

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Ярославский государственный университет
им. П. Г. Демидова

А. Л. ПРИОРОВ, В. В. ХРЯЩЕВ

ОБРАБОТКА И ПЕРЕДАЧА МУЛЬТИМЕДИЙНОЙ ИНФОРМАЦИИ

Учебное пособие

*Рекомендовано
Научно-методическим советом университета для студентов,
обучающихся по направлению Телекоммуникации*

Ярославль 2010

УДК 621.397
ББК 3973.235 я 73
П 76

Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2010 года

Рецензенты:

Меньшиков Б. Н., кандидат технических наук;
кафедра радиотехники и радиосистем Владимирского государственного
университета

Приоров, А. Л. Обработка и передача мультимедийной
П 76 информации: учебное пособие / А. Л. Приоров, В. В. Хрящев;
Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль: ЯрГУ, 2010. – 188
с.

ISBN 978-5-8397-0699-6

Описаны современные стандарты сжатия речевых и аудиосигналов, цифровых изображений и видеопоследовательностей. Учебное пособие предназначено для студентов направления 210400.62 Телекоммуникации, изучающих дисциплину «Обработка и передача мультимедийной информации», специальностей 010801.65 Радиофизика и электроника и 210302.65 Радиотехника физического факультета ЯрГУ. Материал может быть использован при подготовке студентами курсовых и дипломных проектов.

Ил. 58. Табл. 15. Библиогр.: 36 назв.

ISBN 978-5-8397-0699-6 © Ярославский государственный
университет
им. П. Г. Демидова, 2010

ВВЕДЕНИЕ

Проводимые во всех технически развитых странах разработки алгоритмов и аппаратуры сокращения объема и рационального пакетирования видео-, аудио- и сопутствующей информации являются основой создания систем эффективного использования каналов связи, сохранения действующих частотных планов, высвобождения значительной части частотного ресурса для передачи потребителям дополнительных видов услуг – видеотелефона, мобильной и стационарной телеконференц-связи, многопрограммного интерактивного телевидения, телевидения высокой четкости, многопрограммного звукового вещания, цифрового кино [1–6]. Разработка алгоритмов и соответствующей аппаратуры цифрового сжатия различных видов информации для их передачи по каналам связи как альтернативы аналоговым системам проводится уже более 20 лет. Был получен ряд важных результатов в плане разработки алгоритмов сжатия (включая стандарты JPEG, JPEG-2000, JPEG XR, MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264/AVC для статических и динамических изображений различного разрешения) [7–13].

Проблема создания высококачественной системы цифровой передачи видеоинформации – это главным образом проблема быстрого и эффективного ее сжатия – наиболее ресурсопотребляющей части всей системы. Ее эффективность существенно влияет на качество воспроизводимой информации. Кодировующее устройство решает сложные задачи в реальном времени и производит большое количество операций обработки, определяющих качество цифрового преобразования сигналов изображения и звукового сопровождения. Декодер, как правило, менее сложен и должен получать и восстанавливать данные [14–19].

Особенную важность подобные исследования приобрели в связи с переходом к цифровому телерадиовещанию, и в частности к многопрограммному цифровому телевидению, вызванным присоединением России к общеевропейской системе DVB (Распоряжение Правительства Российской Федерации от 25.05.2004 № 706-р «О внедрении в Российской Федерации европейской системы цифрового телевизионного вещания DVB»). Практически во всех странах Европы определены сроки полного перехода на цифровое вещание (в пределе до 2012 г.). Обоснование и сроки внедрения цифровых систем вещания в России приводятся в проекте Концепции федеральной целевой программы «Развитие телерадиовещания в Российской Федерации (2007–2015 гг.)» [4–6].

В этой связи подготовка специалистов в области обработки и передачи мультимедийной информации является весьма актуальной и перспективной задачей [20–36].

Учебное пособие состоит из девяти разделов и охватывает широкий круг вопросов, связанных с кодированием, сжатием, оценкой качества цифровых изображений, видеопоследовательностей, речевых и аудиосигналов.

В первых трех разделах описаны различные методы сжатия изображений. Приводится классификация основных методов сжатия, описываются основные алгоритмы сжатия без потерь информации, подробно рассматриваются алгоритмы кодирования с преобразованием. Описаны стандарты сжатия цветных изображений – алгоритмы JPEG и JPEG2000.

В четвертом разделе рассматриваются вопросы оценки качества сжатых изображений. Приводятся классификация критериев качества, описание ряда современных объективных метрик (универсальный индекс качества, коэффициент структурного подобия, модифицированное пиковое отношение сигнал/шум), результаты сравнения алгоритмов сжатия. Раздел написан на основе анализа современных публикаций российских и зарубежных ученых, а также оригинальных результатов, полученных на кафедре динамики электронных систем Ярославского государственного университета имени П. Г. Демидова (ЯрГУ). Алгоритмы оценки качества, приведенные в данном разделе, реализованы в программной среде PicLab (www.piclab.ru) и активно используются студентами при выполнении лабораторных работ, в рамках курсового и дипломного моделирования.

Пятый и шестой разделы посвящены основным сведениям о видеосигналах и основам видеокодирования. Описываются вопросы пространственной и временной дискретизации, представление видеопоследовательностей в различных цветовых пространствах, общая схема видеокодека, алгоритмы оценки и компенсации движения.

В седьмом разделе кратко описаны стандарты сжатия видеоданных – MPEG-4 и H.264/AVC. Отдельное внимание уделяется вопросам разработки, принятия и использования международных стандартов. Рассмотрены ключевые алгоритмы каждого из указанных стандартов и их специфические особенности. Проведено краткое описание стандартов сжатия предыдущего поколения – MPEG-1, MPEG-2, H.261 и H.263.

В восьмом разделе уделяется внимание вопросам кодирования и оценки качества (разборчивости) речевых сигналов. Проводится

классификация основных методов сжатия речевой информации на основе известных рекомендаций международного союза электросвязи.

В последнем, девятом разделе пособия дается краткое описание современных стандартов сжатия аудиосигналов. Отдельное внимание уделено популярному формату хранения и передачи аудиоданных – алгоритму MP3.

В приложениях приводятся стандарты кодирования мультимедиа по классификации международной организации по сертификации (ISO), типовые тестовые изображения в градациях серого, современные стандарты сжатия речевых сигналов.

В основу учебного пособия положены материалы лекций, в течение ряда лет читавшихся студентам физического факультета ЯрГУ, обучающимся по направлению «Телекоммуникации», а также некоторые результаты работы авторов и их коллег в данном научном направлении. Надеемся, что пособие будет полезно не только студентам и аспирантам телекоммуникационного и радиотехнического профилей, но также инженерам и научным работникам, специалистам в области информатики и программистам, желающим повысить свой уровень в этой интересной и перспективной научной области.

Выражаем глубокую благодарность нашему научному руководителю профессору Ю. А. Брюханову за помощь при создании курса «Обработка и передача мультимедийной информации» и данного пособия. Мы благодарны также члену-корреспонденту РАН Ю. Б. Зубареву, докторам технических наук В. П. Дворковичу, А. А. Ланнэ, В. В. Витязеву, М. А. Щербакову, А. И. Галушкину, А. К. Бернюкову, А. В. Дворковичу, В. И. Джигану, М. К. Чобану, кандидатам технических наук К. Н. Быструшкину, И. В. Апалькову, Е. Ю. Саутову, которые оказали значительное влияние на формирование взглядов авторов в указанном научном направлении. Отдельные слова благодарности сотрудникам, аспирантам и студентам кафедры динамики электронных систем ЯрГУ Ю. Лукашевичу, М. Голубеву, В. Бекреневу, В. Волохову, Л. Шмаглицу, Ф. Черногорову, Г. Коченятову, Н. Павловой, Т. Тихоновой за активное участие в создании оригинал-макета пособия.

Хочется отметить также помощь сотрудников редакционно-издательского отдела ЯрГУ во главе с Л. Н. Селивановой, которые своими замечаниями способствовали повышению стилистического уровня изложения материала.

1. АЛГОРИТМЫ СЖАТИЯ ИЗОБРАЖЕНИЙ

Алгоритмы сжатия занимают значительное место в теории цифровой обработки изображений. Связано это с тем, что изображения, представленные в цифровой форме, требуют для хранения довольно большого объема памяти, а при передаче их по каналам связи требуется значительное время. Так, для изображения с разрешением 1280×1024 точек (типовое разрешение 19-дюймового монитора), с 24 битами на отсчет, потребуется $1280 \times 1024 \times 24 = 3,75$ Мбайт памяти. Чтобы передать его по каналу со скоростью 512 Кбит/с, потребуется около минуты.

В то же время изображения содержат информацию, которая является избыточной, т. е. части памяти и полосы частот канала тратятся на нее. Алгоритмы сжатия уменьшают объем избыточной информации. Их можно разделить на два вида:

- методы сжатия без потерь (рис. 1.1а);
- методы сжатия с потерями (рис. 1.1б).

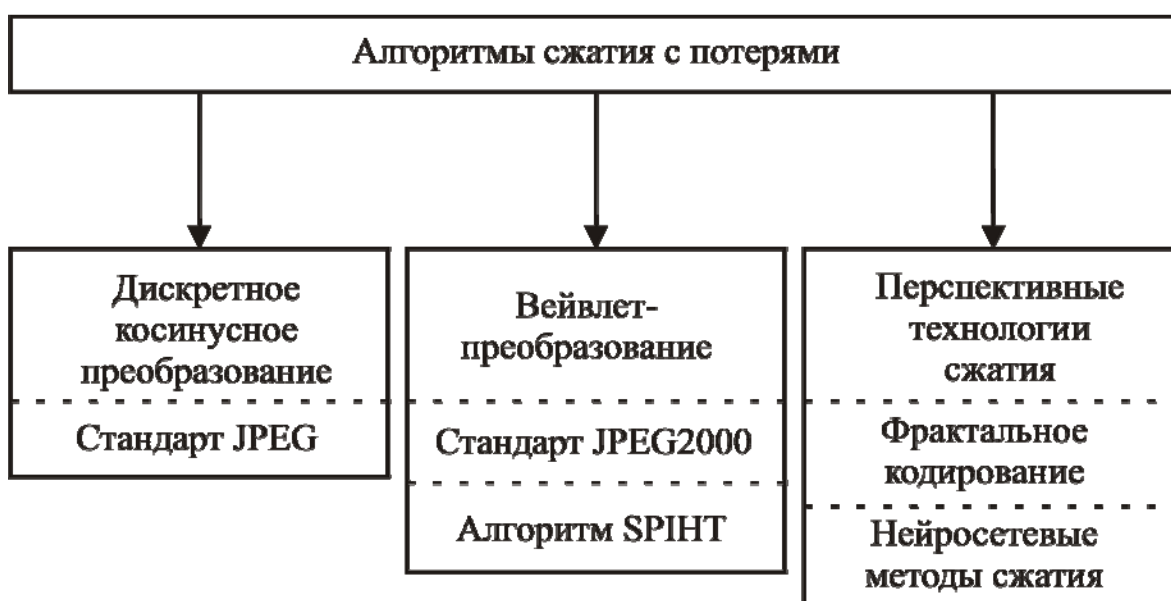
Алгоритмы сжатия без потерь основаны на исключении избыточной статистической информации. К ним относятся: кодирование длин серий (RLE), метод группового кодирования (LZW), хатфмановское кодирование, арифметическое кодирование и др. Сжатие без потерь означает, что процесс обратим, т. е. информацию можно восстановить при декодировании в первоначальном виде.

Методы сжатия с потерями основаны на том, что в изображении содержится информация, слабо воспринимаемая человеческим глазом. Такой вид избыточности называется психофизиологической избыточностью. Снижение психофизиологической избыточности состоит в исключении несущественных частей изображения, которые человеческий глаз не воспринимает в силу психофизиологических особенностей. Критериями исключения такой информации являются параметры рецепторов зрительного анализатора получателя видео-сообщений. При этом часть информации безвозвратно теряется. Восстановить ее при декодировании невозможно.

Например, глаз человека не различает цвет мелких деталей изображения. Следовательно, можно хранить только яркость деталей, а цветность отбросить и таким образом уменьшить объем информации. Этот метод давно применяется в телевидении. Разбиение непрерывного потока изображения на кадры также пример исключения психофизиологической избыточности.



а)



б)

Рис. 1.1. Методы сжатия визуальной информации:
а) без потерь; б) с потерями

В следующих разделах этой главы основные алгоритмы сжатия, применяемые в цифровой обработке изображений, будут рассмотрены более подробно.

1.1. Алгоритмы сжатия без потерь

1.1.1. Алгоритм кодирования длин серий (RLE)

Групповое кодирование – от английского Run Length Encoding (RLE) – один из самых старых и самых простых алгоритмов архивации графики. Изображение в нем вытягивается в ряд по строкам раstra. Сжатие в RLE происходит за счет того, что в исходном изображении встречаются цепочки одинаковых значений отсчетов. Такие цепочки отсчетов заменяются на два числа: первое

число – количество элементов в цепочке (счетчик), второе – значение элемента. Например, если в изображении, которое использует 8 бит на отсчет, встречается цепочка 255, 255, 255, 255, то она заменяется на 4, 255. Такая замена позволяет уменьшить статистическую избыточность изображения.

Следует отметить, что алгоритм RLE рассчитан на искусственные несложные изображения, т. к. в изображениях, где протяженные цепочки встречаются достаточно редко и/или они короткие, данный алгоритм может, напротив, увеличить объем информации вследствие того, что цепочки из одного элемента будут заменяться на два – счетчик и значение.

1.1.2. Алгоритм LZW

Название алгоритм получил по первым буквам фамилий его разработчиков – Lempel, Ziv и Welch. Изображение в нем, как и в алгоритме RLE, вытягивается в ряд по строкам раstra, но сжатие, в отличие от RLE, осуществляется уже не за счет цепочек одинаковых байт, а за счет одинаковых цепочек байт.

Например, если в изображении в разных местах встречается цепочка элементов 255, 120, 0, 95, то ее можно сохранить в некоторой таблице-книге, и во всем изображении заменить эту цепочку на ее индекс в таблице.

Существует множество разновидностей алгоритмов LZW. Рассматриваемый ниже вариант алгоритма будет использовать дерево для представления и хранения цепочек. Это достаточно сильное ограничение на вид цепочек, и далеко не все одинаковые подцепочки в изображении будут использованы при сжатии. Однако в предлагаемом алгоритме выгодно сжимать даже цепочки, состоящие из 2 байт.

Процесс сжатия выглядит достаточно просто. Последовательно считываем отсчеты входного изображения. На основе предыдущих и текущего отсчетов формируем строку и проверяем, есть ли в созданной таблице строк такая строка. Если строка есть, то считываем следующий символ, а если строки нет, то заносим в сжатое изображение код для предыдущей найденной строки, заносим строку в таблицу и начинаем поиск снова.

Рассмотрим подробнее алгоритм компрессии на примере. Пусть исходное изображение {45, 55, 55, 151, 55, 55, 55} содержит 8 бит/отсчет. Создадим таблицу символов, которую заполним всевозможными строками, состоящими из одного элемента. Кроме того, нам потребуется код очистки и код конца информации: 256 и

257 соответственно. Первоначально таблица будет выглядеть следующим образом.

Таблица 1.1

Первоначальная таблица

| Индекс | Значение |
|--------|----------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| | |
| 255 | 255 |
| 256 | 256 |
| 257 | 257 |

В рассматриваемом варианте алгоритма используется 9-битный код и соответственно под коды для строк нам остаются значения от 258 до 511. Добавляемые строки записываются в таблицу последовательно, при этом индекс строки в таблице становится ее кодом.

Согласно алгоритму поместим в выходной файл сначала код очистки – 256, потом добавим к изначально пустой строке «45» и проверим, есть ли строка «45» в таблице. Поскольку мы при инициализации занесли в таблицу все строки из одного символа, то строка «45» есть в таблице. Далее мы читаем следующий отсчет «55» из входного изображения и проверяем, есть ли строка «45, 55» в таблице. Такой строки в таблице пока нет. Мы заносим в таблицу строку «45, 55» (с первым свободным кодом 258) и записываем в поток код 45. Можно коротко представить архивацию так:

«45» – есть в таблице;

«45, 55» – нет. Добавляем в таблицу 258 «45, 55».

В выходной файл – 45;

«55, 55» – нет. В таблицу: 259 «55, 55». В файл – 55;

«55, 151» – нет. В таблицу: 260 «55, 151». В файл – 55;

«151, 55» – нет. В таблицу: 261 «151, 55». В файл – 151;

«55, 55» – есть в таблице;

«55, 55, 55» – нет. В таблицу: 262 «55, 55, 55». В файл – 259.

Последовательность кодов для данного примера, попадающих в выходной поток: {256, 45, 55, 55, 151, 259}.

Таблица, полученная в результате, будет выглядеть так.

Конечная таблица

| Индекс | Значение |
|--------|----------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| | |
| 255 | 255 |
| 256 | 256 |
| 257 | 257 |
| 258 | 45,55 |
| 259 | 55,55 |
| 260 | 55,151 |
| 261 | 151,55 |
| 262 | 55,55,55 |

Особенность LZW заключается в том, что для декомпрессии нам не надо сохранять таблицу строк в файл для распаковки. Алгоритм построен таким образом, что можно восстановить таблицу строк, пользуясь только потоком кодов.

Для каждого кода надо добавлять в таблицу строку, состоящую из уже присутствующей там строки и символа, с которого начинается следующая строка в потоке. Алгоритм декомпрессии осуществляет эту операцию.

Таким образом, на выходе кодера получается $6 \times 9 = 54$ бита. Входное изображение содержало $7 \times 8 = 56$ бит информации.

В отличие от RLE ситуация, когда алгоритм LZW увеличивает изображение, встречается крайне редко. LZW универсален, именно его варианты используются в традиционных файловых архиваторах, например в WinRAR.

1.1.3. Алгоритм Хаффмана

Одним из классических алгоритмов неравномерного кодирования, известных с 1960-х годов, является алгоритм Хаффмана. Работа алгоритма основана на том факте, что символы в сообщении появляются с различной вероятностью. Как следствие, применение кодов различной длины для символов позволяет снизить общий кодовый поток, т. е. сжать данные.

Для алгоритмов, использующих коды различной длины, можно ввести среднюю длину l_{cp} кода, определяемую как математическое ожидание длины элементарного кода:

$$l_{cp} = \sum_{i=1}^r p_i l_i, \quad l_i = l(B_i) - \text{длина кода для } i\text{-го символа.}$$

Коды с $l_{cp} = l_{\min}$ называются кодами с минимальной кодовой избыточностью, или кодами Хаффмана.

При этом средняя длина слов l_{cp} находится в диапазоне

$$\begin{cases} H(A) \leq l_{cp} \leq H(A) + 1 \\ l_{cp} \geq 1 \end{cases} \quad \text{бит/пиксел, где } H(A) - \text{энтропия.}$$

Средняя длина слов не более чем на 1 бит/пиксел больше энтропии, но не менее 1 бит/пиксел (когда энтропия равна нулю).

Коды с минимальной избыточностью дают в среднем минимальное увеличение длин слов при соответствующем кодировании. Рассматриваемый метод позволяет получить подобные коды.

Алгоритм Хаффмана основан на задании вероятности каждого символа, присутствующего в сообщении. Зная эти вероятности, можно построить таблицу кодов, обладающую следующими свойствами:

- различные коды имеют различное количество бит;
- коды символов, обладающих меньшей вероятностью, имеют больше бит, чем коды символов с большей вероятностью;
- коды могут быть декодированы единственным образом.

Процесс кодирования Хаффмана иллюстрируется на рис. 1.2.

Алгоритм кодирования включает в себя следующие шаги.

Шаг 1. Расставляем все буквы входного алфавита в порядке убывания вероятности. Считаем все соответствующие слова B_i из алфавита $\Omega = \{0, 1\}$ пустыми.

Шаг 2. Объединяем два символа a_i и a_j с наименьшими вероятностями p_i и p_j в псевдосимвол $a' \{a_i a_j\}$ с вероятностью $p_i + p_j$. Дописываем 0 в начало слова B_i и 1 в начало слова B_j .

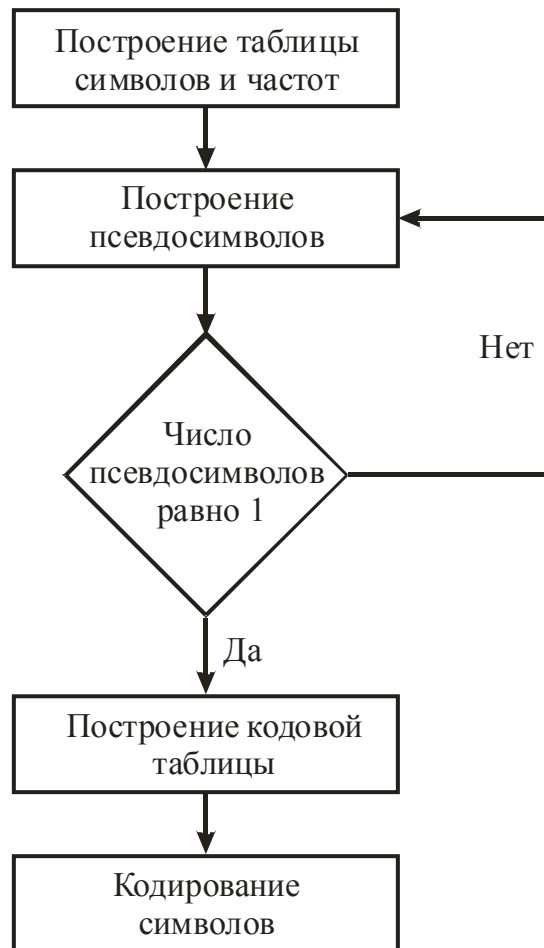


Рис. 1.2. Схема кодирования Хаффмана

Шаг 3. Удаляем из списка упорядоченных символов a_i и a_j , заносим туда псевдосимвол $a' \{a_i a_j\}$. Проводим шаг 2, добавляя при необходимости «1» или «0» для всех слов B_i , соответствующих псевдосимволам, до тех пор, пока в списке не останется 1 псевдосимвол.

Пример. Пусть для кодирования изображения используется 8 уровней квантования, распределение которых определяется гистограммой со следующими данными:

$$P(a_0) = P(a_5) = P(a_6) = P(a_7) = 0,06$$

$$P(a_1) = 0,23$$

$$P(a_2) = 0,3$$

$$P(a_3) = 0,15$$

$$P(a_4) = 0,08$$

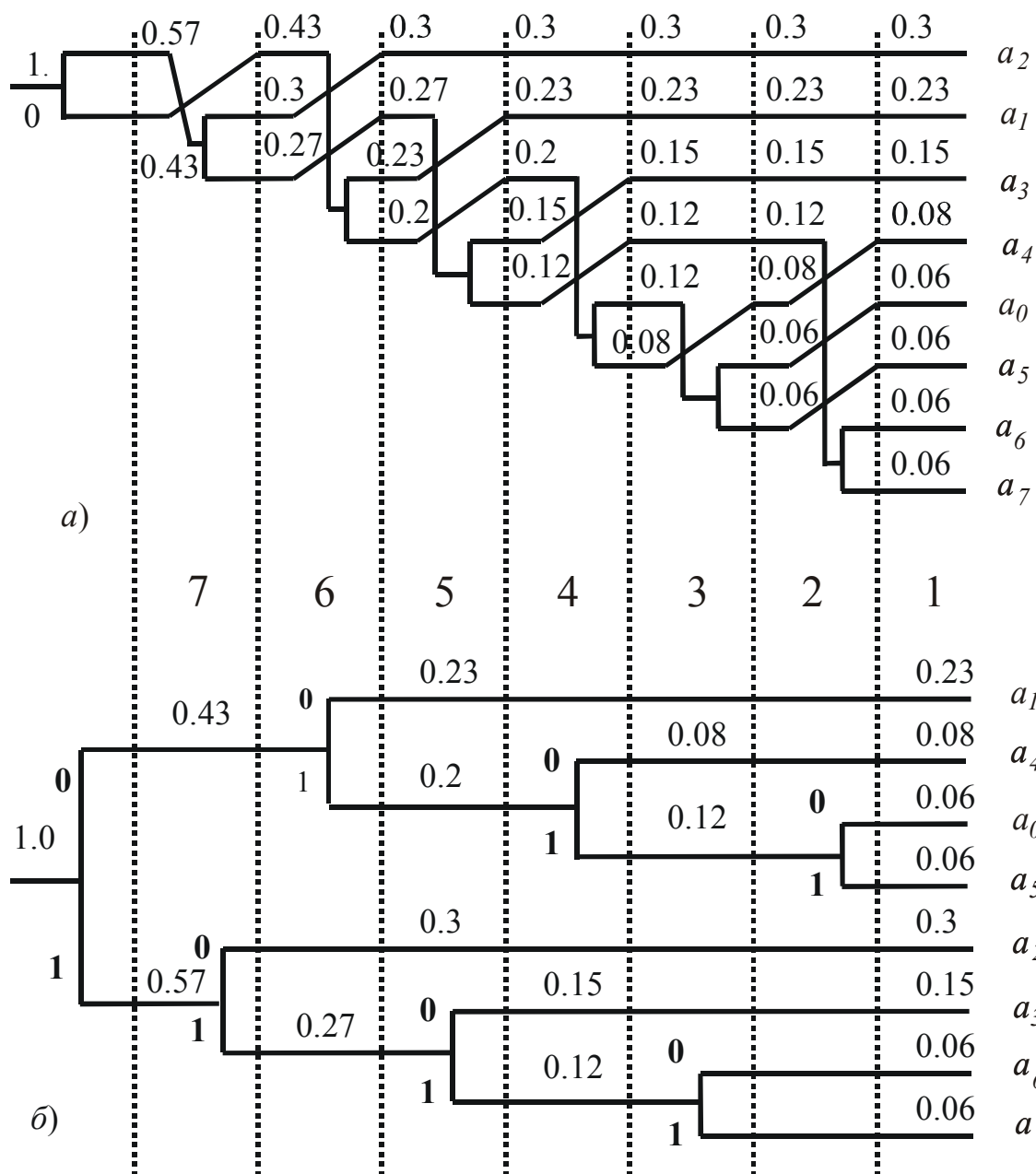


Рис. 1.3. Принцип построения алгоритма хатфмановского кодирования

Дерево строится справа налево следующим образом (рис. 1.3а):

- в секции 1 уровни пикселей сортируются по вероятности от наибольшей к наименьшей сверху вниз; при равенстве $P(a_i) = P(a_j)$ выше ставится уровень $a_i < a_j$;

- в секции 2 две самые нижние ветви объединяются в узел, их вероятности складываются, и узел образует новую ветвь; общее количество ветвей уменьшается на одну, и они вновь сортируются по вероятности от наибольшей к наименьшей;

– в секциях 3 и 4 и т. д. производятся операции, аналогичные проводимым в секции 2, до тех пор, пока не останется одна ветвь с вероятностью, равной 1.

Все это дерево можно перестроить, убрав пересечения (рис. 1.3б).

Кодирование осуществляется движением слева направо по дереву к каждому кодируемому уровню a_i . При этом на каждом узле коду присваивается, например, двоичный «0», если осуществляется шаг вверх, и «1», если осуществляется шаг вниз.

Таким образом, для данного случая наиболее вероятные значения a_1 и a_2 кодируются двухбитовым кодом (00 и 10 соответственно), величины a_3 и a_4 – трехбитовым кодом (110 и 010 соответственно), а наиболее редкие значения a_0, a_5, a_6, a_7 – четырехбитовым кодом (0110, 0111, 1110 и 1111 соответственно).

Нетрудно понять, что эти коды легко различимы.

Если второй бит кода является двоичным нулем, то код двухбитовый. В противном случае количество бит в коде более двух.

Если третий бит кода является двоичным нулем, то код трехбитовый. В противном случае количество бит в коде равно четырем.

Декодер восстанавливает информацию, используя то же самое дерево, двигаясь вверх при получении «0» и вниз при получении «1». Средняя длина слова в данном случае $l_{cp}=2,71$ бит/отсчет при энтропии $H=2,68$ бит/отсчет. При использовании кода равномерной длины потребовалось бы 3 бит/отсчет.

Рассмотренный выше пример показывает высокую эффективность хатфмановской процедуры при относительно равномерном распределении уровней пикселов.

Однако когда энтропия сообщения становится существенно меньше единицы, эффективность хатфмановского кодирования резко снижается, поскольку $l_{cp} \geq 1$, бит/пиксел.

По этой причине при использовании, например, 8-битовой исходной информации об изображении хатфмановское кодирование не позволяет сжать информацию более 8 раз. Идеальное сжатие алгоритмом Хатфмана достигается в том случае, когда вероятности символов точно равны отрицательным степеням числа 2.

Используются неадаптивный и адаптивный варианты хатфмановского кодирования. В первом случае перед передачей сообщения передается таблица плотностей вероятностей, если она заранее неизвестна на приемной стороне. При адаптивном варианте

кодирования таблица плотностей вероятностей вычисляется как на передающей, так и на приемной стороне по мере поступления данных. При этом до начала кодирования предполагается, например, равновероятное распределение уровней элементов. Также существуют кодеры с фиксированными таблицами вероятностей. Они подбираются экспериментальным путем для конкретного вида информации.

Алгоритм Хаффмана служит основой многих программ сжатия текстовой и графической информации, некоторые из них используют непосредственно алгоритм Хаффмана, а другие берут его в качестве одной из ступеней многоуровневого процесса сжатия.

1.1.4. Арифметическое кодирование

Метод Хаффмана является простым и эффективным, однако он порождает наилучшие коды переменной длины, только когда вероятности символов являются степенями числа два, т. е. равна $1/2$, $1/4$ и т. п. В этом случае символу будет присвоен код с целым числом бит. На практике вероятность символа произвольна, поэтому ему следует присвоить код с нецелым числом бит.

Арифметическое кодирование решает эту проблему путем присвоения кода последовательности символов вместо кодирования отдельных символов.

Алгоритм арифметического кодирования состоит из двух проходов по заданным данным. Первый проход используется алгоритмом для построения таблицы символов и частот их возникновения. На втором проходе каждой последовательности символов присваивается код постоянной длины.

Процесс арифметического кодирования иллюстрируется на рис. 1.4.

Процесс кодирования включает в себя следующие шаги для каждой кодируемой последовательности символов:

1. Инициализируется начальный интервал $[0, 1)$.
2. Начальный интервал разбивается на части, пропорциональные вероятностям каждого символа.
3. Выбирается подынтервал в соответствии с первым считанным символом последовательности.
4. Выбранный подынтервал становится текущим интервалом.
5. Пункты 2–4 повторяются для каждого последующего считанного символа из последовательности. В случае если все символы последовательности прочитаны, выходом алгоритма

будет любая точка, однозначно определяющая последний из текущих интервалов.

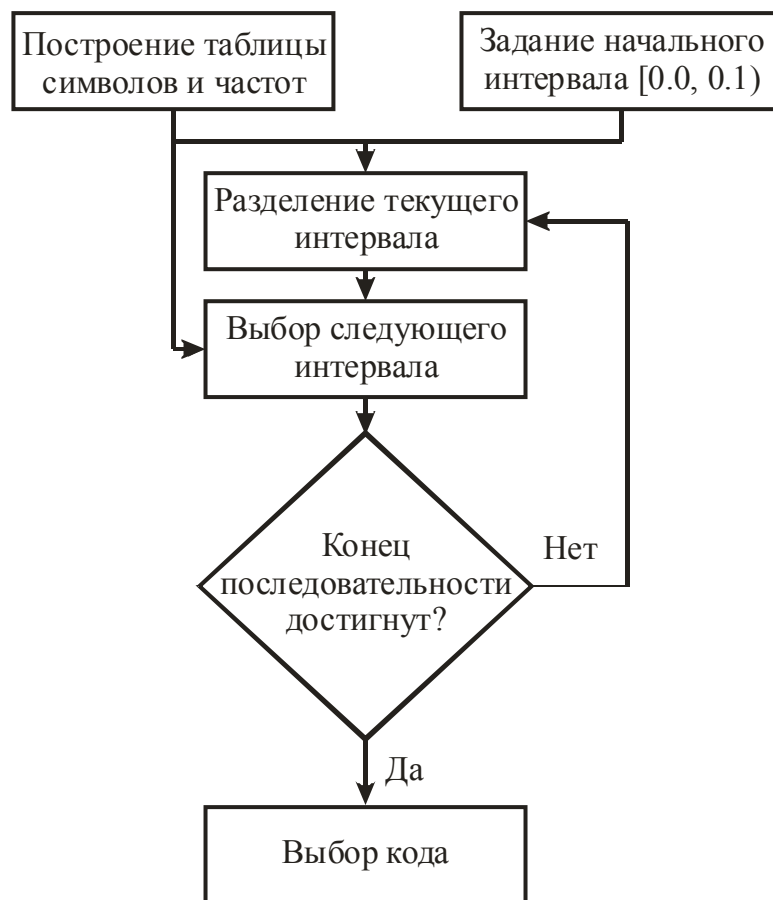


Рис. 1.4. Схема арифметического кодирования

После каждого обработанного символа текущий интервал становится все меньше, поэтому требуется все больше бит, чтобы представить его. Выходом алгоритма арифметического кодирования является единственное число, которое не является объединением индивидуальных кодов последовательности входных символов. Среднюю длину кода можно найти, разделив размер выхода (в битах) на размер входа (в символах).

Следующий пример призван пояснить процесс арифметического кодирования. Мы продемонстрируем шаги алгоритма для последовательности, представляющей строку изображения

[150, 25, 60, 60, 25, 150, 25, 60, 25, 25].

В табл. 1.3 указана информация, подготовленная на предварительном этапе. Для каждого символа сначала вычислена его частота, затем найдена вероятность его появления. Область $[0,1)$

делится между символами; каждый символ получает кусочек или подобласть, равную его вероятности. Символы и частоты таблицы записываются в начало выходного файла до битов кода сжатия (табл. 1.3).

Таблица 1.3

Частоты и вероятности символов последовательности

| Символ | Частота | Вероятность | Область |
|--------|---------|-------------|------------|
| 25 | 5 | 0.5 | [0.5; 1) |
| 60 | 3 | 0.3 | [0.2; 0.5) |
| 150 | 2 | 0.2 | [0; 0.2) |

Процесс кодирования начинается с инициализации двух переменных Low и High, которые отвечают соответственно за нижнюю и верхнюю границы интервала. Для начального интервала значения переменных будут равны соответственно 0 и 1. По мере поступления и обработки символов, переменные Low и High начинают сближаться, уменьшая интервал. Процесс изменения границ интервалов приведен в таблице.

Конечный код – это любое число, однозначно определяющее последний из интервалов [0.13114125, 0.131175), например, 0.13115. Это число переводится в целочисленный вид путем отбрасывания нуля и точки слева. Таким образом, кодом нашей последовательности будет число 13115, поскольку оно содержит наименьшее количество цифр и однозначно определяет последний интервал.

Декодер работает в обратном порядке. Сначала он узнает символы последовательности и частоты, им соответствующие (табл. 1.3). Затем он считывает число и, добавив к нему ноль слева, производит поиск интервала, в котором содержится заданное число.

Таким образом, поскольку каждому интервалу соответствует символ, мы определяем первый символ в последовательности. Далее декодер удаляет эффект распознанного символа из кода последовательности, путем вычитания нижней границы распознанного интервала из кода и деления результата на величину интервала. Мы получаем новое число, являющееся кодом для оставшихся символов последовательности. К нему применяются все выше описанные шаги декодирования. Таким образом, шаг за шагом декодер восстанавливает все символы последовательности.

Анализируя таблицу символов, декодер узнает, сколько раз необходимо повторить свой алгоритм для восстановления всей последовательности символов.

Для того чтобы выяснить степень сжатия, достигаемую с помощью арифметического кодирования, необходимо перевести все результаты в двоичную форму, а затем разделить размер входа на размер выхода в битах. Для нашего примера степень сжатия составила 4,21.

Таблица 1.4

Процесс арифметического кодирования

| Символ | Переменные | Значения переменных |
|--------|------------|--|
| 150 | Low | 0.0 |
| | High | 1.0 |
| 25 | Low | $0.0 + (1.0 - 0.0) \times 0.0 = 0.0$ |
| | High | $0.0 + (1.0 - 0.0) \times 0.2 = 0.2$ |
| 60 | Low | $0.0 + (0.2 - 0.0) \times 0.5 = 0.1$ |
| | High | $0.0 + (0.2 - 0.0) \times 1.0 = 0.2$ |
| 60 | Low | $0.1 + (0.2 - 0.1) \times 0.2 = 0.12$ |
| | High | $0.1 + (0.2 - 0.1) \times 0.5 = 0.15$ |
| 25 | Low | $0.12 + (0.15 - 0.12) \times 0.2 = 0.126$ |
| | High | $0.12 + (0.15 - 0.12) \times 0.5 = 0.135$ |
| 150 | Low | $0.126 + (0.135 - 0.126) \times 0.5 = 0.1305$ |
| | High | $0.126 + (0.135 - 0.126) \times 1.0 = 0.135$ |
| 25 | Low | $0.1305 + (0.1305 - 0.135) \times 0.0 = 0.1305$ |
| | High | $0.1305 + (0.1305 - 0.135) \times 0.2 = 0.1314$ |
| 60 | Low | $0.1305 + (0.1314 - 0.1305) \times 0.5 = 0.13095$ |
| | High | $0.1305 + (0.1314 - 0.1305) \times 1.0 = 0.1314$ |
| 25 | Low | $0.13095 + (0.1314 - 0.13095) \times 0.2 = 0.13104$ |
| | High | $0.13095 + (0.1314 - 0.13095) \times 0.5 = 0.131175$ |
| 25 | Low | $0.13104 + (0.13104 - 0.131175) \times 0.5 = 0.1311075$ |
| | High | $0.13104 + (0.13104 - 0.131175) \times 1.0 = 0.131175$ |
| 25 | Low | $0.1311075 + (0.131175 - 0.1311075) \times 0.5 = 0.13114125$ |
| | High | $0.1311075 + (0.131175 - 0.1311075) \times 1.0 = 0.131175$ |

Таблица 1.5

Процесс арифметического декодирования

| Код | Интервал | Символ |
|---------------------------------|------------|--------|
| 13115 \rightarrow 0.13115 | | |
| 0.13115 | [0.0, 0.2) | 150 |
| $(0.13115 - 0.0)/0.2 = 0.65575$ | [0.5, 1.0) | 25 |
| $(0.65575 - 0.5)/0.5 = 0.3115$ | [0.2, 0.5) | 60 |
| $(0.3115 - 0.2)/0.3 = 0.3716$ | [0.2, 0.5) | 60 |
| $(0.3716 - 0.2)/0.3 = 0.572$ | [0.5, 1.0) | 25 |
| $(0.572 - 0.5)/0.5 = 0.14$ | [0.0, 0.2) | 150 |
| $(0.14 - 0.0)/0.2 = 0.72$ | [0.5, 1.0) | 25 |
| $(0.72 - 0.5)/0.5 = 0.44$ | [0.2, 0.5) | 60 |
| $(0.44 - 0.2)/0.3 = 0.814$ | [0.5, 1.0) | 25 |
| $(0.814 - 0.5)/0.5 = 0.629$ | [0.5, 1.0) | 25 |

Для того чтобы увеличить степень сжатия, применяется адаптивный метод арифметического кодирования, в котором происходит динамическое построение таблицы символов и частот, им соответствующих.

Арифметическое кодирование в чистом виде не применяется для сжатия изображений. Оно входит в состав более сложных стандартов сжатия изображений, таких как JPEG и JPEG2000.

1.2. Алгоритмы кодирования с преобразованием

Методы кодирования с преобразованием, в общем случае, основаны на приведении изображений к виду, наиболее удобному с точки зрения сокращения избыточной информации. При кодировании с преобразованием осуществляется сокращение как психофизиологической, так и статистической избыточности на основе преобразования информации из пространственной области в область трансформант.

Процесс преобразования изображений, как правило, основан на разделении изображения на фрагменты и их отдельной обработке. В связи с тем, что в общем случае изображение характеризуется двумерными зависимостями, исходные фрагменты также должны носить двумерный характер. Размер фрагмента является важным с практической точки зрения параметром. Типичный фрагмент состоит из 8×8 пикселей. Основанием к такому выбору размера фрагмента служит то, что интервал корреляции в изображениях в среднем не превышает 8 пикселей.

Результат преобразования представляет собой спектральные коэффициенты преобразования, называемые трансформантами, которые характеризуются амплитудой и обобщенной пространственной частотой – координатой в спектральной области.

В основу преобразования изображений могут быть положены различные принципы. Наиболее часто используются методы унитарных линейных ортогональных преобразований. Различают ортогональные преобразования изображений, в математическом аппарате которых непосредственно используются статистические свойства видеoinформации (например, преобразование Карунена – Лоэва), и преобразования, в процессе которых статистические характеристики изображения или совсем не учитываются, или учитываются в ограниченной степени (дискретное преобразование Фурье, преобразование Уолша – Адамара, S-преобразование и т. д.). Линейность ортогонального преобразования означает, что операции сложения, вычитания и умножения на скаляр действительны и после

преобразования, а ортогональность означает, что преобразуемый фрагмент представляется ограниченным набором ортогональных функций. Унитарность ортогонального преобразования означает схожесть математических аппаратов при прямом и обратном преобразованиях.

Линейное преобразование можно осуществить в общем случае с непрерывным или дискретным сигналом, процессу преобразования при этом будет соответствовать или интегральная, или матричная форма записи.

Линейные ортогональные преобразования характеризуются тем, что между элементами изображения устраняются статистические зависимости и распределение энергии в преобразованном спектральном фрагменте является неравномерным. Эти особенности используются в процессах кодирования.

Для высокоэффективного кодирования с преобразованием необходимо, чтобы преобразование осуществлялось в соответствии с параметрами источника сигнала. Эффективность кодирования зависит от того, как в процессе выполнения алгоритма преобразования перегруппируется энергия в преобразованном сигнале. При равенстве энтропии исходного и преобразованного для последующей обработки сигналов желательно, чтобы энергия после преобразования в спектральную область была сосредоточена в минимально возможном количестве спектральных коэффициентов (трансформант). При линейных ортогональных преобразованиях оптимальное выполнение указанного условия достигается в случае, когда составляющие векторов матрицы преобразования соответствуют типичным пространственным характеристикам двумерного сигнала в горизонтальном и вертикальном направлениях.

Важным параметром алгоритмов преобразования является объем вычислений. Последний определяет сложность устройства, реализующего преобразование. За исключением преобразования Карунена – Лоэва (ПКЛ), реализация большинства используемых преобразований осуществляется с помощью быстрых алгоритмов, таких как алгоритм быстрого преобразования Фурье (БПФ). Субоптимальные алгоритмы преобразования Адамара и Хаара можно выполнить без операции умножения.

1.2.1. Ухудшение качества изображения

Для кодирования изображений вообще и кодирования с преобразованием в частности не существует мер качества изображения, количественно хорошо коррелированных с

субъективными оценками различных искажений изображения, возникающих в кодере. Однако даже при отсутствии объективных количественных мер можно проанализировать искажения, возникающие в процессе кодирования с преобразованием. Эти искажения можно разделить на три класса:

- возникающие при квантовании трансформант;
- возникающие при приравнивании значений некоторых трансформант нулю (в действительности большая часть сокращения избыточности получается в результате такой низкочастотной фильтрации);
- связанные с тем, что кодирование с преобразованием не является инвариантным к сдвигам. Поскольку обработке подвергаются смежные фрагменты, то может возникнуть нежелательный фрагментарный шум. При сдвиге начала процесса преобразования восстановленные изображения можно различить.

Визуально во многих случаях эти эффекты можно разделить. Квантование трансформант вносит дополнительный шум в изображение. Замена трансформант нулями в кодере эквивалентна низкочастотной фильтрации. При сравнении восстановленного изображения с исходным может оказаться, что часть деталей потеряна. Низкочастотная фильтрация достаточно заметна при больших коэффициентах сжатия. Этот эффект зависит от типа применяемого преобразования, но даже при косинусном преобразовании на малых скоростях искажение на границах фрагментов явно нежелательно. Термин «низкая скорость передачи» означает, что сохраняется только малая часть трансформант, скажем, 10%. Косинусное преобразование менее склонно к образованию фрагментального шума, однако оно также вносит подобные искажения.

Несмотря на отсутствие надежных математических мер, существуют методы оценивания ухудшения качества восстановленных изображений по сравнению с исходными. Весьма наглядным является так называемое «изображение ошибок кодирования», представляющее собой изображение абсолютной разности между исходным и восстановленным изображениями. Для наблюдений сигнал изображения ошибок кодирования обычно усиливается в 10–40 раз. Если информационные потери малы, то и информация, содержащаяся в изображении ошибок кодирования, будет незначительной. Для высококачественных систем кодирования изображение ошибок кодирования представляет собой белый шум.

1.2.2. Преобразование Фурье

Двумерное дискретное преобразование Фурье (ДПФ) матрицы отсчетов изображения определяется в виде ряда

$$F(u, v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \exp\left\{-\frac{2\pi i}{N}(uj + vk)\right\}, \quad (1.1)$$
$$i = \sqrt{-1},$$

а обратное дискретное преобразование Фурье имеет вид

$$f(j, k) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp\left\{\frac{2\pi i}{N}(uj + vk)\right\}.$$

По аналогии с терминологией временного преобразования Фурье переменные u, v называют пространственными частотами. Поскольку ядра преобразования симметричны и разделимы, двумерное преобразование можно выполнить в виде последовательных одномерных преобразований по строкам и столбцам матрицы изображения. Базисными функциями преобразования являются экспоненты с комплексными показателями, которые можно разложить на синусную и косинусную составляющие.

Таким образом,

$$A(j, k, u, v) = \exp\left\{-\frac{2\pi i}{N}(uj + vk)\right\} =$$
$$= \cos\left\{\frac{2\pi}{N}(uj + vk)\right\} - i \sin\left\{\frac{2\pi}{N}(uj + vk)\right\},$$

$$B(j, k, u, v) = \exp\left\{\frac{2\pi i}{N}(uj + vk)\right\} =$$
$$= \cos\left\{\frac{2\pi}{N}(uj + vk)\right\} + i \sin\left\{\frac{2\pi}{N}(uj + vk)\right\}.$$

Для низких частот базисные функции преобразования Фурье являются грубыми аппроксимациями непрерывных синусоид. С повышением частоты сходство с синусоидами теряется. Для

наивысшей частоты базисная функция представляет собой меандр. Можно заметить также избыточность наборов синусных и косинусных составляющих.

Спектр изображения имеет много интересных структурных особенностей. Спектральная составляющая в начале координат частотной плоскости

$$F(0,0) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k)$$

равна увеличенному в N раз среднему (по исходной плоскости) значению яркости изображения. Подставив в равенство (1.1) $u = u + mN$ и $v = v + nN$, где m и n – постоянные, получим

$$F(u + mN, v + nN) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \exp\left\{-\frac{2\pi i}{N}(uj + vk)\right\} \times \exp\{-2\pi i(mj + nk)\}. \quad (1.2)$$

При любых целочисленных значениях m и n второй экспоненциальный множитель равенства (1.2) 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.

Двумерный спектр Фурье изображения является по существу представлением двумерного поля в виде ряда Фурье. Для того чтобы такое представление было справедливым, исходное изображение должно обладать периодической структурой, т. е. 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.

1.2.3. Двумерное преобразование Хартли

Обращение к преобразованию Хартли обусловлено ситуацией, сложившейся в ряде методов обработки информации, в частности

использующих вещественные последовательности данных. Обработку таких данных желательно осуществить в области вещественных чисел с помощью взаимно симметричных прямого и обратного преобразований. В отличие от преобразования Фурье, отображающего вещественные функции в комплексную область, несимметричное по i преобразование Хартли осуществляет прямое и обратное преобразования только в вещественной области и обладает указанной симметрией.

Двумерное преобразование Хартли обладает интересным свойством фундаментального характера. Данное преобразование является вещественным и в двумерном случае. В силу того, что преобразование Фурье является комплексным, для его определения достаточно половины плоскости преобразования. Остальная часть этой плоскости принадлежит сопряженным значениям, которые не содержат дополнительной информации, так как диаметрально противоположные точки характеризуются сопряженными коэффициентами. В плоскости преобразования Хартли, напротив, отсутствуют подобные симметрия и избыточность; информация распределяется по всей области с меньшей в два раза интенсивностью.

Двумерное дискретное преобразование Хартли имеет следующий вид:

$$H(v_1, v_2) = \frac{1}{N_1 N_2} \sum_{\tau_1=0}^{N_1-1} \sum_{\tau_2=0}^{N_2-1} f(\tau_1, \tau_2) \text{cas} \left[\left(\frac{2\pi v_1 \tau_1}{N_1} \right) + \left(\frac{2\pi v_2 \tau_2}{N_2} \right) \right],$$

$$f(\tau_1, \tau_2) = \sum_{v_1=0}^{N_1-1} \sum_{v_2=0}^{N_2-1} H(v_1, v_2) \text{cas} \left[\left(\frac{2\pi v_1 \tau_1}{N_1} \right) + \left(\frac{2\pi v_2 \tau_2}{N_2} \right) \right],$$

где $\text{cas } x = \sin x + \cos x$.

Двумерное дискретное преобразование Фурье в тех же обозначениях определяется как

$$F(v_1, v_2) = \frac{1}{N_1 N_2} \sum_{\tau_1=0}^{N_1-1} \sum_{\tau_2=0}^{N_2-1} f(\tau_1, \tau_2) \exp \left[\left(-j \frac{2\pi v_1 \tau_1}{N_1} \right) + \left(-j \frac{2\pi v_2 \tau_2}{N_2} \right) \right],$$

$$f(\tau_1, \tau_2) = \sum_{v_1=0}^{N_1-1} \sum_{v_2=0}^{N_2-1} F(v_1, v_2) \exp \left[\left(j \frac{2\pi v_1 \tau_1}{N_1} \right) + \left(j \frac{2\pi v_2 \tau_2}{N_2} \right) \right].$$

Преобразование Фурье может быть получено из преобразования Хартли, и наоборот. Пусть F_r – вещественная часть преобразования

Фурье, а F_i – мнимая часть. Тогда $H = F_r - F_i$, т. е. функция H определяется как разность вещественной и мнимой составляющих преобразования Фурье. Соответственно если E – четная, а O – нечетная составляющие функции H , то $F_r = E$, $F_i = O$.

1.2.4. Двумерное дискретное преобразование Уолша

Известно, что, кроме тригонометрических ортогональных функций, в качестве базисных могут применяться и другие ортогональные функции. Для того чтобы стало ясно, почему получили широкое применение выполняемые на основе этих базисных функций преобразования, укажем их особенности, по сравнению с преобразованием Фурье.

Преобразованию Фурье присущ недостаток, которого лишено преобразование Уолша. Этот недостаток заключается в следующем: часто бывает важным сведение к минимуму времени машинного выполнения производимых действий. Создание алгоритма быстрого преобразования Фурье позволило существенно ускорить обработку информации. Но даже для него сохраняется необходимость в выполнении большого количества умножений, занимающих большую часть времени при машинной обработке данных.

Следует иметь в виду и то, что, хотя для ряда функций достаточно удовлетворительная их аппроксимация получается уже при сравнительно небольшом числе коэффициентов ряда Фурье, для многих сигналов чаще всего не выполняется условие быстрой сходимости ряда Фурье. При приближенной аппроксимации функции тригонометрическими рядами Фурье операцию умножения приходится иногда производить миллионы и миллиарды раз, и это занимает основную часть машинного времени.

Кусочно-постоянная базисная функция имеет на выделенном интервале времени постоянное значение. В формулу коэффициента a_k разложения в ряд входит величина, являющаяся для данного интервала разложения постоянной. Если при всех k на заданном интервале значением базисной функции является 1 или -1, то отпадает необходимость выполнения каждый раз операции умножения. При приближенном вычислении коэффициентов разложения функции в ряд по базисным функциям, принимающим значения 1, -1 или 0, должны производиться существенно более простые, чем умножение, операции сложения и вычитания. В значительном упрощении и ускорении обработки информации заключается смысл применения в качестве базисных кусочно-постоянных функций.

Существуют различные способы определения функций Уолша. Рассмотрим способ, основанный на взаимосвязи функций Уолша и функций Радемахера. Последние получаются из синусоидальных функций с помощью соотношения

$$r_k(\theta) = \text{sign}([2^k \theta \pi]), \\ 0 \leq \theta < 1,$$

где аргумент $\theta = t/T_0$ есть безразмерное время, т. е. время, нормированное к произвольному интервалу T_0 , а целое положительное число k – порядок функции. Символом sign обозначается функция знака:

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ -1, & x < 0. \end{cases} \quad (1.3)$$

В соответствии с (1.3) функции Радемахера принимают одно из двух значений (1 или -1) и имеют вид меандра.

Функции Уолша, образующие полную ортонормированную систему, можно сформировать, образуя произведения степеней соответствующих функций Радемахера. Способ построения функций Уолша можно выразить аналитически для любого $N = 2^n$:

$$\text{wal}(\omega\theta) = \prod_{k=1}^n [r_k(\theta)]^{W_{n-k+1} \otimes W_{n-k}}.$$

Для цифровых методов спектрального анализа и обработки сигналов наибольший интерес представляют дискретные функции Уолша. Эти функции являются отсчетами непрерывных функций Уолша. Каждый отсчет расположен в середине связанного с ним элемента непрерывной функции. Длительность элемента равна $1/N$ от интервала $[0,1]$.

Двумерный сигнал можно подвергнуть двумерному спектральному анализу. Для этого сигнал $s[k,l]$ нужно преобразовать в двумерный спектр $S[n,m]$ по системе двумерных базисных функций $\varphi_{n,m}(k,l)$. В качестве последних могут быть применены функции Уолша, имеющие вид

$$\varphi_{n,m}(k,l) = \text{wal}(n,k)\text{wal}(m,l).$$

Тогда преобразования

$$S[n, m] = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} s[k, l] \text{wal}(n, k) \text{wal}(m, l),$$

$$s[k, l] = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} S[n, m] \text{wal}(n, k) \text{wal}(m, l),$$

определяют пару двумерных дискретных преобразований Уолша.

1.2.5. Дискретное косинусное преобразование

К методам линейных ортогональных преобразований, которые в последнее время широко применяются в области цифровой обработки информации, относится и метод дискретного косинусного преобразования (ДКП), основанный на использовании ортогональной системы дискретных косинусных функций возрастающих частот.

Прямое одномерное ДКП будет определяться выражением

$$C(k) = \frac{2}{N} c(k) \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2n+1}{2N} k\pi\right), \quad k \in \{0, 1, \dots, N-1\},$$

а обратное ДКП

$$f(n) = \sum_{k=0}^{N-1} c(k) C(k) \cos\left(\frac{2n+1}{2N} k\pi\right), \quad n \in \{0, 1, \dots, N-1\},$$

где

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k \neq 0 \end{cases}.$$

Матричная форма записи прямого ДКП имеет вид

$$[C(k)] = [\varphi(k, n)][f(n)],$$

где $[C(k)]$ – матрица-столбец спектральных коэффициентов ДКП размером $(1 \times N)$; $[\varphi(k, n)]$ – матрица дискретного множества ортогональных функций ДКП размером $(N \times N)$; $[f(n)]$ – матрица-столбец дискретных значений (отсчетов) сигнала. Соответствующее

обратное одномерное преобразование в матричной форме записи представляется в виде

$$[f(n)] = [\varphi(k, n)]^{-1} [C(k)],$$

а поскольку условие симметричности матрицы преобразования по отношению к ее главной диагонали не выполняется

$$[\varphi(k, n)]^{-1} = [\varphi(k, n)]^T,$$

получим

$$[f(n)] = [\varphi(k, n)]^T [C(k)].$$

Для преобразования двумерных фрагментов изображения определим понятие двумерного преобразования.

Определим прямое двумерное косинусное преобразование как

$$F(u, v) = \frac{4c(u)c(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \cos \frac{(2j+1)u\pi}{2N} \times \\ \times \cos \frac{(2k+1)v\pi}{2N},$$

а обратное как

$$f(j, k) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) c(u) c(v) \cos \frac{(2u+1)j\pi}{2N} \times \\ \times \cos \frac{(2v+1)k\pi}{2N},$$

где

$$c(u), c(v) = \begin{cases} \frac{1}{\sqrt{2}}, & u, v = 0 \\ 1, & u, v \neq 0 \end{cases}.$$

Разделимое ядро ДКП (без учета константы нормирования) имеет вид

$$\begin{aligned}\varphi(k, n, l, m) &= \cos \frac{2n+1}{2N} k\pi \cdot \cos \frac{2m+1}{2N} l\pi, \\ k, n &\in \{0, 1, \dots, N-1\}, \\ l, m &\in \{0, 1, \dots, M-1\}.\end{aligned}$$

Таким образом, прямое ДКП двумерного фрагмента форматом $(N \times M)$ запишется как

$$[C(k, l)] = [\varphi(k, n)][f(n, m)][\varphi(l, m)]^T,$$

где $[f(n, m)]$ – сигнальная матрица размером в общем случае $(N \times M)$; $[C(k, l)]$ – соответствующая матрица спектральных коэффициентов ДКП размером в общем случае $(N \times M)$; $[\varphi(k, n)]$ – квадратная матрица ДКП размером $(N \times N)$:

$$[\varphi(k, n)] = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \cos \frac{2n+1}{2N} k\pi \end{bmatrix}, \quad \begin{aligned} k &\in \{0, 1, \dots, N-1\}, \\ n &\in \{0, 1, \dots, N-1\}, \end{aligned}$$

$[\varphi(l, m)]^T$ – квадратная матрица ДКП размером $(M \times M)$, транспонированная по отношению к матрице

$$[\varphi(l, m)] = \sqrt{\frac{2}{M}} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \cos \frac{2m+1}{2M} k\pi \end{bmatrix}, \quad \begin{aligned} l &\in \{0, 1, \dots, M-1\}, \\ m &\in \{0, 1, \dots, M-1\}. \end{aligned}$$

Соответствующая матричная форма записи обратного двумерного ДКП имеет вид

$$[f(n, m)] = [\varphi(k, n)]^T [C(k, l)][\varphi(l, m)],$$

где символом T обозначен оператор транспонирования.

Преобразования, основанные на гармонических функциях, такие как ДКП и ДПФ, лучше приближаются к оптимальной упаковке информации, достигаемой преобразованием Карунена – Лоэва. Благодаря этому многие системы трансформационного кодирования основываются на ДКП, которое дает хороший компромисс между степенью сжатия и вычислительной сложностью. Кроме того, по

сравнению с другими преобразованиями, ДКП обеспечивает упаковку наибольшего количества информации в наименьшее число коэффициентов для большинства изображений. По сравнению с ДПФ, ДКП минимизирует эффект блочности структуры, проявляющейся в том, что становятся видны границы между соседними блоками. Оно уменьшает этот эффект потому, что его периодичность в $2n$ точек не приводит к разрывам на границе блоков, в отличие от ДПФ.

Дискретное косинусное преобразование обладает лучшей способностью к упаковке информации, по сравнению с ДПФ и преобразованием Уолша – Адамара. Хотя эта ситуация справедлива для большинства изображений, тем не менее оптимальным в смысле минимума среднеквадратичной ошибки восстановленных изображений является преобразование Карунена – Лоэва.

1.2.6. Преобразование Карунена – Лоэва

Метод преобразования непрерывных сигналов в набор некоррелированных коэффициентов разработан Каруненом и Лоэвом. В общем случае преобразование Карунена – Лоэва описывается соотношением

$$\Psi(u, v) = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} F(j, k) A(j, k; u, v),$$

ядро $A(j, k; u, v)$ которого удовлетворяет соотношению

$$\lambda(u, v) A(j, k; u, v) = \sum_{j'=0}^{N-1} \sum_{k'=0}^{N-1} K_F(j, k; j', k') A(j, k; u, v),$$

где $K_F(j, k; j', k')$ – ковариационная функция дискретизированного изображения, а $\lambda(u, v)$ при фиксированных u и v постоянна. Функции $A(j, k; u, v)$ являются собственными функциями ковариационной функции, а $\lambda(u, v)$ – ее собственными значениями. Как правило, выразить собственные функции в явном виде не удастся.

Если ковариационную функцию можно разделить, т. е.

$$K_F(j, k; j', k') = K_C(j, j') K_R(k, k'),$$

то ядро разложения Карунена – Лоэва также делимо и

$$A(j, k; u, v) = A_C(j, u) A_R(k, v).$$

Строки и столбцы матриц, описывающих эти ядра, удовлетворяют следующим уравнениям:

$$\lambda_R(v)A_R(k,v) = \sum_{k'=0}^{N-1} K_R(k,k')A_R(k',v),$$

$$\lambda_C(u)A_C(j,u) = \sum_{j'=0}^{N-1} K_C(j,j')A_C(j',u).$$

В частном случае, когда ковариационная матрица описывает разделимый марковский процесс первого порядка, собственные функции удастся записать в явном виде. Для одномерного марковского процесса с коэффициентом корреляции ρ собственные функции и собственные значения имеют вид

$$A(j,u) = \left[\frac{2}{N + \lambda^2(u)} \right]^{1/2} \sin[w(u)(j - \frac{N-1}{2}) + \frac{(u+1)\pi}{2}]$$

и

$$\lambda(u) = \frac{1 - \rho^2}{1 - 2\rho \cos[w(u)] + \rho^2},$$

где $0 \leq j$, и $u \leq N-1$, а $w(u)$ – корни трансцендентного уравнения

$$\operatorname{tg}(Nw) = \frac{(1 - \rho^2) \sin(w)}{\cos(w) - 2\rho + \rho^2 \cos(w)}.$$

Собственные векторы можно также найти из рекуррентных формул

$$A(0,u) = \frac{\lambda(u)}{1 - \rho^2} [A(0,u) - \rho A(1,u)],$$

$$A(j,u) = \frac{\lambda(u)}{1 - \rho^2} [-\rho A(j-1,u) + (1 + \rho^2)A(j,u) - \rho A(j+1,u)],$$

$$0 < j < N-1,$$

$$A(N-1,u) = \frac{\lambda(u)}{1 - \rho^2} [-\rho A(N-2,u) + A(N-1,u)],$$

положив в качестве начального условия $A(0, u) = 1$ и затем пронормировав полученные собственные векторы.

Преобразование Карунена – Лозва является оптимальным в смысле минимизации среднеквадратичной ошибки восстановленного изображения для любого входного изображения и любого числа сохраненных коэффициентов. Однако поскольку базисные функции ПКЛ зависят от входного изображения, то их вычисление становится нетривиальной задачей. По этой причине ПКЛ для сжатия изображений используется редко.

1.3. Сжатие изображений на основе вейвлет-преобразования

За последние два десятилетия в мире возникло и оформилось новое научное направление, связанное с так называемым вейвлет-преобразованием. Слово «wavelet», являющееся переводом французского «ondelette», означает небольшие волны, следующие друг за другом. В узком смысле вейвлеты – это семейство функций, получающихся путем масштабирования и сдвигов одной, материнской, функции. В широком смысле вейвлеты – это функции, обладающие хорошей частотной локализацией, чье среднее значение равно нулю.

В настоящее время большое развитие получила практика применения вейвлетов для решения задач сжатия и обработки изображений, являющихся нестационарными по своей природе. На данный момент разработано несколько широко известных алгоритмов сжатия изображений (JPEG2000, JPEG XR, EZW, SPIHT, IW44), в основе которых лежит вейвлет-преобразование. Материал настоящего пункта направлен на рассмотрение основных моментов вейвлет-теории и применение последней в области сжатия статических изображений.

1.3.1. Некоторые сведения о вейвлет-преобразовании

1.3.1.1. Непрерывное вейвлет-преобразование

Под непрерывным вейвлет-преобразованием (НВП) функции $f(t) \in L^2(R)$ (здесь $L^2(R)$ – пространство функций конечной энергии, таких что $\int |f(t)|^2 dt < +\infty$) понимается ее скалярное произведение с базисными функциями $\psi_{a,b}(t) = a^{-1/2} \psi\left(\frac{t-b}{a}\right)$, $a \in R^+$, $b \in R$, то есть

разложение по всем возможным сдвигам и сжатиям некоторой функции (порождающего вейвлета):

$$W_f(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt.$$

Если для порождающего вейвлета $\psi(t)$ выполняется равенство $C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$, то возможно обратное преобразование:

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_0^\infty W_f(a, b) \psi_{a,b}(t) \frac{da db}{a^2}.$$

Здесь $\Psi(\omega)$ – преобразование Фурье функции $\psi(t)$.

1.3.1.2. Ортогональное дискретное вейвлет-преобразование

В отличие от непрерывного вейвлет-преобразования в данном случае рассматриваются не все сдвиги и растяжения базисной функции, а только взятые на некоторой дискретной сетке (обычно логарифмической). Здесь необходимо отметить, что если сигнал остается непрерывным, то называть это преобразование дискретным неверно. Если же сигнал – дискретный, то аналогичное преобразование правильно называть дискретным вейвлет-преобразованием (ДВП).

1.3.1.3. Кратномасштабный анализ

Кратномасштабный анализ (КМА) – математическая конструкция, схема представления сигналов. Данная конструкция заключается в представлении пространства в виде бесконечной последовательности вложенных подпространств, являющихся отмасштабированными версиями друг друга и связанными определенными свойствами. Значение КМА заключается также в том, что с его помощью можно дать более точные определения масштабирующей и вейвлет-функциям, а также найти связь между ними.

1.3.1.4. Масштабирующая функция

Масштабирующей (скейлинг) функцией, или «отцовским вейвлетом», называется функция, которая определяется через масштабирующее уравнение вида:

$$\varphi(t) = \sqrt{2} \sum_l h_l \varphi(2t - l),$$

где h_l – масштабные коэффициенты для масштабирующей функции. Отцовский вейвлет порождает семейство масштабирующих функций $\varphi_{j,k} = 2^{j/2} \varphi(2^j t - k)$, где j – параметр масштаба, k – параметр сдвига.

Масштабирующая функция вместе с материнским вейвлетом порождают ортогональный КМА.

1.3.1.5. Материнский (базисный) вейвлет

Материнский вейвлет – это функция, которую можно построить по известной масштабирующей функции, введенной выше:

$$\psi(t) = \sqrt{2} \sum_l g_l \varphi(2t - l),$$

где g_l – масштабные коэффициенты для вейвлет-функции, которые связаны с масштабными коэффициентами для масштабирующей функции следующим образом:

$$g_l = (-1)^l h_{1-l}.$$

Материнский вейвлет порождает семейство вейвлет-функций $\psi_{j,k} = 2^{j/2} \psi(2^j t - k)$, где j – параметр масштаба, k – параметр сдвига.

1.3.1.6. Биортогональные масштабирующие функции и вейвлеты

Биортогональные вейвлет-системы состоят из четырех множеств функций: базиса масштабирующих функций $\{\varphi_{j,k}\}$, двойственного к нему базиса $\{\tilde{\varphi}_{j,k}\}$, базиса вейвлет-функций $\{\psi_{j,k}\}$, двойственного к нему базиса $\{\tilde{\psi}_{j,k}\}$. Условие биортогональности требует, чтобы эти множества функций удовлетворяли следующим свойствам:

$$\left. \begin{aligned} \langle \varphi_{j,k}, \tilde{\psi}_{j,l} \rangle &= 0 \\ \langle \psi_{j,k}, \tilde{\varphi}_{j,l} \rangle &= 0 \end{aligned} \right\} \text{ при всех } j, k, l.$$

$$\left. \begin{aligned} \langle \varphi_{j,k}, \tilde{\varphi}_{j,l} \rangle &= 0 \\ \langle \psi_{j,k}, \tilde{\psi}_{j,l} \rangle &= 0 \end{aligned} \right\} \text{ при всех } k \neq l.$$

Здесь $\langle \bullet \rangle$ – означает скалярное произведение двух функций, т. е. произведение вида:

$$\langle f, g \rangle = \int f(t)g(t)dt.$$

1.3.1.7. Ортогональные масштабирующие функции и вейвлеты

Ортогональные системы масштабирующих и вейвлет-функций являются частным случаем биортогональных, в том случае, когда $\varphi_{j,k} \equiv \tilde{\varphi}_{j,k}$ и $\psi_{j,k} \equiv \tilde{\psi}_{j,k}$.

1.3.1.8. Быстрое вейвлет-преобразование

Быстрое вейвлет-преобразование (БВП), или иерархический алгоритм Малла (Malla), представляет собой эффективный метод реализации вычислений дискретного вейвлет-преобразования, который использует взаимосвязь между коэффициентами ДВП соседних масштабов.

Общая схема осуществления данного преобразования (в одномерном случае) следующая:

Шаг 1. Подаем входной одномерный сигнал $\{c_n^0\}_{n=0}^{N-1}$, где верхний индекс показывает уровень разложения (0), нижний – конкретную точку сигнала, на пару фильтров (низкочастотный и высокочастотный фильтры разложения).

Шаг 2. Результат обработки фильтров подаем на блоки децимации или прореживания, то есть блоки, отвечающие за отбрасывание отсчетов, чей порядковый номер кратен определенному числу. В классическом алгоритме Малла прореживание отсчетов происходит в два раза.

Шаг 3. Результатом обработки Шага 1 и Шага 2 являются два набора вейвлет-коэффициентов $\{c_n^1\}_{n=0}^{M-1}$ и $\{d_n^1\}_{n=0}^{M-1}$

(низкочастотные и высокочастотные соответственно), где в случае четного входного сигнала и его периодического дополнения с краю $M = N/2$. Высокочастотные коэффициенты $\{d_n^1\}_{n=0}^{M-1}$ оставляем без изменения, а низкочастотные коэффициенты $\{c_n^1\}_{n=0}^{M-1}$ преобразуем (если в этом есть необходимость) аналогично преобразованию исходного сигнала, используя **Шаг 1** и **Шаг 2** (то есть в алгоритме выполняется итерация по низкочастотному каналу).

Сказанное выше можно представить математически:

$$\begin{cases} c_n^m = \sum_l h_{l-2n} c_l^{m-1} \\ d_n^m = \sum_l g_{l-2n} c_l^{m-1} \end{cases} \quad (1.4)$$

где индекс m указывает на уровень разложения, а выражение (1.4) представляет собой свертку сигнала или его низкочастотных компонент с импульсными характеристиками КИХ-фильтров (низкочастотного – h и высокочастотного – g).

Структурная схема разложения **Шаг 1 - Шаг 3** представлена на рис. 1.5.

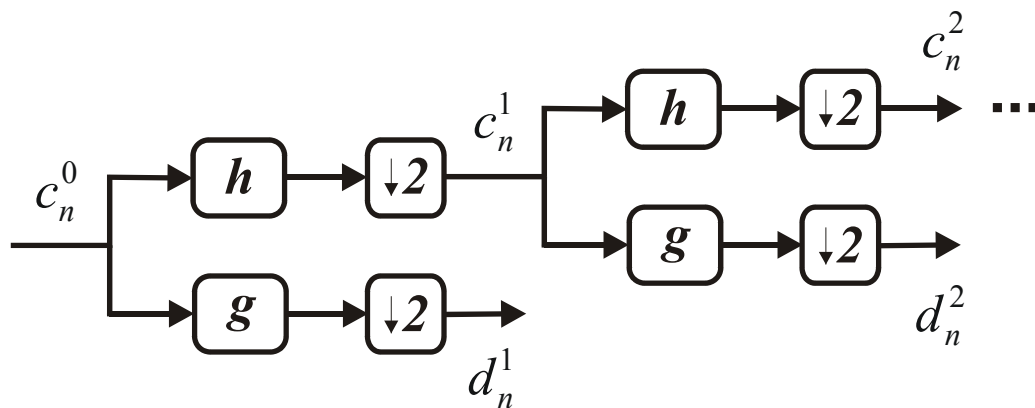


Рис. 1.5. Структурная схема одномерного БВП

Схема восстановления сигнала по известным наборам коэффициентов разложения является симметричной схеме разложения с заменой низкочастотных и высокочастотных фильтров разложения на соответствующие фильтры восстановления, а блоков децимации на блоки интерполяции, то есть блоки, отвечающие за встраивание между отсчетами, чей порядковый номер кратен определенному числу, некоторой константы. В классическом алгоритме Малла интерполяция отсчетов происходит в два раза, а в качестве константы выступает ноль.

Если необходимо осуществить двумерное БВП изображения, то для случая разделимых двумерных фильтров необходимо применить схему разложения вначале к строкам изображения, а затем к его столбцам с уже преобразованными строками. Пример одноуровневой вейвлет-декомпозиции изображения приведен на рис. 1.6.

Декомпозиция изображения (полученная на основе БВП), подобная той, которая представлена на рис. 1.6, может использоваться в алгоритмах JPEG2000 и SPIHT, рассматриваемых далее, для обеспечения сжатия изображений с потерей информации.

1.3.1.9. Целочисленное вейвлет-преобразование

В основе данного метода лежит модификация вейвлет-преобразования, позволяющая производить все вычисления в целочисленном виде. Теоретически при вейвлет-преобразовании потери информации не происходит. Однако при его реализации возникают неизбежные ошибки округления вейвлет-коэффициентов.



Рис. 1.6. а) Тестовое изображение Barbara; б) одноуровневая вейвлет-декомпозиция тестового изображения, где LL – выделенная низкая частота по горизонтали и низкая по вертикали, LH – низкая по горизонтали и высокая по вертикали, HL – высокая по горизонтали и низкая по вертикали, HH – высокая по горизонтали и по вертикали

Целочисленное вейвлет-преобразование позволяет достичь полного контроля над точностью вычислений, поэтому часто его называют обратимым вейвлет-преобразованием.

Общая схема построения данного вейвлет-преобразования состоит в выборе некоторого простого вейвлет-преобразования и изменении по определенному правилу высокочастотных

коэффициентов, для получения целочисленного преобразования, эффективного для кодирования изображения.

Данная схема преобразования может использоваться в алгоритмах JPEG2000 и SPIHT для обеспечения сжатия изображений без потери информации.

1.3.1.10. Лифтинговая схема вычисления вейвлет-преобразования

Лифтинговая схема – это альтернативный пирамиде Малла способ быстрого вычисления вейвлет-преобразования. По некоторым оценкам данная схема снижает вычислительную стоимость в два раза, позволяет экономить память и конструировать вейвлеты, которые другим способом не построить (вейвлеты второго поколения).

Лифтинговая схема состоит из трех звеньев: разбиения, предсказания и обновления, обозначенных на рис. 1.7 буквами S, P, U соответственно. Как видно из рис. 1.7, на вход данной схемы подается некоторый сигнал (в данном случае одномерный) $\{c_n^0\}_{n=0}^{N-1}$, который, претерпевая определенные преобразования в блоках разбиения, предсказания и обновления, порождает низкочастотные $\{c_n^1\}_{n=0}^{M-1}$ и высокочастотные $\{d_n^1\}_{n=0}^{M-1}$ наборы коэффициентов. Более подробно описание работы блоков данной схемы можно найти в соответствующей научно-технической литературе.

Подобного рода конструкция может использоваться в алгоритме JPEG2000 для осуществления сжатия с потерей и без потери информации.

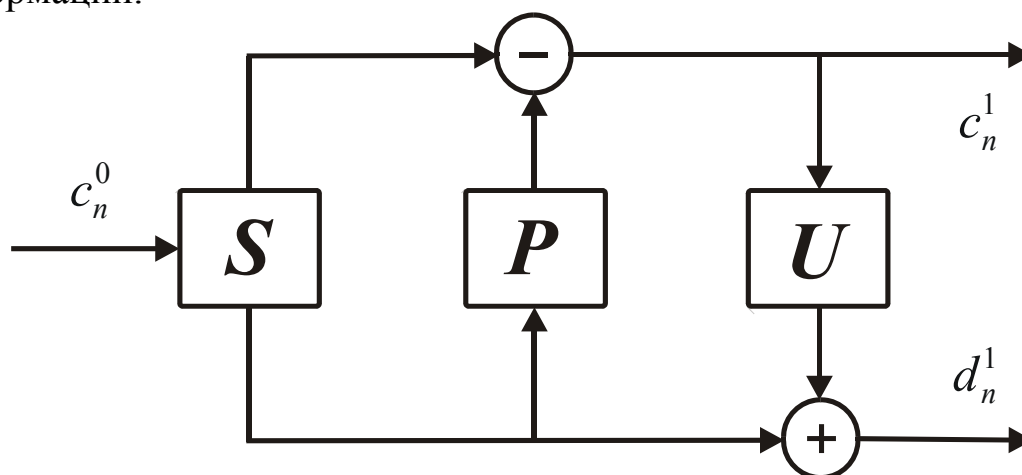


Рис. 1.7. Лифтинговая схема вычисления вейвлет-преобразования

2. СТАНДАРТ JPEG

JPEG (Joint Photographic Expert Group – объединенная группа экспертов в области фотографии) – это первый международный стандарт для сжатия неподвижных цифровых изображений. Целью алгоритма JPEG является сжатие изображений любого размера, в любых цветовых пространствах, при достижении оптимального баланса между коэффициентом компрессии и качеством декодированного изображения. Существует 4 режима работы стандарта JPEG:

1. *Режим без потери качества (Lossless)*. Кодирование осуществляется в один проход, и восстановленное изображение является точной копией оригинала.

2. *Последовательный режим с применением дискретного косинусного преобразования (Sequential)*. Сжатие с потерями осуществляется в один проход с применением дискретного косинусного преобразования. Восстановленное изображение является приближённой копией оригинала.

3. *Прогрессивный режим с применением дискретного косинусного преобразования (Progressive)*. Изображение кодируется и декодируется в несколько проходов. Каждый следующий проход улучшает качество восстановленного изображения по сравнению с предыдущим.

4. *Иерархический режим (Hierarchical)*. Сжатие изображения с несколькими разрешениями для отображения на различных устройствах.

Последние три режима стандарта JPEG являются методами сжатия изображений с потерями качества, поскольку существует ограничение точности вычисления прямого и обратного ДКП, а также искажения, вызванные квантованием коэффициентов. В случае применения последовательного режима без потерь, вместо ДКП используется кодирование, основанное на предсказании, а также отсутствует операция квантования. Иерархический режим использует как кодирование с использованием ДКП, так и метод кодирования, основанный на предсказании. Простейшей формой последовательного режима с применением ДКП является алгоритм JPEG Baseline, использующий в качестве энтропийного кодера алгоритм Хаффмана. Другим вариантом последовательного режима с применением ДКП является алгоритм, основанный на арифметическом кодировании. Алгоритм Baseline JPEG получил наиболее широкое распространение на практике и будет рассмотрен далее более подробно.

В некоторых источниках часто упоминается также алгоритм Motion JPEG для сжатия движущихся изображений. Следует отметить, что данный алгоритм не является отдельным стандартом. Действительно JPEG можно использовать для сжатия последовательности изображений в видеофайлах, основываясь на предположении, что ее можно рассматривать, как последовательность неподвижных кадров, и сжимать кадры независимо друг от друга. Такая процедура сжатия последовательности изображений и получила название – Motion JPEG.

2.1. Создание и развитие стандарта JPEG

Стандарт сжатия JPEG является результатом объединенных усилий международной организации по стандартизации (ISO – International Organization for Standardization), международной электротехнической комиссии (IEC – International Electrotechnical Commission) и международного союза электросвязи (ITU – International Telecommunication Union, прежнее название CCITT). Хронологически развитие JPEG происходило следующим образом (см. табл. 2.1).

Таблица 2.1

Развитие стандарта JPEG

| | |
|-------------|---|
| 1983 | ISO объединяет несколько групп исследователей с целью поиска методов добавления графики фотографического качества к существующим в то время текстовым терминалам. Задачей является создание стандарта сжатия, который передавал бы изображение по цифровой линии со скоростью 64 Кбит/с примерно за 1 с |
| 1986 | Объединение исследовательских групп ISO и CCITT в комитет JPEG |
| 1988 | Выбор экспертами JPEG дискретного косинусного преобразования в качестве основы для будущего стандарта |
| 1992 | Завершение работы над алгоритмом сжатия JPEG |
| 1992 | Инженер американской компании C-Cube Microsystem Эрик Гамильтон (Eric Hamilton) выполнил реализацию алгоритма JPEG в виде спецификации на формат файла JFIF (JPEG File Interchange Format) |
| 1994 | Официальное принятие ISO стандарта JPEG (ISO/IEC 10918-1 ITU-T.Rec.T.81) |

| | |
|-------------|---|
| 1995 | Сертификация нормативных тестов на процедуру кодирования / декодирования JPEG (ISO/IEC 10918-2) |
| 1996 | Первое упоминание о формате файлов SPIFF (Still Picture Interchange File Format) |
| 1997 | Стандарт на расширения для JPEG (ISO/IEC 10918-3) |
| 1999 | Регистрация формата файлов SPIFF как части стандарта JPEG (ISO/IEC 10918-4) |
| 1999 | Сертификация стандарта JPEG-LS для сжатия неподвижных изображений без потерь / практически без потерь (ISO/IEC 14495-1) |
| 2003 | Стандарт на расширения для JPEG-LS (ISO/IEC 14495-2) |
| 2005 | Дополнение к патентной информации относительно стандарта JPEG |

Следует отметить, что уже в начале 1990-х годов существовали алгоритмы, позволяющие осуществлять сжатие изображения с большими коэффициентами компрессии, чем JPEG. Ключевым фактором, учитываемым при разработке стандарта, являлось требование уменьшения вычислительной сложности алгоритма. Простота и сравнительная дешевизна аппаратной реализации JPEG и послужили основой для его широкого распространения.

2.2. Алгоритм сжатия без потерь – JPEG-LS

Некоторые приложения выдвигают специфические требования на качество сжатых изображений. Например, искажение информации недопустимо для сканированных документов, медицинских и некоторых других типов изображений. Этот факт послужил основанием для создания алгоритма JPEG-LS (JPEG Lossless), который не допускает искажений в изображении.

Сжатие алгоритмом JPEG-LS основано на принципах кодирования с предсказанием. Так как для основных типов изображений соседние пиксели обычно коррелированы между собой, следовательно, значительную часть информации о пикселе можно извлечь, основываясь на значениях соседних пикселей A, B и C (рис. 2.1).

Кодирование с предсказанием является простым методом для уменьшения пространственной избыточности. При использовании данного метода значение пикселя определяется, основываясь на

наборе значений уже закодированных соседних пикселей с помощью подходящей модели предсказания.

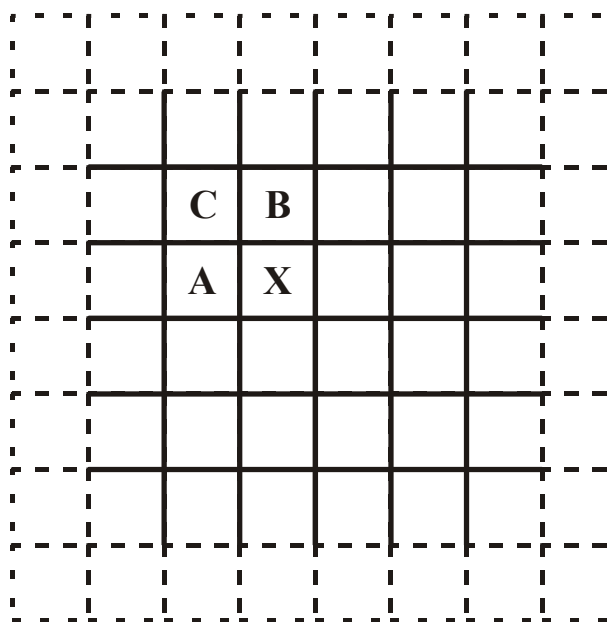


Рис. 2.1. Область соседних пикселей для X

Для идеальной модели предсказанная величина должна в точности совпадать с реальным значением пикселя. В реальных приложениях такая ситуация невозможна. Применяя эффективную модель предсказания, можно определить значение пикселя очень близкое к реальному, и ошибка предсказания будет минимальной. Практические реализации моделей представляют собой линейную комбинацию уже закодированных смежных пикселей. Причина использования уже закодированных пикселей заключается в том, что значения, на которые ссылается кодер и декодер, должны быть уже декодированы на момент восстановления текущего значения. Разность между действительным и предсказанным значениями называется дифференциалом или ошибкой предсказания. Ошибка предсказания кодируется с использованием алгоритма Хаффмана или арифметического кодирования. Общая схема алгоритма JPEG-LS приведена на рис. 2.2.

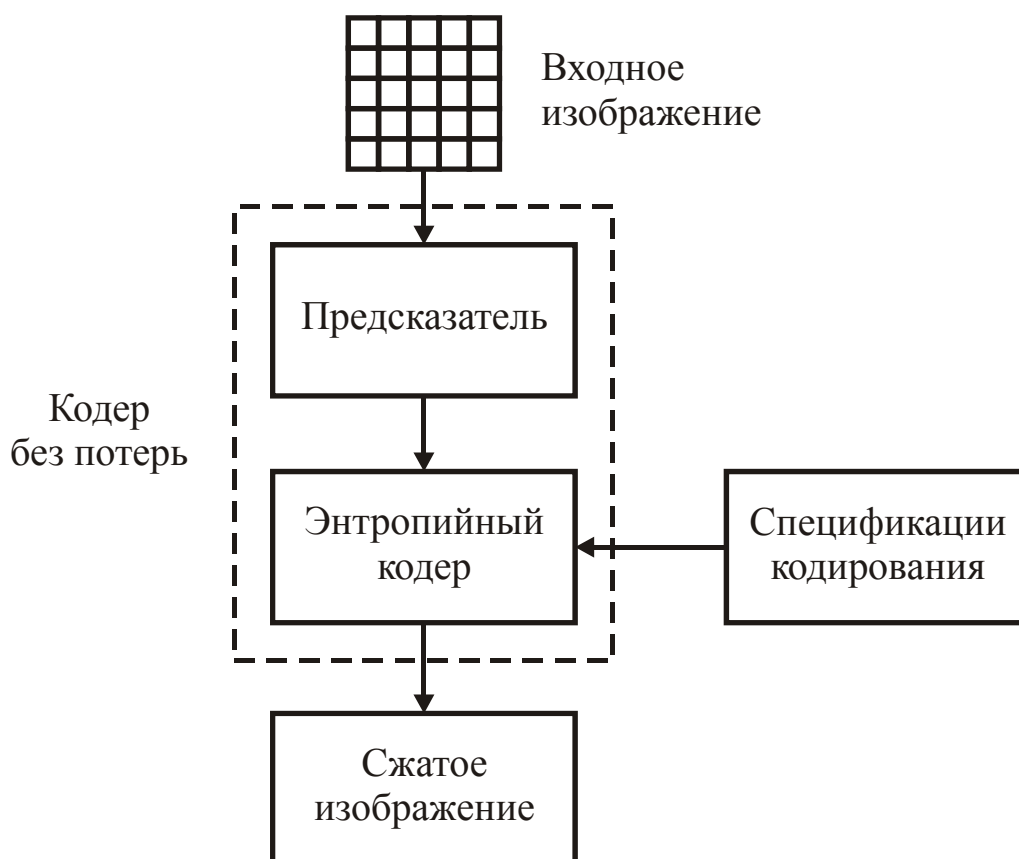


Рис. 2.2. Схема кодера JPEG-LS

2.3. Алгоритм сжатия Baseline JPEG

Алгоритм сжатия Baseline JPEG наиболее часто используется среди всех четырёх режимов стандарта JPEG. Он определён для сжатия полутоновых и полноцветных изображений, представленных в цветовых системах RGB или CMYK. Baseline JPEG позволяет использовать только 8 бит для представления каждой цветовой компоненты. Следует отметить, что в типичном цветном изображении существует значительная пространственная корреляция между цветовыми компонентами. Для достижения наилучшего сжатия корреляция между компонентами цветности уменьшается за счёт преобразования трех компонент RGB изображения в цветовое пространство $YCbCr$. При таком представлении цвета яркостная и цветоразностная компоненты изображения мало коррелированы между собой и содержат значительную часть избыточной информации. Они могут быть сокращены без заметной потери визуального качества восстановленного изображения.

Рассмотрим алгоритм Baseline JPEG более подробно. Основные этапы его работы приведены на рис. 2.3.

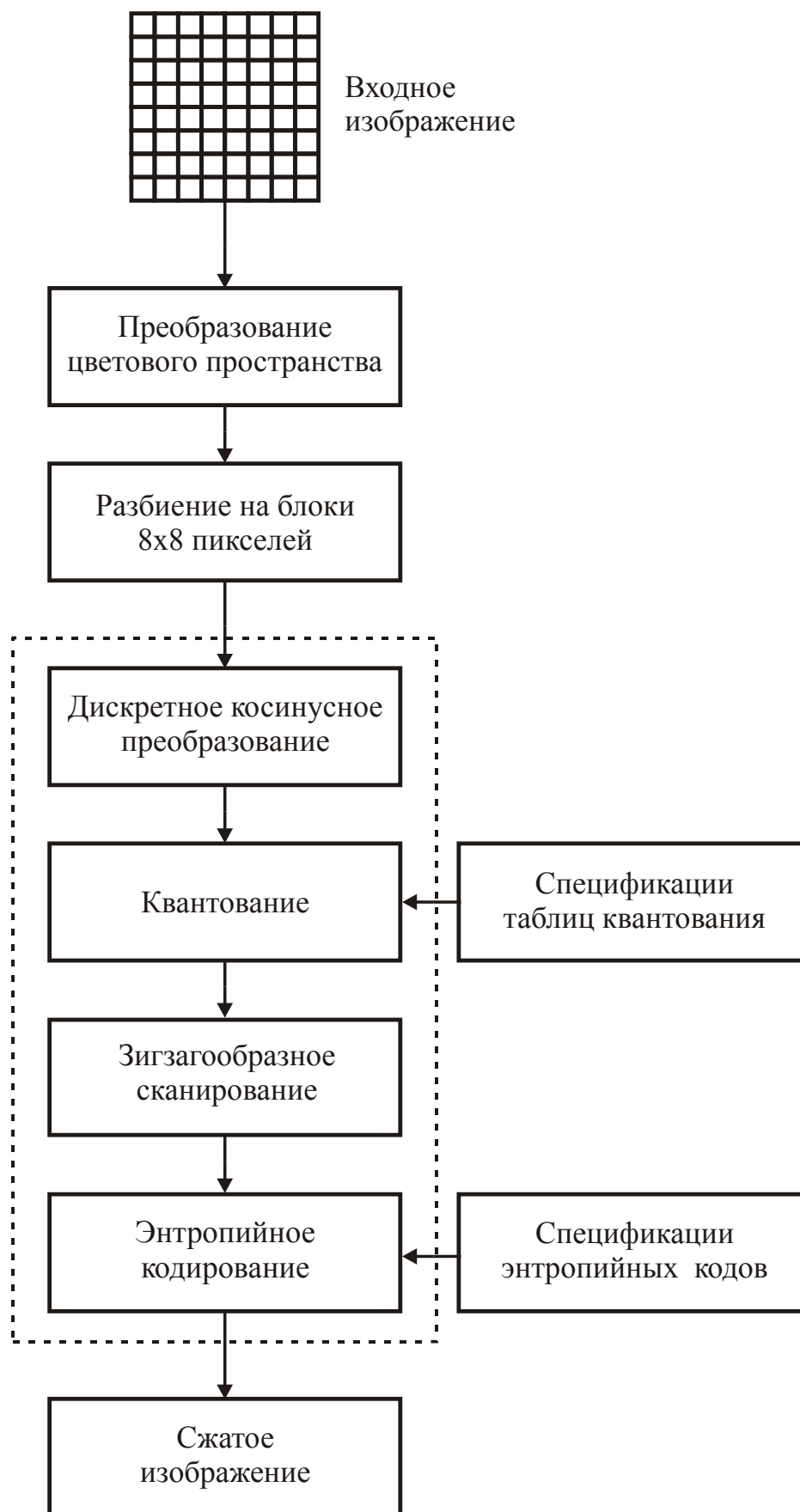


Рис. 2.3. Схема кодирования для алгоритма Baseline JPEG

Пусть сжатию подвергается цветное 24-битное изображение (по 8 бит на каждый из RGB каналов).

Шаг 1. Переводим изображение из цветового пространства RGB в цветовое пространство $YCbCr$. В матричной форме такое преобразование можно представить следующим образом:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,2999 & 0,587 & 0,114 \\ -0,168736 & -0,331264 & 0,5 \\ 0,5 & -0,418688 & -0,081312 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}.$$

Обратное преобразование осуществляется умножением вектора $YCbCr$ на обратную матрицу:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1,402 \\ 1 & -0,34414 & -0,71414 \\ 1 & 1,772 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}.$$

Здесь Y – яркостная составляющая, а Cr , Cb – компоненты, отвечающие за цветность (хроматический красный и хроматический синий). За счет того, что человеческий глаз менее чувствителен к цвету, чем к яркости, появляется возможность выполнить прореживание компонент Cr и Cb по горизонтали и вертикали так, что квадрат из четырех пикселей представляется четырьмя отсчетами яркости, двумя отсчетами Cr и двумя отсчетами Cb (4:2:2). Такое преобразование также используется и в телевизионных системах. На сигналы, отвечающие за цвет, там выделяется более узкая полоса частот.

Шаг 2. Разбиваем исходное изображение на матрицы 8×8 . Формируем из каждой матрицы три рабочие матрицы ДКП – по 8 бит отдельно для каждой компоненты.

Выбор размера блока является важным фактором, влияющим на отклонение восстановленного изображения от оригинала и вычислительную сложность всего алгоритма. Выбор размера 8×8 обусловлен следующими факторами:

- ограничения на вычислительную сложность алгоритма ДКП;
- на интервале в 8 пикселей между пикселями существенна корреляция;
- при данном размере блока достигается приемлемое значение среднеквадратичной ошибки (СКО) для восстановленного блока, что видно из зависимости на рис. 2.4.

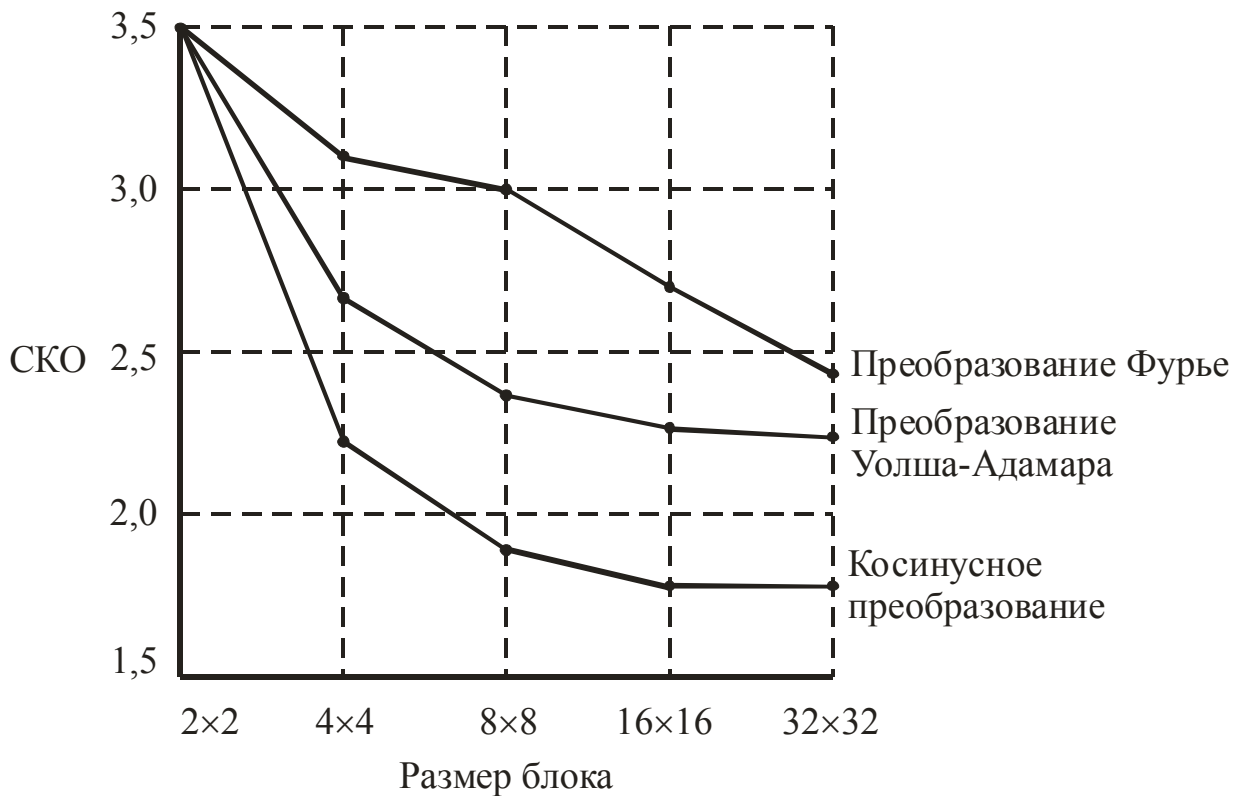


Рис. 2.4. Зависимость нормированной СКО восстановленного изображения от размера блока разложения для трех дискретных преобразований

Для получения больших коэффициентов сжатия выполнение этого шага модифицируется. Изображение делится по компоненте Y , как и в первом случае, а для компонент Cr и Cb матрицы набираются через строчку и через столбец, т. е. из исходной матрицы размером 16×16 получается только одна рабочая матрица ДКП. При этом теряется $3/4$ полезной информации о цветовых составляющих изображения. На результирующем RGB изображении, как показала практика, эта сказывается незначительно.

Шаг 3. Применяем ДКП к каждой рабочей матрице. Для блока размером 8×8 оно вычисляется так:

$$F(u, v) = \frac{1}{4} c(u) c(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \times \\ \times \cos \left[\frac{\pi(2x+1)u}{16} \right] \cos \left[\frac{\pi(2y+1)v}{16} \right],$$

где $u, v = 0, 1, \dots, 7$, а

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k \neq 0 \end{cases}.$$

Получаем матрицу, в которой коэффициенты в левом верхнем углу соответствуют низкочастотной составляющей изображения, а в правом нижнем – высокочастотной.

Шаг 4. Производим квантование в соответствии с заданной матрицей квантования. Рабочая матрица делится на матрицу квантования поэлементно. Получившиеся результаты округляются (Round). Для каждой компоненты (Y , Cb и Cr), в общем случае, задается своя матрица квантования $q[u, v]$

$$Yq[u, v] = Round\left(\frac{Y[u, v]}{q[u, v]}\right).$$

На этом шаге осуществляется управление степенью сжатия и происходят самые большие потери. Задавая матрицы квантования с большими коэффициентами, мы получим больше нулей и, следовательно, большую степень сжатия.

Стандарт Baseline JPEG не фиксирует матрицу квантования. Разработчик может задавать ее по своему усмотрению. Однако стандарт содержит матрицы квантования, построенные на основе психовизуальных экспериментов, которые определены как рекомендации, а не как обязательные требования. Рекомендованные матрицы квантования представлены в Приложении К стандарта Baseline JPEG. Ниже представлены рекомендованные матрицы квантования для яркостной компоненты (2.1) и цветовой компоненты (2.2):

$$\begin{matrix} 16 & 11 & 10 & 16 & 2 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 108 & 99 \end{matrix}, \quad (2.1)$$

$$\begin{array}{cccccccc}
 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\
 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\
 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\
 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\
 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99
 \end{array} \quad (2.2)$$

Эти матрицы квантования обеспечивают хорошие результаты для большинства реальных изображений, заданных с точностью 8 бит для компонент яркости и цветности.

С этапом квантования связаны и специфические эффекты алгоритма Baseline JPEG. При больших значениях коэффициента компрессии потери в низких частотах могут быть настолько велики, что изображение распадется на визуально заметные квадратные области 8×8 . Потери в высоких частотах могут проявиться и в так называемом «эффекте Гиббса», когда вокруг контуров с резким переходом цвета образуется своеобразный «нимб».

Шаг 5. Переводим матрицу 8×8 в 64-элементный вектор при помощи «зигзаг»-сканирования (рис. 2.5), т. е. берем элементы с индексами (0,0), (0,1), (1,0), (2,0)...

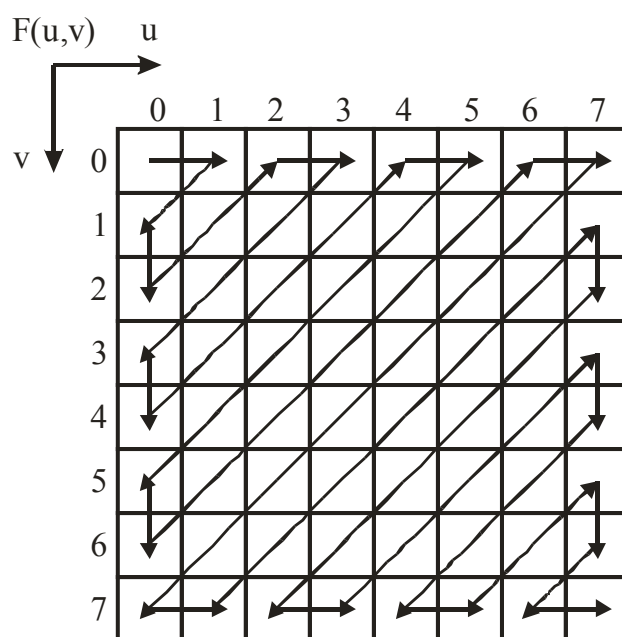


Рис. 2.5. «Зигзаг»-сканирование

Таким образом, в начале вектора получаются коэффициенты матрицы, соответствующие низким частотам, а в конце – высоким.

Шаг 6. Проводим энтропийное кодирование информации. Свертываем полученный вектор с помощью алгоритма группового кодирования. При этом получаем пары типа («пропустить», «число»), где «пропустить» является счетчиком пропускаемых нулей, а «число» – значение, которое необходимо поставить в следующую ячейку. Так, вектор 42 3 0 0 0 -2 0 0 0 0 1 ... будет свернут в пары (0,42) (0,3) (3,-2) (4,1) ...

Свертываем получившиеся пары кодированием по Хаффману с фиксированной таблицей. Постоянные коэффициенты кодируются отдельно, используя схему предсказания.

Процесс восстановления изображения в алгоритме Baseline JPEG полностью симметричен и приведен на рис. 2.6.

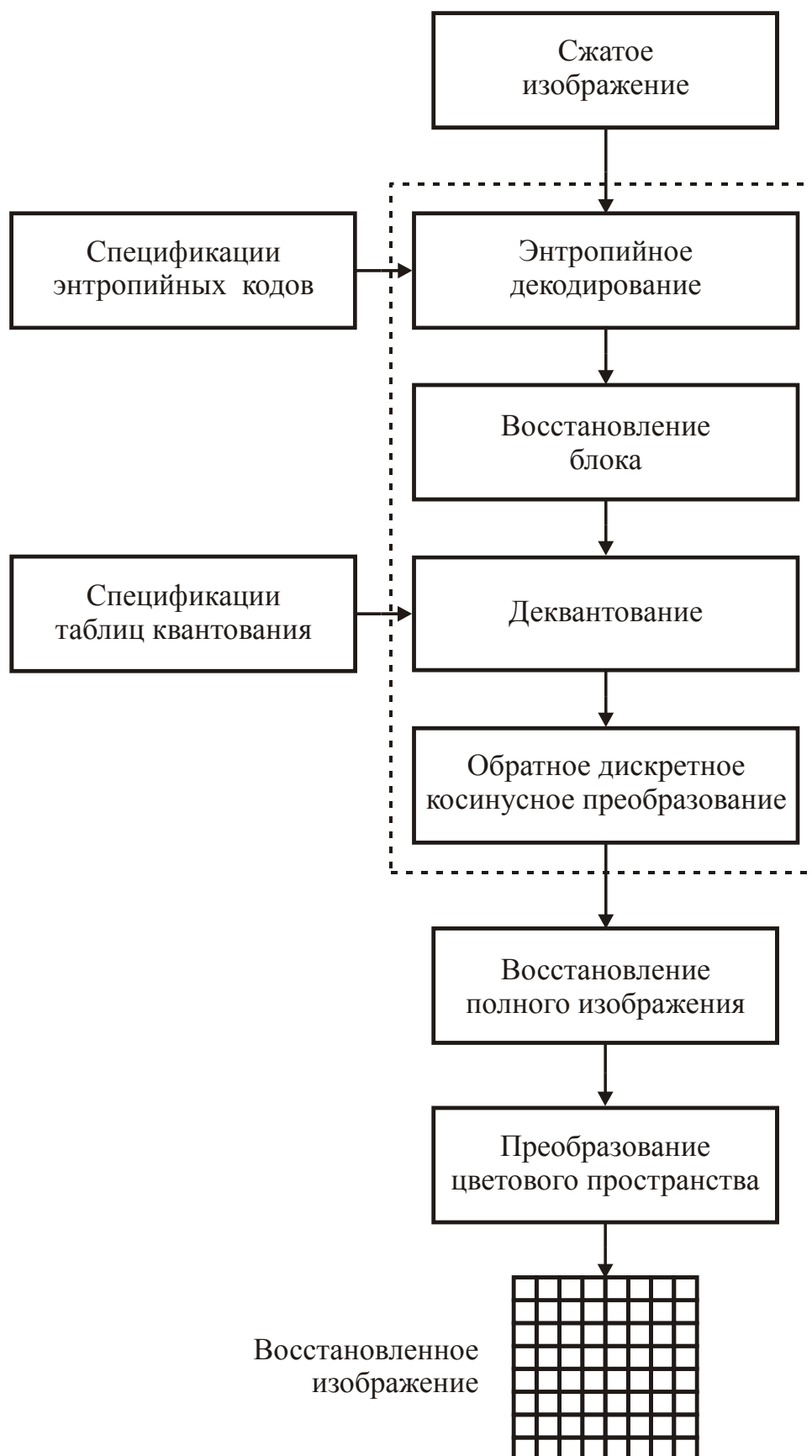


Рис. 2.6. Схема декодирования для алгоритма Baseline JPEG

2.4. Основные характеристики стандарта JPEG

Существенными положительными качествами алгоритма JPEG являются:

- возможность гибкого задания коэффициента сжатия изображения;
- широкая поддержка стандарта во всевозможных графических и других приложениях;
- эффективная реализация стандарта на интегральных микросхемах, что обусловило его широкое распространение в цифровой технике.

Стандарт JPEG предусматривает ряд специальных режимов сжатия. **Режим без потерь** (JPEG-LS) предназначен для применения в случаях, когда требуется идентичное восстановление сжатого изображения. В этом режиме величина каждого пикселя предсказывается в виде отличий от соседних с ним пикселей. Эта величина кодируется с использованием кодов переменной длины. Не используется субдискретизация и преобразование цветов. JPEG-LS, как и любой другой алгоритм сжатия без потерь, обеспечивает существенно худшее сжатие, чем соответствующие алгоритмы с потерями. При кодировании реальных изображений он может обеспечить коэффициент сжатия порядка двух.

Прогрессивный режим может быть использован при передаче изображения по каналу связи с низкой пропускной способностью. В этом случае пользователь может прервать оказавшуюся ненужной передачу данных. Различают два прогрессивных метода:

- режим спектрального выбора, при котором сначала пересылаются компоненты нижних частот каждого из блоков, а затем компоненты более высоких частот (пространственная масштабируемость),
- режим последовательной аппроксимации, при котором сначала пересылается некоторое число старших бит каждой закодированной величины, а затем ее младших бит.

В **иерархическом режиме** изображение хранится в нескольких версиях с различной разрешающей способностью. Например, изображение размером 1000×1000 пикселей может храниться в виде версий размером 10×10 , 100×100 и в полном разрешении. Каждое изображение при этом сохраняет отличия от предыдущей «уменьшенной» версии. Этот режим удобен для быстрого просмотра изображений.

Отрицательной стороной алгоритма JPEG является заметность на восстановленном изображении блочной структуры (8×8) при

повышении степени сжатия изображения. Это связано с тем, что ДКП – блочное преобразование, поэтому проявляется эффект Гиббса – ореолы по границам резких переходов цветов. Этот эффект иллюстрируется на рис. 2.7.



Рис. 2.7. Демонстрация блочной структуры стандарта JPEG:
а) исходное изображение; б) изображение, сжатое с коэффициентом 40

Не очень приятным свойством JPEG является 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 этих сюрпризов JPEG не рекомендуется активно использовать в полиграфии, задавая высокие коэффициенты сжатия. Однако при сжатии изображений, предназначенных для просмотра человеком, он на данный момент незаменим.

3. СТАНДАРТ JPEG2000

JPEG2000 относительно новый стандарт для сжатия изображений, разработанный совместно Международной организацией по сертификации (ISO) и Международной электротехнической комиссией (IEC).

Несмотря на широкую популярность стандарта JPEG в последние два десятилетия, развивающиеся интерактивные мультимедиа приложения в проводных и беспроводных коммуникациях, а также интернет-приложения, потребовали новых возможностей, отсутствующих в JPEG. Принципиальный сдвиг в развитии стандартов сжатия изображений произошел с внедрением в соответствующие алгоритмы кодирования дискретного вейвлет-преобразования (рис. 3.1).

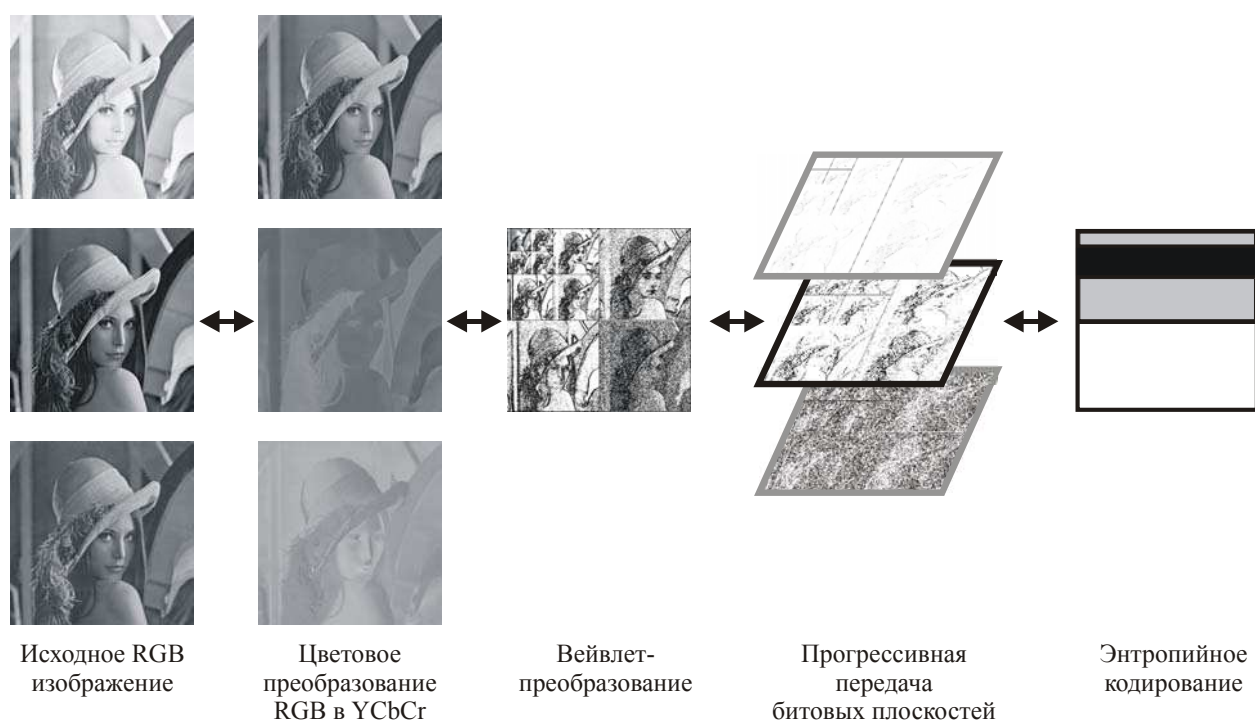


Рис. 3.1. Сжатие изображений с использованием вейвлет-преобразования

Стандарт JPEG2000 показывает свою эффективность в широком спектре приложений: передача изображений в сети Интернет, цифровая фотография, создание цифровых библиотек, базы данных изображений, цветное копирование, сканирование, печать, компьютерная графика, обработка медицинских изображений, мобильная связь третьего поколения (3G), приложения клиент-сервер, электронная коммерция и др.

3.1. Части стандарта JPEG2000

Существующие на сегодня 13 частей алгоритма (часть 7 в процессе развития была отклонена) расширяют основной стандарт JPEG2000 (ISO/IEC 15444-1), описываемый в части 1.

Часть 1. Основная система кодирования (Core Coding System) – определяет базовый набор свойств и синтаксис потокового кода в JPEG2000.

Часть 2. Расширения (Extensions) – добавляет новые свойства в основную систему кодирования.

Часть 3. Стандарт JPEG2000 для движущихся изображений (Motion JPEG2000) – определяет файловый формат MJ2, содержащий последовательность изображений, закодированных в соответствии с основной системой кодирования и использующийся для сжатия видео. Итоговая реализация не является столь эффективной как алгоритмы на основе спецификации MPEG, но предоставляет высококачественный, редактируемый файловый формат, согласующийся со стандартами MPEG4 и QuickTime.

Часть 4. Требования на тестирование (Conformance Testing) – определяет процедуру тестирования для кодирования / декодирования с использованием части 1 стандарта JPEG2000.

Часть 5. Программное обеспечение (Reference Software) – предоставляет разработчикам две программных реализации (на языках Java и C) для тестирования и проверки систем на основе JPEG2000.

Часть 6. Комплексный формат файлов (Compound Image File Format) – определяет файловый формат JPM для компрессии изображений, использующих многослойный смешанный растровый контент согласно спецификации ITU-T. T.44 | ISO 16485.

Часть 7. Отклонена в процессе развития стандарта.

Часть 8. Защищенный JPEG2000 (Secure JPEG2000 – JPSEC) – прописывает аспекты безопасности для фрагментов изображения JPEG2000, используемых в таких приложениях, как криптография, электронная цифровая подпись и др.

Часть 9. Интерактивные документы (Interactivity Tools APIs and Protocols – JPIP) – определяет стандартный механизм взаимодействия и правила эффективного обмена информацией с изображениями стандарта JPEG2000.

Часть 10. 3D и вычисления с плавающей точкой (3D and Floating Point Data – JP3D) – эта часть разработана для трехмерных приложений, например, для восстановления медицинских изображений.

Часть 11. Беспроводной обмен (Wireless – JPWL) – разработана для беспроводных мультимедиа приложений. Основными задачами JPWL являются: защита от ошибок, их обнаружение и коррекция в зашумленной беспроводной среде.

Часть 12. Медиа файл формат (ISO Base Media File Format) – содержит общую часть спецификаций MPEG-4 и JPEG2000.

Часть 13. Спецификация кодера. Представляет канонический пример реализации кодера JPEG2000 (часть 1 описывает только декодер и стандарт кодирования битового потока).

Также существуют усовершенствования (Amendment, Technical Corrigendum) к частям 1 и 2 стандарта JPEG2000. В них описаны дополнительные ограничения на битовый поток, улучшенный алгоритм квантования, а также нюансы работы с различными цветовыми пространствами.

3.2. Создание и развитие стандарта JPEG2000

Впервые использование вейвлет-функций к задачам сжатия изображений было предложено Маллом (Mallat) и Добеши (Daubechies) в 1980-х годах. Хронология развития алгоритмов сжатия на основе вейвлет-преобразования и развитие стандарта JPEG2000 приводится в табл. 3.1.

Таблица 3.1

Развитие стандарта JPEG2000

| | |
|------------------|---|
| 1991 | Шапиро (Shapiro) предложил схему вейвлет-кодирования на основе вложенных нуль-деревьев (EZW – Embedded zerotree wavelet) |
| 1993 | Саид (Said) и Перлман (Pearlman) предложили алгоритм разложения по иерархическим деревьям (SPIHT – Set partitioning in hierarchical trees), значительно превосходящий EZW |
| Май 1994 | Была предложена схема сжатия на обратимых вложенных вейвлетах (CREW – Compression with reversible embedded wavelets), содержащая свойства будущего стандарта JPEG2000 |
| Июль 1995 | Была предложена схема CREW для стандарта JPEG-LS |

| | |
|--------------------|---|
| Март 1996 | Компания Ricoh внесла предложения о новом стандарте сжатия изображений для сертификации в ISO |
| Ноябрь 1996 | Предложение Ricoh было одобрено ISO. Наибольший интерес вызвало понижение битового потока при заданном качестве изображения, а не новые дополнительные свойства алгоритма сжатия |
| Март 1997 | Объявление конкурса на разработку стандарта. Основным критерий отбора победителя – минимизация битового потока |
| Ноябрь 1997 | Рассмотрение 24 поданных заявок. Алгоритм CREW показал лучшие результаты в области сжатия без потери информации и 4–6 место по битовому потоку. В качестве верификационной модели была выбрана схема решетчатого кодирования с квантованием (TCQ – Trellis coded quantization), предложенная университетом Аризоны |
| Март 1998 | Неудачей закончилась попытка модификации ДКП с целью понижения размера битового потока / уменьшения вычислительной сложности |
| Июль 1998 | Были приняты расширения стандарта, предложенные в схеме CREW |
| Ноябрь 1998 | Таубман (Taubman) представил алгоритм вложенного блочного кодирования с оптимизированным усечением (EBCOT - Embedded block coding with optimized truncation). Идея этого метода заключается в разделении каждого частотного диапазона на блоки коэффициентов и выполнении кодирования независимо над каждым из блоков. Этот метод обеспечивает необходимую степень произвольного доступа к потоку бит. Закодированные блоки объединяются в «пакеты», которые вместе образуют «слои качества». Недостатком кодера является размещение в каждом «слое» подобласти бит, определяемых как наиболее значимые. Этот способ нацелен на минимизацию СКО в каждой точке потока бит |
| Март 1999 | Алгоритм EBCOT после длительных обсуждений был включен в верификационную модель |

| | |
|----------------------|---|
| Июль 1999 | Предложен алгоритм Motion JPEG2000 |
| Октябрь 1999 | Выпуск первой черновой версии стандарта |
| Декабрь 1999 | Части 1–5 стандарта JPEG2000 направлены для стандартизации |
| 2 января 2001 | JPEG2000 официально становится международным стандартом ISO |

3.3. Основы кодирования в JPEG2000

Как и JPEG, стандарт JPEG2000 описывается с точки зрения операции декодирования изображений. Это означает, что декодер определён достаточно точно, начиная с маркерных сегментов и заканчивая синтаксисом потока бит в документе стандарта JPEG2000. Детальное описание декодера определяется функциональными возможностями кодера. Тем не менее правильно понять и объяснить разработанные по стандарту документы может только специалист в области цифровой обработки документов. Для того чтобы понять систему декодирования, описанную в документе стандарта, необходимо хорошо изучить алгоритм кодирования. Процедура сжатия разделена на три этапа: предварительная обработка изображения, сжатие и формирование сжатого потока бит. Рассмотрим их более подробно.

3.3.1. Предварительная обработка изображения

Предварительная обработка изображения состоит из трёх основных этапов: разбиение на тайлы (блоки), сдвиг по яркости и последующее преобразование компонент.

Разбиение на тайлы. На этом шаге обрабатываемое изображение делится на прямоугольные неперекрывающиеся блоки – тайлы. Размер всех тайлов одинаков, исключая те, которые находятся на границе изображения (в случае некратности числа пикселей изображения числу пикселей в тайле). Размер тайла может быть произвольным и может принимать значение размера изображения. Если изображение состоит из нескольких цветовых компонент, каждый тайл также состоит из нескольких компонент. Для изображений в градациях серого тайл состоит только из одной

компоненты. После разбиения каждый тайл обрабатывается независимо. В случае сжатия изображения с большими коэффициентами становятся заметными искажения, возникающие на границах тайлов. Подобные искажения существуют для любого алгоритма блочного кодирования. Чем меньше размеры тайла, тем больше искажения на границах и меньше эффективность сжатия. Очевидно, что полное отсутствие деления на тайлы обеспечивает наилучшее визуальное качество. С другой стороны, если размер тайла слишком большой, потребуются большие буферы памяти для выполнения арифметических операций как при программной, так и при аппаратной реализации JPEG2000.

Сдвиг по яркости. Изначально пиксели изображения хранятся как целые величины без знака. Целью сдвига по яркости является преобразование динамического диапазона таким образом, чтобы значения пикселей были центрированы вблизи нуля. Все значения яркости i -ой компоненты $I_i(x, y)$ преобразуются путём вычитания величины $2^{S_{siz}^i - 1}$, в итоге получаем новые значения i -ой компоненты $I'_i(x, y)$.

$$I'_i(x, y) \leftarrow I_i(x, y) - 2^{S_{siz}^i - 1},$$

где S_{siz}^i – точность представления выборок, которая указывается с помощью маркера SIZ (изображение и размер тайла) в сжатом потоке бит. Для некоторых изображений динамический диапазон уже центрирован вблизи нуля, например значения пикселей компьютерных томограмм, поэтому в этом случае никаких преобразований яркости не требуется.

Преобразование компонент. Преобразование компонент эффективно используется для уменьшения корреляции между компонентами в случае цветного изображения. Оно приводит к уменьшению избыточности и в итоге к увеличению коэффициента сжатия. Фактически стандарт JPEG2000 не рассматривает компоненты как цветовые области, и с этой точки зрения стандарт не различает цвета. Тем не менее он определяет преобразование трёх компонент цветовых областей пространства (RGB).

Первая часть стандарта JPEG2000 поддерживает два различных преобразования: обратимое преобразование компонент (Reversible Color Transform, RCT) и необратимое преобразование компонент (Irreversible Color Transform, ICT).

Обратимое преобразование компонент. Для сжатия без потерь необходимо использовать RCT, так как в этом случае значения пикселей должны быть восстановлены абсолютно точно. Несмотря на то что данное преобразование было разработано для сжатия без потерь, его можно использовать и для сжатия с потерями. В случае сжатия с потерями ошибки появляются за счёт основного преобразования или на этапе квантования, но не за счёт RCT. Формулы для расчёта прямого и обратного RCT выглядят следующим образом:

Прямое RCT:

$$\left. \begin{aligned} Y_r &= \left[\frac{R + 2G + B}{4} \right] \\ U_r &= B - G \\ V_r &= R - G \end{aligned} \right\}.$$

Обратное RCT:

$$\left. \begin{aligned} G &= Y_r - \left[\frac{U_r + V_r}{4} \right] \\ R &= V_r + G \\ B &= U_r + G \end{aligned} \right\}.$$

Необратимое преобразование компонент. Данный тип преобразования аналогичен разложению на компоненты яркости и цветности, используемому в алгоритме Baseline JPEG. ICT применяется только для сжатия с потерями. Компонента Y является яркостной компонентой, а в компонентах Cb и Cr заложена информация о цвете. В алгоритме Baseline JPEG возможно уменьшение числа бит, необходимых для представления компонент цветности. Для стандарта JPEG2000 такое не предусмотрено.

3.3.2. Сжатие

После этапа предварительной обработки следует этап сжатия. Схема его работы показана на рис. 3.2. Поток сжатых данных приведен на рис. 3.3. Заметим, что каждая компонента после предварительной обработки сжимается и передается независимо.

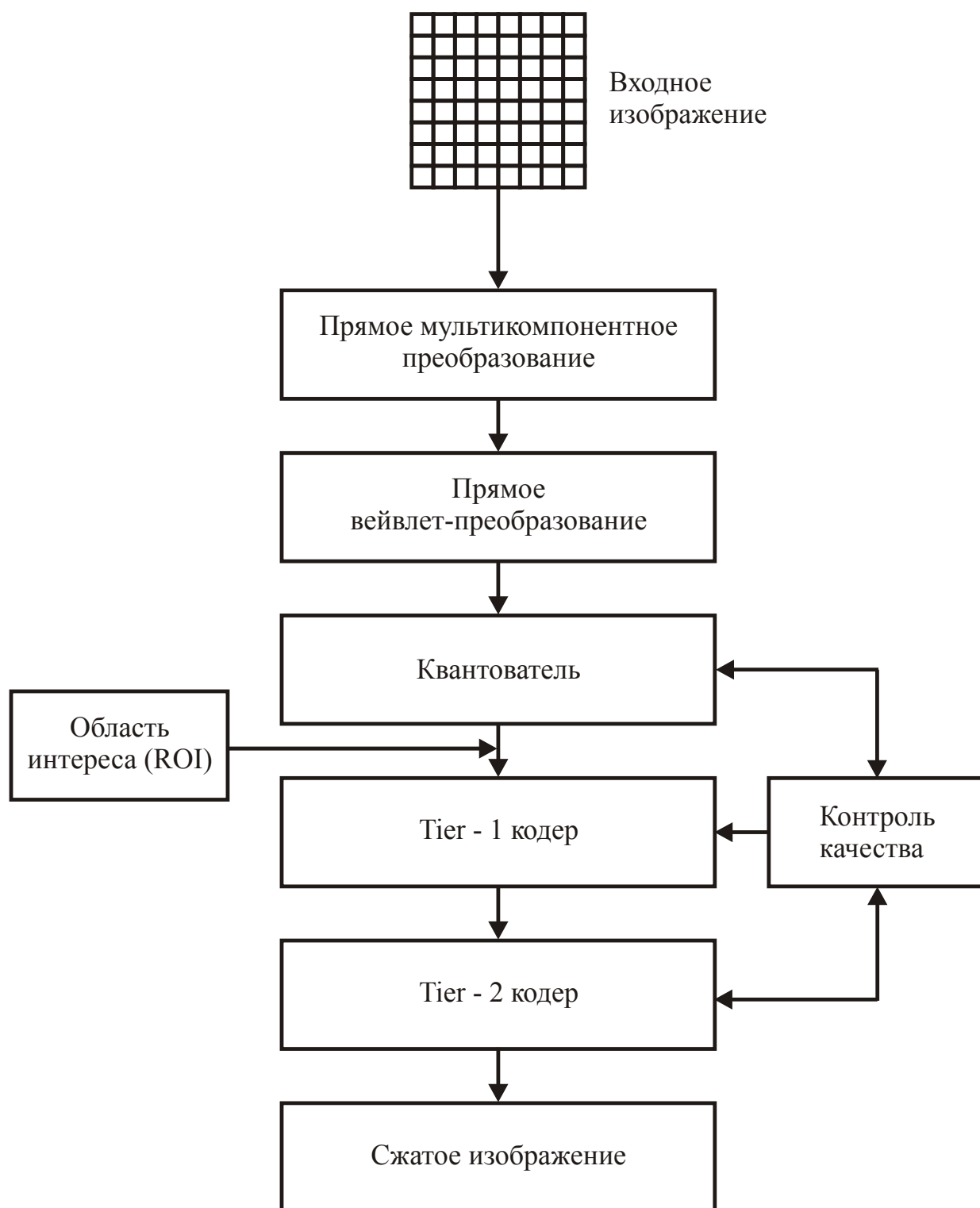


Рис. 3.2. Блок-схема этапа сжатия JPEG2000

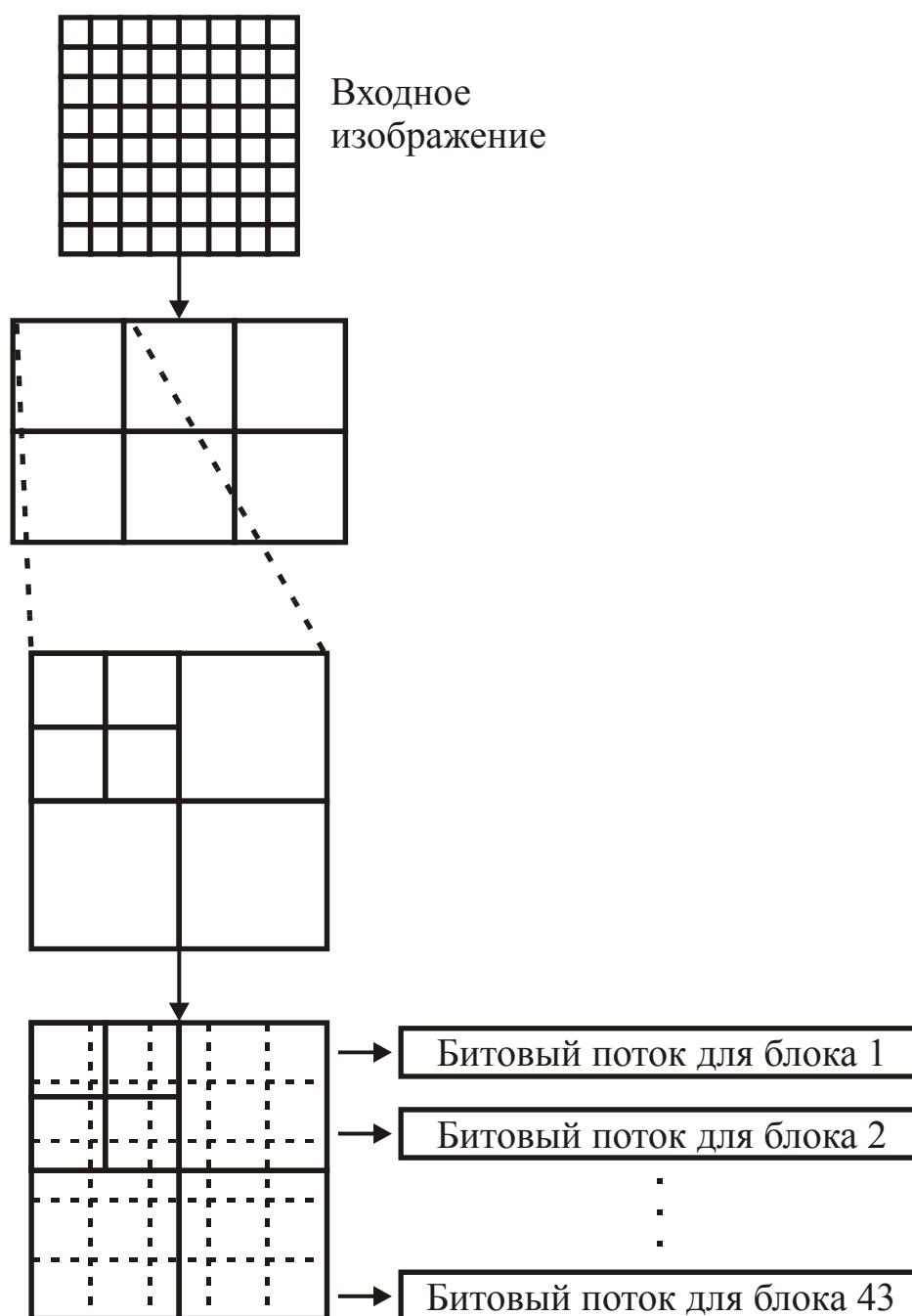


Рис. 3.3. Поток сжатых данных

Этап сжатия включает в себя:

1. Дискретное вейвлет-преобразование.
2. Квантование.
3. Энтропийное кодирование.

В начале каждая компонента преобразуется с помощью подходящего типа дискретного вейвлет-преобразования. Оно непосредственно разлагает каждую компоненту на множество подуровней различного разрешения. Каждый подуровень в случае сжатия с потерями квантуется независимо со своими значениями

коэффициентов квантования. После этого подуровень делится на множество кодовых блоков меньшего размера. Эти кодовые блоки должны иметь одинаковый размер, исключая те блоки, которые находятся на границе. Обычно размер блока составляет 32×32 или 64×64 . После этого к каждому кодовому блоку применяется энтропийное кодирование для получения сжатого потока бит, как показано на рис. 3.3.

Дискретное вейвлет-преобразование. Максимальное число уровней разложения согласно части 1 стандарта JPEG2000 равно 32. В части 1 также отмечено, что для разложения на несколько уровней разрешено использование только преобразований с количеством элементов равным степеням двойки.

Дискретное вейвлет-преобразование для сжатия с потерями. Для сжатия с потерями в стандарте JPEG2000 рекомендовано использование биортогонального вейвлет-фильтра Добеши (9, 7). Под обозначением (9, 7) понимается, что фильтр анализа (разложения) состоит из фильтра нижних частот (ФНЧ) с 9-ю коэффициентами и фильтра верхних частот (АВЧ) с 7-ю коэффициентами. Оба фильтра симметричны. Коэффициенты фильтра анализа (для прямого преобразования) могут быть записаны следующим образом:

для ФНЧ – $[h_{-4}, h_{-3}, h_{-2}, h_{-1}, h_0, h_1, h_2, h_3, h_4]$, где

$$h_4 = h_{-4} = +0,026748757410810;$$

$$h_3 = h_{-3} = -0,016864118442875;$$

$$h_2 = h_{-2} = -0,078223266528988;$$

$$h_1 = h_{-1} = +0,266864118442872;$$

$$h_0 = +0,602949018236358.$$

для АВЧ – $[g_{-3}, g_{-2}, g_{-1}, g_0, g_1, g_2, g_3]$, где

$$g_3 = g_{-3} = +0,0912717631142495;$$

$$g_2 = g_{-2} = -0,057543526228500;$$

$$g_1 = g_{-1} = -0,591271763114247;$$

$$g_0 = +1,115087052456994.$$

Для обратного преобразования используется пара фильтров, называемая фильтрами синтеза. В этой паре ФНЧ имеет 7 коэффициентов, а ФВЧ – 9 коэффициентов.

Коэффициенты ФНЧ можно записать в виде $[h'_{-3}, h'_{-2}, h'_{-1}, h'_0, h'_1, h'_2, h'_3]$, где

$$h'_3 = h'_{-3} = +0,0912717631142495;$$

$$h'_2 = h'_{-2} = -0,057543526228500;$$

$$h'_1 = h'_{-1} = -0,591271763114247;$$

$$h'_0 = +1,115087052456994.$$

Коэффициенты ФВЧ можно записать в виде $[g'_{-4}, g'_{-3}, g'_{-2}, g'_{-1}, g'_0, g'_1, g'_2, g'_3, g'_4]$, где

$$g'_4 = g'_{-4} = +0,026748757410810;$$

$$g'_3 = g'_{-3} = -0,016864118442875;$$

$$g'_2 = g'_{-2} = -0,078223266528988;$$

$$g'_1 = g'_{-1} = +0,266864118442872;$$

$$g'_0 = +0,602949018236358.$$

Обратимое вейвлет-преобразование для сжатия без потерь. Для сжатия без потерь в стандарте JPEG2000 по умолчанию используется вейвлет-фильтр Ле Галла (5, 3). Хотя в основном данный тип фильтров применяется для сжатия без потерь, возможно его использование и для сжатия с потерями. Однако экспериментально было показано, что фильтр Добеши (9, 7)

позволяет получать лучшие результаты в режиме с потерями. Вейвлет-фильтр Ле Галла (5, 3) имеет два фильтра анализа.

ФНЧ $[h_{-2}, h_{-1}, h_0, h_1, h_2]$, где

$$h_2 = h_{-2} = -1/8;$$

$$h_1 = h_{-1} = -1/4;$$

$$h_0 = 3/4.$$

ФВЧ $[g_{-1}, g_0, g_1]$, где

$$g_1 = g_{-1} = -1/2;$$

$$g_0 = 1.$$

Соответствующие фильтры синтеза имеют коэффициенты: ФНЧ $[h'_{-1}, h'_0, h'_1]$, где

$$h'_1 = h'_{-1} = 1/2;$$

$$h'_0 = 1.$$

ФВЧ $[g'_{-2}, g'_{-1}, g'_0, g'_1, g'_2]$, где

$$g'_2 = g'_{-2} = -1/8;$$

$$g'_1 = g'_{-1} = -1/4;$$

$$g'_0 = 3/4.$$

Квантование. После дискретного вейвлет-преобразования, при использовании режима сжатия с потерями все подуровни квантуются. На этом этапе теряется наибольшее количество информации при кодировании. Грубое квантование, то есть квантование с большими коэффициентами, приводит к большему сжатию и, следовательно, к уменьшению точности восстановления изображения, так как происходит потеря значительной части информации. Этап квантования не допускается в случае сжатия без потерь. В части 1

стандарта JPEG2000 описан способ однородного скалярного квантования с мёртвой зоной вблизи нуля. Спецификация определяет различные шаги квантования для каждого подуровня разложения. Шаг квантования Δ_b для подуровня b вычисляется на основе динамического диапазона величин подуровня. Формула однородного скалярного квантования с мёртвой зоной имеет вид

$$q_b(i, j) = \text{sign}(y_b(i, j)) \left\lceil \frac{|y_b(i, j)|}{\Delta_b} \right\rceil,$$

где $y_b(i, j)$ – коэффициент дискретного вейвлет-преобразования для подуровня b и Δ_b – шаг квантования для подуровня b . Все полученные коэффициенты дискретного вейвлет-преобразования $q_b(i, j)$ являются целыми числами со знаком.

Все вычислительные операции до этапа квантования включительно выполняются над данными, представленными в виде двумерных массивов. Далее данные преобразуются в вид, удобный для этапа энтропийного кодирования.

Контроль качества. Несмотря на то что ключевые этапы алгоритма, такие как вейвлет-преобразование, квантование и энтропийное кодирование, строго определены, некоторые части алгоритма доступны для модификации разработчиком кодера. Одной из таких «степеней свободы» является контроль качества. Контроль качества определяет оптимальную битовую скорость по всем кодовым блокам и подуровням для достижения минимальной среднеквадратичной ошибки.

Контроль битовой скорости полностью зависит от кодера и может быть реализован по-разному. Программист может сам определять, как он будет оптимизировать соотношение скорости.

Энтропийное кодирование. Данные, представляющие собой квантованные вейвлет-коэффициенты каждого блока каждого подуровня, сжимаются с помощью энтропийного кодирования. Алгоритм энтропийного кодирования состоит из двух этапов: Tier-1 и Tier-2. Далее дается их краткое описание. В следующем разделе алгоритмы кодирования описываются более подробно.

Tier-1. Блоки кодируются независимо друг от друга. Если точность представления элемента кодового блока равна p , тогда блок представляется в виде p битовых плоскостей, и эти плоскости кодируются, начиная с плоскости, отвечающей за наиболее значимые биты, и заканчивая плоскостью, отвечающей за наименее значимые биты. Каждая битовая плоскость сначала преобразуется с

использованием механизма кодирования битовых плоскостей (BPC – fractional bit-plane coding) для получения промежуточных данных. В стандарте JPEG2000 для этого используется алгоритм EBCOT. Он кодирует битовые плоскости в три прохода таким образом, что данные, закодированные в разных проходах, не перекрываются.

Двоичная величина, определяемая алгоритмом EBCOT, кодируется с использованием арифметического кодирования для получения сжатого кода каждого блока. В этом случае используется разновидность арифметического кодирования, известная как MQ-кодер. Данный тип кодера также применяется в стандарте JBIG2 для сжатия бинарных изображений. Данные, определяемые алгоритмом EBCOT, используются для оценки вероятности, на основе которой MQ-кодер генерирует код.

Tier-2. На данном этапе происходит кодирование данных и формирование битового потока. Целью Tier-2 является эффективное представление слоя данных и добавочной информации для каждого кодового блока. Данные представляют собой последовательно закодированные битовые плоскости каждого тайла. Блок добавочной информации содержит длину кодовых слов, наиболее значимую битовую плоскость, а также точки разделения потоков бит.

3.4. Характеристики стандарта JPEG2000

К основным характеристикам JPEG2000 относятся:

- достижение высокого качества сжатых изображений. Преимущество над стандартом JPEG подтверждается как визуальными, так и численными оценками (например, пиковым отношением сигнал / шум – ПОСШ) и особенно заметно при больших коэффициентах сжатия (30 и выше). Для сжатых изображений, одинаковых с точки зрения субъективного восприятия, JPEG2000 позволяет достичь большей степени компрессии, чем JPEG;

- возможность использования для сжатия / восстановления различных типов изображений: цветных, полутоновых и бинарных (алгоритм JBIG2);

- использование большого динамического диапазона представления пикселей. JPEG2000 может работать с изображениями с различной точностью задания каждой из цветовых компонент. Несмотря на то что в современных приложениях этот диапазон колеблется от 1 до 16 бит, синтаксис стандарта поддерживает его расширение до 38 бит;

– поддержка изображений большого разрешения. Максимальный размер изображения в JPEG2000 ограничен числом $(2^{32}-1) \times (2^{32}-1)$ пикселей, а максимальное число цветовых каналов – 2^{14} .

– возможность сжатия изображений без потери качества, что позволяет, не меняя алгоритм обработки, использовать JPEG2000 в различных приложениях, таких как: архивация медицинских изображений или передача цифровой информации по каналам связи;

– фиксация размера сжатого изображения. JPEG2000 позволяет пользователю точно выбрать желаемый размер сжатого файла;

– выбор области интереса – ROI (Region of Interest). Пользователь может сжать более важную часть изображения произвольной формы с лучшим качеством. Такая ситуация может встретиться, например, при архивации медицинских изображений;

– произвольный доступ и обработка сжатого изображения. Стандарт предоставляет возможность доступа к любой части сжатого изображения с возможностью проведения преобразований (кодирование, поворот, наклон и т. д.) без распаковки всего изображения;

– робастность к ошибкам при передаче бит, что особенно ценно при передаче изображений по зашумленным телекоммуникационным каналам.

Ключевые возможности технологии JPEG2000 демонстрируются на рис. 3.4.

Представление информации, полученное в результате вейвлет-преобразования, удобно тем, что оно обеспечивает возможность получения приблизительных копий изображения без осуществления полного обратного преобразования. Производя ограниченное число обратных декомпозиций (объединение частотных диапазонов), полагая, что все не вовлеченные в преобразование частотные диапазоны содержат исключительно нулевые элементы, мы легко можем получить либо копию изображения в уменьшенном масштабе, либо исходное изображение, но в более низком качестве по сравнению с изображением, полученным в результате полного обратного преобразования. Учитывая тот факт, что блоки вейвлет-коэффициентов кодируются независимо друг от друга, мы получаем возможность частичного декодирования не только на уровне преобразования, но и на уровне интерпретации кода. Для получения приблизительной копии изображения достаточно декодировать часть информации, а затем произвести обратное вейвлет-преобразование.

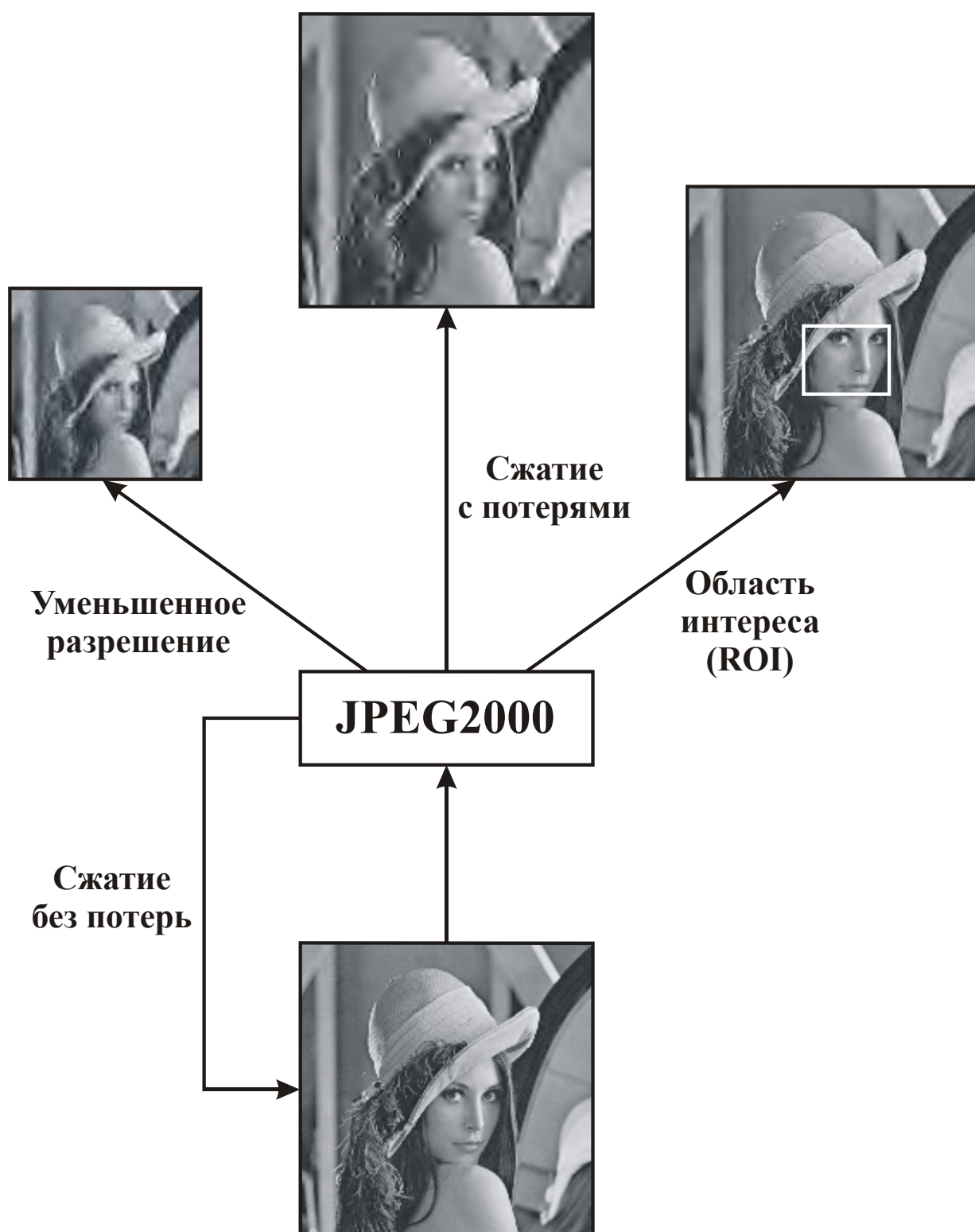


Рис. 3.4. Примеры использования технологии JPEG2000

Произвольный доступ обеспечивается путём деления изображения на тайлы (упомянутые ранее как блоки изображения), а также путём разделения кода тайла на слои, которые характеризуют отдельную область исходного тайла.

Возможности, отличающие JPEG2000 от других стандартов (высокая степень сжатия, возможность быстрого доступа к отдельным частям информации, превосходная структурированность, нетребовательность к ресурсам и многое другое), потенциально обеспечивают широкое использование стандарта в тех областях, где требуется обработка и передача высококачественных изображений.

Стандарт JPEG2000 превосходит по эффективности стандарт JPEG как в случае сжатия с потерями, так и в случае сжатия без потерь. Конечно, эффективность при сжатии без потерь в данном случае оказывается не такой высокой, как, скажем, у стандарта JPEG-LS, однако она вполне приемлема. Что же касается эффективности сжатия с потерями, здесь стандарт позволяет получать результаты, близкие к наилучшим на сегодняшний день результатам для подобного рода методов.

3.5. Алгоритм SPIHT

Алгоритм разложения множества по иерархическим деревьям – SPIHT (Set partitioning in hierarchical trees) – представляет собой метод сжатия изображений, базирующийся на вейвлет-преобразовании. На данный момент он не является стандартом сжатия, таким как JPEG и JPEG2000, но зачастую не менее эффективен, чем выше упомянутые алгоритмы.

Вейвлет-преобразование разлагает изображение на поддиапазоны так, что нижние поддиапазоны соответствуют высоким частотам изображения, а верхние – его низким частотам, в которых концентрируется основная часть энергии изображения. Поэтому можно ожидать, что коэффициенты деталей (высокочастотные коэффициенты) изображения уменьшаются при перемещении от высокого уровня к более низкому. Кроме того, имеется определенное пространственное подобие между поддиапазонами. Эта особенность разложения как раз и используется в алгоритме SPIHT.

3.5.1. Прогрессивная передача в алгоритме SPIHT

Одной из важных особенностей алгоритма SPIHT является прогрессивная передача, которая свойственна большинству методов сжатия изображений. Рассмотрим ее более подробно.

Обозначим пиксели исходного изображения через $P[i, j]$, а множество фильтров, которое может быть использовано для преобразования пикселей в вейвлет-коэффициенты $C[i, j]$, через T .

Вейвлет-коэффициенты образуют некоторый образ C . Само преобразование обозначим как $C = T(P)$.

При прогрессивном методе передачи декодер вначале присваивает значение ноль реконструированному образу \tilde{C} . Затем он принимает преобразованные коэффициенты, декодирует их и использует для получения улучшенного образа \tilde{C} , который, в свою очередь, производит улучшенное изображение $P = T^{-1}(\tilde{C})$. Таким образом, декодер способен очень быстро показать всю картинку в формате с низким качеством, а затем постепенно улучшать качество по мере приема остальной части сжатого изображения и его декодирования. В результате, декодируя 5–10% от размера исходного изображения, можно распознать все его особенности, что недостижимо при последовательной передаче в аналогичных условиях.

Основная цель прогрессивного метода состоит в скорейшей передаче самой важной части информации об изображении, которая дает самое большое сокращение расхождения исходного изображения и реконструированного образа. Для количественного измерения этого расхождения алгоритм SPIHT использует среднеквадратическую ошибку.

Два основных принципа алгоритма SPIHT, используемые при прогрессивной передаче изображения, состоят в том, что, во-первых, кодер должен посылать в первую очередь самые большие (по абсолютной величине) коэффициенты, а во-вторых, самые старшие биты (двоичного представления коэффициентов), так как они несут в себе информацию, которая больше всего сокращает расхождение среднеквадратической ошибки.

3.5.2. Формализованный алгоритм кодера SPIHT

Для упрощения описания основ данного метода вначале предположим, что коэффициенты отсортированы до начала цикла.

Алгоритм 1. Упрощенный SPIHT

Шаг 1. Для заданного изображения вычисляем его вейвлет-преобразование, используя подходящие вейвлет-фильтры, получаем коэффициенты преобразования $C[i, j]$ и представляем их в виде целых чисел фиксированной разрядности.

Шаг 2. Передаем n_{\max} – количество разрядов (считая от нуля), занимаемых максимальным отсчетом разложения, то есть $n_{\max} = \log_2(\max_{i,j}(a_{i,j}))$ (округление вниз).

Шаг 3. Инициализируем список значимых коэффициентов как пустое множество.

Шаг 4. Сортировка. Передаем число l коэффициентов $C[i, j]$, которые удовлетворяют неравенству $2^n \leq |C[i, j]| \leq 2^{n+1}$. Затем передаем l пар координат и l знаков этих коэффициентов.

Шаг 5. Поправка. Передаем $(n-1)$ -ые старшие биты всех коэффициентов, удовлетворяющих неравенству $|C[i, j]| \geq 2^n$. Эти коэффициенты были выбраны на шаге сортировки предыдущей итерации цикла.

Шаг 6. Итерация. Уменьшаем n на 1. Если необходимо сделать еще одну итерацию, переходим на **Шаг 4**.

Обычно последняя итерация совершается при $n = 0$, но кодер может остановиться раньше. Тогда наименее важная часть информации (некоторые менее значимые биты всех вейвлет-коэффициентов) не будет передаваться. В этом заключается естественное отбрасывание информации в алгоритме SPIHT. В противном случае кодер передает весь образ (все биты всех вейвлет-коэффициентов), а декодер может остановить процесс декодирования в любой момент, когда восстанавливаемое изображение достигло требуемого качества. Это качество и предопределяется пользователем или устанавливается декодером автоматически.

Процесс восстановления изображения в данном алгоритме является симметричным.

3.5.3. Алгоритм сортировки разделением множеств

Описанный выше алгоритм очень прост, так как в нем предполагается, что коэффициенты были упорядочены до начала цикла. Поэтому важным для реализации алгоритма SPIHT на практике является метод сортировки вейвлет-коэффициентов, участвующих в прогрессивной передаче.

Сортировка коэффициентов в алгоритме SPIHT основана на факте сравнения в каждый момент времени двух элементов, а каждый результат сравнения – это просто ответ «да» или «нет». Поэтому если кодер и декодер используют один и тот же алгоритм сортировки, то кодер может просто послать декодеру последовательность результатов сравнения «да» или «нет», а декодер будет дублировать работу кодера.

Алгоритм, используемый в SPIHT, основан на том, что нет необходимости сортировать все коэффициенты. Главной задачей этапа сортировки на каждой итерации является выявление

коэффициентов, удовлетворяющих неравенству $2^n \leq |C[i, j]| \leq 2^{n+1}$, с уменьшением n в каждом проходе. Если для данного n выполняется неравенство $|C[i, j]| \geq 2^n$, то говорят, что коэффициент значимый, в противном случае его называют незначимым.

Кодер разделяет все коэффициенты на некоторое количество множеств T_k и выполняет тест на значимость:

$$\max_{(i,j) \in T_k} \{|C[i, j]|\} \geq 2^n ?$$

для каждого множества T_k . Если декодер получает «нет» в качестве ответа (множество является незначимым), то это означает, что все коэффициенты в T_k являются незначимыми. Если ответ «да» (множество значимо), то по некоторому правилу, общему и для кодера, и для декодера, происходит деление T_k на новые множества $T_{k,l}$, и тест на значимость применяется к новым множествам. Этот процесс продолжается до тех пор, пока все значимые множества не будут иметь размер 1 (то есть каждое будет содержать ровно один коэффициент, который является значимым).

Тест на существенность множества T_k можно записать так:

$$S_n(T_k) = \begin{cases} 1, & \max_{(i,j) \in T_k} \{|C[i, j]|\} \geq 2^n, \\ 0, & \text{иначе.} \end{cases}$$

Поскольку результат каждого теста записывается в сжатый файл, то хорошо было бы минимизировать число необходимых тестов. Для достижения этой цели было предложено использовать специальную структуру данных – пространственно ориентированное дерево.

3.5.4. Пространственно ориентированное дерево

Пространственно ориентированное дерево – это структура, которая определяется с использованием пространственных соотношений между вейвлет-коэффициентами на различных уровнях пирамиды поддиапазонов.

На рис. 3.5 показано, как пространственно ориентированное дерево определено в пирамиде вейвлет-коэффициентов, созданной рекурсивным разбиением с четырьмя поддиапазонами. Каждый узел

дерева соответствует пикселю (вейвлет-коэффициенту) и определяется через пиксельные координаты. Его прямые потомки (*отпрыски* – четыре непосредственных потомка данного узла) соответствуют пикселям той же самой пространственной ориентации на следующем более низком уровне пирамиды.

Дерево определено таким образом, что каждый узел либо не имеет ни одного отпрыска (*листья* – вейвлет-коэффициенты, расположенные на первом уровне пирамиды поддиапазонов), либо имеет четыре отпрыска, которые всегда формируют группу 2×2 смежных пикселя. На рис. 3.5 стрелками показаны пространственные взаимосвязи между родительской вершиной и ее четырьмя отпрысками в пирамиде вейвлет-разложения.

Пиксели на самом высоком уровне пирамиды – это *корни* дерева, которые также формируются в группы 2×2 смежных пикселя. Однако их потомство, ответвленное по определенному правилу, является различным, ибо в каждой группе один из корней (обозначенный звездочкой на рис. 3.5) не имеет потомков.

Введем следующие четыре множества координат, которые используются алгоритмом сортировки разделением множеств.

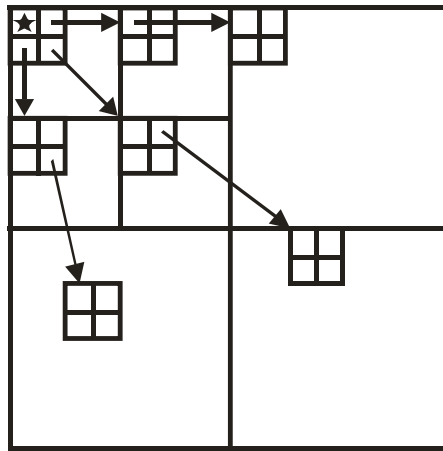


Рис. 3.5. Примеры пространственных взаимосвязей между родительской вершиной и ее четырьмя отпрысками в пирамиде вейвлет-разложения

1. $O(i, j)$: множество координат всех отпрысков узла (i, j) . Если узел (i, j) является листом пространственно ориентированного дерева, то множество $O(i, j)$ пусто.
2. $D(i, j)$: множество координат всех потомков узла (i, j) .
3. $H(i, j)$: множество координат всех корней пространственно ориентированного дерева (узлы на самом высоком уровне пирамиды).

4. $L(i, j) = D(i, j) - O(i, j)$. Это множество содержит всех потомков узла (i, j) за вычетом четырех его отпрысков.

Используя эти обозначения для отпрысков узла (i, j) , не являющегося листом дерева или корнем, обозначенным на рис. 3.5 звездочкой, можно записать:

$$O(i, j) = \{(2i, 2j), (2i, 2j + 1), (2i + 1, 2j), (2i + 1, 2j + 1)\}.$$

Как было отмечено выше, пространственно ориентированные деревья в алгоритме SPIHT используются для создания и разбиения множеств T_k . Последнее осуществляется с использованием следующих правил.

1. Начальными множествами являются $\{(i, j)\}$ и $D(i, j)$ для всех $(i, j) \in H$.
2. Если множество $D(i, j)$ является значимым, то его разбивают на $L(i, j)$ плюс еще четыре одноэлементных множества с $(k, l) \in O(i, j)$.
3. Если множество $L(i, j)$ является значимым, то его разбивают на четыре множества $D(k, l)$, где $(k, l) \in O(i, j)$.

3.5.5. Алгоритм кодирования

Порядок, в котором множества проверяются на значимость, очень важен. Поэтому в практической реализации алгоритма SPIHT вся информация сохраняется в трех упорядоченных списках:

1. LIP – список незначущих точек (*list of insignificant pixel*);
2. LIS – список незначущих множеств (*list of insignificant sets*);
3. LSP – список значущих точек (*list of significant pixel*).

В эти списки заносятся координаты (i, j) так, что в списках LIP , LSP они представляют индивидуальные коэффициенты, а в списке LIS они представляют или множество $D(i, j)$ (запись типа A), или множество $L(i, j)$ (запись типа B).

Ниже полностью представлен модифицированный алгоритм кодирования SPIHT. **Алгоритм 2**, в отличие от **Алгоритма 1**, использует подход деления множеств на этапе сортировки. Его упрощенная блок-схема приведена на рис. 3.6.

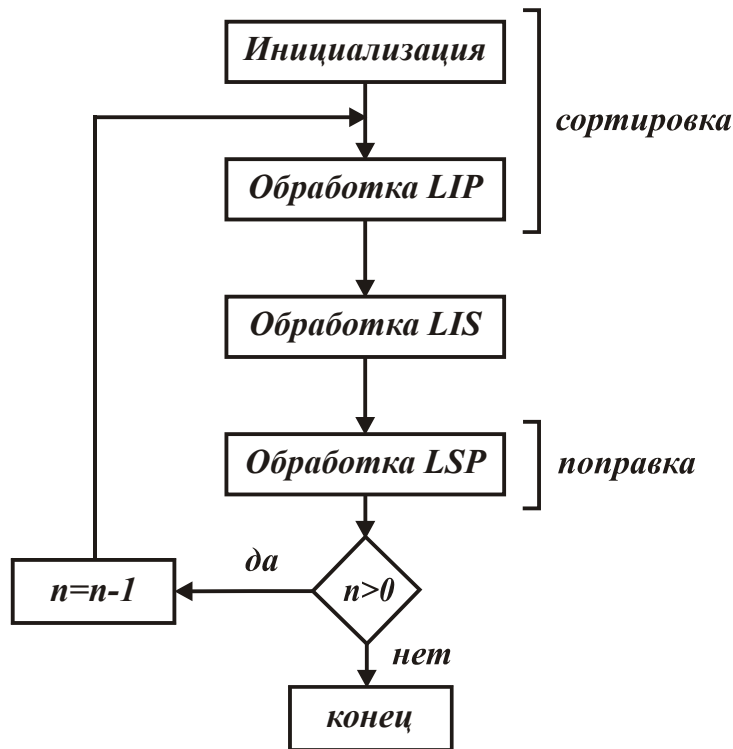


Рис. 3.6. Упрощенная блок-схема модифицированного SPIHT кодера

Алгоритм 2. Модифицированный алгоритм SPIHT на основе подхода разделения множеств

Шаг 1. Для заданного сжимаемого изображения вычисляем его вейвлет-преобразование, используя подходящие вейвлет-фильтры, получаем коэффициенты преобразования $C[i, j]$ и представляем их в виде целых чисел фиксированной разрядности.

Шаг 2. Передаем n_{\max} – количество разрядов (считая от нуля), занимаемых максимальным отсчетом разложения, то есть $n_{\max} = \log_2(\max_{i,j}(a_{i,j}))$ (округление вниз).

Шаг 3. Инициализируем список LSP как пустой и добавляем координаты $(i, j) \in H$ в LIP , а в список LIS координаты $(i, j) \in H$, которые имеют *потомков*, записи типа A .

Шаг 4. Сортировка.

4.1. Для каждой записи из LIP выполняем:

4.1.1. Передача $S_n(i, j)$;

4.1.2. Если $S_n(i, j) = 1$, то перемещаем (i, j) в LSP и передаем знак $C[i, j]$;

4.2. Для каждой записи (i, j) из LIS выполняем:

4.2.1. Если запись имеет тип A , то

- Передача $S_n(D(i, j))$;
- Если $S_n(D(i, j)) = 1$, то

- Для каждого $(k, l) \in O(i, j)$ выполняем:
 - Передача $S_n(k, l)$;
 - Если $S_n(k, l) = 1$, то добавляем (k, l) в LSP и передаем знак $C[k, l]$;
 - Если $S_n(k, l) = 0$, то добавляем (k, l) в конец LIP ;
- Если $L(i, j) \neq 0$, перемещаем (i, j) в конец LIS как запись типа B и переходим на шаг 4.2.2, в противном случае удаляем запись (i, j) из LIS ;

4.2.2. Если запись имеет тип B , то

- Передача $S_n(L(i, j))$;
- Если $S_n(L(i, j)) = 1$, то
 - Добавляем каждый $(k, l) \in O(i, j)$ в конец LIS как запись типа A ;
 - Удаляем (i, j) из LIS .

Шаг 5. Поправка. Для каждой записи (i, j) из LSP за исключением тех, которые были включены в последний проход сортировки, то есть с тем же n , передаем n -й наиболее значимый бит $|C[i, j]|$.

Шаг 6. Итерация. Уменьшаем n на 1. Если необходимо сделать еще одну итерацию, переходим на **Шаг 4**.

Здесь еще раз отметим, что алгоритм декодирования SPIHT является симметричным алгоритму кодирования, причем кодер и декодер работают всегда синхронно друг с другом. Более тонкие моменты работы декодера можно найти в соответствующей научно-технической литературе.

3.5.6. Основные характеристики алгоритма SPIHT

Существенными положительными сторонами рассмотренного алгоритма SPIHT являются:

1. Возможность восстановления всего изображения при прекращении приема битового потока. При этом изображение будет максимально хорошего качества для данного числа бит. Это применимо для передачи по каналам с потерями, а также для приложений вещания.

2. Возможность быстрого просмотра изображения в удаленной базе данных. Для поиска достаточно и грубой копии, а при нахождении нужного изображения оно декодируется полностью.

3. Возможность регулирования скорости передачи и размера сжатого файла (с точностью до байта).

4. Сжатие некоторых изображений в 20–30 раз без визуально воспринимаемых потерь.

Недостатки алгоритма SPIHT:

1. Использование списковых структур, что усложняет реализацию данного алгоритма на цифровых сигнальных процессорах, т. к. при использовании списков происходит динамическая работа с памятью.

2. Приспособленность SPIHT к изображениям с малым числом контуров и резких переходов. В связи с этим рассматриваемый алгоритм начинает значительно проигрывать, например, алгоритму JPEG2000 на изображениях со сложной пространственной структурой.

В заключение необходимо отметить, что работа алгоритма SPIHT в значительной мере зависит от количества уровней вейвлет-разложения, используемого банка вейвлет-фильтров, арифметического кодирования. Более подробную информацию о возможностях оптимизации алгоритма можно найти, например, в работах зарубежных (A. Said, W. A. Pearlman, N. Sprihan, S. Grgic, M. Grgic, S. Jayaraman) и отечественных (М. К. Чобану, В. Ю. Кобелев, В. А. Волохов, И. С. Мочалов) авторов.

Для сравнения результатов работы алгоритмов JPEG, JPEG2000 и SPIHT используются субъективные и объективные критерии оценки качества изображений. Рассмотрим их более подробно.

4. ОЦЕНКА КАЧЕСТВА СЖАТЫХ ИЗОБРАЖЕНИЙ

4.1. Классификация критериев оценки качества

На современном этапе развития цифровой обработки изображений существует несколько критериев оценки их качества (рис. 4.1). Качество столь сложного объекта, как изображение, является очень важным, но вместе с тем нечетким понятием. С одной стороны, качество может рассматриваться как характеристика самого изображения и определяться его собственными свойствами (статистическими, семантическими, структурными). Соответствующие критерии являются либо субъективными, либо опираются на объективные характеристики изображения: форму и параметры распределения яркости, оценки искажения деталей и т. д.

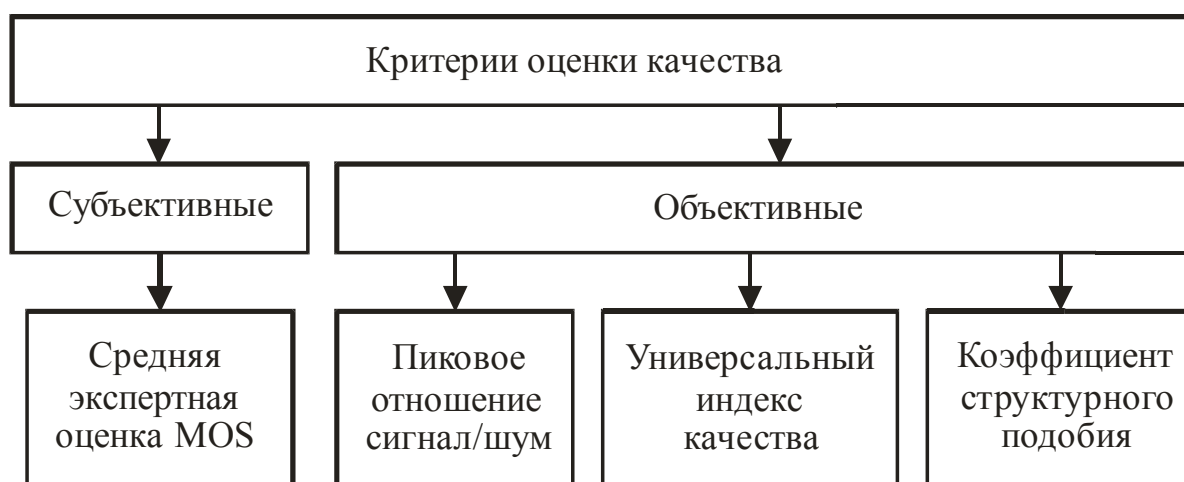


Рис. 4.1. Классификация критериев оценки качества изображений

С другой стороны, качество рассматривается как мера близости двух изображений: оцениваемого и эталонного. Такой подход более конструктивен, он позволяет оценивать количественные изменения значений яркости, уровень искажений изображений при преобразованиях (фильтрация, сжатие с потерями и т. д.), т. е. оценивается качество самого средства преобразования – алгоритма или устройства (системы). Это очень важно при построении систем обработки изображений, оценке качества работы соответствующих алгоритмов.

С того момента, как исследователи стали широко использовать различные средства обработки изображений, наиболее надежным и верным способом оценки качества изображения стала субъективная экспертиза. Средняя экспертная оценка (Mean Opinion Score, MOS), являющаяся субъективной, долгое время была наилучшим методом

оценивания качества изображения. Однако метод MOS дорог, так как требует участия большого количества людей и, как правило, не пригоден для использования в реальных ситуациях.

Сложность и дороговизна субъективных экспертных оценок делают привлекательным автоматическое измерение качества с помощью определенных алгоритмов. Такие меры качества должны хорошо согласовываться с результатами субъективных оценок для широкого класса изображений, не подразумевая при этом проведение чересчур сложных вычислений. Также желательно, чтобы эти метрики имели простую аналитическую форму и их можно было бы применять в качестве критериев оптимальности при выборе параметров системы обработки изображений.

Основной задачей исследования объективных систем оценки качества является создание компьютерной модели, которая могла бы автоматически и достоверно оценивать качество изображений. Другими словами, идеальный алгоритм должен оценивать качество так, как оценил бы его человек.

Очевидно, развитие подобных алгоритмов несет в себе большой потенциал в их последующем применении.

Во-первых, подобные алгоритмы найдут применение в оценке и сравнительном анализе систем обработки изображений.

Во-вторых, алгоритмы оценки качества могут быть интегрированы в системы обработки и передачи информации с целью их оптимизации в режиме реального времени.

В процессе разработки методов оценки часто возникает выбор между точностью результатов и сложностью алгоритма, зависящий от конкретной задачи. Например, если бы существовала система, которая могла бы оценивать все значимые аспекты в человеческом восприятии качества, то такая система могла бы точно оценивать визуальное качество. Однако наши знания о зрительной системе человека (ЗСЧ) ограничены. В процессе углубления знаний в данной области будут появляться алгоритмы, все более достоверно оценивающие качество изображений.

Численные методы оценки качества изображений по отношению к субъективным метрикам часто являются недостаточно адекватными. Поэтому до последнего времени идея автоматической оценки качества изображений считалась малоперспективной.

4.2. Субъективные критерии качества

Наиболее распространенным способом оценки качества изображений является субъективная экспертиза. Существуют два основных вида экспертных оценок: абсолютные и сравнительные. В первом случае наблюдатель должен оценить качество изображения по какой-то заранее определенной шкале. При некоторых методиках процесс оценивания облегчается тем, что наблюдателю предложен набор эталонных изображений. Существуют и другие методики, когда наблюдатель вынужден принимать решение только на основании своего ощущения.

При сравнительных оценках наблюдатель должен ранжировать набор конкретных изображений, т. е. расставить их в ряд по убыванию качества. Полученные оценки обрабатываются и усредняются. Существуют специальные приемы, исключающие «привыкание» экспертов в процессе экспериментов. Обычно в роли экспертов выступают специалисты в области обработки цифровых изображений, которые могут дать наиболее обоснованную оценку. Предполагается, что опытные наблюдатели замечают даже небольшие погрешности изображения, которые неспециалист может не заметить.

На результаты субъективных экспертных оценок влияют характер рассматриваемых изображений и окружающая обстановка (условия эксперимента). Если наблюдатель видел подобные изображения, то он склонен более строго оценивать погрешности, поскольку имеет сложившееся представление о структуре изображения. С другой стороны, в незнакомом изображении искажения могут оказаться незамеченными, пока наблюдателю не будет на них указано. Ясно, что условия эксперимента должны в максимальной степени соответствовать наблюдениям в реальной обстановке. Поэтому международный союз электросвязи принял нормативный документ ITU-R BT.500-11, носящий рекомендательный характер, целью которого является обеспечение одинаковых условий проведения экспертиз в разных лабораториях.

Существует большое количество различных шкал оценки, они описаны в указанной рекомендации. Выбор конкретной шкалы зависит от цели исследования.

Одной из наиболее распространенных методик оценки является мера DMOS (difference mean opinion score), которая вычисляется как разность между средней оценкой оригинала и средней оценкой текущего искаженного изображения (MOS). Как следует из определения, для оценки используется одностимуляционный метод, при котором оригинальное изображение входит в набор

предлагаемых эксперту изображений и оценивается наряду с остальными. Оценка производится по шкале качества, согласно которой наблюдатель должен оценить общее качество изображения, изменяющееся от «очень плохого» до «очень хорошего».

Шкала качества:

1. Очень плохо.
2. Плохо.
3. Средне.
4. Хорошо.
5. Очень хорошо.

Результатом экспертных оценок является линейное отображение значения, определенного по шкале качества, в диапазон оценок от 1 до 100. Считается, что для получения надежной оценки качества изображения необходимо опросить не менее двадцати наблюдателей.

Проведение экспертизы весьма сложная и дорогая задача, при этом ее результаты приблизительны. Но главный недостаток субъективного критерия – отсутствие количественных оценок. Он не позволяет решать задачи оптимизации изображений в пространстве непрерывно изменяющихся параметров. Здесь возможен только лишь небольшой перебор параметров.

Желательно, чтобы критерий имел простую аналитическую форму и просто вычислялся по предъявляемым изображениям. Этому требованию удовлетворяет ряд «объективных» критериев, рассматриваемых ниже.

4.3. Объективные критерии качества

Сложность и дороговизна субъективных экспертных оценок делают привлекательным автоматическое измерение качества с помощью определенных правил. Многие разработчики систем цифровой обработки изображений часто опираются на объективные (алгоритмические) меры качества. Объективные меры качества должны хорошо согласовываться с результатами субъективных оценок для широкого класса изображений, не подразумевая при этом проведение чересчур сложных вычислений. Также желательно, чтобы эти метрики имели простую аналитическую форму и их можно было бы применять в качестве критериев оптимальности при выборе параметров системы обработки изображений.

Во многих практических приложениях система оценки качества не имеет возможности получить изображение-оригинал. В этом случае довольно перспективным направлением развития является создание критерия, способного оценивать качество «вслепую». Такие критерии называются *неэталонными*. Однако разработка подобных критериев является очень трудной задачей. Человек способен легко оценивать качество даже при отсутствии оригинала. Это объясняется тем, что человеческая память содержит огромное количество информации о том, какими должны быть, а какими не должны быть изображения. Поэтому считается, что эффективная неэталонная модель оценки качества изображения возможна только при наличии априорной информации о типах искажений изображения.

На практике разработка эффективного метода оценки качества для конкретной задачи обычно проще, нежели разработка обобщенного метода, так как типы искажений заранее известны. Например, для изображений с блочной структурой (JPEG, MPEG) оценка заметности блоков может быть довольно хорошей оценкой качества изображения в целом.

4.3.1. Пиковое отношение сигнал / шум

На данный момент наиболее популярной объективной мерой служит пиковое отношение сигнал / шум (ПОСШ), определяемое по следующим формулам (для изображений с 256 градациями яркости):

$$\text{ПОСШ} = 20 \log_{10} \frac{255}{\sqrt{\text{СКО}}},$$

$$\text{СКО} = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2,$$

где N – число пикселей в изображении, x_i и y_i – значения пикселей исходного и восстановленного изображений соответственно, а СКО – среднеквадратичная ошибка. Высокое значение ПОСШ означает определенную схожесть восстановленного и исходного изображений, но оно не дает гарантию того, что зрителю понравится восстановленный образ. Большим минусом использования ПОСШ в системах цифровой обработки изображений и, в частности, в цифровых фильтрах является то, что данная величина не имеет абсолютного значения. Бессмысленно говорить о том, что если ПОСШ равно, например, 25 дБ, то это хорошо. Величина ПОСШ используется только для сравнения различных алгоритмов обработки

или для изучения влияния параметров на эффективность того или иного алгоритма.

Следует иметь в виду, что критерий ПОСШ будет характеризовать «среднее» качество изображения в целом, а на различных его фрагментах ошибки в принципе могут различаться.

Широкое применение данной оценки обусловлено тем, что подсчет этой характеристики математически прост и, следовательно, не требует больших вычислительных затрат.

4.3.2. Проблемы использования ПОСШ в качестве критерия оценки качества

При практическом использовании критерия ПОСШ возникает следующий ряд проблем:

1. Восприятие качества изображений не зависит от пространственных зависимостей между исходным и искаженным сигналом. Как результат, изменение порядка частей изображения не должно повлиять на результат оценки.

2. Визуальное восприятие не зависит от любых связей между сигналом изображения и сигналом ошибки. Это значит, что для сигнала с одинаковой ошибкой не имеет значения, каково исходное изображение. Результат будет одинаков.

3. Визуальное восприятие качества определяется только величиной сигнала ошибки. Как следствие, изменения в сигнале ошибки не влияют на результат оценки.

4. Все составляющие сигнала одинаково важны для визуальной оценки качества.

Иллюстрация этих положений приведена на рис. 4.2 – 4.4.

На рис. 4.2 изображение (б) создано добавлением белого Гауссова шума к исходному изображению (а). В изображении (в) амплитудные (яркостные) составляющие сигнала оставлены такими же, как и в изображении (а), но пространственный порядок изменен. Изображение (г) получено из изображения (б) использованием процедуры перераспределения, как и в случае изображения (в). Другими словами, координатные преобразования, используемые при переходе из (б) в (г), такие же, как и при переходе из (а) в (в). Значения ПОСШ для изображений (б) и (г) относительно (а) и (в) соответственно абсолютно одинаковы. Однако изображение (г) воспринимается более зашумленным, нежели (б).

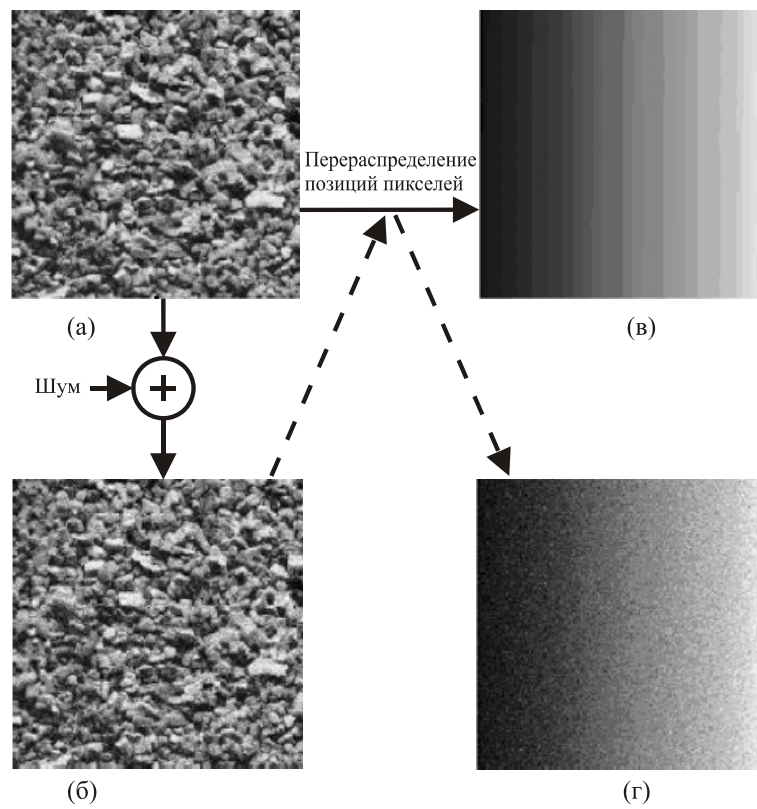


Рис. 4.2. Визуальное различие при добавлении Гауссова шума

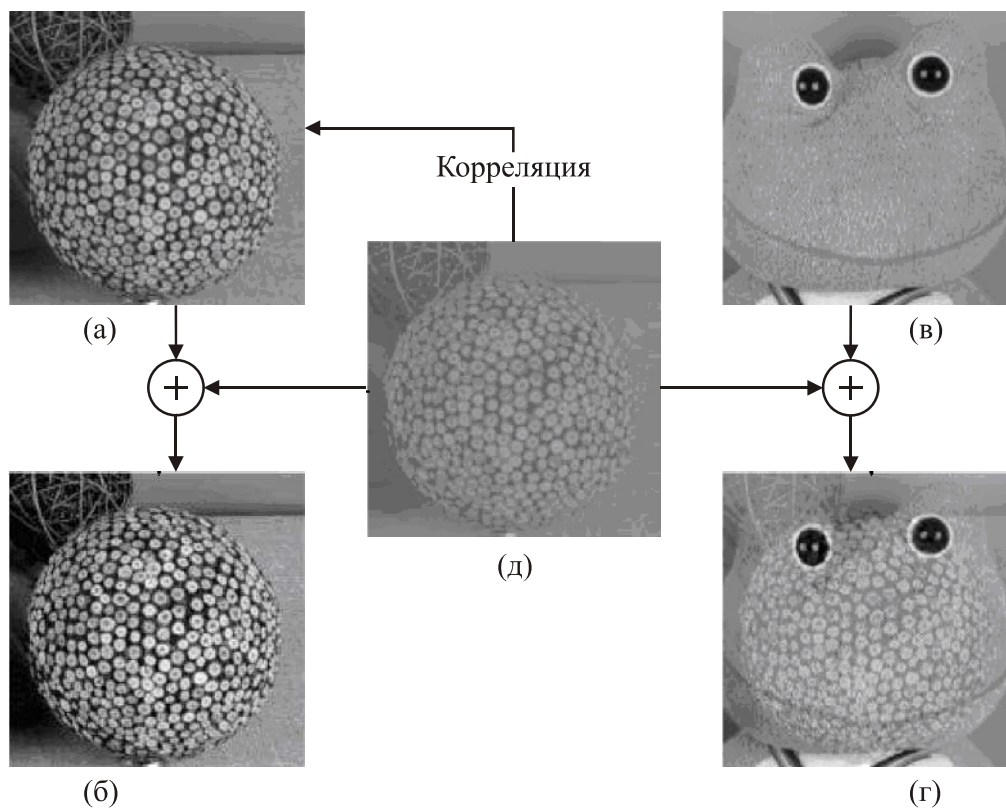


Рис. 4.3. Визуальное различие при добавлении коррелированного шума

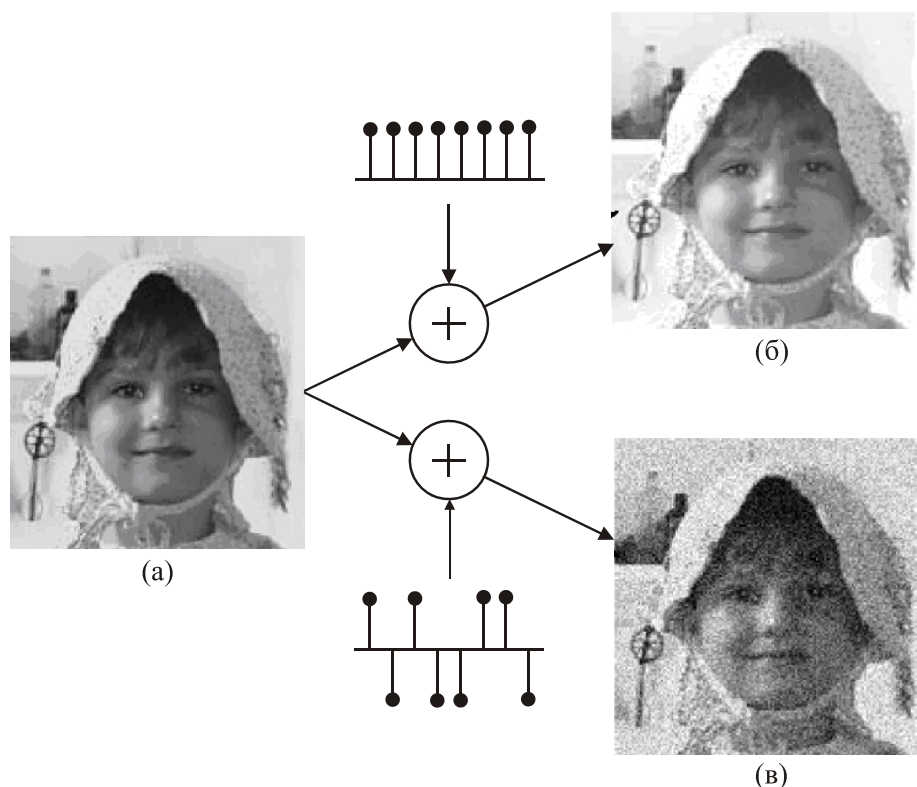


Рис. 4.4. Визуальное различие при добавлении к исходному изображению (а) константы (б) и знакопеременной последовательности (в)

На рис. 4.3 одинаковый сигнал ошибки добавлен к исходному изображению (а) и (в). Сигнал ошибки (д) выбран таким образом, что полностью коррелирует с изображением (а). Оба искаженных сигнала (б) и (г) имеют одинаковое значение ПОСШ. Однако визуальное искажение изображения (г) кажется сильнее, нежели (б).

На рис. 4.4 два искаженных изображения созданы из одного исходного. Первое изображение подвержено добавлению постоянной компоненты ко всем значениям пикселей. Второе изображение получено путем добавления ко всем значениям пикселей исходного изображения константы со случайным знаком. Визуальное качество изображений значительно различается, в то время как ПОСШ полностью игнорирует знаки и значения таких оценок качества для обоих изображений равны.

Другим примером может служить рис. 4.5., где исходное тестовое изображение «Эйнштейн» (а) изменено несколькими типами искажений: средний сдвиг яркости (б), увеличение контраста (в), искажение импульсным шумом (г), искажение Гауссовым шумом (д), размытие (е), JPEG-сжатие (ж). Необходимо отметить, что изображения (б) – (ж) имеют одинаковое значение СКО, но, очевидно, различное визуальное восприятие качества. Также СКО чувствительна к пространственным изменениям, изменению

масштаба и вращению. Это легко заметить на рис. 4.6, где то же самое эталонное изображение «Эйнштейн» подвержено изменению путем пространственного сдвига влево (а), зум-эффекта (б) и вращения (в).

Объясняются столь некоррелированные результаты тем, что пиксельная структура, используемая в ПОСШ, неточно отображает систему человеческого визуального восприятия. Психофизические и психофизиологические исследования показали, что функции зрительной системы человека не учитываются в ПОСШ.

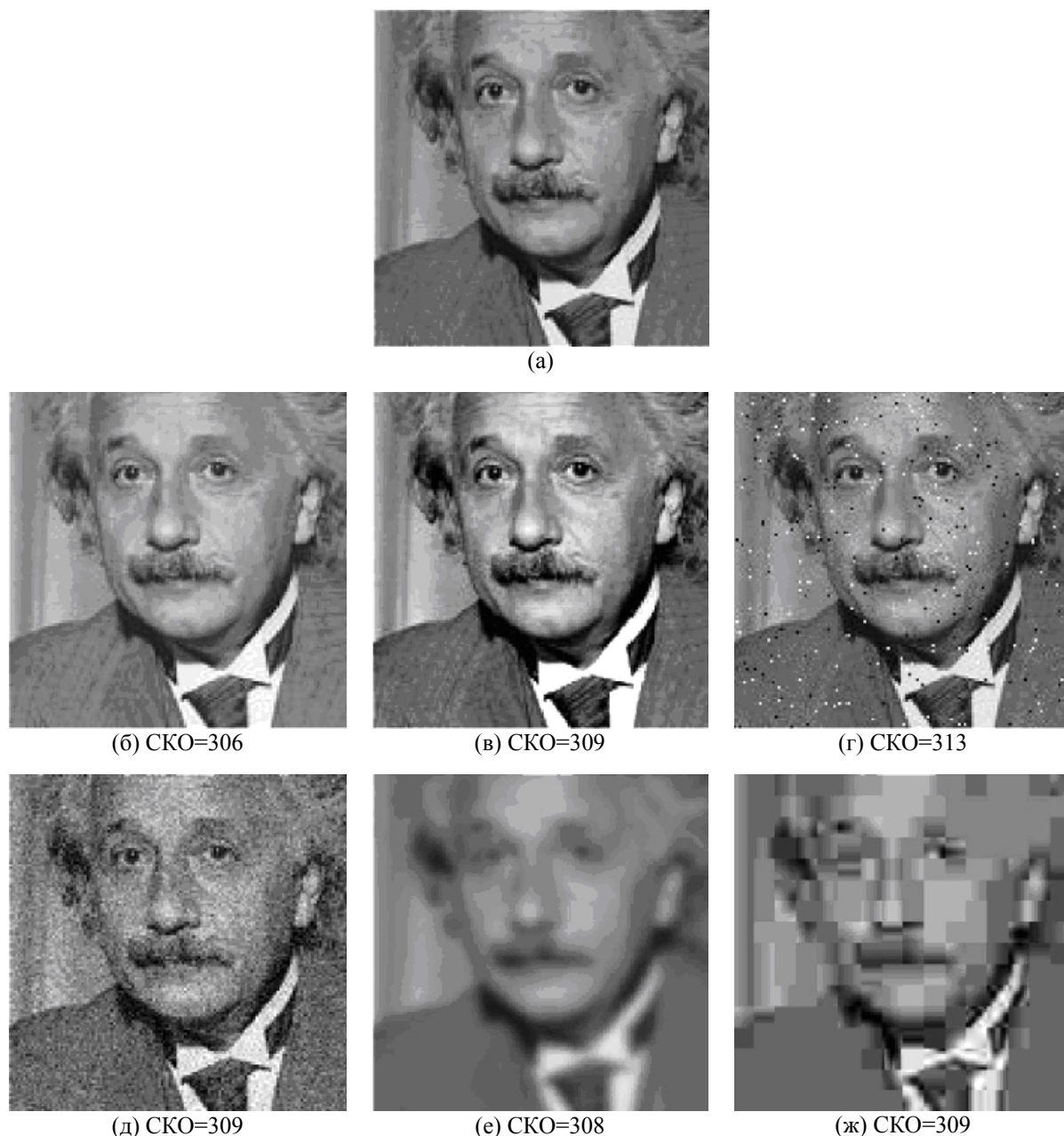


Рис. 4.5. Изображение «Эйнштейн» с близкими значениями SKO

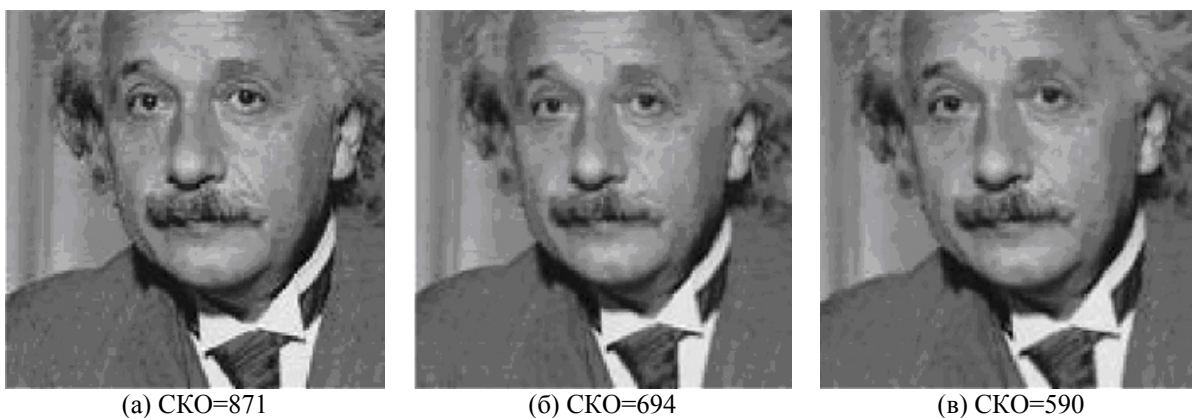


Рис. 4.6. Реакция СКО на пространственные изменения

4.3.3. Модификация ПОСШ с учетом характеристик зрительной системы человека

Проведенные исследования показали, что система визуального восприятия человека более восприимчива к искажениям на низких частотах, чем в высокочастотной области. Этот факт был учтен при разработке стандарта JPEG. В результате была спроектирована специальная таблица корректирующих величин, которая применяется для квантования коэффициентов ДКП в блоках 8×8 . Согласно этой таблице, высокочастотные коэффициенты квантуются с большим шагом квантования, что приводит к лучшему качеству восстановленного изображения, чем если применять для всех коэффициентов одинаковое квантование.

Выражение для СКО с плавающим окном 8×8 с ДКП, принимая в рассмотрение ЗСЧ, имеет вид:

$$\text{СКО}_M = \frac{\sum_{i=1}^{I-7} \sum_{j=1}^{J-7} \sum_{m=1}^8 \sum_{n=1}^8 ((X[m,n]_{i,j} - X[m,n]_{i,j}^e) T_c[m,n])^2}{(I-7)(J-7)64},$$

где $X_{i,j}$ – коэффициенты ДКП блока 8×8 , для которого координаты верхнего левого элемента равны i и j , $X_{i,j}^e$ – коэффициенты ДКП соответствующего окна оригинального изображения, T_c – матрица корректирующих факторов. В качестве такой матрицы используется матрица, построенная с использованием таблицы квантования стандарта JPEG.

Матрица T_c обладает следующими двумя свойствами:

1. Соотношения ее коэффициентов обратно пропорциональны соответствующим коэффициентам таблицы квантования JPEG.

2. Суммарный корректирующий фактор матрицы $\sum_{m=1}^8 \sum_{n=1}^8 \frac{T_c[m,n]^2}{64}$

равен единице. Это означает, что в случае однородного распределения искажений по всем пространственным частотам значения $СКО_M$ совпадают с $СКО$.

На основе полученного результата можно записать выражение для критерия $ПОСШ_M$, учитывающего особенности системы визуального восприятия человека:

$$ПОСШ_M = 10 \lg(255^2 / СКО_M).$$

Использование рассмотренной модификации $ПОСШ$ увеличивает корреляцию с субъективной оценкой экспертов. Однако в этом случае значительно возрастают временные затраты, требуемые на вычисление объективной оценки.

4.3.4. Универсальный индекс качества

В последние годы сделано множество работ по созданию более приемлемых алгоритмических мер визуального качества обрабатываемых изображений, которые бы стояли ближе к результатам субъективных оценок. Одним из таких критериев является универсальный индекс качества (УИК) изображений, используемый для определения подобности восстановленного изображения некоторому оригиналу в зависимости от вида и степени искажения входного изображения. Универсальность данного критерия заключается в том, что он не только отражает некоторую схожесть обработанного изображения по отношению к оригиналу, но и должным образом учитывает различные виды искажений. В отличие от многих других алгоритмов данный метод не привязан к специфике изображения и искажениям, присутствующим в нем. Он основывается на статистическом анализе отдельных блоков входного сигнала и дальнейшем сравнении полученных результатов со значениями эталонного изображения. Недостатком данного метода является только то, что, как и в случае вычисления $ПОСШ$, для нахождения УИК помимо тестового (входного) изображения необходимо иметь некоторый эталонный сигнал, соответствующий данному изображению.

Определение УИК для одномерных сигналов

Рассмотрим две последовательности $\{x_i\}$ и $\{y_i\}$, где $i = 1, 2, \dots, N$, соответствующие стационарному входному и выходному сигналам, тогда универсальный индекс качества будет определяться комбинацией статистических характеристик (математического ожидания, дисперсии и корреляционной функции) соответствующих последовательностей по следующей формуле:

$$\text{УИК} = \frac{4 \sigma_{xy} \bar{x} \bar{y}}{[\sigma_x^2 + \sigma_y^2][(\bar{x})^2 + (\bar{y})^2]}, \quad (4.1)$$

где

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i,$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i,$$

$$\sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2,$$

$$\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2,$$

$$\sigma_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}).$$

Таким образом, получается, что значение УИК изменяется в интервале $[-1, 1]$, при этом $\text{УИК}=1$ соответствует наилучшему качеству сигнала. Важно отметить, что максимальное значение $\text{УИК}=1$ получается только тогда, когда $y_i = x_i$ на протяжении всего сигнала ($i = 1, 2, \dots, N$), а минимальное значение $\text{УИК}=-1$, в случае если $y_i = 2\bar{x} - x_i$, $i = 1, 2, \dots, N$. Другими словами, максимальное значение $\text{УИК}=1$ соответствует такому сигналу, который в точности совпадает с оригиналом.

Определенный таким образом индекс качества учитывает в себе три искажающих фактора: степень коррелированности отчетов двух сигналов, изменение значений математического ожидания и среднеквадратичных отклонений сигнала относительно оригинала. Такое представление позволяет наиболее точно оценивать качество

восстановленного сигнала и величину искажений, присутствующих в нем. В результате этого исходное выражение (4.1) может быть представлено в виде произведения трех множителей:

$$\text{УИК} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2}. \quad (4.2)$$

Первый множитель есть коэффициент взаимной корреляции сигналов x и y , который определяет степень их линейной подобности и принимает значения в интервале $[-1, 1]$, при этом максимальное значение достигается в случае, если $y_i = ax_i + b$, где $a > 0$, $b > 0$ и $i = 1, 2, \dots, N$. Если восстановленный сигнал линейно соотносится с входным, но в нем присутствуют какие-либо искажения, они будут учитываться во втором и третьем множителях. Второй множитель (4.2) определяет соотношение между математическими ожиданиями входного и восстановленного сигнала и принимает значения в интервале $[0; 1]$, при этом максимум достигается в случае их равенства. В случае если сигналом является изображение, этот коэффициент отвечает за изменения его яркости.

Значения среднеквадратичных отклонений σ_x и σ_y определяют динамический диапазон изменения сигналов x и y соответственно, вследствие этого они могут рассматриваться как характеристики контраста, в том случае если сигналом является изображение. Таким образом, третий множитель в выражении (4.2) для УИК соответствует искажению контрастности восстановленного сигнала относительно оригинала и принимает значения в интервале $[0; 1]$.

Предложенный индекс качества обладает хорошей согласованностью со значениями субъективных оценок, к тому же, в отличие от ПОСШ, он имеет некоторое абсолютное значение, равное единице. То есть при показаниях УИК, близких к единице, можно говорить о высоком качестве восстановленного сигнала. Однако для прямого использования (4.1) в роли критерия качества цифровых изображений необходимо ввести некоторые ограничения.

Применение УИК для оценки качества изображений

Следует отметить, что предложенная в выражении (4.1) структура УИК соответствует одномерным стационарным сигналам. Изображение является двумерным и в общем случае нестационарным сигналом, поэтому прямое использование формулы (4.1) для

вычисления УИК в данном случае недопустимо. Для решения данной проблемы предлагается выделять в изображении отдельные локальные области размером $A \times B$, в пределах которых изображение можно считать стационарным и вычислять статистические характеристики (математическое ожидание, дисперсию и корреляционную функцию) внутри этой области, а уже на основании их вычислять значение $УИК_i$, соответствующее выделенному блоку. Такой подход обеспечивает наиболее правильное и точное суждение о качестве восстановленного изображения.

В результате этого процесс нахождения УИК для конкретного изображения будет сводиться к следующему: «скользящее окно» (маска) размером $A \times B$ начиная из верхнего левого угла проходит пиксель за пикселем по всем строкам и столбцам этого изображения, на каждом шаге вычисляется значение статистических характеристик и собственно сам $УИК_i$ для текущего положения окна. УИК, характеризующий качество изображения в целом, будет определяться как среднее арифметическое значений $УИК_i$ на протяжении всего изображения, т. е.

$$УИК = \frac{1}{M} \sum_{i=1}^M УИК_i ,$$

где M – число блоков, внутри которых вычислялся УИК.

Такой подход к нахождению численных оценок качества особенно хорош для изображений, имеющих блочную структуру. Как известно, одним из таких типов изображений является формат JPEG, основанный на дискретном косинусном преобразовании отдельных блоков размером 8×8 пикселей. В результате этого для оценки качества, например JPEG-изображений, возможно использование УИК с размером «скользящего окна», равным размеру блоков ДКП.

Именно благодаря комбинированной структуре УИК (4.2) оценка качества восстановленных изображений осуществляется наиболее точно. В ряде случаев показания УИК просто незаменимы, так как показания ПОСШ и СКО в некоторых случаях противоречат действительности. Это наблюдается, например, при изменении яркости изображений, трансформации гистограмм, гамма-коррекции и т. д. Особенно важно то, что УИК позволяет дифференцировать (различать) изображения, имеющие одинаковые значения ПОСШ и СКО, что способствует построению действительно оптимальных фильтров.

Исходя из формулы (4.1) для вычисления УИК следует, что в некоторых случаях могут возникать ситуации, когда значение нового

индекса качества имеет неопределенный характер, математически это означает неопределенность типа «ноль на ноль». Причиной этого могут служить два фактора. Знаменатель и числитель могут обращаться в ноль, во-первых при совместном равенстве нулю дисперсий для каких-либо блоков разбиения изображений, во-вторых, при совместном равенстве нулю математических ожиданий внутри этих блоков. На практике это встречается довольно часто, поэтому для разрешения таких неопределенных ситуаций предлагается исключать их из рассмотрения и не включать такие значения в конечный результат (рис. 4.7).

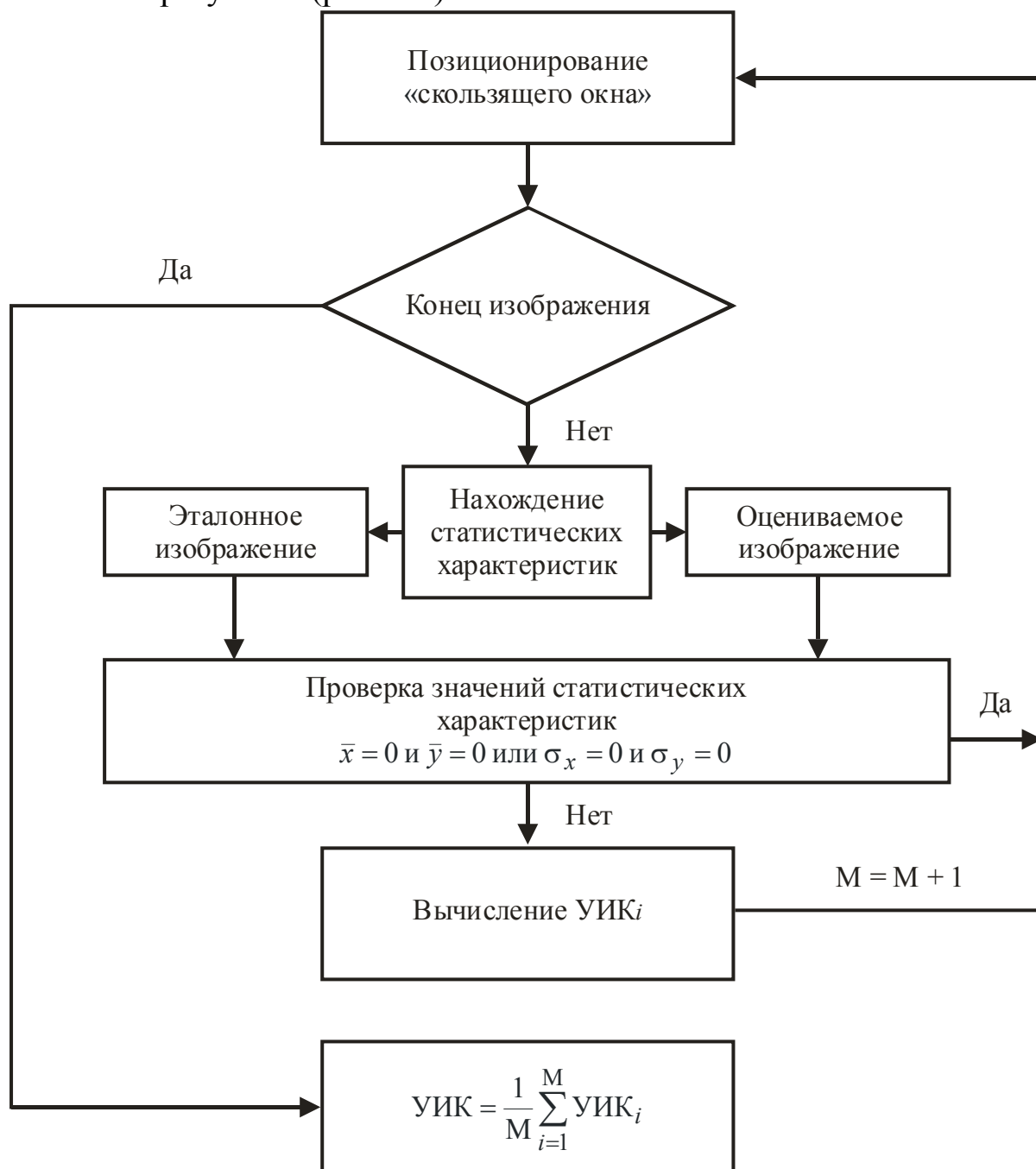


Рис. 4.7. Алгоритм нахождения универсального индекса качества для изображений

4.3.5. Коэффициент структурного подобия

Сигналы естественных изображений хорошо структурированы. Выборки, взятые из сигналов изображений, имеют сильную зависимость, и эти зависимости несут важную информацию о структуре: объектах, их расположении, текстурах. Главная идея, лежащая в основе метода структурного подобия, заключается в том, что зрительная система человека хорошо приспособлена к выделению структурной информации, и поэтому измерение степени структурного подобия должно обеспечивать хорошее приближение к визуально воспринимаемому качеству изображения.

Для того чтобы преобразовать общий принцип структурного подобия в конкретный алгоритм оценки качества изображения, необходимо разделить структурные и неструктурные искажения.

Неструктурными искажениями считаются те из них, которые не вносят изменения в структуру объектов на изображении, остальные называются структурными.

Изображение с измененным контрастом (рис. 4.8б) сохраняет фактически всю существенную информацию об объектах исходного (эталонного) изображения (рис. 4.8а). Таким образом, согласно принципу структурного подобия, такое искажение – неструктурное.

Изображение, искаженное артефактами сжатия с потерями по стандарту JPEG (рис. 4.8д), имеет то же самое СКО, что и изображение с измененным контрастом (рис. 4.8б), однако можно видеть, что артефакты сжатия значительно изменяют структуру объектов на изображении. Искажения такого рода являются структурными.

На рис. 4.8 исходное изображение представлено вектором в пространстве значений пикселей этого изображения. Любые искажения исходного изображения можно понимать как добавление вектора искажения к «эталонному» вектору. В частности, векторы искажения одинаковой длины определяют гиперсферу равного СКО в пространстве изображений. Длина вектора искажения не является эффективной оценкой его степени, тогда как направление имеет значение для визуально воспринимаемого качества.

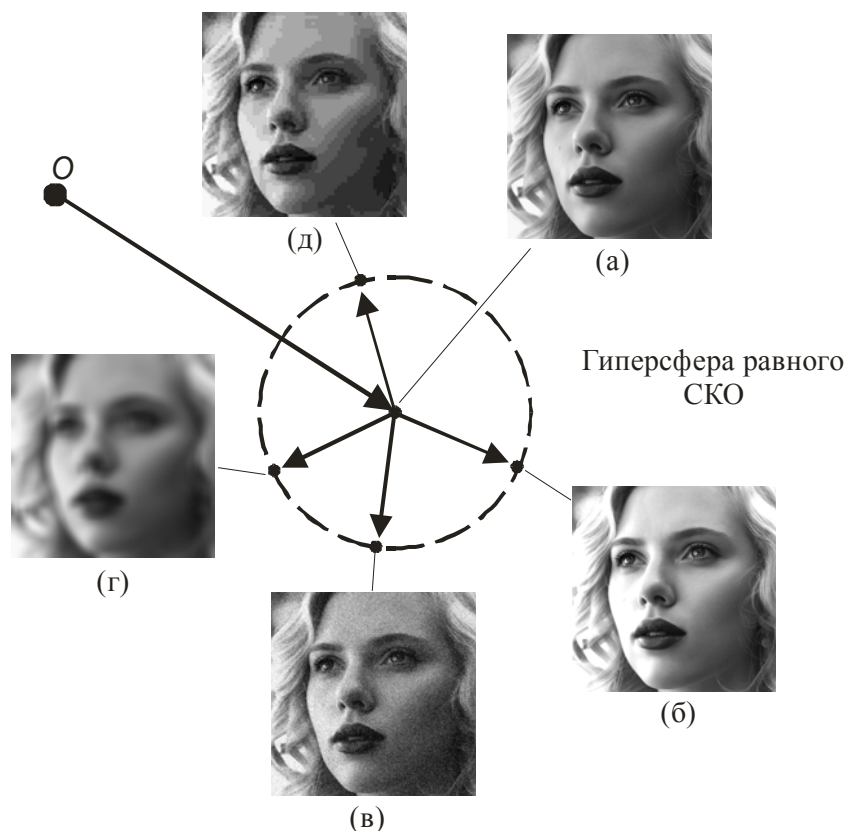


Рис. 4.8. Структурные и неструктурные искажения: а) оригинальное изображение; б) увеличение контраста; в) внесение шума; г) размытие; д) артефакты сжатия по стандарту JPEG

На рис. 4.9 показано, что искажения яркости и контраста отделяются от структурных искажений в рассмотренном пространстве. Векторы яркостных искажений параллельны вектору изображения равной яркости $x_1 = x_2 = \dots = x_N$, который перпендикулярен гиперплоскости $\sum_{i=1}^N x_i = 0$. Изменения контраста определяются направлением $\mathbf{x} - \bar{x}$. Таким образом, векторы искажений яркости и контраста определяют в исходном пространстве плоскость неструктурных искажений. Структурные же искажения соответствуют повороту этой плоскости на некоторый угол (рис. 4.9).

Коэффициент структурного подобия представляет собой результат сравнения эталонного и оцениваемого изображений, обозначенных как \mathbf{x} и \mathbf{y} соответственно. Алгоритм структурного подобия учитывает три составляющие: искажения яркости, искажения контраста и искажения структуры.

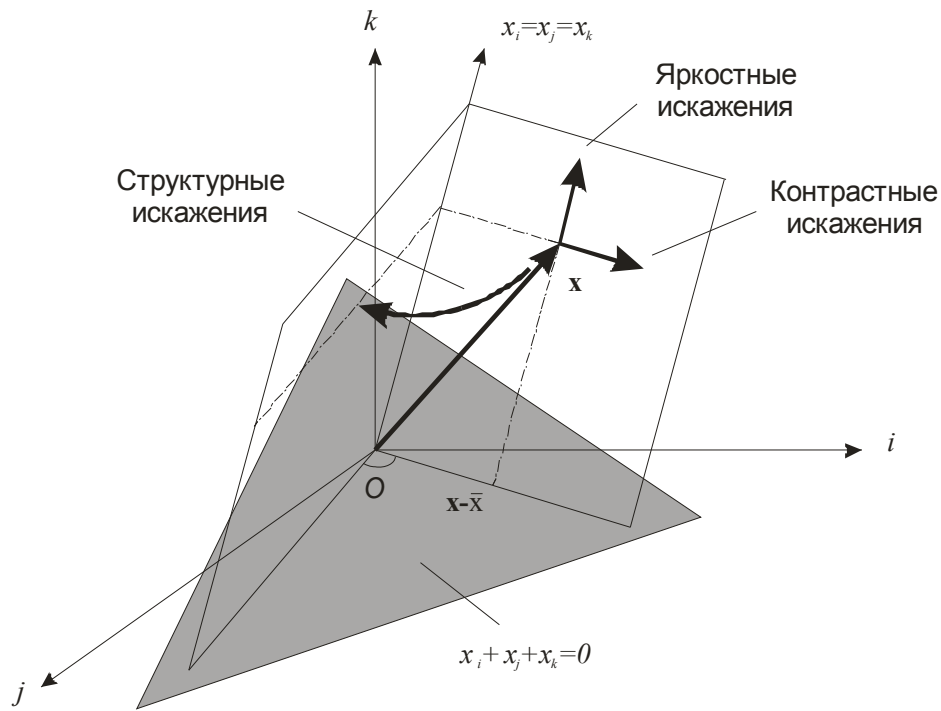


Рис. 4.9. Разделение структурных и неструктурных искажений
в пространстве изображения

Локальная яркость каждого изображения в пикселе с координатами (x, y) находится как усредненное значение яркости в локальной окрестности этого пикселя:

$$\mu_x = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad (4.3a)$$

$$\mu_y = \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i, \quad (4.3б)$$

где N – количество пикселей в рассматриваемой окрестности.

Функция $l(x, y)$, характеризующая искажения яркости, будет определяться через значения μ_x и μ_y :

$$l(x, y) = l(\mu_x, \mu_y).$$

Если вычесть среднее значение μ_x из окрестности сигнала x , то результат $x - \mu_x$ будет являться проекцией вектора x на

гиперплоскость $\sum_{i=1}^N x_i = 0$, изображенную на рис. 4.9. В качестве оценки локальной контрастности сигнала используется среднеквадратичное отклонение:

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2},$$

$$\sigma_y = \left(\frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2 \right)^{1/2}.$$

Функция $c(x, y)$, характеризующая искажения контраста, будет определяться через значения σ_x и σ_y :

$$c(x, y) = c(\sigma_x, \sigma_y).$$

Структурное подобие между эталонным и оцениваемым изображениями определяется через нормированные на свои средние значения соответствующие среднеквадратичные отклонения:

$$s(x, y) = s\left(\frac{x - \mu_x}{\sigma_x}, \frac{y - \mu_y}{\sigma_y}\right).$$

Все три составляющие объединяются в общую локальную оценку следующим образом:

$$S(x, y) = f[l(x, y), c(x, y), s(x, y)].$$

На функцию $S(x, y)$ накладываются следующие ограничения:

1. **Симметрия** $S(x, y) = S(y, x)$. Когда определяется подобие между двумя сигналами, изменение порядка входящих сигналов не должно влиять на результирующее измерение.

2. **Ограниченность** $S(x, y) \leq 1$. Верхняя граница служит показателем того, насколько близко два сигнала к тому, чтобы быть идентичными.

3. **Единственный максимум**: $S(x, y) = 1$. Функция $S(x, y)$ принимает максимальное значение тогда и только тогда, когда $\mathbf{x} = \mathbf{y}$.

Для удовлетворения этих условий входящие в состав $S(x, y)$ функции $l(x, y)$, $c(x, y)$, $s(x, y)$ определяются следующим образом:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (4.4)$$

где константа C_1 включена для того, чтобы избежать обращения в бесконечность, когда значение $\mu_x^2 + \mu_y^2$ близко к нулю. В частности, значение C_1 выбирают равным $(K_1 \cdot L)^2$, где L – динамический диапазон значений пикселей изображения, а $K_1 \ll 1$ – малая константа. Заметим, что уравнение (4.4) подчиняется трем свойствам, описанным выше.

Функция искажения контраста имеет подобный вид:

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (4.5)$$

где $C_2 = (K_2 \cdot L)^2$ и $K_2 \ll 1$. Эти определения также удовлетворяют трем свойствам, описанным выше.

Функция структурных искажений вводится следующим образом:

$$s(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (4.6)$$

где

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y).$$

Так же, как и для двух предыдущих функций, в выражение (4.6) вводится малая константа C_3 . Геометрически $s(x, y)$ – это косинус угла между векторами $(x - \mu_x)$ и $(y - \mu_y)$.

Три рассмотренные составляющие (4.4), (4.5) и (4.6) объединяются в локальный коэффициент структурного подобия изображений x и y :

$$S(x, y) = l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma,$$

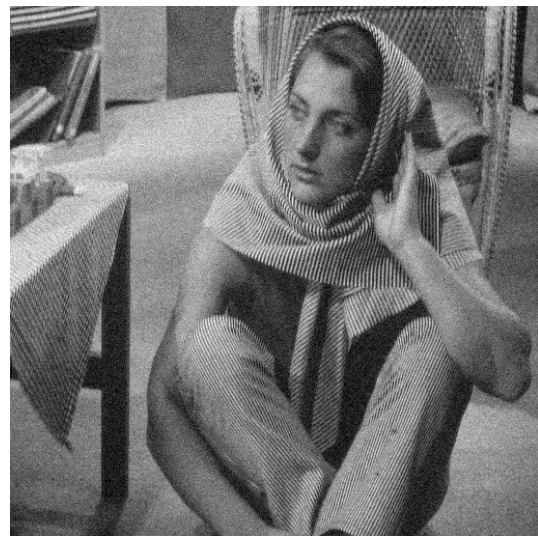
где $\alpha > 0$, $\beta > 0$, $\gamma > 0$ – параметры, которые регулируют относительную значимость отдельных составляющих.

На практике, как правило, используются следующие значения параметров: $\alpha = \beta = \gamma = 1$, $C_3 = C_2 / 2$, $K_1 = 0,01$, $K_2 = 0,03$.

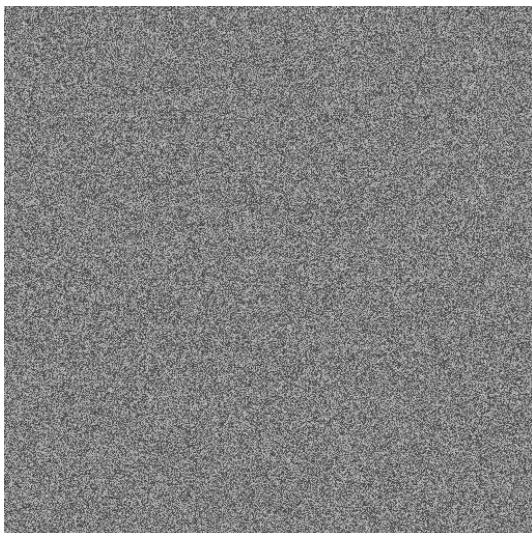
При вычислении КСП скользящим окном может быть получена карта структурного подобия для заданной пары изображений (эталонное-оцениваемое). Пример такой карты приведен на рис. 4.10. Более светлые участки на карте КСП соответствуют большему структурному подобию и, следовательно, большему качеству изображения.



а)



б)



в)



г)

Рис. 4.10. Структурное качество изображения: а) оригинал; б) зашумленное изображение; в) карта абсолютных ошибок; г) карта КСП, окно 7×7

На искаженном изображении (рис. 4.10б) шум гораздо лучше виден на однотонных областях, что отражается на карте КСП (рис. 4.10г) в виде более темных областей.

Для определения КСП всего изображения используется усредненное значение карты КСП.

4.4. Сравнительный анализ алгоритмов сжатия изображений относительно критериев ПОСШ и УИК

Проведем сравнительный анализ работы описанных выше алгоритмов сжатия изображений: JPEG, JPEG2000 и SPIHT.

В процессе анализа использовались три изображения с различной степенью детализации из тестового набора изображений, приведенные на рис. 4.11.

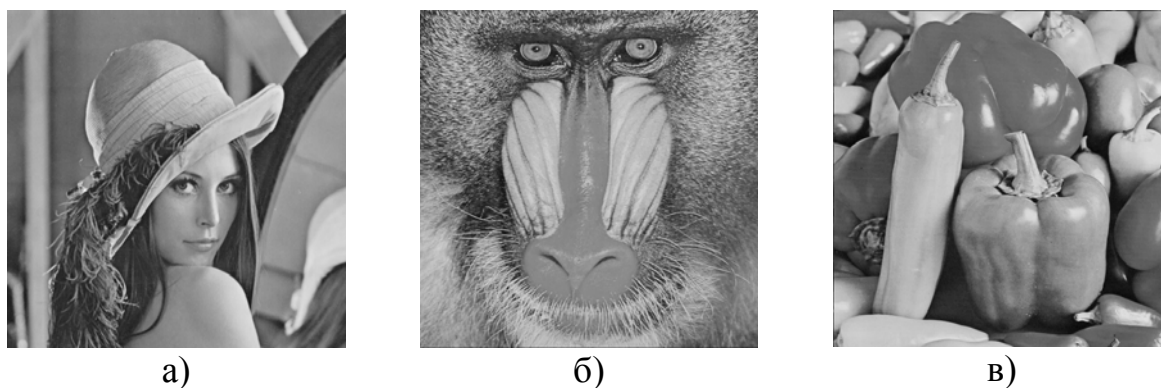


Рис. 4.11. Тестовые изображения: а) «Лена»; б) «Бабуин»; в) «Перцы»

В качестве критериев сравнения результатов работы использовались объективные метрики качества: ПОСШ сжатого изображения; оценки по шкале УИК.

На рис. 4.12. в качестве примера приведены зависимости объективных оценок качества восстановленного тестового изображения «Лена» от коэффициента сжатия K . Данные зависимости для других типов изображений не иллюстрируются ввиду их идентичности с приведенными.

В результате проведенного исследования можно сделать следующие выводы:

1. Для всех трех типов тестовых изображений наиболее высокие оценки как по шкале ПОСШ, так и по шкале УИК получились при использовании алгоритма JPEG2000.

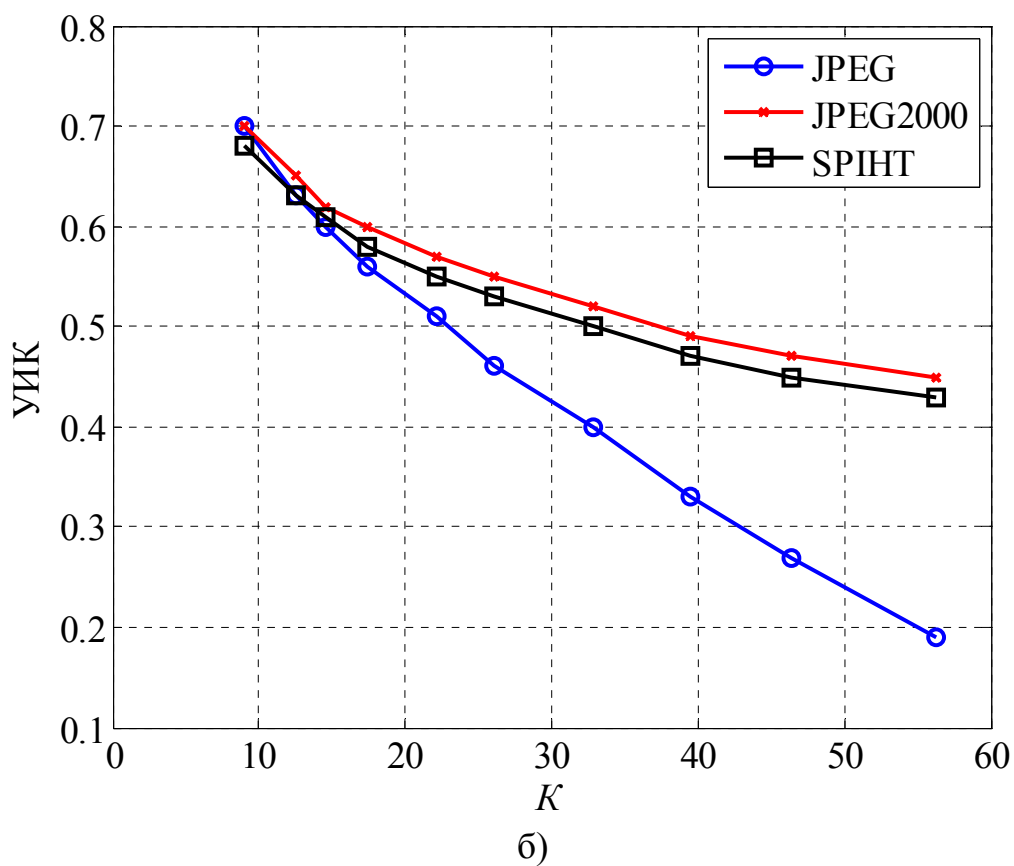
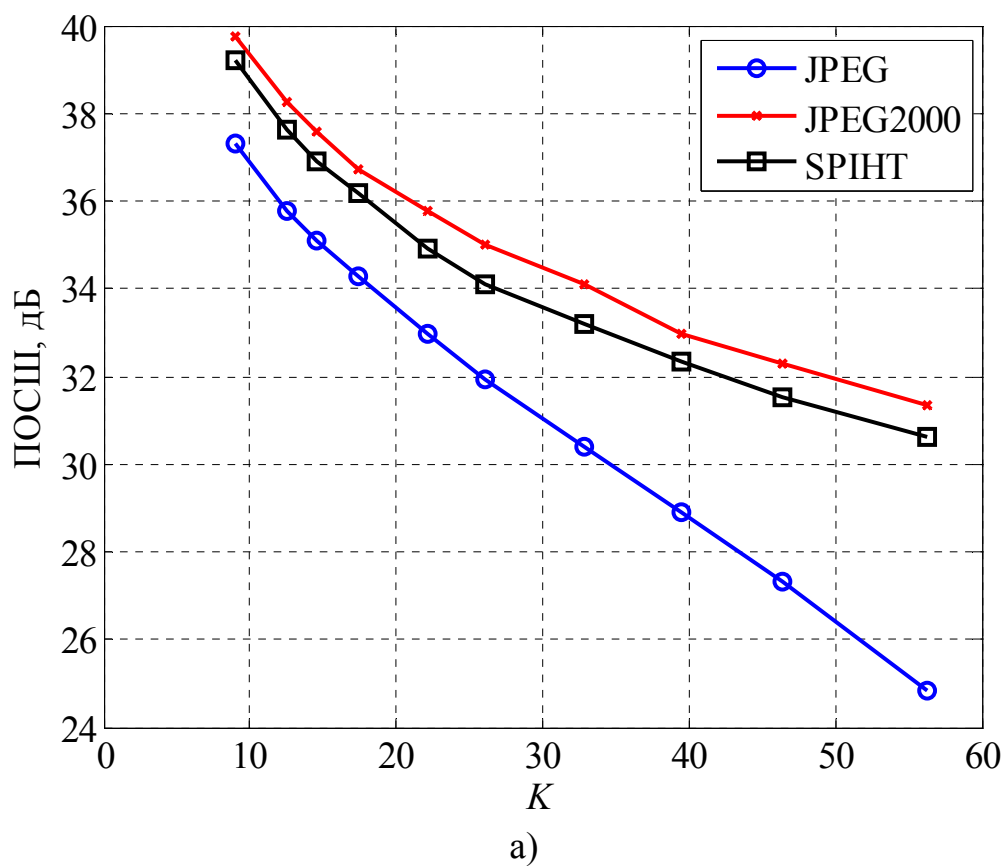


Рис. 4.12. Сравнение эффективности работы алгоритмов сжатия для изображения «Лена»: а) по шкале ПОСШ; б) по шкале УИК

2. Результаты моделирования показывают, что при одинаковых коэффициентах сжатия алгоритм SPIHT сопоставим с алгоритмом JPEG2000, оценки по ПОСШ и УИК различаются менее чем на 3%.

3. Проигрыш в качестве JPEG-изображений в диапазоне наиболее часто используемых на практике коэффициентов сжатия (2–20) более заметен по шкале ПОСШ, чем по шкале УИК.

4. С точки зрения оценки УИК при малом сжатии все алгоритмы предоставляют одинаковое качество, однако при высоком сжатии ($K \geq 20$) стандарт JPEG сильно уступает двум другим алгоритмам.

Для дополнительной проверки пункта 4 вышеизложенных выводов на рис. 4.13. представлены визуальные результаты сжатия изображения «Перцы». При малых коэффициентах сжатия разница между декодированными изображениями незаметна. Напротив, при больших коэффициентах сжатия (в данном случае $K = 46,7$) JPEG-изображение обладает ярко выраженной блочной структурой, что свидетельствует о потере большой части информации. Эффект размытия, вносимый в изображения алгоритмами JPEG2000 и SPIHT при данных степенях сжатия, менее заметен.

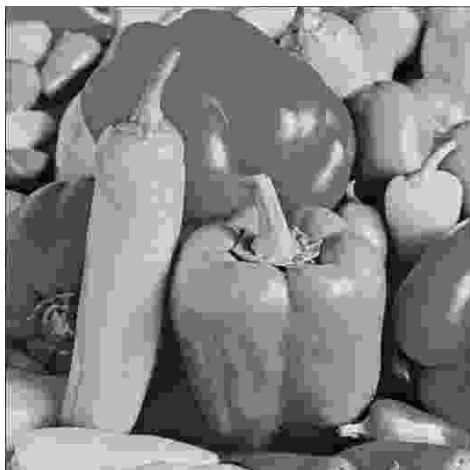
Проведем анализ алгоритмов сжатия изображений для трех типов тестовых изображений. Исследуя графики на рис 4.14–4.16, можно сделать следующие выводы:

1. Стандарт JPEG сжимает низкочастотные изображения с меньшими потерями качества, чем высокочастотные. Изображение «Бабуин» имеет много резких перепадов, поэтому плохо сжимается алгоритмом JPEG на высоких коэффициентах сжатия. В этом случае при делении на блоки нарушаются естественные границы.

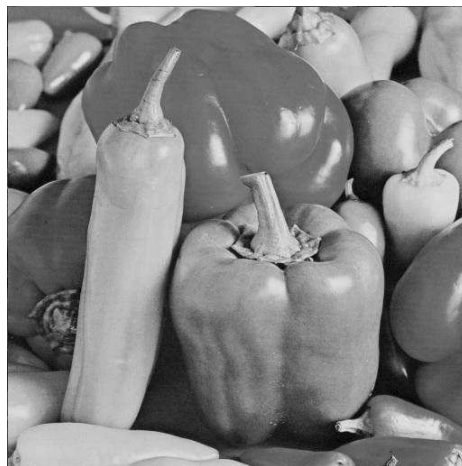
2. При сжатии изображения «Бабуин» коэффициенты, отвечающие за мелкие (особенно диагональные) детали, находятся при работе алгоритма SPIHT в нижнем правом углу плоскости битов. Иными словами, нижние уровни пространственно ориентированного дерева содержат коэффициенты, имеющие достаточно большие значения, что приводит к потере значительной части информации и, следовательно, к уменьшению ПОСШ.

3. На тестовых изображениях «Лена» и «Перцы» алгоритм SPIHT работает так же, как и алгоритм JPEG2000, из-за отсутствия в этих изображениях резких перепадов.

4. Стандарт JPEG2000 хорошо сжимает любые типы изображений, но на высоких коэффициентах сжатия качество лучше для низкочастотных изображений.



JPEG($K = 46,7$), ПОСШ = 27,32,
УИК = 0,25



JPEG($K = 9$), ПОСШ = 36,01,
УИК = 0,64



JPEG2000 ($K = 46,7$), ПОСШ = 32,27,
УИК = 0,42



JPEG2000 ($K = 9$), ПОСШ = 38,00,
УИК = 0,67



SPIHT ($K = 46,7$), ПОСШ = 31,46,
УИК = 0,40



SPIHT ($K = 9$), ПОСШ = 37,23,
УИК = 0,64

Рис. 4.13. Визуальный анализ алгоритмов сжатия JPEG, JPEG2000 и SPIHT для тестового изображения «Перцы»

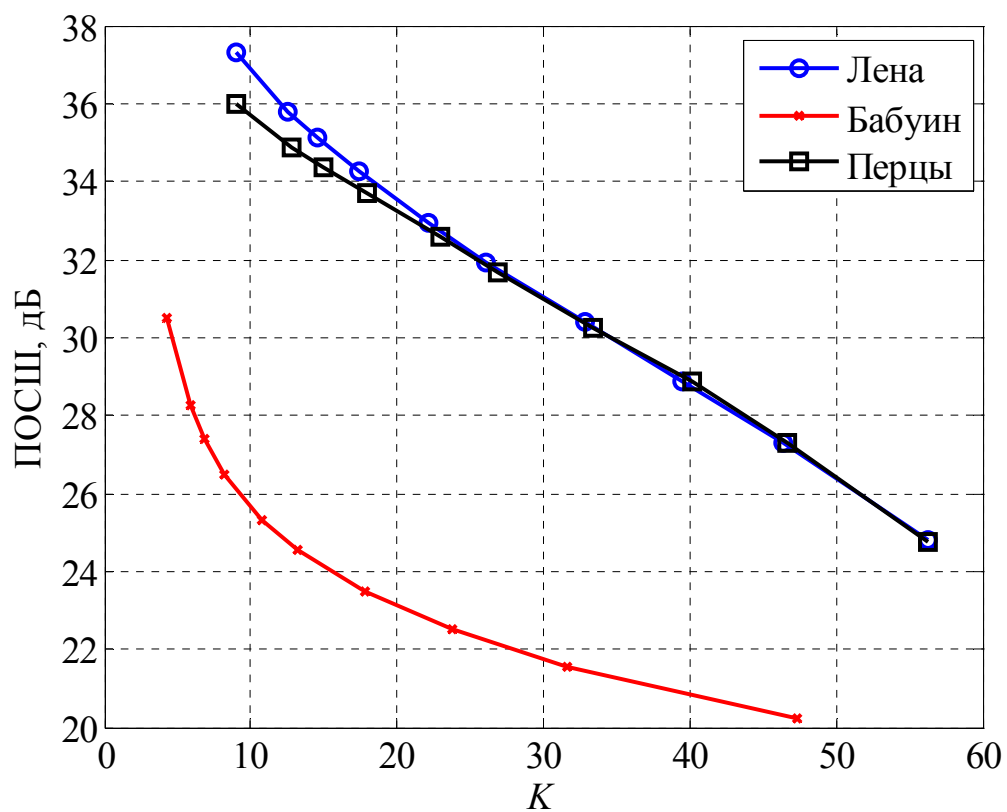


Рис. 4.14. Работа алгоритма JPEG для всех тестовых изображений

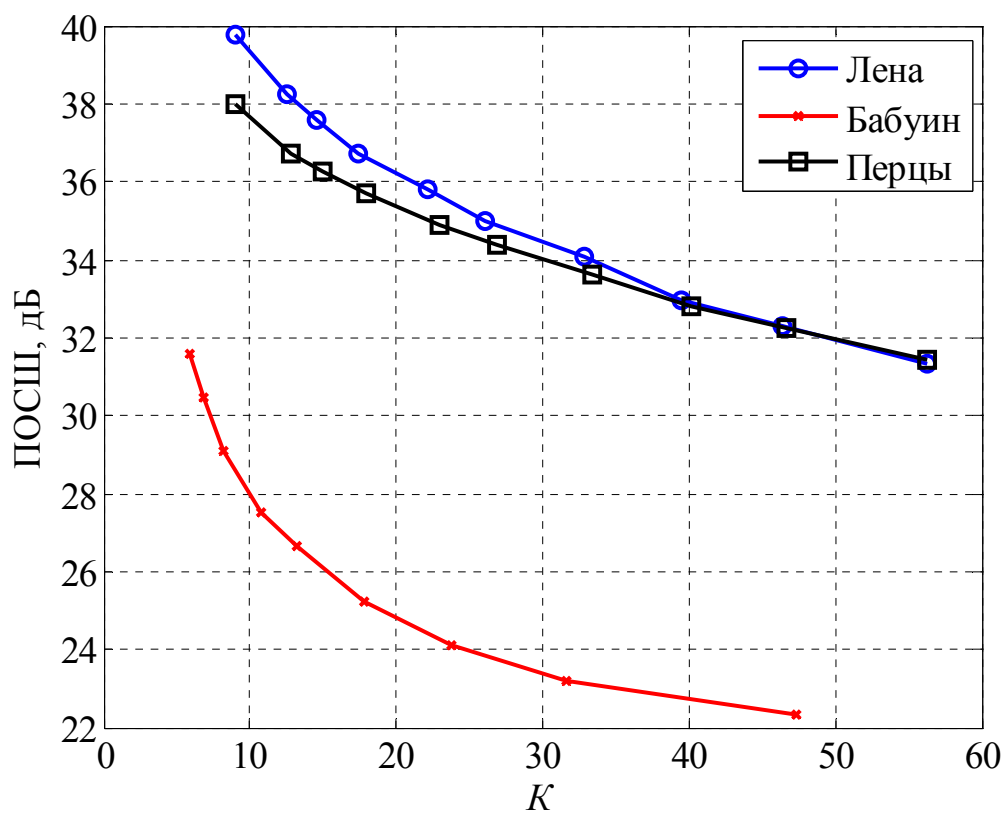


Рис. 4.15. Работа алгоритма JPEG2000 для всех тестовых изображений

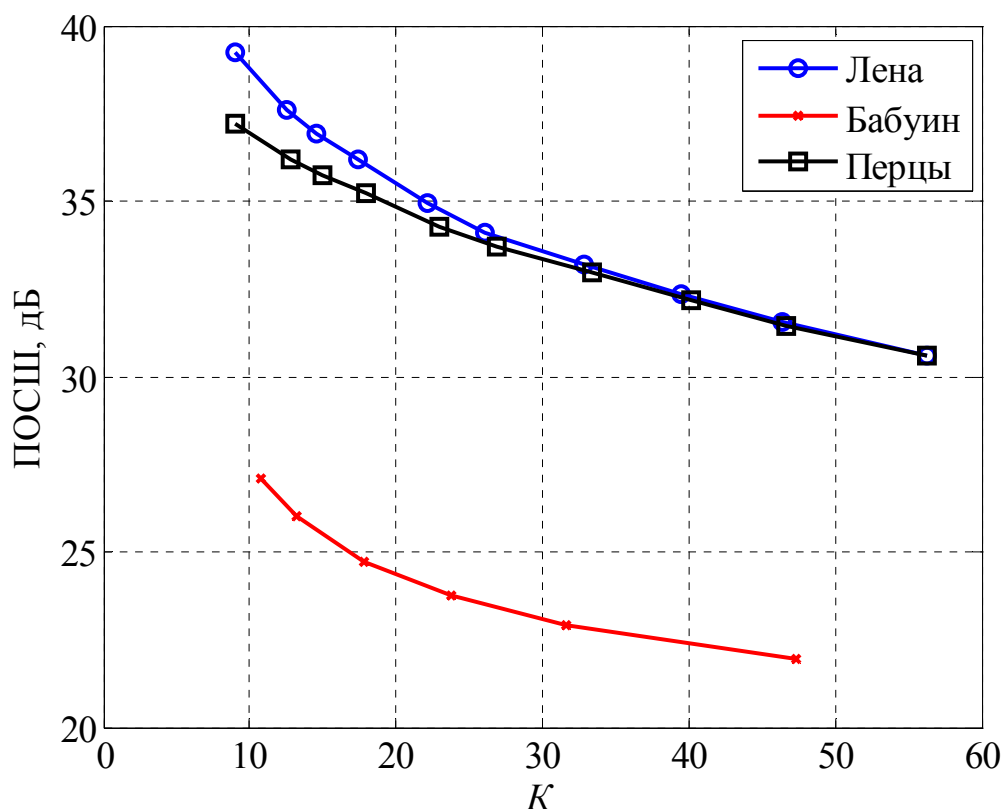


Рис. 4.16. Работа алгоритма SPIHT для всех тестовых изображений

Таким образом, проведенный анализ алгоритмов сжатия изображений относительно критериев ПОСШ и УИК позволяет сделать следующие выводы:

1. Стандарт JPEG проигрывает алгоритмам сжатия на основе вейвлет-преобразования 2–7 дБ по шкале оценок ПОСШ. Эта разница увеличивается с ростом степени компрессии и визуально различима начиная с $K = 25$.

2. Алгоритм SPIHT целесообразно использовать в случае сжатия малоконтрастных изображений и изображений без резких границ и мелких деталей (особенно диагональных). Для данного типа изображений можно добиться лучшей работы алгоритма путем совмещения его с энтропийным кодированием.

3. В общем случае предпочтительным для практического использования является стандарт JPEG2000, который позволяет добиваться хороших результатов вне зависимости от особенностей изображений.

5. ОСНОВНЫЕ СВЕДЕНИЯ О ЦИФРОВЫХ ВИДЕОСИГНАЛАХ

Процесс видеокодирования заключается в компрессии и декомпрессии цифрового видеосигнала. Цифровое видео представляет собой запись реальных сцен окружающего нас мира, снятых в некоторые последовательные моменты времени. Сцена фиксируется (сэмплируется) в каждый момент времени в виде кадра (образа всей визуальной сцены в этот момент времени) или полукадра (состоящего из строк кадров с четными или нечетными номерами пространственного сэмплирования). Сэмплирование по времени производится с некоторыми промежутками (например, через интервалы 1/25 или 1/30 секунды).

Типичная сцена из окружающего нас реального мира состоит из различных объектов, каждый из которых имеет свою форму, размеры, глубину, структуру, плотность и освещенность. Цвет и яркость естественной видеосцены могут достаточно плавно меняться в определенной степени внутри сцены («непрерывный тон»). Характеристики типичной естественной видеосцены (рис. 5.1), включает как пространственные характеристики (изменение текстуры или структуры внутри сцены, числа и формы объектов, цвета и т. п.), так и временные характеристики (движение объектов, изменение освещенности, перемещение видеокамеры или ее фокуса и т. д.).



Рис. 5.1. Кадр тестовой видеопоследовательности «Stockholm»

5.1. Пространственное и временное сэмплирование

Естественная видеосцена является в большой степени непрерывной по времени и пространственно однородной. Представление визуальной сцены в цифровой форме предполагает ее пространственное сэмплирование (обычно с помощью прямоугольной сетки в плоской проекции видеоизображения) и временное сэмплирование (в виде последовательности неподвижных изображений (кадров) или компонентов кадров, зафиксированных через регулярные промежутки времени) (рис. 5.2). Каждый пространственно-временной сэмпл (элемент изображения или пиксел) представляется в виде числа или множества чисел, которые обозначают яркость (светимость) и цвет данного сэмпла. Чтобы получить двумерное сэмплированное изображение, видеокамера фокусирует двухмерную проекцию на датчике или сенсоре, которым, например, может быть матрица прибора с зарядовой связью (ПЗС-матрица). В случае захвата (фиксации) цветного изображения каждая цветовая компонента выделяется особым цветным фильтром и проецируется на отдельную ПЗС-матрицу.

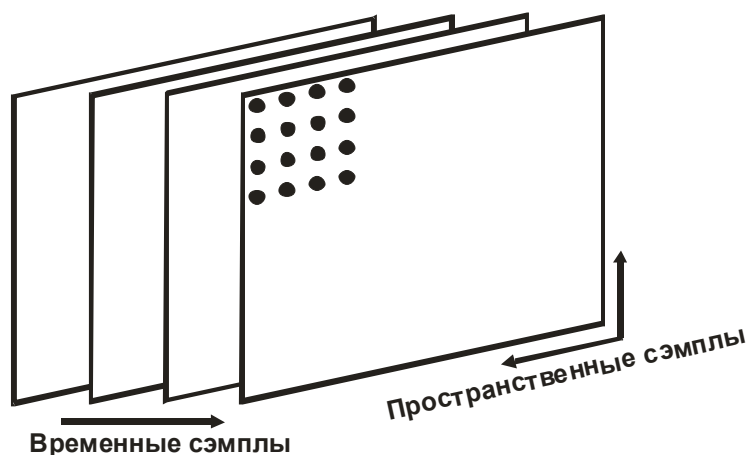


Рис. 5.2. Пространственное и временное сэмплирование

Выходом ПЗС-матрицы служит аналоговый видеосигнал – переменный электрический сигнал, представляющий видеоизображение. Сэмплирование сигнала по точкам пространства производит сэмплированное изображение или сэмплированный кадр, имеющий определенные числовые значения в соответствующих точках изображения. Наиболее популярным форматом сэмплированного изображения является прямоугольник, в котором точки сэмплирования располагаются на квадратной или прямоугольной решетке.

Движущееся видеоизображение захватывается (фиксируется) в виде прямоугольных «снимков» сигнала через периодические интервалы времени. Повторное проигрывание последовательности кадров производит эффект движения. Чем выше скорость временного сэмплирования (частота кадров), тем более гладким выглядит движение изображения снятой сцены, однако для это требуется зафиксировать и записать больше снимков (сэмплов). Частота кадров меньше 10 кадров в секунду иногда используется при видеокоммуникациях на низкой битовой скорости (при относительно малых объемах передаваемых данных). Однако при этом изображение движется резкими скачками и выглядит неестественным. Частота от 10 до 20 кадров в секунду является более типичной для видеокоммуникаций: изображение движется более сглаженно, однако его прерывистость остается заметной глазу, особенно при резких перемещениях некоторых частей сцены. Сэмплирование с частотой от 25 до 30 кадров в секунду является стандартным в телевидении (с чересстрочной разверткой для улучшения передачи движения); частота кадров от 50 до 60 в секунду производит совершенно гладкое движущееся изображение (платой при этом служит высокая скорость передачи данных).

5.2. Кадры и полукадры

Видеосигнал может быть просэмплирован в виде серии целых кадров (прогрессивное сэмплирование) или в виде последовательности чересстрочных полукадров (чересстрочное сэмплирование). В чересстрочной видеопоследовательности половина пикселей кадра (полукадр) фиксируется за каждый временной интервал сэмплирования. Полукадры состоят из нечетных или четных строк полного видеокадра, а чересстрочный видеоряд (рис. 5.3) состоит из последовательности полукадров, в каждом из которых находится половина информации целого видеокадра. Преимущество такого метода сэмплирования заключается в том, что имеется возможность передать в два раза больше полукадров за секунду, чем целых кадров в эквивалентной прогрессивной последовательности при той же общей скорости передачи данных. При этом наблюдается определенное сглаживание движения в кадре. Например, видеопоследовательность в системе PAL состоит из 50 полукадров в секунду, и движение в кадре выглядит существенно более плавным, чем в эквивалентной видеопоследовательности целых кадров, содержащей 25 кадров в секунду.

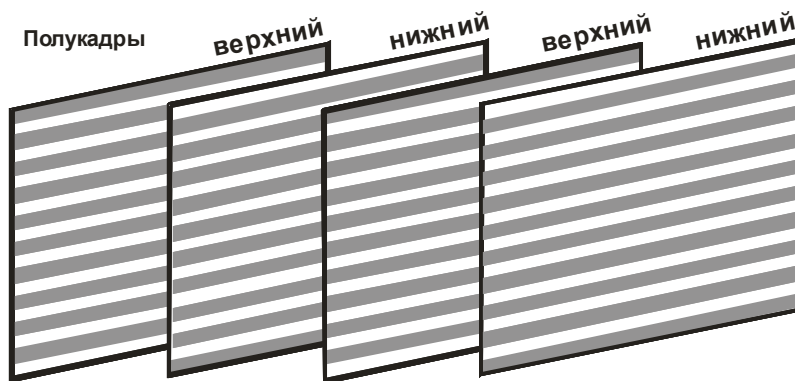


Рис. 5.3. Видеопоследовательность при чересстрочном сэмплировании

5.3. Цветовые пространства

Большинство цифровых видеоприложений приспособлены для отображения цветного видео, поэтому необходимо иметь определенный алгоритм для захвата и представления цветовой информации. Для монохромного изображения достаточно задать всего одно число для выражения яркости или светимости каждого пространственного пиксела. А цветное изображение требует не менее трех чисел на один пиксел для точной передачи его цвета. Метод, выбранный для представления яркости (или светимости) и цвета, определяется цветовым пространством.

5.3.1. Цветовое пространство RGB

В цветовом пространстве RGB пиксели цветного изображения представляются с помощью трех чисел, указывающих относительное соотношение красного (Red), зеленого (Green) и синего (Blue) цветов (три основные компоненты видимого света). Любой цвет можно получить с помощью комбинации красного, зеленого и синего цветов в соответствующей пропорции. На рис. 5.4 показаны красная, зеленая и голубая компоненты цветного изображения: красная компонента состоит из всех красных элементов, зеленая компонента образована всеми зелеными элементами, а синяя включает все синие. Захват изображения RGB основан на фильтровании красной, зеленой и голубой компонент видеосцены и их фиксировании с помощью отдельных сенсорных матриц. Цветные электронно-лучевые трубки и жидкокристаллические дисплеи отображают RGB-изображения, отдельно освещая красные, зеленые и голубые компоненты каждого пиксела в соответствии с интенсивностью каждого из них. Если

смотреть на экран с расстояния обычного зрителя, то различные компоненты сливаются в единый «правильный цвет».



Рис. 5.4. Красный, зеленый и синий компоненты цветного изображения

5.3.2. Цветовое пространство YC_bC_r

Известно, что органы зрения человека менее чувствительны к цвету предметов, чем к их яркости. В цветовом пространстве RGB все три цвета считаются одинаково важными, и они обычно сохраняются с одинаковым разрешением. Однако можно отобразить цветное изображение более эффективно, отделив яркость от цветовой информации и представив ее с большим разрешением, чем цвет.

Цветовое пространство YC_bC_r и его вариации (иногда его обозначают YUV) являются популярным методом эффективного представления цветных изображений. Буква Y обозначает яркостную компоненту, которая вычисляется как взвешенное усреднение компонент R , G и B по следующей формуле:

$$Y = k_r R + k_b B + k_g G,$$

где k – соответствующий весовой множитель.

Цветовая (хроматическая) информация может быть представлена компонентами *цветовых разностей* (хроматичностей), т. е. каждая компонента цветности – это разность между компонентами R , G , B и компонентой Y :

$$\begin{aligned} C_b &= B - Y, \\ C_g &= G - Y, \\ C_r &= R - Y. \end{aligned}$$

Полное описание цветного изображения задается величиной Y (яркостной компонентой) и тремя хроматическими разностями C_b , C_g и C_r , которые выражают разность между интенсивностью соответствующего цвета и средней светимостью каждого пикселя изображения. На рис. 5.5. показаны хроматические компоненты

(красный, зеленый и синий), соответствующие цветовым компонентам RGB на рис. 5.4. Здесь серый цвет обозначает близкую к нулю разность, более яркий соответствует положительной разности, а темный указывает на отрицательную разность. Хроматические компоненты имеют большие значения только там, где имеется существенное расхождение между цветовыми компонентами и яркостью изображения. Обратите внимание на большие значения синей и красной компонент.



Рис. 5.5. Компоненты C_r , C_g и C_b

Число $C_r + C_g + C_b$ является постоянным, поэтому только две из трех хроматических компонент необходимо хранить или передавать, поскольку третью компоненту всегда легко вычислить, зная две другие. В цветовом пространстве YC_bC_r используются компонента яркости (Y) и две хроматические компоненты (C_b и C_r). Преимущество пространства YC_bC_r по сравнению с RGB заключается в том, что компоненты C_b и C_r можно представлять с *меньшим разрешением*, чем Y , так как глаз человека менее чувствителен к цвету предметов, чем к их яркости. Это позволяет сократить объем информации, требуемый для представления хроматических компонент, без заметного ухудшения качества передачи цветовых оттенков изображения. Простой зритель не заметит никаких различий между изображением RGB и его представлением в цветовом пространстве YC_bC_r с огрубленным разрешением хроматических компонент. Использование большего разрешения для компоненты яркости по сравнению с хроматическими компонентами является простым, но весьма эффективным приемом при сжатии изображений.

RGB -изображение легко конвертировать в YC_bC_r для уменьшения объема сохраняемых или передаваемых данных. Однако перед тем, как отображать картинку на экране, необходимо сделать обратную конвертацию YC_bC_r в RGB . Соответствующие вычисления для прямого и обратного преобразования совершаются с помощью соотношений (5.1) и (5.2). Отметим, что множитель k_g получается из

соотношения $k_g + k_b + k_r = 1$, а величина компоненты G – вычитанием суммы C_b и C_r из Y .

$$Y = k_r R + (1 - k_b - k_r)G + k_b B,$$

$$C_b = \frac{0,5}{1 - k_b}(B - Y), \quad (5.1)$$

$$C_r = \frac{0,5}{1 - k_r}(R - Y),$$

$$R = Y + \frac{1 - k_r}{0,5} C_r,$$

$$G = Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} C_b - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} C_r, \quad (5.2)$$

$$B = Y + \frac{1 - k_b}{0,5} C_b.$$

Рекомендация ITU-T BT.601 предлагает следующие коэффициенты: $k_b = 0,114$ и $k_r = 0,229$. Используя эти значения в выражениях (5.1) и (5.2), получаем следующие широко распространенные формулы:

$$Y = 0,299R + 0,587G + 0,114B,$$

$$C_b = 0,564(B - Y),$$

$$C_r = 0,713(R - Y),$$

$$R = Y + 1,402C_r,$$

$$G = Y - 0,344C_b - 0,714C_r,$$

$$B = Y + 1,772C_b.$$

5.3.3. Форматы сэмплирования YC_bC_r

На рис. 5.6 показаны три образца сэмплирования для компонент Y , C_b и C_r , которые поддерживаются стандартами MPEG-4 и H.264. Формат 4:4:4 подразумевает, что все три компоненты (Y , C_b и C_r) имеют одинаковое разрешение и, следовательно, сэмплы всех компонентов присутствуют в каждом пикселе. Число в пропорции означает относительную долю каждой компоненты при сэмплировании в *горизонтальном* направлении, т. е. для каждой из четырех компонент яркости отбирается по четыре хроматические компоненты. Сэмплирование по формату 4:4:4 означает полную точность в передаче хроматических компонент. При сэмплировании по формуле 4:2:2 хроматические компоненты по вертикали имеют

одинаковое разрешение с яркостью, а по горизонтали они имеют половину от разрешения яркости (числа 4:2:2 означают, что на каждые четыре сэмпла яркости Y по горизонтали отбирается только две компоненты C_b и две компоненты C_r). Формат 4:2:2 используется для высококачественного цветного видео.

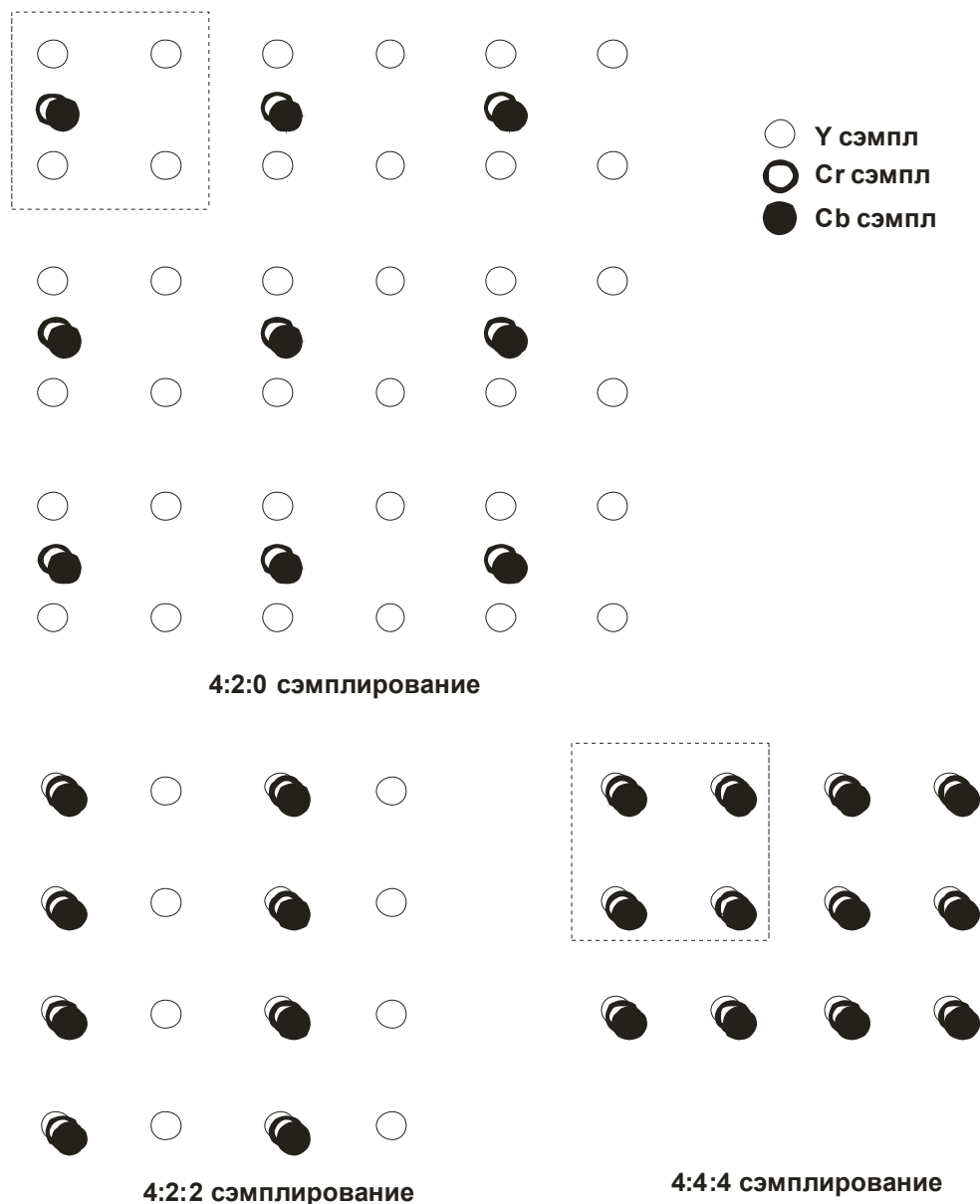


Рис. 5.6. Форматы 4:2:0, 4:2:2, 4:4:4 (прогрессивное сэмплирование)

В популярном формате сэмплирования 4:2:0 каждая компонента C_b и C_r имеет и по вертикали, и по горизонтали половину разрешения по сравнению с Y . Пропорция 4:2:0 выглядит несколько странной, поскольку эти числа не имеют обычной интерпретации, а само это выражение просто является данью исторической традиции, когда под этим «кодом» подразумевался именно этот формат сэмплирования, который отличается от форматов 4:4:4 и 4:2:2. Цветное

сэмплирование 4:2:0 широко используется во многих потребительских приложениях, таких как видеоконференции, цифровое телевидение и DVD-диски. Поскольку хроматические компоненты отбираются в четыре раза реже компонент яркости, то пространство 4:2:0 YC_bC_r требует в два раза меньше сэмплов по сравнению с форматом видео 4:4:4 (или $R:G:B$).

Формат сэмплирования 4:2:0 иногда называют «12 бит на пиксел». Чтобы понять такое наименование, достаточно рассмотреть группу из четырех пикселов (см. участки, обведенные пунктирными линиями, на рис. 5.6). При сэмплировании 4:4:4 требуется ровно 12 сэмплов для Y , C_b и C_r , т. е. $12 \times 8 = 96$ бит информации; в среднем $96 / 4 = 24$ бит на пиксел. А при использовании формата 4:2:0 будет необходимо лишь шесть сэмплов: четыре для Y и по одному C_b и C_r , т. е. $6 \times 8 = 48$ бит информации; в среднем $48 / 4 = 12$ бит на пиксел.

При чересстрочном сэмплировании по формату 4:2:0 компоненты Y , C_b и C_r полного видеокadra распределяются по двум полукадрам. На рис. 5.7 показан метод распределения сэмплов Y , C_b и C_r на пары чересстрочных полукадров, применяемый в стандартах MPEG-4 и H.264. Из этого рисунка ясно, что общее число сэмплов для двух полукадров равно числу сэмплов для полных кадров при прогрессивном сэмплировании.



Рис. 5.7. Распределение сэмплов 4:2:0 между верхним и нижним кадрами

6. ОСНОВЫ ВИДЕОКОДИРОВАНИЯ

При сжатии (кодировании) видео происходит уменьшение объема цифрового видеосигнала, который отображает это видеоизображение. «Сырой», несжатый цифровой видеосигнал требует огромной битовой скорости для его передачи в реальном масштабе времени (примерно 216 Мбит/с для представления несжатого видео стандартного аналогового телевизионного качества). Поэтому компрессия видео необходима для его практического хранения и передачи по цифровым сетям.

Для сжатия цифрового видео необходимо иметь две дополняющие друг друга системы: компрессор (кодер) и декомпрессор (декодер). Кодер преобразует источник видеоданных в сжатую форму (занимающую меньший объем) для дальнейшей передачи или хранения видео, а декодер делает обратное преобразование, возвращая сжатую форму видеоданных к ее исходному виду, пригодному для показа зрителю. Пару кодер / декодер принято называть «кодэк» (КОдер / ДЕКОдер) (рис. 6.1).



Рис. 6.1. Кодер / декодер

Методы видеосжатия с потерями основаны на удалении *субъективной* избыточности, т. е. тех элементов изображения, которые можно удалить без заметного влияния на зрительское восприятие видео.

Большинство методов видеокодирования используют оба типа избыточности, временную и пространственную, для достижения эффекта сжатия. Во временной области имеется значительная корреляция (сходство) между видеокадрами, зафиксированными в близкие моменты времени. Смежные по времени кадры (следующие по порядку друг за другом) имеют высокую степень корреляции, особенно при большой частоте кадров. В пространственной области также наблюдается высокая зависимость величин пикселей (сэмплов), лежащих близко друг к другу (рис. 6.2).



Рис. 6.2. Пространственная и временная корреляции кадров

Стандарты H.264 и MPEG-4 Visual (которые будут кратко описаны в разделе 7) имеют много общих черт. Оба стандарта подразумевают «модель» кодека, которая использует компенсацию движения на основе блоков, преобразование, квантование и энтропийное кодирование.

6.1. Видеокодек

Видеокодек кодирует исходное изображение или видеопоследовательность в сжатой форме, а также декодирует сжатую видеопоследовательность, производя цифровую видеокопию, которая или совпадает, или близка к исходной видеопоследовательности. Если декодированная последовательность совпадает с исходной, то процесс кодирования называется кодированием без потерь. Если же эти последовательности отличаются друг от друга, то процесс называется кодированием с потерями.

Видеокодер (рис. 6.3) состоит из трех основных функциональных единиц: *временной модели*, *пространственной модели* и *энтропийного кодера*. На вход временной модели подается несжатый цифровой видеосигнал. Временная модель стремится

сократить временную избыточность, используя схожесть между последовательными видеокадрами, обычно строя прогноз для следующего кадра по соседним кадрам последовательности. В стандартах MPEG-4 и H.264 прогноз формируется по одному или нескольким предыдущим или будущим кадрам, при этом делается коррекция расхождения между кадрами (предсказание компенсации движения). Выходом временной модели служит остаточный кадр (производимый вычитанием кадра-прогноза из подлинного текущего кадра) и некоторое семейство числовых параметров модели, обычно множество векторов движения, описывающих, как это движение было скомпенсировано.

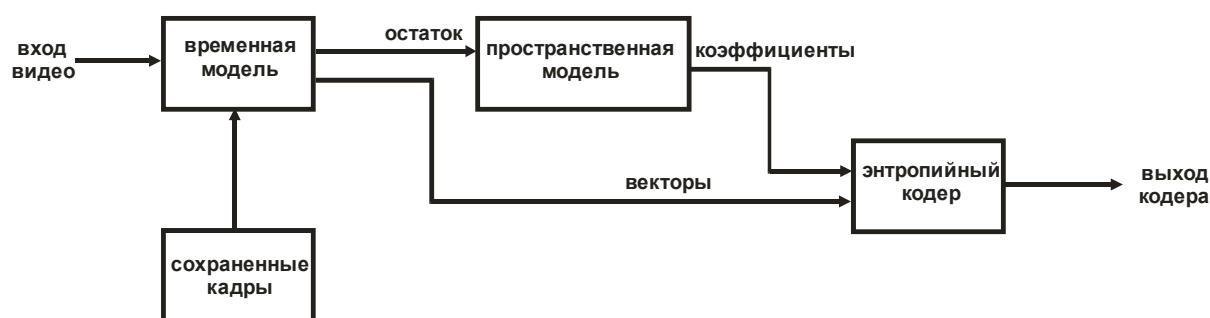


Рис. 6.3. Блок-схема видеокодера

Остаточный кадр является входом для пространственной модели, которая использует схожесть или подобие соседних пространственных сэмплов этого кадра, сокращая тем самым пространственную избыточность. В стандартах MPEG-4 и H.264 это осуществляется применением специальных преобразований к остаточному кадру и квантованием результата. Преобразование переводит сэмпл в другую область, в которой он представляется в виде последовательности коэффициентов используемого преобразования. Коэффициенты квантуются методом удаления несущественных значений; при этом оставляется небольшое число существенных коэффициентов, которые обеспечивают более компактное представление остаточного кадра. Выходом пространственной модели выступает семейство квантованных коэффициентов преобразования.

Параметры временной модели (векторы движения) и пространственной модели (коэффициенты преобразования) сжимаются энтропийный кодером. При этом удаляется статистическая избыточность данных (например, кодируя часто встречающиеся векторы и коэффициенты более короткими двоичными кодами) и формируется сжатый битовый поток или файл, который можно передавать по сетям или хранить на носителях

цифровых данных. Файл сжатой последовательности состоит из закодированных векторов движения, закодированных остаточных коэффициентов и некоторого информационного заголовка.

Видеодекодер реконструирует видеокадр по сжатому битовому потоку данных. Коэффициенты и векторы движения декодируются энтропийным декодером, после чего пространственная модель декодирует полученные данные, формируя некоторую версию остаточного кадра. Декодер использует параметры векторов движения вместе с одним или несколькими ранее декодированными кадрами для построения прогноза текущего кадра, а сам кадр реконструируется добавлением остаточного кадра к этому кадру-прогнозу.

6.2. Временная модель

Целью временной модели является удаление избыточности между передаваемыми кадрами. Для этого создается кадр-прогноз, который вычитается из текущего кадра. Выходом этого процесса является остаточный кадр (разность кадров). При этом чем лучше прогноз, тем меньше энергии содержится в остаточном кадре. Остаточный кадр кодируется и посылается декодеру, который воспроизводит кадр-прогноз по некоторым уже построенным кадрам, складывает его с остаточным кадром и таким образом восстанавливает текущий кадр. Кадр-прогноз формируется по одному или нескольким прошлым или будущим кадрам («ссылочным кадрам»). Как правило, точность прогноза может быть улучшена компенсированием движения между ссылочными кадрами и текущим кадром.

6.2.1. Прогноз по предыдущему видеокадру

Простейший метод временного прогнозирования основан на предыдущем кадре, который и служит кадром-прогнозом текущего кадра. Два последовательных кадра некоторой видеопоследовательности показаны на рис. 6.4 и 6.5. Кадр 1 является прогнозом кадра 2, и остаточный кадр, полученный вычитанием прогноза (кадра 1) из текущего (кадра 2), показан на рис. 6.6. На этом рисунке нейтральный серый цвет соответствует нулевым разностям пикселей, а более яркий или более темный оттенок серого цвета соответствует положительным или отрицательным разностям. Видно, что при таком простейшем прогнозе остаточный кадр содержит много энергии (на что указывают яркие и темные области кадра), а это означает, что в нем остался значительный объем информации,

которую можно будет сжимать после временной модели. Много оставшейся энергии образуется из движения объектов в кадре, т. е. лучший прогноз можно построить на основе *компенсации* движения между двумя этими кадрами.



Рис. 6.4. Кадр 1



Рис. 6.5. Кадр 2

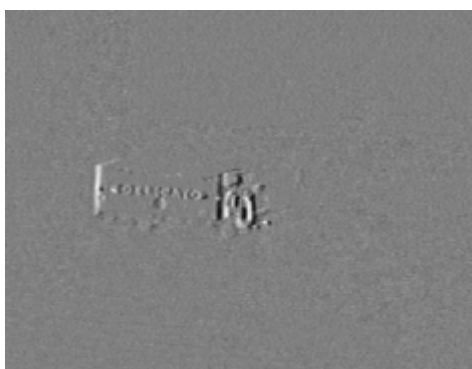


Рис. 6.6. Разность кадров 1 и 2

6.2.2. Изменения вследствие движения

Отличия между кадрами могут появиться из-за движения объектов (не меняющих свою форму, например автомобилей, и деформируемых, например рук или ног прохожих), перемещения видеокамеры (панорамная съемка, наклоны, наезды, изменения фокуса, повороты и т. п.), изменения перекрытия объектов (например,

части сцены, закрытой движущимся предметом), изменения освещенности и в силу многих других причин. За исключением изменений перекрытия и освещенности объектов съемки эти отличия соответствуют перемещению групп пикселей по кадру. Имеется возможность проследить траекторию каждого пикселя между последовательными кадрами. Порождаемое при этом поле траекторий пикселей называется *оптическим потоком*. Однако этот метод не очень практичен для компенсации движения по ряду причин. Точное вычисление оптического потока является довольно долгим (более точные методы используют итеративные процедуры для каждого пикселя), кроме того, придется пересылать все векторы оптического потока декодеру для декодирования и восстановления кадра-прогноза (в результате чего повышается объем пересылаемой информации, сводящий на нет преимущества маломощного остаточного кадра).

6.2.3. Оценка и компенсация движения на основе блоков

На практике широко используется метод компенсации движения, который компенсирует перемещение прямоугольных областей или «блоков» текущего кадра. При этом выполняется следующая процедура для каждого фиксированного блока, состоящего из $M \times N$ пикселей обрабатываемого кадра.

1. Поиск на ссылочном кадре (предыдущем или следующем, ранее закодированном и переданном декодеру) «подходящего» блока из $M \times N$ пикселей. Это делается путем сравнения фиксированного $M \times N$ -блока с некоторыми или со всеми $M \times N$ -блоками области поиска (обычно она представляет собой некий регион с центром в середине этого выбранного блока) и нахождения «самого подходящего» блока. Популярным критерием схожести блоков служит энергия остатка, получаемого вычитанием блока-кандидата из фиксированного $M \times N$ -блока, т. е. выбирается блок-кандидат, минимизирующий энергию остатка. Этот процесс поиска подходящего блока называется *оценкой движения*.

2. Выбранный кандидат становится прогнозом текущего $M \times N$ -блока и его вычитают из этого блока для получения остаточного $M \times N$ -блока (*компенсации движения*).

3. Остаточный блок кодируется и передается декодеру, и декодер получает координаты вектора смещения текущего блока по отношению к позиции блока-кандидата (*вектор движения*).

Декодер использует вектор движения для нахождения блока-прогноза, декодирует остаточный блок и складывает его с прогнозом для реконструкции версии исходного блока.

Компенсация движения на основе блоков очень популярна по многим причинам. Этот метод достаточно прост и легко поддается программной реализации. Он хорошо вписывается в прямоугольный формат кадров, а также в процедуры преобразования изображений на основе блоков (например, в дискретное косинус-преобразование). Компенсация движения определяет достаточно приемлемую временную модель для многих видеопоследовательностей. Имеется, однако, ряд недостатков. Например, «реальные» объекты редко имеют четкие прямоугольные границы. Объекты часто перемещаются на нецелое число пиксельных позиций между кадрами, и многие типы перемещений трудно скомпенсировать, используя методы на основе блоков (например, деформацию объектов, их повороты, изменение масштаба, т. е. приближение или удаление объектов от видеокамеры, сложные движения, такие как клубы дыма). Несмотря на эти недостатки, компенсация движения на основе блоков является базой временных моделей во всех современных стандартах цифрового видеокодирования.

6.2.4. Прогноз макроблока с компенсированным движением

Макроблок, соответствующий области из 16×16 пикселей на кадре, является базисной единицей для прогноза с компенсацией движения во многих важных визуальных кодовых стандартах, включая MPEG-1, MPEG-2, MPEG-4 Visual, H.263 и H.264. Для исходного видеоматериала в формате 4:2:0 структура макроблока показана на рис. 6.7. Область из 16×16 пикселей исходного кадра состоит из 256 сэмплов яркости (сгруппированные в 4 блока сэмплов 8×8), 64 синих хроматических компонент (один блок 8×8) и 64 красных хроматических компонент (8×8), что дает ровно шесть блоков размером 8×8 . Стандарты MPEG-4 Visual и H.264 оперируют с кадрами с помощью единиц макроблоков.

Оценка движения. Оценка движения макроблока состоит из поиска области пикселей 16×16 на ссылочном кадре, который похож на текущий макроблок. Ссылочным кадром является уже декодированный кадр, который хронологически может идти раньше или позже текущего. Область поиска на ссылочном кадре имеет центр в середине положения текущего макроблока. Найденный блок 16×16 пикселей, имеющий наименьшее расхождение с текущим макроблоком, выбирается в качестве «подходящего» макроблока.

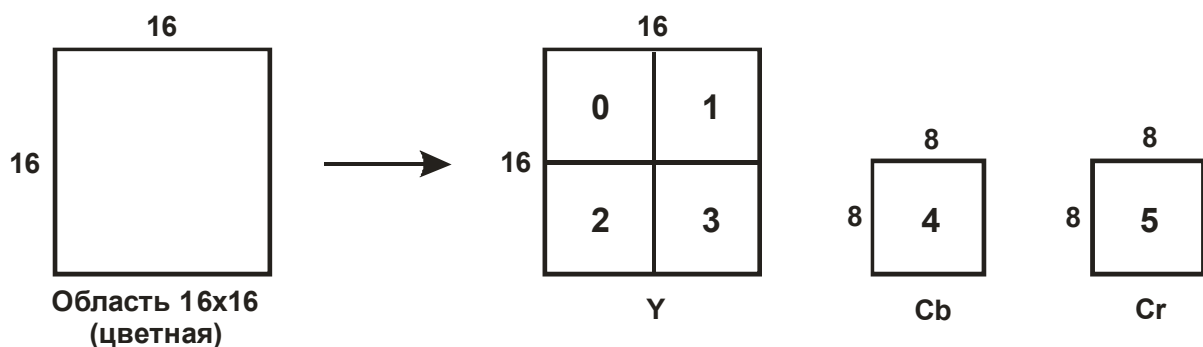


Рис. 6.7. Макроблок 4:2:0

Компенсация движения. Выбранный «наилучший» макроблок на ссылочном кадре вычитается из текущего блока, и их разность, называемая *остаточным макроблоком* (яркости и хроматичности), кодируется и передается вместе с вектором движения, задающим позицию наилучшего макроблока по отношению к текущему макроблоку. Внутри кодера остаточный макроблок кодируется и декодируется, а затем добавляется к наилучшему макроблоку для формирования реконструированного макроблока, который сохраняется в ссылочном буфере для использования в будущих прогнозах компенсации движения.

Существует много вариаций базового процесса оценки движения и компенсации движения. Ссылочным кадром может быть предыдущий кадр (в хронологическом порядке), будущий кадр или комбинация двух или более ранее декодированных кадров. Если в качестве ссылочного выбран будущий кадр, то необходимо его закодировать до текущего кадра (т. е. кадры должны быть закодированы не в хронологическом порядке). Там, где имеется существенное расхождение между ссылочным и текущим кадрами (например, при смене сцен), можно кодировать макроблок без компенсации движения. Таким образом, кодер выбирает или моду intra (кодирование без компенсации движения), или моду inter (кодирование с компенсацией движения) для каждого макроблока. Движущиеся объекты на видеосцене редко следуют строго вдоль границ блоков пикселей 16×16, поэтому более эффективным может быть использование переменных размеров блоков для оценки и компенсации движения. Объекты могут также перемещаться на нецелое число пикселей между кадрами (например, на 2,78 пикселей вместо 2,0 в горизонтальном направлении). Поэтому лучший прогноз можно построить интерполяцией ссылочного кадра по позициям подпикселей перед поиском наилучшего положения прогноза.

6.2.5. Размер блока компенсации движения

Результаты моделирования показывают, что чем меньше размер блока компенсации движения, тем лучше результат этой компенсации. Однако уменьшение размера блока влечет за собой увеличение сложности (необходимо совершить больше операций поиска) и числа векторов движения, необходимых для декодирования. Посылка каждого нового вектора движения требует дополнительных бит, а увеличение числа векторов движения сведет на нет выигрыш от сокращения энергии остаточного кадра. Эффективный компромисс заключается в приспособлении размеров блоков к характеристикам изображений, т. е. можно выбирать большие размеры блоков для ровных однородных областей кадра, а для областей со многими деталями и со сложным движением следует уменьшать размер блоков. Стандарт H.264 использует адаптивный механизм для выбора размеров блоков компенсации движения. Стандарт MPEG-4 включает в себя ряд инструментов, реализующих компенсацию движения на основе областей.

7. СТАНДАРТЫ MPEG-4 и H.264

Для успешной работы со стандартами сжатия видеоданных полезно понимать процесс их создания. В этом разделе будет рассмотрена роль исследовательских групп ISO/IEC MPEG (Moving Picture Experts Group) и ITU-T VCEG (Video Coding Experts Group) при разработке стандартов MPEG-4 и H.264. Мы обсудим механизмы выбора тех или иных параметров стандартов, а также рассмотрим движущие силы (технические и коммерческие), стоящие за развитием данных технологий, постараемся объяснить, как нужно понимать стандарты и как следует извлекать из них полезную информацию. Будут рассмотрены ключевые алгоритмы каждого из указанных стандартов и их специфические особенности. Также будет проведено краткое описание стандартов сжатия предыдущего поколения – MPEG-1, MPEG-2, H.261 и H.263.

7.1. Развитие стандартов

За разработку, поддержку и обновление семейства международных стандартов ISO/IEC 14496 (MPEG-4) отвечает группа экспертов по движущимся изображениям ISO/IEC MPEG. Разработка стандарта H.264 (известного также под наименованиями MPEG-4 Part 10, Advanced Video Coding и H.26L) была обеспечена на основе совместной работы группы MPEG и рабочей группы ITU-T VCEG.

Группа MPEG разработала в прошлом весьма успешные стандарты MPEG-1 и MPEG-2 для кодирования видео и аудио. В ее активе также находится разработка стандарта MPEG-7, стандартизация MPEG-21. Группа VCEG разработала первый широко используемый стандарт для видеотелефонии (H.261), за которым последовало создание H.263 и инициация разработки ранней стадии проекта H.26L. Эти две группы создали совместную команду JVT (Joint Video Team) для завершения проекта H.26L и придания ему формы международного стандарта (H.264/MPEG-4 Part 10), который был опубликован одновременно ISO/IEC и ITU-T.

7.1.1. ISO MPEG

Формально MPEG – это 11-я группа 29-го подкомитета первого объединенного технического комитета JTC1 (Joint Technical Committee 1) и ее официальное название: ISO/IEC JTC1/SC29/WG11.

Группа MPEG ответственна за разработку стандартов для сжатия, обработки и представления движущихся изображений и

звука. Она разработала целую серию важных стандартов, начиная с MPEG-1 (сжатие видео и аудио для CD-проигрывателей), за которым последовал MPEG-2 (хранение и передача изображения и звука «телевизионного качества»). Стандарт MPEG-4 (кодирование аудиовизуальных объектов) является самым новейшим стандартом, работающим с аудиовизуальным кодированием. Два других проекта стандартизации (MPEG-7 и MPEG-21) имели дело с представлением мультимедийного контента и общей мультимедийной среды. Группа MPEG пользуется заслуженной известностью за ее вклад в сжатие видео и звука. Стандарт MPEG-2 стал очень популярным благодаря применению в цифровом телевидении и в DVD-видео, а стандарт кодирования звука MPEG Layer 3 (более известный, как MP3) активно используется для хранения и распространения разнообразного звукового и музыкального контента.

Эксперты для группы MPEG набираются из различных коммерческих компаний, исследовательских организаций, институтов, имеющих мировую известность. Членство в группе MPEG и участие в ее конференциях ограничено рамками делегаций от национальных корпораций по стандартам (National Standards Bodies). Компания или институт, желающие участвовать в группе MPEG, должны обратиться в национальные корпорации по стандартам. Присоединение к корпорации по стандартам, поддержка национальных конференций и участие в международных встречах группы MPEG – все это требует значительных материальных и временных затрат. Однако данная деятельность открывает доступ к приватным документам группы MPEG (включая возможность ознакомления с черновыми вариантами стандартов до их официальной публикации, что дает шанс захватить лидерство на потенциальном рынке новых технологий), а также возможность влияния на формирование будущих стандартов.

Группа MPEG продвигает свою работу во время регулярных встреч, которые проходят четыре раза в год. Обычно такие встречи длятся пять дней (с понедельника по пятницу), включая рабочие встречи технических групп, предваряющие основную повестку дня. Успешные и перспективные инициативы одобряются и, если необходимо, включаются в процесс стандартизации. Результаты встреч оформляются в виде комплекта входных и выходных документов. Входные документы включают в себя результаты работы подгрупп на предварительных встречах, предложения по добавлениям или изменениям в разрабатываемые стандарты, а также предложения от других корпораций. Выходные документы включают отчеты конференций и черновые версии стандартов.

7.1.2. ITU-T VCEG

Организация ITU-T разрабатывает стандарты и рекомендации для телекоммуникационной области. Подгруппа 16 ITU-T отвечает за мультимедийные сервисы, системы и терминалы. Рабочий сектор 3 подгруппы 16 занимается медийным кодированием, а проект, послуживший основой для стандарта H.264, был описан в вопросе 6. Таким образом, официальное название интересующей нас группы – VCEG ITU-T SG16 Q.6. Ранее эта рабочая группа была известна как группа экспертов по кодированию при малой скорости (LBCEG, Low Bitrate Coding Experts Group).

Группа VCEG разработала ряд стандартов, относящихся к передаче видео в телекоммуникационных и компьютерных сетях. За стандартом H.261 для видеоконференций последовал более эффективный стандарт H.263. Позднее появились его продвинутые версии (известные под неофициальными обозначениями H.263+ и H.263++), которые существенно расширили возможности H.263. Новый проект по стандартизации, известный под кодом H.26L, вылился в разработку, оформление и публикацию рекомендации H.264. Начиная с 2001 года эта работа выполнялась группой VCEG в кооперации с группой MPEG. Новый стандарт сжатия видео – AVC (Advanced Video Coding) был опубликован совместно под именем ITU-T H.264 и ISO/IEC MPEG-4 Part 10.

Членство в VCEG открыто для всех заинтересованных сторон (при условии одобрения председателем). Члены группы VCEG встречаются каждые три месяца для рассмотрения новых начинаний и предложений, некоторые из них включаются в черновые проекты стандартов. В отличие от группы MPEG, входные и выходные документы VCEG (а теперь и JVT) являются общедоступными.

7.1.3. JVT

Объединенная команда по видео – JVT (Joint Video Team) – состоит из членов двух групп: ISO/IEC JTC1/SC29/WG11 (MPEG) и ITU-T SG16 Q.6 (VCEG). JVT была образована из-за необходимости группы MPEG иметь расширенный инструментарий для видеокодирования. Ключевой механизм кодирования в MPEG-4 Part 2 основан на довольно старой технологии (исходный стандарт H.263 был опубликован в 1995 году). В силу возросшей мощности процессоров и в связи с развитием исследований по видеокодированию стало ясно, что возможен прогресс в эффективности видеокодирования. После изучения нескольких

конкурирующих технологий, в июне 2001 года стало очевидным, что тестовая модель кодека H.26L является наилучшим выбором, удовлетворяющим всем требованиям MPEG. Была достигнута договоренность между группами MPEG и VCEG о создании команды JVT для завершения проекта H.26L. Встречи JVT происходят одновременно со встречами MPEG и/или VCEG, и соответствующие рекомендации и материалы направляются обеим родительским группам. Главной целью JVT было опубликование рекомендации H.264/MPEG-4 Part 10. После того, как работа над стандартом была завершена, внимание группы переключилось на расширение поддержки цветовых пространств и повышение точности сэмплирования.

7.1.4. История развития

В табл. 7.1 отражены важные этапы в развитии стандартов MPEG-4 и H.264.

Таблица 7.1

История развития MPEG-4 и H.264

| | |
|-------------|---|
| 1993 | Запуск проекта MPEG-4. Получены первые результаты по проекту H.263 |
| 1995 | Приглашение MPEG-4 представлять предложения по эффективному видеокодированию и контекстно-зависимым функциям. H.263 выбран ядром инструментов для обработки видео |
| 1998 | Приглашение представлять предложение для H.26L |
| 1999 | Публикация стандарта MPEG-4. Определение начальной тестовой модели TM1 для H.26L |
| 2000 | Приглашение MPEG представлять предложения по инструментам расширенного кодирования |
| 2001 | Публикация стандарта MPEG-4 Edition 2. H.26L выбран стержнем для MPEG-4 Part 10. Сформирована команда JVT |
| 2002 | Поправки 1 и 2 (профили студийного и потокового видео) к MPEG-4 Edition 2. Зафиксирован технический контент для H.264 |
| 2003 | Опубликован стандарт H.264/MPEG-4 Part 10 |

Изначально целью стандарта MPEG-4 было гибкое и эффективное обновление более ранних стандартов MPEG-1 и MPEG-2. Позднее, в 1994 году, стало ясно, что стандарт H.263 способен обеспечить более производительную и перспективную технологию сжатия. Поэтому группа MPEG переориентировала свои интересы и решила охватить кодирование и функциональность на основе объектов и сделать это существенным элементом нового стандарта MPEG-4. К моменту завершения проекта MPEG-4 (1998–1999 гг.) рабочая группа ITU-T начала разработку нового стандарта видеокодирования под кодовым названием H.26L (буква L обозначает Long Term). Развитие тестовой модели H.26L было принято за основу для предложенного в 2001 году стандарта MPEG-4 Part 10.

7.1.5. Процесс разработки стандартов

Разработка стандарта для видеокодирования проходит через ряд этапов. Сначала рабочий план согласуется со множеством функциональных и производительных особенностей. Для того чтобы сделать выбор базовой технологии стандарта, на втором этапе могут быть организованы конкурсные испытания, в которых заинтересованным сторонам предлагается представить для оценивания решения поставленных тестовых задач. Для стандартов MPEG-4 и H.264 в качестве базовой была выбрана технология видеокодирования на основе поблочной компенсации движения с последующим преобразованием и квантованием остаточных блоков. После принятия решения о ключевом алгоритме стандарта требуются дополнительные большие усилия по его детальной разработке. Алгоритмы, методы и конкретные функции предлагаются компаниями и организациями, имеющими определенный интерес по разработке новых стандартов, и передаются техническим группам по стандартизации MPEG, VCEG или JVT для испытания и оценивания. На каждой встрече рабочей группы происходит обсуждение всех поступивших разработок, которые могут приниматься или отклоняться экспертами. В результате такой работы происходит существенное усовершенствование будущего стандарта.

Разрабатывается программная модель («справочная программа»), в которой реализуются согласованные на данный момент функции. Одновременно с этим пишется документация по этой программе («тестовая модель»). Справочная программа и тестовая модель проверяются и дорабатываются, и каждый обновленный программный блок добавляется в стандарт. В какой-то

момент группа стандартизации решает, что документы тестовой модели уже достаточно созрели и их можно внедрять в черновую версию стандарта. После дальнейших улучшений и серии голосований документы публикуются в виде готового международного стандарта.

Окончательная доводка международного стандарта достигается в результате процесса, в котором участвуют сотни специалистов (членов группы по стандартизации), после многих модификаций и компромиссов. Окончательный документ формируется под воздействием многих движущих сил. С технической стороны вклад в работу групп MPEG и VCEG делается ведущими исследовательскими институтами, университетами и компаниями из сферы видеокодирования. Группы стандартизации стараются освоить и внедрить все лучшее, что известно исследователям на данном этапе развития технологий, для достижения требуемого сжатия и производительности при существующих вычислительных ограничениях. С коммерческой точки зрения исходный рабочий план группы стандартизации должен предвидеть и предвосхищать технологические требования будущих практических приложений. Технические предложения должны быть оценены с точки зрения эффективности затрат на их разработку и внедрение. Многие компании, представленные на конференциях групп стандартизации, весьма заинтересованы в результатах процесса стандартизации. В некоторых случаях конкретные предложения компаний основываются на их собственных патентах, которые могут принести большой доход от лицензирования при внедрении в международный стандарт. Поэтому группа JVT просила у своих членов обязательно идентифицировать любые патентованные технологии в поступающих предложениях, для того чтобы быть уверенной в том, что основной профиль стандарта H.264 может быть реализован без отчислений держателям патентов, без чего трудно рассчитывать на широкое распространение стандарта.

С практической точки зрения команды стандартизации работают в очень узких временных рамках. Поэтому часто бывает необходим определенный компромисс по некоторым аспектам рабочего плана для завершения всего проекта в установленный срок, поскольку очень важно иметь уверенность в том, что стандарт будет конкурентоспособным при его реализации. В некотором роде стандартизация MPEG-4 и H.264 походила на состязание во времени с другими коммерческими (нестандартными) системами и пакетами кодирования изображений и видео, которые продолжают непрерывно разрабатываться и совершенствоваться.

7.2. Использование стандарта

Документация стандарта MPEG-4 (Second Edition) содержит 539 страниц формата А4. Стандарт H.264 изложен на 250 страницах того же формата. Стандарты сосредоточены на деталях, чтобы добиться наилучшей совместимости и согласованности систем и продуктов на их основе. Однако степень детализации в данных документах мешает извлечению из них необходимой информации. В этом параграфе будет дан обзор содержимого стандартов и рекомендации, как их интерпретировать и применять.

7.2.1. Что регламентирует стандарт?

Так же, как и более ранние стандарты видеокодирования, MPEG-4 и H.264 не предоставляют описания видеокодеков. Вместо этого стандарты подробным образом описывают *синтаксис* кодированного битового потока (двоичные коды и значения, из которых состоит поток), *семантику* элементов этого синтаксиса (что они означают) и *процесс*, на основании которого можно декодировать элементы синтаксиса и извлекать из них визуальную информацию.

Специфицировав синтаксис и семантику кодового потока, стандарт определяет, что в нем допустимо, а что – нет. Правильный кодер должен генерировать битовый поток, который может быть корректно декодирован с помощью предписанного стандартом процесса декодирования. Вместе с синтаксисом, семантикой и процессом декодирования в обоих стандартах описывается также *рекомендуемый декодер*, который вместе с некоторыми заданными уровнями операций накладывает определенные практические ограничения на конкретный кодер. Эти ограничения включают, например, максимальную битовую скорость кодирования, максимальный размер изображения и т. п. Это важно с точки зрения ограничений на объем памяти и вычислительные ресурсы, используемые декодером.

7.2.2. Понимание стандартов

Стандарты MPEG-4 и H.264 не похожи на подробные руководства. Они скорее формулируют те цели, которые необходимо достигнуть для совместимости между собой кодеков, созданных различными производителями. Например, «правильный» битовый поток H.264, сгенерированный кодеком, должен быть декодируемым

любым «правильным» декодером H.264. Имея в виду, что такую задачу выполнить непросто и стремясь преодолеть возможные препятствия, документы по стандартам пишутся их авторами длинно, сложно и трудночитаемо.

Изучив стандарт, можно разработать его декодер, поскольку процесс декодирования описан со всеми деталями. Процесс кодирования вовсе не специфицирован, что представляет разработчикам определенную гибкость для выбора их собственного метода кодирования. Однако синтаксис и процесс декодирования накладывают значительные ограничения на разработку кодера. Поэтому полезно иметь некоторые рабочие сведения о разработке хотя бы одного кодека, который полностью удовлетворяет требованиям данного стандарта.

Видеокодирование является существенным компонентом индустрии развлекательного мультимедиа, тем не менее стандарты представления цифрового видео могут оказаться весьма сложными для понимания разработчиком. Все это способствует появлению различных учебников, руководств, онлайн-ресурсов и книг, которые призваны помочь разобраться в указанных стандартах.

Полезный подход к пониманию и «декодированию» стандартов состоит в том, чтобы близко познакомиться с некоторой реализацией, полностью сделанной по правилам данного стандарта. Справочные программы доступны для обоих стандартов MPEG-4 и H.264. Имеется также исходный код программы кодера, который генерирует правильный битовый поток, и декодер, переводящий закодированный поток по правилам, предписанным процессом декодирования.

7.2.3. Подчинение стандартам

Для *кодера* подчинение стандарту видеокодирования означает, что он производит битовый поток, который удовлетворяет всем требованиям, предписанным синтаксисом, причем этот поток можно декодировать с помощью процесса декодирования стандарта в рамках ограничений на такие параметры, как битовая скорость кодирования. Для *декодера* подчинение стандарту предполагает, что он может декодировать «правильный» битовый поток, который удовлетворяет всем ограничениям и имеет правильную точность всех параметров. Понятие подчинения стандартам далее уточняется в *профилях* (согласованных подмножествах синтаксиса и инструментов в рамках стандарта) и *уровнях* (согласованных пределах производительности).

Хорошее руководство по подчинению стандарту должно обеспечивать взаимодействие со справочными программами

(программными реализациями MPEG-4 и H.264, доступными для их свободного скачивания разработчиками). Если кодер и декодер третьей стороны надежно взаимодействуют со справочными программами кодирования и декодирования, то это указывает на *возможную* подчиненность этого кодека стандарту. Однако этого еще недостаточно для признания такого подчинения. Справочные программы работают в режиме офлайн, кроме того, входы и выходы этих программ оформлены в виде файлов, что делает трудным их использование в режиме реального времени, которое может, например, потребоваться для «живой» трансляции видео по сетям. Кроме того, известны случаи несогласованности моделей справочных программ со спецификациями из документов стандартов (хотя группы экспертов прилагают все усилия к тому, чтобы минимизировать эти проблемы, исправляя ошибки, неточности и расхождения). Наконец, некоторые проблемы совместимости могут обнаружиться только при определенных условиях обработки конкретных типов видеоданных.

Обмен тестовыми материалами (выходами кодеров и декодеров) разных разработчиков между собой является правильным и доступным путем проверки и выявления широкого спектра потенциальных проблем совместимости и подчинения стандартам. Промышленные корпорации, такие, например, как Internet Streaming Media Alliance (ISMA) и MPEG-4 Industry Forum (MPEGIF), всемерно способствуют взаимодействию между командами тестеров различных компаний и организаций.

В MPEG-4 тестирование на совместимость и подчиненность битовых потоков и декодеров описано в части 4 (Тестирование совместимости). Считается, что некоторый битовый поток подчиняется стандарту, если он содержит «правильные» синтаксические элементы данного профиля и не выходит за пределы, предписанные используемым уровнем. Декодер подчиняется стандарту, если он способен декодировать любой битовый поток, подчиняющийся ограничениям выбранной комбинации профиль / уровень.

7.3. Обзор стандарта MPEG-4 (Part 2)

MPEG-4 Part 2 представляет собой объемный документ, описывающий широкий класс функций, относящихся к кодированию и представлению визуальной информации. Стандарт описывает следующие типы данных:

- движущиеся изображения (прямоугольные кадры);

- видеообъекты (области произвольной формы с движущимися предметами);
- двухмерные и трехмерные сеточные объекты (представляющие деформируемые объекты);
- анимированные лица и фигуры людей;
- статические текстуры (неподвижные изображения).

Разработаны различные инструменты кодирования для представления этих типов данных в сжатом виде. Имея разнообразные семейства инструментов, стандарт MPEG-4 может поддерживать различные приложения, например:

- «наследие» основных видеоприложений, таких как цифровое телевидение, видеоконференции и хранение видео;
- видеоприложения на основе объектов, в которых видеосцена может быть скомпонована различными видеообъектами, закодированными независимо друг от друга;
- представление компьютерной графики с использованием двухмерной и трехмерной деформируемой сеточной геометрии и анимированных лиц и фигур людей;
- «гибридные» видеоприложения, в которых комбинируются естественные видеосцены, неподвижные изображения и графика, сгенерированная компьютером;
- потоковое видео, передаваемое через Интернет и каналы мобильной связи;
- высококачественная видеопродукция, изготовленная и распространяемая для студийного употребления.

Несмотря на многообразие инструментов, задаваемых стандартом, алгоритмическим ядром MPEG-4 Part 2 является относительно простой механизм видеокодирования, использующий кодек на основе блоков с компенсацией движения и с последующим DST-преобразованием, квантованием и энтропийным кодированием. Синтаксис этого стержневого кодека (при некоторых ограничениях) идентичен ядру H.263. Большинство остальных функций и инструментов, поддерживаемых стандартом, получают добавлением некоторых деталей (за исключением инструментария кодирования сеток), неподвижных изображений и параметров анимации лиц и фигур, которые разрабатываются отдельно.

Рассмотрим содержание основных разделов стандарта.

Введение дает обзор целевых приложений, подходов и типов данных с особым акцентом на двухмерные и трехмерные сеточные объекты.

Разделы с 1-го по 5-ый являются преамбулой к описанию технических деталей стандарта. Раздел 1 описывает рамки и границы стандарта, раздел 2 содержит ссылки на другие документы стандарта, раздел 3 состоит из полезного списка терминологии и определений, раздел 4 перечисляет символы и сокращения, используемые в стандарте, а раздел 5 объясняет основные соглашения для описания синтаксиса стандарта.

Раздел 6 описывает синтаксис и семантику стандарта MPEG-4. Здесь определены различные структуры элементов битового потока вместе с предполагаемыми форматами несжатого видео. Описан синтаксис «правильного» битового потока, определено, какие синтаксические элементы являются частью стандарта и в каких последовательностях они могут встречаться. Там, где это необходимо, приводится семантика (смысл и допустимые значения) всех синтаксических элементов. Раздел 6 определяет допустимые параметры битового потока MPEG-4, и «правильный» кодер должен производить битовые потоки, отвечающие этим правилам.

Раздел 7 посвящен описанию процесса декодирования битовых потоков MPEG-4. Этот раздел определяет последовательность шагов, требуемых для декодирования правильного потока и для его преобразования в визуальные сцены или в визуальные объекты. Прежде всего описываются основные шаги декодирования остаточных данных (текстур), параметров форм и компенсации движения. Затем рассматриваются специальные инструменты, необходимые для кодирования других объектов (чересстрочное видеокодирование, кодирование неподвижных изображений и масштабируемых объектов и т. п.). Далее в разделе 7 определяется, как следует декодировать битовый поток MPEG-4. «Правильный» декодер должен выполнять все эти или эквивалентные им процедуры.

В *разделе 8* обсуждаются методы композиции объектов, закодированных независимо друг от друга, при создании синтетических видеосцен.

В *разделе 9* дается определение важных понятий стандарта – *профиль* и *уровень*. Это специфические подмножества инструментов кодирования (профили) и рабочие характеристики (уровни), которые кодек может выбирать и использовать. Эти точки соответствия особенно важны для разработчиков и производителей, поскольку представляется маловероятным, что конкретному приложению потребуется все разнообразие инструментов, обеспечиваемых стандартом.

Оставшаяся часть документации (около 200 страниц) состоит из 15 *приложений*. Некоторые из них носят *нормативный* характер, т. е.

являются важными частями стандарта. Прежде всего это спецификации дискретного косинусного преобразования и дискретного вейвлет-преобразования, используемых стандартом, таблицы кодов переменной длины, применяемых в битовых потоках MPEG-4, а также буферная проверка видео – VBV (Video Buffering Verifier), которая накладывает ограничения на рабочие параметры кодека. Остальные приложения являются *информационными*. В них разъясняются некоторые детали стандарта, но они не содержат новых определений. В виде примеров выделены описания методов устойчивости против ошибок, предложения по предварительной и последующей видеообработке, указания по организации алгоритмов контроля битовой скорости, а также список компаний, обладающих полезными патентами, на основе которых можно реализовывать некоторые функции стандарта.

7.4. Обзор стандарта H.264 (MPEG-4 Part 10)

Стандарт H.264 имеет более узкие границы применения по сравнению с MPEG-4. Он был разработан для эффективного кодирования и помехоустойчивой передачи видео прямоугольного формата. Его первоначальной задачей было обеспечение функциональности, схожей с H.263+ и MPEG-4 (простой профиль), но с лучшим сжатием и с более надежным механизмом передачи видеоданных. Целевые приложения включают двусторонние видеокommunikации (видеоконференции и видеотелефонию), видео высокого качества и организацию видеопотоков через сети пакетной передачи данных. Поддержка устойчивой передачи данных по сетям встроена в стандарт, который разработан так, чтобы его можно было легко встраивать в разные компьютерные платформы.

Этот стандарт описан гораздо подробнее своих предшественников (включая MPEG-4) в целях минимизации возможных разночтений. Детальный уровень представления стандарта сильно удлиняет его документирование и делает некоторые разделы плохо читаемыми.

Во *введении* перечислены некоторые целевые приложения, объяснены концепции профилей и уровней, дано краткое описание кодовому представлению.

Разделы с 1-го по 5-ый являются преамбулой к основному тексту, в них даются ссылки на остальные части документа, список терминов и определений, перечислены принятые сокращения, описаны арифметические операции и математические функции.

Раздел 6 дает описание входных и выходных форматов. Стандарт поддерживает обработку прогрессивного и чересстрочного видеοформата 4:2:0. Здесь же приводится описание форматов сэмплирования и форматов кодированного битового потока, а также определяется порядок обработки видеοкадров. В этом разделе описываются методы нахождения «близких» кодируемых элементов, которые необходимы для процесса декодирования.

В *разделе 7* рассматриваются синтаксис и семантика стандарта. Структура и последовательность синтаксических элементов описаны аналогично MPEG-4. Определяется семантика (смысл и допустимые значения) всех синтаксических элементов.

Раздел 8 посвящен основным процессам декодирования слоев (базовых кодовых единиц H.264). Первый параграф имеет дело с процессами более высокого уровня, например с обнаружением границ изображений и с обработкой ссылочных изображений. За этим следуют intra- и inter-прогнозирование, декодирование коэффициентов преобразования, реконструкция и встроенное деблокирующее фильтрование.

Раздел 9 описывает синтаксический разбор кодированного потока (т. е. как следует извлекать из него синтаксические элементы). Здесь имеется два основных метода: первый использует коды переменной длины, а второй основан на контекстно-адаптивном двоичном арифметическом кодировании.

Приложение А определяет три профиля, поддерживаемые H.264 (базовый, основной и расширенный), и специфицирует точки производительности (уровни).

Приложение В определяет формат потока байтов (последовательностей закодированных единиц данных, подходящих для трансляции по сетям с коммутацией каналов).

Приложение С описывает гипотетический справочный декодер. По аналогии с MPEG-4 буферная проверка видео (VBV) накладывает ограничения на скорость битового потока и частоту кадров декодируемого видео.

Приложения D и E содержат дополнительную информацию по усовершенствованию (SEI, Supplemental Enhancement Information) и информацию по используемости видео (VUI, Video Usability Information). Эта «посторонняя» информация поможет декодеру улучшить качество изображения, но она не существенна для корректного декодирования видеοпоследовательности.

7.5. Сравнение MPEG-4 и H.264

В табл. 7.2 приведены основные различия двух стандартов. Она не является полной, но в ней выделены важные особенности подходов MPEG-4 и H.264 к кодированию видеоданных.

Таблица 7.2

Основные различия MPEG-4 и H.264

| Сравнение | MPEG-4 | H.264 |
|---|--|---|
| Поддерживаемые типы данных | Прямоугольные кадры и полукадры видео, видеообъекты произвольной формы, неподвижные текстуры и «спрайты», гибридные синтетические и натуральные видеообъекты, сеточные двухмерные и трехмерные объекты | Прямоугольные кадры и полукадры видео |
| Число профилей | 19 | 3 |
| Степень сжатия | Средняя | Высокая |
| Поддержка потокового видео | Масштабируемое кодирование | Переключение слоев |
| Минимальный размер блока компенсации движения | 8×8 | 4×4 |
| Точность векторов движения | Половина и четверть пиксела | Четверть пиксела |
| Сравнение | MPEG-4 | H.264 |
| Преобразование | Дискретное косинусное блоков 8×8 | Дискретное косинусное блоков 4×4 |
| Встроенный деблокинговый фильтр | Нет | Да |

MPEG-4 делает акцент на вариабельность, а H.264 нацелен на эффективность и надежность. MPEG-4 обеспечивает высокую приспособляемость своих технических инструментов и ресурсов кодирования, давая возможность работать с широким спектром видеоданных, включающим прямоугольные видеокадры («традиционный» видеоматериал), видеообъекты (области визуальных сцен произвольной формы), состоящие как из обычных изображений, так и из гибридов «природных» объектов и объектов, сгенерированных компьютером. MPEG-4 обеспечивает свое функционирование посредством набора инструментов кодирования, организованных в «профили», которые состоят из рекомендуемых групп модулей для конкретных приложений. Классы профилей включают «простые» профили (кодирующие прямоугольные видеокадры), профили на основе объектов (кодирующие визуальные объекты прямоугольной формы), текстурные профили (кодирующие неподвижные изображения или текстуры), масштабируемые профили (кодирующие многократные разрешения или качественные уровни) и студийные профили (кодирующие высококачественные студийные приложения).

По сравнению с гибким подходом MPEG-4 стандарт H.264 сконцентрирован на эффективности сжатия видеокадров. Ключевыми признаками стандарта являются следующие элементы: эффективность сжатия (обеспечивающая значительное улучшение компрессии по сравнению со всеми предыдущими стандартами), эффективность передачи данных (со множеством встроенных деталей, поддерживающих надежную и устойчивую передачу по различным каналам и сетям) и сфокусированность на популярных приложениях видеосжатия. В настоящее время этим стандартом поддерживается только три профиля (по сравнению с 19 в MPEG-4), каждый из которых приспособлен к своему классу популярных приложений видеокommunikаций. Базовый профиль может быть использован в «разговорных» приложениях типа видеоконференции, расширенный профиль добавляет некоторые инструменты, которые могут быть полезны в потоковом видео в сетях, а основной профиль включает инструменты для широковещательных приложений и для хранения видеоданных.

7.6. Родственные стандарты

7.6.1. MPEG-1 и MPEG-2

Стандарт MPEG-1 был разработан для приложений просмотра и хранения видео на компакт-дисках CD. На такой диск помещается около 70 минут аудио-видео при скорости записи 1,4 Мбит/с. Стандарт MPEG-1 был задуман для поддержки Video CD – формата для любительского видео, который мог бы конкурировать с видео на аналоговых VHS-кассетах. MPEG-1 использует компенсацию движения на основе блоков, дискретное косинусное преобразование и квантование. Стандарт был оптимизирован для сжатия видео со скоростью около 1,2 Мбит/с. Стандарт MPEG-1 не получил коммерческого развития, потому что качество видеоизображения было не намного лучше, чем у аналоговых видеокассет, что не побудило потребителей перейти на новую цифровую технологию.

За MPEG-1 последовал стандарт MPEG-2, который был нацелен на потенциально более широкий рынок широковещательного цифрового телевидения. Он был основан на MPEG-1, но при этом имел несколько существенных изменений для поддержки целевых приложений, включая более эффективную компрессию чересстрочного (а также прогрессивного) видео, более гибкий синтаксис и более мощную системную часть стандарта. В стандарте MPEG-2 были впервые реализованы концепции профилей и уровней (задающие точки соответствия и пределы производительности) как способ поощрения взаимодействия разработчиков приложений без ущемления гибкости стандарта.

MPEG-2 имел (и имеет) большой коммерческий успех. Он стал применяться во всем мире для широковещательной цифровой телевизионной трансляции по кабельным сетям, спутниковым и наземным каналам. Он обеспечил кодирование фильмов в формате DVD-Video, который смог вытеснить видеопленку VHS (в чем потерпели неудачу MPEG-1 и Video-CD).

Во время написания приложений под MPEG-1 и MPEG-2 многие разработчики искали стандарт кодирования для видеопродуктов следующего поколения, в идеале обеспечивающий лучшее сжатие и большую устойчивость к передаче по сетям по сравнению со старыми стандартами. Оба новых стандарта – MPEG-4 и H.264 – являются претендентами для использования во многих ключевых мультимедийных приложениях настоящего и будущего поколений.

7.6.2. H.261 и H.263

H.261 был первым распространенным стандартом, разработанным ITU-T для поддержки видеотелефонии и видеоконференций по сетям ISDN с коммутацией каналов. Эти сети работают на скорости, кратной 64 кбит/с, и H.261 был разработан для осуществления достаточно простого видеокодирования на этой битовой скорости. Стандарт использовал известную гибридную модель DPCM/DCT с компенсацией движения с целопиксельной точностью.

В целях дальнейшего улучшения степени сжатия H.261 рабочая группа ITU-T разработала стандарт H.263. Это обеспечило лучшее сжатие, чем у H.261, с поддержкой базового видеокачества на скоростях до 30 кбит/с. H.263 вошел в группу стандартов, обеспечивающих передачу видео по сетям с коммутацией каналов и пакетов. Базовая модель кодирования H.263 (гибрид DPCM/DCT с полупиксельной компенсацией движения) была принята в качестве ядра простого профиля стандарта MPEG-4. Оригинальная модель H.263 включала четыре дополнительные моды кодирования (описанные в приложении к стандарту), а в выпуск 2 был добавлен ряд дальнейших деталей, например увеличенная степень сжатия и помехоустойчивая передача по зашумленным каналам. Термины H.263+ и H.263++ используются для обозначения кодеков, которые поддерживают некоторые дополнительные моды кодирования. Выбор возможных мод кодека был оставлен за разработчиками. Такой широкий спектр возможных комбинаций мод (а значит, и проблемы совместимости кодеков разных производителей) вынудил разработчиков стандарта H.264 остановиться на достаточно простой структуре из трех профилей.

8. КОДИРОВАНИЕ РЕЧЕВЫХ СИГНАЛОВ

Современные достижения в области создания высокопроизводительных процессоров цифровой обработки сигналов обеспечили возможность создания высококачественных цифровых систем связи. Ежегодно усилиями ученых и инженеров снижается информационная емкость передаваемых по каналам связи речевых сигналов, что приводит к постоянному удешевлению услуг связи. Кроме того, потребности в уменьшении скорости передачи (компрессии) речевых сигналов возникают в оборонной области и других силовых ведомствах для обеспечения закрытой связи.

При построении цифровых систем передачи сообщений принципиальным моментом является определение полосы частот, требуемой для обеспечения заданного качества воспроизведения переданного сообщения. Вообще говоря, для высококачественной передачи речевого сообщения требуется полоса не менее 10 кГц. Но для достижения удовлетворительного уровня разборчивости при передаче речи по телефонным каналам достаточно передать спектр в полосе 300...3400 Гц. Именно такой спектр звуковых частот обычно передается в современных системах передачи речевой информации, хотя в ряде случаев с целью повышения качества используется и более широкая полоса (например, в рекомендации G.722).

Известно, что для речевого сигнала характерна высокая степень избыточности, что обеспечивает высокую степень помехоустойчивости. В то же время переход на цифровые методы обработки речевых сигналов и передача данных по цифровым каналам связи позволяет существенно сократить объем передаваемой информации (осуществить компрессию), что в свою очередь позволяет снизить капитальные затраты на создание каналов связи, увеличить их пропускную способность. Кроме того, цифровая передача речи позволяет обеспечить высокую степень конфиденциальности передачи информации как за счет алгоритмов обработки речевого сигнала, так и за счет возможности дополнительной криптографической обработки цифровых данных.

8.1. Кодирование формы волны речевого сигнала

Принцип работы этого типа кодеков заключается в том, что они пытаются восстановить исходный сигнал сколь возможно точно без использования знаний о самом сигнале. Этот тип кодеков обладает невысокой сложностью и производит речь на скоростях кодирования

выше 16 кбит/с. Когда же скорость потока данных ниже этого уровня, то качество восстановленной речи сильно падает.

8.1.1. Рекомендация ITU-T G.711

Рекомендация, утвержденная в 1984 г., описывает кодек, использующий импульсно-кодovou модуляцию – PCM (Pulse Code Modulation) с частотой дискретизации 8000 кадров/секунду. Так как каждый кадр состоит из 8 бит, то кодек G.711 создает поток 64 кбит/с. Для снижения шумов квантования и улучшения преобразования сигналов с небольшой амплитудой при кодировании используется нелинейное квантование (командирование) по уровню согласно специальному псевдологарифмическому закону. При этом в Европе используется так называемый А-закон, а в Северной Америке и Японии – μ -закон.

Кодек G.711 широко распространён в системах традиционной телефонии с коммутацией каналов. Несмотря на то что рекомендация G.711 в стандарте H.323 является основной и первичной, в шлюзах IP-телефонии данный кодек применяется редко из-за высоких требований к полосе пропускания и задержкам в канале передачи. Использование G.711 в системах IP-телефонии обосновано лишь в тех случаях, когда требуется обеспечить максимальное качество кодирования речевой информации при небольшом числе одновременных разговоров.

8.1.2. Рекомендация ITU-T G.726

Она основана на адаптивной дифференциальной импульсно-кодовой модуляции – ADPCM (Adaptive Differential Pulse Code Modulation). Рекомендация G.726 была принята в 1990 г. Она заменила собой два других алгоритма – G.721, который описывает передачу голоса на основе ADPCM с битрейтом в 32 кбит/с, и G.723, который описывает аналогичный алгоритм с битрейтом в 24 и 40 кбит/с.

Алгоритм ADPCM основан на том, что в исходном PCM сигнале, передающем речь, существует значительная корреляция между соседними отсчетами. Поэтому если кодировать не саму амплитуду сигнала, а ее изменение по сравнению с предыдущим значением, то можно обойтись меньшим числом разрядов. Для речевых данных это позволяет уменьшить требуемое количество бит на отсчет примерно на 25 %. Кроме того, адаптивное изменение шага квантования позволяет еще больше уменьшить требования к полосе

пропускания при заданном отношении сигнал / шум. Скорость входного (PCM) битового потока 64 кбит/с. После ADPCM-кодирования скорости выходного потока могут быть 40, 32, 24 или 16 кбит/с, с размером кадра 5, 4, 3 или 2 бита соответственно. Наиболее часто используемый среди них – 32 кбит/с, он дает практически такое же качество воспроизведения речи, как и G.711.

Все методы кодирования, основанные на определенных предположениях о форме сигнала, плохо работают в ситуации, когда сигнал может передаваться с резкими скачками амплитуды. Именно такой вид имеет аудиосигнал, генерируемый модемами или факсимильными аппаратами. Современные системы обмена информацией, поддерживающие цифровые линии связи, умеют распознавать факсимильный обмен и передают соответствующие сигналы непосредственно в цифровом виде, не преобразуя их в аудиосигнал. Кодек G.726 в первую очередь используется на международных линиях связи в телефонных сетях. Он также является стандартным кодеком для беспроводных телефонных систем стандарта DECT.

8.1.3. Рекомендация ITU-T G.722

Исторически решающими факторами в выборе типичной полосы для передачи речевого трафика в диапазоне 0,3–3,4 кГц были экономические соображения и нехватка телефонных каналов. В цифровой телефонии существует принципиальная возможность использовать спектр речи за пределами полосы 0,3–3,4 кГц и тем самым повысить ее качество. Широкополосная передача обеспечивает передачу четкой и естественной речи.

ITU-T G.722 является стандартом кодирования широкополосной речи (ширина полосы – 7кГц). Широкополосное сжатие приводит к значительному улучшению качества телефонной речи. Технология кодека основана на ADPCM. Входной речевой сигнал предварительно разбивается в нем на две равные подполосы по 3,5 кГц. Далее кодек G.722 работает подобно G.726. Независимые кодеры ADPCM обрабатывают временные ряды сокращенных полос частот от двух фильтров и выдают скорости в 48 кбит/с и 16 кбит/с соответственно на выходе низкой и высокой полос. Кодер работает как на скоростях 56 и 48 кбит/с, так и с номинальной скоростью 64 кбит/с. Эта рекомендация была принята в 1988г. и в настоящее время считается устаревшей.

Более новая версия данного алгоритма известна как рекомендация G.722.1 (принята в 1999г.) Эта рекомендация

обеспечивает широкополосный речевой сигнал, более близкий по качеству к FM радиовещанию, чем к обычной телефонной связи. G.722.1 определяет работу кодека на скоростях 24 и 32 кбит/с при полосе пропускания 50Гц – 7кГц.

Рекомендация G.722.2 (принята в 2002г.) – современный и часто используемый вариант аналогичного кодека, также известный как Adaptive Multi Rate – WideBand (AMR-WB), предлагает возможность быстрого изменения скорости сжатия при изменении пропускной способности сети передачи данных. G.722.2 определяет 9 различных режимов скорости передачи от 6,6 кбит/с до 23,85 кбит/с.

8.2. Кодирование речи с низкой скоростью цифрового потока

Ниже будет рассмотрен ряд успешно реализованных вокодеров с низкими скоростями цифрового потока, которые были стандартизованы для применения в специальных коммуникационных приложениях (связь в оборонной сфере, спутниковая связь). Вокодер представляет собой устройство, осуществляющее параметрическое кодирование речевых сигналов. Компрессия речевых сигналов на передающем конце канала связи производится в анализаторе, выделяющем из речевого сигнала медленно меняющиеся составляющие, которые передаются по каналу связи в виде кодовых посылок. На приемном конце с помощью источников сигналов, управляемых принятыми параметрами, синтезируется речевой сигнал.

Для параметрического описания речи обычно используется подход, основанный на вычислении параметров, описывающих передаточную функцию речевого тракта человека и функцию возбуждения. Обычно для кодирования речи используются 8–10 параметров, вычисляемых на интервалах порядка 5–40 мс. Кроме того, вычисляется параметр, характеризующий изменение амплитуды или мощности сигнала, период основного тона речи, а также признак типа «тон / шум / пауза», характеризующий способ возбуждения речевого сигнала.

8.2.1. Стандарт MIL-STD 3005

В 1993 году на консорциуме в области цифровой обработки речи Министерства обороны США была принята трехлетняя программа по выбору нового речевого кодера со скоростью 2,4 кбит/с вместо устаревшего алгоритма LPC-10e. В ходе проведенного в последующие три года тестирования измерялись: доступность,

качество, узнаваемость говорящего, коммуникабельность и вычислительная сложность предлагаемых алгоритмов.

Данный проект привел к выбору в 1996 году MELP-вокодера (Mixed-Excitation Linear Predictive), работающего на скорости 2,4 кбит/с, в качестве нового стандарта MIL-STD 3005. MELP-вокодер использует кадр размером 22,5 мс и основан на традиционной параметрической модели кодирования с линейным предсказанием. В табл. 8.1 представлены параметры, передаваемые от кодера к декодеру в стандарте MIL-STD 3005.

Таблица 8.1

Параметры, передаваемые декодеру в стандарте
MIL-STD 3005

| Параметр | Озвученный кадр | Неозвученный кадр |
|--|-----------------|-------------------|
| Линейные спектральные частоты | 25 | 25 |
| Амплитуды преобразования Фурье | 8 | - |
| Коэффициенты усиления (2 за кадр) | 8 | 8 |
| Период основного тона, общая озвученность кадра | 7 | 7 |
| Озвученность по полосам | 4 | — |
| Флаг неперериодичности | 1 | — |
| Защита от ошибок | — | 13 |
| Синхробит | 1 | 1 |
| Всего за кадр 22.5мс | 54 | 54 |

Кроме того, данный алгоритм содержит пять дополнительных особенностей:

- смешанное возбуждение;
- использование аperiодических импульсов;
- адаптивное улучшение спектрального оценивания;
- размывание импульсов возбуждения;
- моделирование амплитуд коэффициентов Фурье.

Реализация смешанного возбуждения позволяет симулировать частотно-зависимую интенсивность звука, используя адаптивную фильтрацию. Главной целью смешанного возбуждения является

устранение призвуков, которые имеют место в LPC-вокодерах, особенно при широкополосном акустическом шуме.

MELP-кодер осуществляет синтез, используя периодические и аperiodические импульсы. Аperiodические импульсы применяются наиболее часто в переходных областях, между озвученными и неозвученными сегментами речевого сигнала.

Адаптивный фильтр основан на использовании полюсов фильтра линейного предсказания. Он расширяет формантную структуру синтетической речи и улучшает соответствие между синтетической и натуральной формами волны. Это также придает синтетической речи более натуральное звучание.

Размывание импульсов осуществляется с помощью фильтра с постоянными параметрами. Этот фильтр распределяет энергию возбуждения в пределах периода основного тона, снижая этим шероховатости в синтезированной речи.

Первые десять амплитуд Фурье определяются из пиковых значений преобразования Фурье сигнала-остатка. Информация, содержащаяся в этих коэффициентах, улучшает точность производимой речевой модели на наиболее важных низких частотах. Они увеличивают качество синтетической речи, особенно мужской, на фоне шумов.

В показателях доступности, качества, распознаваемости речи и коммуникабельности производительность нового MELP-алгоритма была значительно выше, чем у более раннего LPC-10e кодера с аналогичной скоростью в 2,4 кбит/с. Новый кодер по эффективности сравним с алгоритмом FS1016 (скорость 4,8 кбит/с) при более низкой скорости цифрового потока.

8.2.2. Стандарт NATO STANAG 4591

Деятельность Министерства обороны США подтолкнула организацию североатлантического договора (NATO) произвести схожий процесс обновления и стандартизации для битрейтов в 2,4 и 1,2 кбит/с, что и было произведено в 1997 году. Инженеры трех государств, входящих в североатлантический договор, оценивали предложенные алгоритмы с помощью тестов на качество, понятность, узнаваемость речи и языковую зависимость.

Модернизированный MELP-кодер получил название MELPe (enhanced MELP) и выбран как стандарт STANAG 4591 для скоростей в 2,4 и 1,2 кбит/с в 2001 году. Новый MELPe-кодер был основан на алгоритме MIL-STD 3005, используя схожие технологии квантования и расположения бит и обеспечивал полную совместимость с

существующим стандартом. Кроме того, производительность была улучшена за счет введения алгоритма шумоподавления, основанного на логарифмическом спектральном расстоянии. Предложенный MELPe-кодер был дополнен возможностью работы на скорости 1,2 кбит/с, с применением суперкадра из трех последовательных кадров по 22.5 мс. Такой подход позволил низкоскоростной модификации показать результаты, схожие со стандартной версией (со скоростью в 2,4 кбит/с), ценой введения дополнительной задержки в кодировании.

В 2006 году NATO STANAG 4591 был расширен для обеспечения работы на скорости 600 бит/с. Так же, как и версия для 1,2 кбит/с, этот кодер основан на MELPe-кодере с более длинным суперкадром для анализа. В такой ситуации четыре последовательных кадра группируются в суперкадр длиной 90 мс.

В целом семейство NATO STANAG 4591 MELPe-кодеров предоставляет возможность переговоров с коммуникационным качеством при значениях битрейта в 2,4 кбит/с, 1,2 кбит/с и 600 бит/с. Производительность MELPe (со скоростью в 2,4 кбит/с) лучше, чем у FS1016 (4,8 кбит/с). На скорости в 600 бит/с производительность MELPe лучше, чем у LPC-10e (2,4 кбит/с).

8.2.3. Работы ITU-T в области стандартизации кодера с битрейтом 4 кбит/с

Закончим этот пункт обсуждением достижений сектора стандартизации ITU-T в разработке кодера, обеспечивающего голосовую связь на скорости 4 кбит/с для широкого диапазона телекоммуникационных приложений. Результаты данного проекта были представлены общественности в 1994 году.

К сожалению, достижение заявленных целей оказалось более сложным, чем предполагалось. В параметрических низкоуровневых кодерах с битрейтом 4 кбит/с совершенствование кодеков было сфокусировано на лучшем представлении параметров конкретной модели, т. е. более быстром обновлении кадра, более точном моделировании спектральных амплитуд, использовании комплексных коэффициентов Фурье (или информации о фазе волны).

Другие подходы к проблеме заключались в уходе от параметрической модели в целом и работе с более скоростными кодерами формы сигнала, основанными на CELP (Code-Excited Linear Prediction), для снижения скорости передачи.

Третьей альтернативой является гибридный кодер, использующий как параметрическое кодирование, так и кодирование формы волны в разное время. С того момента как параметрические

кодеры стали хорошо обрабатывать периодическую голосовую речь, а кодеры формы волны стали хорошо моделировать переходные кадры, гибридные кодеры используют каждый из них в той ситуации, когда тот предоставляет наилучшую производительность. Гибридный MELP/CELP кодер лежал в основе претендента с наивысшей производительностью.

Стандарт ITU-T с битрейтом 4 кбит/с так и не увидел свет, т. к. ни один из рассмотренных кандидатов не смог достичь высокого качества во всех тестирующих лабораториях и на всех языках.

8.3. Гибридное кодирование

Появление гибридных кодеков обусловлено необходимостью закрыть «пустоту», образовавшуюся между современными вокодерами, работающими на скоростях до 2,4 кбит/с, и кодерами формы волны, функционирующими от скорости 16 кбит/с и выше. Наиболее успешными и распространенными стали кодеки, работающие во временной области по принципу анализа-через-синтез AbS (Analysis-by-Synthesis). В таких алгоритмах используется сигнал, форма которого подобрана таким образом, чтобы наиболее точно отобразить форму исходного сигнала (рис. 8.1). Метод кодирования AbS был впервые представлен в 1982 г. под аббревиатурой MPE (Multi-Pulse Excited), т. е. кодеки, возбуждаемые множественными импульсами. Позднее появились кодеки, возбуждаемые регулярными импульсами RPE (Regular-Pulse Excited), а также возбуждаемые кодовой последовательностью с линейным предсказанием CELP (Code-Excited Linear Predictive). Три этих основных кодека гибридного типа кратко рассматриваются далее.



Рис. 8.1. Упрощенная схема кодера, использующего схему анализ-через-синтез

Кодеки AbS работают путем разделения входного речевого сигнала на кадры, обычно длительностью 20 мс. Для каждого кадра определяются параметры для фильтра синтеза, а затем параметры возмущения фильтра. Это делается путем нахождения сигнала возмущения, который проходит через фильтр синтеза и сводит к минимуму ошибку (разницу) между входным и восстановленными сигналами. Отсюда и название «анализ-путем-синтеза», т. е. кодер анализирует входной сигнал путем подбора (синтеза) наиболее близкого к нему сигнала. Для каждого кадра кодер передает декодеру информацию, о параметрах фильтра синтеза и сигнале возбуждения. Тем самым на принимающей стороне заданный сигнал возбуждения проходит через фильтр синтеза с заданными параметрами, воссоздавая исходный сигнал.

Фильтр синтеза обычно представляет собой линейный фильтр с бесконечной импульсной характеристикой (БИХ) и передаточной функцией вида:

$$H(z) = \frac{1}{A(z)},$$

где $A(z) = 1 - \sum_{i=1}^p a_i z^{-i}$ – ошибка предсказания исходного сигнала фильтром синтеза; p – порядок фильтра синтеза (обычно около 10).

Фильтр синтеза также может представлять собой фильтр основного тона для моделирования долгосрочных периодичностей в вокализированной речи. Вместо этого данные о долгосрочных периодичностях речи могут быть смоделированы путем включения адаптивной кодовой книги в алгоритм возмущения таким образом, что сигнал возмущения $u(n)$ включал бы в себя компоненту формы $G u(n-\alpha)$, где α – период тона. МРЕ и RPE кодеры будут работать и без фильтра основного тона, хотя их производительность улучшается при введении такового. Однако для CELP кодеров использование фильтра тона является принципиальным.

Отличительным свойством AbS-кодеков является выбор функции возмущения $u(n)$ для фильтра синтеза. В кодеках с многоимпульсным возмущением $u(n)$ для каждого кадра имеет вид фиксированного количества ненулевых импульсов. Задача кодера состоит в том, чтобы определить положение и амплитуду этих импульсов внутри каждого кадра, а затем передать эту информацию декодеру. Теоретически возможно определить наилучшие значения этих параметров, но на самом деле это приведет к значительному усложнению алгоритма кодирования и, как следствие, к задержке. На

практике применяются субоптимальные методы определения указанных параметров, обычно используется около 4 импульсов на каждые 5 мс, что дает хорошее качество восстанавливаемой речи, при скорости кодирования порядка 10 кбит/с.

Так же как и в случае с многоимпульсным возмущением, в кодеках возбуждаемых регулярными импульсами используется набор ненулевых импульсов возмущения в качестве сигнала $u(n)$. Однако в данном случае импульсы расположены друг от друга на фиксированном расстоянии. Поэтому кодеру необходимо определить положение только первого импульса, а также амплитуду каждого импульса. Очевидно, что в этом случае меньшее количество информации требуется декодеру для восстановления исходного сигнала, а следовательно, возможно улучшение качества работы кодека без увеличения скорости кодирования. Так, на скорости около 10 кбит/с RPE- кодек может использовать 10 импульсов через каждые 5 мс, по сравнению с 4 импульсами для MPE-кодека. При этом, однако, сложность кодеков, возбуждаемых регулярными импульсами, выше сложности кодеков с многоимпульсным возмущением.

Несмотря на то что описанные выше два типа кодеков позволяют получать хорошее качество восстанавливаемой речи, они работают только на скоростях 10 кбит/с и выше. Невозможность их использования на более низких скоростях кодирования обусловлена тем, что информация об импульсах возмущения (амплитуде и их позициях) имеет достаточно большой объем. В случае же более грубого представления (квантования) амплитуд импульсов происходит быстрое ухудшение качества синтезируемой речи.

В настоящее время наиболее распространенным алгоритмом, дающим хорошее качество речи, на скоростях ниже 10 кбит/с являются кодеки, работающие на базе алгоритма возбуждения кодовой последовательностью с линейным предсказанием (CELP). Этот подход был предложен Шредером (Schroeder) и Аталом (Atal) в 1985 году и отличается тем, что в качестве возбуждающего сигнала используется алгоритм на основе векторного квантования. Сигнал возмущения (кодированное слово) выбирается из кодовой книги, представленный в ней индексом. Критерием выбора кодового слова является наименьшее евклидово расстояние между входным сигналом (тоже вектором) и этим кодовым словом. Размер кодовой книги обычно составляет 10 бит (1024 элемента книги), а разница (между исходным сигналом и выбранным кодовым словом) кодируется 5 битами. Из этого следует, что скорость, необходимая для передачи информации о сигнале возмущения, значительно снижается, так как необходимо передать лишь 15 бит информации на каждый отсчет, по

сравнению с 47 битами, используемыми, например в GSM RPE кодеке.

Изначально кодовая книга алгоритма CELP содержала белые Гауссовы последовательности, так как считалось, что возбуждающие сигналы именно такого вида могут обеспечить высокое качество воссоздаваемой речи. Однако главным остается тот факт, что для выбора кодового слова (а значит, и функции возмущения) необходимо было проверить каждую последовательность в фильтре синтеза на выполнение условия минимума евклидова расстояния до исходного сигнала. Это в свою очередь означает, что сложность кодека на базе CELP намного выше порога вычислительной сложности, позволяющего работать в реальном времени. Для примера необходимо сказать, что в год изобретения этого алгоритма кодирования (1985 г.) компьютеру Cray-1 CPU требовалось 125 секунд для обработки 1 секунды речевого сигнала. С тех пор была проделана большая работа по снижению вычислительной сложности CELP (в основном путем изменения структуры кодовой книги), а также значительно повысилось быстродействие современных цифровых сигнальных процессоров. Все это привело к тому, что алгоритм CELP стал доступен для работы в реальном времени даже на одиночных и недорогих процессорах.

На сегодняшний день разработан ряд модификаций алгоритма CELP известных под аббревиатурами ACELP (Algebraic CELP), RCELP (Relaxed CELP), LD-CELP (Low-Delay CELP) и VSELP (Vector CELP). Они служат основой для целого ряда международных стандартов в области сжатия речевых сигналов. AMR, EFR, AMR-WB (G.722.2), VMR-WB, EVRC, EVRC-B, SMV, TETRA, PCS 1900, MPEG-4 CELP и рекомендации ITU-T G.729, G.729.1 (первый этап кодирования), G.723.1 основаны на ACELP. Рекомендация ITU-T G.728 с битрейтом в 16 кбит/с на LD-CELP, а алгоритмы IS-54, IS-136 (D-AMPS), GSM (Half Rate speech) и iDEN на VSELP.

8.4. Методы оценки разборчивости речи

Речевой сигнал имеет двойственную структуру: с одной стороны, это обычный акустический сигнал, объективные акустические параметры которого вызывают определенные субъективные ощущения. Взаимодействия между ними, в соответствии с общими психофизическими законами, неоднозначны и нелинейны.

С другой стороны, речевой сигнал имеет особую структуру, в которой закодирована семантическая (смысловая) информация.

Поэтому процесс слухового восприятия речи представляет собой прежде всего процесс расшифровки и распознавания семантического и эмоционального содержания информации, содержащейся в речевом сигнале. Исследование этого процесса, то есть того, как мозг переводит акустические признаки речевого сигнала в его фонетическое и смысловое содержание, является в настоящее время одной из самых актуальных проблем в современной науке.

Современные достижения в цифровой обработке сигналов позволили достичь значительных успехов в этой области и получить практические результаты в компьютерном распознавании и синтезе речевых сигналов. Понимание процессов слухового восприятия речевого сигнала и расшифровки его смыслового содержания являются чрезвычайно важными, поскольку в процессе работы с речью необходимо понимание того, какие признаки в них являются наиболее критичными для передачи смыслового содержания. Однако, поскольку современные технологии расшифровки и синтеза речевых сигналов появились только в последние годы и еще достаточно сложны для применения, на протяжении уже нескольких десятилетий используются интегральные методы оценки правильной передачи смысловой информации, заключенной в речевом сигнале, – это методы оценки разборчивости. Оценка разборчивости важна для построения различных коммуникационных систем (радиовещательных, телефонных и др.). Поэтому остановимся на субъективных и объективных методах разборчивости речи.

В соответствии с международными стандартами, в частности ISO/TR 4870, под разборчивостью понимается «степень, с которой речь может быть понята (расшифрована) слушателями». Под этим понимается степень, с которой слушатели могут идентифицировать (понять смысл) фраз, слов, слогов и фонем. В соответствии с этим различаются виды разборчивости: фонемная, слоговая, словесная и фразовая, которые, однако, все связаны друг с другом и могут быть пересчитаны одна в другую.

При передаче речевого сигнала происходит неизбежная потеря информации. Хотя речевой сигнал обладает определенной избыточностью, однако различные шумы, искажения и реверберационные помехи могут привести к настолько значительной потере информации, что это сделает невозможным понимание смысла речи. Следует отметить, что «слышимость» и «разборчивость речи» – это разные понятия. Речь может звучать очень громко и быть прекрасно слышна, но быть при этом совершенно неразборчивой (например, в залах вокзалов, аэропортов и др.). Поэтому для оценки разборчивости речи разрабатываются специальные методы, отличные

от оценок ее громкости, и разработкой этих методов занимаются крупные международные организации: ISO, AES, IEC и др.

Все известные в настоящее время методы оценок разборчивости могут быть разделены на две большие группы: субъективные экспертные методы (ГОСТ 50840-95 и ГОСТ 51061-97, стандарт ANSI S3.2 и др.), и объективные методы, основные из которых: Alcons – процент артикуляционных потерь согласных (percentage Articulation Loss of Consonants); AI – индекс артикуляции (articulation Index); STI – индекс передачи речи (speech transmission index); RASTI – быстрый индекс передачи речи (rapid speech transmission index); SII – индекс разборчивости речи (speech intelligibility index) и др. (стандарты ISO/TR-4870, ANSI S3.2, S3.5; IEC 268-16) .

Таким образом, на разборчивость речи в различных условиях влияют следующие основные факторы: отношение сигнал / шум, время реверберации, уровень прямого звука, отношение энергии ранних и поздних отражений, частотный диапазон системы звукоусиления, формы АЧХ и ФЧХ, характеристики направленности, уровень нелинейных (особенно интермодуляционных) искажений, равномерность покрытия площади прослушивания.

Процент потери разборчивости зависит прежде всего от отношения уровня речевого сигнала к уровню шума (S/N), которое должно быть выше определенного уровня, чтобы можно было понять смысловое содержание речи. Степень маскировки шумом будет зависеть от отношения S/N и от спектрального состава шума. Для широкополосного шума (20–4000 Гц) зависимость процента словесной разборчивости от S/N показана на рис. 8.2. Из него видно, что процент словесной разборчивости будет больше 80% только при отношении $S/N > 12$ дБ.

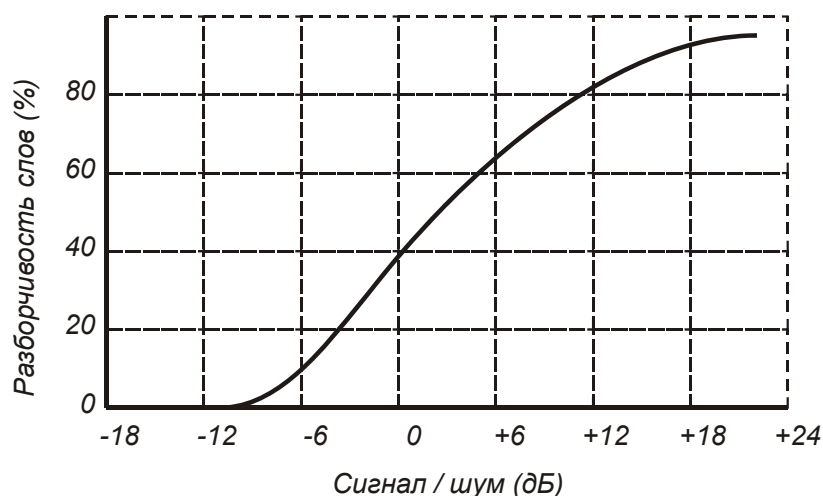


Рис. 8.2. Зависимость уровня разборчивости речи от отношения сигнал / шум при широкополосном шуме

Если шум узкополосный, то степень маскирования речи и потеря разборчивости зависят от частотной полосы (рис. 8.3), то более «опасными», чем высокочастотные (1800–2500 Гц) шумы, являются низкочастотные шумы (135–400 Гц).

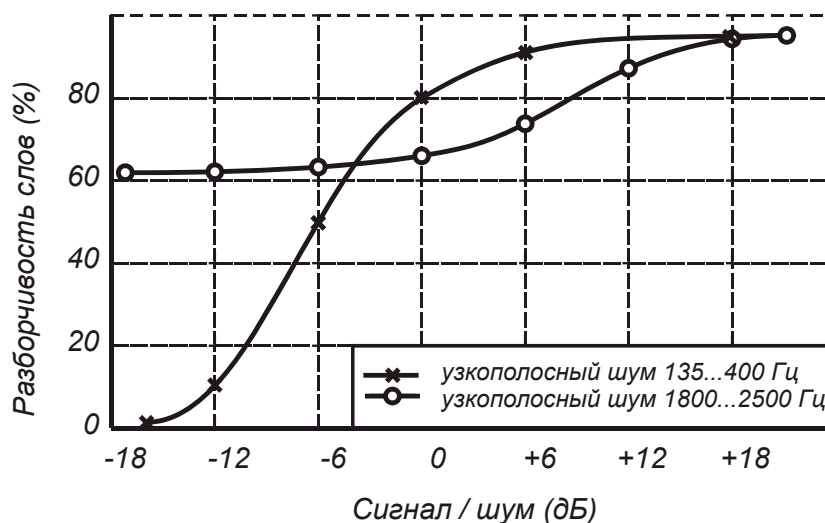


Рис. 8.3. Зависимость уровня разборчивости речи от отношения сигнал / шум при узкополосном шуме

Для количественной оценки разборчивости речи применяются как субъективные методы (экспертные оценки), так и объективные (расчет целого ряда параметров). Хотя за последние годы введено достаточно много новых объективных критериев и созданы специальные компьютерные программы для их реализации, оценки разборчивости речи с помощью квалифицированных экспертов по-прежнему остаются наиболее достоверными и все новые объективные критерии сравниваются с ними.

8.4.1. Субъективные методы оценки разборчивости

Оценка процента разборчивости (артикуляции) зависит от ряда факторов, основные из которых следующие:

- выбор для прослушивания элементов речи (звуки, слоги, слова, фразы), наиболее полно отражающих статистику данного языка;
- подбор состава экспертов и степень их тренированности;
- качество голоса диктора, его дикция, интонация и др.;
- требования к помещению и условиям в нем (уровню шумов и др.);
- методика проведения измерений и методы статистической обработки результатов.

Именно эти требования и задаются в различных стандартах, так как только при их точном соблюдении можно получить повторяемость результатов.

Для регламентации таких испытаний введены отечественные стандарты: ГОСТ 50840-95 «Передача речи по трактам связи. Методы оценки качества, разборчивости и узнаваемости», ГОСТ 51061-97 «Системы низкоскоростной передачи речи по цифровым каналам. Параметры качества речи и методы измерений», международные стандарты ISO/TR4870, IEC 268-16, американский стандарт ANSI S3.2-1989 «Method for measurement the Intelligibility of Speech Over Communication Systems» (имеется также новая редакция R-1999) .

Стандартизованные правила прежде всего касаются отбора испытательного материала: специально составленных таблиц фраз, слов или слогов, которые записываются или передаются диктором для оценки помещения, системы звукоусиления или других систем коммуникации. В зависимости от типа используемых при испытаниях элементов речи различается разборчивость звуковая, слоговая, словесная и фразовая. Все эти виды разборчивости при испытании одной и той же системы будут выражаться разными числовыми величинами, так как процент правильных оценок для предсказуемого сообщения всегда выше, чем для непредсказуемого. Степень предсказуемости при прослушивании фразы выше, чем при слушании отдельных слов или слогов, поскольку если часть фразы не услышана, то можно догадаться по смыслу о ее содержании. В связи с этим находятся и соотношения соответствующих видов разборчивости: фразовая – выше словесной, словесная – выше слоговой, слоговая – выше фонемной.

Для измерения различных типов разборчивости существуют свои характерные особенности. Например, измерение разборчивости слов целесообразно проводить в сетях, работающих в тяжелых условиях, например, при связи с подвижными объектами или в присутствии сильных шумов, а также для систем связи, процент разборчивость слогов у которых очень мал.

При проведении таких испытаний специально подобранные дикторы (с хорошей дикцией, правильной речью, с хорошим слухом) зачитывают в определенном ритме стандартизованные слоговые таблицы в выбранном помещении – с естественной акустикой или через звукоусилительную систему. Желательно, чтобы эксперты были незнакомы с дикторами, так как разборчивость у знакомых дикторов выше за счет запоминания экспертами их интонации, дикции и др. Количество дикторов должно быть не менее четырех, причем желательно, чтобы они имели минимальную разницу по акустическим

характеристикам голосов. Для проведения испытаний группа слушателей размещается в разных местах помещения и записывает прослушиваемый текст. Отношение правильно записанных на слух фонетических элементов к общему количеству переданных и определяет процент разборчивости.

Для получения статистически достоверных результатов необходимо привлечение достаточно большого числа слушателей. В стандартах ГОСТ принята численность группы слушателей в 20 человек, позволяющая получить статистически надежные результаты.

Большое влияние на результаты определения разборчивости речи оказывает не только количественный состав группы слушателей, но и другие факторы: образование, профессия, социальная принадлежность, а также память и сообразительность. Все слушатели должны обладать нормальным слухом, быть носителями данного языка и должны быть знакомы со всеми тестовыми словами. Возрастной состав ограничен 35 годами. В процессе испытаний могут привлекаться как тренированная бригада экспертов, показания которой проверены на эталонной системе, так и нетренированные слушатели (при этом их количество должно быть больше).

В отечественных стандартах по оценке качества передачи речи по каналам связи ГОСТ 50840-95 и ГОСТ 51061-97 используется измерение слоговой разборчивости речи и измерение фразовой разборчивости при нормальном и ускоренном темпах произнесения.

В соответствии с указанными стандартами по среднему проценту разборчивости все системы передачи речи делятся на 4 класса – эта классификация приведена в табл. 8.2.

Таблица 8.2

Классификация трактов связи в зависимости от типа и степени разборчивости

| Вид разборчивости | Качество речевого тракта, % (класс системы связи) | | | |
|-------------------|---|-------------------|------------------------------|------------------|
| | Отличное (класс 1) | Хорошее (класс 2) | Удовлетворительное (класс 3) | Слабое (класс 4) |
| Звуковая | >80 | 55–80 | 40–55 | 25–40 |
| Слоговая | >90 | 82–90 | 75–82 | 64–75 |
| Словесная | >98 | 93–98 | 87–93 | 75–87 |
| Фразовая | >99 | 97–99 | 95–97 | 90–95 |

Наряду с разборчивостью, часто указываются и другие субъективные факторы, влияющие на качество восприятия речи. К ним относятся: громкость речи, эхо, порхающее эхо, нарушение локализации, тембровые искажения, повышенный уровень шума и плохие акустические условия в зоне расположения источника звука. Следует заметить, что громкость, эхо и шум являются факторами, которые непосредственно определяют разборчивость речи и косвенно оцениваются при субъективной оценке разборчивости.

В международных стандартах, в частности ANSI S3.2-89, предлагается использовать пять дикторов и пять экспертов, удовлетворяющих указанным выше требованиям, но процедура предъявления речевого материала значительно сложнее.

Таким образом, процедура организации субъективных экспертиз по оценке разборчивости речи – дело сложное, длительное и достаточно дорогостоящее, хотя и наиболее достоверное. Поэтому за последние годы большое внимание было уделено созданию объективных методов оценки разборчивости, что позволило внедрить в практику целый ряд новых достаточно эффективных компьютерных методов расчета разборчивости речи в различных условиях.

8.4.2. Тесты MOS

Международным аналогом тестов в области субъективной оценки качества речи (а точнее, ее разборчивости) с привлечением группы экспертов является методика MOS (Mean Opinion Score), ход проведения которой установлен в рекомендациях ITU-T P.800 и P.830.

Основной методикой измерения разборчивости в MOS является тест ACR (Absolute Category Rating), который схож с артикуляционными тестами, за исключением трех основных моментов. Отличие состоит в том, что, во-первых, MOS хронологически появился намного позже упомянутых выше ГОСТов и, как следствие, при проведении тестов работает с электронными базами речевых эталонов, а их запись происходит независимо; во-вторых, в российских исследованиях разборчивость измерялась в процентах верно принятых элементов речи к общему их количеству, что контрастирует с тестами ACR, где оценка выставляется в баллах по шкале от 1 до 5. Данная шкала показана в табл. 8.3.

Последним и наиболее важным с точки зрения работы экспертов является то обстоятельство, что в MOS субъективным образом последовательно сравниваются эталонный и искаженный сетью сигналы (в отличие от описанных выше тестов, где мерой

разборчивости является количество распознанных верно элементов речи).

Таблица 8.3

Шкала качества MOS

| Оценка (Баллы) | Качество речи (концентрация внимания, требуемая для понимания образцов речи) |
|-------------------|--|
| 5 | Идеальное (концентрация внимания не требуется) |
| 4 | Хорошее (необходимо незначительное внимание) |
| 3 | Удовлетворительное (умеренная концентрация внимания) |
| 2 | Плохое (необходимо полное внимание) |
| 1 | Ужасное (смысл определить невозможно) |

Наряду с ACR, в MOS также используются методы DCR (Degradation Category Rating), CCR (Comparison Category Rating) и QRD (Quantal-Response Delectability). Опишем особенности каждого из них.

- *Degradation Category Rating (DCR)*. По сравнению с ACR метод DCR имеет большую чувствительность при оценке речевых сигналов с малыми искажениями (например, внешний шум), что позволяет применять данный метод для сравнения высококачественных сетей передачи речевых сообщений.
- *Comparison Category Rating (CCR)*. Ничем не отличается от DCR, за исключением того, что эталонный и искаженный сигналы появляются в случайном порядке следования (имеется в виду случайность внутри пары: эталон – искаженный сигнал). Кроме того, количество «инвертированных» и обычных пар одинаково, но заранее эксперт не знает, какая пара им оценивается. Как следствие перечисленных выше отличий изменяется шкала оценки (табл. 8.4).

Таблица 8.4

Шкала качества CCR

| Оценка (Баллы) | Качество образца речи по сравнению с идущим первым в паре |
|-------------------|---|
| 3 | Намного лучше |
| 2 | Лучше |
| 1 | Незначительно лучше |
| 0 | Приблизительно одинаково |
| -1 | Незначительно хуже |
| -2 | Хуже |
| -3 | Намного хуже |

- Quantal-Response Detectability (QRD). Является инструментом для получения информации о наличии в передаваемом сигнале эффектов искажения аналоговой природы, например, таких как эхо. В этом методе также присутствует особая шкала качества. Она показана в табл. 8.5.

Таблица 8.5

Шкала качества QRD

| Оценка | Восприятие эффектов искажения аналоговой природы |
|--------|--|
| А | Вызывают раздражение |
| В | Наблюдаются (обнаруживаются) |
| С | Не наблюдаются |

Таким образом, можно сказать, что процедура организации субъективных экспертиз по оценке качества (разборчивости) речи – дело сложное, длительное и достаточно дорогостоящее, т. к. требует привлечения большого числа людей, воссоздания сети связи, использования специальных лабораторий и соблюдения большого числа требований. Но главным недостатком субъективных тестов является тот факт, что их нельзя использовать для постоянного или хотя бы регулярного мониторинга сетей связи.

Однако при этом данные тесты являются наиболее достоверными среди всех существующих критериев оценки. Это подтверждается тем, что все современные объективные алгоритмы оценки качества передаваемой речи сравниваются именно с результатами MOS или подобными.

8.5. Объективные методы оценки качества речи

Классические объективные метрики, используемые для измерения качества речевого сигнала, в основном применимы и полезны в условиях линейных инвариантных к сдвигу систем, т. е. систем, для которых выполняются условия аддитивности и гомогенности, а также инвариантности к сдвигу во времени. С практической точки зрения это означает отсутствие в сети устройств, значительно преобразующих форму сигнала (то есть когда ухудшающие факторы носят только линейный или квазилинейный характер). Некоторые из методов оценки могут быть адаптированы для оценки систем с нелинейными искажениями путем анализа в кратких промежутках времени, однако применение такого рода метрик все равно остается ограниченным. Все описанное выше

говорит о том, что использование объективных метрик на практике не всегда дает исследователю понимание о восприятии абонентом переданного речевого сигнала. Иными словами, метрика может показать, что произошли сильные искажения, а разборчивость речи останется на прежнем уровне.

8.5.1. Отношение сигнал / шум

Для количественной оценки качества воспроизведения речевого сигнала обычно используется нормированный показатель погрешности, характеризующий средний квадрат ошибки воспроизведения, усредненный по времени и приведенной к дисперсии всего сообщения σ_c^2 :

$$\bar{\delta}^2 = \sigma_c^2 / \sigma_{ш}^2,$$

где $\sigma_{ш}^2$ – дисперсия шумов.

Отношение сигнал / шум (Signal-to-noise ratio, *SNR*) определяется как величина, обратная погрешности:

$$SNR = 10 \cdot \lg(\bar{\delta}^2)^{-2}, \text{ дБ}$$

При объективных методах оценки для анализа качества систем передачи речи необходимо оценивать отношение мощности сигнала к суммарной мощности шума – SNR_{Σ} .

Поскольку дисперсии сигнала σ_c^2 и шума $\sigma_{ш}^2$ вычисляются за полное время речевого теста, то данный показатель называется долгосрочным *SNR*.

Однако в случае адаптивного кодирования (например, при ADPCM) более целесообразно использование сегментного *SNR*, которое определяется следующим соотношением:

$$SNR_{сез} = \frac{1}{M} \sum_{i=1}^M 10 \cdot \lg(1 + \frac{\sigma_{ci}^2}{\sigma_{шi}^2}),$$

где σ_{ci}^2 и $\sigma_{шi}^2$ – дисперсии выборок речевого сообщения и ошибки, вычисленные на *i*-ом сегменте длительности *T*; *M* – общее количество сегментов в исследуемом сигнале.

Следует отметить, что исследования цифровых методов передачи речи и особенно различных адаптивных методов

кодирования выявили серьезное расхождение между субъективными экспертными оценками и SNR . Это объясняется различным характером искажений, создаваемых неадаптивными и адаптивными системами передачи. В неадаптивных системах имеет место стационарный шум с уровнем, независящим от уровня сигнала, качество сети передачи при этом определяется, главным образом, по восприятию шума в паузах речи. В адаптивных системах шумы незанятого канала могут быть незаметны, при этом большее значение имеют нестационарные шумы.

Следует заметить, что критерий SNR полезен только в случае, когда кодирование сигнала происходит без изменения формы волны, т. е. отсутствует такая обработка, как низкоскоростное кодирование или сжатие. В последних двух случаях SNR и сегментное SNR имеют низкую степень достоверности. Это является одной из причин, по которой для оценки разборчивости речи требуются качественно иные методы.

8.5.2. Логарифмическое спектральное расстояние

Логарифмическое спектральное расстояние (Log Spectral Distance, LSD) – это в достаточной мере универсальный критерий оценки качества речевых сигналов. LSD вычисляется в частотной области и, как следствие, не зависит от фазовых сдвигов во временной области между эталонным и оцениваемым сигналами. Формула логарифмического спектрального расстояния, измеряемого в дБ, будет выглядеть следующим образом:

$$LSD_{дБ} = \frac{1}{M} \sum_{i=0}^{M-1} \sqrt{\left\{ \frac{1}{(L/2)+1} \sum_{j=0}^{L/2} \left[10 \log_{10} |X(j,i)| - 10 \log_{10} |\hat{X}(j,i)| \right]^2 \right\}},$$

где M – количество сегментов речи, L – длина одного сегмента, $|X(j,i)|$ – спектральная мощность i -го сегмента эталонного сигнала, $|\hat{X}(j,i)|$ – спектральная мощность i -го сегмента искаженного сигнала.

Логарифмическое спектральное расстояние характеризует степень удаленности одного сигнала от другого. Таким образом, данная метрика качества речи является специализированным видом евклидова расстояния. LSD имеет обратную логику по сравнению с SNR . Чем выше отношение сигнал/шум, тем лучше качество выходного сигнала. В случае же с LSD чем меньше его показатель, тем ближе находятся спектры сигналов и, как следствие, тем лучше

качество выходного (в случае с кодированием – восстановленного) сигнала. Обе описанные метрики требуют использования эталонного (исходного) сигнала и поэтому являются эталонными.

Необходимо заметить, что существуют ситуации, когда объективные метрики не отражают правильным образом истинное качество речи, например, когда спектрально распределение шума более или менее повторяет спектр речевого сигнала (такое возможно, например, для шума квантования). В этом случае шум экранируется речью и замечен в гораздо меньшей степени, чем некоррелированный шум.

8.5.3. Современные методы объективной оценки качества речи

Новое поколение объективных критериев оценки качества речи своим появлением обязано тому, что субъективные и классические объективные методы оценки, как было указано выше, не соответствуют потребностям операторов современной связи. Для адекватной оценки уровня качества работы телефонной сети с пакетной коммутацией данных необходим постоянный ее мониторинг. Очевидно, что привлечение экспертов в этом случае не решает поставленной задачи и даже в случае его проведения потребует больших финансовых вливаний. Поэтому каждому провайдеру услуг связи необходим инструмент (метрика) для автоматической, т. е. компьютерной, оценки качества речи.

Разработка такого рода алгоритмов началась в середине 1990-х годов, и наиболее оптимальным решением поставленной задачи стали методы оценки качества речи с использованием эталонных образцов. На рис. 8.4 показана наиболее общая схема алгоритмов эталонной оценки качества речи.

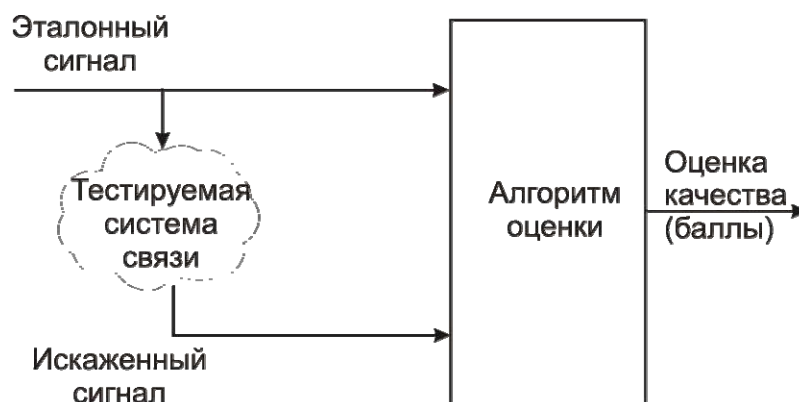


Рис. 8.4. Схема алгоритмов эталонной оценки качества речи

В соответствии с этим методом эталонный (входной) сигнал вводится в тестируемую систему связи, после чего, претерпев определенные искажения (характерные именно для этого типа системы), сигнал обрабатывается алгоритмом оценки, который после ряда математических преобразований выдает численную оценку качества искаженного (выходного) сигнала.

К эталонным сигналам, в свою очередь, также предъявляются определенные требования, которые во многом схожи с тестовыми речевыми образцами, используемыми в субъективных тестах. Это должны быть сигналы с частотой дискретизации не больше 8 или 16 кГц, квантованные с помощью 16-ти битовой РСМ. Кроме того, на длительность тестовых сигналов также накладываются ограничения, которые, однако, в различных рекомендациях и стандартах различны. Так, например, в рекомендации ITU-T P.800 длительность речевого эталона равна от 2,5 до 5 секунд, а в рекомендации ITU-T P.862 длительность фрагментов речи установлена в диапазоне от 8 до 16 секунд. Описанные ограничения помогают повысить уровень точности данного класса алгоритмов оценки.

В целом к эталонным алгоритмам оценки качества речи, разработанным за последние 15 лет, относятся следующие: Perceptual Speech Quality Measurement (PSQM), Perceptual Analysis Measurement System (PAMS), и Perceptual Evaluation of Speech Quality (PESQ). Алгоритм PESQ обладает наивысшей корреляцией (по Пирсону) с оценками экспертов, что определяет выбор именно этого критерия эталонной оценки качества речи для проведения тестовых исследований телекоммуникационного оборудования и мониторинга сетей связи на современном этапе развития. Коэффициент корреляции с оценками MOS составляет 0,935, по данным, приведенным в рекомендации ITU-T P.862, описывающей данный алгоритм.

В алгоритме PESQ большинство сетевых искажений учитывается с помощью временного и амплитудного выравнивания, а также посредством использования психоакустической модели человека, т. е. учета особенностей восприятия звуков человеком. В табл. 8.6 приведены потенциальные области применения PESQ, а также диапазон факторов, при которых данный алгоритм может работать неадекватно.

Области применения алгоритма PESQ

| Факторы воздействия на сигнал | | Применение | |
|---|--|--|--|
| для которых PESQ продемонстрировал высокую степень достоверности | | | |
| <ul style="list-style-type: none">– Ошибки канала передачи данных– Потери пакетов– Маскировка потерь пакетов CELP-кодеками– Различные скорости кодирования– Внешний шум– Джиттер | | <ul style="list-style-type: none">– Оценка кодеков– Выбор кодеков– Тестирование сетей (реальных, прототипных и эмулированных) в реальном времени | |
| для которых PESQ работает некорректно | | | |
| <ul style="list-style-type: none">– Потеря громкости– Задержка в тестах двустороннего разговора– Девиация уровня сигнала после выравнивания– Эхо | <ul style="list-style-type: none">– Технологии кодирования, при которых тишиной заменяется более 25% активной речи | <ul style="list-style-type: none">– В устройствах без возможности введения эталонного сигнала– Двусторонняя оценка | |

8.5.4. Структура алгоритма PESQ

Подробная блок-схема PESQ представлена на рис. 8.5–8.7. В частности, на рис. 8.5 изображен алгоритм выравнивания, рис. 8.6 показывает структуру психоакустической модели, а на рис. 8.7 изображена конечная часть PESQ – выставление оценки о качестве исследуемого сигнала и определение «плохих» интервалов.

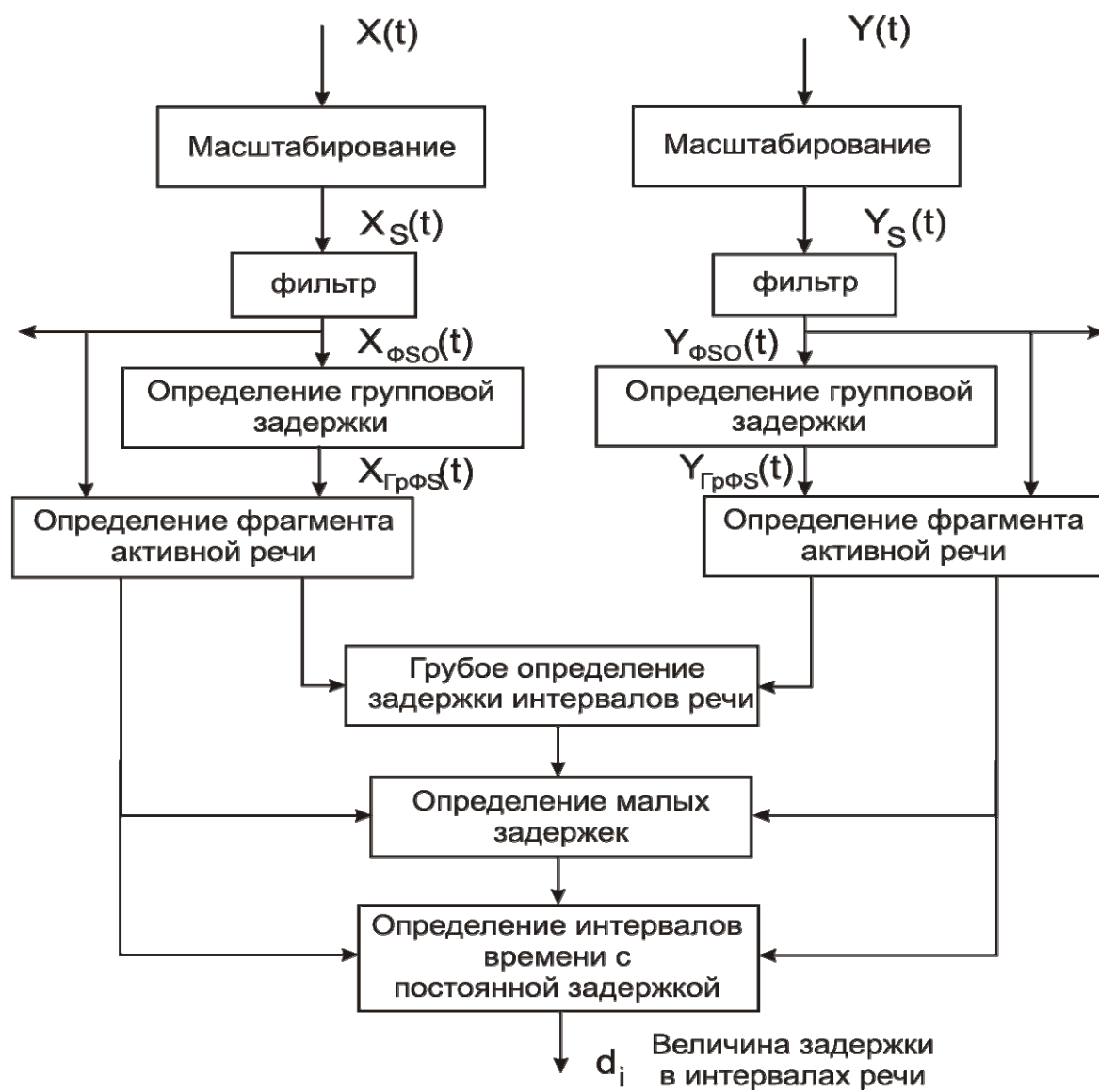


Рис. 8.5. Алгоритм выравнивания по уровню и по времени

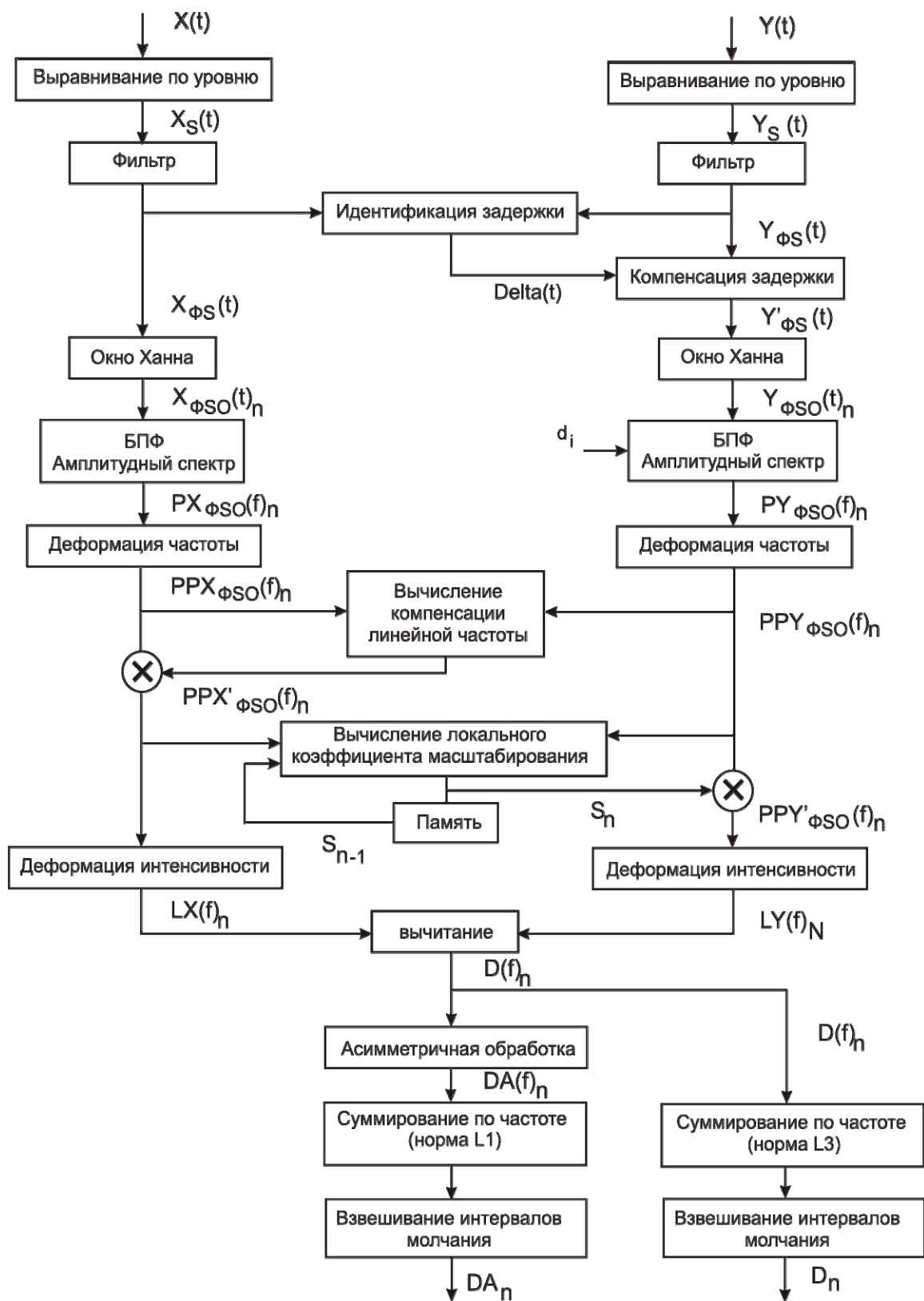


Рис. 8.6. Структура психоакустической модели

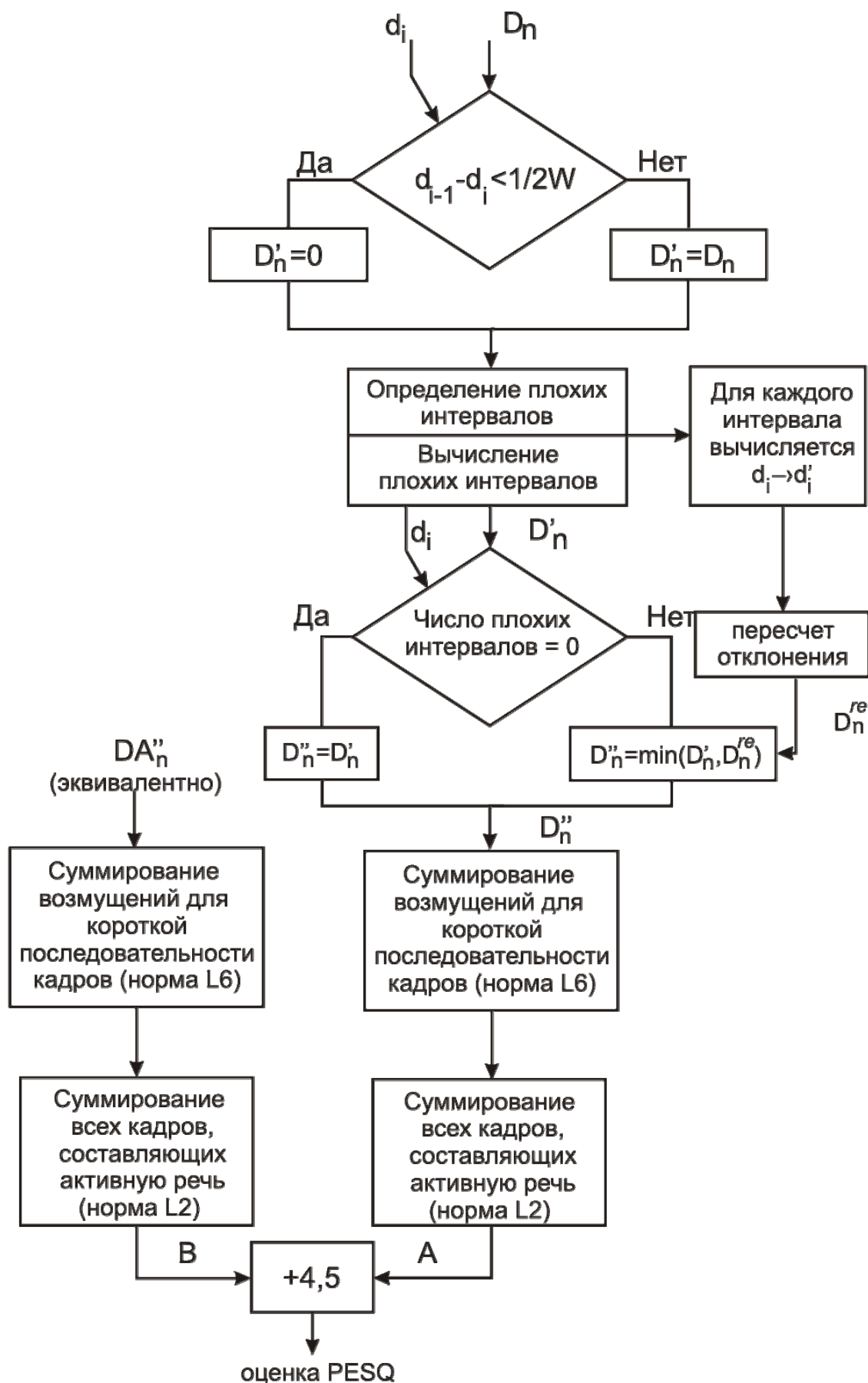


Рис. 8.7. Процедура выставления оценки качества и определение «плохих» интервалов

9. КОДИРОВАНИЕ АУДИОИНФОРМАЦИИ

При сравнении основных идей перцептуального (основанного на особенностях человеческого слуха) аудиокодирования и кодирования речи необходимо рассмотреть несколько аспектов. В то время как речевые кодеры сфокусированы на использовании модели источника сигнала, аудиокодеры могут не использовать такие модели, но при этом прибегать к использованию более общих средств для описания характеристик исходного сигнала. С другой стороны, перцептуальные аудиокодеры используют особенности восприятия звука человеком и путем обработки шумов кодирования и их маскирования их слышимость минимизируется.

Другое различие между аудио и речевыми кодерами лежит в ширине используемой ими полосы частот. В то время как музыкальные сигналы содержат заметные для восприятия компоненты на частотах до 16 кГц, достижением речевых кодеров для голосовой связи является его низкая требовательность. Важно то, что приложения голосовой коммуникации используют узкую полосу частот (300 Гц – 3,4 кГц). К сожалению, некоторая важная информация человеческой речи располагается вне этих границ. Для устранения таких эффектов и повышения удобства длительных разговоров может использоваться более широкая полоса частот (50 Гц – 7 кГц).

В этом разделе производится краткий обзор нескольких успешно стандартизованных перцептуальных аудиокодеров. Ключевую роль в данном вопросе играет экспертная группа MPEG и группа стандартизации ISO/IEC.

9.1. Кодеры MPEG-1 и MPEG-2

В 1992 г. группа MPEG завершила работу над своим первым международным стандартом кодирования видео и связанных с ним аудиодорожек. Этот стандарт известен как MPEG-1. Его звуковая составляющая (ISO/IEC 11172-3) определяет общий стандарт кодирования аудиосигналов. Она является основной технологией для многих важных приложений, основанных на сжатии аудио. Стандарт определяет три режима работы (3 слоя) с целью соответствия требованиям широкого диапазона приложений аудиокодирования.

Слой 2 определяет кодек со средней производительностью сжатия и вычислительной сложностью. Впервые он стал применяться в радиовещании, включая систему Европейской цифровой

аудиотрансляции (DAB). Кодек обеспечивал соответствующее радиотрансляции качество звука при скорости цифрового потока 128 кбит/с на канал.

В соответствии с этим слой 1 схемы кодирования является упрощенной версией слоя 2 кодера с низкой вычислительной сложностью и невысоким уровнем сжатия. Основным применением данной схемы являлась цифровая запись для потребителей (запись на цифровые компактные кассеты). Слой 1 обеспечивал трансляционное качество при скорости цифрового потока 192 кбит/с на канал.

Слой 3 схемы кодирования был разработан для обеспечения максимально возможного сжатия путем более точных вычислений, требующих большей вычислительной мощности. Передача звука по цифровым сетям с интегрированными услугами (ISDN) и спутниковое радиовещание стали первыми приложениями, где применялся данный кодек. Слой 3 стал очень популярным с развитием сети Интернет для хранения персональных аудиозаписей и получил отдельное название – MP3.

9.2. Формат кодирования MP3

MP3 (более точно, англ. MPEG-1/2/2.5 Layer 3 (но не MPEG-3) – третий формат кодирования звуковой дорожки MPEG) – лицензируемый формат файла для хранения аудиоинформации.

На данный момент MP3 является самым известным и популярным из распространенных форматов цифрового кодирования звуковой информации с потерями. Формат может проигрываться практически в любой популярной операционной системе, на практически любом портативном аудиоплеере, а также поддерживается всеми современными моделями музыкальных центров и DVD-плееров.

В формате MP3 используется алгоритм сжатия с потерями, разработанный для существенного уменьшения размера данных, необходимых для воспроизведения записи и обеспечения качества воспроизведения, очень близкого к оригинальному (по мнению большинства слушателей), хотя меломаны говорят об осязаемом различии. При создании MP3 со средним битрейтом 128 кбит/с в результате получается файл, размер которого примерно равен 1/10 от оригинального файла с аудио-CD. MP3-файлы могут создаваться с высоким или низким битрейтом, который влияет на качество файла-результата. Принцип сжатия заключается в снижении точности некоторых частей звукового потока, что практически неразличимо для слуха большинства людей. При этом на первом этапе строится

диаграмма звука в виде последовательности коротких промежутков времени, затем на ней удаляется информация, не различимая человеческим ухом, а оставшаяся информация сохраняется в компактном виде. Данный подход похож на метод сжатия, используемый при сжатии изображений в формат JPEG.

9.2.1. История создания MP3

MP3 разработан рабочей группой института Фраунгофера (нем. Fraunhofer-Institut für Integrierte Schaltungen) под руководством Карлхайнца Бранденбурга и университета Эрланген-Нюрнберг в сотрудничестве с AT&T Bell Labs и Thomson (Джонсон, Штолл, Деери и др.). Основой разработки MP3 послужил экспериментальный кодек ASPEC (Adaptive Spectral Perceptual Entropy Coding). Первым кодировщиком в формат MP3 стала программа L3Enc, выпущенная летом 1994 года. Спустя один год появился первый программный MP3-плеер – Winplay3.

При разработке алгоритма тесты проводились на вполне конкретных популярных композициях. Основной стала песня Сюзанны Веги «Tom's Diner». Отсюда возникла шутка, что «MP3 был создан исключительно ради комфортного прослушивания любимой песни Бранденбурга», а Сюзанну Вегу стали называть «мамой MP3».

9.2.2. Качество MP3

MP3 является форматом сжатия с потерями, т. е. часть звуковой информации, которую (согласно психоакустической модели) ухо человека воспринять не может или воспринимают не все люди, из записи удаляется безвозвратно. Степень сжатия можно варьировать, в том числе в пределах одного файла. Интервал возможных значений битрейта составляет 8–320 кбит/с. Для сравнения: поток данных с обычного компакт-диска формата Audio-CD равен приблизительно 689 кбит/с при частоте дискретизации 44100 Гц. В прошлом было распространено мнение, что запись с битрейтом 128 кбит/с подходит для музыкальных произведений, предназначенных для прослушивания большинством людей, обеспечивая качество звучания Audio-CD. На данный момент часто встречаются MP3-файлы с битрейтом 192 кбит/с, что может косвенно говорить, что большинство считает этот битрейт достаточным. Реально воспринимаемое «качество» зависит от исходного аудиофайла, слушателя и его аудиосистемы. Некоторые меломаны предпочитают сжимать музыку с «максимальным качеством» – 320 кбит/с, либо даже переходить на

кодеки без потерь, например FLAC. Также среди меломанов иногда бытует мнение, что некоторые сэмплы (фрагменты аудиозаписи) не поддаются качественному сжатию с потерями: на всех возможных битрейтах не составляет особого труда отличить сжатое аудио от оригинала. Однако есть и серьезные возражения против такой позиции.

9.2.3. Версии, режимы и управление кодированием MP3

Существует три версии MP3 формата для различных нужд: MPEG-1, MPEG-2 и MPEG-2.5. Отличаются они возможными диапазонами битрейта и частоты дискретизации:

- 32–320 кбит/с при частотах дискретизации 32000 Гц, 44100 Гц и 48000 Гц для MPEG-1 Layer 3;
- 16–160 кбит/с при частотах дискретизации 16000 Гц, 22050 Гц и 24000 Гц для MPEG-2 Layer 3;
- 8–160 кбит/с при частотах дискретизации 8000 Гц и 11025 Гц для MPEG-2.5 Layer 3.

Так как формат MP3 поддерживает двухканальное кодирование (стерео), существует 4 режима:

- *Стерео* – двухканальное кодирование, при котором каналы исходного стереосигнала кодируются независимо друг от друга, но распределение бит между каналами в общем битрейте может варьироваться в зависимости от сложности сигнала в каждом канале.
- *Моно* – одноканальное кодирование. Если закодировать двухканальный материал этим способом, различия между каналами будут полностью нивелированы, так как два канала смешиваются в один, он кодируется и он же воспроизводится в обоих каналах стереосистемы.
- *Двухканальный* – два независимых канала, например звуковое сопровождение на разных языках. Битрейт делится на два канала. Например, если заданный битрейт 192 кбит/с, то для каждого канала он будет равен только 96 кбит/с.
- *Объединенное стерео (Joint Stereo)* – оптимальный способ двухканального кодирования. Например, в одном из режимов левый и правый каналы преобразуются в их сумму (L+R) и разность (L-R). Для большинства звуковых файлов насыщенность

канала с разностью (L-R) получается намного меньше канала с суммой (L+R). Также тут свою роль играет восприятие звука человеком, для которого различия в направлении звука намного менее примечательны. Поэтому такой режим позволяет либо сэкономить на битрейте канала (L-R) или улучшить качество на том же битрейте, поскольку на канал суммы (L+R) отводится большая часть битрейта. Бытует мнение, что данный режим не подходит для звукового стереоматериала, в котором в двух каналах воспроизводится субъективно абсолютно различный материал, так как он стирает различия между каналами.

Относительно битрейта существуют следующие три модификации алгоритма MP3.

- *CBR (Constant Bit Rate)* – постоянный битрейт, который задается пользователем и не изменяется при кодировании произведения. Таким образом, каждой секунде произведения соответствует одинаковое количество закодированных бит данных (даже при кодировании тишины). Режим CBR может быть полезен для потоков мультимедиа данных по ограниченному каналу; в таком случае кодирование использует все возможности канала данных. Для хранения данный режим кодирования не является оптимальным, так как он не может выделить достаточно места для сложных отрезков исходного произведения, при этом бесполезно тратя место на простых отрезках. Повышенные битрейты (выше 256 кбит/с) могут решить данную проблему, выделив больше места для данных, но зато и пропорционально увеличивая размер файла.
- *VBR (Variable Bit Rate)* – переменный битрейт, который динамически изменяется программой-кодером при кодировании в зависимости от насыщенности кодируемого аудиоматериала и установленного пользователем качества кодирования (например, тишина закодируется с минимальным битрейтом). Этот метод MP3-кодирования является самым прогрессивным и до сих пор развивается и улучшается, так как аудиоматериал разной насыщенности может быть закодирован с определенным качеством, которое обычно выше, чем при установке среднего значения в методе CBR. Плюс к тому, размер файла уменьшается за счет фрагментов, не требующих высокого битрейта. Минусом данного метода кодирования является сложность предсказания размера выходного файла. Но этот недостаток VBR-кодирования незначителен в сравнении с его достоинствами. Также минусом является то, что VBR считает

«незначительной» звуковой информацией более тихие фрагменты, таким образом получается, что если слушать очень громко, то эти фрагменты будут некачественными, в то время как CBR делает с одинаковым битрейтом и тихие, и громкие фрагменты. Формат VBR постоянно улучшается, благодаря постоянному совершенствованию математической модели кодеков.

- *ABR (Average Bit Rate)* – усредненный битрейт, который является гибридом VBR и CBR: битрейт в кбит/с задается пользователем, а программа варьирует его, постоянно подгоняя под заданный битрейт. Таким образом, кодер будет с осторожностью использовать максимально и минимально возможные значения битрейта, так как рискует не вписаться в заданный пользователем битрейт. Это является явным минусом данного метода, так как сказывается на качестве выходного файла, которое будет немного лучше, чем при использовании CBR, но намного хуже, чем при использовании VBR. С другой стороны, этот метод позволяет наиболее гибко задавать битрейт (может быть любым числом между 8 и 320, против исключительно кратных 16 чисел метода CBR) и вычислять размер выходного файла.

9.3. Кодеры MPEG-2 AAC и MPEG-4 HE-AAC

После завершения работ над многоканальной версией MPEG-2 в 1994 году, группа стандартизации MPEG начала работы над другим многоканальным стандартом, который не обязан был обеспечивать совместимость с MPEG-1. Процесс стандартизации был завершен в 1997 году и известен как MPEG-2 Advanced Audio Coding (MPEG-2 AAC). Исследования доказали высокую производительность новой схемы кодирования для стерео- и монофонических записей. Кодек обеспечивал трансляционное качество при скорости цифрового потока 64 кбит/с на канал. Кроме того, кодек мог работать с широким диапазоном частот дискретизации (8 – 96 кГц) и скоростями потока от 16 кбит/с до 128 кбит/с на канал.

Технология MPEG-2 AAC была выбрана в качестве основы MPEG-4 аудиокодирования и была расширена рядом нововведений. В контексте кодирования речи несколько таких расширений представляют особый интерес:

- *MPEG-4 масштабируемое аудиокодирование*: одной из ключевых особенностей предоставляемых MPEG-4 является

масштабируемость, т. е. возможность декодирования лишь части полного цифрового потока. MPEG-4 масштабируемые цифровые потоки состоят из нескольких вложенных слоев информации, каждый из которых возникает от своего кодера. Интересным является тот факт, что MPEG-4 аудио определяет конфигурации, включающие и речевые и перцептуальные аудиокодеры, и может комбинировать их в одной системе.

- *MPEG-4 AAC с малой задержкой*: другим требованием к MPEG-4 аудиокодированию была возможность обеспечивать полнокачественное аудиокодирование с задержками кодирования / декодирования сравнимыми с речевыми кодерами, чтобы обеспечивать двустороннюю связь в реальном времени. Это было достигнуто с помощью ответвления AAC, называемого MPEG-4 AAC с малой задержкой.
- *Расширение полосы пропускания*: идея расширения полосы пропускания является одним из наиболее привлекательных дополнений к MPEG-4. Эта технология позволяет восстановить аппроксимацию исходного высокочастотного содержимого путем передачи низкочастотных компонент вместе с небольшим количеством дополнительной информации, отображающей высокочастотную спектральную область. Совмещение AAC кодера с расширением полосы пропускания и высокоэффективного AAC (HE-AAC) позволяет добиться привлекательного звучания звука при скорости цифрового потока всего лишь 24 кбит/с на канал.
- *Параметрическое стерео*: другим шагом на пути к повышению эффективности MPEG-4 AAC кодеров является совмещение с параметрическим стереокодированием (PS), который был разработан как часть MPEG-4 технологии высококачественного параметрического аудиокодирования. Данный инструмент позволяет эффективно воссоздавать стереосигнал путем передачи монофонической записи и небольшого количества сторонней информации, описывающей пространственную стереокартину. Основанный на данной технологии кодек расширяет передаваемый монофонический сигнал в стерео. Комбинированный с HE-AAC кодером кодек обеспечивает хорошее звучание при 32 кбит/с на стереоканал и ниже.

9.4. Сравнительный анализ качества кодирования

Учитывая то, что аудиокодирование сфокусировано на достижении наибольшего сжатия при сохранении субъективного

качества звучания, логичным становится вопрос о том, какие улучшения были произведены за прошлые десятилетия. На рис. 9.1 изображены скорости цифровых потоков, использовавшихся для кодирования стереосигнала в хорошем качестве.

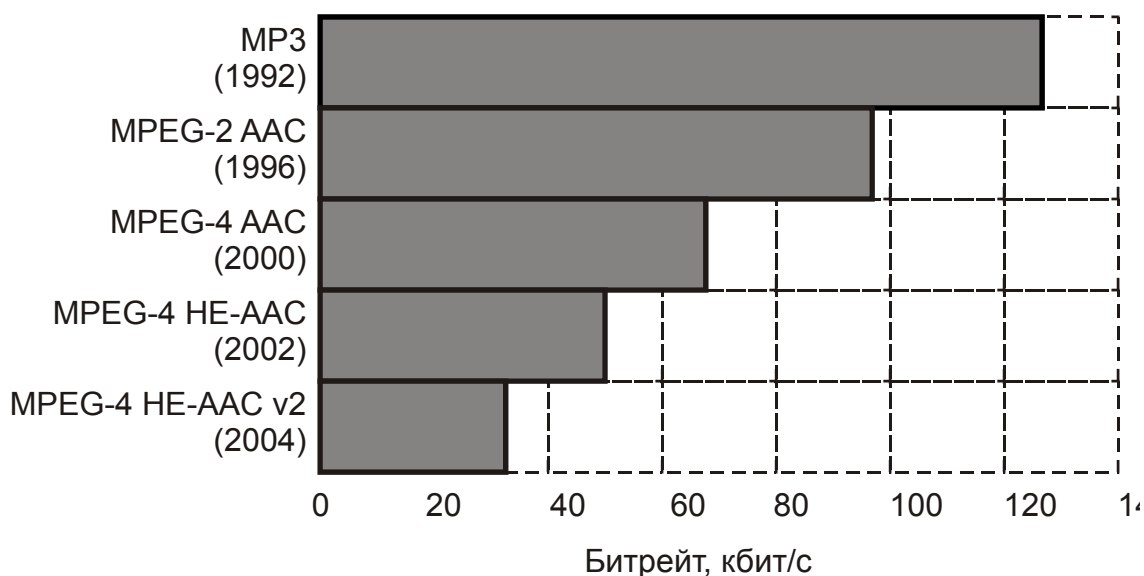


Рис. 9.1. Скорости цифрового потока для кодирования в хорошем качестве

Из рис. 9.1. следует снижение скорости цифрового потока с 128 кбит/с (MP3) до 32 кбит/с (HE-AAC v2 с параметрическим стереосигналом), которое характеризует прогресс в области кодирования аудиосигналов, достигнутый с момента выпуска формата MP3.

ЛИТЕРАТУРА

1. Ярославский Л. П. Введение в цифровую обработку изображений. М.: Советское радио, 1979.
2. Прэтт У. Цифровая обработка изображений. М.: Мир, 1982.
3. Цифровая обработка телевизионных и компьютерных изображений / под ред. Ю. Б. Зубарева, В. П. Дворковича. М.: МЦНТИ, 1997.
4. Борисов Ю. И., Дворкович В. П., Зубарев Ю. Б. Технические проблемы и перспективы внедрения цифрового телевидения и радиовещания в России. Ч. 1 // Broadcasting. Телевидение и радиовещание. 2007. № 1. С. 22–27.
5. Борисов Ю. И., Дворкович В. П., Зубарев Ю. Б. Технические проблемы и перспективы внедрения цифрового телевидения и радиовещания в России. Ч. 2 // Broadcasting. Телевидение и радиовещание. 2007. № 2. С. 14–18.
6. Зубарев Ю. Б., Дворкович В. П., Дворкович А. В. Проблемы и перспективы внедрения информационных мультимедийных систем в России // Электросвязь. 2004. № 10. С. 11–16.
7. Шлихт Г. Ю. Цифровая обработка цветных изображений. М.: Эком, 1997.
8. Абламейко С. В., Лагуновский Д. М. Обработка изображений: технология, методы, применение. – Минск: Амалфея, 2000.
9. Приоров А. Л., Апальков А. В., Хрящев В. В. Цифровая обработка изображений: учеб. Пособие. Ярославль, 2007.
10. Приоров А. Л., Ганин А. Н., Хрящев В. В. Цифровая обработка изображений: учеб. Пособие. Ярославль, 2001.
11. Красильников Н. Н. Цифровая обработка изображений. М.: Вузовская книга, 2001.
12. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: Диалог-МИФИ, 2002.
13. Цифровое преобразование изображений / Под ред. Р. Е. Быкова. М.: Горячая линия – Телеком, 2003.
14. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005.
15. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. М.: Техносфера, 2006.
16. ITU-R Recommendation BT.500-11. Methodology for the subjective assessment of the quality of television pictures. ITU-T, 2002.
17. Ричардсон Я. Видеокодирование. H.264 и MPEG-4 – стандарты нового поколения. М.: Техносфера, 2005.

18. Сэломон Д. Сжатие данных, изображений и звука. М.: Техносфера, 2004.
19. Вернер М. Основы кодирования. М.: Техносфера, 2004.
20. The USC-SIPI Image Database (<http://sipi.usc.edu/database>).
21. Test Video (<http://media.xiph.org/video/derf/>).
22. ISO/IEC 14496-2, Coding and audio-visual objects. Part 2: Visual. 2001.
23. ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced video coding. 2003.
24. Беллами Дж. К. Цифровая телефония; пер. с англ. М.: Экотрендз, 2004.
25. Быков С. Ф., Журавлев В. И., Шалимов И. А. Цифровая телефония: учеб. пособие для вузов. М.: Радио и связь, 2003.
26. Гольдштейн Б. С., Пинчук А. В., Суховицкий А. Л. IP-телефония. М.: Радио и связь, 2001.
27. Ковалгин Ю. А., Вологдин Э. И., Цифровое кодирование звуковых сигналов. СПб.: Корона-Принт, 2004.
28. Радзишевский А. Ю. Основы аналогового и цифрового звука. М.: Вильямс, 2006.
29. ITU Recommendation P.800 Methods for subjective determination of transmission quality.
30. ITU Recommendation P.830 Subjective performance assessment of telephone-band and wideband digital codecs.
31. ITU Recommendation P.56 Objective measurement of active speech level.
32. ITU Recommendation P.861 Objective quality measurement of telephone-band (300-3400 Hz) speech codecs.
33. ITU Recommendation P.862 Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.
34. ГОСТ Р 50840-95. Передача речи по трактам связи. Методы оценки качества, разборчивости и узнаваемости.
35. ГОСТ Р 51061-97. Системы низкоскоростной передачи речи по цифровым каналам. Параметры качества речи и методы измерений.
36. Алдошина И. Основы психоакустики. Ч. 17: Слух и речь; Ч. 4. Субъективные и объективные методы оценки разборчивости речи // Звукорежиссер. 2002. № 5.

ПРИЛОЖЕНИЕ 1

Стандарты кодирования мультимедиа и гипермедиа информации ISO (рабочая группа – JTS1/SC29)

| JPEG | | | |
|---|----------------------|-----------------|--|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Цифровое сжатие и кодирование неподвижных изображений | 1994 | ISO/IEC 10918-1 | Требования и руководящие указания |
| | 1995 | ISO/IEC 10918-2 | Сертификация нормативных тестов на процедуру кодирования / декодирования |
| | 1997 | ISO/IEC 10918-3 | Стандарт на расширения для JPEG |
| | 1999 | ISO/IEC 10918-4 | Регистрация профилей JPEG; профилей, цветовых пространств, типов сжатия и др. для формата SPIFF (Still Picture Interchange File Format). |

| JBIG | | | |
|---|----------------------|---------------|--|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Кодированное представление изображений и аудио информации | 1993 | ISO/IEC 11544 | Сжатие черно-белых (bi-level) изображений с прогрессивной разверткой |

| JPEG-LS | | | |
|--|----------------------|-----------------|------------------------|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Сжатие неподвижных изображений без потерь / практически без потерь | 1999 | ISO/IEC 14495-1 | Базовые принципы |
| | 2003 | ISO/IEC 14495-2 | Расширения |

| JBIG-2 | | | |
|--|----------------------|---------------|----------------------------|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Сжатие черно-белых изображений без потерь / практически без потерь | 2001 | ISO/IEC 14492 | Описание кодера и декодера |

| JPEG-2000 | | | |
|---|----------------------|------------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Система кодирования изображений JPEG-2000 | 2004 | ISO/IEC 15444-1 | Часть 1: Основная система кодирования (Core Coding System) – определяет базовый набор свойств и синтаксис потокового кода в JPEG-2000. |
| | 2004 | ISO/IEC 15444-2 | Часть 2: Расширения (Extensions) – добавляет новые свойства в основную систему кодирования |
| | 2002 | ISO/IEC 15444-3 | Часть 3: Motion JPEG2000 определяет файловый формат MJ2, содержащий последовательность изображений, закодированных в соответствии с основной системой кодирования и используется для сжатия видео |
| | 2004 | ISO/IEC 15444-4 | Часть 4: Нормативные тесты (определяют процедуру тестирования для кодирования и декодирования с использованием части 1 стандарта JPEG-2000) |
| | 2003 | ISO/IEC 15444-5 | Часть 5: Справочные программы для тестирования и проверки систем на основе JPEG-2000 |
| | 2003 | ISO/IEC 15444-6 | Часть 6: Комплексный формат файлов (Compound Image File Format) – определяет файловый формат JPM для компрессии сложных изображений |
| | 2005 | ISO/IEC 15444-9 | Часть 9: Интерактивные инструменты, APIs и протоколы |
| | 2005 | ISO/IEC 15444-12 | Часть 12: Основной формат медиа файлов |

| MPEG-1 | | | |
|--|----------------------|-----------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Кодирование движущихся изображений и сопутствующего звука для хранения цифрового медиа со скоростью записи до порядка 1,5 Мбит/с | 1993 | ISO/IEC 11172-1 | Part 1: Systems (базовые вопросы) |
| | 1993 | ISO/IEC 11172-2 | Part 2: Video (кодирование визуальных объектов) |
| | 1993 | ISO/IEC 11172-3 | Part 3: Audio (кодирование аудио объектов) |
| | 1995 | ISO/IEC 11172-4 | Part 4: Нормативные тесты |
| | 1998 | ISO/IEC 11172-5 | Part 5: Программная реализация |

| MHEG | | | |
|--|----------------------|-----------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Кодирование мультимедиа и гипермедиа информации | 1997 | ISO/IEC 13522-1 | Part 1: Представление объектов MHEG |
| | 1997 | ISO/IEC 13522-3 | Part 3: Представление обмена сценариями MHEG |
| | 1996 | ISO/IEC 13522-4 | Part 4: Процедура регистрации MHEG |
| | 1997 | ISO/IEC 13522-5 | Part 5: Поддержка интерактивных применений базового уровня |
| | 1998 | ISO/IEC 13522-6 | Part 6: Поддержка интерактивных применений повышенного уровня |
| | 2001 | ISO/IEC 13522-7 | Part 7: Взаимодействие и нормативные тесты для ISO/IEC 13522-5 |
| | 2001 | ISO/IEC 13522-8 | Part 8: XML для ISO/IEC 13522-5 |

| MPEG-2 | | | |
|--|----------------------|------------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Родовое кодирование движущихся изображений и сопутствующей аудио информации | 2000 | ISO/IEC 13818-1 | Part 1: Systems (базовые вопросы) – описывает синхронизацию и мультиплексирование видео и аудио |
| | 2000 | ISO/IEC 13818-2 | Part 2: Video (кодирование визуальных объектов) |
| | 1998 | ISO/IEC 13818-3 | Part 3: Audio (кодирование аудиообъектов) |
| | 2004 | ISO/IEC 13818-4 | Part 4: Нормативные тесты |
| | 2005 | ISO/IEC 13818-5 | Part 5: Программная реализация |
| | 1998 | ISO/IEC 13818-6 | Part 6: Расширения для DSM-CC (Digital Storage Media Command and Control) |
| | 2006 | ISO/IEC 13818-7 | Part 7: Расширенное аудиокодирование (AAC) |
| | 1996 | ISO/IEC 13818-9 | Part 9: Расширение для декодеров в приложениях реального времени |
| | 1999 | ISO/IEC 13818-10 | Part 10: Нормативные расширения для DSM-CC (хранение цифрового медиа – управление и контроль) |
| | 2004 | ISO/IEC 13818-11 | Part 11: IPMP (управление правами интеллектуальной собственности) в системах MPEG-2 |

| MPEG-4 | | | |
|--|----------------------|------------------|--|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Кодирование аудио-видео объектов | 2004 | ISO/IEC 14496-1 | Part 1: Systems (базовые вопросы: описание сцены, мультиплексирование аудио, видео и связанной информации, синхронизация, организация буферов, права интеллектуальной собственности) |
| | 2004 | ISO/IEC 14496-2 | Part 2: Visual (кодирование визуальных объектов) |
| | 2005 | ISO/IEC 14496-3 | Part 3: Audio (кодирование аудиообъектов) |
| | 2004 | ISO/IEC 14496-4 | Part 4: Нормативные тесты (тесты на соответствие, процедуры тестирования, проверка битового потока) |
| | 2001 | ISO/IEC 14496-5 | Part 5: Справочные программы |
| | 2000 | ISO/IEC 14496-6 | Part 6: Сеансовый протокол для мультимедийного потока (DMIF) |
| | 2004 | ISO/IEC 14496-7 | Part 7: Оптимизированные справочные программы для кодирования аудио-, видеообъектов |
| | 2004 | ISO/IEC 14496-8 | Part 8: Передача кодированных данных MPEG-4 по сетям протокола IP |
| | 2004 | ISO/IEC 14496-9 | Part 9: Описание инструментов кодирования VHDL в MPEG-4 |
| | 2005 | ISO/IEC 14496-10 | Part 10: Расширенное видеокодирование (AVC) |
| | 2005 | ISO/IEC 14496-11 | Part 11: Описание сцены и ядро приложения |
| | 2005 | ISO/IEC 14496-12 | Part 12: Основной файловый формат медиа |
| | 2004 | ISO/IEC 14496-13 | Part 13: Управление правами интеллектуальной собственности и расширение защиты |
| | 2003 | ISO/IEC 14496-14 | Part 14: Форматы файлов MPEG-4 |
| | 2004 | ISO/IEC 14496-15 | Part 15: Форматы файлов AVC |
| | 2006 | ISO/IEC 14496-16 | Part 16: Расширение рамок анимации (AFX) |
| | 2006 | ISO/IEC 14496-17 | Part 17: Форматы потокового текста |
| | 2004 | ISO/IEC 14496-18 | Part 18: Сжатие шрифтов и потоки |
| | 2004 | ISO/IEC 14496-19 | Part 19: Поток синтезированной структуры |
| | 2006 | ISO/IEC 14496-20 | Part 20: Представление сцен для простых применений (LSeR) и формат простой группировки (SAF) |
| | 2006 | ISO/IEC 14496-21 | Part 21: Расширение рамок графики (GFX) |
| | 2007 | ISO/IEC 14496-22 | Part 22: Формат открытых шрифтов |

| MPEG-7 | | | |
|---|----------------------|------------------|--|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Интерфейс описания мультимедийного контента | 2002 | ISO/IEC 15938-1 | Part 1: Systems (бинарный формат кодирования описаний) |
| | 2002 | ISO/IEC 15938-2 | Part 2: Язык описания определений (дескрипторов) |
| | 2002 | ISO/IEC 15938-3 | Part 3: Visual (дескрипторная схема, отвечающая исключительно за визуальные дескрипторы) |
| | 2002 | ISO/IEC 15938-4 | Part 4: Audio (дескрипторная схема, отвечающая исключительно за аудио-дескрипторы) |
| | 2003 | ISO/IEC 15938-5 | Part 5: Схемы описания мультимедиа |
| | 2003 | ISO/IEC 15938-6 | Part 6: Справочные программы |
| | 2003 | ISO/IEC 15938-7 | Part 7: Нормативные тесты |
| | 2002 | ISO/IEC 15938-8 | Part 8: Извлечение и использование описаний MPEG-7 |
| | 2005 | ISO/IEC 15938-9 | Part 9: Профили и уровни |
| | 2005 | ISO/IEC 15938-10 | Part 10: Определение схемы |
| | 2005 | ISO/IEC 15938-11 | Part 11: Схемы профилей MPEG-7 |

| MPEG-21 | | | |
|--------------------------|----------------------|------------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Структура мультимедиа | 2004 | ISO/IEC 21000-1 | Part 1: Концепция, технологии и стратегия |
| | 2005 | ISO/IEC 21000-2 | Part 2: Описание цифровых сообщений |
| | 2003 | ISO/IEC 21000-3 | Part 3: Распознавание цифровых сообщений |
| | 2006 | ISO/IEC 21000-4 | Part 4: Компоненты управления правами интеллектуальной собственности (IPMP) |
| | 2004 | ISO/IEC 21000-5 | Part 5: Язык выражения прав |
| | 2004 | ISO/IEC 21000-6 | Part 6: Словарь сведений о правах |
| | 2004 | ISO/IEC 21000-7 | Part 7: Адаптация цифровых сообщений |
| | 2006 | ISO/IEC 21000-8 | Part 8: Справочные программы |
| | 2005 | ISO/IEC 21000-9 | Part 9: Форматы файлов |
| | 2006 | ISO/IEC 21000-10 | Part 10: Обработка цифровых сообщений |
| | 2004 | ISO/IEC 21000-11 | Part 11: Оценочные инструменты для технологий устойчивой связи |
| | 2005 | ISO/IEC 21000-12 | Part 12: Тестовое основание для средств доставки MPEG-21 |
| | 2006 | ISO/IEC 21000-15 | Part 15: Event Reporting |
| | 2005 | ISO/IEC 21000-16 | Part 16: Бинарный формат |
| | 2006 | ISO/IEC 21000-17 | Part 17: Фрагментарное распознавание средств MPEG |

| MPEG-A | | | |
|----------------------------------|----------------------|-----------------|--|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Формат применения мультимедиа | 2007 | ISO/IEC 23000-1 | Part 1: Назначение форматов применения мультимедиа |
| | 2006 | ISO/IEC 23000-2 | Part 2: Формат применения музыкального проигрывателя MPEG |

| MPEG-B | | | |
|------------------------------|----------------------|-----------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Системные технологии MPEG | 2006 | ISO/IEC 23001-1 | Part 1: Бинарный формат MPEG для XML |

| MPEG-C | | | |
|-------------------------|----------------------|-----------------|---|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Видеотехнологии MPEG | 2006 | ISO/IEC 23002-1 | Part 1: Точные требования для применения integer-output 8x8 обратного дискретного косинусного преобразования |

| MPEG-D | | | |
|-------------------------|----------------------|-----------------|------------------------|
| Название | Год опубликования | Стандарты ISO | Информация о стандарте |
| Аудиотехнологии MPEG | 2007 | ISO/IEC 23003-1 | Part 1: Окружение MPEG |

ПРИЛОЖЕНИЕ 2

Набор тестовых изображений среды PicLab



1) Lenna



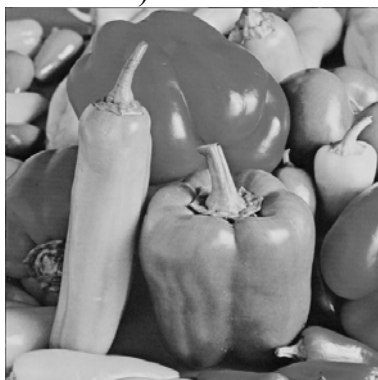
2) Scarlett



3) Elaine



4) Barbara



5) Peppers



6) Tank



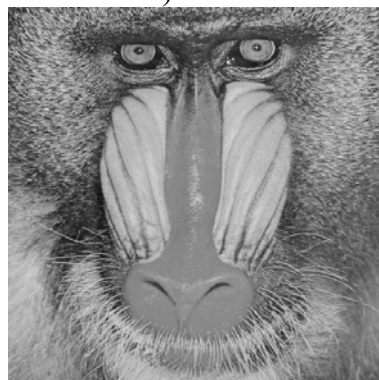
7) Boat



8) Stream and Bridge



9) San Diego



10) Baboon

ПРИЛОЖЕНИЕ 3

Обзорная таблица речевых кодеков

| Кодек | Год | Кем стандартизован | Битрейт, кбит/с | Ширина полосы, кГц | Алгоритм. задержка, мс | Вычислительная сложность | MOS |
|-------------------|------|--------------------|----------------------------|--------------------|------------------------|--------------------------|----------------|
| G.711 | 1977 | ITU-T | 64 | 0 – 4 | - | 0,2 MIPS | 4,3 |
| GSM-FR | 1986 | ETSI | 13 | 0 – 4 | 20 | 4,5 MIPS | 3,5 |
| G.728 | 1992 | ITU-T | 16 | 0 – 4 | 0,625 | 36 MIPS | 3,69 |
| GSM-HR | 1993 | ETSI | 5,6 | 0 – 4 | 25 | 25 MIPS | 3,5 |
| G.723.1 | 1995 | ITU-T | 6,3 5,3 | 0 – 4 | 37,5 | 19 MIPS | 3,9 3,65 |
| G.729 | 1996 | ITU-T | 8 | 0 – 4 | 15 | 22 MIPS | 3,92 |
| GSM-EFR | 1997 | ETSI | 12,2 | 0 – 4 | 20 | 18 MIPS | 3,8 |
| GSM-AMR | 1999 | ETSI | от 4,75 до 12,2 | 0 – 4 | 25 | 20 MIPS | - 3,96 |
| iLBC | 2002 | IETF | 13,3 15,2 | 0 – 4 | 25 | 15 MIPS | - 3,9 |
| AMR-WB G.722.2 | 2002 | 3GPP ITU-T | от 6,6 до 23,85 12,2 | 0 – 8 | 26 | 38 WMOPS | - - 4,14 |
| VMR-WB | 2002 | 3GPP2 | 6,6 8,85 12,65 | 0 – 8 | 33,75 | 38 WMOPS | - |
| G.729.1 | 2006 | ITU-T | от 8 до 32 | 0 – 4 0 – 8 | 48,9375 | 36 WMOPS | 3,7 - |

ITU-T (МСЭ-T) – Международный союз электросвязи, сектор стандартизации электросвязи;

ETSI – Европейский институт по стандартизации в области телекоммуникаций;

IETF (Internet Engineering Task Force) – открытое международное сообщество проектировщиков, учёных, сетевых операторов и провайдеров, которое занимается развитием протоколов и архитектуры Интернет;

3GPP (3rd Generation Partnership Project) – консорциум, разрабатывающий спецификации для мобильной телефонии третьего поколения;

3GPP2 (3rd Generation Partnership Project 2) – консорциум, разрабатывающий спецификации для CDMA2000;

MIPS (Million Instructions Per Second) – величина, показывающая сколько миллионов инструкций в секунду выполняет процессор;

WMOPS (Weighted Million Operations Per Second) – величина, показывающая сколько миллионов взвешенных операций в секунду выполняет процессор.

СОДЕРЖАНИЕ

| | |
|---|----|
| 1. АЛГОРИТМЫ СЖАТИЯ ИЗОБРАЖЕНИЙ | 6 |
| 1.1. Алгоритмы сжатия без потерь | 7 |
| 1.1.1. Алгоритм кодирования длин серий (RLE) | 7 |
| 1.1.2. Алгоритм LZW | 8 |
| 1.1.3. Алгоритм Хаффмана | 10 |
| 1.1.4. Арифметическое кодирование | 15 |
| 1.2. Алгоритмы кодирования с преобразованием | 19 |
| 1.2.1. Ухудшение качества изображения | 20 |
| 1.2.2. Преобразование Фурье | 22 |
| 1.2.3. Двумерное преобразование Хартли | 23 |
| 1.2.4. Двумерное дискретное преобразование Уолша | 25 |
| 1.2.5. Дискретное косинусное преобразование | 27 |
| 1.2.6. Преобразование Карунена-Лоэва | 30 |
| 1.3. Сжатие изображений на основе вейвлет-преобразования | 32 |
| 1.3.1. Некоторые сведения о вейвлет-преобразовании | 32 |
| 1.3.1.1. Непрерывное вейвлет-преобразование | 32 |
| 1.3.1.2. Ортогональное дискретное вейвлет-преобразование .. | 33 |
| 1.3.1.3. Кратномасштабный анализ | 33 |
| 1.3.1.4. Масштабирующая функция | 34 |
| 1.3.1.5. Материнский (базисный) вейвлет | 34 |
| 1.3.1.6. Биортогональные масштабирующие функции и вейвлеты | 34 |
| 1.3.1.7. Ортогональные масштабирующие функции и вейвлеты | 35 |
| 1.3.1.8. Быстрое вейвлет-преобразование | 35 |
| 1.3.1.9. Целочисленное вейвлет-преобразование | 37 |
| 1.3.1.10. Лифтинговая схема вычисления вейвлет- преобразования | 38 |
| 2. СТАНДАРТ JPEG | 39 |
| 2.1. Создание и развитие стандарта JPEG | 40 |
| 2.2. Алгоритм сжатия без потерь – JPEG-LS | 41 |
| 2.3. Алгоритм сжатия – Baseline JPEG | 43 |
| 2.4. Основные характеристики стандарта JPEG | 51 |
| 3. СТАНДАРТ JPEG2000 | 53 |
| 3.1. Части стандарта JPEG2000 | 54 |
| 3.2. Создание и развитие стандарта JPEG2000 | 55 |
| 3.3. Основы кодирования в JPEG2000 | 57 |
| 3.3.1. Предварительная обработка изображения | 57 |
| 3.3.2. Сжатие | 59 |
| 3.4. Характеристики стандарта JPEG2000 | 66 |

| | |
|--|-----|
| 3.5. Алгоритм SPIHT | 69 |
| 3.5.1. Прогрессивная передача в алгоритме SPIHT | 69 |
| 3.5.2. Формализованный алгоритм кодера SPIHT | 70 |
| 3.5.3. Алгоритм сортировки разделением множеств | 71 |
| 3.5.4. Пространственно ориентированное дерево | 72 |
| 3.5.5. Алгоритм кодирования | 74 |
| 3.5.6. Основные характеристики алгоритма SPIHT | 76 |
| 4. ОЦЕНКА КАЧЕСТВА СЖАТЫХ ИЗОБРАЖЕНИЙ | 78 |
| 4.1. Классификация критериев оценки качества | 78 |
| 4.2. Субъективные критерии качества | 80 |
| 4.3. Объективные критерии качества | 81 |
| 4.3.1. Пиковое отношение сигнал/шум | 82 |
| 4.3.2. Проблемы использования ПОСШ в качестве критерия оценки качества | 83 |
| 4.3.3. Модификация ПОСШ с учетом характеристик зрительной системы человека | 87 |
| 4.3.4. Универсальный индекс качества | 88 |
| 4.3.5. Коэффициент структурного подобия | 93 |
| 4.4. Сравнительный анализ алгоритмов сжатия изображений относительно критериев ПОСШ и УИК | 99 |
| 5. ОСНОВНЫЕ СВЕДЕНИЯ О ЦИФРОВЫХ ВИДЕОСИГНАЛАХ | 105 |
| 5.1. Пространственное и временное сэмплирование | 106 |
| 5.2. Кадры и полукадры | 107 |
| 5.3. Кадры и полукадры | 108 |
| 5.3.1. Цветовое пространство RGB | 108 |
| 5.3.2. Цветовое пространство YCbCr | 109 |
| 5.3.3. Форматы сэмплирования YCbCr | 111 |
| 6. ОСНОВЫ ВИДЕОКОДИРОВАНИЯ | 114 |
| 6.1. Видеокодек | 115 |
| 6.2. Временная модель | 117 |
| 6.2.1. Прогноз по предыдущему видеокадру | 117 |
| 6.2.2. Изменения вследствие движения | 118 |
| 6.2.3. Оценка и компенсация движения на основе блоков | 119 |
| 6.2.4. Прогноз макроблока с компенсированным движением | 120 |
| 6.2.5. Размер блока компенсации движения | 122 |
| 7. СТАНДАРТЫ MPEG-4 и H.264 | 123 |
| 7.1. Развитие стандартов | 123 |
| 7.1.1. ISO MPEG | 123 |
| 7.1.2. ITU-T VCEG | 125 |
| 7.1.3. JVT | 125 |
| 7.1.4. История развития | 126 |

| | |
|---|-----|
| 7.1.5. Процесс разработки стандартов | 127 |
| 7.2. Использование стандарта | 129 |
| 7.2.1. Что регламентирует стандарт? | 129 |
| 7.2.2. Понимание стандартов | 129 |
| 7.2.3. Подчинение стандартам | 130 |
| 7.3. Обзор стандарта MPEG-4 (Part 2) | 131 |
| 7.4. Обзор стандарта H.264 (MPEG-4 Part 10) | 134 |
| 7.5. Сравнение MPEG-4 и H.264 | 136 |
| 7.6. Родственные стандарты | 138 |
| 7.6.1. MPEG-1 и MPEG-2 | 138 |
| 7.6.2. H.261 и H.263 | 139 |
| 8. КОДИРОВАНИЕ РЕЧЕВЫХ СИГНАЛОВ | 140 |
| 8.1. Кодирование формы волны речевого сигнала | 140 |
| 8.1.1. Рекомендация ITU-T G.711 | 141 |
| 8.1.2. Рекомендация ITU-T G.726 | 141 |
| 8.1.3. Рекомендация ITU-T G.722 | 142 |
| 8.2. Кодирование речи с низкой скоростью цифрового потока | 143 |
| 8.2.1. Стандарт MIL-STD 3005 | 143 |
| 8.2.2. Стандарт NATO STANAG 4591 | 145 |
| 8.2.3. Работы ITU-T в области стандартизации кодера с битрейтом 4 кбит/с | 146 |
| 8.3. Гибридное кодирование | 147 |
| 8.4. Методы оценки разборчивости речи | 150 |
| 8.4.1. Субъективные методы оценки разборчивости | 153 |
| 8.4.2. Тесты MOS | 156 |
| 8.5. Объективные методы оценки качества речи | 158 |
| 8.5.1. Отношение сигнал-шум | 159 |
| 8.5.2. Логарифмическое спектральное расстояние | 160 |
| 8.5.3. Современные методы объективной оценки качества речи | 161 |
| 8.5.4. Структура алгоритма PESQ | 163 |
| 9. КОДИРОВАНИЕ АУДИОИНФОРМАЦИИ | 167 |
| 9.1. Кодеры MPEG-1 и MPEG-2 | 167 |
| 9.2. Формат кодирования MP3 | 168 |
| 9.2.1. История создания MP3 | 169 |
| 9.2.2. Качество MP3 | 169 |
| 9.2.3. Версии, режимы и управление кодированием MP3 | 170 |
| 9.3. Кодеры MPEG-2 AAC и MPEG-4 HE-AAC | 172 |
| 9.4. Сравнительный анализ качества кодирования | 173 |
| Литература | 175 |
| ПРИЛОЖЕНИЯ | 177 |

Учебное издание

Приоров Андрей Леонидович, **Хрящев** Владимир Вячеславович

ОБРАБОТКА И ПЕРЕДАЧА МУЛЬТИМЕДИЙНОЙ ИНФОРМАЦИИ

Учебное пособие

Редактор, корректор М. В. Никулина

Компьютерная верстка Ю. А. Лукашевич

Подписано в печать 05.02.10. Формат 60×84/16. Бумага офсетная.
Гарнитура «Times New Roman». Усл. печ. л. 10,93 Уч.-изд. л. 10,48.
Тираж 100 экз. Заказ .

Оригинал-макет подготовлен
в редакционно-издательском отделе ЯрГУ
Ярославский государственный университет
150000, Ярославль, ул. Советская, 14

Отпечатано
ООО «Ремдер» ЛР ИД № 06151 от 26.10.01.
150049, г. Ярославль, пр. Октября, 94, оф. 37,
тел. (4852) 73-35-03