2$  и выше, а в пасмурную погоду эта величина падает до  $10\,000 \text{ лм}/\text{м}^2$ . Безоблачной ночью в полнолуние освещенность земной поверхности составляет около  $0,1 \text{ лм}/\text{м}^2$ . В типичных служебных помещениях поддерживается уровень освещенности порядка  $1000 \text{ лм}/\text{м}^2$ . Типичные значения коэффициента отражения составляют: 0,01 для черного бархата; 0,65 для нержавеющей стали; 0,80 для поверхности стены, окрашенной в ровный белый цвет; 0,9 для посеребренной металлической поверхности; и 0,93 для снега.

#### **5.2.4. Общий подход к математическому описанию цифрового изображения**

С формальной точки зрения цифровое изображение представляется в виде матрицы действительных чисел  $F$  с элементами  $f_{ij}$  (каждому элементу которой ставится в соответствие некоторое число – код яркости, обычно от 0 до 255), где  $ij$  – целочисленные координаты [1,5]. Причем принято за начало

координат принимать левый верхний угол изображения, где  $i = 0, j = 0$ . Следующим значением координат вдоль первой строки изображения будет точка ( $i = 0, j = 1$ ). Важно иметь в виду, что обозначение (0,1) используется лишь для указания на второй отсчет в первой строке, и не означает, что это фактические значения физических координат точек дискретизации.

С использованием введенных обозначений можно компактно записать полное цифровое изображение размерами  $M \times N$  в форме матрицы  $F$ :

$$F = \begin{bmatrix} f_{00} & f_{01} & \dots & f_{0,N-1} \\ f_{10} & f_{11} & \dots & f_{1,N-1} \\ \dots & \dots & \dots & \dots \\ f_{M-1,0} & f_{M-1,1} & \dots & f_{M-1,N-1} \end{bmatrix} = \begin{bmatrix} f_{ij} \end{bmatrix}.$$

Каждый элемент этой матрицы называется элементом изображения или пикселим (pixel от английского picture element) [1].

Для выполнения процесса оцифровки изображения необходимо принять решения относительно размера  $M$  и  $N$ , а также числа уровней (градаций) яркости  $L$ , разрешенных для каждого пикселя. Для  $M$  и  $N$  не существует специальных требований помимо того, что они должны быть положительными целочисленными значениями. Однако значение  $L$ , по соображениям удобства построения оборудования для обработки, хранения и дискретизации, обычно выбирают равным целочисленной степени двойки  $L = 2^k$ .

Дискретные уровни яркости, как правило, располагаются с постоянным шагом (т. е. используется равномерное квантование) и принимают целые значения в интервале  $[0, L - 1]$ . Иногда интервал значений яркости называют динамическим диапазоном изображения, а об изображениях, интервал яркостей которых занимает значительную часть всего диапазона уровней серого, говорят как об изображениях с большим динамическим диапазоном. Если заметная доля пикселей обладает таким свойством, изображение имеет высокий контраст. Наоборот, изображение с малым динамическим диапазоном обычно выглядит тусклым, размытым и серым.

Общее количество битов  $b$ , необходимое для хранения цифрового изображения, определяется по формуле:

$$b = M \times N \times K.$$

Главным фактором, определяющим пространственное разрешение изображения является дискретизация. По существу, пространственное разрешение – это размер мельчайших различимых деталей на изображении. Предположим, что построен чертеж, состоящий из вертикальных линий с шириной  $W$ ,

разделенных промежутками также с шириной  $W$ . Парой линий будем называть одну линию с примыкающим с одной стороны промежутком. Таким образом, ширина пары линий составляет  $2W$ , и на единице длины размещается  $1/2W$  таких пар. Широко используемое определение разрешения состоит именно в указании максимального числа различимых пар линий на единицу длины; например, 100 пар линий на миллиметр.

Яркостным (или полутоновым) разрешением, аналогично, называется мельчайшее различимое изменение яркости, но процесс измерения различных уровней яркости является в высшей степени субъективным. Если при дискретизации изображений имеется относительно большая свобода действий при выборе числа отсчетов (т. е. частоты дискретизации), то при выборе числа градаций яркости приходится в значительной степени учитывать особенности аппаратуры; по этим причинам число градаций обычно выбирается равным степени 2. Наиболее частым решением является выбор 8-битного представления (256 градаций яркости), но в некоторых приложениях используется 16 бит, иногда можно встретить системы, в которых квантование уровней яркости изображения проводится с 10 или 12 битами точности, но это скорее исключение, чем правило.

### 5.3. Обработка бинарных изображений

#### 5.3.1. Улучшение качества бинарных изображений

Пусть  $F = f_{ij}$  – растровое изображение, представляющее собой прямоугольную матрицу. Если  $f_{ij} \in \{0, 1\}$  и принимает только два значения, то изображение называется бинарным и состоит только из черных и белых пикселей [1] (рис. 5.23).

Рис. 5.23. Бинарные изображения  
в увеличенном растровом виде  
и в матричной форме



$1$	$1$	$1$	$1$	$1$	$1$
$0$	$1$	$1$	$1$	$1$	$0$
$0$	$0$	$1$	$1$	$0$	$0$
$0$	$1$	$1$	$1$	$1$	$0$
$0$	$0$	$1$	$1$	$0$	$0$
$0$	$0$	$1$	$1$	$0$	$0$

Первым этапом обработки является устранение шумов на изображении для того, чтобы они в дальнейшем не влияли на качество распознавания.

Возникновение шумов на исходном растровом представлении объясняется следующими причинами [1]:

- наличием на считываемом документе следов «грязи»;
- невысоким качеством отдельных элементов документа;
- возникновением погрешностей сканирования;
- неверным выбором порога.

Можно выделить следующие наиболее часто встречающиеся шумы на изображении (рис. 5.24) [1]:

- неоднородность контура объекта;
- изолированные черные пятна небольших размеров;
- изолированные пустоты внутри объектов;
- разрывы объектов;
- слияние нескольких объектов.

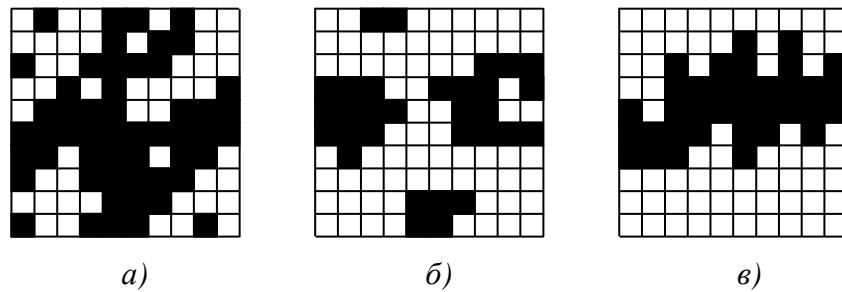


Рис. 5.24. Типы шумов: а) слияние объектов с изолированными пятнами и пустотами, б) разрыв объектов, в) неоднородность контура объекта

Как оценить качество фильтрации изображения? Если бы документ был идеален с точки зрения шумов, то естественным критерием была бы минимизация отклонения отфильтрованного изображения от оригинала. Однако на практике документ не является идеальным изображением. Поэтому выделяются некоторые эвристические критерии, например улучшение качества изображения для его визуального восприятия, для дальнейшего распознавания и т. д.

Существует целый ряд подходов к устранению шумов на изображении. Однако принципиально все они основаны на анализе окрестностей, т.е. являются в основном локальными методами.

Вторым подходом для улучшения качества изображений является логическая фильтрация [1]. Суть ее заключается в том, что значение центрального элемента изменяется в зависимости от количества однотипных с ним элементов в заданной окрестности.

Обычно используют окрестность  $3 \times 3$ . В заданной окрестности может быть  $2^8 = 256$  различных комбинаций черных и белых пикселей. Для каждой из этих комбинаций следует применять некоторое предопределенное решение: изменять цвет данной точки или нет. Естественно, что выбирается некоторый порог. Если, скажем, для черного центрального пикселя число черных пикселей в окрестности меньше порога, то он становится белым. Дополнительная сложность возникает из-за контроля связности объ-

ектов. Величина порога естественным образом влияет на результат обработки. Однако, варьируя значение порога, можно усилить подавление отдельных шумов на изображении и ослабить подавление других. Пример применения логической фильтрации приведен на рис. 5.25.

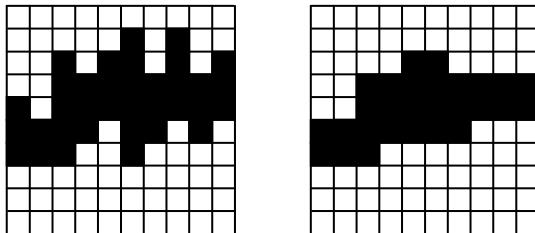


Рис. 5.25. Пример применения логической фильтрации

Алгоритмы, рассмотренные выше, имеют недостаток, заключающийся в достаточно большом времени работы, так как они предполагают последовательный просмотр всего изображения.

Ясно, что часто нет необходимости просматривать все изображение и достаточно исследовать лишь его объекты и их некоторую окрестность. Еще один путь повышения быстродействия – контурный анализ изображения и подавления контурных шумов объектов (при высоком качестве считывающего устройства и считываемого документа шумы на растровом изображении существуют только на контурах).

### 5.3.2. Уточнение бинарных изображений

Задача выделения средних линий (скелетов) изображений является одной из основных задач предварительной обработки. Средние линии позволяют описывать геометрические особенности объектов и удобны для последующей обработки.

Термин «утоньшение» (thinning) является наиболее общим термином для обозначения процесса преобразования линий или других объектов изображения, имеющих ширину в несколько пикселей, в линии единичной ширины [1]. Другими наиболее употребляемыми терминами для обозначения этой операции являются термины «скелетизация», «преобразование средних осей», «преобразование осей симметрии» и др. После своего возникновения около 30 лет назад, решению задачи утоньшения посвящены десятки работ. В первых работах утоньшение использовалось для оптического распознавания символов, кодирования и передачи факсимильных данных. Затем появились приложения для обработки изображений хромосом, отпечатков пальцев, обработки

рентгенограмм и др. В связи с быстрым ростом количества растровых сканирующих систем наблюдается широкое использование уточнения для кодирования чертежей, карт, печатных плат и других графических документов.

К операции уточнения предъявляются, как правило, три основных требования [1]:

- связность объектов изображения и фона должна быть сохранена (connectivity criterion);
- концы средней линии должны располагаться как можно ближе к их истинному положению (endpoint problem);
- центральные линии объектов должны быть выделены достаточно точно (symmetrical shrink).

Как правило, все существующие алгоритмы удовлетворяют этим требованиям. Эти алгоритмы можно условно разбить на несколько групп на основе идеи или метода, в них заложенных.

Самая большая группа алгоритмов основана на идее итеративного удаления внешних слоев или контурных точек объектов до тех пор, пока на изображении останутся только точки скелета. Итеративные алгоритмы используют маску (как правило, размером  $3 \times 3$ ), которая перемещается по всему изображению и в каждый момент времени маска сопоставляется с соответствующим участком изображения, чтобы определить новое значение центрального пикселя. Таким образом, в результате просмотра всего изображения удаляется один (или несколько) из внешних слоев объекта.

Количество просмотров изображения и, как следствие, время работы итеративных алгоритмов зависят от максимальной ширины объектов изображения. Пример приведен на рис. 5.26.

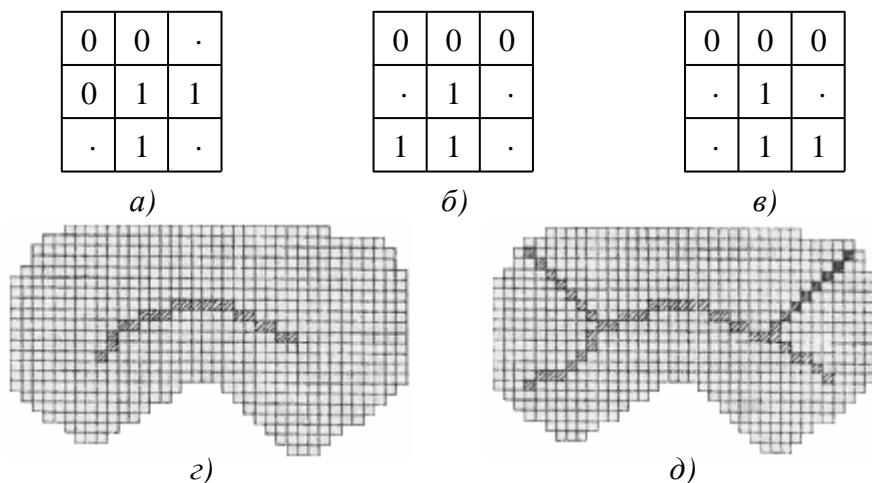


Рис. 5.26. Уточнение бинарного изображения:

а) – в) примеры масок; г) результат уточнения при использовании трех масок; д) результат уточнения при использовании двух масок а) и б)

Алгоритмы данной группы можно разделить на два класса [1]: параллельные и последовательные. В параллельных алгоритмах окно располагается одновременно во всех пикселях изображения, и при его обработке не используются новые (полученные на данной итерации) значения пикселей. При работе последовательных алгоритмов пиксели обрабатываются последовательно, и последнее правило не соблюдается.

Было доказано, что итеративные параллельные алгоритмы для сохранения связности требуют маски размерами  $5 \times 5$ . Итеративные сканирующие последовательные алгоритмы, как правило, не имеют проблем с сохранением связности, так как при обработке текущего пикселя используются результаты обработки соседних с ним пикселей. Однако здесь возникают проблемы с симметричным сжатием объектов. Дальнейшим развитием является использование окон размеров больших, чем  $3 \times 3$ , причем одновременно осуществляется удаление не одного, а  $(k - 1)^2$  пикселей. Однако при  $k > 3$  наблюдается ухудшение качества скелетизированного изображения.

Время уточнения нелинейно изменяется при изменении разрешающей способности сканера. Увеличение разрешения дает линейное возрастание ширины линии и квадратичное возрастание числа пикселей в изображении. Вследствие этого мы будем наблюдать кубическое увеличение времени работы итеративных алгоритмов при увеличении разрешающей способности сканирующего устройства.

### 5.3.3. Выделение контуров бинарных изображений

Алгоритмы выделения контуров можно условно разбить на две группы: отслеживающие и сканирующие [1]. Отслеживающие алгоритмы основаны на том, что, на изображении отыскивается объект (первая встретившаяся точка объекта) и контур объекта отслеживается и векторизуется. Достоинством данных алгоритмов является их простота, к недостаткам можно отнести их последовательную реализацию и некоторую сложность при поиске и обработке внутренних контуров. Пример отслеживающего алгоритма – «алгоритма жука» – приведен на рис. 5.27. Жук начинает движение с белой области по направлению к черной. Как только он попадает на

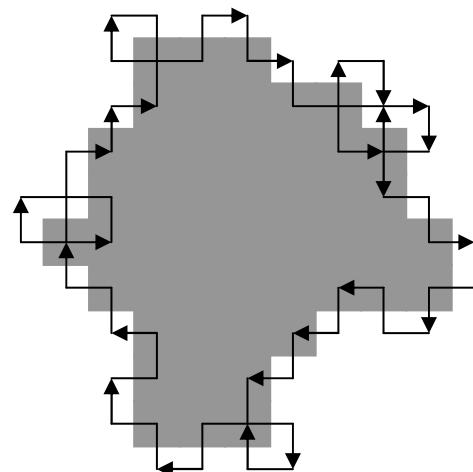


Рис. 5.27. Пример работы «алгоритма жука»

черный элемент, он поворачивает налево и переходит к следующему элементу. Если этот элемент белый, то жук поворачивается направо, иначе – налево. Процедура повторяется до тех пор, пока жук не вернется в исходную точку. Координаты точек перехода с черного на белое и с белого на черное описывают границу объекта.

Сканирующие алгоритмы основаны на просмотре (сканировании) всего изображения и выделении контурных точек без отслеживания контура объекта.

#### 5.3.4. Связность

Основной этап при формировании символического описания изображения по массиву элементов или набору простейших признаков заключается в определении геометрических соотношений и связности между элементами, относительно которых предполагается, что они принадлежат одному классу [1]. Для двоичной картинки, представленной на рис. 5.28, кольцо из четырех элементов, согласно всем принятым определениям связности, делит эту картинку на три области: белые элементы с внешней стороны кольца, белые элементы внутри кольца и черные элементы самого кольца. Говорят, что элементы внутри каждой области связаны друг с другом. Смысл этого понятия легко уяснить, обратившись к рис. 5.28, *a*, но если рассматривать рис. 5.28, *b*, то возникает неоднозначность.

Определяют ли все черные элементы кольцо или же они представляют собой четыре прямолинейных отрезка? Ответ на этот вопрос до некоторой степени зависит от желаемого определения связности.

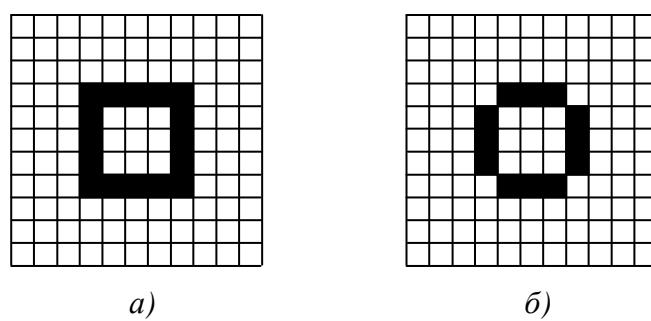


Рис. 5.28. Определение связности:  
*a* – кольцо; *б* – неоднозначная фигура

Возвращаясь к более общему случаю многоградационных изображений, рассмотрим рис. 5.28, *a*, на котором представлен некоторый элемент (элемент *A*), окруженный восемью соседними элементами (от *B* до *I*). Предположим, что элемент *A* обладает свойством *S*, установленным на основе некоторого простейшего описания (яркости, цвета, текстуры и т. д.).

По определению четырехсвязности (под четырехсвязностью здесь понимается связность по четырем направлениям: вверх, вниз и влево, вправо) элемент  $A$  и элемент  $B$  связаны, если оба обладают свойством  $S$ . Аналогично четырехсвязность можно установить между элементом  $A$  и элементами  $E, C$  и  $D$ , граничащими с  $A$  по ребру, при условии, что оба члена пары обладают одним и тем же свойством. Восьмисвязность позволяет связывать элемент  $A$  с одним из его соседей по диагонали, например с элементом  $F$ , граничащим с  $A$  в точке, если оба они обладают одинаковым свойством.

На рис. 5.28, б в соответствии с определением четырехсвязности имеется четыре несвязных черных прямолинейных отрезка, а согласно определению восьмисвязности, изображено кольцо из связных черных элементов. Заметим, однако, что при восьмисвязности белые элементы, расположенные внутри кольца на рис. 5.28, б, связаны с белыми элементами с внешней стороны кольца. Таким образом, возникает парадокс. Если бы черные элементы связывались по принципу восьмисвязности в кольцо, то следовало бы ожидать разделения внутренних и внешних белых элементов этого кольца. Чтобы разрешить эту дилемму, можно для элементов со свойством  $S$  определить восьмисвязность, а принцип четырехсвязности установить для элементов, обладающих свойством  $\bar{S}$  ( $\bar{S}$  – дополнение множества  $S$ ), или наоборот.

Рассмотрим рис. 5.29. Пусть заштрихованный элемент обладает свойством  $S$ , а незаштрихованный – свойством  $\bar{S}$ . Тогда элемент  $A$  на рис. 5.29, б называется изолированным, если для него не соблюдается принцип восьмисвязности относительно любого из его соседей. На рис. 5.29, в элемент  $A$  является внутренним элементом, для которого выполняется принцип четырехсвязности относительно каждого из его соседей  $B, C, D, E$ .

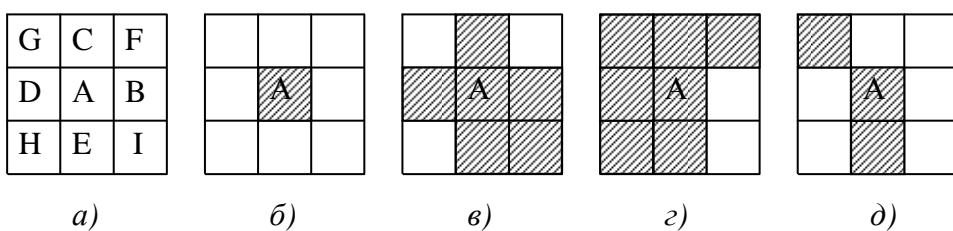


Рис. 5.29. Определения связных элементов изображения:  
а – обозначения элементов окрестности; б – изолированный элемент;  
в – внутренний элемент; г – граничный элемент; д – элемент дуги

Граничный элемент, как показано на рис. 5.29, г, не обладает четырехсвязностью, по крайней мере, с одним из ближайших соседей. Следуя этому определению, элемент  $C$  из окрестности  $A$  не может быть классифи-

цирован как граничная точка. Рис. 5.29, *д* иллюстрирует определение точки дуги; элемент *A* обладает четырехсвязностью только со своими верхним и нижним (или правым и левым) соседями. Дуговой концевой элемент обладает четырехсвязностью лишь с одним соседом. Минимально связная дуга по определению есть множество точек дуги, для которых каждая внутренняя точка дуги (не являющаяся ее концом) обладает восьмисвязностью лишь с двумя соседями.

Обозначим  $S_8(a_{ij})$  как множество всех соседей  $a_{ij}$  (кроме собственно  $a_{ij}$ ), называемое 8-соседями  $a_{ij}$ . Соседи с нечетными номерами прямые соседи  $a_{ij}$ , или 4-соседи, обозначаемые как  $S_4(a_{ij})$ ; соседи с четными номерами – это непрямые соседи  $a_{ij}$ , которые обозначаются как  $S_8(a_{ij})$ . В общем случае под понятием соседства понимается  $S_8$ . Множество  $S_8(a_{ij})$  называется 8-окрестностью  $a_{ij}$ , а множество  $S_4(a_{ij})$  называется 4-окрестностью  $a_{ij}$ . Топология на цифровой плоскости определяется с помощью отношения соседства (рис. 5.30).

$a_4$	$a_3$	$a_2$
$a_5$	$a_{ij}$	$a_1$
$a_6$	$a_7$	$a_8$

Рис. 5.30. Восьмисвязные соседи пикселя  $a_{ij}$

Пиксель из  $B$ , имеющий всех соседей из  $B$ , – внутренний пиксель. Совокупность всех внутренних пикселей  $B$  называется ядром или внутренностью  $B$ . Все пиксели  $B$ , не являющиеся внутренними, называются контурными пикселями (рис. 5.31).

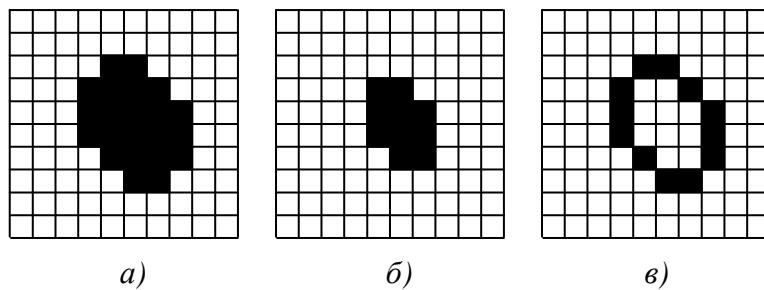


Рис. 5.31. Исходный объект – *а*), его ядро – *б*) и контур – *в*)

Два пикселя ( $B$  или  $W$ ) называются связными, если они являются соседями (расстояние между ними равно 1) в выбранной метрике. Связная компонента изображения – это связное множество пикселей в соответствии с выбранным типом метрики.

В зависимости от выбранного типа метрики, существуют 8-связные линии (рис. 5.32, *a*) и 4-связные линии (рис. 5.32, *б*), а также линии со смешанным типом связности (рис. 5.32, *в*).

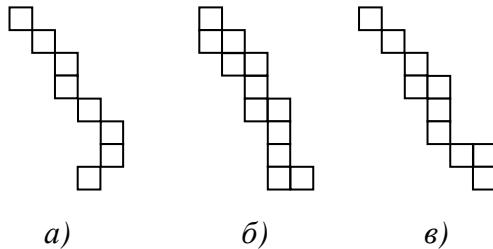


Рис. 5.32. Вид линии в зависимости от типа связности

Для сохранения связности объекта и фона используются различные типы связности. Обычно 8-связность используется для *B*, а 4-связность – для *W* или наоборот.

В обработке бинарных изображений используются следующие понятия связности и характеристических чисел [1]:

$$A_4(a_{ij}) = \sum_{k=1}^4 a_{2k-1} - \text{количество единичных элементов в } S_4;$$

$$A_8(a_{ij}) = \sum_{k=1}^8 a_k - \text{количество единичных элементов в } S_8;$$

$$B_8(a_{ij}) = \sum_{k=1}^8 a_k a_{k+1} - \text{количество единичных двоек в } S_8;$$

$$C_8(a_{ij}) = \sum_{k=1}^4 a_{2k-1} a_{2k} a_{2k+1} - \text{количество единичных троек в } S_8;$$

$$N_{c4}(a_{ij}) = A_4(a_{ij}) - C_8(a_{ij}) - \text{число 4-связности.}$$

Примеры различных пикселей приведены на рис. 5.33. Конечные пиксели и пиксели ветвления называют особыми пикселями (точками).

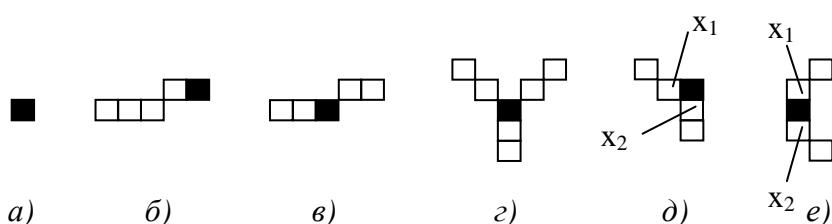


Рис. 5.33. Примеры: *а*) – изолированного пикселя, *б*) – конечного пикселя, *в*) – связующего пикселя, *г*) – пикселя ветвления, *д*) – углового пикселя, *е*) – нормального пикселя

Однако в некоторых случаях утоньшение не может быть полностью проведено до единичной ширины объекта, как для объекта на рис. 5.34 [1].

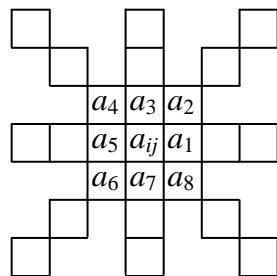


Рис. 5.34. Конфигурация, которая не может быть уточнена до единичной ширины

#### 5.4. Вопросы и задания для самопроверки

1. Что понимают под цифровым изображением?
2. Что такое цифровая обработка изображений?
3. Является ли изображение сигналом? Поясните свой ответ.
4. В каких областях используется цифровая обработка изображений?

Приведите примеры.

5. Назовите ключевые точки лица человека, которые используются при распознавании лица человека по его фотопортрету.
6. От каких факторов зависит архитектура системы обработки изображений? Можете ли вы выделить универсальный доминирующий фактор? Поясните свой ответ.
7. Представьте диаграмму обобщенной системы обработки изображений.
8. Представьте упрощенную схему глаза человека в разрезе и расскажите о строении глаза.
9. Поясните термин «контрастная чувствительность глаза», что он характеризует?
10. Приведите примеры оптических иллюзий.
11. Как называют элемент изображения?
12. Какое изображение называется бинарным?
13. Назовите причины возникновения шумов на растровом изображении.
14. Перечислите и охарактеризуйте типы шумов. От какого типа шумов сложнее (проще) всего избавиться?

15. Поясните суть логической фильтрации бинарного изображения.
16. Что такое уточнение изображений и какие требования предъявляют к данной операции?
17. Представьте маску фильтра размером  $5 \times 5$  для удаления шумов на бинарном изображении.
18. Приведите примеры масок для скелитизации бинарного изображения.
19. Сформулируйте алгоритм выделения контуров на бинарном изображении.
20. Назовите основные типы связности элементов в изображении. Приведите примеры пикселей и характеристических чисел, использующихся в обработке бинарных изображений.

## МОДУЛЬ 6. МЕТОДЫ ПРОСТРАНСТВЕННОЙ ОБРАБОТКИ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ

Цель модуля – изучение студентами алгоритмов пространственной обработки полутоновых изображений, включая градационные преобразования и алгоритмы улучшения изображений.

В результате изучения модуля студенты *должны знать*:

- основные типы градационных преобразований;
- алгоритмы пространственной фильтрации;
- методы улучшения контраста;
- линейные и нелинейные алгоритмы выделения контуров.

Студенты *должны уметь*:

- применять алгоритмы пространственной обработки полутоновых изображений в соответствии с поставленной задачей.

### 6.1. Градационные преобразования

#### 6.1.1. Представление и особенности обработки

Пусть  $F = \{f_{ij} \mid i = 0, \dots, M-1, j = 0, \dots, N-1\}$  – растровое изображение, представляющее собой прямоугольную матрицу,  $f_{ij}$  элемент изображения (пиксель). Если  $f_{ij} \in \{0, 1, \dots, L-1\}$ , то изображение называется полутоновым, и каждый пиксель может принимать  $L$  оттенков серого (градаций яркости) [1]. Пример полутонового изображения показан на рис. 6.1.

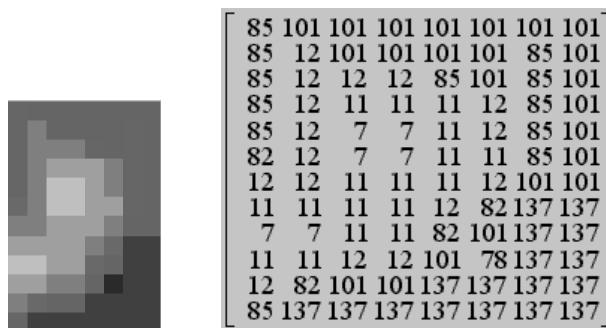


Рис. 6.1. Пример полутонового изображения в матричной форме и в увеличенном растровом виде

Как и для бинарного изображения, основным понятием для полутонового изображения является понятие объекта и фона, однако в силу природы изображения их определение более сложно по сравнению с бинарными.

Все основное множество методов, которые решают эту задачу, делится на методы обработки в частотной и пространственной областях. Термин *пространственная область* относится к плоскости изображения как таковой, и данная категория объединяет подходы, основанные на прямом манипулировании пикселями изображения [5]. К преимуществам методов обработки изображений в пространственной области относится возможность быстрой обработки в масштабе реального времени телевизионного видеосигнала, а к недостаткам – ограниченность функциональных возможностей и недостаточная эффективность. Хотя способы обработки изображений в частотной области и достаточно развиты, но требуют значительных вычислительных затрат и для решения практических задач применяются реже.

В общем виде процессы пространственной обработки изображений описываются выражением:

$$g_{ij} = T[f_{ij}],$$

где  $f_{ij}$  – элементы входного изображения  $F$ ;

$g_{ij}$  – элементы обработанного изображения  $G$ ;

$T$  – оператор преобразования входного изображения в окрестности точки с координатами  $(ij)$ .

Под окрестностью вокруг точки понимают квадратную или прямоугольную область изображения, которая центрирована в точке с координатами  $(ij)$ . Простейшая форма оператора достигается в случае, когда окрестность имеет размеры  $1 \times 1$ . При этом значение  $g$  зависит только от значения  $f$  в точке  $(ij)$  и оператор  $T$  становится функцией градационного преобразования (функцией преобразования интенсивностей или функцией отображения). На основе градационных (поэлементных) преобразований строятся хотя и простые, но достаточно эффективные преобразования изображений (получение негативного изображения, логарифмическое и степенное преобразования и др.). Увеличение значений окрестности приводит к значительно большей гибкости при обработке изображений. При этом используется маска (апerture, окно, ядро, окрестность), которая представляет собой двумерный массив заданного размера, чаще всего размером  $3 \times 3$ . Такие методы называют обработкой или фильтрацией на основе маски.

### 6.1.2. Бинаризация

Бинаризация – это преобразование полутонового изображения к однокрасочному (монохромному или бинарному) [1].

Преобразование цветного изображения в полутоновое заключается в получении яркости каждой точки по формуле:

$$Y = 0,3R + 0,59G + 0,11B,$$

и последующем копировании на все три канала полученной величины:

$$(R = B = G = Y),$$

где  $R, G, B$  – значение красного, зеленого и синего цветов в обрабатываемой точке.

Пусть  $f_{ij}$  – полутоновое изображение,  $t$  – порог и  $b_0, b_1$  – два бинарных значения (для бинарного черно-белого  $b_0 = 0, b_1 = 255$ ). Результат порогового разделения – бинарное изображение  $g_{ij}$ , полученное следующим образом:

$$g_{ij} = \begin{cases} b_0, & \text{if } f_{ij} \leq t; \\ b_1, & \text{if } f_{ij} > t. \end{cases}$$

Как видно, основной задачей является выбор значения  $t$  с помощью некоторого критерия. Это значение может выбираться как одинаковым для всего изображения, так и различным для различных его частей. Если значения объектов и фона режима достаточно однородны по всему изображению, то может использоваться одно пороговое значение для всего изображения. Использование единственного значения порога для всех пикселей изображения называется глобальным пороговым разделением.

Существует много способов выбора порогового значения. Один из наиболее популярных и широко используемых – метод мод, в котором используется гистограмма яркостей пикселей на изображении. Для изображения с хорошо отличимыми объектами и фоновым режимом гистограмма будет иметь два различных пика (рис. 6.2), т. е. бимодальную форму.

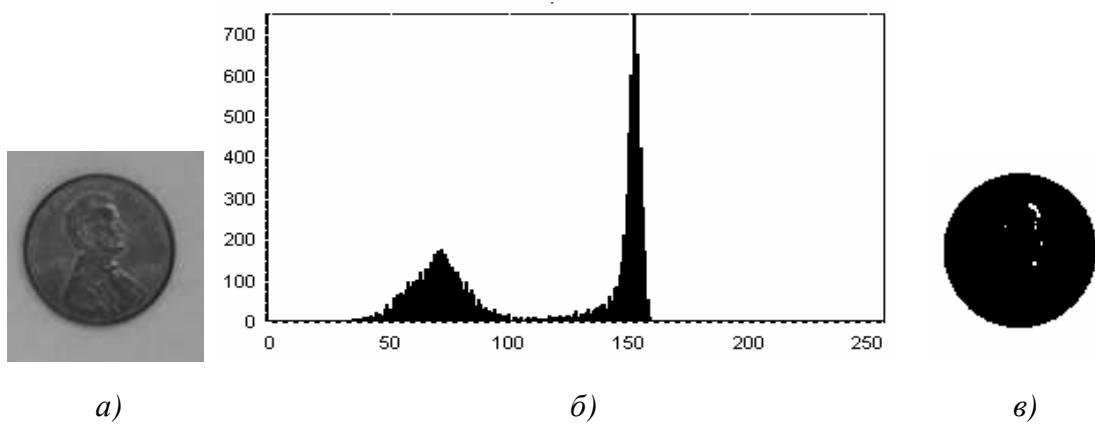


Рис. 6.2. Пример глобального порогового разделения:  
а) исходное изображение; б) гистограмма изображения;  
в) результата бинаризации при  $t = 110$

Впадина между пиками может быть найдена как минимум между двумя максимумами, а соответствующее ему значение интенсивности выбирается как порог, который лучше всего разделяет два пика.

Однако имеется ряд недостатков в методе глобального порогового разделения, основанном на форме распределения яркостей.

1. Изображение не всегда содержит хорошо различимые объект и фон из-за недостаточного контраста и наличия шума.

2. В случае редко расположенных графических объектов, каковыми являются графические изображения, пик, соответствующий объектам, будет значительно меньше пика, соответствующего фону.

Такое различие нередко затрудняет нахождение долины между двумя пиками. Кроме того, надежные модовые методы определения глобального порога сами по себе являются отдельной проблемой.

Существует ряд методов, в которых порог не определяется непосредственно, а гистограмма преобразуется в гистограмму с более глубокими впадинами и более острыми пиками так, чтобы стало возможным определение порога. Общая особенность этих методов – то, что новая гистограмма получена с помощью весовых коэффициентов, применяемых к пикселям, в соответствии с локальными свойствами изображения.

*Локальное пороговое разделение.* Методика локального порогового разделения основана на разделе первоначального изображения на меньшие части и определении порога для каждой части изображения. В результате получается бинарное изображение с разрывами серого уровня на границах фрагментов. Для устранения неоднородностей должны применяться сглаживающие методики.

Основная проблема всех алгоритмов локального порогового разделения – выбор размера фрагмента изображения. Выбранный фрагмент должен иметь достаточно значительные размеры, чтобы получить наиболее достоверную оценку среднего значения, но в тоже время размеры должны быть не слишком велики для предотвращения сглаживания неоднородностей изображения. Однако объект изображения нередко изменяется в размере, так что применение фиксированного размера фрагмента также затруднительно. Для устранения этой проблемы используется информация, зависящая от областей для проверки того, что результаты бинаризации дают ожидаемые результаты. Если результат бинаризации неудовлетворительный, то размер фрагмента может измениться и бинаризация будет проведена снова.

Известны также многопороговые алгоритмы бинаризации полутоновых изображений. Использование как локального, так и глобального порогового разделения при бинаризации не дает идеального результата. В зависимости от качества исходного изображения, на бинарном изображении могут присутствовать такие искажения, как разрывы прямых, неровные края областей и лишние пиксели.

Несмотря на широкий спектр алгоритмов порогового разделения, не существует единственного метода, являющегося наилучшим в применении ко всем типам изображений. Для простых задач, где характеристики изображений не изменяются сильно в пределах изображения или от изображения к изображению, достаточно применения простых методов. Для более сложных задач, отличающихся наличием шума или изменяющимися характеристиками изображения, обычно требуются более сложные (и требующие большего времени вычислений) алгоритмы. В коммерческих программных продуктах сосредоточены различные алгоритмы порогового разделения, использующие как локальные, так и глобальные методы порогового разделения.

### **6.1.3. Преобразование изображения в негатив**

Для изображения с диапазоном яркостей  $[0, \dots, L-1]$  выполняется согласно выражению [5]:

$$g_{ij} = L - 1 - f_{ij}.$$

Такой переворот уровней яркостей цифровых изображений создает эффект фотографического негатива. Преобразование используется для усиления белых или серых деталей на фоне темных областей изображения, особенно когда темные области имеют преобладающие размеры и широко применяется при обработке цифровых медицинских снимков (рис. 6.3 [5]).

### **6.1.4. Логарифмическое преобразование**

Логарифмическое преобразование определяется как:

$$g_{ij} = c \log(1 + f_{ij}),$$

где  $c$  – константа,

$$f \in [0, \dots, L-1].$$

Использование логарифма позволяет узкий диапазон малых значений яркости преобразовать в более широкий диапазон выходных значений. Для больших значений входного сигнала верно противоположное утверждение. Такой тип преобразования используется для растяжения диапазона значений темных пикселей на изображении с одновременным сжатием диапазона значений ярких пикселей. При использовании обратного логарифмического преобразования происходит растяжение диапазона значений ярких пикселей на изображении с одновременным сжатием диапазона значений темных пикселей. Логарифмическая функция имеет важную особенность – позволяет сжимать динамический диапазон изображений, имеющих большие вариации в

значениях пикселей. Классическим примером большого динамического диапазона пикселей является спектр Фурье изображения, поэтому для отображения спектра используется логарифмическое преобразование, позволяющее отобразить большее количество видимых спектральных составляющих.

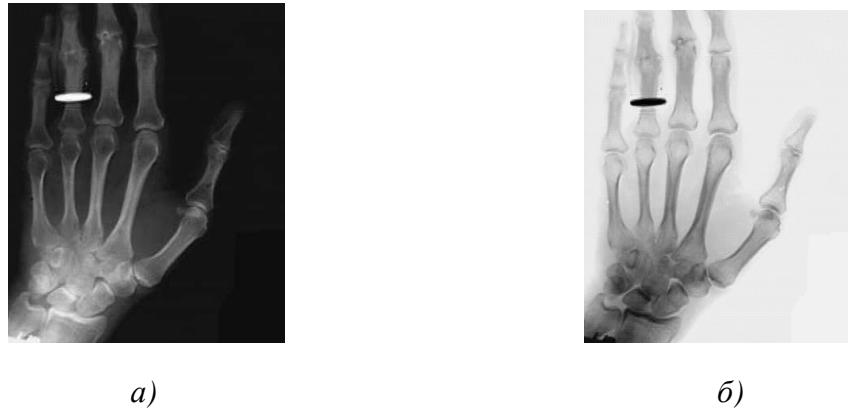


Рис. 6.3. Рентгенограмма кисти руки человека:  
а) исходное изображение; б) негативное изображение

### 6.1.5. Степенные преобразования

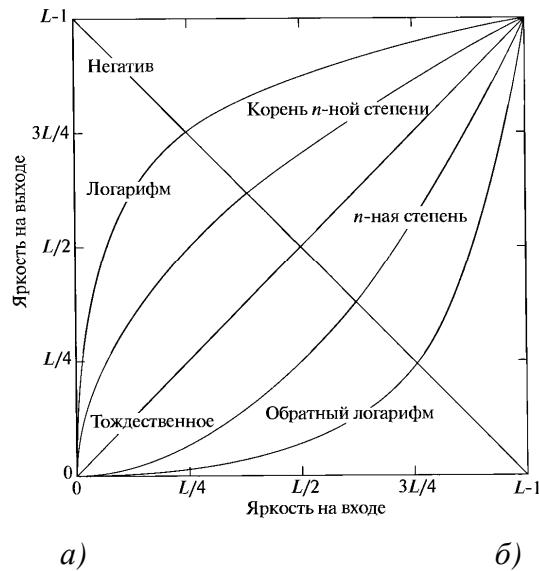
Степенные преобразования описываются выражением:

$$g_{ij} = cf_{ij}^{\gamma},$$

где  $c, \gamma$  – положительные константы.

Следует отметить, что уравнение приводит к тождественному преобразованию при  $c = \gamma = 1$ . В отличие от логарифмического преобразования, здесь возможно целое семейство преобразований, получаемых изменением величины  $\gamma$ . Преобразования со значениями  $\gamma < 1$  дают прямо противоположный эффект по сравнению с преобразованиями, которые получены при  $\gamma > 1$ . Следует отметить, что амплитудная характеристика многих устройств, используемых для ввода, печати или визуализации изображений соответствует степенному закону. Процедура, используемая для коррекции такой степенной характеристики, называется гамма-коррекцией (отсюда и обозначение показателя степени). Например, ЭЛТ-мониторы имеют степенную зависимость яркости от напряжения с показателем степени в диапазоне от 1,8 до 2,5, то есть изображения на мониторе будут темнее, чем они есть на самом деле. Некоторые компьютерные системы имеют встроенную частичную гамма-коррекцию. Кроме этого гамма-коррекция используется для универсального управления контрастом, в частности, при обработке медицинских изображений.

В результате применения логарифмического или степенного преобразования изменяется лишь яркость изображения. На рис. 6.4. [5] представлены функции градационных преобразований.



a)

б)

Рис. 6.4. Функции градационных преобразований:

а) логарифмического преобразования; б) степенного преобразования

### 6.1.6. Кусочно-линейные функции преобразования

Характеристики применяемых на практике процедур изменения диапазона яркостей на основе кусочно-линейных процедур преобразования (препарирование изображений) приведены на рис. 6.5. Очевидно, что преобразование с пороговой характеристикой (рис. 6.5, а) переводит полуточновое изображение в бинарное. Смыл других преобразований, использующих функции, представленные на рис. 6.5, очевиден из их анализа. Например, преобразование рис. 6.5, б выполняет яркостный срез изображения, выделяя те его участки, где яркость соответствует выделенному интервалу. При этом остальные участки имеют яркость, соответствующую уровню черного. Перемещая выделенный интервал по яркостной шкале и изменяя его ширину, можно детально исследовать содержание картины. Преобразование рис. 6.5, в повышает яркость точек из выбранного диапазона, однако сохраняет яркости других уровней. Преобразование, приведенное на рис. 6.5, г, позволяет повысить детальность наблюдаемой картины в выбранном диапазоне яркостей, однако в отличие от предыдущего здесь выходное изображение использует полный динамический диапазон. Данное преобразование представляет собой линейное контрастирование, применяемое к изциальному диапазону входного изображения. Как и в пре-

дальнем варианте, участки, не попавшие в этот диапазон, образуют после препарирования черный фон. Иногда наглядность изображения повышается применением преобразования типа пилообразного контрастирования (рис. 6.5, *д*). При этом различные яркостные диапазоны одновременно подвергаются локальному яркостному контрастированию. Однако необходимо иметь в виду, что данное преобразование, как и некоторые другие, может сопровождаться появлением ложных контуров на получаемом препарате.

Главное преимущество кусочно-линейных функций заключается в том, что их форма может быть сколь угодно сложной, однако с увеличением их сложности необходимо задавать большее количество параметров, описывающих эти функции.

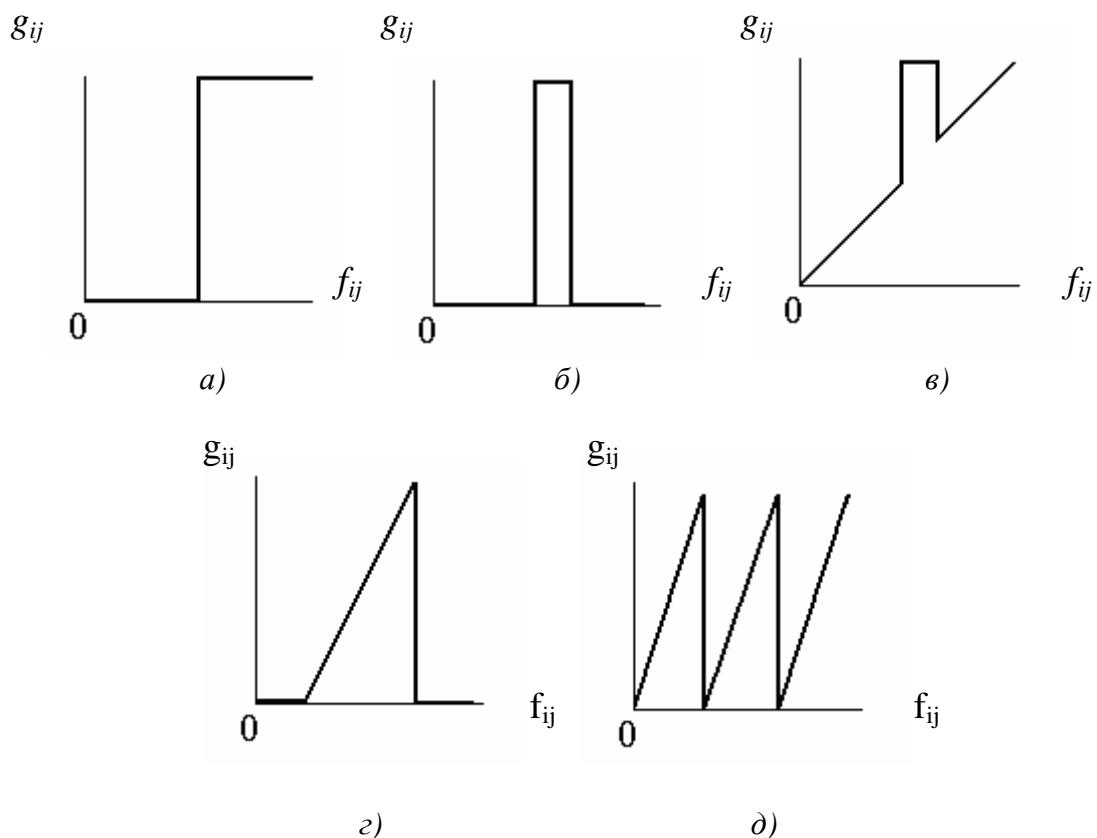
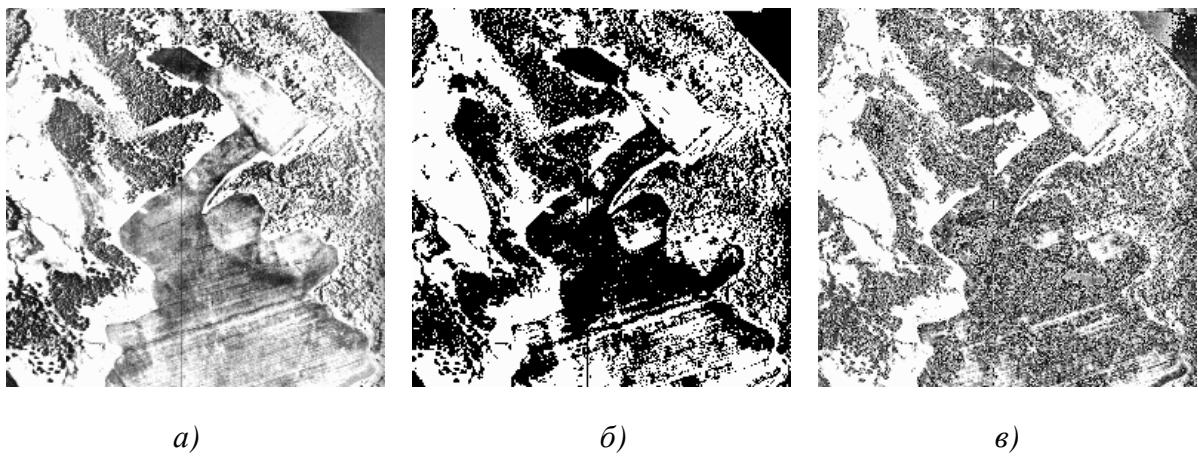


Рис. 6.5. Примеры кусочно-линейных функций

На рис. 6.6 приведены результаты бинаризации и пилообразного преобразования аэрофотоснимка участка Земли. Первое преобразование приводит к выявлению границ отдельных участков, создавая общее интегральное представление о наблюдаемой сцене. Второе, наоборот, дает возможность наблюдения мелких деталей на всех участках изображения.



a)

б)

в)

Рис. 6.6. Примеры применения кусочно-линейных функций

Рассмотренные выше процедуры обработки изображений часто используются также в автоматических системах обработки визуальной информации, поскольку подготавливаемый при этом препарат может содержать всю информацию, необходимую для последующей (вторичной) обработки. Например, если при наблюдении из космоса требуется автоматически обнаружить на изображении некоторый объект, имеющий известную конфигурацию, то для этого может быть достаточно бинарного препарата, передающего эту конфигурацию.

### 6.1.7. Соляризация изображения

При данном виде обработки преобразование имеет вид:

$$g_{ij} = k \cdot f_{ij} \cdot (f_{\max} - f_{ij}),$$

где  $f_{\max}$  – максимальное значение входного изображения,

$k$  – константа, позволяющая управлять динамическим диапазоном преобразованного изображения.

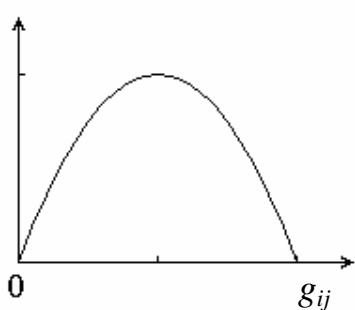


Рис. 6.7. Функция, описывающая соляризацию

Функция, описывающая данное преобразование, является квадратичной параболой, ее график при  $k=1$  приведен на рис. 6.7. При  $g_{\max} = f_{\max}$  динамические диапазоны изображений совпадают, что может быть достигнуто при  $k = 4/f_{\max}$ .

Как следует из рис. 6.7, смысл соляризации заключается в том, что участки исходного изображения, имеющие уровень белого или близкий к нему уровень яркости, после обра-

ботки имеют уровень черного. При этом сохраняют уровень черного и участки, имеющие его на исходном изображении. Уровень же белого на выходе приобретают участки, имеющие на входе средний уровень яркости (уровень серого). Пример применения соляризации приведен на рис. 6.8.



Рис. 6.8. Пример соляризации

На рис. 6.8, *a* показано исходное изображение, а на рис. 6.8, *б* – результат его соляризации. На втором этапе обработки здесь применено линейное контрастирование при  $g_{\min} = 0$  и  $g_{\max} = 255$ . Обработка приводит к повышению четкости деталей изображения: улучшены изображения глаз, повышен контраст на переходе «лицо – волосы» и т. д.

## 6.2. Фильтрация изображений

### 6.2.1. Принципы и особенности пространственной фильтрации

Изображения, сформированные различными информационными системами, искажаются действием помех. Ослабление действия помех достигается фильтрацией. При фильтрации яркость (сигнал) каждой точки исходного изображения, искаженного помехой, заменяется некоторым другим значением яркости, которое признается в наименьшей степени искаженным помехой [5,12].

Задача заключается в том, чтобы найти эффективную вычислительную процедуру, которая позволяла бы достигать наилучших результатов. Существует ряд методов и алгоритмов фильтрации, что связано с большим разнообразием сюжетов, которые приходится описывать различными математическими моделями. Кроме того, применяются различные критерии оптимальности, что также ведет к разнообразию методов фильтрации. На-

конец, даже при совпадении моделей и критериев очень часто из-за математических трудностей не удается найти оптимальную процедуру. Сложность нахождения точных решений порождает различные варианты приближенных методов и процедур.

В практике цифровой обработки изображений широко используется масочная фильтрация. В качестве маски используется множество весовых коэффициентов, заданных во всех точках окрестности, обычно симметрично окружающих рабочую точку кадра. Распространенным видом окрестности, часто применяемым на практике, является квадрат  $3 \times 3$  с рабочим элементом в центре (рис. 6.9). На рис. 6.10 представлены элементы области и изображения под маской. При цифровой обработке изображений используется декартова система координат с началом в левом верхнем углу кадра и с положительными направлениями из этой точки вниз и вправо.

$$\begin{array}{ccc}
 w_{(-1,-1)} & w_{(-1,0)} & w_{(-1,1)} \\
 w_{(0,-1)} & w_{(0,0)} & w_{(0,1)} \\
 w_{(1,-1)} & w_{(1,0)} & w_{(1,1)}
 \end{array}
 \quad
 \begin{array}{ccc}
 f_{(i-1,j-1)} & f_{(i-1,j)} & f_{(i-1,j+1)} \\
 f_{(i,j-1)} & f_{(i,j)} & f_{(i,j+1)} \\
 f_{(i+1,j-1)} & f_{(i+1,j)} & f_{(i+1,j+1)}
 \end{array}$$

Рис. 6.9. Коэффициенты маски с относительными значениями координат

Рис. 6.10. Элементы области изображения под маской

Сущность процедуры фильтрации заключается в смещении маски фильтра по изображению с шагом один пиксель слева направо, сверху вниз. При этом отклик задается суммой произведений коэффициентов фильтра на соответствующие значения пикселей в области, покрытой маской фильтра. Для маски размером  $3 \times 3$  отклик  $r$  определяется:

$$\begin{aligned}
 r = & w_{(-1,-1)}f_{(i-1,j-1)} + w_{(-1,0)}f_{(i-1,j)} + w_{(-1,1)}f_{(i-1,j+1)} + \\
 & + w_{(0,-1)}f_{(i,j-1)} + w_{(0,0)}f_{(i,j)} + w_{(0,1)}f_{(i,j+1)} + w_{(1,-1)}f_{(i+1,j-1)} + \\
 & + w_{(1,0)}f_{(i+1,j)} + w_{(1,1)}f_{(i+1,j+1)}.
 \end{aligned}$$

В общем случае фильтрация изображения  $f_{ij}$  размером  $M \times N$  с использованием маски размером  $m \times n$  описывается выражением:

$$g_{ij} = \sum_{s=-a}^a \sum_{t=-b}^b w_{st} f_{i+s, j+t},$$

где  $a = (m - 1) / 2$ ,  $b = (n - 1) / 2$ .

Таким образом, можно использовать маски размерности не только  $3 \times 3$ , но и большей размерности, например,  $5 \times 5$ ,  $7 \times 7$  и т. п.

Процедура линейной фильтрации в частотной области аналогична операции свертки, поэтому линейную пространственную фильтрацию называют сверткой маски с изображением.

Если центр маски находится на границе изображения, то ее края выходят за пределы изображения. Для обработки краев изображений используются следующие приемы [1].

*Дополнение средним значением.* Все недостающие отсчеты считаются равными среднему значению отсчетов последовательности: статистическому среднему или среднему по заданной последовательности. Наиболее часто встречающаяся разновидность этого способа – дополнение нулями. Недостатком его является то, что при экстраполяции средним значением почти всегда имеются большие разрывы в величине отсчетов сигнала на краях последовательности.

*Доопределение повторением крайних отсчетов последовательности.* Это экстраполяция нулевого порядка. Она не дает скачков на краях, но повторение крайних отсчетов при экстраполяции на большое число дополнительных отсчетов сильно искажает структуру сигнала и результат фильтрации на краях.

*Экстраполяция более высокого порядка.* Недостающие отсчеты определяются как взвешенная сумма крайних заданных отсчетов последовательности. Веса могут выбираться из статистических соображений по критерию минимума среднеквадратической ошибки экстраполяции, если известна статистическая корреляционная функция сигнала, или классическими методами, сохраняющими значения первых, вторых и более высоких разностей сигнала. Метод неудобен сложностью вычислений.

### **6.2.2. Низкочастотная фильтрация**

В спектре шума, как правило, содержатся более высокие пространственные частоты, чем в спектре обычного изображения. Следовательно, простая усредняющая или сглаживающая пространственная фильтрация может служить эффективным средством подавления высокочастотных шумов.

Пусть отсчеты полезного изображения мало меняются в пределах маски. На изображение накладывается аддитивный шум, отсчеты которого случайны и независимы (или слабо зависимы) со статистической точки зрения. В этом случае механизм подавления шума с использованием фильтра-

ции состоит в том, что при суммировании шумы компенсируют друг друга. Эта компенсация будет происходить тем успешнее, чем большее число членов в сумме, то есть чем больше размер (апертура) маски.

При импульсной помехе механизм подавления состоит в том, что импульс «расплывается» и становится менее заметным на общем фоне. На рис. 6.11 приведены сглаживающие массивы трех разновидностей, часто называемые шумоподавляющими масками [12].

$$H1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad H2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad H3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Рис. 6.11. Сглаживающие маски

Эти массивы нормированы для получения единичного коэффициента передачи, чтобы процедура подавления шума не вызывала смещения средней яркости обработанного изображения. На рис. 6.12 показан пример работы маски  $H1$  [1].

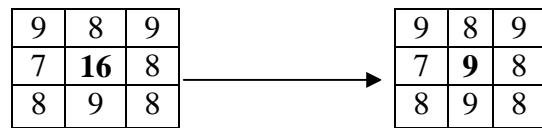


Рис. 6.12. Пример работы маски  $H1$

Для фрагментов, представленных на рис. 6.12, центральный элемент определяется следующим образом:

$$r = \frac{1}{9} (1 \cdot 9 + 1 \cdot 8 + 1 \cdot 9 + 1 \cdot 7 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 8 + 1 \cdot 9 + 1 \cdot 8) = 9.$$

Однако часто в пределах апертуры значения полезного изображения все же изменяются заметным образом. Это бывает, в частности, когда в пределы маски попадают контуры. Как уже было сказано, с физической точки зрения маски  $H1 - H3$  являются фильтрами низких частот, подавляющими высокочастотные гармоники и шума, и полезного изображения. Это приводит не только к ослаблению шума, но и к размыванию контуров на изображении.

Для устранения эффекта размывания контуров при подавлении шума следует переходить к нелинейной обработке. Примером нелинейного фильтра для подавления шума служит медианный фильтр.

Медианная фильтрация представляет собой эвристический метод обработки, ее алгоритм не является математическим решением строго сформулированной задачи. Поэтому уделяется большое внимание анализу эффективности обработки изображений на ее основе и сопоставлению с другими методами.

При медианной фильтрации используется двумерное окно (апертура фильтра), обычно имеющее центральную симметрию, при этом его центр располагается в текущей точке фильтрации. На рис. 6.13 показаны два примера наиболее часто применяемых вариантов окон в виде креста и в виде квадрата. Размеры апертуры принадлежат к числу параметров, оптимизируемых в процессе анализа эффективности алгоритма.



Рис. 6.13. Примеры окон при медианной фильтрации изображений

Обозначим рабочую выборку в виде одномерного массива  $W = \{w_1, w_2, \dots, w_n\}$ ; число его элементов равняется размеру окна, а их расположение произвольно. Обычно применяют окна с нечетным числом точек  $n$  (это автоматически обеспечивается при центральной симметрии апертуры и при вхождении самой центральной точки в ее состав). Если упорядочить последовательность  $\{w_k, k = \overline{1, n}\}$  по возрастанию, то ее медианой будет тот элемент выборки, который занимает центральное положение в этой упорядоченной последовательности. Полученное таким образом число и является продуктом фильтрации для текущей точки кадра. Формальное обозначение описанной процедуры представляется в виде:

$$g^* = med(w_1, w_2, \dots, w_n).$$

Рассмотрим пример. Предположим, что выборка имеет вид:  $W = \{9, 8, 9, 7, 16, 8, 8, 9, 8\}$ , а элемент 16, расположенный в ее центре, соответствует текущей точке фильтрации. Большое значение яркости в этой точке кадра может быть результатом воздействия импульсной (точечной) помехи. Упорядоченная по возрастанию выборка имеет при этом вид  $\{7, 8, 8, 8, 8, 9, 9, 9, 16\}$ , следовательно,  $g^* = 8$ . Влияние соседних пикселей на результат фильтрации в текущей точке привело к игнорированию импульсного выброса яркости, что принято рассматривать как эффект фильтрации.

Если импульсная помеха не является точечной, а покрывает некоторую локальную область, то она также может быть подавлена с помощью медианной фильтрации, если размер этой локальной области будет меньше, чем половина размера апертуры фильтра. Поэтому для подавления импульсных помех, поражающих локальные участки изображения, следует увеличивать размеры апертуры медианного фильтра. При подавлении независимого гауссовского шума эффективность медианной фильтрации ниже эффективности линейной фильтрации.

Медианная фильтрация в меньшей степени сглаживает границы изображения, чем любая линейная фильтрация.

Предположим, что апертура фильтра находится вблизи границы, разделяющей светлый и темный участки изображения, при этом ее центр располагается в области темного участка. Тогда, вероятнее всего, рабочая выборка будет содержать большее количество элементов с малыми значениями яркости, и, следовательно, медиана будет находиться среди тех элементов рабочей выборки, которые соответствуют этой области изображения. Ситуация меняется на противоположную, если центр апертуры смещен в область более высокой яркости. Но это означает наличие чувствительности у медианного фильтра к перепадам яркости.

### 6.2.3. Подчеркивание границ

Известно, что фотографическое или телевизионное изображение с подчеркнутыми границами оказывается субъективно более приятным, чем фотометрически совершенная репродукция. Повышение резкости изображений может быть достигнуто путем численного дифференцирования. С принципиальной точки зрения, величина отклика оператора производной в точке изображения пропорциональна степени разрыва изображения в данной точке. Таким образом, дифференцирование изображения позволяет усилить перепады яркости и другие разрывы (шумы, помехи) и не подчеркивать области с медленными изменениями яркостей.

Для вычисления двумерной второй производной и наложение результата на изображение (высокочастотная фильтрация) используются три маски (рис. 6.14) [12]:

$$H4 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad H5 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad H6 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Рис. 6.14. Маски для выполнения высокочастотной фильтрации

Одной из задач фильтрации изображений с подъемом высоких частот является случай, когда исходное изображение темнее, чем требуется. В этом случае можно варьировать коэффициент усиления высоких частот  $U > 1$  (рис. 6.15) для увеличения общей яркости изображения. Примеры обработки изображений показаны на рис. 6.16 – 6.21.

$$H7 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & U+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad H8 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & U+8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad H9 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & U+4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Рис. 6.15. Маски высокочастотной фильтрации с изменением яркости



Рис. 6.16. Исходное изображение

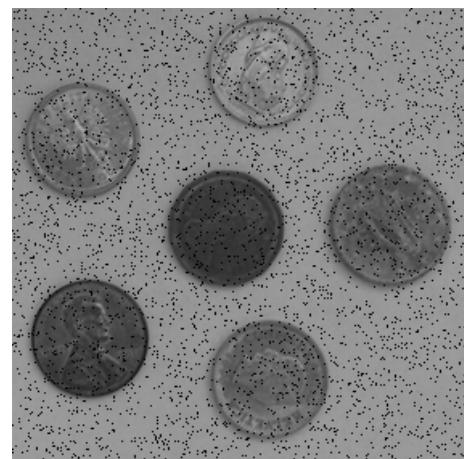


Рис. 6.17. Изображение, искаженное 5 % импульсным шумом

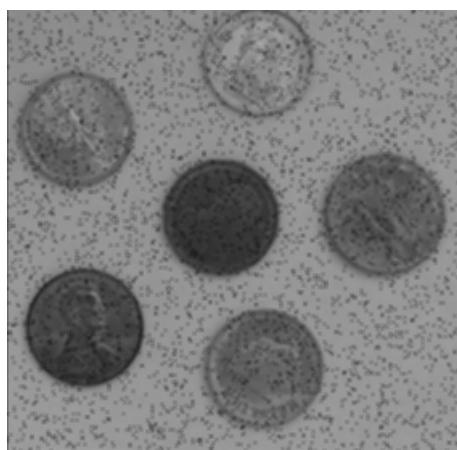


Рис. 6.18. Результат применения фильтра  $H1$



Рис. 6.19. Результат медианной фильтрации (маска  $3 \times 3$ )



Рис. 6.20. Результат подчеркивания границ (маска  $H5$ )



Рис. 6.21. Результат подчеркивания границ с увеличением яркости (маска  $H8$ ,  $U = 2$ )

Анализ результатов обработки показывает, что импульсные помехи не могут эффективно подавляться сглаживающими фильтрации, но полностью удаляются подходящим медианным фильтром. Высокочастотная фильтрация не только улучшает детальность объектов на изображении, но и подчеркивает высокочастотные шумы.

### 6.3. Улучшение изображений на основе арифметико-логических операций и изменения гистограмм

#### 6.3.1. Улучшение на основе арифметико-логических операций

В применении к изображениям, арифметико-логические операции являются поэлементными преобразованиями над двумя или более изображениями (за исключением операции НЕ, которая выполняется над одним изображением и выполняет ту же самую операцию, что и негативное преобразование) [5]. Например, разность двух изображений дает новое изображение, значение элемента с координатами  $(i, j)$  которого есть разность значений элементов двух изображений в той же самой точке. Известно, что логические операции И, ИЛИ, НЕ являются базовыми, а любая другая логическая операция может быть получена с использованием только этих трех основных операций.

Таблица 6.2

#### Логические операторы

Оператор	Выражение
NOT (НЕ)	$c = \bar{a}$
OR (ИЛИ)	$c = a + b$
AND (И)	$c = a \cdot b$
XOR (суммирование по модулю 2)	$c = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b$
SUB (разность)	$c = a - b = a \cdot \bar{b}$

При выполнении логических операций над полутоновыми изображениями значения элементов обрабатываются как строки двоичных значений. Например, выполнение операции НЕ над 8-битовым черным пикселям (строка из восьми нулей) приведет к получению белого пикселя (строка из восьми единиц). Промежуточные значения преобразуются похожим образом, изменяя нули на единицы и наоборот.

Примеры обработки изображений представлены на рис. 6.22, на которых логическая единица соответствует черному цвету, логический ноль соответствует белому цвету.

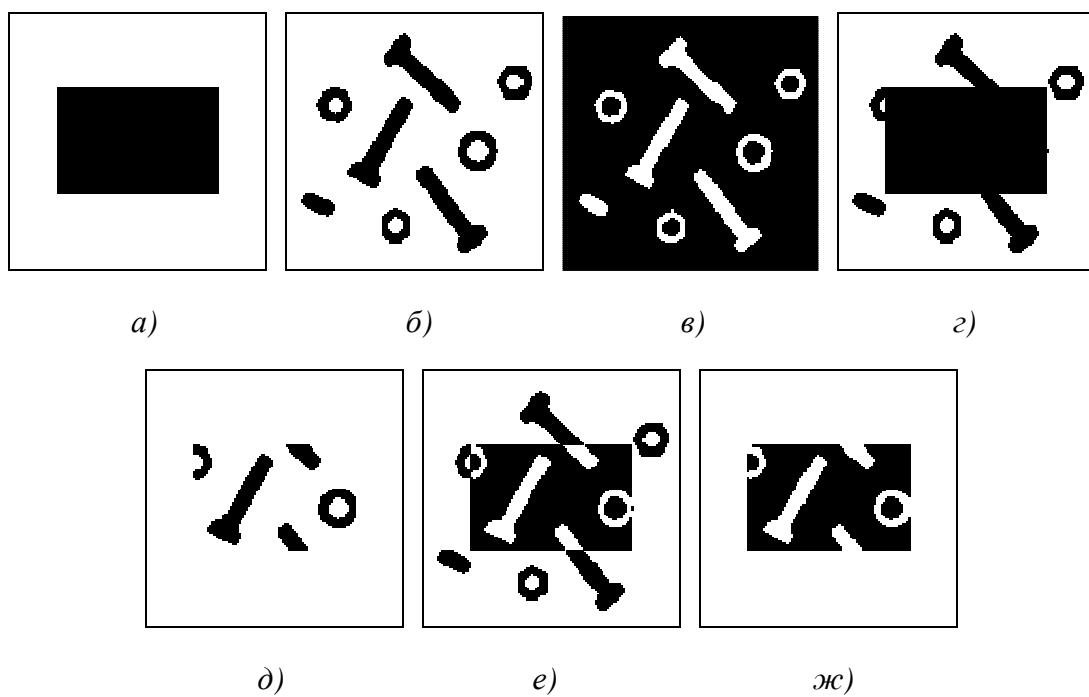


Рис. 6.22. Примеры обработки бинарных изображений с использованием логических операций: а) изображение  $F$ ; б) изображение  $H$ ; в) результат применения операции НЕ к изображению  $H$ ; г) результат применения операции ИЛИ к изображениям  $F$  и  $H$ ; д) результат применения операции И к изображениям  $F$  и  $H$ ; е) результат применения операции XOR к изображениям  $F$  и  $H$ ; жс) результат применения операции SUB к изображениям  $F$  и  $H$

Операция вычитания используется для *маскирования*, то есть для выделения части изображения. Маскирование иногда имеет отношение к *обработке по области интереса*. С точки зрения задачи улучшения, маскирование в основном используется для изолирования области обработки. Это делается, чтобы выделить данную область и отличить ее от остальной части изображения. Логические операции также часто используются в морфологических преобразованиях.

Из четырех арифметических операций наиболее часто используемыми для улучшения изображений являются вычитание и сложение. Помимо очевидной операции умножения изображения на константу с целью увеличения среднего уровня его яркости, умножение изображений находит применение в задаче улучшения в первую очередь как операция маскирования, которая является более общей, чем логические маски, обсуждавшиеся в предыдущем абзаце. Другими словами, умножение одного изображения на другое может более универсально использоваться для коррекции уровня яркости, чем двоичные маски.

### 6.3.2. Глобальные методы улучшения контраста

Слабый контраст – один из наиболее распространенных дефектов фотографических, сканированных и телевизионных изображений, обусловленный ограниченностью диапазона воспроизводимых яркостей. Путем цифровой обработки контраст можно повысить, изменяя яркость каждого элемента изображения и увеличивая диапазон яркостей [12].

Как уже было отмечено, если для цифрового представления каждого отсчета изображения отводится 1 байт запоминающего устройства, то входной или выходной сигналы могут принимать одно из 256 значений в диапазоне 0 – 255. Предположим, что минимальная и максимальная яркости исходного изображения равны  $f_{\min}$  и  $f_{\max}$  соответственно. Если эти параметры или один из них существенно отличаются от граничных значений яркостного диапазона, то визуализированная картина выглядит как слабоконтрастная.

В реальных изображениях обычно существует перекос в сторону малых уровней яркости большинства элементов изображения ниже средней. На темных участках подобных изображений детали часто оказываются неразличимыми. Одним из методов улучшения таких изображений является видоизменение гистограммы. Этот метод предусматривает преобразование яркостей исходного изображения с тем, чтобы гистограмма распределения яркостей обработанного изображения приняла желаемую форму.

Одним из методов улучшения контраста является линейная растяжка гистограммы, когда уровням исходного изображения, лежащим в интервале  $[f_{\min}, f_{\max}]$ , присваиваются новые значения с тем, чтобы охватить весь возможный интервал изменения яркости, в данном случае  $[0, 255]$ . При линейном контрастировании используется линейное поэлементное преобразование вида:

$$g_{ij} = a \cdot f_{ij} + b,$$

параметры которого  $a$  и  $b$  определяются желаемыми значениями минимальной  $g_{\min}$  и максимальной  $g_{\max}$  выходной яркости. Решая систему уравнений:

$$\begin{cases} g_{\min} = a \cdot f_{\min} + b, \\ g_{\max} = a \cdot f_{\max} + b. \end{cases}$$

Для рис. 6.19 находим  $f_{\min} = 29$  и  $f_{\max} = 177$ . Далее необходимо определить  $a$  и  $b$  таким образом, чтобы  $g_{\min} = 0$ ,  $g_{\max} = 255$ . Путем решения системы уравнений получаем  $a = 1,72$  и  $b = -50$ .

Относительно параметров преобразования  $a$  и  $b$ , выражение для линейного преобразования можно привести к виду:

$$g_{ij} = \frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}} (g_{\max} - g_{\min}) + g_{\min}.$$

При нормализации гистограммы на весь максимальный интервал уровней яркости  $[0, 255]$  растягивается не вся гистограмма, лежащая в пределах  $[f_{\min}, f_{\max}]$ , а ее наиболее интенсивный участок  $[f'_{\min}, f'_{\max}]$ , из рассмотрения исключаются малоинформационные начальный и конечный участки. Например, для рассматриваемого изображения на основе анализа гистограммы исходного изображения примем  $f'_{\min} = 45$  и  $f'_{\max} = 160$ .

Целью выравнивания гистограммы (линеаризации, эквализации) является такое преобразование, чтобы, в идеале, все уровни яркости приобрели бы одинаковую частоту, а гистограмма яркостей отвечала бы равномерному закону распределения. Кроме этого существует метод видоизменения гистограмм, который обеспечивает экспоненциальную или гиперболическую форму распределения яркостей улучшенного изображения. Применяются также «локальные» методы улучшения, на основе обработки части изображения.

Пусть изображение имеет формат:  $N$  пикселов по горизонтали и  $M$  по вертикали, число уровней квантования яркости равно  $L$ . Общее число пикселов равно  $N \times M$ , на один уровень яркости попадает, в среднем,  $y = N \times M / L$  пикселов. Например,  $N = M = 512$ ,  $L = 256$ . В этом случае  $y = 1024$ . Расстояние  $\Delta f$  между соседними дискретными уровнями яркости в гистограмме исходного изображения одинаковое, но на каждый уровень выпадает различное число пикселов. При эквализации гистограммы расстояние  $\Delta g$  между соседними уровнями различно, но число пикселов на каждом уровне, в среднем, одинаковое и примерно равно  $y$ .

Рассмотрим алгоритм эквализации на примере. Пусть уровень яркости 0 на исходном изображении имеют 188 пикселов, уровень 1 – 347 пикселов, уровень 2 – 544 пикселя. В сумме это 1079 пикселов, то есть приблизительно  $y$ . Присвоим всем этим пикселям уровень 0. Пусть на исходном изображении число пикселов с уровнями яркости 3 и 4 в сумме приблизительно также равно  $y$ . Этим пикселям присваивается уровень 1. С другой стороны, пусть число пикселов с уровнем 45 на исходном изображении составляет 3012, то есть приблизительно  $3y$ . Всем этим пикселям присваивается некоторый одинаковый уровень  $g_i$ , не обязательно равный 45, а соседние два уровня остаются незаполненными. Рассмотренные процедуры выполняются для всех уровней яркости.

Результаты улучшения контраста изображения и гистограммы полученных изображений показаны на рис. 6.23 – 6.26.

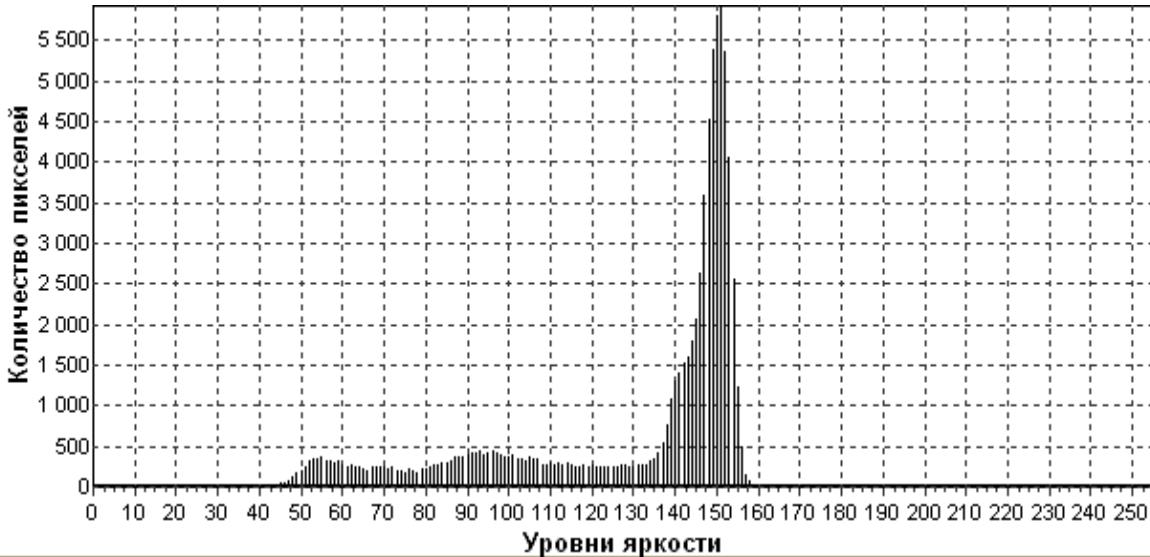


Рис. 6.23. Гистограмма исходного изображения (рис. 6.16)

Для изменения гистограммы широко используются также методы, представленные в табл. 6.3 [12].

Таблица 6.3

#### Характеристики передачи уровней, используемые при видоизменении гистограмм

Плотность вероятности на выходе	Характеристика передачи уровней
Равномерная (линейная) $p_g(g) = 1/(g_{\max} - g_{\min}), g_{\min} \leq g \leq g_{\max}$	$g_{ij} = [g_{\max} - g_{\min}] \frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}} + g_{\min}$
Экспоненциальная $p_g(g) = \alpha \exp\{-\alpha(g - g_{\min})\}, g \geq g_{\min}$	$g_{ij} = g_{\min} - (1/\alpha) \ln[1 - \frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}]$

Окончание табл. 6.3

Плотность вероятности на выходе	Характеристика передачи уровней
Рэлея $p_g(g) = \frac{g - g_{\min}}{\alpha^2} \exp\left\{-\frac{(g - g_{\min})^2}{2\alpha^2}\right\}, g \geq g_{\min}$	$g_{ij} = g_{\min} + \left[ 2\alpha^2 \ln\left(\frac{1}{1 - \frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}}\right)\right]^{1/2}$
Степени 2/3 $p_g(g) = (1/3)[g^{-2/3} / (g_{\max}^{1/3} - g_{\min}^{1/3})]$	$g_{ij} = ([g_{\max}^{1/3} - g_{\min}^{1/3}] [\frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}] + g_{\min}^{1/3})^3$
Гиперболическая $p_g(g) = 1/g[\ln(g_{\max}) - \ln(g_{\min})]$	$g_{ij} = g_{\min} [g_{\max} / g_{\min}]^{\left(\frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}\right)}$

Здесь  $\alpha$  – коэффициент преобразования.

Следует отметить, что степень улучшения субъективного восприятия зависит от характера изображения.



Рис. 6.24. Результат линейной растяжки гистограммы изображения, представленного на рис. 6.16



Рис. 6.25. Результат нормализации гистограммы изображения, представленного на рис. 6.16

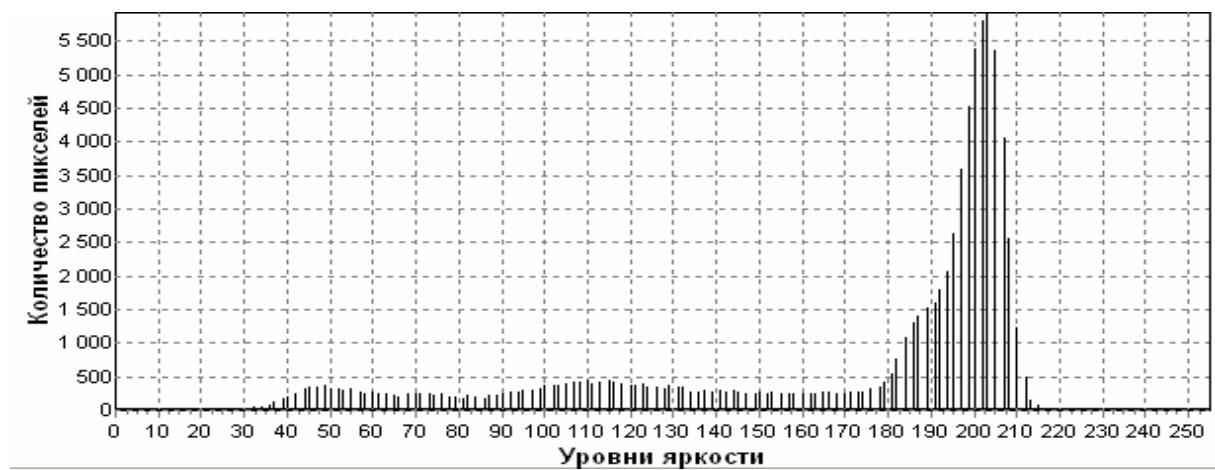


Рис. 6.26. Линейная растяжка гистограммы

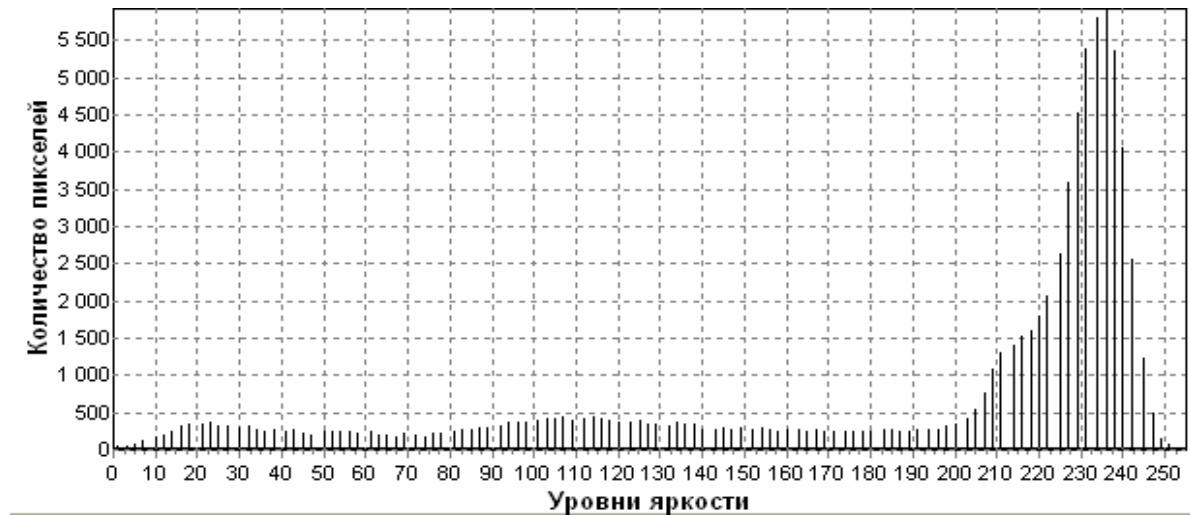


Рис. 6.27. Нормализация гистограммы ( $f'_{\min} = 45, f'_{\max} = 160$ )

Форма гистограммы дает много сведений о свойствах изображения. Например, узкая гистограмма указывает на низкий контраст, а бимодальная гистограмма предполагает наличие областей разной яркости.

### 6.3.3. Метод локального улучшения контраста

Основная идея технологии повышения качества изображений, которая основывается на преобразовании локальных контрастов, состоит в том, что для каждого элемента изображения сначала определяется локальный контраст, а потом происходит его нелинейное усиление и восстановление яркости данного элемента изображения из уже скорректированного локального контраста.

Алгоритм предполагает выполнение следующих шагов:

1. Выбирается элемент  $f_{ij}$  изображения  $F$ ;
2. Вычисляется локальный контраст элемента по формуле:

$$g_{ij}^1 = \frac{\left| \bar{f}_{ij}^1 - \bar{f}_{ij}^2 \right|}{\bar{f}_{ij}^1 + \bar{f}_{ij}^2},$$

где  $\bar{f}_{ij}^1 = \frac{1}{n^2} \sum_{k=0}^{W_1-1} \sum_{l=0}^{W_1-1} f_{kl};$

$$\bar{f}_{ij}^2 = \frac{1}{m^2} \sum_{k=0}^{W_2-1} \sum_{l=0}^{W_2-1} f_{kl};$$

$$m = 3n, n > 1 \& n = 1(\text{mod } 2).$$

Таким образом, области  $W_1$  и  $W_2$  представляют собой скользящие окна в виде квадрата с центром в элементе с координатами  $(ij)$ . Скользящее окно  $W_1$  размещено внутри скользящего окна  $W_2$ .

Следует отметить, что форма и размеры скользящего окна, в общем случае, могут быть произвольными. В большинстве методов размеры апертуры остаются постоянными на протяжении реализации всего метода. Однако существуют методы обработки изображений с адаптивным скользящим окном. Известно, что степень обработки сигналов зависит от размеров апертуры фильтра, а именно при малом размере апертуры фильтра операции усреднения подвергается меньшее число значений, что обеспечивает лучшее сохранение контрастных деталей сигнала. Но при этом шум будет сглажен хуже. И наоборот, при большом размере апертуры фильтра сглаживание шума будет происходить лучше, но при этом возможна «потеря» некоторых контрастных деталей, присутствующих в исходном сигнале. Таким образом, качество обработки изображения можно улучшить путем выбора наиболее подходящих размеров апертуры. К сожалению, реальные изображения не являются стационарными: они содержат как монотонные области, так и изображения перепада, поэтому на практике очень трудно подобрать оптимальные размеры апертуры фильтра. В подобных ситуациях качество обработки можно улучшить, если использовать апертуру с большими размерами в монотонных областях обрабатываемого изображения и апертуру с малыми размерами вблизи областей перепада.

3. Третий этап состоит в нелинейном усилении локального контраста:

$$g_{ij}^2 = \Phi(g_{ij}^1),$$

где  $\Phi(g_{ij}^1)$  – нелинейная монотонная функция.

За функции  $\Phi(g_{ij}^1)$  принимаются степенные, экспоненциальные, логарифмические и гиперболические функции.

4. На четвертом этапе происходит восстановление элемента изображения с координатами  $(ij)$  скорректированного контрастом  $g_{ij}^2$ . Для этого используют выражение:

$$g_{ij} = \begin{cases} \bar{f}_{ij}^2 \frac{1 - g_{ij}^2}{1 + g_{ij}^2}, & \text{if } \bar{f}_{ij}^1 \leq \bar{f}_{ij}^2, \\ \bar{f}_{ij}^2 \frac{1 + g_{ij}^2}{1 - g_{ij}^2}, & \text{if } \bar{f}_{ij}^1 > \bar{f}_{ij}^2. \end{cases}$$

Экспериментальные исследования и компьютерное моделирование данного метода показали его недостаточную эффективность, поскольку при таком подходе результирующее изображение получается размытым.

Для устранения возможного эффекта размытия изображения можно использовать вместо усредненного значения  $\bar{f}_{ij}^1$  значение центрального элемента  $f_{ij}$ . В этом случае область  $W_1$  вырождается в центральный элемент  $f_{ij}$  и принимает размеры  $n=1$ . Учитывая это, можно записать:

$$g_{ij}^1 = \frac{\left| f_{ij} - \bar{f}_{ij}^2 \right|}{f_{ij} + \bar{f}_{ij}^2}, \quad g_{ij} = \begin{cases} \bar{f}_{ij}^2 \frac{1 - g_{ij}^2}{1 + g_{ij}^2}, & \text{if } f_{ij} \leq \bar{f}_{ij}^2, \\ \bar{f}_{ij}^2 \frac{1 + g_{ij}^2}{1 - g_{ij}^2}, & \text{if } f_{ij} > \bar{f}_{ij}^2. \end{cases}$$

## 6.4. Сегментация изображений посредством выделения линий и контуров

### 6.4.1. Общие принципы

Сегментация – процесс, при котором изображение разбивается на взаимно непересекающиеся множества, называемые областями, каждая из которых называется однородной относительно установленного критерия однородности [13]. Согласно определению, сегментация подразделяет изображение на составляющие его области или объекты. Сегментация сложных изображений представляет собой одну из самых трудных задач обработки изображений. Конечный результат ряда прикладных процедур компьютерной обработки изображений во многом определяется точностью сегментации, следовательно, необходимо использовать лишь эффективные алгоритмы сегментации изображений.

В общем, алгоритмы сегментации изображений основываются на одном из двух базовых свойств сигнала яркости [5]: разрывы и однородность.

В первом случае подход состоит в разбиении изображения на основании резких изменений (разрывов) яркости, которые являются важными простейшими признаками, поскольку они часто определяют очертания изображенных объектов. Локальные разрывы значений яркости называются яркостными перепадами или яркостными контурами. Вторая категория методов использует разбиение изображения на области, однородные с точки зрения выбранных критериев.

На рис. 6.28 [12] показаны участки изображения с быстрым изменением яркости от низкого уровня к высокому в одномерном и двумерном случаях. В одномерном случае перепад характеризуется высотой, углом наклона и координатой центра склона. Перепад существует, если его угол наклона и высота больше некоторого заданного порога. Для двумерного случая важна также ориентация перепада по отношению к оси  $x$ . Идеальный детектор перепада при обработке областей изображения, представленных на рис. 6.28, должен указывать на наличие перепада в единственной точке, расположенной в центре склона.

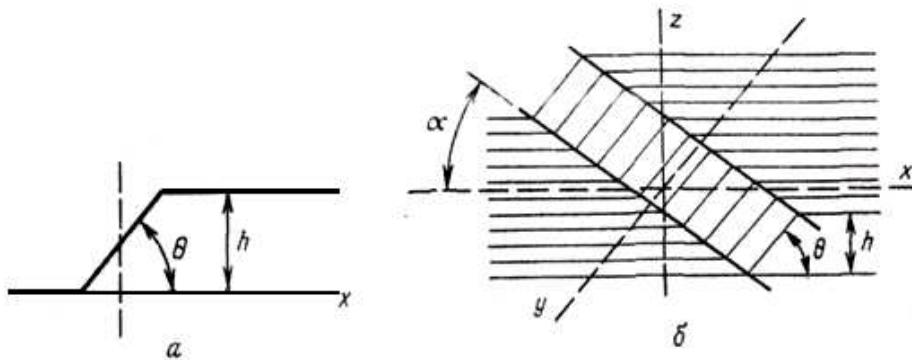


Рис. 6.28. Одномерный (а) и двумерный (б) перепады

Общий подход к обнаружению перепадов на изображении можно сформулировать следующим образом. Исходное изображение представленное цифровой матрицей  $F$  с элементами  $f_{ij}$  подвергается линейной или нелинейной обработке с тем, чтобы усилить перепады яркости. В результате образуется массив чисел  $G$  с элементами  $g_{ij}$ , описывающий изображение с подчеркнутыми изменениями яркостей. Затем выполняется операция сравнения с порогом и определяется положение элементов изображения с ярко выраженными перепадами. Если  $g_{ij} < T_{ij}^L$ , то имеет место нисходящий перепад, а при  $g_{ij} \geq T_{ij}^U$  – восходящий перепад. Величины  $T_{ij}^L$  и  $T_{ij}^U$  представляют собой нижнее и верхнее пороговые значения. Эти значения можно сделать переменными в плоскости изображения для компенсации влияния сильных изменений яркости на результаты обнаружения перепадов.

Выбор порога является одним из ключевых вопросов выделения перепадов. При слишком высоком уровне порога не будут обнаружены структурные элементы с низким контрастом, а при слишком низком – шум будет ложно принят за перепад. Для обозначения положения перепадов на изображении часто формируют *конту르ный препарат – массив элементов*.

Таким образом, перепады яркостей на изображениях являются характеристиками границ областей объектов – контуров. Задача выделения контуров состоит в построении бинарного изображения, содержащего лишь контура. Например, положение точек восходящих препаратов можно отмечать белыми точками на черном фоне.

Следует отметить ряд проблем, возникающих при выделении контуров: в процессе выделения контура объекта могут быть его разрывы в местах, где яркость меняется недостаточно быстро; из-за различных искажений изображения (шума, размытости и т. д.) или неэффективных алгоритмов обработки могут ошибочно обнаруживаться контура «ложных объектов» или формироваться широкие контурные линии; из-за дискретности цифровых изображений контуры представляются линией единичной (а не нулевой) ширины, то есть не являются бесконечно тонкими (имеется неоднозначность в положении контура величиной в один пиксель).

#### 6.4.2. Линейные методы контрастирования

Как правило, принято различать три типа границ (контуров) – вертикальные, горизонтальные и диагональные. Для каждого из этих видов существуют фильтры, которые особенно эффективно выделяют соответствующие контуры. В отличие от задачи обострения контуров (высокочастотный фильтр) здесь основной целью является не улучшение изображения, а наоборот, контуры должны быть отделены от всего изображения так, чтобы выходное изображение состояло только из контуров.

Один из наиболее простых способов контрастирования заключается в вычислении дискретных разностей, что аналогично непрерывному пространственному дифференцированию [5]. Подчеркивание вертикальных перепадов осуществляется горизонтальным (построчным) дискретным дифференцированием (вычислением дискретных производных). В результате формируется некоторое изображение – массив элементов:

$$g_{ij} = f_{ij} - f_{i,j+1}.$$

Аналогично осуществляется подчеркивание горизонтальных перепадов. В результате получается массив элементов:

$$g_{ij} = f_{ij} - f_{i+1,j}.$$

Диагональное подчеркивание можно получить путем вычисления разностей уровней диагональных пар элементов изображения.

Горизонтальное подчеркивание перепада также можно выполнить, вычисляя разности яркостей элементов вдоль строки изображения по формуле:

$$g_{ij} = [f_{ij} - f_{i,j-1}] - [f_{i,j+1} - f_{ij}] \text{ или } g_{ij} = 2f_{ij} - f_{i,j-1} - f_{i,j+1}.$$

Подобные выражения можно записать и для изменения яркости по вертикали и диагонали.

Для обнаружения линий на изображениях можно использовать набор масок, показанный на рис. 6.29 [5,12].

$$H10 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad H11 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad H12 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad H13 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

горизонтальная                          +45°                          вертикальная                          -45°

Рис. 6.29. Маски для обнаружения линий

Название масок соответствует их наиболее сильному отклику при скольжении по изображению. Для первой маски наибольший отклик будет на линии толщиной в один пиксель, которая проходит горизонтально через центр маски, а яркость фона изображения одинакова. Направления масок можно также выявить по расположению максимальных коэффициентов в определенном направлении, например, для горизонтальной маски максимальные коэффициенты (2) расположены по горизонтали и т. д.

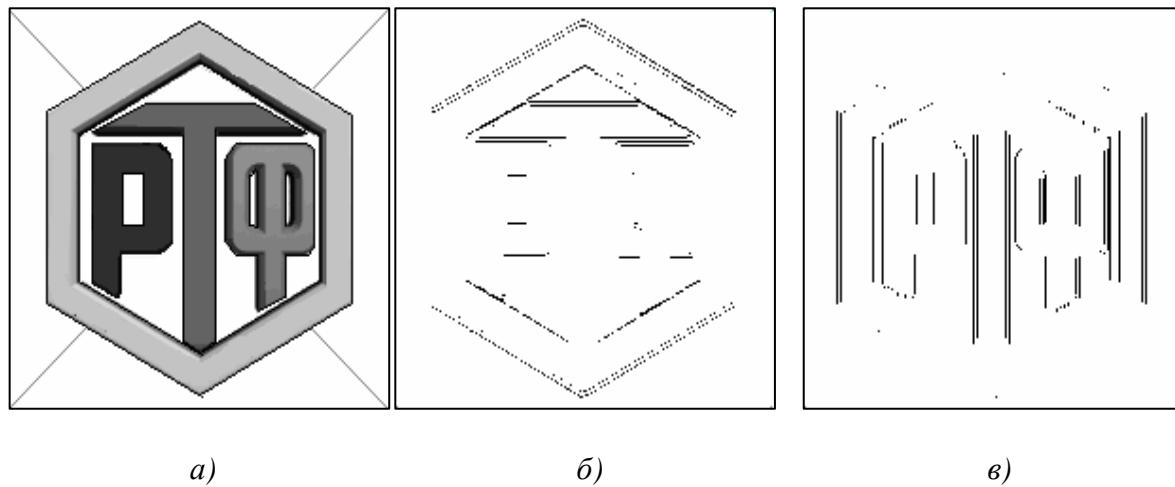
Для выделения всех линий заданного направления необходимо применить маску к изображению, в целом сравнивая абсолютное значение результата отклика маски с заданным порогом. Таким образом, решение о наличии элемента линии на изображении принимается на основе выражения:

$$|R| \geq T,$$

где  $R$  – результат отклика маски;

$T$  – заданный порог.

Оставшиеся при этом точки на изображении и соответствуют наибольшим значениям отклика, которые в случае наличия на изображении линий толщиной в один пиксель наиболее близки к направлению, определяемому маской. Результат работы масок представлен на рис. 6.30.



a)

б)

в)

Рис. 6.30. Пример выделения линий заданного направления:  
 а) исходное изображение; б) результат подчеркивания горизонтальных линий;  
 в) результат подчеркивания вертикальных линий

Для выделения контуров на изображении в заданном направлении используют следующие курсовые градиентные маски (рис. 6.31) [12].

$H10 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$H11 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$H12 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$
Север <b>Юго-восток</b> $H13 = \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	<b>Юг</b> $H14 = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	<b>Юго-запад</b> $H15 = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$
<b>Запад</b>  $H16 = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	<b>Северо-запад</b>  $H17 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	

Рис. 6.31. Маски для выделения перепадов яркости на изображении

Название курса указывает на направление склона перепада яркости, при котором маска дает максимальный отклик. Например, восточная градиентная маска дает максимальный отклик при изменении яркости в горизонтальном направлении слева направо. Градиентные маски обладают нулевым суммарным весом, поэтому в областях изображения с постоянной яркостью они дают нулевой отклик. На рис. 6.32 приведены результаты работы некоторых масок для изображения, представленного на рис. 6.31, а.

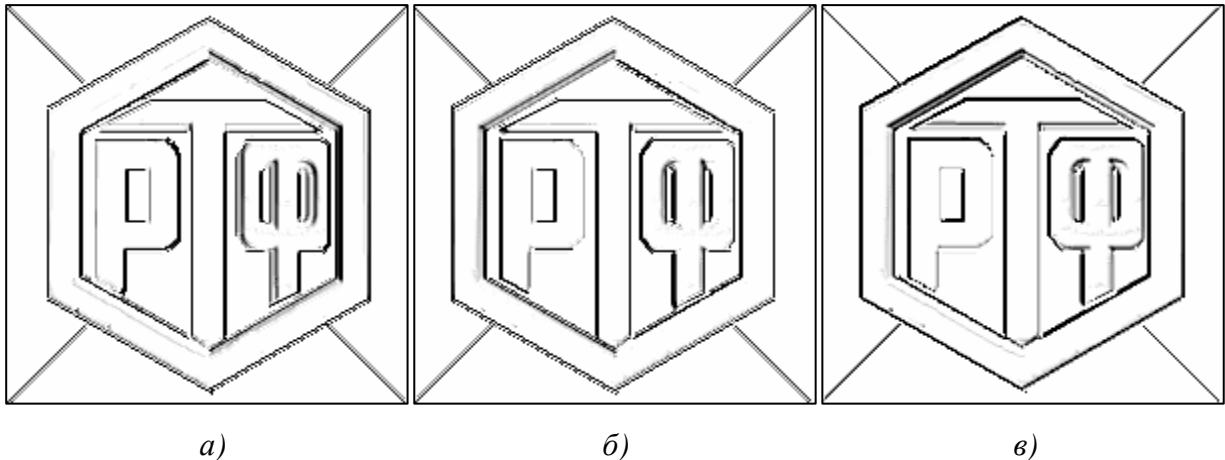


Рис. 6.32. Результаты обработки изображения курсовыми градиентными масками:  
а) результат использования маски «Восток»; б) результат использования маски  
«Запад»; в) результат использования маски «Северо-запад»

Метод усиления края по Лапласу отличается тем, что он не зависит от направления границы, т. е. края определяются независимо от их типа. Усиление края по Лапласу образует более резкий край, чем при других алгоритмах усиления. Дополнительно определяются края, имеющие как положительное, так и отрицательное изменение яркости.

Функция Лапласа в общем виде записывается следующим образом:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2},$$

где  $\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2}$  – вторые частные производные по  $x$  и  $y$  соответственно.

Для дискретных функций вторые производные могут быть аппроксимированы следующим образом:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - 2f(x, y) + f(x-1, y);$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - 2f(x, y) + f(x, y-1).$$

Таким образом, лапласиан можно записать в виде:

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y).$$

Кроме аналитической записи, лапласиан может быть записан и в виде ядра свертки (маски) (рис. 6.33) [5,12]. Результаты применения представлены на рис. 6.34.

$$L1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad L2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad L3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Рис. 6.33. Маски Лапласа для контрастирования

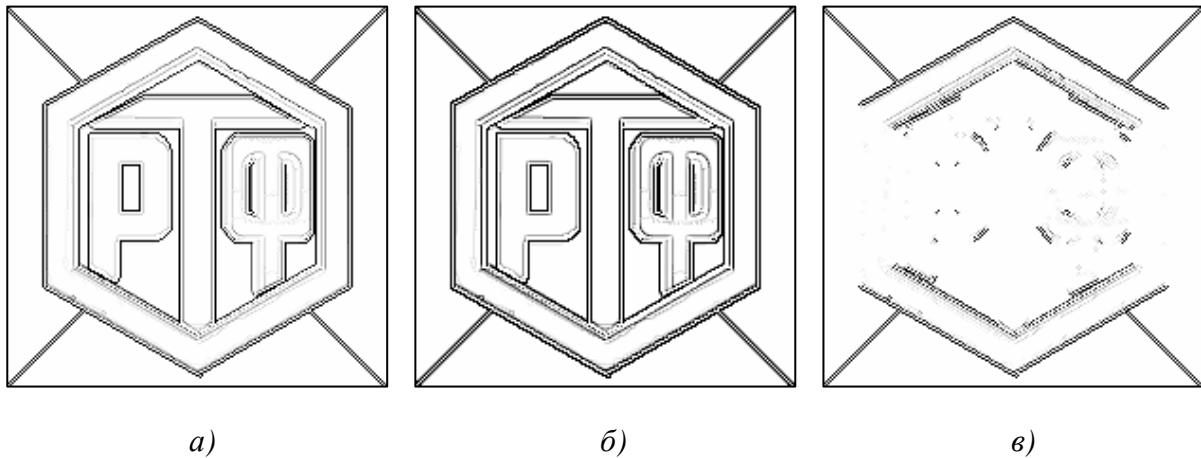


Рис. 6.34. Результат применения масок Лапласа:  
a) результат использования маски  $L1$ ; б) результат использования маски  $L2$ ;  
в) результат использования маски  $L3$

Операторы Лапласа реагируют на перепады яркости в виде ступенчатого перепада и на «крышеобразный» перепад. Они также выделяют изолированные точки, тонкие линии, их концы и острые углы объектов. Линия подчеркивается в 2 раза ярче, чем ступенчатый перепад, конец линии в 3 раза, а точка в 4 раза ярче. Оператор Лапласа не инвариантен к ориентации перепадов: например, отклик оператора на наклонный перепад в диагональном направлении почти вдвое больше, чем в горизонтальном и вертикальном.

#### 6.4.3. Нелинейные методы контрастирования

В нелинейных системах обнаружения перепадов для контрастирования перед пороговым ограничением используются нелинейные комбинации значений яркости элементов изображения. В большинстве методов ог-

раничиваются обработкой окном размером  $2 \times 2$  или  $3 \times 3$ . Среди существующих нелинейных методов можно выделить алгоритмы: Робертса, Превитта, Собела, Кирша и Уоллиса [5,12].

Ряд нелинейных операторов выделения контуров использует вычисление модуля градиента яркости:

$$\nabla f = g_{ij} = \sqrt{X^2 + Y^2},$$

где  $X$  и  $Y$  – частные производные, характеризующие скорость изменения яркости по двум направлениям.

Роберт для выделения перепадов предложил следующую операцию нахождения величин  $X$  и  $Y$ :

$$\begin{aligned} X &= f_{ij} - f_{i+1,j+1}, \\ Y &= f_{i,j+1} - f_{i+1,j}. \end{aligned}$$

При построении оператора Робертса используется тот факт, что для определения перепада яркости можно использовать производные (разности) в любых двух взаимно перпендикулярных направлениях. Отмечая тот из четырех элементов изображения, расположенных около обнаруженной точки, который имеет наибольшее значение яркости, можно получить информацию о приблизительной ориентации перепада.

Реализация масок размером  $2 \times 2$  неудобна, так как у них нет четко выраженного центрального элемента. Метод, при котором используется маски размером  $3 \times 3$ , предложил Превитт. Алгоритм используется модуль вектора градиента и для нахождения величин  $X$  и  $Y$  применяет выражения:

$$\begin{aligned} Y &= (f_{(i-1,j+1)} + f_{(i,j+1)} + f_{(i+1,j+1)}) - (f_{(i-1,j-1)} + f_{(i,j-1)} + f_{(i+1,j-1)}), \\ X &= (f_{(i-1,j-1)} + f_{(i-1,j)} + f_{(i-1,j+1)}) - (f_{(i+1,j-1)} + f_{(i+1,j)} + f_{(i+1,j+1)}). \end{aligned}$$

Собел модифицировал данный алгоритм и предложил использовать весовой коэффициент 2 для средних элементов, что позволяет уменьшить эффект сглаживания за счет придания большего веса средним точкам. Выражения для нахождения величин  $X$  и  $Y$  в операторе Собела имеют вид:

$$\begin{aligned} Y &= (f_{(i-1,j+1)} + 2f_{(i,j+1)} + f_{(i+1,j+1)}) - (f_{(i-1,j-1)} + 2f_{(i,j-1)} + f_{(i+1,j-1)}), \\ X &= (f_{(i-1,j-1)} + 2f_{(i-1,j)} + f_{(i-1,j+1)}) - (f_{(i+1,j-1)} + 2f_{(i+1,j)} + f_{(i+1,j+1)}). \end{aligned}$$

$$H18 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad H19 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Рис. 6.35. Маски оператора Робертса

$$H20 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

a)

$$H21 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

б)

$$H22 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

6

$$H23 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2)

Рис. 6.36. Маски нелинейных операторов:  
 $a, b$  – маски Превитта;  $c, \varepsilon$  – маски Собела

Представленные маски применяются для получения составляющих градиента  $X$  и  $Y$ . Для вычисления величины градиента эти составляющие, а соответственно, и маски, необходимо использовать совместно. Для уменьшения объема вычислений можно применять другие приближенные выражения:

$$g_{ij} = \max\{|X|, |Y|\}.$$

Для оператора Робертса приближенные выражения не являются одинаково чувствительными к границам с любой ориентацией. Для строго вертикальных или горизонтальных линий все выражения дают одинаковые результаты, но для линий с наклоном  $45^\circ$  приближенные значения могут отличаться от точного в  $\sqrt{2}$  раз. Для операторов Превитта и Собела вопрос изотропности не возникает, так как сами маски инвариантны лишь для поворотов на углы,  $90^\circ$ .

Следует отметить, что операторы Превитта и Собела можно изменить так, чтобы они давали максимальный отклик для контуров, направленных диагонально (маски представлены на рис. 6.37).

$$H24 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

a)

$$H25 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

б)

$$H26 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

6

$$H23 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

2)

Рис. 6.37. Нелинейные маски для выделения диагональных контуров:  
 $a, b$  – маски Превитта;  $c, g$  – маски Собела

Нелинейный оператор, предложенный Киршем, определяется следующим выражением:

$$g_{ij} = \max\{1, \max_{k=0 \div 7} [ |5S_k - 3T_k| ]\},$$

где  $S_k = A_k + A_{k+1} + A_{k+2}$ ,

$$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7}.$$

$A_0$	$A_1$	$A_2$
$A_7$	$f_{(i,j)}$	$A_3$
$A_6$	$A_5$	$A_4$

Рис. 6.38. Обозначение элементов для оператора Кирша

Индексы у слагаемых  $A_k$  подсчитываются по модулю 8. По существу оператор Кирша дает максимальное значение курсового градиента в некоторой точке изображения без учета значения  $f_{ij}$ .

Уоллис предложил нелинейный метод обнаружения перепадов, основанный на гомоморфной обработке изображения. Согласно этому методу, точка находится на перепаде, если величина логарифма от яркости в этой точке превосходит среднее значение логарифмов яркостей четырех ближайших соседних элементов на некоторое фиксированное значение. Элемент контрастированного изображения определяется как:

$$g_{ij} = \log(f_{ij}) - \frac{1}{4} \log(f_{i-1,j}) - \frac{1}{4} \log(f_{i,j+1}) - \frac{1}{4} \log(f_{i+1,j}) - \frac{1}{4} \log(f_{i,j-1}),$$

или, что эквивалентно:

$$g_{ij} = \frac{1}{4} \log \left\{ \frac{f_{ij}^4}{f_{i-1,j} f_{i,j+1} f_{i+1,j} f_{i,j-1}} \right\}.$$

Сравнение  $g_{ij}$  с верхним и нижним пороговыми значениями эквивалентно сравнению дроби в скобках выражения с видоизмененным порогом. Поэтому не требуется точно вычислять значения логарифмов. Основное преимущество логарифмического детектора перепадов состоит в том, что он не чувствителен к мультипликативным изменениям уровня яркости.

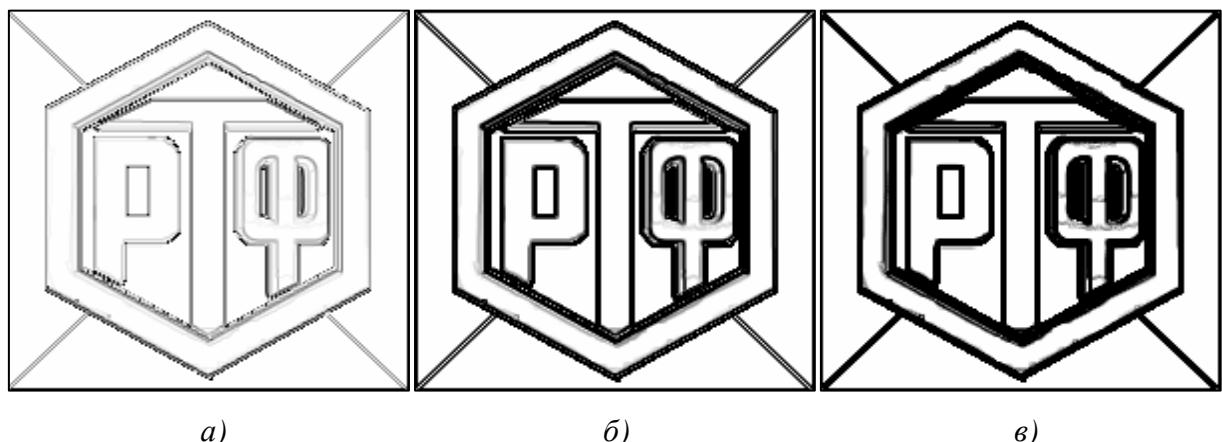


Рис. 6.39. Результаты применения:

а) оператора Робертса; б) оператора Собела; в) оператора Кирша

Эффективность детекторов обнаружения перепадов яркостей трудно оценить вследствие большого числа предложенных методов, из-за трудностей в определении наилучших параметров, связанных с каждым методом, а также из-за отсутствия единого критерия эффективности [12].

При разработке критерия эффективности для детекторов перепада целесообразно делать различие между обязательной и вспомогательной информацией, которую необходимо получить с помощью таких детекторов. Очевидно, совершенно необходимо определять положение перепада. Другая представляющая интерес информация включает высоту и крутизну перепада яркости, а также его пространственную ориентацию. Полезной информацией является также достоверность решения о перепаде, например о степени соответствия фрагмента истинного изображения модели идеального перепада. К сожалению, имеется немногих детекторов, дающих этот полный набор данных.

Существуют три основных типа ошибок, связанных с определением положения перепадов яркости [12,16]: 1) пропуск истинных перепадов; 2) ошибка в определении их положения; 3) принятие шумовых выбросов за перепад (ложное обнаружение).

Общая стратегия в задачах обнаружения сигналов заключается в установлении некоторой границы вероятности ложного обнаружения, являющегося следствием воздействия шума, с последующей максимизацией вероятности правильного обнаружения. Распространение этой идеи на обнаружение перепадов ведет к установлению порога обнаружения на таком уровне, при котором вероятность ложного обнаружения, обусловленного одним лишь шумом, не превосходит некоторого заданного значения. Вероятность обнаружения истинного перепада можно легко вычислить, сравнивая контурные изображения, полученные с помощью идеального и реального детекторов перепада. Определение величины штрафа за ошибки в локализации перепадов представляет собой более трудную задачу. Ясно, что детекторы, дающие «смазанный» перепад, должны быть «оштрафованы». Следует отдавать предпочтение детекторам, которые локализуют перепады, хотя и дают при этом небольшую ошибку. Точность в определении положения перепада можно оценить величиной [12,16]:

$$R = (1 / I_N) \sum_{i=1}^{I_A} 1 / (1 + \alpha d^2),$$

где  $I_N = \max(I_I, I_A)$ ;

$I_I$  и  $I_A$  – число точек перепадов в идеальном и реальном контурных препаратах;

$\alpha$  – масштабный множитель;

$d$  – расстояние между точкой действительного перепада и линией, состоящей из точек идеального перепада, измеренное по нормали к этой линии.

Значение критерия нормализовано так, что  $R = 1$  для точно выделенного перепада. Масштабный множитель можно подобрать таким образом, чтобы установить штрафы для перепадов, истинное положение которых отличается от обнаруженного. Множитель  $1/I_N$  обеспечивает штраф за слизанные или разбитые контуры. Например, если  $\alpha = 1/9$ , то при обнаружении вертикального перепада, отстоящего на один элемент изображения относительно истинного положения  $R = 0,90$ , сдвиг на два элемента приводит к значению  $R=0,69$ . При  $\alpha = 1/9$  слизанный контур шириной в три элемента изображения, центр которого совпадает с центром истинного вертикального перепада, дает значение  $R = 0,93$ , а при ширине в пять элементов  $R = 0,84$ . Более высокое значение  $R$  для слизанного контура по сравнению со значением для сдвинутого перепада представляется разумным, поскольку имеется возможность путем последующей обработки сделать слизанный контур более тонким.

Некоторые детекторы позволяют определять высоту, крутизну и ориентацию перепада. Для этих детекторов полезно определить среднеквадратическое отклонение этих параметров от их истинных значений, усредненное по всем истинным точкам перепадов, которые совпадают с обнаруженными точками.

## **6.5. Связывание контуров. Представление и описание изображений**

На практике идеально выделить контур достаточно сложно по причине шумов, разрывов самих контуров из-за неоднородности освещения, а также прочих эффектов, нарушающих непрерывность яркостной картины. Поэтому алгоритмы обнаружения контуров обычно дополняются процедурами связывания, чтобы сформировать из множества контурных точек содержательные контуры.

### **6.5.1. Локальная обработка**

Один из простейших подходов к связыванию точек контура состоит в анализе характеристик пикселей в небольшой окрестности каждой точки  $(i, j)$  изображения, которая была отмечена как контурная точка с помощью какого-либо из известных алгоритмов. Все точки, являющиеся сходными в соответствии с некоторыми заранее заданными критериями, связываются и образуют контур, состоящий из пикселей, отвечающих этим критериям.

При таком анализе используются следующие два основных параметра для установления сходства пикселей контура: величина отклика оператора градиента, определяющая значение пикселей контура; направление вектора градиента.

Первый параметр задается значением градиента изображения [5]:

$$\nabla f = |\nabla f| = \sqrt{X^2 + Y^2},$$

где  $X$  и  $Y$  – частные производные, характеризующие скорость изменения яркости по двум направлениям.

Величину  $\nabla f$  также называют модулем градиента.

Таким образом, пиксель контура, имеющий координаты  $(i_0, j_0)$  и расположенный внутри заданной окрестности точки  $(i, j)$ , считается сходным по модулю градиента с пиксelem  $(i, j)$ , если

$$|\nabla f_{ij} - \nabla f_{i_0j_0}| \leq E,$$

где  $E$  – заданный неотрицательный порог.

Направление (угол) вектора градиента задается выражением:

$$\alpha_{ij} = \arctg\left(\frac{Y}{X}\right).$$

Пиксель контура с координатами  $(i_0, j_0)$ , расположенный внутри заданной окрестности точки  $(i, j)$ , считается сходным по направлению градиента с пиксelem  $(i, j)$ , если:

$$|\alpha f_{ij} - \alpha f_{i_0j_0}| \leq A,$$

где  $A$  – заданный неотрицательный угловой порог.

Направление контура в точке  $(i, j)$  *перпендикулярно* направлению вектора градиента в этой точке.

Пиксель в заданной окрестности объединяется с центральным пиксelem  $(i, j)$ , если выполнены критерии сходства и по величине, и по направлению. Этот процесс повторяется в каждой точке изображения, с одновременным запоминанием найденных связанных пикселей при движении центра окрестности. Простой способ учета данных состоит в том, что каждому множеству связываемых пикселей контура присваивается свое значение яркости.

### 6.5.2. Глобальный анализ с помощью преобразования Хафа

Пусть задано  $n$  точек на изображении (плоскости). Предположим, что требуется найти подмножества этих точек, лежащие на прямых линиях. Одно из возможных решений этой задачи состоит в том, что в начале находятся все прямые, определяемые каждой парой точек, и затем ищутся все подмножества точек, близких к конкретным прямым. Трудности реализации такой процедуры вызываются необходимостью нахождения  $n / (n - 1) / 2 \sim n^2$  прямых, а затем выполнения  $n / (n - 1) / 2 \sim n^3$  сравнений каждой точки со всеми прямыми. Вычислительная сложность данного подхода позволяет применить его лишь в самых тривиальных прикладных задачах.

Хаф предложил альтернативный подход (который принято называть *преобразованием Хафа*) [5]. Возьмем точку  $(x_i, y_i)$  из заданного множества  $n$  точек и рассмотрим общее уравнение прямой на плоскости в форме с угловым коэффициентом:  $y = ax + b$ . Очевидно, что через точку  $(x_i, y_i)$  проходит бесконечно много прямых, удовлетворяющих уравнению  $y_i = ax_i + b$  при различных значениях  $a$  и  $b$ . Однако если переписать это уравнение в виде  $-b = -x_i a + y_i$  и рассмотреть плоскость  $ab$ , называемую *пространством параметров*, то для заданной пары получаем уравнение *единственной* прямой. Более того, другой точке  $(x_j, y_j)$  также соответствует своя прямая в пространстве параметров, и эти две прямые пересекаются в некоторой точке  $(a', b')$ , такой, что  $a'$  – угловой коэффициент, а  $b'$  – точка пересечения с осью  $y$  прямой, проходящей через точки  $(x_i, y_i)$  и  $(x_j, y_j)$  в плоскости  $xy$ . На самом деле, каждой точке прямой, проходящей через точки  $(x_i, y_i)$  и  $(x_j, y_j)$ , в пространстве параметров соответствует своя прямая линия, причем все они пересекаются в точке  $(a', b')$  (рис. 6.40).

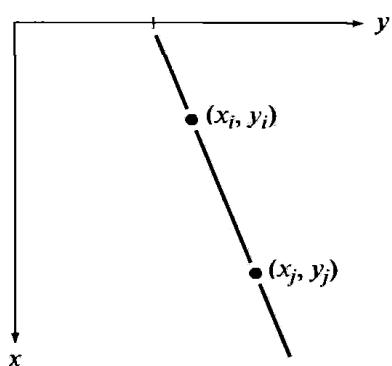


Рис. 6.39. Плоскость  $xy$

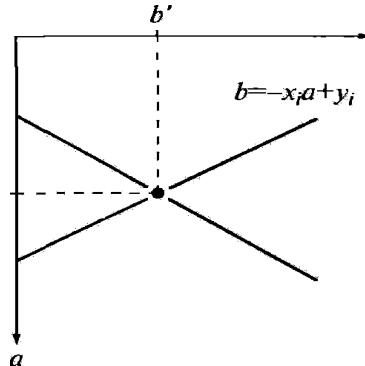


Рис. 6.40. Пространство параметров

Привлекательность преобразования Хафа с точки зрения вычислений проистекает из возможности разбиения пространства параметров на так называемые **ячейки накопления** (рис. 6.41), где  $(a_{\max}, a_{\min})$  и  $(b_{\max}, b_{\min})$  – предполагаемые диапазоны возможных значений углового коэффициента и ординаты точки пересечения прямой с осью  $y$ . В ячейке  $(p, q)$  накапливается значение  $A(p, q)$  для прямоугольника в пространстве параметров, соответствующего точке  $(a_p, b_q)$ . Первоначально значения во всех ячейках накопления равны нулю. Затем для каждой точки  $(x_i, y_i)$  из заданного множества  $n$  точек в плоскости изображения полагаем параметр  $a$  равным поочередно каждому дискретному значению  $a_p$  в разрешенном интервале на оси  $a$  и находим соответствующее ему значение  $b$ , решая уравнение  $b = -x_i a + y_i$ . После этого найденное значение округляется до ближайшего дискретного значения на оси  $b$ . Если выбор значения  $a_p$  приводит к допустимому решению  $b_q$ , увеличиваем накопленное значение в соответствующей ячейке:  $A(p, q) = A(p, q) + 1$ . После выполнения описанной процедуры для всех анализируемых точек  $(x_i, y_i)$ , записанное в ячейке  $(p, q)$  значение  $A(p, q) = Q$  означает, что в плоскости  $xy$  имеется  $Q$  точек, лежащих на прямой  $y = a_p x + b_q$ . Точность попадания точек на эту прямую определяется размерами ячеек накопления на плоскости  $ab$ .

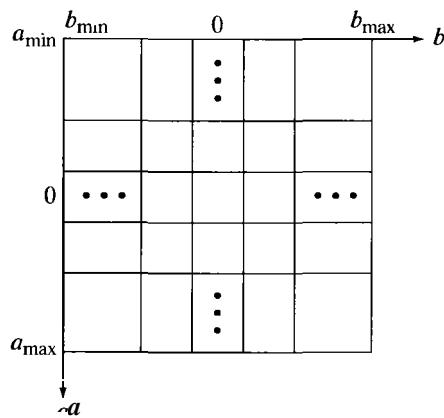


Рис. 6.41. Разбиение плоскости параметров для применения преобразования Хафа

Разбиение оси  $a$  на  $K$  дискретных значений дает для каждой точки  $(x_i, y_i)$  значений параметра  $b$ , соответствующих  $K$  возможным значениям  $a$ . Для  $n$  точек заданного множества данный метод требует выполне-

ния  $nK$  вычислений. Таким образом, трудоемкость рассмотренной процедуры линейна по  $n$ , что намного меньше количества вычислений, требуемых для указанного в начале этого раздела прямого метода (если только  $K$  не оказывается сопоставимым с  $n$  по порядку величины).

Если прямая близка к вертикали, то использование уравнения  $y = ax + b$  для ее представления затруднительно, поскольку при этом угловой коэффициент стремится к бесконечности. Один из способов обойти эту трудность состоит в представлении прямой с помощью нормали:

$$x \cos \theta + y \sin \theta = \rho.$$

Рис. 6.42 [5] дает геометрическую интерпретацию параметров, использованных в выше представленном уравнении. Такое представление используется при построении таблицы ячеек накопления точно так же, как и в случае уравнения в форме с угловым коэффициентом. Однако, вместо прямых линий, геометрические места точек, лежащих на одной прямой, в плоскости  $\rho\theta$  представляют собой синусоидальные кривые. Как и ранее,  $Q$  точек, лежащих на прямой  $\theta_q + y \sin \theta_q = \rho_q$ , порождают в пространстве параметров  $Q$  синусоидальных кривых, пересекающихся в точке  $(\rho_p, \theta_q)$ . Если последовательно придавать  $\theta$  всевозможные дискретные значения и находить соответствующие им значения  $\rho$ , будет выполнено  $Q$  приращений содержимого ячейки накопления  $(\rho_p, \theta_q)$ , отвечающей параметрам  $(\rho_p, \theta_q)$ . Рис. 6.42, б иллюстрирует способ разбиения пространства параметров в этом случае.

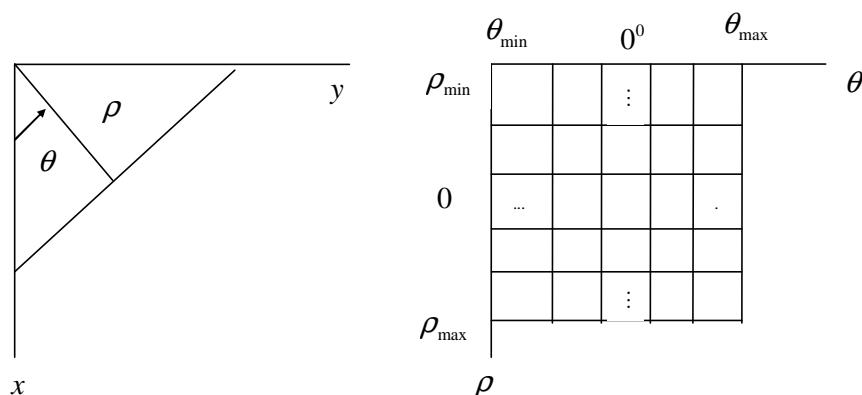


Рис. 6.42: а) представление прямой с помощью нормали;  
б) разбиение плоскости  $\rho\theta$  на ячейки накопления

Диапазон значений углов  $\theta$  составляет  $\pm 90^\circ$ ; угол измеряется по отношению к оси  $x$ . Таким образом, в соответствии с рис. 6.42, *a*, для горизонтальной прямой  $0 = 0^\circ$ , а  $p$  будет равно координате точки ее пересечения с осью  $x$ . Аналогично, для вертикальной прямой  $p$  есть координата точки ее пересечения с осью  $y$ , а  $\theta = 90^\circ$  при  $p > 0$  или  $\theta = -90^\circ$  при  $p < 0$ .

Преобразование Хафа можно применить к любой функции вида  $g(v, e) = 0$ , где  $v$  – вектор координат, а  $e$  вектор коэффициентов. Например, точки, лежащие на окружности:

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2,$$

также могут обнаруживаться описанным выше методом.

Основное отличие состоит в увеличении числа параметров до трех  $c_1, c_2, c_3$ , что приводит к трехмерному пространству параметров с кубическими ячейками, накапливаемые значения в которых имеют вид  $A(p, q, r)$ . Процедура состоит в последовательном присваивании параметрам  $c_1$  и  $c_2$  всех сочетаний допустимых дискретных значений, нахождением для каждой пары значения  $c_3$ , которое бы удовлетворяло выше представленному уравнению, и увеличением накопленного значения в ячейке, отвечающей тройке  $(c_1, c_2, c_3)$ . Ясно, что сложность преобразования Хафа пропорциональна числу координат и коэффициентов в данном функциональном представлении. Возможны дальнейшие обобщения преобразования Хафа на случаи обнаружения кривых, не имеющих простого аналитического представления, как в случае применения этого преобразования к полуточновым изображениям.

Связывание контуров на основе преобразования Хафа заключается в следующем [5]:

1. Вычисляется модуль градиента изображения в каждой точке, который подвергается пороговому преобразованию, в результате чего формируется бинарное изображение.

2. Выполняется разбиение (дискретизация) пространства параметров  $\rho\theta$  на ячейки накопления.

3. Для всех ненулевых пикселей двоичного изображения, полученного в п. 1, находятся образы в пространстве параметров  $\rho\theta$  и осуществляется процедура накопления.

4. Анализируются накопленные значения и отыскиваются ячейки с наибольшей концентрацией точек.

5. Исследуются отношения между пикселями изображения, отвечающих выбранным ячейкам накопления (в основном на предмет их связности).

Понятие связности в данном случае обычно базируется на вычислении расстояний между несвязными пикселями, обнаруженными при обходе множества элементов изображения, соответствующего данной ячейке накопления. Разрыв в некоторой точке считается значимым, если расстояние между этой точкой и ее ближайшим соседом превышает заданный порог.

### 6.5.3. Представление и описание

После сегментации изображения на области, с помощью выше рассмотренных алгоритмов, полученные совокупности пикселей обычно описываются и представляются в форме, удобной для последующей компьютерной обработки. По существу, при выборе способа представления областей возникает следующая альтернатива: область можно представить ее внешними характеристиками (т. е. границей) или область можно представить внутренними характеристиками (т. е. совокупностью элементов изображения, составляющих эту область). Однако выбор способа представления является только частью задачи преобразования данных в форму, удобную для компьютерной обработки. Следующая задача состоит в том, чтобы описать область, исходя из выбранного способа представления. Например, область может быть представлена своей границей, а граница описана с помощью таких характеристик, как длина границы, направления прямых, соединяющих угловые точки, и число вогнутостей границы.

Внешнее представление обычно выбирается в тех случаях, когда основное внимание обращено на характеристики формы области. Внутреннее представление выбирается, если интерес представляют свойства самой области, например, цвет и текстура. Иногда приходится использовать оба способа представления одновременно. В любом случае, выбранные для описания признаки (дескрипторы) должны быть как можно менее чувствительными к изменению размеров области и ее перемещению по полю изображения (сдвиг, поворот).

*Цепные коды.* С помощью цепных кодов граница представляется в виде последовательности соединенных отрезков, для которых указаны длина и направление. Как правило, такое представление основывается на отрезках с 4- или 8-связностью. Направление каждого отрезка кодируется числом в соответствии со схемой нумерации, например, как изображено на рис. 6.43 [5].

Цифровые изображения обычно регистрируются и обрабатываются в виде сетки с одинаковым шагом дискретизации в направлениях осей  $x$  и  $y$ , поэтому цепной код можно построить путем прослеживания границы в направлении по часовой стрелке, присваивая номера направлений отрезкам, соединяющим каждую пару пикселей. В общем случае, такой метод является неприемлемым, т. к. получаемая цепочка кодов оказывается слишком длинной, и любые малые возмущения вдоль границы области, вызванные наличием шума или несовершенством алгоритма сегментации, приводят к изменениям в кодовой последовательности, которые не связаны с общей формой границы.

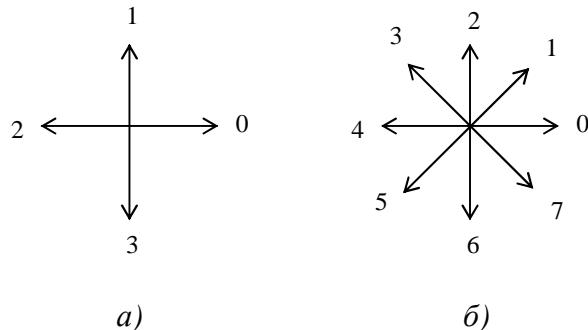


Рис. 6.43. Нумерация направлений:

*а)* для четырехсвязного цепного кода; *б)* для восьмисвязного цепного кода

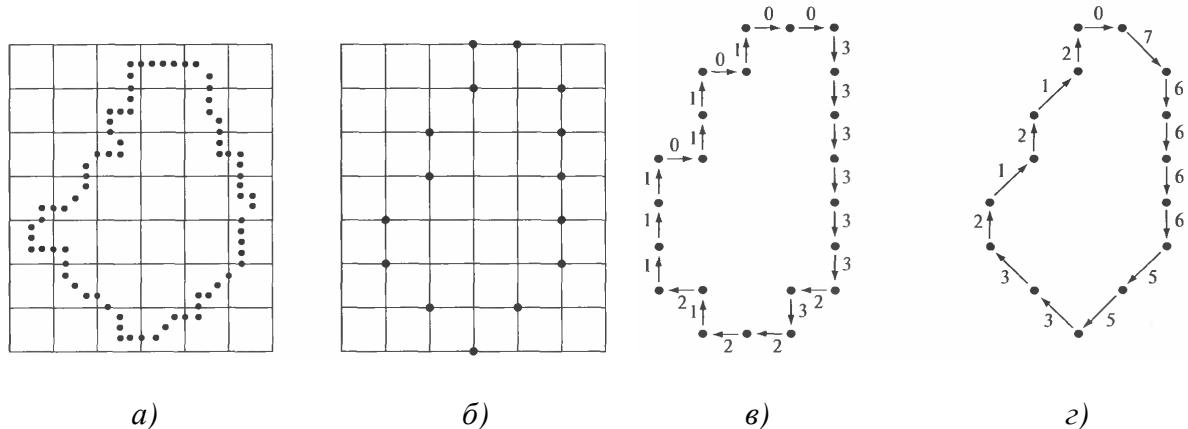


Рис. 6.44. Пример формирования цепного кода:

- а)* дискретная граница с наложенной укрупненной сеткой дискретизации;
- б)* результат новой дискретизации; *в)* цепной код с 4-мя направлениями;
- г)* цепной код с 8-мью направлениями

На практике применяется подход, использующий повторную дискретизацию границы с увеличенным шагом сетки (рис. 6.44, *а*). После этого в процессе обхода границы строятся отрезки, соединяющие узлы укрупненной сетки, выбираемые по признаку их близости к первоначальной границе, что ил-

люстрирует рис. 6.44, б. Полученная таким способом граница с пониженным разрешением может затем представляться с помощью четырех или восьмивсвязного кода, показанных на рис. 6.44, в, г соответственно. В качестве начальной точки на рис. 6.44, в, г произвольно выбран левый верхний угол границы, а сама граница представляет собой кратчайший допустимый четырех- или восьмивсвязный путь на сетке из рис. 6.44, б. Граница на рис. 6.44, в представляется цепным кодом 0033...01, а на рис. 6.44, г – кодом 0766...12. Точность представления границы полученным кодом определяется шагом дискретизации новой сетки.

Цепной код границы области зависит от начальной точки, но его можно сделать инвариантным к ее выбору. Для этого цепной код рассматривается как циклическая последовательность номеров направлений отрезков, и начальная точка переопределяется таким образом, чтобы при начале отсчета с нее получалась линейная запись, соответствующая целому числу наименьшей величины. Цепной код можно сделать инвариантным относительно поворота, если вместо самого кода рассматривать его первую разность, которая формируется путем вычитания значений направления для всех пар соседних элементов кодовой последовательности. Каждая разность вычисляется циклически (против часовой стрелки), так что, например, для цепного кода с 4 направлениями 10103322 первая разность задается последовательностью 3133030. Если и сам код трактовать как циклическую последовательность, то первый элемент разностной последовательности должен вычисляться как циклическая разность последнего и первого элемента исходного цепного кода, и тогда первой разности отвечает код 33133030. Нормировка по размерам области может быть достигнута путем изменения шага сетки, применяемой для новой дискретизации.

Такие способы нормировки будут точными только в том случае, если границы сами по себе инвариантны относительно поворота и изменения масштаба, что редко достигается на практике. Например, при дискретизации одного и того же объекта в двух различных ориентациях получаются границы отличающейся формы, причем степень рассогласования тем больше, чем выше разрешение изображения. Этот эффект можно уменьшить, выбирая длину элементов цепи большей, чем расстояния между пикселями оцифрованного изображения, а также ориентируя сетку вторичной дискретизации вдоль главных осей описываемого объекта.

*Сигнатуры.* Сигнатура есть описание границы объекта с помощью одномерной функции, которое может строиться различными способами [5]. Один из простейших состоит в построении зависимости расстояния от цен-

троида (т. е. от некоторой средней точки объекта, например, его центра тяжести) до границы объекта в виде функции угла, как иллюстрирует рис. 6.45. Независимо от способа построения сигнатуры, основная идея состоит в том, чтобы свести представление границы к одномерной функции, которую предположительно описать легче, чем исходную двумерную границу.

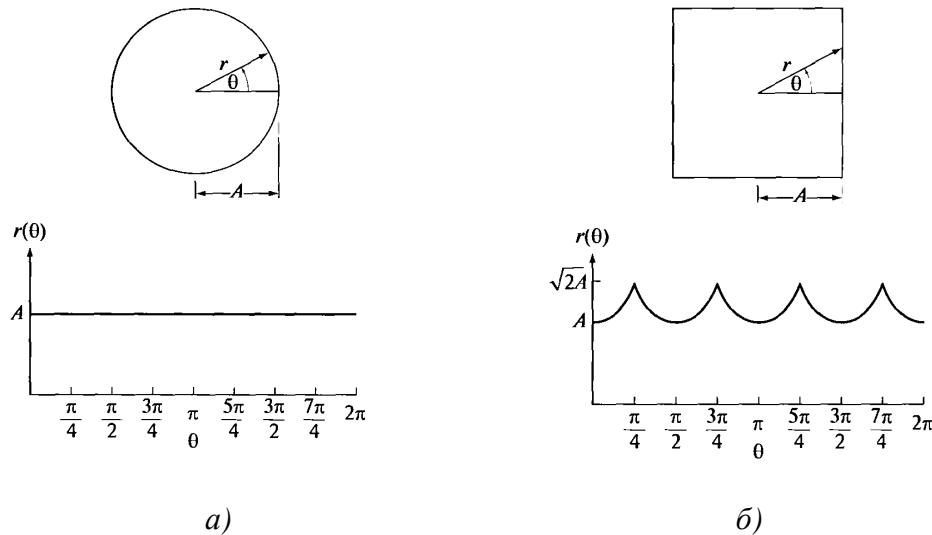


Рис. 6.45. Сигнатуры «угол-расстояние»: а) в случае  $r(\theta) = \text{const}$  ;  
б) сигнатаура состоит из повторяющихся  $\pi / 2$  участков

Сигнатуры, рассмотренные выше, инвариантны по отношению к параллельному переносу, однако зависят от поворота и изменения масштаба. Инвариантность к повороту можно получить, найдя способ выбора одной и той же начальной точки для построения сигнатуры, независимо от ориентации фигуры. Например, выбирать в качестве начальной точки максимально удаленную от центроида, если она оказывается единственной и не зависящей от искажений, возникающих при поворотах интересующих фигур. Другой способ заключается в выборе максимально удаленной от центроида точки на собственной оси фигуры. Такой метод требует большего объема вычислений, но является более устойчивым, т. к. направление собственной оси фигуры определяется с учетом всех точек ее контура.

*Дескрипторы.* Одним из простейших дескрипторов границы является ее длина. Общее число пикселей границы является грубым приближением ее длины. Для кривой, представленной цепным кодом с единичными шагами дискретизации по обоим направлениям, сумма числа вертикальных, горизонтальных и умноженных на  $\sqrt{2}$  диагональных составляющих, дает точное значение длины границы.

Диаметр границы  $B$  определяется соотношением:

$$Diam(B) = \max_{i,j} [D(p_i, p_j)],$$

где  $D$  – мера расстояния,

$p_i, p_j$  – суть точки границы.

Дескрипторами границы являются значение ее диаметра и направление отрезка, соединяющего две экстремальные точки, которые определяют диаметр (этот отрезок называется большой осью границы). Малая ось границы определяется как отрезок, перпендикулярный большой оси и имеющий такую (минимальную) длину, что проведенный через концы обеих осей прямоугольник со сторонами, параллельными этим осям, полностью содержит в себе всю границу. Такой прямоугольник называется базовым прямоугольником, а отношение длины большой оси к длине малой – эксцентризитетом границы, величина которого также является дескриптором.

На рис. 6.46 приведена К-точечная дискретная граница на плоскости  $xy$ . Начиная с ее произвольной точки  $(x_0, y_0)$ , обход границы выполняется против часовой стрелки, и координаты встречающихся точек границы обозначаются как  $(x_0, y_0), \dots, (x_{K-1}, y_{K-1})$ . Границу можно представить в виде последовательности координатных пар  $s_K = (x_k, y_k)$ , где  $k = 0, \dots, K - 1$ . Далее, каждую пару координат можно рассматривать как комплексное число так, что:

$$s_k = x_k + jy_k.$$

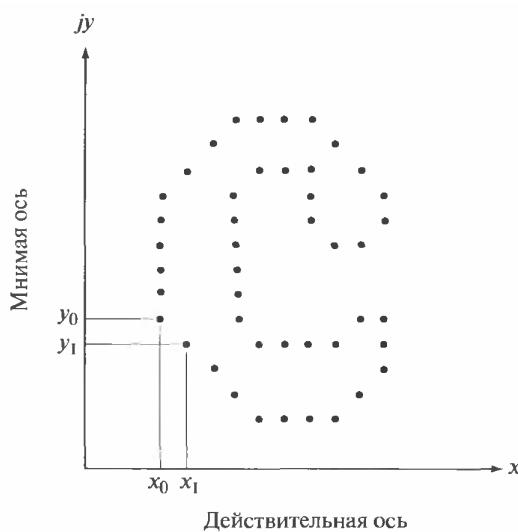


Рис. 6.46. Дискретная граница и ее представление в виде комплексной последовательности

Таким образом,  $x$  и  $y$  рассматриваются как действительная и мнимая оси для последовательности комплексных чисел. Несмотря на изменившийся способ интерпретации этой последовательности, сущность границы осталась прежней. Такое представление имеет одно крупное преимущество: оно позволяет свести двумерную задачу к одномерной.

Как известно, одномерное ДПФ может быть определено на основе выражения:

$$a_u = \frac{1}{K} \sum_{k=0}^{K-1} s_k e^{j2\pi uk/K}, \quad u \in 0, \dots, K-1.$$

Комплексные коэффициенты  $a_u$  называются фурье-дескрипторами границы [5]. Обратное преобразование Фурье, примененное к этим коэффициентам, позволяет восстановить границу.

Если вместо всех коэффициентов Фурье используются только первые  $P$  из них. Это равносильно тому, что в выше представленном уравнении принимается  $a_u = 0$  при  $u > P - 1$ . Хотя при вычислении каждой компоненты используется лишь  $P$  членов,  $k$  по-прежнему пробегает весь диапазон от 0 до  $K-1$ , т. е. в приближенной границе будет то же самое число точек, хотя для восстановления их координат используется меньшее число членов. Известно, что высокочастотные составляющие описывают мелкие детали, тогда как низкочастотные компоненты определяют общую форму границы. Следовательно, чем меньше  $P$ , тем больше деталей границы теряется.

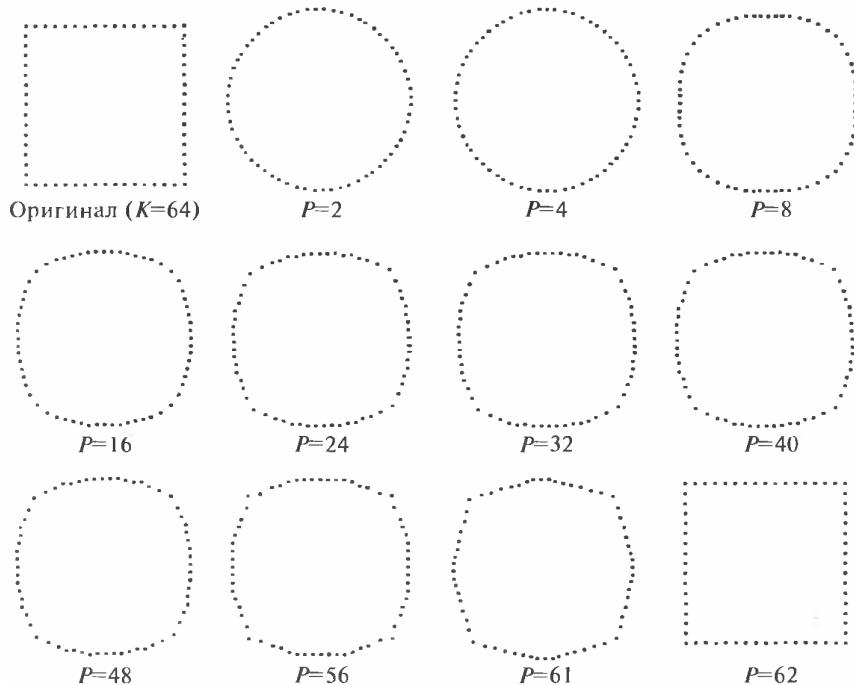


Рис. 6.47. Примеры восстановления границы по Фурье-дискрипторам

На рис. 6.47 изображена граница квадратной формы, состоящая из  $K = 64$  точек, а также результаты ее восстановления с помощью обратного преобразования Фурье при разных значениях  $P$ . Очевидно, что лишь начиная со значений  $P$  около 8, восстановленная граница становится больше похожей на квадрат, чем на круг. Также, слабо остаются выраженными углы, пока  $P$  не достигает значения приблизительно 56, когда угловые точки начинают «выдаваться» из последовательности. Наконец, при  $P = 61$  кривые начинают выпрямляться, и при добавлении еще одного коэффициента восстанавливается почти точная копия оригинала. Таким образом, нескольких низкочастотных коэффициентов достаточно для описания общей формы границы, однако для точного восстановления резких деталей, например, углов и прямолинейных участков, требуется значительно большее число высокочастотных коэффициентов.

В тех случаях, когда результат зависит от порядка обработки точек границы, ставится дополнительное требование, чтобы дескрипторы не зависели от выбора начальной точки. Фурье-дескрипторы сами по себе не инвариантны к указанным геометрическим изменениям, однако измененные дескрипторы могут быть получены несложными преобразованиями. Например, применительно к повороту, поворот точки комплексной плоскости на угол  $\theta$  относительно начала координат равносителен умножению соответствующего числа на  $e^{j\theta}$ . Выполнение этой операции для каждой точки  $s_k$  приводит к повороту всей последовательности на угол  $\theta$  относительно начала координат. Повернутая последовательность  $s_k e^{j\theta}$  характеризуется фурье-дескрипторами следующего вида:

$$a_u^r = \frac{1}{K} \sum_{k=0}^{K-1} s_k e^{j\theta} e^{-j2\pi uk/K} = a_u e^{j\theta},$$

для  $u = 0, \dots, K - 1$ .

Следовательно, поворот объекта приводит просто к умножению всех коэффициентов на одинаковую мультипликативную константу  $e^{j\theta}$ .

В таблице 6.4 [5] приведены выражения для Фурье-дескрипторов последовательности точек границы  $s_k$  после ее поворота, параллельного переноса, изменения масштаба и смены начальной точки. Символ  $\Delta_{xy}$  обозначает число  $\Delta_{xy} = \Delta_x + j\Delta_y$ , поэтому запись  $s_k^t = s_k + \Delta_{xy}$  соответствует последовательности, переопределенной путем параллельного переноса:

$$s_k^t = [x_k + \Delta_x] + j[y_k + \Delta_y].$$

Другими словами, параллельный перенос состоит в прибавлении постоянного смещения к координатам всех точек границы. Параллельный перенос не оказывает влияния на все дескрипторы, кроме первого ( $u = 0$ ), у которого значение дельта-функции  $\delta(u)$  будет ненулевым.

Таблица 6.4

#### Основные свойства дескрипторов Фурье

Преобразование	Граница	Фурье-дескрипторы
Тождественное	$s_k$	$a_u$
Поворот	$s_k^r = s_k e^{j\theta}$	$u_u^r = u_u e^{j\theta}$
Параллельный перенос	$s_k^t = s_k + \Delta_{xy}$	$a_u^t = a_u + \Delta_{xy} \delta_u$
Изменение масштаба	$s_k^s = \alpha s_k$	$a_u^s = \alpha a_u$
Смена начальной точки	$s_k^p = s_{k-k_0}$	$a_u^p = a_u e^{-j2\pi k_0 u / K}$

Последняя строка таблицы показывает, что изменение начальной точки влияет на все дескрипторы по-разному, хотя и известным способом, поскольку  $a_u$  умножается на член, зависящий от  $u$ .

## 6.6. Обработка изображений с помощью математической морфологии

### 6.6.1. Обработка бинарных изображений

Математическая морфология – относительно новый подход в обработке изображений, суть которого заключается в том, что исходное изображение рассматривается как множество, и к нему применяются теоретико-множественные операции [1,5]. Основные морфологические операции являются аналогами операций сложения и вычитания Минковского.

Пусть  $X \subseteq E$  есть множественное представление двоичного входного изображения, и пусть  $B \subseteq E$  есть компактное множество малого размера и простой формы (например,  $d$ -мерная сфера). Множество  $B$  называется структурирующим элементом. Пусть  $X \pm b = \{b : x \in X \text{ and } b \in B\}$  выражает векторный перенос  $X$  на  $\pm b \in E$ .

Фундаментальными морфологическими операторами для множеств являются наращение (dilation)  $\oplus$  и эрозия (erosion) –  $X$  с помощью  $b$ , которые определяются как [5]:

– расширение:  $X \oplus B = \bigcup_{b \in B} X + b$ ;

– эрозия:  $X - B = \bigcap_{b \in B} X - b$ .

Альтернативное определение операции расширения выглядит так:

$$G(j, k) = \bigcup_{m} \bigcup_{n} X(m, n) \cap b(j - m + 1, k - n + 1).$$

Операция эрозии также может быть определена иначе:

$$G(j, k) = \bigcap_{m} \bigcap_{n} X(m, n) \cup \neg b(j - m + 1, k - n + 1).$$

Операции расширения и эрозии обладают свойствами. Приведем некоторые из них:

- 1)  $X \oplus B = B \oplus X$ ;
- 2)  $X - B \neq B - X$ ;
- 3)  $\neg(X - B) \neq \neg X \oplus B$ ;
- 4) если  $X \subseteq B$ , то  $X \oplus C \subseteq B \oplus C$ ;
- 5) если  $X \subseteq B$ , то  $X - C \subseteq B - C$ .

Пример применения операций расширения и эрозии приведен на рисунке 6.48, для изображения, представленного на рис. 6.30, а.

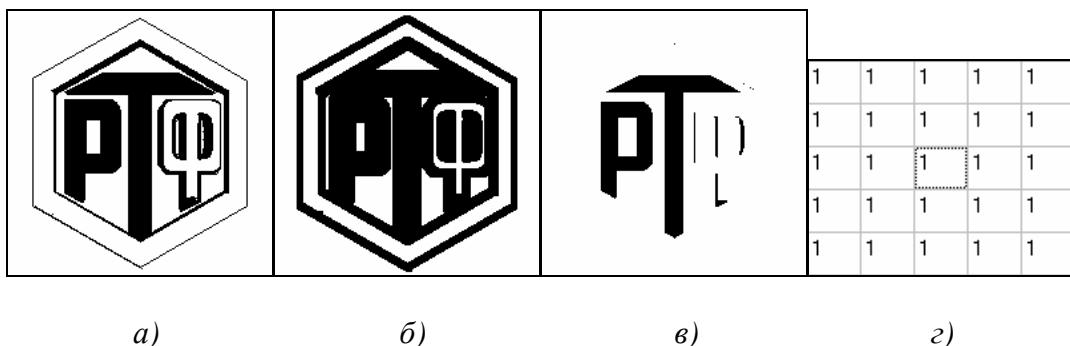


Рис. 6.48. Результаты применения операций математической морфологии:

- а) исходное бинарное изображение; б) результат применения операции расширения;
- в) результат применения операции эрозии; г) структурирующий элемент

Применение только операции расширения или эрозии обычно не дает эффекта в обработке изображений. Следует отметить, что в результате выполнения эрозии на изображениях исчезнут объекты, диаметр которых меньше диаметра структурирующего элемента. В результате же выполнения расширения могут закраситься впадины в объектах и пустоты в изображениях, чей диаметр меньше структурирующего элемента.

Кроме этого, следует отметить одно очень важное свойство морфологических операций. В результате их выполнения в изображении остаются лишь те симметрии, которые присутствуют в структурирующем элементе.

те. Поэтому, для того чтобы в результате обработки изображения не внести в него новых искажений, структурирующий элемент должен быть близок к кругу (симметричен по всем направлениям). В этом случае в изображении останутся все его симметрии.

Для того чтобы использовать аппарат морфологии в обработки изображений, применяют комбинации расширения и эрозии.

Отмыканием множества  $A$  множеством  $B$  ( $A^{\circ}B$ ) называется последовательное применение операций эрозии и расширений  $X \circ B = (X - B) \oplus B$ .

Замыканием множества  $A$  множеством  $B$  ( $A \bullet B$ ) называется последовательное применение операций расширения и эрозии  $X \bullet B = (X \oplus B) - B$ .

Результаты применения операций отмыкания и замыкания с использованием структурирующего элемента типа квадрат размером девять на девять элементов представлены на рис. 6.49.

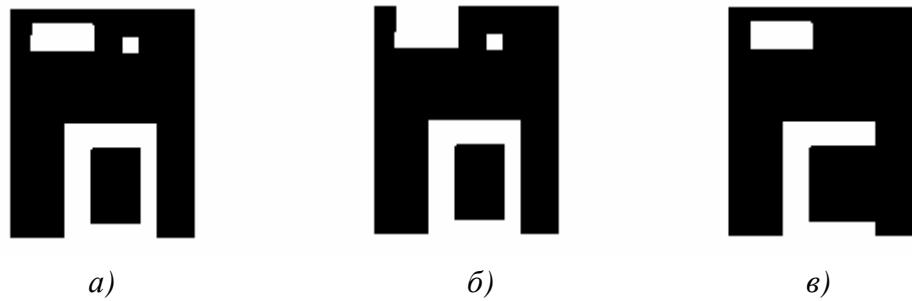


Рис. 6.49. Результат обработки: а) исходное изображение;  
б) результат применения операции отмыкания;  
в) результат применения операции замыкания.

На практике операция закрывания применяется для смыкания дыр в объектах, но только таких дыр, диаметры которых не превышают диаметра структурирующего элемента. Операция открывания может быть использована для разделения объектов на бинарном изображении. Здесь сложность состоит в правильном выборе размеров структурирующего элемента.

Указанные операции несут значительную нагрузку и являются основными морфологическими фильтрами, которые широко используются при обработке изображений.

С использованием операций математической морфологии на изображениях можно удалять изолированные пиксели и импульсный шум. Для проведения этой операции дважды используется эрозия. Значение каждого пикселя может быть вычислено так:  $A = A - Erosion(A, B_1, B_2)$ , где  $B_1$  и  $B_2$  – два структурирующих элемента:

$$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

С помощью базовой операции эрозии возможно нахождение восьмисвязного контура.

Для нахождения восьмисвязного контура можно воспользоваться формулой:

$$A = A - Erosion(A, N_8),$$

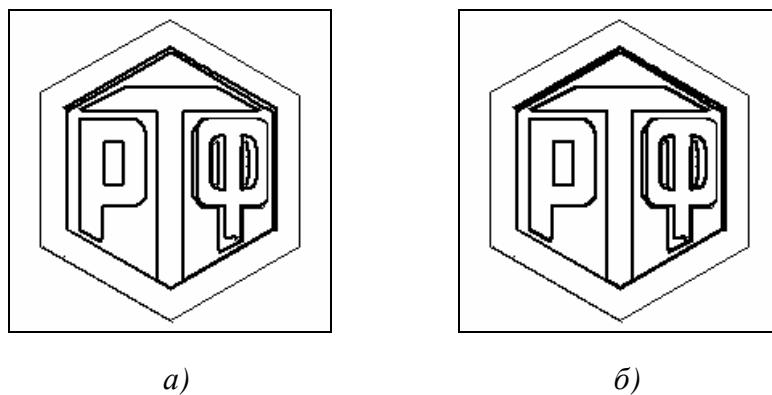
$$\text{где } N_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Для нахождения четырехсвязного контура можно воспользоваться формулой:

$$A = A - Erosion(A, N_4),$$

$$\text{где } N_4 = \begin{bmatrix} - & 1 & - \\ 1 & - & 1 \\ - & 1 & - \end{bmatrix}.$$

На рис. 6.50 показаны результаты контурного анализа для изображения, представленного на рис. 6.30, а.



а)

б)

Рис. 6.50. Результаты контурного анализа:

а) нахождение восьмисвязного контура; б) нахождение четырехсвязного контура

### 6.6.2. Обработка полутоновых изображений

Методы математической фильтрации могут быть расширены на полутоновые изображения. Как и для бинарных изображений, здесь применяются операции – расширение, эрозия, открытие и закрытие.

Для простоты считается, что коэффициенты в структурирующем элементе  $B$  равны нулю.

Тогда расширение для полутоновых изображений запишется в следующем виде:

$$D_G(A, B) = \max_B \{a[m - j, n - k]\} = \max_B(A).$$

Также определим операцию эрозия:

$$E_G(A, B) = \min_B \{a[m - j, n - k]\} = \min_B(A).$$

По аналогии с бинарными изображениями открывание есть последовательное применение операций эрозии и расширения, а закрывание – последовательное применение операций расширения и эрозии.

Открывание:

$$O_G(A, B) = \max_B (\min_B(A)).$$

Закрывание:

$$C_G(A, B) = \min_B (\max_B(A)).$$

Результаты применения рассмотренных операции представлены на рис. 6.51, структурирующий элемент размерности пять на пять элементов.

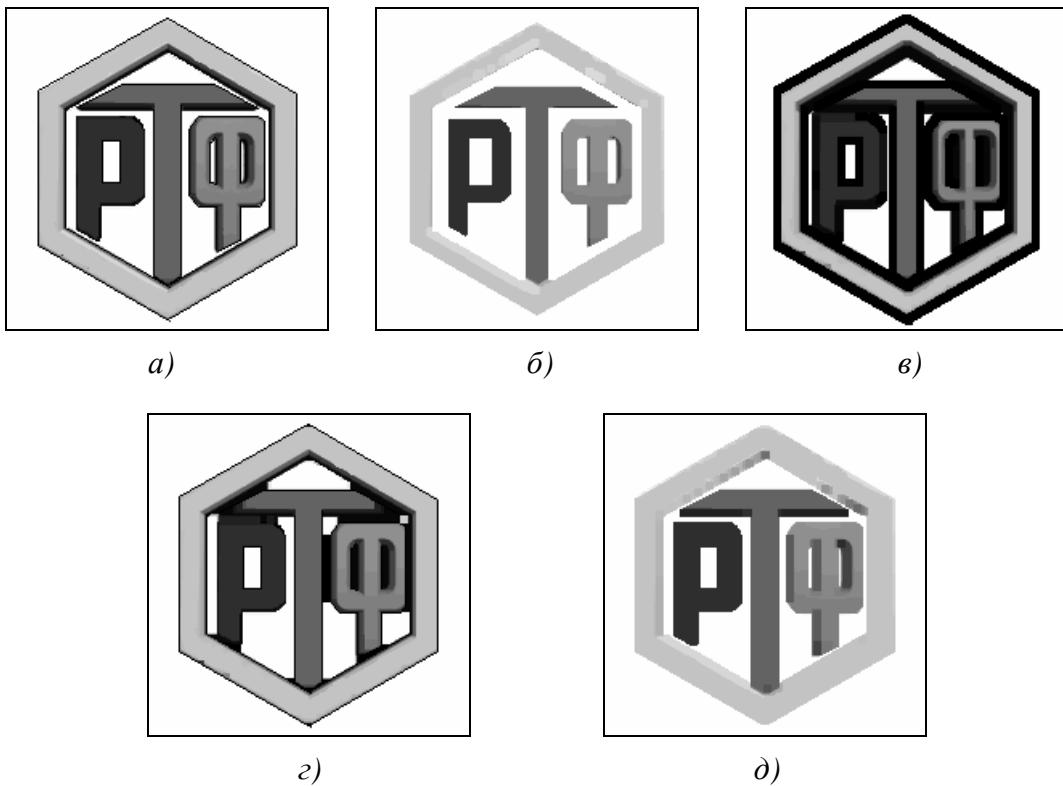


Рис. 6.51. Результаты применения операций математической морфологии:  
 а) исходное изображение; б) результат применения операции расширения;  
 в) результат применения операции эрозии; г) результат применения операции  
 открывания; д) результат применения операции закрывания

Следует отметить, что данные операции являются операциями фильтрации.

С помощью математической морфологии можно проводить сглаживание изображения. Определяя математически, эта операция есть последовательное применение открывания и закрывания к исходному изображению:

$$MorphSmooth = C_G(O_G(A, B), B) = \min(\max(\max(\min(A)))) .$$

Следующей очень важной операцией является нахождение морфологического градиента, который может быть использован для выделения контуров на изображении и определяется следующим образом:

$$Grad(A, B) = \frac{1}{2} * (\max(A) - \min(A)) .$$

Фильтр Лапласа через операции расширения и эрозии:

$$L(A, B) = \frac{1}{2} * (D_G(A, B) + E_G(A, B) - 2 * A) = \frac{1}{2} * (\max(A) + \min(A) - 2 * A) .$$

Изолирование (выделение) полутоновых объектов может быть выполнено при помощи преобразования, предложенного Мейером [5]:

$TopHat_1(A, B) = (A - \max(\min(A)))$  – для светлых объектов на темном фоне;

$TopHat_2(A, B) = (\min(\max(A)) - A)$  – для темных объектов на светлом фоне.

На практике размеры структурирующего элемента должны быть выбраны большими, чем размеры объектов на изображении.  $TopHat(A, B)$  есть неотрицательное число.

Для увеличения локального контраста можно воспользоваться формулой:

$$c[m, n] = 255 * \frac{A - \min(A)}{\max(A) - \min(A)} .$$

На рис. 6.52 показаны обработки изображения, представленного на рис.6.51, а.

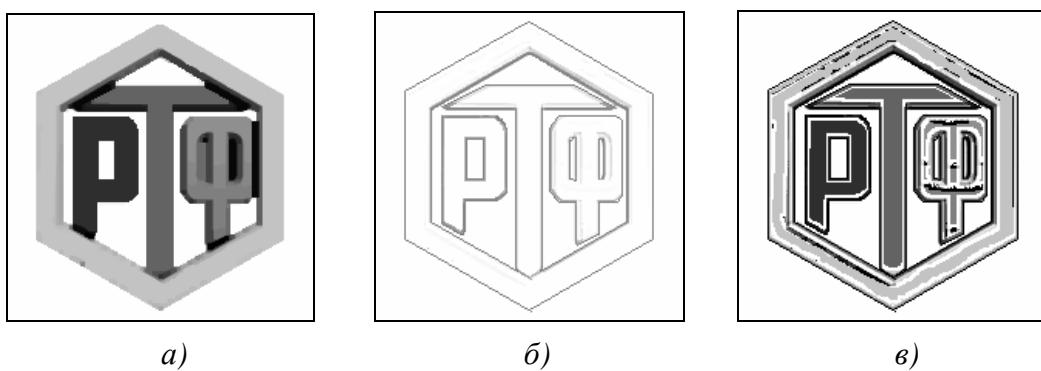


Рис. 6.52. Результаты применения операций математической морфологии:  
а) результат применения операции сглаживания; б) результат применения фильтра Лапласа; в) увеличение локального контраста

## **6.7. Вопросы и задания для самопроверки**

1. В чем сущность методов пространственной и методов частотной обработки изображений?
2. Что такое бинаризация изображений и в чем трудности данной процедуры?
3. Каково назначение градационного преобразования?
4. Представьте алгоритм преобразования цветного изображения в полутона (полутонового в бинарное).
5. В чем схожесть и отличие логарифмического и степенного преобразований?
6. Представьте алгоритм аддитивной пороговой бинаризации.
7. Приведите примеры практического использования градационных преобразований.
8. Приведите примеры кусочно-линейных функций, поясните принцип их работы.
9. Что такое соляризация изображения и где она может применяться?
10. В чем заключается сущность пространственной фильтрации изображений?
11. Какая возникает проблема при масочной фильтрации изображений и какими способами она может решаться?
12. Приведите примеры НЧ фильтров размером  $3 \times 3$  и  $5 \times 5$ ?
13. После применения какого фильтра на изображении будем наблюдать наибольшее размытие контуров и почему?

$$M_1 = \frac{1}{14} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad M_2 = \frac{1}{12} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad M_3 = \frac{1}{10} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 6 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

14. Приведите пример двумерного БИХ-фильтра.
15. В чем сущность медианной фильтрации?
16. Приведите примеры масок для ВЧ-фильтрации.
17. Сравните эффективность использования линейной и нелинейной фильтрации для различного рода шумов.
18. Какие методы используются для улучшения контраста изображения и как они описываются математически?

19. Определите коэффициенты  $a$  и  $b$  при линейной растяжке гистограммы, если  $f_{\min} = 56$  и  $f_{\max} = 158$ .
20. Какой будет результат при повторном применении процедуры эквализации гистограммы распределения яркостей к цифровому изображению?
21. Что такое сегментация изображений? Для решения каких практических задач она применяется?
22. На основе каких свойств сигнала яркости базируются алгоритмы сегментации?
23. Сформулируйте обобщенный алгоритм выделения контуров на изображении?
24. Какие проблемы возникают при выделении линий и контуров на изображении и как можно их минимизировать?
25. Представьте маску для выделения горизонтальных (вертикальных) линий, отличную от приведенных в пособии.
26. По какому признаку можно определить направления масок для подчеркивания линий заданного направления? Поясните свой ответ на примере.
27. Каков будет результат работы курсовой градиентной маски «Север», если значение центрального коэффициента изменить на « $-1$ » (« $0$ »)?
28. Приведите достоинства и недостатки операторов Лапласа.
29. Можно ли применить представленные маски для подчеркивания контуров на изображении? Поясните свой ответ.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

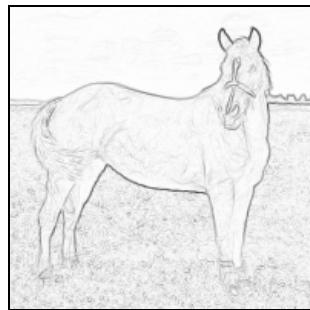
30. На каком рисунке представлен результат применения алгоритма Робертса к исходному изображению?



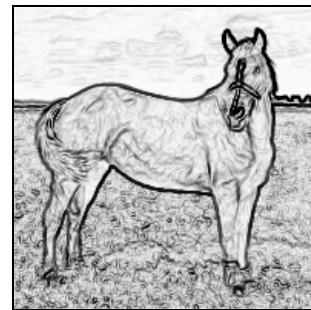
Исходное изображение



*a)*



б)



в)

31. Приведите сравнение качественных характеристик методов подчеркивания контуров.
32. В чем сущность метода связывания контуров на основе локальной обработки?
33. Для каких целей может использоваться преобразование Хафа?
34. Представьте алгоритм связывания контуров на основе преобразования Хафа.
35. Приведите пример описание изображения с помощью цепного кода.
36. Что такое сигнатура и для чего она применяется?
37. Что понимают под Фурье-дискрипторами? Представьте их основные свойства.
38. Приведите базовые операции математической морфологии.
39. Представьте примеры процедур обработки бинарных изображений на основе математической морфологии.
40. Представьте примеры процедур обработки полутоновых изображений на основе математической морфологии.
41. На каком рисунке представлен результат применения операции эрозии к исходному изображению? Поясните ответ.



Исходное изображение



а)



б)



в)

## **МОДУЛЬ 7. ОБРАБОТКА И СЖАТИЕ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ДИСКРЕТНЫХ ПРЕОБРАЗОВАНИЙ**

Цель модуля – изучение студентами алгоритмов обработки полуточновых изображений в частотной области, включая фильтрацию и сжатие.

В результате изучения модуля студенты *должны знать*:

- алгоритм и особенности дискретного двумерного преобразования Фурье;
- основные шаги и особенности алгоритмов сжатия JPEG и JPEG 2000.

Студенты *должны уметь*:

- применять дискретные преобразования для обработки изображений в частотной области;
- а также иметь представление о критериях, предъявляемых к алгоритмам сжатия.

Существуют три основные области применения двумерных преобразований для обработки изображений [5]. Во-первых, преобразования используются для выделения характерных признаков изображения. Так, например, постоянная составляющая спектра Фурье пропорциональна средней яркости изображения, а высокочастотные составляющие характеризуют величину и ориентацию его контуров. Другой областью применения преобразований является кодирование изображений, когда ширина спектра уменьшается за счет отбрасывания или грубого квантования малых по величине коэффициентов преобразования. Третья область приложений это сокращение размерности при выполнении вычислений. Иначе говоря, в процессе обработки (например, фильтрации) малые коэффициенты преобразования можно отбросить без заметного ухудшения качества обработки.

### **7.1. Двумерное преобразование Фурье. Фильтрация изображений в частотной области**

#### **7.1.1. Дискретное двумерное преобразование Фурье**

Дискретное двумерное преобразование Фурье матрицы отсчетов изображения  $S$  размером  $N \times N$  с элементами  $s_{ik}$  определяется как [5]:

$$f_{uv} = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} s_{ik} \exp\left\{-\frac{2\pi j}{N}(ui + vk)\right\},$$

где  $j = \sqrt{-1}$  ;

$f_{uv}$  – спектральные коэффициенты преобразования;

$u, v = 0, 1, \dots, N - 1$ .

Двумерное дискретное обратное преобразование Фурье имеет вид:

$$s_{ik} = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f_{uv} \exp\left\{-\frac{2\pi j}{N}(ui + vk)\right\}.$$

Фурье-спектр, фаза и энергетический спектр определяются так же, как и в одномерном случае:

$$|f_{uv}| = \left[ R_{uv}^2 + I_{uv}^2 \right]^{1/2}, \quad \Phi_{uv} = \arctg \left[ \frac{I_{uv}}{R_{uv}} \right], \quad P = p_{uv} = |f_{uv}|^2 = R_{uv}^2 + I_{uv}^2,$$

где  $R(u, v)$  и  $I(u, v)$  – величины, которые обозначают действительную и мнимую части величины  $f_{uv}$  соответственно.

Двумерное преобразование можно выполнить в виде последовательных одномерных преобразований по строкам и столбцам матрицы изображения. Базисными функциями преобразования, как известно, являются экспоненты с комплексными показателями, которые, можно разложить на синусную и косинусную составляющие на основе формулы Эйлера:

$$\begin{aligned} \exp\left\{-\frac{2\pi j}{N}(ui + vk)\right\} &= \cos\left\{\frac{2\pi}{N}(ui + vk)\right\} - j \sin\left\{\frac{2\pi}{N}(ui + vk)\right\}; \\ \exp\left\{\frac{2\pi j}{N}(ui + vk)\right\} &= \cos\left\{\frac{2\pi}{N}(ui + vk)\right\} + j \sin\left\{\frac{2\pi}{N}(ui + vk)\right\}. \end{aligned}$$

Спектр изображения имеет ряд структурных особенностей. Например, спектральная составляющая в начале координат частотной плоскости (постоянная составляющая) равна увеличенному в  $N$  раз среднему (по исходной плоскости) значению яркости изображения:

$$f_{00} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} s_{ik}.$$

Подставив в равенство прямого преобразования Фурье  $u = u + mN$  и  $v = v + nN$ , где  $m$  и  $n$  – постоянные, получим:

$$f_{u+mN, v+nN} = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} s_{ik} \exp\left\{-\frac{2\pi j}{N}(ui + vk)\right\} \exp\left\{-2\pi j(mi + nk)\right\}.$$

При любых целочисленных значениях второй экспоненциальный множитель равенства превращается в единицу. Таким образом, при  $m, n = 0, \pm 1, \pm 2, \dots$ :

$$f_{u+mN, v+nN} = f_{u, v},$$

что свидетельствует о периодичности частотной плоскости.

Рис. 7.1, *a* [5] иллюстрирует этот результат.

Двумерный спектр Фурье изображения является по существу представлением двумерного поля в виде ряда Фурье. Для того чтобы такое представление было справедливым, исходное изображение также должно обладать периодической структурой [5], т. е. (как показано на рис. 7.1, *b*) иметь рисунок, повторяющийся по вертикали и горизонтали. Таким образом, правый край изображения примыкает к левому, а верхний край – к нижнему.

Из-за разрывов значений яркости в этих местах в спектре изображения возникают дополнительные составляющие, лежащие на координатных осях частотной плоскости. Эти составляющие не связаны со значениями яркости внутренних точек изображения, но они необходимы для воспроизведения его резких границ.

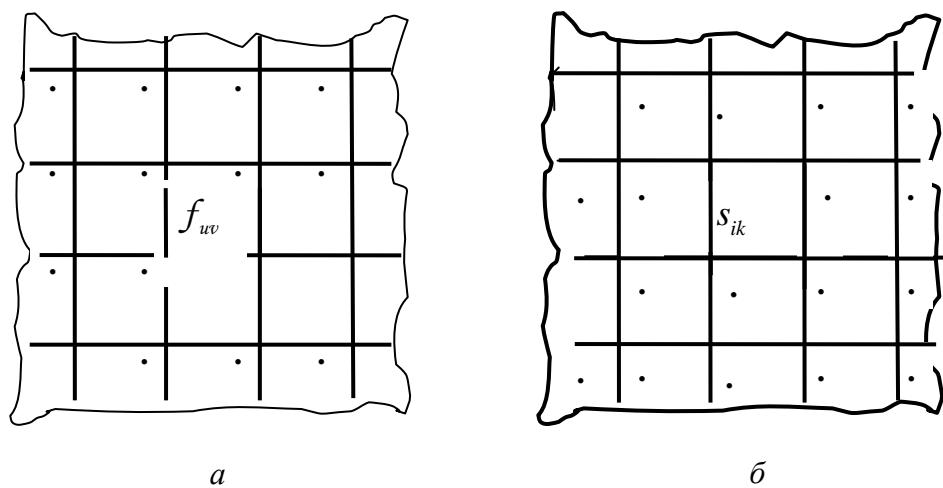


Рис. 7.1. Периодическое продолжение изображения и спектра Фурье:  
*а* – спектр; *б* – исходное изображение

Если массив отсчетов изображения описывает поле яркости, то числа будут действительными и положительными. Однако спектр Фурье этого изображения в общем случае имеет комплексные, значения. Поскольку спектр содержит  $2N^2$  компонент, представляющих действительную и минимую части или фазу и модуль спектральных составляющих для каждой частоты, может показаться, что преобразование Фурье увеличивает размерность изображения. Это, однако, не так, поскольку *спектр* обладает симметрией относительно комплексного сопряжения (аналогично спектру одномерных сигналов). Из-за наличия комплексно-сопряженной симметрии почти половина спектральных составляющих оказывается избыточной, т. е. их можно сформировать из остальных составляющих.

Поскольку динамический диапазон составляющих спектра достаточно широк, то спектры необходимо подвергнуть сжатию. Сжатие динамического диапазона можно осуществить путем ограничения больших спектральных составляющих или логарифмическим преобразованием всех составляющих спектра согласно соотношению:

$$d_{uv} = \log[a + b|f_{uv}|],$$

где  $a$  и  $b$  – масштабные постоянные.

При математическом анализе непрерывных сигналов начало координат частотной плоскости обычно помещают в ее геометрическом центре. Двумерный дискретный Фурье-спектр, найденный с помощью компьютера, путем простой перестановки коэффициентов можно изменить так, чтобы начало координат также оказалось в центре массива. Тот же результат можно получить по-другому, если отсчеты изображения предварительно умножить на коэффициенты вида  $(-1)^{i+k}$ . Тогда квадранты спектра, рассчитанного согласно формуле прямого преобразования Фурье, в процессе вычисления автоматически меняются местами.

Хотя преобразование Фурье имеет много полезных для анализа свойств изображений, как известно, у него есть и два существенных недостатка [5]: во-первых, все вычисления приходится производить не с действительными, а с комплексными числами и, во-вторых, ряды сходятся медленно. Последнее замечание, весьма существенное для задач кодирования изображений, можно пояснить, записав:

$$s_{ik} = \lim_{\substack{U_T \rightarrow (N+1)/2 \\ V_T \rightarrow (N+1)/2}} \left[ \frac{1}{N} \underbrace{\sum_u \sum_v f_{uv}}_{u, v \in W(U_T, V_T)} \exp \left\{ \frac{2\pi j}{N} (ui + vk) \right\} \right],$$

где  $W(U_T, V_T)$  – низкочастотная область частотной плоскости, которая на рис. 7.2 заштрихована.

Если верхние границы  $U_T$  и  $V_T$  зафиксированы, а размер изображения  $N$  сравнительно невелик, то заключенная в квадратные скобки величина может сильно отличаться от  $s_{ik}$ , пока  $U_T$  и  $V_T$  не станут достаточно большими. Качественно можно указать, что плохая сходимость преобразования Фурье обусловлена скачками изображения, возникающими на линиях перехода от левого края изображения к правому и от верхнего к нижнему. Эти разрывы приводят к появлению в спектре больших составляющих с высокими пространственными частотами.

Каждый элемент Фурье-образа  $f_{u,v}$  содержит все отсчеты функции  $s_{ik}$ , умноженные на значения экспоненциальных членов. Поэтому обычно, за исключением тривиальных случаев, невозможно установить прямое соответствие между характерными деталями изображения и его образа, однако некоторую взаимосвязь частотных составляющих Фурье-образа и пространственных характеристик изображения можно установить [5]. Например, поскольку частота прямо связана со скоростью изменения сигнала, то интуитивно ясно, что частоты в Фурье-преобразовании связаны с вариацией яркости на изображении. Низкие частоты, отвечающие точкам вблизи начала координат Фурье-преобразования, соответствуют медленно меняющимся компонентам изображения. На изображении комнаты, например, они могут соответствовать плавным изменениям яркости стен и пола. По мере удаления от начала координат, более высокие частоты начинают соответствовать все более и более быстрым изменениям яркости, которые суть границы объектов и другие детали изображения, характеризуемые резкими изменениями яркости, такие как шум. На рис. 7.3 представлен пример преобразования Фурье изображения.

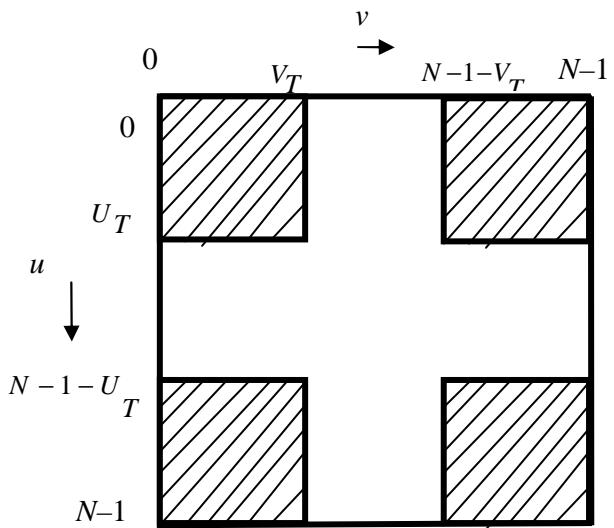
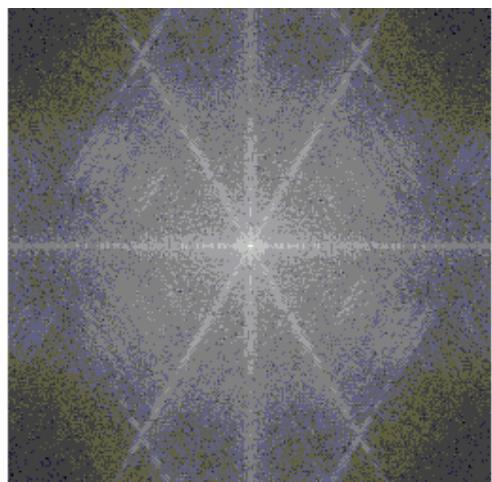


Рис. 7.2. Низкочастотная область частотной плоскости



a)

б)

Рис. 7.3. Пример преобразования Фурье: а) изображение; б) спектр

На рис. 7.4 представлено увеличенное примерно в 2500 раз изображение интегральной схемы [5]. Помимо конструкции, видны две характерные детали – контуры, проходящие под углом примерно  $\pm 45^\circ$ , и пара белых пятен – дефектов. В Фурье-спектре на 7.4, б хорошо видны диагональные составляющие, которые отвечают упомянутым контурам. При внимательном рассмотрении области, расположенной вдоль вертикальной оси, можно заметить частотную составляющую, слегка повернутую против часовой стрелки. Наличие этой составляющей обусловлено контурами оксидных пятен, причем угол поворота этой частотной составляющей, связан с отклонением длинного белого пятна от горизонтали, а также положение нулей этой частотной составляющей, связанное с малым поперечным размером оксидных пятен.



Рис. 7.4. Пример применения Фурье-преобразования к анализу изображений:  
а) изображение поврежденной интегральной схемы;  
б) Фурье-спектр изображения, представленного на рис. 7.4, а

### 7.1.2. Основы фильтрации в частотной области

Процедура фильтрации в частотной области проста и состоит из следующих шагов [5,6]:

- 1) исходное изображение умножается на  $(-1)^{i+k}$ , чтобы его Фурье-преобразование оказалось центрированным;
- 2) вычисляется прямое ДПФ  $f_{u,v}$  изображения, полученного после первого шага;
- 3) функция  $f_{u,v}$  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  $(-1)^{i+k}$ .

Причина, по которой множитель  $h_{u,v}$  называется *фильтром*, состоит в том, что он подавляет некоторые частоты преобразования, оставляя при этом другие без изменения.

Пусть  $s_{i,k}$  обозначает входное изображение после шага 1, и пусть  $f_{u,v}$  есть его Фурье-образ. Тогда Фурье-образ выходного изображения определяется выражением [5]:

$$g_{u,v} = h_{u,v} f_{u,v}.$$

Умножение функций двух переменных  $h$  и  $f$  осуществляется *поэлементно*. Это означает, что первый элемент функции  $h$  умножается на первый элемент функции  $f$ , второй элемент функции  $h$  – на второй элемент функции  $f$ , и т. д. В общем случае, компоненты фильтра  $h$  являются комплексными величинами. В случае если компоненты фильтра содержат только действительные величины, то действительная и мнимая части функции  $f$  умножаются на одну и ту же действительную функцию фильтра  $h$ . Такие фильтры называются фильтрами *нулевого фазового сдвига*. Как и следует из названия, эти фильтры не меняют фазу Фурье-преобразования.

Фильтрованное изображение получается вычислением обратного преобразования Фурье от Фурье-образа  $g_{u,v}$ .

Искомое изображение получается выделением действительной части из последнего результата и умножения на  $(-1)^{i+k}$ , чтобы скомпенсировать эффект от умножения входного изображения на ту же величину. Обратное Фурье-преобразование в общем случае является комплексным. Однако в случае вещественного входного изображения и вещественной передаточной функции фильтра мнимые части всех значений обратного Фурье-преобразования должны равняться нулю. Однако на практике значения обратного Фурье-преобразования, как правило, содержат паразитную мнимую составляющую, что связано с ошибками округления при вычислениях. Этой составляющей необходимо пренебречь.

Описанная процедура фильтрации схематически отображена на рис. 7.5 в несколько более общем виде, включающем стадии предварительной и заключительной обработки. Помимо умножения на  $(-1)^{i+k}$ , такая обработка может включать обрезание входного изображения так, чтобы его размеры приняли ближайшие четные значения по отношению к исходным (это необходимо для правильного центрирования Фурье-преобразования), яркостное масштабирование, преобразование формата входных данных в формат с плавающей точкой и преобразование формата выходных данных в восьмибитовые целые. Возможны многоступенчатые процедуры фильтрации, а также разнообразные операции предварительной и заключительной обработки.

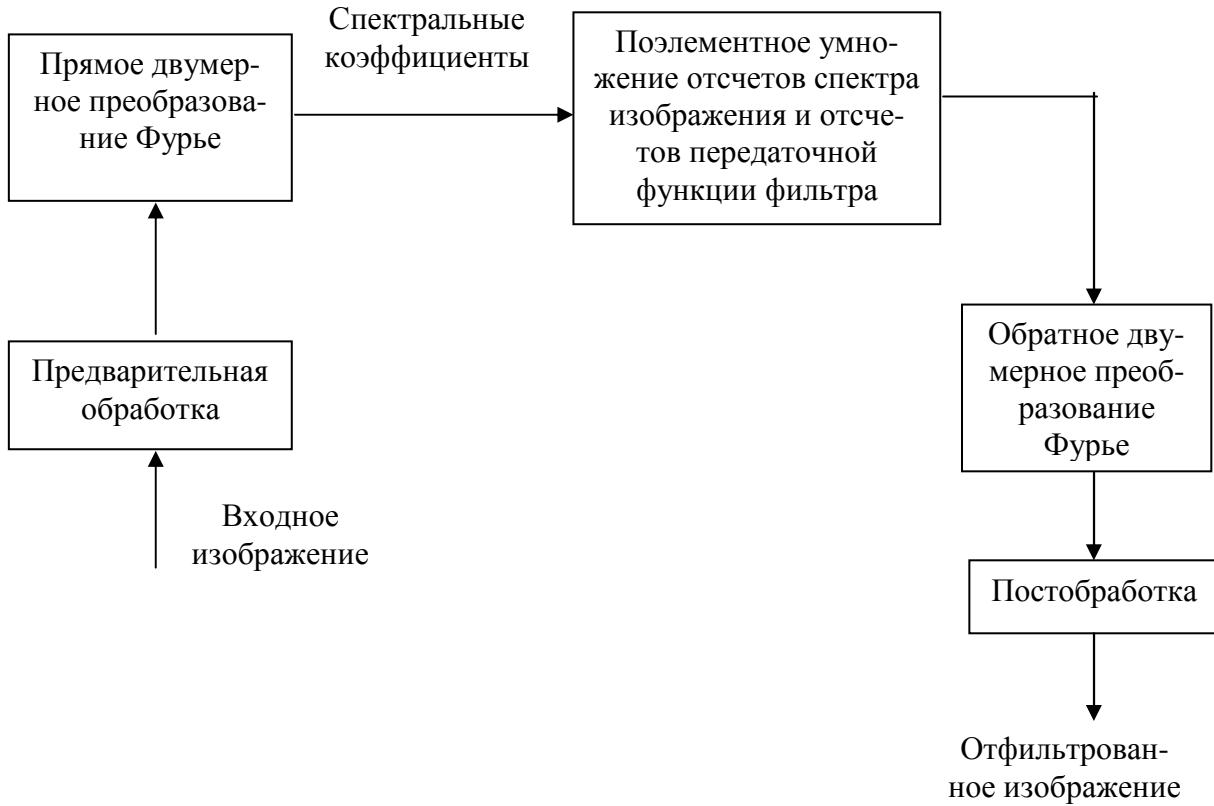


Рис. 7.5. Основные этапы фильтрации изображений в частотной области

Простейшим примером фильтрации изображений является обращение в нуль среднего значения на изображении. Известно, что среднее значение задается величиной  $f_{00}$ . Если обнулить этот член в частотной области и выполнить обратное преобразование, то среднее значение полученного изображения будет равно нулю. В предположении, что изображение было предварительно отцентрировано так, можно осуществить такую операцию умножением всех значений  $f_{u,v}$  на следующую функцию фильтра [5]:

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

Все действие такого фильтра сводится к тому, что значение  $f_{00}$  будет обращено в нуль, а остальные частотные компоненты Фурье-преобразования останутся незатронутыми, что и требуется. Обработанное изображение (с нулевым средним) затем может быть получено применением обратного Фурье-преобразования.

Такой фильтр называется *фильтр-пробка* [5], поскольку он представляет собой постоянную функцию с вырезом (дырой) в начале координат. Результат обработки изображения, представленного на рис. 7.3, дан-

ным фильтром показан на рис. 7.6. Анализ рис. 7.6 показывает падение общей яркости в результате принудительного обнуления среднего значения; а также побочный результат, состоящий в выделении контуров. В действительности среднее значение яркости *выводимого на экран* изображения не может равняться нулю, поскольку для этого некоторые элементы изображения должны быть отрицательными, а монитор не может оперировать с отрицательными величинами. Рисунок 7.6 представлен стандартным образом, при котором наименьшее отрицательное значение соответствует нулю, или черному цвету, а остальные значения увеличены с учетом этого. Узкополосные режекторные фильтры полезны, когда удается идентифицировать пространственные эффекты на изображении, вызываемые локализованными компонентами частотного представления.

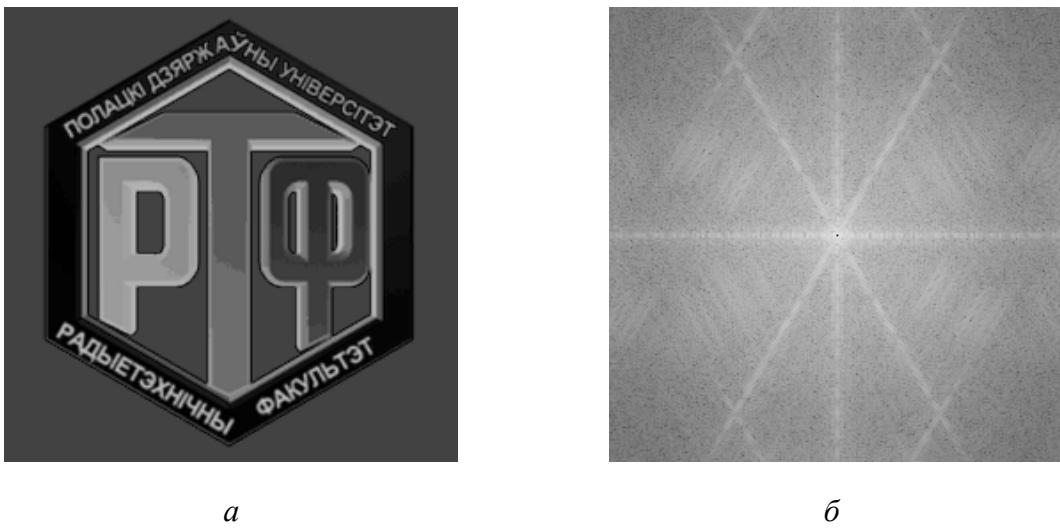


Рис. 7.6. Результат фильтрации изображения, представленного на рис. 7.3 с помощью фильтра-пробки (а), и спектр обработанного изображения (б)

Как известно, низкочастотные составляющие Фурье-преобразования отвечают за формирование крупных деталей изображения, а высокие частоты ответственны за мелкие детали – контуры и шум.

Таким образом, после применения низкочастотной фильтрации, изображение, по сравнению с исходным, представляется менее резким, т. к. содержит меньше мелких деталей, поскольку высокие частоты подавлены. Аналогично после применения высокочастотной фильтрации, на изображении уменьшаются изменения яркости в пределах больших гладких областей и выделяются переходные зоны быстрого изменения яркости (т. е. контуры). Такое изображение выглядит более резким.

## **7.2. Сжатие изображений. Спектральные методы сжатия**

### **7.2.1. Особенности изображений, учитываемые при сжатии**

Очевидно, что изображения требуют для хранения гораздо большего объема памяти, чем текст. Например, не очень качественная иллюстрация на обложке книги размером  $500 \times 800$  точек занимает 1,2 Мб – столько же, сколько художественная книга из 400 страниц (60 знаков в строке, 42 строки на странице) [3]. Причем, чем более качественным является изображение, тем больше объема памяти требуется для его хранения. Эта особенность изображений определяет актуальность алгоритмов архивации графики.

Кроме этого следует учитывать, что человеческое зрение при анализе изображения оперирует контурами, общим переходом цветов и сравнительно нечувствительно к малым изменениям в изображении. Таким образом, можно создать эффективные алгоритмы архивации изображений, в которых декомпрессированное изображение не будет совпадать с оригиналом, однако человек этого не заметит. Данная особенность человеческого зрения позволила создать специальные алгоритмы сжатия, ориентированные только на изображения. Эти алгоритмы позволяют сжимать изображения с высокой степенью сжатия и незначительными с точки зрения человека потерями.

Можно заметить, что изображение, в отличие, например, от текста, обладает избыточностью в 2-х измерениях. Т. е. как правило, соседние точки, как по горизонтали, так и по вертикали, в изображении близки по цвету. Кроме этого, можно использовать подобие между цветовыми плоскостями  $R$ ,  $G$  и  $B$  в алгоритмах, что дает возможность создать еще более эффективные алгоритмы. Таким образом, при создании алгоритма компрессии графики используются особенности структуры изображения.

### **7.2.2. Требования к алгоритмам компрессии**

Следует сразу отметить, что приложение определяет характер использования изображений (либо большое количество изображений хранится и используется, либо изображения скачиваются по сети, либо изображения велики по размерам, и необходима возможность получения лишь части...). Характер использования изображений задает степень важности следующих ниже противоречивых требований к алгоритму сжатия [3].

1. Высокая степень компрессии. Однако далеко не для всех приложений актуальна высокая степень компрессии. Кроме этого, некоторые алгоритмы дают лучшее соотношение качества к размеру файла при высоких степенях компрессии, однако проигрывают другим алгоритмам при низких степенях.

2. Высокое качество изображений. Выполнение этого требования напрямую противоречит выполнению предыдущего.

3. Высокая скорость компрессии. Это требование для некоторых алгоритмов с потерей информации является взаимоисключающим с первыми двумя. Интуитивно понятно, что чем больше времени будет анализироваться изображение для получения наивысшей степени компрессии, тем лучше будет результат. И, соответственно, чем меньше времени потратиться на компрессию (анализ), тем ниже будет качество изображения и больше его размер.

4. Высокая скорость декомпрессии. Достаточно универсальное требование, актуальное для многих приложений.

5. Масштабирование изображений. Данное требование подразумевает легкость изменения размеров изображения до размеров окна активного приложения. Дело в том, что одни алгоритмы позволяют легко масштабировать изображение прямо во время декомпрессии, в то время как другие не только не позволяют легко масштабировать, но и увеличивают вероятность появления неприятных артефактов после применения стандартных алгоритмов масштабирования к декомпрессированному изображению. Например, можно привести пример «плохого» изображения для алгоритма JPEG – это изображение с достаточно мелким регулярным рисунком (пиджак в мелкую клетку). Характер вносимых алгоритмом JPEG искажений таков, что уменьшение или увеличение изображения может дать неприятные эффекты.

6. Возможность показать огрубленное изображение (низкого разрешения), используя только начало файла. Данная возможность актуальна для различного рода сетевых приложений, где перекачивание изображений может занять достаточно большое время, и желательно, получив начало файла, корректно показать preview. Примитивная реализация указанного требования путем записывания в начало изображения его уменьшенной копии заметно ухудшит степень компрессии.

7. Устойчивость к ошибкам. Данное требование означает локальность нарушений в изображении при порче или потере фрагмента передаваемого файла. Данная возможность используется при широковещании (broadcasting – передача по многим адресам) изображений по сети, то есть в тех случаях, когда невозможно использовать протокол передачи, повторно запрашивающий данные у сервера при ошибках. Например, если передается видеоряд, то было бы неправильно использовать алгоритм, у которого сбой приводил бы к прекращению правильного показа всех последующих кадров. Данное требование противоречит высокой степени архивации, поскольку интуитивно понятно, что мы должны вводить в поток избыточную информацию. Однако для разных алгоритмов объем этой избыточной информации может существенно отличаться.

8. Учет специфики изображения. Более высокая степень архивации для класса изображений, которые статистически чаще будут применяться в приложении.

9. Редактируемость. Под редактируемостью понимается минимальная степень ухудшения качества изображения при его повторном сохранении после редактирования. Многие алгоритмы с потерей информации могут существенно испортить изображение за несколько итераций редактирования.

10. Небольшая стоимость аппаратной реализации. Эффективность программной реализации. Данные требования к алгоритму реально предъявляют разработчики большинства информационных систем. Таким образом, на практике для каждой задачи можно сформулировать набор приоритетов из требований, изложенных выше, который и определит наиболее подходящий для конкретной задачи алгоритм (либо набор алгоритмов) для ее решения.

### **7.2.3. Критерии сравнения алгоритмов**

Характеристики алгоритма относительно некоторых требований приложений, сформулированные выше, зависят от конкретных условий, в которые будет поставлен алгоритм. Так, степень компрессии зависит от того, на каком классе изображений алгоритм тестируется. Аналогично, скорость компрессии нередко зависит от того, на какой платформе реализован алгоритм. Преимущество одному алгоритму перед другим может дать, например, возможность использования в вычислениях алгоритма технологий нижнего уровня, а это возможно далеко не для всех алгоритмов. Так, JPEG существенно выигрывает от применения таких технологий, а LZW нет. Кроме того, некоторые алгоритмы распараллеливаются легко, а некоторые нет.

Таким образом, невозможно составить универсальное сравнительное описание известных алгоритмов [3]. Это можно сделать только для типовых классов приложений при условии использования типовых алгоритмов на типовых платформах. Однако такие данные необычайно быстро устаревают.

Так, например, еще в 1994, интерес к показу огрубленного изображения, используя только начало файла (требование 6), был чисто абстрактным. Реально эта возможность практически нигде не требовалась и класс приложений, использующих данную технологию, был крайне невелик. С распространением Internet, который характеризуется передачей изображений по каналам связи, использование Interlaced GIF (алгоритм LZW) и Progressive JPEG (вариант алгоритма JPEG), реализующих эту возможность, резко возросло. То, что новый алгоритм (например, wavelet) поддерживает такую возможность, существеннейший плюс для него сегодня.

Основные критерии следующие [3]:

1. Худшая, средняя и лучшая степень сжатия. То есть доля, на которую возрастет изображение, если исходные данные будут наихудшими; некая среднестатистическая степень для того класса изображений, на который ориентирован алгоритм; и, наконец, лучшая степень. Последняя необходима лишь теоретически, поскольку показывает степень сжатия наилучшего (как правило, абсолютно черного) изображения, иногда фиксированного размера.

2. Класс изображений, на который ориентирован алгоритм. Иногда указано также, почему на других классах изображений получаются худшие результаты.

3. Симметричность. Отношение характеристики алгоритма кодирования к аналогичной характеристике при декодировании. Характеризует ресурсоемкость процессов кодирования и декодирования. Наиболее важной является симметричность по времени: отношение времени кодирования ко времени декодирования. Иногда требуется симметричность по памяти.

4. Есть ли потери качества? И если есть, то за счет чего изменяется степень сжатия? У большинства алгоритмов сжатия с потерей информации существует возможность изменения степени сжатия.

5. Характерные особенности алгоритма и изображений, к которым его применяют. Здесь могут указываться наиболее важные для алгоритма свойства, которые могут стать определяющими при его выборе.

Следует отметить, что один и тот же алгоритм часто можно реализовать разными способами. Многие известные алгоритмы, такие как RLE, LZW или JPEG, имеют десятки различающихся реализаций. Кроме того, у алгоритмов бывает несколько явных параметров, варьируя которые, можно изменять характеристики процессов архивации и разархивации. При конкретной реализации эти параметры фиксируются, исходя из наиболее вероятных характеристик входных изображений, требований на экономию памяти, требований на время архивации и т. д. Поэтому у алгоритмов одного семейства лучшая и худшая степени сжатия могут отличаться, но качественно картина не изменится.

Первыми для архивации изображений стали применяться традиционные алгоритмы сжатия, т. е. те, что использовались и используются в системах резервного копирования, при создании дистрибутивов и т. п. Эти алгоритмы архивировали информацию без изменений. Однако основной тенденцией в последнее время стало использование новых классов изображений. Старые алгоритмы перестали удовлетворять требованиям, предъявляемым к архивации. Многие изображения практически не

сжимались, хотя «на взгляд» обладали явной избыточностью. Это привело к созданию нового типа алгоритмов – сжимающих с потерей информации. Как правило, степень сжатия и, следовательно, степень потери качества в них можно задавать. При этом достигается компромисс между размером и качеством изображений.

Одна из серьезных проблем машинной графики заключается в том, что до сих пор *не найден* адекватный критерий оценки потерь качества изображения. А теряется оно постоянно – при оцифровке, при переводе в ограниченную палитру цветов, при переводе в другую систему цветопредставления для печати, и, что для нас особенно важно, при архивации с потерями. Можно привести примеры критериев: среднеквадратичное отклонение значений пикселов, максимальное отклонение, мера отношения сигнала к шуму и т. д. (меры сравнения изображений будут рассматриваться далее).

Лучше всего потери качества изображений оценивают глаза человека. Отличной считается архивация, при которой невозможно на глаз различить первоначальное и разархивированное изображения. Хорошей – когда сказать, какое из изображений подвергалось архивации, можно только сравнивая две находящихся рядом картинки. При дальнейшем увеличении степени сжатия, как правило, становятся заметны побочные эффекты, характерные для данного алгоритма. На практике, даже при отличном сохранении качества, в изображение могут быть внесены регулярные специфические изменения.

В существующих алгоритмах сжатия изображений широко используются спектральные методы, основанные на применении известных дискретных двукратных преобразований.

#### **7.2.4. Основы спектральных методов сжатия**

Спектральные методы сжатия (эффективного кодирования) изображений основаны на том, что цифровой эквивалент аналогового сигнала путем соответствующего линейного ортогонального преобразования может быть приведен к виду, который наиболее удобен с точки зрения сокращения избыточности информации. В этом случае повышение эффективности кодирования связано с тремя факторами:

- в процессе преобразования ряд спектральных коэффициентов (трансформант) становится настолько малым по величине, что их можно отбросить без заметного ухудшения качества восстанавливаемых изображений;

- в процессе преобразования осуществляется декорреляция данных, обеспечивающая повышение эффективности статистического кодирования;

– различное нелинейное квантование коэффициентов преобразования позволяет существенно сократить объем передаваемой информации без заметного ухудшения качества передаваемого изображения при его восстановлении.

Сущность спектральных методов эффективного кодирования изображений заключается в том, что кодируется и передается не само изображение, а значения трансформант, получающихся при ортогональном преобразовании этого изображения. В процессе преобразований изображения, имеющего сильные корреляционные связи между смежными элементами, происходит процесс декорреляции, так что значения коэффициентов преобразования оказываются практически некоррелированными. В отличие от исходного изображения, для которого характерно в среднем равномерное распределение энергии между элементами, распределение энергии между спектральными коэффициентами резко неравномерно. Основная доля энергии приходится на коэффициенты с малыми номерами (низкие пространственные частоты), и лишь небольшая ее часть – на прочие. В дальнейшем коэффициенты, имеющие малую амплитуду, опускаются либо квантуются на малое число уровней, что позволяет использовать меньшее число разрядов кода для передачи. При приеме выполняются обратные преобразования. Предварительно по принятому коду восстанавливается матрица трансформант, а затем путем обратного ортогонального преобразования изображение восстанавливается.

В общем случае эффективное кодирование посредством преобразований реализуется путем двух последовательных операций. Первая операция состоит в линейном преобразовании статистически зависимых элементов изображения. С помощью второй операции осуществляется кодирование определенным способом выбранных коэффициентов преобразования.

Для кодирования и сжатия изображений можно использовать такие двухмерные преобразования, как дискретное преобразование Фурье, преобразование Адамара, дискретное косинусное преобразование (ДКП), дискретные преобразования Хартли, Хаара и ряд других.

### **7.2.5. Дискретное косинусное преобразование**

Дискретное косинусное преобразование непосредственно связано с дискретным преобразованием Фурье. Недостаток ДПФ заключается в том, что спектральные коэффициенты носят комплексный характер. Однако можно осуществить такое преобразование множества отсчетов сигнала  $\{s_n\}$ , в котором используется только реальная часть ядра преобразования ДПФ, т. е. только члены, связанные с  $\cos$ .

Используя запись ДПФ, получаем выражение ДКП [3]:

$$\begin{cases} C_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n, \\ C_k = \sqrt{\frac{2}{N}} \cdot \sum_{n=0}^{N-1} s_n \cdot \cos \left[ \frac{2n+1}{2N} k\pi \right], \end{cases}$$

где  $k = \{1, 2, \dots, N-1\}$ .

В соответствии с выражением, спектральные коэффициенты одномерного ДКП порядка  $N$  можно определить через вычисления спектральных коэффициентов ДПФ порядка  $2N$ . Формулу для одномерного обратного ДКП найдем из условий ортогональности системы дискретных косинусных функций. Используя взаимосвязь с ДПФ, обратное ДКП можно записать как:

$$s_n = \frac{1}{\sqrt{N}} \hat{C}_0 + \sqrt{\frac{2}{N}} \cdot \operatorname{Re} \left\{ \sum_{k=1}^{2N-1} \exp \left[ j \frac{k\pi}{2N} \right] \hat{C}_k \overline{W}^{kn} \right\},$$

где  $\hat{C}_k = \begin{cases} C_k & \text{для } k \in \{1, 2, \dots, N-1\}, \\ 0 & \text{для } k \in \{N, N+1, \dots, 2N-1\}, \end{cases}$ ,

$$\overline{W} = \exp \left( j \frac{2\pi}{2N} \right), \quad n \in \{1, 2, \dots, N-1\}.$$

Упрощенная формула обратного одномерного ДКП имеет вид

$$s_n = \frac{1}{\sqrt{N}} \hat{C}_0 + \sqrt{\frac{2}{N}} \cdot \sum_{k=1}^{N-1} C_k \cos \left[ \frac{2n+1}{2N} k\pi \right], \quad n \in \{0, 1, \dots, N-1\}.$$

Если перераспределить нормирующие коэффициенты выражений, константу нормирования включить в выражение для прямого преобразования, тогда прямое преобразование ДПК будет определяться выражением

$$C(k) = \frac{2}{N} c(k) \sum_{n=0}^{N-1} s_n \cos \left[ \frac{2n+1}{2N} k\pi \right], \quad k \in \{0, 1, \dots, N-1\},$$

а обратное преобразование:

$$s_n = \sum_{k=0}^{N-1} c_k C_k \cos \left[ \frac{2n+1}{2N} k\pi \right], \quad n \in \{0, 1, \dots, N-1\},$$

где  $c_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{для } k = 0, \\ 1 & \text{для } k \in \{1, 2, \dots, N-1\}. \end{cases}$

Прямое ДКП двухмерного фрагмента изображения размером  $N \times N$  в матричной форме запишется как

$$C_{k_1, k_2} = \Phi_{k_1, n_1} s_{n_1, n_2} \Phi_{k_2, n_2}^T,$$

$$k_1, n_1, k_2, n_2 \in \{0, 1, \dots, N-1\},$$

где  $C_{k_1, k_2}$  – матрица спектральных коэффициентов ДКП размером  $N \times N$ ;

$s_{n_1, n_2}$  – сигнальная матрица размером  $N \times N$ ;

$\Phi_{k_1, n_1}$  – матрица ДКП размером  $N \times N$ :

$$\Phi_{k_1, n_1} = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \cos \left[ \frac{2n_1 + 1}{2N} \cdot k_1 \pi \right] \end{bmatrix}, k_1 \in \{1, 2, \dots, N-1\}, n_1 \in \{0, 1, \dots, N-1\};$$

$\Phi_{k_2, n_2}$  – матрица ДКП размером  $N \times N$ :

$$\Phi_{k_2, n_2} = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \cos \left[ \frac{2n_2 + 1}{2N} \cdot k_2 \pi \right] \end{bmatrix}, k_2 \in \{1, 2, \dots, N-1\}, n_2 \in \{0, 1, \dots, N-1\}.$$

Ядро ДКП для  $N = 4$  можно определить следующим образом:

$$(\Phi_{k,n})_4 = \sqrt{\frac{2}{4}} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \cos \left[ \frac{2n+1}{2 \cdot 4} \cdot k \pi \right] \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{8} & \cos \frac{3\pi}{8} & \cos \frac{5\pi}{8} & \cos \frac{7\pi}{8} \\ \cos \frac{2\pi}{8} & \cos \frac{6\pi}{8} & \cos \frac{10\pi}{8} & \cos \frac{14\pi}{8} \\ \cos \frac{3\pi}{8} & \cos \frac{9\pi}{8} & \cos \frac{15\pi}{8} & \cos \frac{21\pi}{8} \end{bmatrix} =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{8} & \cos \frac{3\pi}{8} & -\cos \frac{3\pi}{8} & -\cos \frac{\pi}{8} \\ \cos \frac{2\pi}{8} & -\cos \frac{2\pi}{8} & -\cos \frac{2\pi}{8} & -\cos \frac{2\pi}{8} \\ \cos \frac{3\pi}{8} & -\cos \frac{\pi}{8} & -\cos \frac{\pi}{8} & -\cos \frac{3\pi}{8} \end{bmatrix} = \begin{bmatrix} 0,5 & 0,5 & 0,5 & 0,5 \\ 0,653 & 0,271 & -0,271 & -0,653 \\ 0,5 & -0,5 & -0,5 & 0,5 \\ 0,271 & -0,653 & 0,653 & -0,271 \end{bmatrix}.$$

Аналогичным образом можно вычислить ядро ДКП для  $N = 8$ :

$$(\Phi_{k,n})_8 = \begin{bmatrix} 0,354 & 0,354 & 0,354 & 0,354 & 0,354 & 0,354 & 0,354 & 0,354 \\ 0,490 & 0,416 & 0,278 & 0,098 & -0,098 & -0,278 & -0,416 & -0,490 \\ 0,462 & 0,191 & -0,191 & -0,462 & -0,462 & -0,191 & 0,191 & 0,462 \\ 0,416 & -0,098 & -0,490 & -0,278 & 0,278 & 0,490 & 0,098 & -0,416 \\ 0,354 & -0,354 & -0,354 & -0,354 & -0,354 & -0,354 & -0,354 & -0,354 \\ 0,278 & -0,490 & 0,098 & 0,416 & -0,416 & -0,098 & 0,490 & -0,278 \\ 0,191 & -0,462 & 0,462 & -0,191 & -0,191 & 0,462 & -0,462 & 0,191 \\ 0,098 & -0,278 & 0,416 & -0,490 & 0,490 & -0,416 & 0,278 & -0,098 \end{bmatrix}.$$

### 7.3. Стандарты сжатия JPEG и JPEG 2000

#### 7.3.1. Основные шаги стандарта сжатия JPEG

JPEG – Joint Photographers Experts Group – сжатие с потерей информации.

Рассмотрим работу алгоритма сжатия JPEG при кодировании полуточного изображения, представленного набором минимальных элементов изображений (пикселов) с числом градаций яркости в 256 уровней [3].

1. Работа алгоритма сжатия JPEG начинается с *разбиения изображения на квадратные блоки* размером  $8 \times 8 = 64$  пиксела.

Выбор размера блока обусловлен тем, что при меньшем размере блока эффект кодирования будет небольшим (при размере  $1 \times 1$  – вообще отсутствует), а при большом – свойства изображения в пределах блока будут сильно изменяться при увеличении степени сжатия и эффективность кодирования снова снизится.

На рис. 7.7 изображено несколько таких блоков (в виде матриц цифровых отсчетов), взятых из различных участков изображения. В дальнейшем эти блоки будут обрабатываться и кодироваться независимо друг от друга.

2. Второй этап сжатия – применение ко всем блокам *дискретного косинусного преобразования*.

В результате применения к блоку изображения размером  $8 \times 8$  пикселов дискретного косинусного преобразования (путем умножения матрицы пикселей на матрицу коэффициентов) получим двумерный спектр, также имеющий размер  $8 \times 8$  отсчетов.

Таким образом, 64 числа, представляющие отсчеты изображения, превратятся в 64 числа, представляющие отсчеты его ДКП-спектра.

Спектр сигнала (как известно, изображение есть двумерный сигнал) – величины коэффициентов, с которыми соответствующие спектральные составляющие в сумме формируют сигнал. Как известно, отдельные спектральные составляющие, на которые раскладывается сигнал, часто называют базисными функциями.

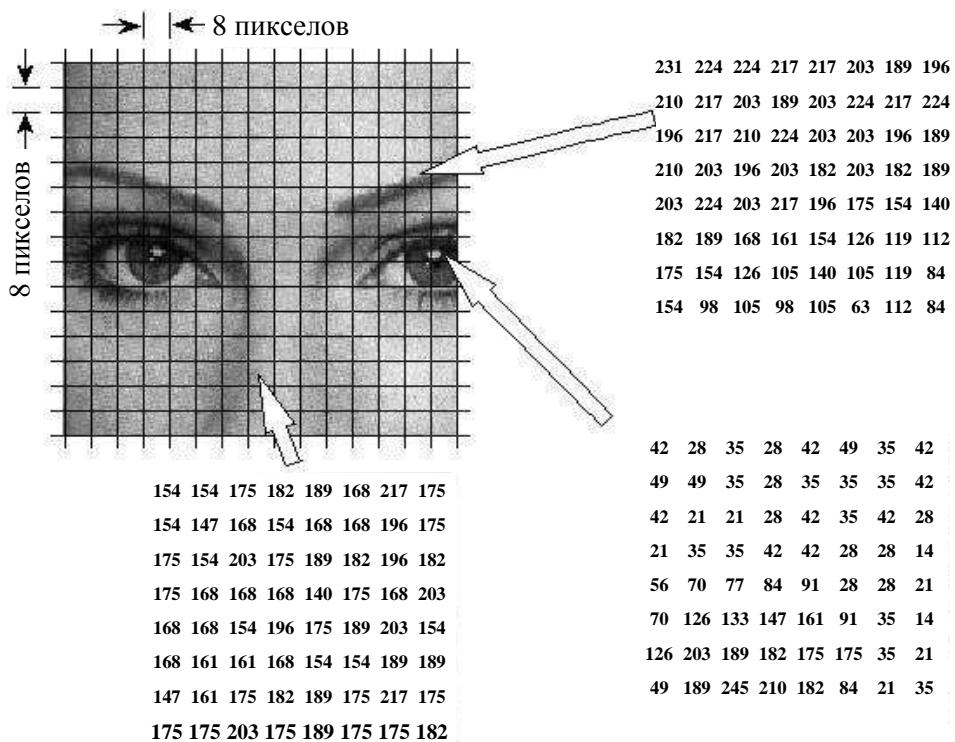


Рис. 7.7. Пример разбиения изображения на блоки

Самая низкочастотная базисная функция, соответствующая индексам (0,0), изображена в левом верхнем углу рисунка, самая высокочастотная – в нижнем правом.

В результате, к примеру, компонента ДКП -спектра с индексами (0,0) будет представлять собой просто сумму всех элементов блока изображения, то есть среднюю для блока яркость. Чем ниже и правее в матрице ДКП его компонента, тем более высокочастотным деталям изображения она соответствует.

Для того, чтобы получить исходное изображение по его ДКП-спектру (выполнить обратное преобразование), нужно теперь базисные функции умножить на спектральные коэффициенты. В приведенной ниже табл. 7.1 видны числовые значения одного из блоков изображения и его ДКП-спектра.

Таблица 7.1

**Пример ДКП для блока изображения**

Элементы цифрового изображения							
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	161	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
161	162	161	163	162	157	157	157
162	162	161	161	163	158	158	15
Результат ДКП							
235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	-1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

Особенность полученного спектра заключается в том, что наибольшие его значения сосредоточены в левом верхнем углу табл. 7.1 (низкочастотные составляющие), правая же нижняя его часть (высокочастотные составляющие) заполнена относительно небольшими числами.

Если выполнить обратное преобразование, получим тот же самый блок изображения.

Однако анализ результата ДКП показывает, что значения многих коэффициентов равны или близки к нулю. Они соответствуют амплитудам высоких частот, которые (обычно) могут быть отброшены или, по крайней мере, округлены до ближайшего целого значения.

3. Следующим этапом работы алгоритма JPEG является *квантование* (табл. 7.2).

Квантование заключается в делении каждого коэффициента ДКП на некоторое число в соответствии с матрицей квантования. Эта матрица может быть фиксированной либо, для более качественного и эффективного сжатия, получена в результате анализа характера исходной картинки. Чем больше числа, на которые происходит деление, тем больше в результате деления будет нулевых значений, а значит, сильнее сжатие и заметнее потери.

Таблица 7.2

**Пример квантования**

Ранее полученный результат ДКП							
235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	-1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4
Таблица квантования							
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99
Результат квантования							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

От выбора таблицы квантования будет в значительной степени зависеть как эффективность сжатия – число нулей в квантованном (округленном) спектре, так и качество восстановленной картинки. Рассмотренные таблицы включены в стандарт и получены опытным путем.

4. Следующим этапом работы алгоритма JPEG является преобразование  $8 \times 8$  матрицы ДКП-спектра в линейную последовательность.

Делается это таким образом, чтобы сгруппировать по возможности вместе все большие значения и все нулевые значения спектра. Для этого нужно прочесть элементы матрицы коэффициентов ДКП в порядке, изображенном на рис. 7.8, то есть зигзагообразно – из левого верхнего угла к правому нижнему. Эта процедура называется *зигзаг-сканированием*.

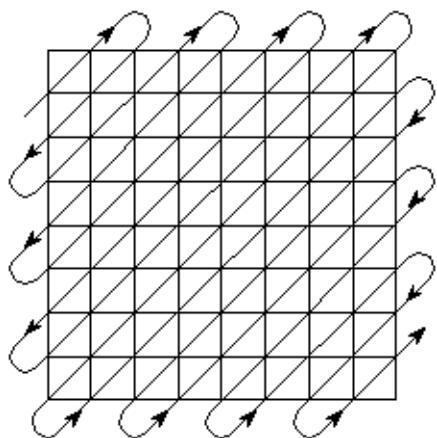


Рис. 7.8. Зигзаг-сканирование

В результате такого преобразования квадратная матрица  $8 \times 8$  квантованных коэффициентов ДКП превратится в линейную последовательность из 64 чисел, большая часть из которых – это идущие подряд нули. Известно, что такие потоки можно очень эффективно сжимать путем *кодирования длин повторений*. Именно так это и делается.

5. На следующем, пятом этапе JPEG-кодирования, получившиеся цепочки нулей подвергаются кодированию длин повторений.

6. Последним этапом работы алгоритма JPEG является *кодирование получившейся последовательности алгоритмом Хаффмена*.

В результате получается новая последовательность, размер которой существенно меньше размера массива исходных данных.

Последние два этапа кодирования обычно называют вторичным сжатием, и именно здесь происходит неразрушающее статистическое кодирование, и с учетом характерной структуры данных – существенное уменьшение их объема.

Декодирование данных сжатых согласно алгоритму JPEG производится точно так же, как и кодирование, но все операции следуют в обратном порядке.

После неразрушающей распаковки методом Хаффмена и расстановки линейной последовательности в блоки размером  $8 \times 8$  чисел *спектральные компоненты деквантируются* с помощью сохраненных при кодировании таблиц квантования. Для этого распакованные 64 значения ДКП умножаются на соответствующие числа из таблицы.

После этого каждый блок подвергается обратному косинусному преобразованию, процедура которого совпадает с прямым и отличается только знаками в матрице преобразования.

Последовательность действий при декодировании и полученный результат иллюстрируются в табл. 7.3

Очевидно, что восстановленные данные несколько отличаются от исходных.

Таблица 7.3

**Декодирование блока изображения**

Квантованные данные							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
Деквантованные данные							
240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
Результат обратного ДКП							
144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158
Для сравнения – исходные данные							
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	161	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
161	162	161	163	162	157	157	157
162	162	161	161	163	158	158	15

На рис. 7.9 представлены результаты применения алгоритма JPEG.

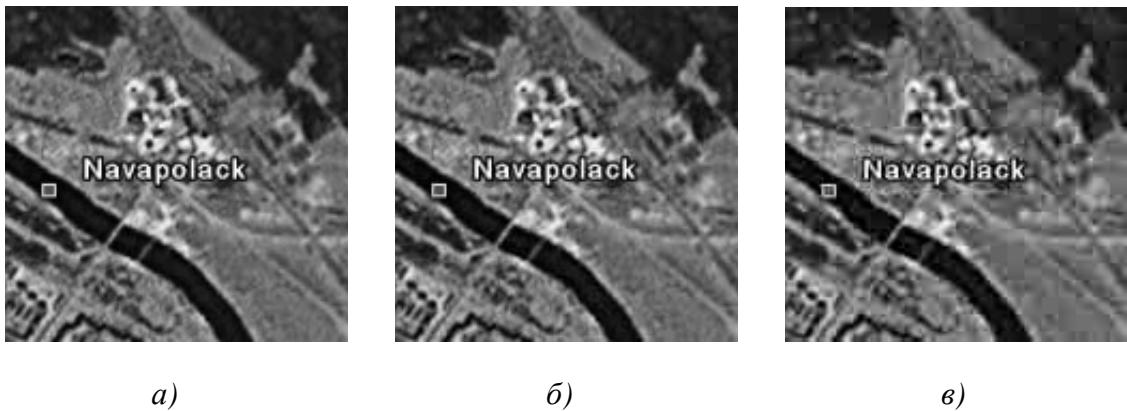


Рис. 7.9. Пример сжатия изображения алгоритмом JPEG:  
а) исходное изображение; б) сжатое алгоритмом JPEG со средним качеством;  
в) сжатое алгоритмом JPEG с низким качеством

Если исходное изображение представлено в цветовом пространстве RGB, то предварительно, перед сжатием, оно переводится в пространство YCrCb (YUV). В этом пространстве Y-яростная составляющая, Cr – компонента хроматического красного цвета, Cb – компонента хроматического синего цвета. За счет того, что человеческий глаз менее чувствителен к цвету, чем к яркости, появляется возможность архивировать массивы Cr и Cb с большими потерями.

Упрощенно перевод из RGB в YCrCb можно представить:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}.$$

Обратный перевод осуществляется:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} * \left( \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \right).$$

К достоинствам алгоритма сжатия JPEG относят [3]:

- возможность задания в широких пределах (от 2 до 200) степени сжатия;
- возможность работы с изображениями любой разрядности;
- симметричность процедур сжатия – распаковки.

К недостаткам можно отнести наличие ореола на резких переходах цветов – эффект Гиббса, а также распадение изображения на отдельные квадратики  $8 \times 8$  при высокой степени сжатия.

Формат JPEG стандартизирован в 1991г., но даже в то время существовали алгоритмы, сжимающие сильнее при меньших потерях качества, однако действия разработчиков стандарта были ограничены мощностью существовавшей на тот момент техники. На персональном компьютере алгоритм должен работать меньше минуты на среднем изображении и алгоритм должен быть симметричным относительно времени архивации и разархивации [3].

Для JPEG характерно также то, что нередко горизонтальные и вертикальные полосы на дисплее абсолютно не видны и могут проявиться только при печати в виде муарового узора. Он возникает при наложении наклонного раstra печати на горизонтальные и вертикальные полосы изображения. Из-за этого недостатка JPEG не рекомендуется активно использовать в полиграфии, задавая высокие коэффициенты матрицы квантования. Однако при архивации изображений, предназначенных для просмотра человеком, он на данный момент незаменим.

Кроме того, если имеющиеся изображения, допустим, в 8-битном формате GIF перевести в 24-битный JPEG, а потом обратно в GIF для просмотра, то потеря качества произойдет дважды при обоих преобразованиях. Тем не менее, выигрыш в размерах архивов зачастую настолько велик (в 3 – 20 раз), а потери качества настолько малы, что хранение изображений в JPEG оказывается очень эффективным.

Хотя JPEG и является стандартом ISO, формат его файлов не был зафиксирован. Пользуясь этим, производители создают свои, несовместимые между собой форматы, и, следовательно, могут изменить алгоритм. Так, внутренние таблицы алгоритма, рекомендованные ISO, заменяются ими на другие. Кроме этого, легкая неразбериха присутствует при задании степени потерь. Например, при тестировании выясняется, что «отличное» качество, «100%» и «10 баллов» дают существенно различающиеся картинки. При этом «100%» качества не означают сжатие без потерь. Встречаются также варианты JPEG для специфических приложений.

Как стандарт ISO JPEG начинает широко использоваться при обмене изображениями в компьютерных сетях. Поддерживается алгоритм JPEG в форматах Quick Time, PostScript Level 2, Tiff 6.0 и т. д., т. е. на данный момент занимает видное место в системах мультимедиа.

### **7.3.2. Особенности стандарта сжатия JPEG 2000**

Алгоритм JPEG 2000 разработан той же группой экспертов в области фотографии, что и JPEG. Формирование JPEG как международного стандарта было закончено в 1992 году. В 1997 стало ясно, что необходим новый, более гибкий и мощный стандарт, который и был доработан к зиме 2000 года. Основные отличия алгоритма в JPEG 2000 от алгоритма в JPEG заключаются в следующем [3]:

1. Лучшее качество изображения при одинаковой степени сжатия. Или большая степень сжатия при том же качестве для высоких степеней сжатия. Фактически это означает заметное уменьшение размеров графики «Web-качества», используемой большинством сайтов.

2. Поддержка кодирования отдельных областей с лучшим качеством. Известно, что отдельные области изображения критичны для восприятия человеком (например, глаза на фотографии), в то время как качеством других можно пожертвовать (например, задний план). При «ручной» оптимизации увеличение степени сжатия проводится до тех пор, пока не будет потеряно качество в какой-то важной части изображения. Сейчас появляется возможность задать качество в критичных областях, сжав остальные области сильнее, т. е. мы получаем еще большую окончательную степень сжатия при субъективно равном качестве изображения.

3. Вместо ДКП используется вейвлет-преобразование. Помимо указанного повышения степени сжатия это позволило избавиться от 8-пиксельной блочности, возникающей при повышении степени сжатия. Кроме этого, плавное проявление изображения теперь изначально заложено в стандарт (Progressive JPEG, активно применяемый в Интернет, появился много позднее JPEG).

4. Для повышения степени сжатия в алгоритме используется арифметическое сжатие вместо алгоритма Хаффмена. Изначально в стандарте JPEG также было заложено арифметическое сжатие, однако позднее оно было заменено менее эффективным сжатием по Хаффмену, поскольку арифметическое сжатие было защищено патентами. Сейчас срок действия основного патента истек, и появилась возможность улучшить алгоритм.

5. Поддержка сжатия без потерь. Помимо привычного сжатия с потерями новый JPEG теперь будет поддерживать и сжатие без потерь. Таким образом, становится возможным использование JPEG для сжатия медицинских изображений, в полиграфии, при сохранении текста под распознавание OCR системами и т. д.

6. Поддержка сжатия однобитных (двухцветных) изображений. Для сохранения 1-битных изображений (рисунки тушью, отсканированный текст и т. п.) ранее повсеместно рекомендовался формат GIF, поскольку сжатие с использованием ДКП весьма неэффективно к изображениям с резкими переходами цветов. В JPEG при сжатии 1-битная картинка приводилась к 8-битной, т. е. увеличивалась в 8 раз, после чего делалась попытка сжимать, нередко менее чем в 8 раз. Сейчас можно рекомендовать JPEG 2000 как универсальный алгоритм.

7. На уровне формата поддерживается прозрачность. Плавно накладывать фон при создании WWW страниц теперь можно будет не только в GIF, но и в JPEG 2000. Кроме того, поддерживается не только 1 бит прозрачности (пиксел прозрачен/непрозрачен), а отдельный канал, что позволит задавать плавный переход от непрозрачного изображения к прозрачному фону.

Кроме этого, на уровне формата поддерживаются включение в изображение информации о копирайте, поддержка устойчивости к битовым ошибкам при передаче и широковещании, можно запрашивать для декомпрессии или обработки внешние средства (plug-ins), можно включать в изображение его описание, информацию для поиска и т. д.

Таким образом, в целом, базовая схема JPEG 2000 схожа с базовой схемой JPEG, а основные особенности следующие:

- 1) вместо дискретного косинусного преобразования используется дискретное вейвлет-преобразование;
- 2) вместо кодирования по Хаффману используется арифметическое сжатие;
- 3) в алгоритм изначально заложено управление качеством областей изображения;
- 4) не используется явно дискретизация компонент  $U$  и  $V$  после преобразования цветовых пространств, поскольку при DWT можно достичь того же результата, но более аккуратно.

#### **7.4. Вопросы и задания для самопроверки**

1. Запишите и поясните выражения для двукратного двумерного преобразования Фурье.
2. Какие особенности имеет спектр изображений?
3. Для решения каких задач используется представление изображений в частотной области?
4. Сформулируйте алгоритм фильтрации изображений в частотной области.

5. Приведите примеры характеристик фильтров, используемых при фильтрации изображений в частотной области?
6. Сформулируйте основные требования, предъявляемые к алгоритмам сжатия.
7. Каковы основные критерии сравнения алгоритмов сжатия изображений?
8. В чем заключается сущность спектральных методов сжатия изображений?
9. Запишите выражения для прямого и обратного ДКП? В чем отличие его от ДПФ?
10. Назовите основные шаги алгоритма JPEG?
11. Какими факторами обусловлен выбор размера блока обрабатываемого изображения в алгоритме JPEG?
12. Запишите правило для перевода изображения из RGB системы в YCrCb.
13. Каковы недостатки алгоритма JPEG?
14. В чем отличие алгоритма JPEG 2000 от алгоритма в JPEG?
15. Представьте базовую структуру алгоритма JPEG 2000.

## **8. СРАВНЕНИЕ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ И ОБНАРУЖЕНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ**

Цель модуля – изучение студентами мер, вычисляющих сходство между изображениями, и алгоритмов поиска объектов на изображениях.

В результате изучения модуля студенты *должны знать*:

- требования к мерам, вычисляющим сходство между изображениями;
- основные функции схожести корреляционного типа;
- методы сокращения вычислительных затрат при поиске и сравнении изображений.

Студенты *должны уметь*:

- реализовывать эффективные алгоритмы для поиска и сравнения объектов;
- а также иметь представление обо всех основных подходах к сравнению изображений и их особенностях.

### **8.1. Сравнение изображений**

Прикладные системы обработки изображений, использующие сравнение или поиск [12,13,16]:

- системы машинного зрения для технологических процессов;
- системы для диагностики и мониторинга состояния объектов;
- системы поиска и распознавания объектов для обработки медицинских изображений, изображений карт земной поверхности, чертежей, фотоснимков и др. сложных изображений;
- системы видеонаблюдения (детектирование движущихся объектов).

#### **8.1.1. Метрические характеристики**

При сравнении и анализе изображений, при построении их цифровых моделей приходится вычислять расстояния между точками, точками и множествами точек, подсчитывать площади, объемы и другие геометрические величины, вычислять различные геометрические оценки и топологические отношения. Ключевую роль при этом играет понятие расстояния. Это исключительно геометрическое понятие, однако оно широко используется для оценки степени сходства даже таких негеометрических объектов, как изображения и образы. При этом полагается, что чем меньше расстояние между сравниваемыми объектами, тем более они схожи между собой. Однако это справедливо лишь при удачно выбранном метрическом пространстве, т. е. множестве сравниваемых признаков и базисной функции расстояния.

Расстояние – это вещественная функция  $d[(x_i, y_i), (x_j, y_j)]$  двух точек  $(x_i, y_i)$  и  $(x_j, y_j)$ , обладающая следующими свойствами [12]:

$$\begin{aligned} d[(x_i, y_i), (x_j, y_j)] &\geq 0, \\ d[(x_i, y_i), (x_j, y_j)] &= d[(x_j, y_j), (x_i, y_i)], \\ d[(x_i, y_i), (x_j, y_j)] + d[(x_j, y_j), (x_k, y_k)] &\geq d[(x_i, y_i), (x_k, y_k)]. \end{aligned}$$

Известно довольно много функций, удовлетворяющих условиям. Большинство обычных метрик, встречающихся в задачах анализа изображений, имеет следующий вид:

– евклидово расстояние:

$$d_E = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2};$$

– абсолютное расстояние:

$$d_M = |x_i - x_j| + |y_i - y_j|;$$

– максимальное расстояние:

$$d_X = \max \{|x_i - x_j|, |y_i - y_j|\}.$$

В случае дискретного изображения разности координат  $x_i - x_j$  и  $y_i - y_j$  представляют собой целые числа, а евклидово расстояние обычно не целочисленно. Это обстоятельство неизбежно ведет к ошибке вследствие округления или усечения числа при цифровой обработке.

### 8.1.2. Проблема сравнения видеоданных

Проблема построения меры, позволяющей оценить степень сходства кривых и различных объектов по их изображениям, а также изображений и видеосигналов, в целом очень актуальна и сложна, она имеет как теоретическое, так и практическое значение [13].

Сравнение бинарных изображений изучено достаточно полно, для частных задач получены интересные результаты. Однако попытки сравнения полутонаовых изображений путем адаптации функций, предложенных для бинарных изображений, показывают слабость таких оценок. Поэтому те немногие исследования, которые посвящены этой проблеме, носят частный характер. В результате при изменении природы видеоданных, их размерности, диапазонов допустимых значений и прочих параметров пользователям приходится искать новые меры для оценки сходства.

Пусть дано несколько цифровых изображений  $A, B_1, B_2, \dots$  одинакового размера  $N \times N$  с одинаковым диапазоном допустимых значений яркости  $G$ . Требуется определить, какое из цифровых изображений  $B_i$  более других похоже на цифровое изображение  $A$ , а в общем случае – насколько похожи сцены, передаваемые изображениями  $B_i$  и изображением  $A$ ?

Эта проблема стала еще более актуальной в последние годы, так как растут возможности получения и хранения изображений различного рода. Накапливаются цифровые фотографии людей, снимки из космоса и т. д. Цифровые изображения «складываются» в огромные базы данных, содержащие изображения. Возникает задача поиска изображения в базе по его словесному описанию или по образцу в виде некого изображения.

Количество возможных вариантов только черно-белых изображений при заданных размерах  $N \times N$  и диапазоне значений  $G$  равно  $G^{N \times N}$ . Для средних размеров изображений  $1024 \times 1024$  и если  $G = 256$  возможных оттенков у полутоновых изображений –  $256^{1024 \times 1024}$ , для цветных изображений следует учитывать, что  $G = 256 \times 3$  оттенков. На рис. 8.1. представлены для сравнения три очень простых изображения. Определите, какое из двух более схоже с изображением  $A$ .

Следствием этого гигантского объема информации является отсутствие универсального определения или численной оценки подобия изображений.

Каждый исследователь использует свою модель подобия.

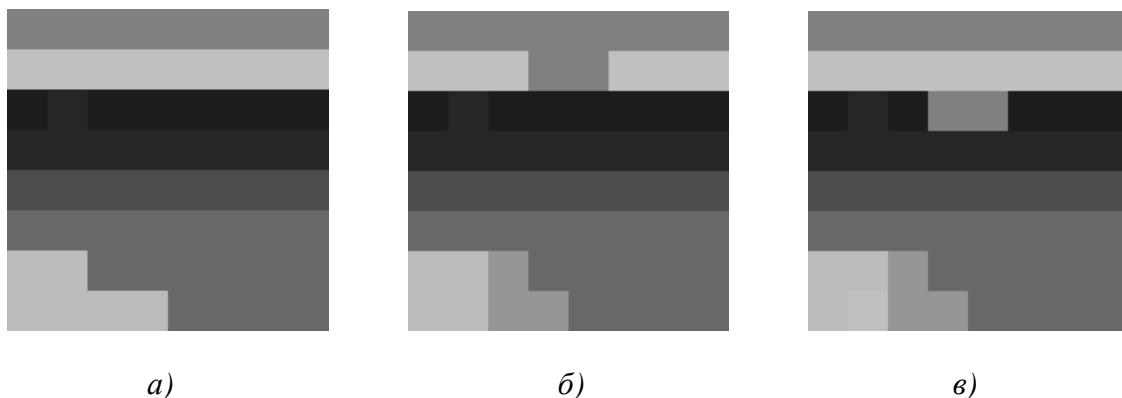


Рис. 8.1. Изображения для сравнения:  
а) изображение  $A$ ; б) изображение  $B_1$ ; в) изображение  $B_2$

Существуют три основных подхода к сравнению изображений [13]:

1. Человеческое восприятие, дающее субъективную оценку.
2. Математические меры, опирающиеся на модель зрения человека.
3. Объективные меры, построенные на теоретических моделях.

Наиболее широко при автоматической обработке используется третий подход. Методы и алгоритмы, сравнивающие изображения или их отдельные части, можно сгруппировать в зависимости от ключевых понятий, используемых в целях сравнения, следующим образом: уровень точек; уровень пикселей; точки интереса; сегменты и кривые.

Методы первого уровня базируются на операциях теории множеств (объединение, пересечение и принадлежность), исходными данными для них служат изображения, представленные в виде множеств точек. Такой подход к сходству можно назвать теоретико-множественным и он достаточно эффективен при вычислении сходства бинарных изображений. Легко проверить, что теоретико-множественные отношения плохо оценивают сходства похожих, но сдвинутых друг относительно друга мелкоконтрастных изображений одной и той же сцены.

Методы второго уровня можно разделить на следующие подуровни: классические корреляционные функции, меры на базе локальных расстояний. В соответствии с предложенной классификацией к этому же уровню примыкают и исследования, связанные с вычислением и анализом оптических потоков (optical flow).

В случае сравнения на локальном уровне, уровне пикселей изображений результат не зависит от контекста и особенностей изображаемых сцен. Перечисленные и многие другие меры сравнения, особенно имеющие теоретико-множественную природу, не могут быть использованы для сравнения полутоновых, а тем более, цветных изображений без значительной доработки. Кроме того, можно выделить ряд задач обработки и анализа изображений, которые решаются посредством использования той или иной меры сравнения цифровых изображений целиком, как единых объектов.

### **8.1.3. Требования к мерам, вычисляющим сходство**

Чтобы быть уверенным в том, что полученная оценка сходства изображений отражает реальное сходство изображаемых сцен, выбранная функция схожести должна удовлетворять некоторым требованиям и ее свойства должны быть известны. В качестве функции схожести может использоваться мера расстояния либо функция схожести. Если два изображения абсолютно идентичны, то мера расстояния  $M$  между ними равна 0, а нормализованная функция схожести  $R$  равна 1; следовательно:  $M = 1 - R$  или  $R = 1 - M$ .

Пусть  $A = \{a_{ij}\}$ ,  $B = \{b_{ij}\}$  и  $C = \{c_{ij}\}$  обозначают полутоновые изображения. Желательно, чтобы мера  $M$  обладала следующими свойствами [13].

1. Метричность:

- 1)  $M(A, B) = 0 \Leftrightarrow A = B$ ;
- 2)  $M(A, B) = M(B, A)$ ;
- 3)  $M(A, B) \leq M(A, C) + M(C, B)$ .

2. Нормализованность значений:

$$M(A, B) = 1 \Leftrightarrow a_{ij} = 0 \& b_{ij} = G, \forall(i, j).$$

3. Устойчивость к шуму: если изображение  $B_1$  совпадает с изображением  $B_2$  за исключением  $k$  точек, т. е.:

$$M((B_1 \setminus \{b'_{ij} | l = 1..k\}), (B_2 \setminus \{b'_{ij} | l = 1..k\})) = 0,$$

тогда

$$M(A, B_1) = M(A, B_2) \pm \sigma,$$

где  $\sigma$  – достаточно мало.

4. Усредненность оценки, ее независимость от небольших различий между изображениями, т. е. если

$$\begin{aligned} \left\{a_{ij} \setminus \{a_{ij}^l | l = 1..k_1\}\right\} &= \left\{b_{ij} \setminus \{b_{ij}^l | l = 1..k_1\}\right\}, \\ \left\{a_{ij} \setminus \{a_{ij}^m | l = 1..k_2\}\right\} &= \left\{c_{ij} \setminus \{c_{ij}^m | l = 1..k_2\}\right\}, \\ k_1 << k_2 \quad \text{и} \quad \sum_l |b_{ij}^l - a_{ij}^l| &\approx \sum_m |c_{ij}^m - a_{ij}^m|, \end{aligned}$$

тогда

$$M(A, B) \approx M(A, C).$$

5. Монотонность оценки:

$$M(A, B) < M(A, C)$$

означает, что изображение  $A$  больше похоже на изображение  $B$ , чем на изображение  $C$ .

6. Мера  $M$  не должна резко изменяться при небольших изменениях форм изображаемых объектов, при небольших сдвигах и вращениях.

7. Имеет место:

$$M(A, B) \geq M(A, C),$$

если  $B$  получено из  $A$  путем добавления шума, а  $C$  есть восстановленная версия  $B$ .

## 8. Вычисление меры должно быть достаточно быстрым.

Совокупность требований, которым удовлетворяет некоторая мера сходства  $M$ , определяет более строгую либо слабую (на качественном уровне) оценку сходства. Мера является строгой, если она удовлетворяет большинству описанных требований, и слабой, если она не удовлетворяет некоторым требованиям. Например, если мера является квазиметрикой (т. е. не выполняется третья аксиома метрики), то она является более слабой по сравнению с мерой, обладающей такими же свойствами, но дополнительно являющейся метрикой. Тем не менее, выполнение всех аксиом метрики для меры не является обязательным условием при ее практических применениях и не является достаточным условием сильной оценки. Если мера удовлетворяет неравенству треугольника, из этого вытекает следующее свойство: пусть изображения  $A$  и  $B$  похожи и  $A$  и  $C$  также похожи, тогда степень сходства изображений  $B$  и  $C$  не будет превышать величины:

$$M(A, B) + M(A, C) \geq M(B, C).$$

Метрика Хаусдорфа, используемая в качестве меры сходства, очень чувствительна к шуму. Следовательно, сама по себе метричность не гарантирует строгости самой оценки сходства, но в то же время мера, удовлетворяющая совокупности других требований, но не являющаяся метрикой, может в большинстве случаев дать лучшую оценку сходства изображений, чем иная мера, являющаяся метрикой.

Назначение требований – попытка приблизить аналитическую оценку, выраженную одним числом, к субъективному понятию человека о схожести изображений.

Согласно вышеизложенным требованиям, очень похожие цифровые видеоданные должны иметь меру сходства, близкую к нулю, или функцию схожести, близкую к единице.

### 8.1.4. Функции схожести корреляционного типа

Для сравнения двух изображений  $O = \{o_{ij}\}$  и  $B^* = \{b_{ij}^*\}$  можно использовать следующие типы функций схожести [9,13].

1. Нормированная функция корреляции:

$$R^{COR} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} o_{ij} b_{ij}^*}{\sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij})^2} \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (b_{ij}^*)^2}}.$$

2. Нормированная усредненная корреляционная функция:

$$R^{\overline{COR}} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij} - \overline{o})(b_{ij}^* - \overline{b}^*)}{\sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij} - \overline{o})^2} \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (b_{ij}^* - \overline{b}^*)^2}},$$

где  $\overline{o}$  и  $\overline{b}^*$  – средние значения изображений  $O$  и  $B^*$ :

$$\overline{o} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} o_{ij}, \quad \overline{b}^* = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{ij}^*.$$

3. Функция на основе суммы квадратов разностей:

$$R^{SSD} = 1 - \frac{1}{LN^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij} - b_{ij}^*)^2,$$

где  $L$  – диапазон допустимых значений яркости.

4. Функция на основе взвешенной суммы квадратов разностей:

$$R^{SSD'} = 1 - \frac{1}{L} \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij} - b_{ij}^*)^2}{\sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij})^2} \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (b_{ij}^*)^2}}.$$

5. Функция на основе метрики Хаусдорфа:

$$R^H = 1 - \frac{1}{L} \max_{ij} |o_{ij} - b_{ij}^*|, \quad i \in 0 \dots N-1, \quad j \in 0 \dots N-1.$$

6. Функция на основе городской метрики:

$$R^C = 1 - \frac{1}{LN^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |o_{ij} - b_{ij}^*|.$$

7. Функция на основе усредненной городской метрики:

$$R^{C'} = 1 - \frac{1}{LN^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |o_{ij} - b_{ij}^* - \overline{o} + \overline{b}^*|.$$

8. Функция на основе среднеквадратичной погрешности:

$$R^e = 1 - \frac{1}{LN^2} \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij} - b_{ij}^*)^2}.$$

9. Минимаксная функция схожести:

$$R^{SM} = \min_{i,j \in 0, N-1} \left( \frac{\min(o_{ij}, b_{ij}^*)}{\max(o_{ij}, b_{ij}^*)} \right).$$

10. Минимаксная мультипликативная:

$$R^M = \prod_{i=0}^{N-1} \prod_{j=0}^{N-1} \frac{\min(o_{ij}, b_{ij}^*)}{\max(o_{ij}, b_{ij}^*)}.$$

11. Минимаксная аддитивная:

$$R^S = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{\min(o_{ij}, b_{ij}^*)}{\max(o_{ij}, b_{ij}^*)}.$$

12. Минимаксная аддитивная степенная:

$$R^{SDP} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{\min(o_{ij}^P, b_{ij}^{*P})}{\max(o_{ij}^P, b_{ij}^{*P})}, p > 1.$$

13. Минимаксная усредненная аддитивная:

$$\begin{cases} \text{если } |o_{ij} - \bar{o}| > |b_{ij} - \bar{b}|: \quad \overline{R^S} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{b_{ij} - \bar{b}}{o_{ij} - \bar{o}}; \\ \text{иначе:} \quad \overline{R^S} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{o_{ij} - \bar{o}}{b_{ij} - \bar{b}}. \end{cases}$$

### 8.1.5. Алгоритм поиска объектов

Поиск объектов на изображении методом сопоставления с эталоном является одним из основных методов обнаружения. При этом эталон сравнивается со всеми объектами, находящимися на изображении, путем последовательного перемещения по изображению, как правило, слева направо, сверху вниз. Полное совпадение эталона с какой-либо частью изображения бывает редко из-за действия шумов и искажений, вызванных пространственной дискретизацией и квантованием яркости, а также вследствие отсутствия априорной информации относительно точной формы и структуры объекта, ко-

торый требуется обнаружить. Поэтому для оценки схожести объекта и фрагмента изображения используется функция схожести. В случае если рассчитанная величина больше порогового значения, то принимается решение о наличии объекта на изображении. Таким образом, исходными данными являются: эталонное изображение  $O$ , обрабатываемое изображение  $B$ , значение пороговой величины  $T$ . Следует отметить, что одной из проблем является точность позиционирования объекта на изображении, которая связана с размытостью основного корреляционного пика и наличием интенсивных боковых выбросов при анализе изображения в смеси с шумом. В связи с этим, выбор оптимальной пороговой величины в значительной мере определяет эффективность поиска объектов в целом, даже незначительно изменение пороговой величины может привести к увеличению вероятности ложного обнаружения либо к увеличению вероятности пропуска объекта (рис. 8.2).

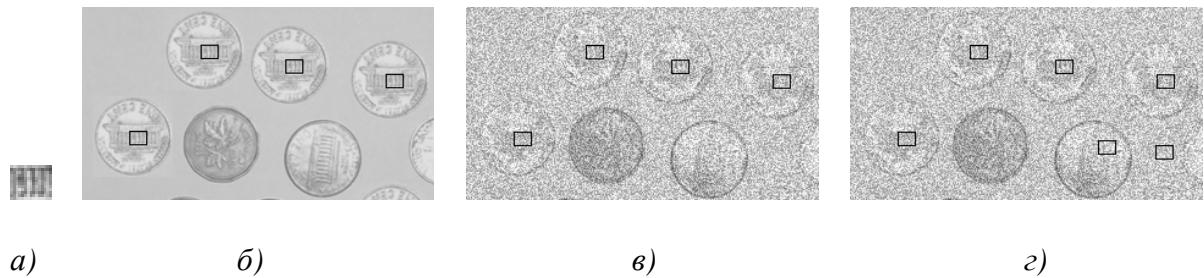


Рис. 8.2. Результаты поиска: *а)* объект; *б)* слабоконтрастное изображение,  $T = 0,99$ ; *в)* изображение с шумом 30%,  $T = 0,9779$ ; *г)* изображение с шумом 30%,  $T = 0,9777$

Алгоритм поиска объекта  $O = o_{ij}$ ,  $i \in [0..N]$ ,  $j \in [0..N]$  на изображении  $B = b_{kq}$ ,  $k \in [0..M_1]$ ,  $q \in [0..M_2]$  в общем случае требует выполнения следующих шагов.

1. Выделение фрагмента  $B_1^*$  размером  $N_1 \times N_2$  с левой верхней части растрового изображения.
2. Вычисление меры сходства для выделенного фрагмента  $B^*$  и объекта  $O$ , сравнение ее с поровым значением и принятие решения о соответствии фрагмента объекту.
3. Сдвиг вправо или вниз на один элемент на принятом изображении, выделение следующего фрагмента размером  $N_1 \times N_2$  и переход к п. 2., если число сдвигов меньше  $(M_1 - N_1) \times (M_2 - N_2)$ . Иначе поиск завершен.

На рис. 8.3 представлены объект и изображение, а на рис. 8.4 – графические функции схожести для данного объекта и изображения.



*a)*

*б)*

Рис. 8.3. Обрабатываемые изображения: *а*) объект; *б*) фрагмент карты

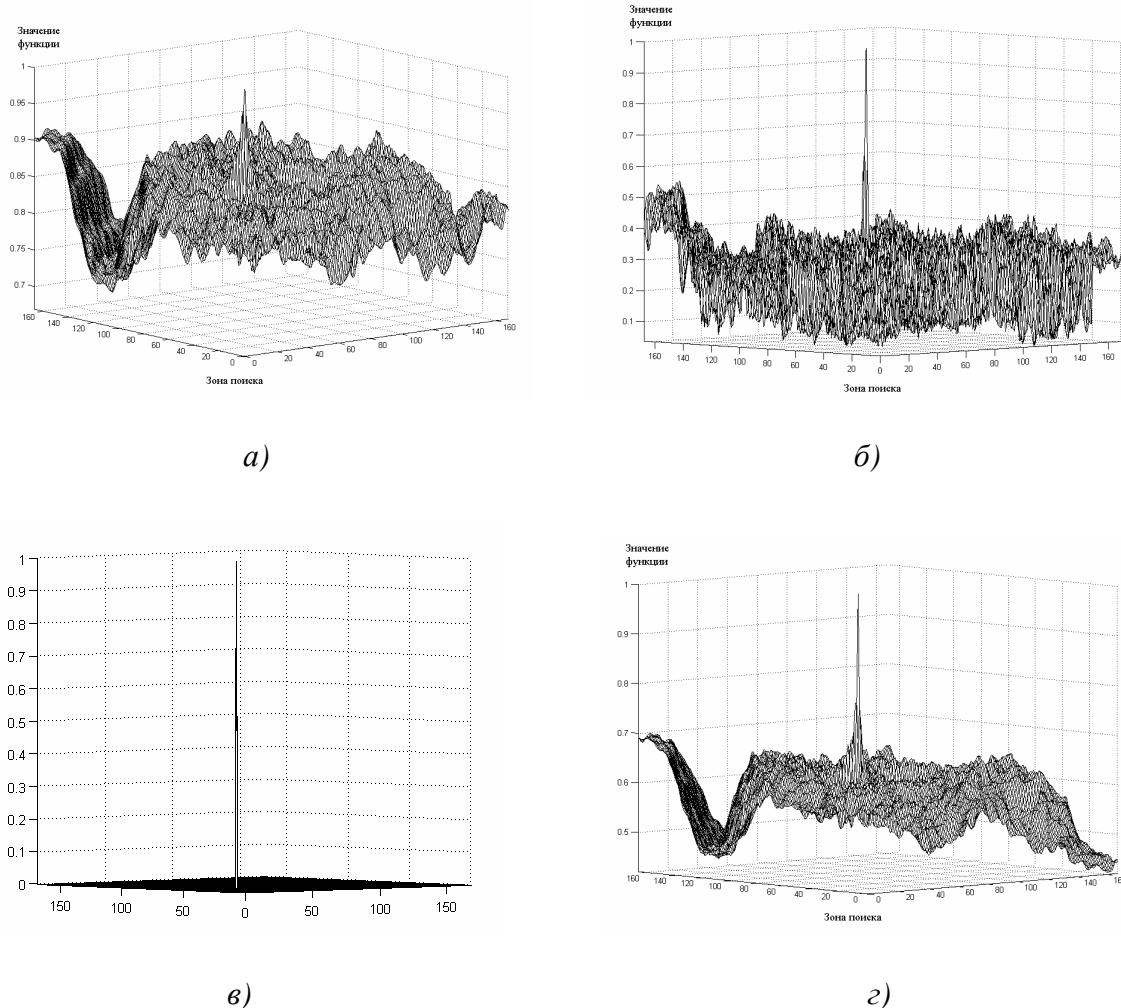


Рис. 8.4. Графики функций схожести для объекта и изображения, представленных на рис. 8.3: *а*) корреляционная функция; *б*) функция на основе метрики Хаусдорфа; *в*) минимаксная мультипликативная; *г*) минимаксная аддитивная

Главный недостаток метода сопоставления с эталоном заключается в необходимости использования огромного количества эталонов для учета изменений объектов, возникающих при их повороте и увеличении (уменьшении) размеров. Этот недостаток корреляционных методов обработки приводит к тому, что при обработке требуются значительные вычислительные, а следовательно и временные затраты для решения практических задач.

## **8.2. Поиск объектов на изображениях**

### **8.2.1. Методы сокращения вычислительных затрат при поиске объектов на статических изображениях**

1. Применение различных двукратных преобразований. Вычисление двумерной корреляции в этом случае требует выполнения следующих действий:

- выполнить прямое преобразование (в базисе Фурье, Хаара, Уолша-Адамара и т. д.), т. е. вычислить спектры для анализируемого фрагмента изображения и эталона путем умножения растровых матриц на матрицу коэффициентов используемого базиса;
- вычислить поточечное произведение полученных векторов;
- произвести обратное преобразование – умножить полученные вектора на матрицу обратного преобразования применяемого базиса.

Использование быстрых двумерных ортогональных преобразований позволяет значительно уменьшить объем вычислений и, следовательно, сократить временные затраты при поиске объектов. Однако этим методам присущи два основных недостатка: первый – двумерная нормированная корреляционная функция зачастую имеет довольно размытый максимум, что затрудняет точное определение координат объекта на изображении; второй – корреляционный поиск не позволяет решать задачу обнаружения объектов при различиях, связанных с поворотом объекта или с изменением масштаба объекта на изображении относительно эталона.

2. Совмещение операции вычисления функции схожести с операцией сравнения с порогом, т. е. вычисление функций прекращается для анализируемой позиции объекта на изображении, если полученное значение функции схожести на некотором шаге меньше установленного порога. В данном случае снижение вычислительных затрат достигается за счет ухудшения помехоустойчивости.

3. Использование характеристик определенного класса изображений. Например, для бинарных изображений процедуру поиска можно значительно упростить, если применять простые арифметические действия вместо спектрального преобразования.

4. Использование моментов строк и столбцов растровых матриц изображений.

### 8.2.3. Обнаружение повернутых объектов

При решении задачи поиска объекта на изображении входной сигнал является стохастическим, т. е. его можно оценить с помощью статистических законов. Одной из наиболее употребляемых характеристик стохастического сигнала является амплитудная плотность, которая характеризует среднее значение сигнала, определяемое в результате усреднения по времени и соответствующее первому моменту плотности распределения амплитуд. Очевидно, что за равное время усреднения одинаковым моментам плотности распределения амплитуд может соответствовать бесконечное множество форм сигналов, поэтому для растровой матрицы изображения  $A = \{a_{i,j}\}$  размером  $N \times N$  вводятся моменты строк  $r^A$ :

$$r_i^A = \frac{1}{N} \sum_{j=0}^{N-1} a_{i,j}, \quad i = 0, \dots, N-1 \quad (1)$$

и столбцов  $c^A$ :

$$c_j^A = \frac{1}{N} \sum_{i=0}^{N-1} a_{i,j}, \quad j = 0, \dots, N-1. \quad (2)$$

Два изображения будем считать идентичными тогда и только тогда, когда у них совпадают вектора моментов строк и столбцов. Таким образом, для определения схожести исходного изображения и повернутого необходимо оценить соответствие векторов моментов строк и столбцов. При этом можно использовать ранее рассмотренные функции схожести. Например, нормированная функция корреляции для моментов строк определяется согласно выражению:

$$R_{r^A, r^B}^{COR} = \frac{\sum_{i=0}^{N-1} r_i^A r_i^B}{\sqrt{\sum_{i=0}^{N-1} (r_i^A)^2} \sqrt{\sum_{i=0}^{N-1} (r_i^B)^2}}.$$

Общая функция соответствия рассчитывается с использованием формулы:

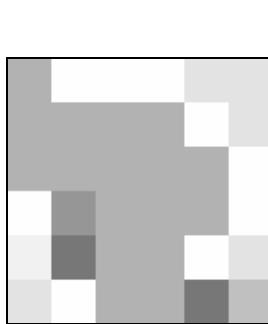
$$R(A, B) = \left( \frac{\sum_{i=0}^{N-1} r_i^A r_i^B}{\sqrt{\sum_{i=0}^{N-1} (r_i^A)^2} \sqrt{\sum_{i=0}^{N-1} (r_i^B)^2}} \right) \cdot \left( \frac{\sum_{j=0}^{N-1} c_j^A c_j^B}{\sqrt{\sum_{j=0}^{N-1} (c_j^A)^2} \sqrt{\sum_{j=0}^{N-1} (c_j^B)^2}} \right) = R_{r^A, r^B}^{COR} R_{c^A, c^B}^{COR}.$$

На основе выражений (1) и (2) и рис. 8.5 можно получить условия сравнения моментов строк и столбцов эталона  $A$  и повернутого против часовой стрелки на угол  $\Theta = 90^\circ$  изображения  $B$ :

$$\Theta = 90 \quad r_i^B = \frac{1}{N} \sum_{j=0}^{N-1} b_{i,j} = \frac{1}{N} \sum_{j=0}^{N-1} a_{j,N-i-1} = c_{N-i-1}^A. \quad (3)$$

$$c_j^B = \frac{1}{N} \sum_{i=0}^{N-1} a_{j,N-i-1} = r_j^A. \quad (4)$$

На рис. 8.5 приведен пример исходного изображения и повернутого на  $90^\circ$  против часовой стрелки, а также показаны соответствующие им растровые матрицы, для которых рассчитаны моменты строк и столбцов.



a)

$r_i^A$	209,5	189,2	190,8	190,8	202	230,5
232,8	<b>178</b>	<b>255</b>	<b>255</b>	<b>255</b>	<b>227</b>	<b>227</b>
199	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>255</b>	<b>227</b>
190,8	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>255</b>
199	<b>255</b>	<b>150</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>255</b>
199,7	<b>241</b>	<b>119</b>	<b>178</b>	<b>178</b>	<b>255</b>	<b>227</b>
191,5	<b>227</b>	<b>255</b>	<b>178</b>	<b>178</b>	<b>119</b>	<b>192</b>

б)



в)

$r_i^B$	232,8	199	190,8	199	199,7	191,5
230,5	<b>227</b>	<b>227</b>	<b>255</b>	<b>255</b>	<b>227</b>	<b>192</b>
202	<b>227</b>	<b>255</b>	<b>178</b>	<b>178</b>	<b>255</b>	<b>119</b>
190,8	<b>255</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>
190,8	<b>255</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>
189,2	<b>255</b>	<b>178</b>	<b>178</b>	<b>150</b>	<b>119</b>	<b>255</b>
209,5	<b>178</b>	<b>178</b>	<b>178</b>	<b>255</b>	<b>241</b>	<b>227</b>

г)

Рис. 8.5: а) исходное изображение; б) исходное изображение в цифровом виде; в) изображение, повернутое на  $90^\circ$ ; г) повернутое изображение в цифровом виде

Согласно выражениям (3) и (4) для определения соответствия эталона и повернутого на угол  $\Theta = 90^\circ$  изображения необходимо рассчитать:

- функцию схожести (в рассматриваемом примере – функцию корреляции) для вектора моментов строк  $r_i^B$  изображения  $B$  и вектора момен-

тов столбцов  $c_j^A$  изображения  $A$ , причем элементы последнего вектора должны быть прочитаны в обратном порядке:

$$R_{r^B, c^A}^{COR} = \frac{\sum_{i=0}^{N-1} r_i^B c_{N-i-1}^A}{\sqrt{\sum_{i=0}^{N-1} (r_i^B)^2} \sqrt{\sum_{i=0}^{N-1} (c_i^A)^2}};$$

- функцию схожести для вектора моментов столбцов  $c_j^B$  изображения  $B$  и вектора моментов строк  $r_i^A$  изображения  $A$ :

$$R_{c^B, r^A}^{COR} = \frac{\sum_{i=0}^{N-1} c_i^B r_i^A}{\sqrt{\sum_{i=0}^{N-1} (c_i^B)^2} \sqrt{\sum_{i=0}^{N-1} (r_i^A)^2}}.$$

Таким образом, общая функция соответствия рассчитывается с использованием формулы:

$$R(A, B^{90}) = R_{r^B, c^A}^{COR} R_{c^B, r^A}^{COR}.$$

Выражения для сравнения моментов строк и столбцов эталона  $A$  и повернутого изображения на угол  $\Theta = 180^\circ$  и  $\Theta = 270^\circ$  изображения  $B$  представлены ниже:

$$\Theta = 180^\circ \quad r_i^B = r_{N-i-1}^A. \quad c_j^B = c_{N-j-1}^A.$$

$$\Theta = 270^\circ \quad r_i^B = c_i^A. \quad c_j^B = r_{N-j-1}^A.$$

Рассмотрим случай зеркального отражения изображений. Если изображение  $A' = \{a'_{i,j}\}$  является зеркальным отражением изображения  $A = \{a_{i,j}\}$  относительно вертикальной оси, то выполняется равенство:

$$a'_{i,j} = a_{i,N-j-1}.$$

В результате получаем:

$$r_i^{A'} = \frac{1}{N} \sum_{j=0}^{N-1} a_{i,N-j-1} = r_i^A; \quad c_j^{A'} = \frac{1}{N} \sum_{i=0}^{N-1} a_{i,N-j-1} = c_{N-j-1}^A.$$

Соотношения между моментами строк и столбцов эталона и повернутого изображения, которое предварительно зеркально отражено относительно вертикальной оси, описываются выражениями:

$$\begin{array}{lll} \Theta=90^\circ & r_i^{A'} = c_j^A . & c_j^{A'} = r_i^A . \\ \Theta=180^\circ & r_i^{A'} = r_{N-i-1}^A . & c_j^{A'} = c_j^A . \\ \Theta=270^\circ & r_i^{A'} = c_{N-i-1}^A . & c_i^{A'} = r_{N-i-1}^A . \end{array}$$

### 8.2.3. Обнаружение движения на видеопоследовательностях

Системы видеонаблюдения все шире используются для решения различных практических задач, среди которых – контроль доступа в секретные объекты или в определенное время, мониторинг скоростного режима автомобилей, контроль качества в промышленности и т. д. Развитие таких систем предполагает автоматическое выделение, отслеживание и распознавание движущихся объектов в условиях действия различного рода шумов и возмущений.

Для обнаружения движущихся объектов в видеопоследовательности может использоваться алгоритм, требующий выполнения следующих шагов: формирование кадра фона, используя итеративный алгоритм; вычитание фона и получение бинарной маски движения; морфологическая обработка бинарной маски движения с целью снижения шума и получения более четкой формы движущихся объектов; локализация движущихся объектов.

Для построения фонового кадра выполняется разбиение изображения динамической сцены на два класса: элементы (пиксели) заднего плана, описывающие фон, и элементы (пиксели) переднего плана, принадлежащие движущимся объектам. Алгоритм, использующий  $N$  кадров входной видеопоследовательности, требует выполнения следующих шагов:

1. Захват  $N$  исходных базовых кадров видеоряда и формирование вектора  $w$  базовых кадров:

$$w = S^1, S^2, \dots, S^N = S^k, S^{k+L}, \dots, S^{k+(N-1)\cdot L},$$

где  $N \geq 3 \& N = 1(\text{mod } 2)$ ,

$L$  – межкадровый интервал,

$k$  – номер первого кадра из  $N$ .

2. Проверка условия: если  $l = 0(\text{mod } 2), l \in N-1, N-2, \dots, 1$ , то:

2.1. Вычисление вектора, состоящего из  $l$  бинарных масок (матриц) движения. Мaska движения  $M^k = \{m_{ij}^k\}$  определяется путем сравнения уровней цветовых составляющих красного, зеленого и синего соответствующих пикселей в соседних кадрах  $S^k = \{s_{ij}^k\}$  и  $S^{k+1}$  на основе выражения:

$$m_{ij}^k = \begin{cases} 1, & \text{если } \frac{\min(s_{ij}^k, s_{ij}^{k+1})}{\max(s_{ij}^k, s_{ij}^{k+1})} \leq T, \\ 0, & \text{в противном случае,} \end{cases}$$

где  $T$  – уровень порога.

2.2. Фильтрация бинарного изображения путем применения процедуры морфологического отмыкания.

2.3. Иначе, формируется вектор, состоящий из  $l$  вспомогательных опорных кадров:

2.3.1. Формирование вектора вспомогательных матриц движения, где в матрице  $M^k$  отображены движущиеся объекты  $S^{k+1}$  анализируемого кадра по правилу:

$$m_{ij}^k = m_{ij}^k \cdot m_{ij}^{k+1}.$$

2.3.2. Формирование вектора опорных кадров путем удаления из  $S^{k+1}$  анализируемого кадра всех пикселей, с координатами, соответствующими движущимся объектам и помещения на их место пикселей с  $S^k$  кадра с теми же координатами:

$$s_{ij}^k = \begin{cases} s_{ij}^k, & \text{if } m_{ij}^k = 1, \\ s_{ij}^{k+1}, & \text{if } m_{ij}^k = 0. \end{cases}$$

2.4. Контроль числа шагов алгоритма по правилу: если  $l=1$ , то фоновый кадр построен и осуществляется переход к пункту 3 алгоритма, иначе:  $l=l-1$  и переход к пункту 2 алгоритма.

3. Обновление опорного кадра следующим образом:

3.1. Удаление первого базового кадра из  $w$  и циклический сдвиг влево кадров в векторе на одну позицию.

3.2. Захват кадра из видеопоследовательности и добавление его в вектор  $w$  на позицию  $S^{k+(N-1)\cdot L}$ .

3.3. Переход к пункту 2 алгоритма.

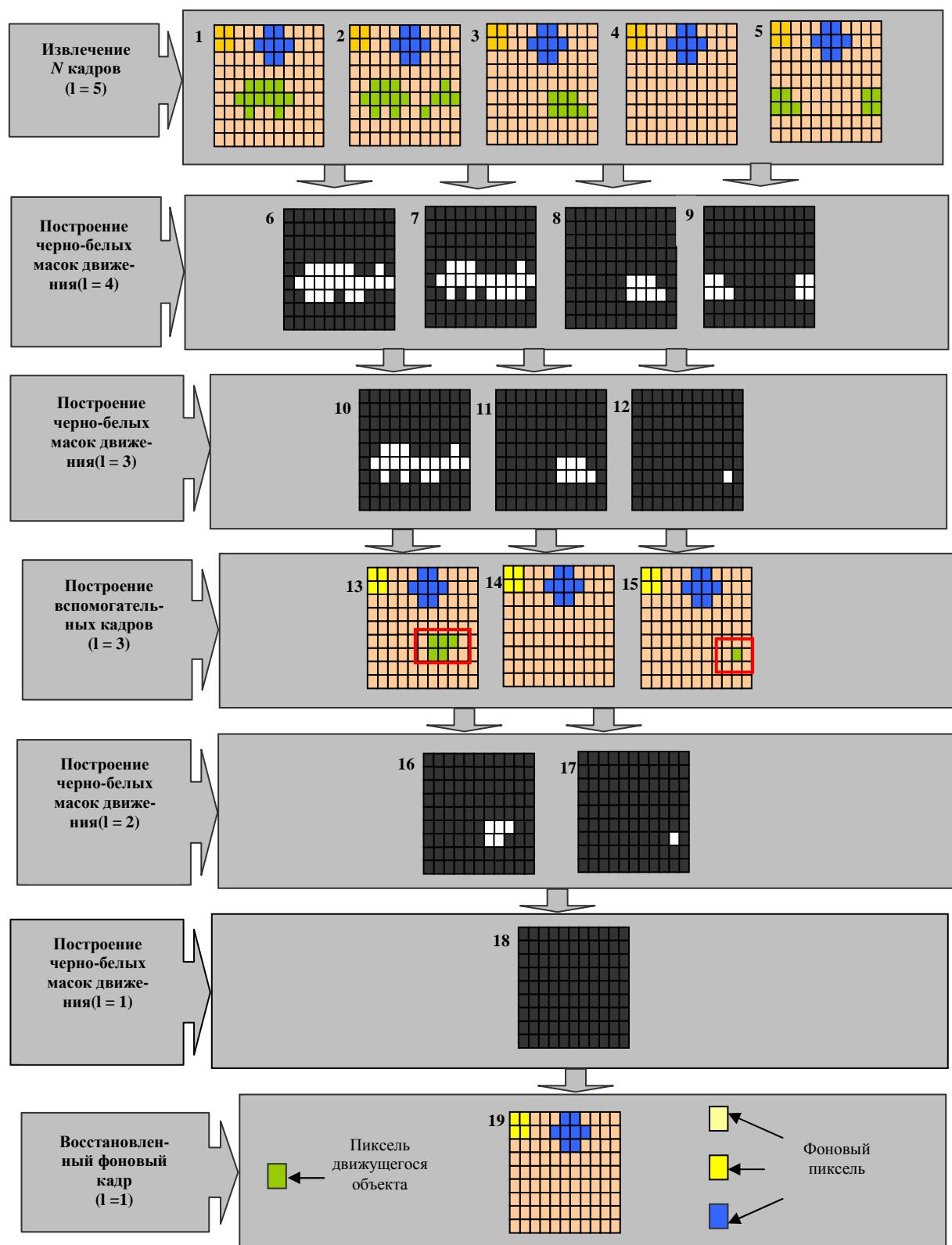


Рис. 8.6. Схема построения фонового кадра

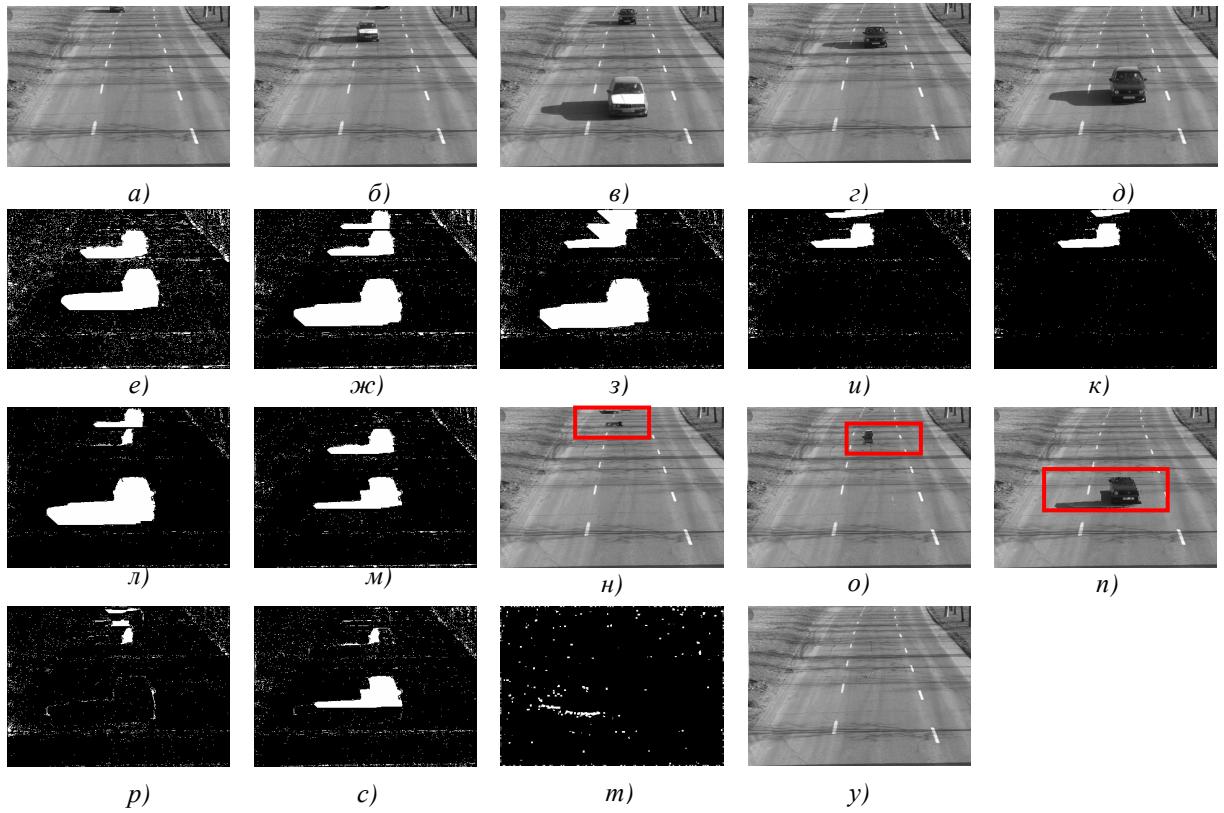


Рис. 8.7. Пример построения кадра фона для видеопоследовательности, полученной при естественном освещении ( $N = 5$  кадров,  $L = 50$  кадров,  $T = 90\%$ );  $a, b, v, \vartheta, \partial$  – 5 исходных анализируемых кадра;  $e, ж, з, и$  – бинарные маски движения при  $l = 4$ ;  $к, л, м$  – бинарные маски движения при  $l = 3$ ;  $н, о, п$  – вспомогательные фоновые кадры при  $l = 3$ ;  $p, с$  – бинарные маски движения при  $l = 2$ ;  $m$  – бинарная маска движения при  $l = 1$ ;  $y$  – полученный фоновый кадр

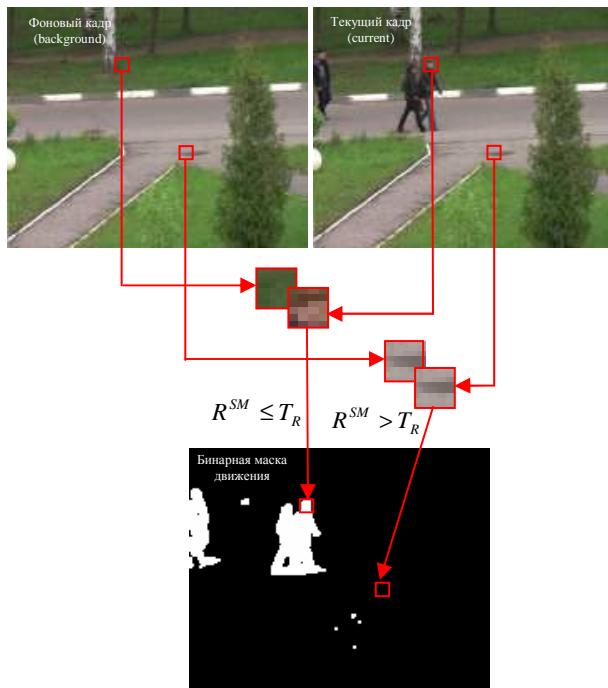


Рис. 8.8. Схема обнаружения движущихся объектов

Для обнаружения движения требуется определить функцию схожести для обрабатываемых фрагментов фонового и текущего кадров, и сравнить полученное значение с пороговой величиной, на основе чего и принимается решение о принадлежности фрагмента к движущемуся объекту.

На рис. 8.8 представлен пример обнаружения движущихся областей.

### **8.3. Вопросы и задания для самопроверки**

1. Что такое расстояние и каковы его особенности в приложении к задачам обработки изображений?
2. В чем заключается проблема сравнения видеоданных?
3. Назовите основные подходы к сравнению изображений и охарактеризуйте их.
4. Каковы основные требования, предъявляемые к мерам сходства?
5. Назовите и запишите основные функции схожести, которые Вам известны?
6. Сформулируйте алгоритм поиска объектов на изображении методом сопоставления с эталоном.
7. Назовите достоинства и недостатки алгоритма поиска объектов на изображении методом сопоставления с эталоном.
8. Какие функции схожести можно использовать для поиска объектов на изображении? Представьте их математически.
9. Как уменьшить недостатки алгоритма поиска объектов на изображении методом сопоставления с эталоном?
10. Какая из известных Вам функций схожести характеризуется минимальными вычислительными затратами? Назовите недостатки этой функции схожести.
11. Дайте определения моментам строк и столбцов изображения.
12. Запишите выражения для определения соотношений между моментами строк и столбцов эталона и повернутого изображения, которое предварительно зеркально отражено относительно горизонтальной оси?
13. Запишите выражение для определения соответствия эталона и повернутого на угол  $\Theta = 180^\circ$  ( $90^\circ$ ) изображения с использованием минимаксной аддитивной усредненной функции схожести?
14. В чем заключается сложность реализации процедур обнаружения движения?
15. Сформулируйте известный Вам алгоритм обнаружения движения на видеопоследовательностях?

## **МОДУЛЬ 9. РАСПОЗНАВАНИЕ ОБРАЗОВ**

Цель модуля – изучение студентами основных методов распознавания образов, включая технологии построения классификаторов.

В результате изучения модуля студенты *должны знать*:

- основные алгоритмы распознавания образов;
- технологии построения классификаторов для распознавания образов.

Студенты *должны уметь*:

- применять алгоритмы цифровой обработки сигналов и изображений для построения цифровых распознающих комплексов;
- а также иметь представление о нейронных сетях и алгоритмах распознавания на их основе.

Задача распознавания дискретных образов (символов, цифр, условных знаков) изображений находится в стадии разработки и еще далека от своего полного решения, что порождает появление новых алгоритмов и методов, позволяющих так или иначе разрешить возникающие в задаче проблемы.

Распознавание – это отнесение конкретного объекта (реализации), представленного значениями его свойств (признаков), к одному из фиксированного перечня образов (классов) по определенному решающему правилу в соответствии с поставленной целью [5].

Под машинным распознаванием образов понимаются методы, позволяющие относить образы к тем или иным классам – автоматически или с минимальным вмешательством человека [5].

Под образом подразумевается некоторая упорядоченная совокупность дескрипторов. Классом образов (или просто классом) называется совокупность образов, обладающих некоторыми общими свойствами.

Классом образов (или просто классом) называется совокупность образов, обладающих некоторыми общими свойствами. В практических задачах получили распространение три формы упорядоченного представления признаков: в виде векторов признаков (для количественных дескрипторов), в виде символьных строк, а также в виде деревьев (строки и деревья применяются для структурных описаний).

Таким образом, задача распознавания образов заключается в классификации некоторой группы объектов на основе определенных требований. Требования эти могут быть различными, так как в различных ситуациях возникает необходимость в определенном типе классификатора. Например, при распознавании английских букв образуется 26 классов образов. Однако, чтобы отличить при распознавании буквы от китайских иероглифов, нужны лишь два класса образов.

Проблема распознавания получила широкое распространение при решении различных задач в различных областях народного хозяйства: распознавание рукописных букв или слов, медицинская диагностика, диагностика неисправностей в технических системах, обнаружение цели, прогноз погоды, классификация сейсмических волн и др.

## **9.1. Основы распознавания образов**

### **9.1.1. Основные определения и понятия**

В общем случае, распознавание может осуществляться любой системой (живой или неживой), выполняющей следующие функции: измерение значений признаков, производство вычислений, реализующих решающее правило. При этом перечень образов, информативных признаков и решающие правила либо задаются распознающей системе извне, либо формируются самой системой. Вспомогательная функция распознающих систем – оценка риска потерь. Без этой функции невозможно построить оптимальные решающие правила, выбрать наиболее информативную систему признаков, которые используются при распознавании, и др.

Используемые обозначения:

$S \leq S < \infty$  – множество распознаваемых образов (классов), называемое иногда алфавитом;

$X$  – признаковое (выборочное) пространство;

$N$  – размерность признакового пространства (количество признаков, характеризующих распознаваемые объекты);

$D(X)$  – множество решающих правил, по которым осуществляется отнесение распознаваемого объекта (реализации) к тому или иному образу;

$R$  – риск потерь при распознавании.

Количество распознаваемых образов  $S$  всегда конечно и не может быть меньше двух. Гипотетически, конечно, можно рассматривать случай  $S = 1$ , но он является вырожденным, т. к. все реализации относят к одному и тому же образу. Для этого не нужно измерять значения каких бы то ни было признаков, решающее правило тривиально, а практический смысл решения подобного рода задачи распознавания вряд ли можно усмотреть.

Перечень образов может задаваться распознающей системе извне (учителем). Например, если система предназначена для автоматического стенографирования, то распознаваемыми образами являются фонемы – элементы устной речи.

Во многих случаях распознающая система сама формирует перечень распознаваемых образов. В литературе этот процесс называют обучением без учителя, самообучением, кластерным анализом (таксономией).

Размерность признакового пространства  $N$  обычно стремятся сделать как можно меньше, поскольку при этом сокращается количество требуемых измерений, упрощаются вычисления, формирующие и реализующие решающие правила, повышается статистическая устойчивость результатов распознавания. Вместе с тем уменьшение  $N$ , ведет к росту риска потерь. Поэтому формирование признакового пространства является компромиссной задачей, которую можно разделить на две части: формирование исходного признакового пространства и минимизация размерности этого пространства. В части, касающейся минимизации размерности, существуют формальные методы, алгоритмы и программы. Что же касается исходного пространства, то его формирование пока что основано на опыте, интуиции, а то и везении. Теоретически обоснованные подходы к решению этой задачи в литературе не встречаются.

Одной из наиболее изучаемых задач в теории распознавания образов является задача построения решающих правил. Основная цель, которая при этом преследуется, – минимизация риска потерь.

Риск потерь  $R$  фактически является критерием, по которому формируется наиболее информативное признаковое пространство и наиболее эффективные решающие правила. И алфавит, и признаки, и решающие правила должны быть такими, чтобы по возможности минимизировать риск потерь. Этот критерий (характеристика распознающей системы) является составным. В него, в общем случае, входят потери (штрафы) за ошибки распознавания и затраты на измерения признаков распознаваемых объектов. В частном, наиболее широко используемом случае в качестве риска потерь фигурирует средняя вероятность ошибки распознавания или максимальная компонента матрицы вероятностей ошибок. На практике, конечно, речь идет не о вероятностях, а об их выборочных оценках.

Итак,  $X$  можно представить как некоторое пространство размерности  $N$  с определенной в этом пространстве метрикой. Любой объект (реализация) представляется в виде точки (вектора) в этом пространстве. Проекция этой точки на  $i$ -ю ось координат соответствует значению  $i$ -го признака. Например, множество прямоугольников со сторонами, параллельными осям координат, можно представить множеством точек в двухмерном признаковом пространстве (рис. 9.1) с евклидовой метрикой, где  $x_1$  – длина горизонтальной стороны,  $x_2$  – длина вертикальной стороны. Если нужно распознавать два образа – вертикально и горизонтально вытянутые прямоугольники, то решающее правило в виде биссектрисы угла в начале координат  $D(x_1, x_2)$  эту задачу выполняет. Все точки (объекты), лежащие выше – левее  $D(x_1, x_2)$ , относятся к образу «вертикально вытянутые прямоугольники», ниже – правее – «горизонтально вытянутые прямоугольники».

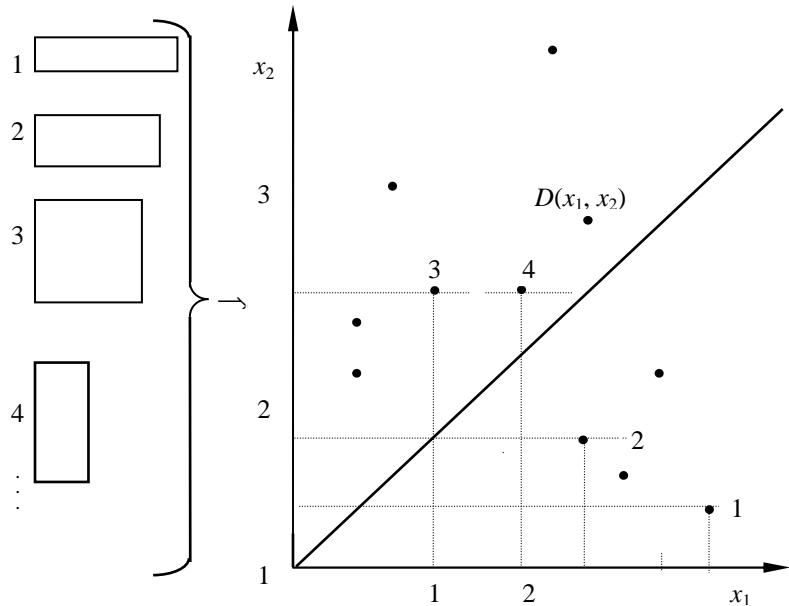


Рис. 9.1. Множество прямоугольников и их представление в признаковом пространстве

### 9.1.2. Классификация методов распознавания объектов изображений

В зависимости от формы представления обрабатываемой информации, в процессе развития области обработки изображений сложились три подхода к распознаванию дискретных знаков [1]:

- распознавание по контурному представлению;
- распознавание по скелетизированному представлению;
- распознавание по растровому представлению.

Использование контурного представления обусловлено тем, что в случае низкого уровня зашумленности границы, контур объекта содержит наиболее полную объективную информацию. Однако даже тогда, когда объекты изображений представлены в виде контуров идеальным образом, возникают определенные трудности при распознавании, связанные с тем, что объекты могут иметь достаточно сложную форму, а различные части объекта – различную значимость при определении семантики. Все это, в совокупности с различными шумами, всегда присутствующими на реальных изображениях, вызывает новые исследования в этой области, а в результате – появление множества различных алгоритмов.

По виду анализа алгоритмы распознавания объектов по контуру принято делить на три группы:

- статистические,
- структурные,
- синтаксические.

В качестве примера статистического анализа может служить оценивание характеристик, полученных при преобразовании Фурье. Структурный анализ представляет собой исследование строения и взаимного расположения частей контура. Наиболее перспективным для применения представляются алгоритмы, осуществляющие структурный и/или синтаксический анализ контуров.

Использование скелетизированного представления объектов для решения задачи распознавания обусловлено не только сокращением хранимой информации об объектах, но в существенной мере тем, что такое представление позволяет выявлять некоторые геометрические свойства, характеризующие форму образа, упрощая процедуру его описания и распознавания (однако следует учитывать, что при этом теряется часть информации об объекте, иногда полезной). Также сохраняются в явном виде различные пространственные и топологические отношения, характеризующие специфику структуры объектов. Последнее обстоятельство предопределяет использование структурно-лингвистического подхода к распознаванию дискретных объектов, основанного на использовании особенностей структуры образов объектов.

Существуют два основных способа использования этих особенностей для распознавания [1]:

1. Описание распознаваемого объекта строится на языке признаков (нескоторых отношений между непроизводными элементами (НЭ) изображения), а распознавание осуществляется путем проверки условий близости в эталонном признаковом пространстве.
2. Использование формальной грамматики как средства описания образов и процедуры грамматического разбора как средства анализа описаний с целью распознавания.

Учитывая выделение двух этих способов, принято говорить соответственно о структурных и лингвистических методах. Главным преимуществом структурных методов является простота реализации непосредственно процесса распознавания и классификации. Фактически задача сводится к геометрическому поиску в многомерном пространстве признаков некоторой области, представляющей класс образов, в которую попадает вектор признаков анализируемого объекта. Существуют различные приемы осуществления такого поиска. В некоторых случаях для проведения классификации используются так называемые разделяющие функции, представляющие собой в геометрической интерпретации некоторые поверхности в пространстве признаков, отделяющие одну область от нескольких других (или всех оставшихся) областей, представляющих классы образов.

Довольно часто применяется также последовательная дихотомия, разбивающая каждый раз оставшееся множество классов на две группы. При этом естественным является использование логических древовидных структур. В каждой вершине логического дерева производится тестирование значений некоторых логических предикатов (или решающих функций), по результатам которого выбирается для продолжения одна из двух или более ветвей, выходящих из данной вершины. Каждая ветвь ведет либо к терминальной вершине, соответствующей определенному классу образов (в этом случае процесс классификации заканчивается), либо к вершине, в которой необходимо произвести следующее тестирование.

В случае, когда признаки задаются двоичными или целыми переменными, можно обойтись без вычисления каких-либо функций, используя для классификации непосредственно значения признаков. Наиболее удобным средством для этого является сравнение с эталоном, когда значение вектора признаков, представляющего распознаваемый образ, сравнивается последовательно со значением признаков, описывающих эталонные образы. При этом для сокращения глобального перебора в пространстве эталонов часто производится предварительная классификация (в общем виде напоминающая последовательную дихотомию). В этом случае на основе анализа значений одного или нескольких признаков осуществляется последовательное разбиение всего множества эталонов на группы, и окончательная классификация осуществляется в одной из этих групп уже на основе меньшего количества признаков.

Несмотря на возможность существенного ускорения процесса распознавания за счет разработки эффективных процедур принятия решений, все же центральным и наиболее важным и сложным вопросом при использовании структурных методов является правильный выбор множества признаков. Общепризнанным является тот факт, что относительно простое решающее правило, примененное к хорошо выбранным признакам, может дать лучший результат, чем сложные и мощные методы статистического анализа при использовании признаков, выбранных менее тщательным образом.

Вопрос выбора множества признаков включает в себя три основных аспекта:

1. Наличие априорной информации о связи выбиравшего признака с тем классом объектов, для распознавания которого он будет использоваться. В этом плане выбор признаков осуществляется, как правило, на основе визуального анализа специфики представления подлежащих распознаванию образов и полностью определяется квалификацией и интуицией человека. Для распознавания скелетизированных изображений можно использовать следующие типы признаков:

- метрические (длина, ширина, ориентация, угол);
- аналитические (параллельность, прямолинейность, кривизна);
- структурные (различное чередование конструктивных графических элементов);
- топологические (вложенность, соседство, пересечение, примыкание, наложение и др.).

2. Необходимость учета информативности при выборе признака, т. е. количество информации о структуре распознаваемого образа. Никакая изобретательность при разработке системы распознавания не может компенсировать выбор бедных в смысле информативности признаков. В этом плане существует противоречие между информативностью признака и вопросом реализации его вычисления. С другой стороны, чем больше информации заложено в признаке, тем более трудоемкой является процедура его вычисления. Но с другой стороны, введение жестких ограничений на трудоемкость процедуры вычисления признака может привести к существенной потере его информативности.

3. Третий аспект состоит в том, что для каждого класса часто нет необходимости использовать все множество выбранных признаков и можно обойтись его подмножеством, что нередко приводит к существенному снижению трудоемкости вычислений.

Лингвистические методы возникли как аппарат решения задачи обработки графических изображений, не являющихся неделимыми с общечеловеческой точки зрения, но представляющих собой некоторым образом организованные сцены. При использовании лингвистических методов распознавания основными этапами их реализации являются следующие:

- выбор и выделение непроизводных элементов (НЭ);
- построение описаний объектов или сцен в терминах отношений соответственно НЭ или объектов;
- анализ описания (грамматический разбор).

Выбор НЭ зависит от характера самого объекта, вида представления исходных данных, типа решаемой прикладной задачи. При использовании векторной формы представления скелетизированных изображений естественным является использование в качестве НЭ характерных участков линий (прямолинейных и криволинейных сегментов, уголков, перегибов и т. п.), задаваемых со своими семантическими атрибутами: длиной, кривизной, толщиной и т. д. Такие формы, как правило, легко выделяются достаточно простыми аналитическими методами.

Аппаратом построения описания изображения являются формальные грамматики, которые в общем случае можно определить как четверку:

$$\langle Vn, Vt, S, R \rangle,$$

где  $V_t$  – терминальный словарь, это, как правило, набор НЭ;

$V_n$  – нетерминальный словарь, символы, обозначающие классы исходных образов или некоторых совокупностей НЭ;

$S$  – начальный символ, исходный распознаваемый образ;

$R$  – правила подстановки.

Правило из множества  $K$  в простейшем случае может означать замену одного символа цепочкой терминальных и/или нетерминальных символов; при этом отношения между элементами задаются порядком их следования в цепочке. Грамматика, использующая такого типа правила, называется цепочечной. В более сложных случаях применение правил подстановки требует определенных логических предикатов или отношений, заданных на множестве элементов описания. Такие правила реализуются обычно с помощью графовых, матричных и других более сложных грамматик.

Выбор грамматики для реализации процедуры распознавания в каждом конкретном случае зависит от специфических свойств распознаваемых изображений.

Анализ структурного описания изображения в рамках заданной грамматики осуществляется с помощью процедуры грамматического разбора. Она позволяет ответить на вопрос, является ли представленное предложение синтаксически правильным и, следовательно, принадлежит ли описываемый им образ заданному классу.

Анализ методов структурно-лингвистического подхода показывает целесообразность использования структурных методов для распознавания дискретных объектов, а лингвистических – для описания более сложных пространственно-логических отношений между объектами и распознавания определенным образом организованных сцен (т. е. различных совокупностей объектов, несущих единую семантическую нагрузку).

Распознавание по растровому представлению осуществляется с помощью так называемых корреляционных методов и заключается в том, что растровое представление объекта сравнивается с фрагментом изображения, содержащим этот объект. Данный подход является относительно простым, однако очень чувствителен к шуму, масштабу и повороту объекта, и требует значительных вычислительных затрат, что препятствует его широкому распространению.

## 9.2. Основные методы распознавания

### 9.2.1. Метод сопоставления с эталоном. Выбор признаков

Метод сопоставления с эталоном предполагает, что в памяти машины хранится некоторое множество образов по одному из каждого класса. Входной, подлежащий распознанию образ сравнивается с эталоном каждого класса на основе выбранного критерия соответствия. Если входной образ в большей степени соответствует эталону  $i$ -го класса образов, чем любому другому эталону, то входной образ классифицируется как принадлежащий  $i$ -му классу образов. В ряде случаев вместо сравнения входного образа с эталонами, классификация основывается на некотором множестве отобранных замеров, называемых информативными признаками, которые предполагаются инвариантными или малочувствительными по отношению к изменениям и искажениям. В этом случае процесс распознавания можно рассматривать как состоящий из двух задач.

Первая задача заключается в выборе подмножества признаков и их упорядочиванию в заданном множестве измерений. Вторая задача заключается в классификации (принятии решения о принадлежности входного образа тому или иному классу), которая основывается на измерениях отобранных признаков.

Упрощенная блок-схема системы распознавания приведена на рис. 9.2. [5]

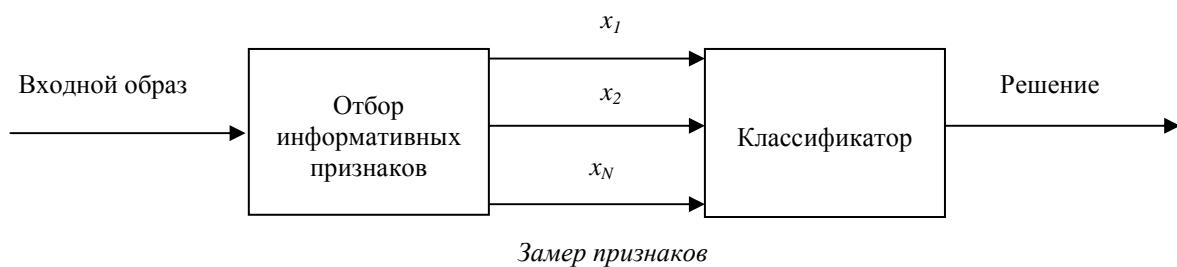


Рис. 9.2. Блок-схема системы распознавания

Таким образом, подход, заключающийся в сопоставлении с эталоном, можно рассматривать как частный случай второго подхода - подхода измерения признаков, при этом эталоны хранятся в виде измеренных признаков, а в классификаторе используется специальный критерий сопоставления. Математически задача классификации может быть сформулирована с помощью разделяющей функции.

Пусть  $\omega_1, \dots, \omega_m$  обозначают  $m$  возможных классов образов, подлежащих распознаванию и пусть  $X' = [x_1, x_2, \dots, x_n]$  вектор замеров признаков, где  $x_k$  предоставляет  $k$ -ый замер.

Тогда разделяющая функция  $D_j(x)$ , относящаяся к классу образов  $\omega_j, j = \overline{1, m}$ , такова, что если входной образ представленный вектором признаков  $X$ , принадлежит классу  $\omega_i$ , то величина  $D_i(x)$  должна быть наибольшей.

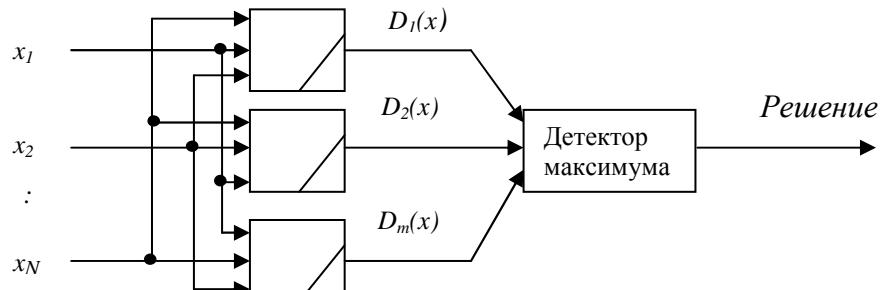
Пусть  $X \sim \omega_i$  обозначает, что вектор признаков  $X$  входного образа принадлежит классу  $\omega_i$ . Тогда можно записать, что для всех  $X \sim \omega_i$ :

$$D_i(x) > D_j(x), \quad i, j = \overline{1, m}, \quad i \neq j.$$

Таким образом, в пространстве признаков  $\Omega_x$  граница разбиений, называемая решающей границей между областями, относящимися соответственно к классам  $\omega_i, \omega_j$ , выражается уравнением:

$$D_i(x) - D_j(x) = 0.$$

Общая схема рассмотренного классификатора представлена на рис. 9.3, а типичный двумерный пример приведен на рис. 9.4.



Вычислители разделяющих функций

Рис. 9.3. Общая схема классификатора

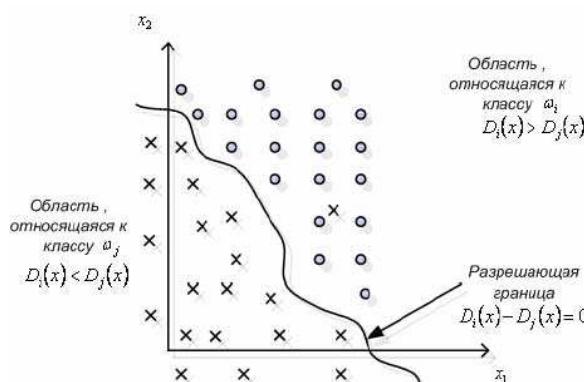


Рис. 9.4. Двумерный пример классификации

*Классификатор по минимальному расстоянию.* Важный класс составляют линейные классификаторы, в которых в качестве критерия классификации используется расстояние между входным образом и множеством опорных векторов или эталонных точек в пространстве признаков. Предположим, что задано  $m$  опорных векторов  $R_1, R_2, \dots, R_m$ , где  $R_j$  соответствует классу образов  $\omega_j$ . При классификации по минимальному расстоянию относительно  $R_1, R_2, \dots, R_m$  входной сигнал  $X$  предполагается принадлежащим  $\omega_i$ , т. е.  $X \sim \omega_i$ , если  $|X - R_i|$  – минимально, где  $|X - R_i|$  – есть расстояние между  $X$  и  $R_i$ .

Расстояние можно определить, например, следующим образом:

$$|X - R_i| = \left[ (X - R_i)^T (X - R_i) \right]^{1/2},$$

где  $T$  – обозначает операцию транспонирования вектора.

Из последнего соотношения следует, что:

$$|X - R_i|^2 = XX^T - X^T R_i - XR_i^T + R_i^T R_i.$$

Так как  $X^T X$  не зависит от  $i$ , то соответствующая разделяющая функция для классификатора по минимальному расстоянию имеет вид:

$$D_i(x) = X^T R_i + XR_i^T - R_i^T R_i, i = \overline{1, m}.$$

Как видно из этого соотношения, классификатор по минимальному расстоянию является линейной функцией. В свою очередь, свойства классификатора по минимальному расстоянию конечно зависят от того, как выбраны опорные векторы.

### 9.2.2. Структурные методы распознавания

При структурном подходе объекты описываются не множеством числовых значений признаков  $x$ , а структурой объекта. На рис. 9.5 представлено изображение и описание его иерархической структуры.

Иерархия предполагает описание сложных объектов с помощью более простых подобъектов. Те, в свою очередь, могут быть описаны с помощью подобъектов следующего уровня, и т. д. Этот подход основан на аналогии между структурой объектов и синтаксисом языков. Он приемлем тогда, когда простейшие подобъекты вычленять и распознавать легче, чем изображение (объект) в целом. Правила композиции простейших (непроизводных) элементов при описании объекта в целом называют граммати-

кой языка описания объектов. Распознавание объекта состоит в распознавании непроизводных его элементов и синтаксическом анализе (грамматическом разборе) «предложения», описывающего данный объект.

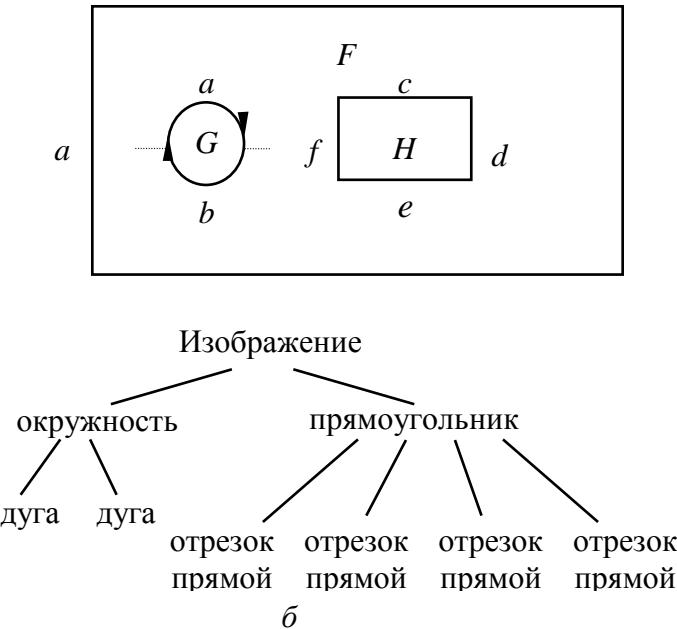


Рис. 9.5. Пояснение структурного подхода:  
а) изображение; б) иерархическое структурное описание изображения

Преимущество лингвистического подхода проявляется в том случае, если удается большое количество сложных объектов представлять с помощью небольшого множества непроизводных элементов и грамматических правил (например, распознавание устных слов по последовательности фонем). На рис. 9.6 представлен пример описания объекта (рис. 9.6, а) при помощи операции композиции «составления цепочки» из непроизводных элементов (9.6, б):  $a + a + a + b + b + c + c + c + d + d$ .

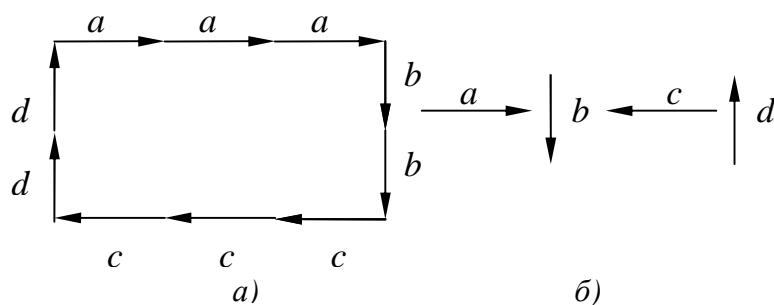


Рис. 9.6. Пример описания объекта:  
а) прямоугольник; б) непроизводные элементы

На рис. 9.7 приведен более сложный пример структурного описания цифры 9.

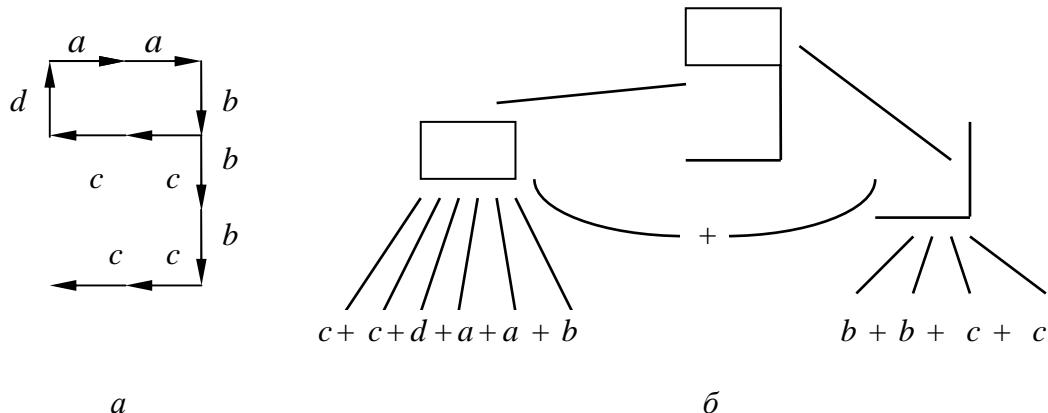


Рис. 9.7. Пример описания объекта:

а) изображение цифры 9; б) структурное описание цифры 9

Грамматика языка описания объектов формируется на этапе обучения на основе обучающей выборки. Теоретической базой данного подхода является теория формальных языков и лежащих в их основе порождающих грамматик.

### 9.2.3. Нейронные сети

Образы, принадлежащие к известным классам и используемые для оценивания упомянутых параметров, называются обучающими, а множество таких образов для каждого класса – обучающей выборкой этого класса [5]. Процесс, в ходе которого с помощью обучающей выборки строятся дискриминантные функции, называется обучением.

В реальных задачах статистические свойства классов образов зачастую неизвестны или не поддаются оценке. На практике для таких задач более эффективными оказываются методы, в которых необходимые дискриминантные функции строятся непосредственно в ходе обучения. Это устраняет необходимость использовать предположения о функциях плотности распределения вероятностей или о каких-то других вероятностных параметрах рассматриваемых классов.

Главной особенностью рассматриваемого материала является использование большого числа простейших нелинейных вычислительных элементов (называемых нейронами), которые организованы в виде сетей, напоминающих предположительный способ соединения нейронов в мозге человека. Применяемые модели известны под различными названиями, в частности, нейронные сети, нейрокомпьютеры, модели параллельной распределенной обработки, нейроморфные системы, многослойные самоан-

страивающиеся сети и модели с межсоединениями. Можно воспользоваться этими сетями в качестве среды, в которой осуществляется адаптивная настройка параметров дискриминантных функций путем последовательного предъявления обучающих выборок образов из различных классов.

Развитие нейронных сетей восходит к началу 40-х годов, чему примером является работа Мак-Каллока и Питтса [5]. Они предложили модель нейрона в виде двоичного порогового устройства, а в качестве основы для моделирования нейронных систем – стохастические алгоритмы, в которых происходят внезапные переходы нейронов из состояния 0 в состояние 1 и наоборот. В середине 50-х – начале 60-х годов Розенблаттом был создан новый класс так называемых обучающихся машин, что пробудило значительный интерес исследователей и инженеров к теории распознавания образов. Причиной большого интереса к таким машинам, названным персепtronами, было построение математических доказательств того факта, что при обучении персептрана с помощью линейно разделимых обучающих выборок (т. е. выборок образов, для которых разделяющей поверхностью может быть гиперплоскость), сходимость к решению достигается за конечное число итеративных шагов. Решение имеет вид набора коэффициентов уравнений гиперплоскостей, правильно разделяющих классы, представленные образами из обучающей выборки.

Однако, простой персептрон, как и некоторые его обобщения, были для того времени просто недостаточно мощными для решения большинства практически важных задач распознавания образов. Последующие попытки увеличить мощность машин, подобных персептрану, за счет рассмотрения нескольких слоев таких устройств в принципе выглядели привлекательными, однако им не хватало эффективных алгоритмов обучения, вроде тех, которые вызвали интерес к самому персептрану. Более поздние результаты в области разработки новых алгоритмов обучения для многослойных персепtronов, полученные Румельхартом, Хинтоном и Уильямсом, существенно изменили положение дел. Предложенный ими основной метод, который часто называют обобщенным дельта-правилом обучения посредством обратного распространения ошибки, предлагает эффективный способ обучения многослойных машин. Хотя для такого алгоритма обучения не удается доказать конечную сходимость к правильному решению, как это было сделано для однослойного персептрана, обобщенное дельта-правило успешно было применено для решения большого числа задач, представляющих практический интерес. Благодаря этим успехам многослойные машины, подобные персептрану, стали одной из главных моделей нейронных сетей, используемых в настоящее время.

*Персептрон для двух классов.* В самой простой форме, при обучении персептрана, строится линейная дискриминантная функция, осуществляющая дихотомию двух линейно разделимых обучающих выборок. Рис. 9.8 [5] схематически показывает модель персептрана в случае двух классов образов. Выходной сигнал (реакция) этого элементарного устройства базируется на взвешенной сумме его входных сигналов, имеющей вид:

$$d(x) = \sum_{i=1}^n w_i x_i + w_{n+1},$$

которая является линейной дискриминантной функцией по отношению к компонентам вектора признаков. Коэффициенты  $w_i$ ,  $i = 1, 2, \dots, n+1$ , называемые весами, изменяют (масштабируют) входные сигналы перед тем, как они суммируются и подаются на пороговое устройство. В этом смысле веса аналогичны синапсам в нервной системе человека. Функцию, которая отображает результат суммирования в конечный выходной сигнал устройства, иногда называют активирующей функцией.

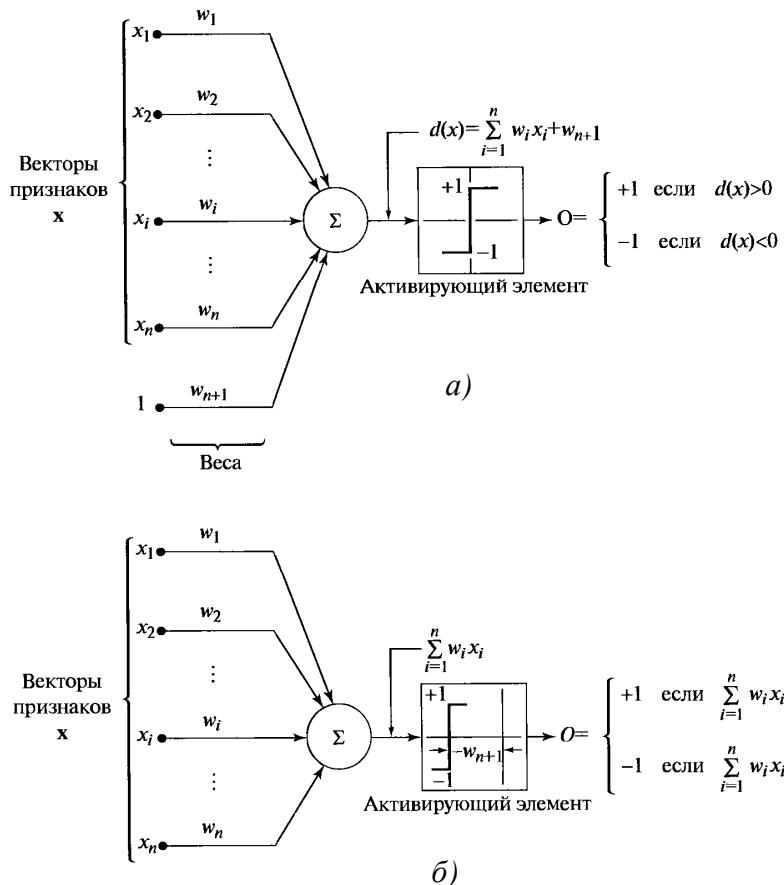


Рис. 9.8. Два эквивалентных представления модели персептрана для двух классов образов

Если  $d(x) > 0$ , пороговое устройство устанавливает на выходе персептрана сигнал +1, указывающий, что объект  $x$  опознан как принадлежащий классу  $\omega_1$ ; при  $d(x) < 0$  на выходе устанавливается сигнал -1. Если  $d(x) = 0$ , то объект  $x$  лежит на разделяющей поверхности между двумя классами, что является условием неопределенности. Уравнение разделяющей поверхности, реализуемой персептроном, получается приравниванием к нулю выражения:

$$d(x) = \sum_{i=1}^n w_i x_i + w_{n+1} = 0,$$

что представляет собой уравнение гиперплоскости в  $n$ -мерном пространстве признаков. С геометрической точки зрения, первые  $n$  коэффициентов задают направление гиперплоскости, а последний, свободный член  $w_{n+1}$ , пропорционален расстоянию от начала координат до гиперплоскости в перпендикулярном направлении. Следовательно, при  $w_{n+1} = 0$  разделяющая гиперплоскость проходит через начало координат в пространстве признаков. Аналогично, если  $w_j = 0$ , гиперплоскость проходит параллельно координатной оси  $x_j$ .

Выходной сигнал порогового устройства на рис. 9.8, *a* зависит от знака функции  $d(x)$ . Вместо того чтобы исследовать знак всей функции, можно сравнивать член с суммой в правой части уравнения со свободным членом  $w_{n+1}$ ; в этом случае выходной сигнал системы формируется в соответствии с законом:

$$O = \begin{cases} +1, & \text{если } \sum_{i=1}^n w_i x_i > -w_{n+1}, \\ -1, & \text{если } \sum_{i=1}^n w_i x_i < -w_{n+1}. \end{cases}$$

Такая реализация эквивалентна модели, изображенной на рис. 9.8, *a*, и показана на рис. 9.8, *б*. Отличие состоит в том, что пороговая функция смешается на величину  $-w_{n+1}$  и константа больше не присутствует в числе входов сумматора. При другом часто применяемом построении вектор признаков расширяется посредством добавления к нему еще одной  $(n+1)$ -ой компоненты, которая всегда равна 1, независимо от класса, к которому принадлежит объект. Иначе говоря, строится расширенный вектор, признаков объекта  $y$ , такой что  $y_i = x_i$ ,  $i = 1, 2, \dots, n$ ;  $y_{n+1} = 1$ .

Тогда выше рассмотренное уравнение примет вид:

$$d(y) = \sum_{i=1}^{n+1} w_i y_i = w^T y,$$

где  $y = (y_1, y_2, \dots, y_n, 1)^T$  – расширенный вектор признаков;  
 $w = (w_1, w_2, \dots, w_{n+1})^T$  – весовой вектор.

Такая запись обычно является более удобной с точки зрения обозначений. Однако какая бы формулировка ни использовалась, главная задача всегда состоит в нахождении вектора  $w$  по данной обучающей выборке образов каждого из двух классов.

*Алгоритмы обучения.* Рассмотрим итерационный алгоритм получения весового вектора, являющегося решением для двух линейно разделимых обучающих выборок [5]. Пусть имеются две обучающие выборки расширенных векторов признаков объектов, принадлежащих классам  $\omega_1$  и  $\omega_2$  соответственно, и пусть  $w(1)$  – начальный весовой вектор, который можно выбрать произвольно. Тогда на  $k$ -ом шаге итерации, если  $y(k) \in \omega_1$  и  $w^T(k)y(k) \leq 0$ , заменяют  $w(k)$  на

$$w(k+1) = w(k) + c y(k),$$

где  $c$  – положительный коэффициент коррекции.

Напротив, если  $y(k) \in \omega_2$  и  $w^T(k)y(k) \geq 0$ , заменяют  $w(k)$  на

$$w(k+1) = w(k) - c y(k).$$

В остальных случаях оставляют  $w(k)$  неизменным:

$$w(k+1) = w(k).$$

Данный алгоритм вносит изменения в вектор  $w$  только в тех случаях, когда рассматриваемый на  $k$ -ом шаге обработки обучающей последовательности объект классифицируется ошибочно. Корректирующий коэффициент считается положительным и в данном случае постоянным. Такой алгоритм иногда называют правилом постоянного коэффициента коррекции.

Сходимость алгоритма наступает, когда обучающие выборки обоих классов целиком проходят через машину без единой ошибки. Алгоритм с постоянным коэффициентом коррекции сходится за конечное число шагов, если две используемые обучающие выборки являются линейно разделимыми.

Рассмотрим пример алгоритма обучения персептрона. Используем две обучающие выборки, представленные на рис. 9.9, *a*, каждая из которых состоит из двух образов. Описанный алгоритм обучения должен завершиться успешно, так как эти две выборки линейно разделимы. Прежде чем применять алгоритм, необходимо выполнить расширение образов, в результате чего получают обучающие выборки  $\{(0,0,1)^T, (0,1,1)^T\}$  для класса  $\omega_1$ , и  $\{(1,0,1)^T, (1,1,1)^T\}$  для класса  $\omega_2$ . Полагая,  $c=1$   $w(1)=0$  образы предъявляются в порядке нижеприведенной последовательности шагов:

$$w^T(1)y(1) = [0, 0, 0] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0. \quad w(2) = w(1) + y(1) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

$$w^T(2)y(2) = [0, 0, 1] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1. \quad w(3) = w(2) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

$$w^T(3)y(3) = [0, 0, 1] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 1. \quad w(4) = w(3) - y(3) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$

$$w^T(4)y(4) = [-1, 0, 0] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = -1. \quad w(5) = w(4) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$

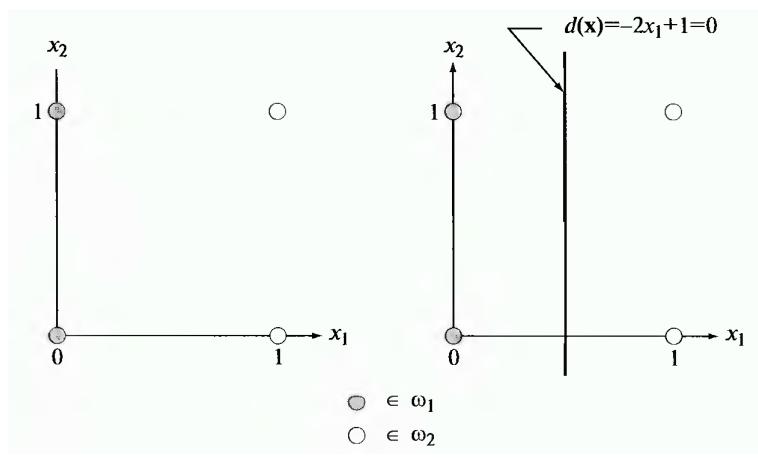


Рис. 9.9. Пример обучения персептрана: *a*) образы из двух классов; *б*) разделяющая поверхность, построенная в результате обучения

Следует отметить, что по причине ошибок классификации на первом и третьем шагах были внесены изменения в весовой вектор. Поскольку решение считается полученным только в том случае, когда ошибки классификации не возникают при предъявлении обученному алгоритму всей выборки целиком, то выборку необходимо предъявить снова. Процесс обучения продолжается вышеописанным способом, считая  $y(5) = y(1)$ ,  $y(6) = y(2)$ ,  $y(7) = y(3)$ ,  $y(8) = y(4)$  и так далее. Сходимость достигается при  $k = 14$ ; полученным решением является весовой вектор  $w(14) = (-2, 0, 1)^T$ . Соответствующая дискриминантная функция определяется уравнением  $d(y) = -2y_1 + 1$ . Полагая, что  $y_i = x_i$ , возвращаемся к исходному пространству признаков, где уравнение дискриминантной функции будет иметь вид  $d(x) = -2x_1 + 1$ . Приравнивая дискриминантную функцию к нулю, получаем уравнение разделяющей поверхности, показанной на рис. 9.9, б.

*Неразделимые классы [5].* На практике линейно разделимые классы являются скорее редким исключением, чем правилом. Поэтому в 60-х – 70-х годах значительные усилия исследователей были направлены на разработку методов, предназначенных для работы с неразделимыми классами объектов. С успехами последних продвижений в области обучения нейронных сетей, многие из таких методов для неразделимых классов стали представлять лишь исторический интерес; однако один из ранних методов имеет непосредственное отношение к теме обсуждения: это дельта-правило в его первоначальной формулировке. Данный метод обучения персепtronов, известный как метод Уидроу-Хоффа или как дельта-правило наименьшего среднего квадрата, на каждом шаге обучения минимизирует ошибку между фактической и желаемой реакциями.

Рассмотрим целевую функцию [5]:

$$J(w) = \frac{1}{2} (r - w^T y)^2,$$

где  $r$  – желаемая реакция (т. е.  $r = +1$ , если расширенный вектор признаков  $y$  объекта из обучающей выборки принадлежит классу  $\omega_1$ , и  $r = -1$ , если  $y$  принадлежит классу  $\omega_2$ ).

Задача состоит в том, чтобы путем последовательных приращений корректировать весовой вектор  $w$  в направлении, противоположном градиенту функции  $J(w)$ , чтобы найти минимум этой функции, который достигается при  $r = w^T y$ ; т. е. минимум соответствует безошибочной класси-

ификации. Если обозначить через  $w(k)$  весовой вектор на  $k$ -ом шаге итерации, то в общем виде алгоритм градиентного спуска можно записать следующим образом в виде рекуррентного соотношения:

$$w(k+1) = w(k) - \alpha \left[ \frac{\partial J(w)}{\partial w} \right]_{w=w(k)},$$

где  $w(k+1)$  – новое значение вектора  $w$ ;

параметр  $\alpha > 0$  – задает величину коррекции.

Используя выражение для целевой функции, имеем:

$$\frac{\partial J(w)}{\partial w} = -(r - w^T y)y.$$

Подстановка этого результата в рекуррентное соотношение дает:

$$w(k+1) = w(k) + \alpha [r(k) - w^T(k)y(k)]y(k),$$

причем начальный весовой вектор  $w(1)$  выбирается произвольно.

По определению, изменение («дельта») весового вектора есть векторная величина:

$$\Delta w = w(k+1) - w(k),$$

и можно записать алгоритм дельта-коррекции:

$$\Delta w = \alpha e(k)y(k),$$

где  $e(k) = r(k) - w^T(k)y(k)$  есть величина ошибки, допущенной при использовании весового вектора  $w(k)$  для распознавания объекта  $y(k)$ .

Выше представленное равенство указывает величину ошибки для весового вектора  $w(k)$ . Если заменить его на  $w(k+1)$ , оставляя объект тем же самым, ошибка станет равной:

$$e(k) = r(k) - w^T(k+1)y(k).$$

Тогда величина изменения ошибки составит:

$$\begin{aligned} \Delta e(k) &= [r(k) - w^T(k+1)y(k)] - [r(k) - w^T(k)y(k)] = \\ &= -[w^T(k+1) - w^T(k)]y(k) = -\Delta w^T y(k). \end{aligned}$$

Однако,  $\Delta w = \alpha e(k)y(k)$ , поэтому:

$$\Delta e = -\alpha e(k)y^T(k)y(k) = -\alpha e(k)\|y(k)\|^2.$$

Следовательно, при изменении весов происходит уменьшение ошибки с коэффициентом  $\alpha \|y(k)\|^2$ . С предъявлением следующего образа начнется новый цикл адаптации, в котором следующая ошибка уменьшится с коэффициентом  $\alpha \|y(k+1)\|^2$ , и т. д.

От выбора параметра  $\alpha$  зависит устойчивость и скорость сходимости алгоритма. Для устойчивости алгоритма необходимо, чтобы  $0 < \alpha < 2$ . На практике используется интервал значений  $0,1 < \alpha < 1$ . Хотя доказательство этого факта здесь не приводится, алгоритм сходится к решению, минимизирующему средний квадрат ошибки на образах обучающей выборки. Если классы линейно разделимы, то вышеописанный алгоритм не обязательно строит решение в виде разделяющей гиперплоскости. Иначе говоря, решение, минимизирующее средний квадрат ошибки, не обязательно является решением в смысле теоремы о сходимости персептрона. Такая неопределенность – цена за возможность применения алгоритма, который в данной конкретной постановке сходится и для разделимых, и для неразделимых классов.

Рассмотренные два алгоритма обучения персептрона могут быть распространены на случаи, когда имеется более двух классов, а также используются нелинейные дискриминантные функции [5].

### **9.3. Вопросы и задания для самопроверки**

1. Что такое распознавание образов (машинное распознавание)?
2. Что понимают под образом и классом образов при распознавании?
3. В каких областях используются методы распознавания объектов изображений? Приведите примеры.
4. Что такое обучение с учителем и обучение без учителя при распознавании образов?
5. Классифицируйте методы распознавания и объясните сущность каждого из них.
6. Представьте алгоритм метода сопоставления с эталоном при распознавании.
7. В чем принцип работы классификатора по минимальному расстоянию?
8. Поясните принцип структурного подхода при распознавании образов.
9. Представьте две модели персептрона для двух классов образов.
10. Поясните алгоритм обучения в нейронных сетях при распознавании.

## **ЛИТЕРАТУРА**

1. Абламейко, С. В. Обработка изображений: технология, методы применение / С. В. Абламейко, Д. М. Лагуновский. – Минск: Институт технической кибернетики НАН Беларуси, 1999. – 300 с.
2. Айфичер, Э. Цифровая обработка сигналов: практический подход / Э. Айфичер, Б. Джервис; пер. с англ. – 2-е изд. – М.: Издательский дом «Вильямс», 2004 – 992 с.
3. Ватолин, Д. Методы сжатия данных. Алгоритмы сжатия изображений / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М., 2002.
4. Глинченко, А. С. Цифровая обработка сигналов: учеб. пособие: в 2 ч. / А. С. Глинченко. – Красноярск: ИПЦ КГТУ, 2001. – 184 с.
5. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Дж. Вудс. – М.: Техносфера, 2005. – 1072 с.
6. Гонсалес, Р. Цифровая обработка изображений в среде MATLAB/ Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006.
7. Крот, А. М. Быстрые алгоритмы и программы цифровой спектральной обработки сигналов и изображений / А. М. Крот, Е. Б. Минервина. – Минск, 1995. – 407 с.
8. Лосев, В. В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки: учеб. пособие для вузов. – Минск: Выш. шк., 1990. – 132 с.
9. Методы обработки изображений. Методические указания к лабораторным работам по дисциплине «Цифровая обработка сигналов и изображений» для студентов специальности 1-40 02 01 «Вычислительные машины, системы и сети» / Сост. Р. П. Богуш. – Новополоцк: УО «ПГУ», 2008. – 40 с.
10. Новиков, Л. В. Основы вейвлет-анализа сигналов: учебное пособие. – СПб.: МОДУС+, 1999.
11. Солонина, А. И. Основы цифровой обработки сигналов: курс лекций / А. И. Солонина, Д. А. Улахович, Е. Б. Соловьева. – 2-е изд., испр. и перераб. – СПб.: БХВ-Петербург, 2005. – 768 с.
12. Прэtt, У. Цифровая обработка изображений: в 2-х кн. / У. Прэtt; пер. с англ. – М.: Мир, 1982.
13. Старовойтов, В. В. Локальные геометрические методы цифровой обработки и анализа изображений / В. В. Старовойтов. – Минск: ИТК НАН Беларуси, 1997.

14. Цифровые сигнальные процессоры семейства TMS320VC5510 [Электронный ресурс]. – 2007. – Режим доступа: [http://www.compel.ru/catalog/dsp/dsp\\_tms320/tms320c55x/tms320vc55010\\_10a?layout=print](http://www.compel.ru/catalog/dsp/dsp_tms320/tms320c55x/tms320vc55010_10a?layout=print), свободный. – Дата доступа: 12.04.2007.
15. Цифровые сигнальные процессоры TMS320VC5510A [Электронный ресурс]. – 2007. – Режим доступа: [http://sub.chipdoc.ru/html.cgi/txt/ic/Texas\\_Instruments/micros/dsp/c5000/320c55x/TMS320VC5510\\_5510A.htm](http://sub.chipdoc.ru/html.cgi/txt/ic/Texas_Instruments/micros/dsp/c5000/320c55x/TMS320VC5510_5510A.htm), свободный. – Дата доступа: 12.04.2007.
16. Ярославский, Л. П. Цифровая обработка сигналов в оптике и голографии: введение в цифровую оптику / Л. П. Ярославский. – М.: Радио и связь, 1987. – 296 с.

## СОДЕРЖАНИЕ

Введение .....	3
<b>Модуль 1. Основные понятия и определения.....</b>	<b>10</b>
1.1. Обобщенная схема цифровой обработки сигналов и основные типы сигналов.....	10
1.1.1. Обобщенная схема цифровой обработки сигналов .....	10
1.1.2. Основные типы сигналов и их математическое описание.....	14
1.2. Типовые дискретные сигналы и их математическое описание. Основная полоса частот и нормирование частоты.....	16
1.2.1. Типовые дискретные сигналы.....	16
1.2.2. Основная полоса частот. Нормирование частоты .....	19
1.3. Вопросы и задания для самопроверки .....	20
<b>Модуль 2. Дискретные преобразования и алгоритмы ЦОС .....</b>	<b>21</b>
2.1. Дискретное преобразование Фурье и его свойства .....	21
2.1.1. Основы преобразования Фурье.....	21
2.1.2. Дискретные экспоненциальные функции.....	24
2.1.3. Дискретное преобразование Фурье и его свойства .....	29
2.2. Ключевые операции ЦОС: свертка и корреляция .....	33
2.2.1. Импульсная и переходная характеристики .....	33
2.2.2. Линейная свертка .....	35
2.2.3. Циклическая свертка.....	38
2.2.4. Вычисление сверток при помощи дискретных преобразований .....	38
2.2.5. Корреляция.....	41
2.3. Алгоритмы быстрого преобразования Фурье .....	42
2.3.1. Алгоритм БПФ с прореживанием по времени .....	43
2.3.2. Алгоритм БПФ с прореживанием по частоте.....	52
2.3.3. Алгоритм действительного БПФ .....	56
2.4. Преобразование Уолша-Адамара .....	59
2.4.1. Функции Уолша.....	59
2.4.2. Преобразование Уолша-Адамара и его свойства .....	63
2.4.3. Быстрое преобразование Уолша-Адамара.....	65
2.5. Преобразование Хаара.....	69
2.6. Элементы вейвлет-преобразования сигналов .....	73
2.6.1. Основы вейвлет-преобразования.....	73
2.6.2. Матричное представление.....	76
2.7. Вопросы и задания для самопроверки .....	84
Лабораторная работа № 1 .....	87

Лабораторная работа № 2 .....	88
Лабораторная работа № 3 .....	89
<b>Модуль 3. Цифровые фильтры .....</b>	<b>90</b>
3.1. Методы математического описания и типы ЦФ .....	90
3.1.1. Основы Z-преобразования.....	90
3.1.2. Рекурсивные и нерекурсивные ЦФ .....	94
3.1.3. Передаточная функция .....	96
3.2. Структуры ЦФ .....	97
3.2.1. Основные структуры РЦФ .....	97
3.2.2. Каскадная структура .....	101
3.2.3. Параллельная структура .....	102
3.2.4. Структуры НЦФ .....	102
3.3. Частотные характеристики и элементы синтеза ЦФ .....	104
3.3.1. Общие сведения.....	104
3.3.2. Характеристики КИХ-фильтров и БИХ-фильтров .....	105
3.4. Реализация алгоритмов ЦОС на процессорах.....	109
3.4.1. Компьютерные архитектуры обработки сигналов .....	110
3.4.2. Реализация КИХ-фильтров .....	112
3.4.3. Расчет БПФ .....	115
3.4.4. Структура и описание процессора TMS320VC5510/5510A .....	117
3.5. Вопросы и задания для самопроверки .....	120
Лабораторная работа № 4 .....	121
<b>Модуль 4. Спектральный анализ сигналов .....</b>	<b>128</b>
4.1. Анализ спектра на основе ДПФ .....	128
4.1.1. Параметры анализаторов спектра.....	128
4.1.2. Базовая структура анализатора спектра на основе ДПФ и БПФ.....	129
4.1.3. Частотная характеристика анализатора спектра на основе ДПФ ....	133
4.2. Весовые функции. Связь ДПФ с цифровой фильтрацией при анализе спектра.....	137
4.2.1. Весовые функции для гармонического спектрального анализа.....	137
4.2.2. Основные параметры весовых функций при спектральном анализе .....	138
4.2.3. Элементы анализа спектра с помощью ДПФ и цифровой фильтрации ..	140
4.3. Корреляционные характеристики дискретных случайных сигналов и их статистические оценки .....	142
4.3.1. Корреляционные и спектральные характеристики дискретных случайных сигналов .....	142
4.3.2. Статистические оценки корреляционных характеристик дискретных случайных сигналов .....	145

4.3.3. Статистические оценки спектральной плотности мощности дискретных случайных сигналов .....	146
4.3.4. Коррелограммный метод вычисления спектральной плотности мощности дискретных случайных сигналов .....	147
4.3.5. Вычисление спектральной плотности мощности методом периодограмм .....	147
4.4. Вопросы и задания для самопроверки .....	150
<b>Модуль 5. Введение в цифровую обработку изображений. Обработка бинарных изображений .....</b>	<b>151</b>
5.1. Направления, развитие и области применения цифровой обработки изображений.....	151
5.1.1. Направления обработки изображений и развитие методов.....	151
5.1.2. Применение методов цифровой обработки изображений для решения прикладных задач.....	155
5.1.3. Структура системы обработки изображений .....	162
5.2. Элементы зрительного восприятия и цифрового представления .....	164
5.2.1. Строение глаза человека и формирование изображения в глазу .....	164
5.2.2. Зрительные явления .....	169
5.2.3. Считывание и регистрация изображения .....	173
5.2.4. Общий подход к математическому описанию цифрового изображения ...	175
5.3. Обработка бинарных изображений .....	177
5.3.1. Улучшение качества бинарных изображений.....	177
5.3.2. Уточнение бинарных изображений.....	179
5.3.3. Выделение контуров бинарных изображений .....	181
5.3.4. Связность.....	182
5.4. Вопросы и задания для самопроверки .....	186
<b>Модуль 6. Методы пространственной обработки полутоновых изображений .....</b>	<b>188</b>
6.1. Градационные преобразования.....	188
6.1.1. Представление и особенности обработки.....	188
6.1.2. Бинаризация .....	189
6.1.3. Преобразование изображения в негатив.....	192
6.1.4. Логарифмическое преобразование .....	192
6.1.5. Степенные преобразования.....	193
6.1.6. Кусочно-линейные функции преобразования.....	194
6.1.7. Соляризация изображения.....	196
6.2. Фильтрация изображений.....	197
6.2.1. Принципы и особенности пространственной фильтрации.....	197

6.2.2. Низкочастотная фильтрация .....	199
6.2.3. Подчеркивание границ.....	202
6.3. Улучшение изображений на основе арифметико-логических операций и изменения гистограмм .....	204
6.3.1. Улучшение на основе арифметико-логических операций.....	204
6.3.2. Глобальные методы улучшения контраста .....	206
6.3.3. Метод локального улучшения контраста .....	210
6.4. Сегментация изображений посредством выделения линий и контуров...	212
6.4.1. Общие принципы .....	212
6.4.2. Линейные методы контрастирования .....	214
6.4.3. Нелинейные методы контрастирования .....	218
6.5. Связывание контуров. Представление и описание изображений .....	223
6.5.1. Локальная обработка.....	223
6.5.2. Глобальный анализ с помощью преобразования Хафа.....	225
6.5.3. Представление и описание .....	229
6.6. Обработка изображений с помощью математической морфологии...	236
6.6.1. Обработка бинарных изображений .....	236
6.6.2. Обработка полутооновых изображений .....	239
6.7. Вопросы и задания для самопроверки .....	242
<b>Модуль 7. Обработка и сжатие изображений с использованием дискретных преобразований .....</b>	<b>245</b>
7.1. Двумерное преобразование Фурье. Фильтрация изображений в частотной области .....	245
7.1.1. Дискретное двумерное преобразование Фурье.....	245
7.1.2. Основы фильтрации в частотной области .....	250
7.2. Сжатие изображений. Спектральные методы сжатия.....	254
7.2.1. Особенности изображений, учитываемые при сжатии .....	254
7.2.2. Требования к алгоритмам компрессии .....	254
7.2.3. Критерии сравнения алгоритмов .....	256
7.2.4. Основы спектральных методов сжатия .....	258
7.2.5. Дискретное косинусное преобразование .....	259
7.3. Стандарты сжатия JPEG и JPEG 2000.....	262
7.3.1. Основные шаги стандарта сжатия JPEG.....	262
7.3.2. Особенности стандарта сжатия JPEG 2000 .....	270
7.4. Вопросы и задания для самопроверки .....	271
<b>Модуль 8. Сравнение цифровых изображений и обнаружение объектов на изображениях .....</b>	<b>273</b>
8.1. Сравнение изображений .....	273

8.1.1. Метрические характеристики .....	273
8.1.2. Проблема сравнения видеоданных.....	274
8.1.3. Требования к мерам, вычисляющим сходство.....	276
8.1.4. Функции схожести корреляционного типа .....	278
8.1.5. Алгоритм поиска объектов.....	280
8.2. Поиск объектов на изображениях .....	283
8.2.1. Методы сокращения вычислительных затрат при поиске объектов на статических изображениях.....	283
8.2.2. Обнаружение повернутых объектов .....	284
8.2.3. Обнаружение движения на видеопоследовательностях .....	287
8.3. Вопросы и задания для самопроверки .....	291
<b>Модуль 9. Распознавание образов.....</b>	<b>292</b>
9.1. Основы распознавания.....	293
9.1.1. Основные определения и понятия.....	293
9.1.2. Классификация методов распознавания объектов изображений....	295
9.2. Основные методы распознавания.....	300
9.2.1. Метод сопоставления с эталоном. Выбор признаков.....	300
9.2.2. Структурные методы распознавания .....	302
9.2.3. Нейронные сети.....	304
9.3. Вопросы и задания для самопроверки .....	312
Литература .....	313

*Учебное издание*

БОГУШ Рихард Петрович

## ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Учебно-методический комплекс  
для студентов специальности 1-40 02 01  
«Вычислительные машины, системы и сети»

*Редактор Н. В. Байко*

*Дизайн обложки В. А. Виноградовой*

---

Подписано в печать 23.04.09. Формат 60×84 1/16. Гарнитура Таймс. Бумага офсетная.  
Ризография. Усл.-печ. л. 18,56. Уч.-изд. л. 17,93. Тираж 60. Заказ № 662.

---

Издатель и полиграфическое исполнение:  
учреждение образования «Полоцкий государственный университет»

ЛИ 02330/0133020 от 30.04.04      ЛП № 02330/0133128 от 27.05.04

211440 г. Новополоцк, ул. Блохина, 29