

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

«Образование»

А.Ю. Тропченко, А.А. Тропченко

МЕТОДЫ СЖАТИЯ ИЗОБРАЖЕНИЙ, АУДИОСИГНАЛОВ И ВИДЕО

**Учебное пособие
по дисциплине «Теоретическая информатика»**



Санкт-Петербург

2009

Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие – СПб: СПбГУ ИТМО, 2009. – 108 с.

Учебное пособие охватывает основные методы сжатия изображений, аудиосигналов и видео, которые составляют отдельную важную группу методов обработки данных. Материал пособия разбит на 4 раздела, включая введение. В каждом разделе, кроме введения, приведены теоретические сведения об используемых для решения соответствующих задач математических методах. Пособие может быть использовано при подготовке магистров по направлению 230100.68 «Информатика и вычислительная техника», а также инженеров по специальности 23.01.01.65.

Рекомендовано Советом факультета Компьютерных технологий и управления
10 ноября 2009 г., протокол №4



В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

©Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2009

© Тропченко А.Ю., Тропченко А.А., 2009

СОДЕРЖАНИЕ

	стр.
Введение	4
1. МЕТОДЫ СОКРАЩЕНИЯ ИНФОРМАЦИОННОЙ ИЗБЫТОЧНОСТИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ.....	5
1.1. Представление цифровых изображений	5
1.2. Классификация методов сжатия. Основные характеристики.....	8
1.3. Алгоритмы сжатия изображений без потерь.....	15
1.3.1. Сжатие способом кодирования серий (RLE).....	15
1.3.2. Сжатие по методу Хаффмана.....	15
1.3.3. Алгоритм Лемпеля-Зива (LZ-compression).....	16
1.3.4. Алгоритм Лемпеля-Зива-Велча (Lempel-Ziv-Welch -LZW).....	17
1.3.5. Алгоритм JBIG.....	17
1.3.6. Алгоритм Lossless JPEG	18
1.4. Алгоритмы сжатия с потерями	18
1.4.1. Метод усеченного блочного кодирования (УБК)	18
1.4.2. Сжатие по стандарту JPEG.....	20
1.4.3. Сжатие по методу WIC (Wavelet Image Compression).....	26
1.4.4. Фрактальное сжатие изображений	31
2. АЛГОРИТМЫ СЖАТИЯ АУДИОСИГНАЛОВ	43
2.1. Параметры звукового сигнала.....	43
2.1.1. Критические полосы слуха.....	44
2.1.2. Пороги слышимости при маскировке	45
2.1.3. Цифровой звук.....	47
2.2. Основные виды алгоритмов сжатия аудиосигналов.....	48
2.3. Степень ухудшения качества сигнала при квантовании.....	54
2.4. Алгоритмы сжатия аудиосигналов.....	56
2.4.1. Нелинейная ИКМ	56
2.4.2. Кодирование в частотных поддиапазонах	61
2.4.3. Аудио MPEG-1.....	62
2.5. Общая схема аудиокодера и аудиодекодера MPEG	63
2.6. Анализ возможностей алгоритмов сжатия аудиосигналов	69
3. СТАНДАРТЫ СЖАТИЯ ВИДЕОДАНЫХ.....	74
3.1. Особенности сжатия видеоданных.....	74
3.2. Основные процедуры сжатия видеоданных	77
3.2.1. Цветовые пространства и их преобразование	78
3.2.2. Форматы семплирования	81
3.2.3. Устранение пространственной статистической избыточности.....	85
3.3. Модель DPCM/DCT видеокодека.....	86
3.4. Анализ возможностей современных стандартов сжатия видеоданных	91
Литература	105

Введение

Бурное развитие средств вычислительной техники привело к появлению различных мультимедийных приложений и программ, в которых используются тексты, изображения, анимированные фрагменты и звук. Для хранения звука требуется места меньше, чем для видео, но больше, чем для текста и изображений.

В системах связи и радиовещания ограничены возможности выбора свободных частот диапазонов, поэтому остро стоит задача понижения скорости цифровых аудио и видео потоков без снижения субъективного качества воспроизведения.

В области цифровой звукозаписи для повышения качества воспроизведения звука требуется повышение частоты дискретизации и числа разрядов без увеличения размеров носителя и сокращения времени записи [4].

Эти и другие задачи привели к созданию алгоритмов сжатия аудио сигналов, основанных на различных моделях слухового восприятия.

В настоящее время кодирование звука используется в европейских цифровых системах радиовещания DAB (Digital Audio Broadcasting – цифровое аудио вещание), DRM (Digital Radio Mondiale – всемирное цифровое радио), американской Dolby AC-3 (ATSC, Advanced Television System Committee – комитет по улучшенным телевизионным системам), оптических дисках системы DVD-Audio, магнитооптических мини-дисках, звуковых дисках системы MP3, в сети Internet для передачи высококачественного звука.

Наиболее широкое распространение получили следующие алгоритмы компрессии цифровых аудио данных, используемых в новейших системах цифрового радиовещания и звукового сопровождения телевидения:

- MPEG-1 ISO/IEC 11172-3;
- MPEG-2 ISO/IEC 13818-3 и 13818-7 AAC;
- MPEG-4 ISO/IEC 14496-3;
- ATSC Dolby AC-3 (A/52).

Успехи цифровой видеоиндустрии (прежде всего широковещательного цифрового видео и DVD-видео) базировались на международном стандарте ISO/IEC 13818, широко известном под аббревиатурой MPEG-2 (назван так по имени рабочей группы экспертов по движущимся изображениям, разработавшей этот стандарт, - Moving Picture Experts Group). Необходимость лучшего сжатия породило разработку дальнейших стандартов видеосжатия, известных под названиями ISO/IEC 14496 Part 2 (MPEG-4 Visual) и Рекомендация организации ITU-E H.264/ISO/IEC 144496 Part 10 (сокращенно H.264). Стандарты MPEG-4 Visual и H.264 имеют общее происхождение и многие общие черты. Они оба были разработаны на основе более ранних стандартов сжатия. Однако они развивают старые стандарты в существенно различных направлениях.

1. МЕТОДЫ СОКРАЩЕНИЯ ИНФОРМАЦИОННОЙ ИЗБЫТОЧНОСТИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

В последнее время наблюдается бурное развитие телекоммуникационных систем, предназначенных для приема и передачи видеоданных.

Решение подобной задачи стало возможным благодаря существенному увеличению емкости памяти и вычислительной мощности технических средств, входящих в состав телекоммуникационных систем. В состав таких средств входят универсальная или специализированная ЭВМ, специализированные устройства ввода-вывода изображений, средства для хранения или архивации видеоинформации и соответствующее программное обеспечение. В общем случае комплекс подобных средств должен также обеспечивать ввод, вывод и передачу изображений различной физической природы.

Под вводом изображения понимаются процедуры преобразования исходного изображения к виду, удобному для вычислительной системы. Ввод может производиться как со стандартных периферийных устройств ЭВМ (ВЗУ, сканеров), так и с нестандартных по отношению к ЭВМ устройств. К последним относятся, например, телевизионные камеры и ПЗС-линейки.

Под выводом изображения понимается оперативная визуализация на видеомониторе, архивация с целью долговременного хранения и документирование необходимой информации.

Передача изображений включает в себя обмен изображениями между различными блоками системы обработки и обмен изображениями по каналам передачи данных между системой и устройствами, не входящими в ее состав.

Следует отметить, что выполнение различных функций может быть возложено на функционально-ориентированные рабочие станции на базе персональных ЭВМ, подключенных к локальной сети с выходом в глобальную сеть (рис.1.1).

1.1. Представление цифровых изображений

Компьютерное изображение в его цифровом представлении является набором значений интенсивностей светового потока, распределенных по конечной площади.

Для простоты рассмотрим сначала монохромные изображения. Интенсивность излучаемой световой энергии с единицы поверхности в точке с координатами (ξ, η) изображения можно представить некоторым числом $B(\xi, \eta)$. Единичный элемент изображения, характеризуемый определенным значением (ξ, η) , называется пикселем, а величина $z = f(\xi, \eta)$ - яркостью [4].

Прежде чем рассмотреть алгоритмы сжатия изображений, необходимо определить что в дальнейшем будет пониматься под изображением.

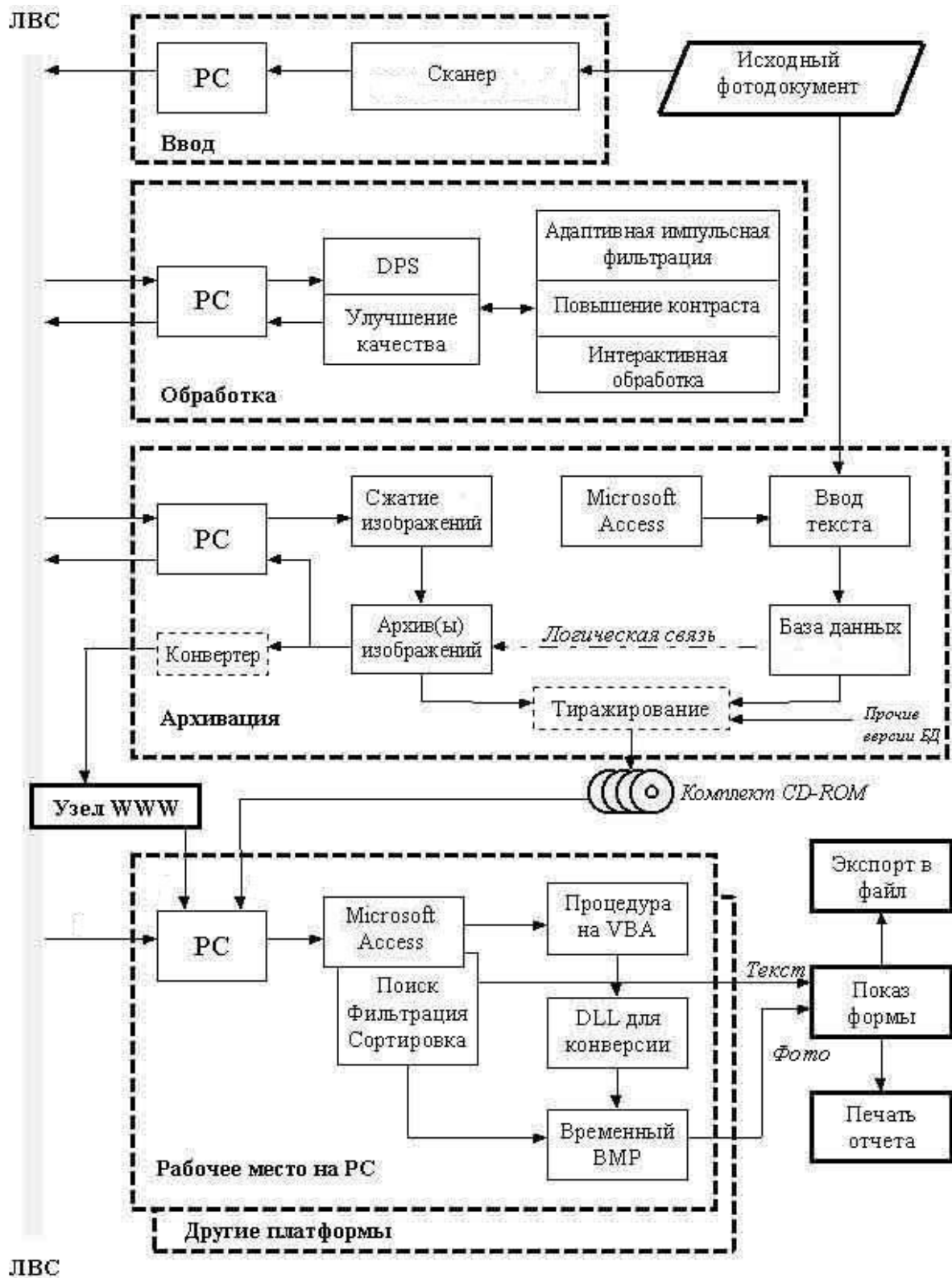


Рис.1.1. Технические средства телекоммуникационной системы для передачи видеоданных и их функциональное назначение.

С математической точки зрения, изображения в градациях серого можно представить как вещественную функцию I двух вещественных переменных x и y . Функция $I(x,y)$ изображения в общем случае определяется в прямоугольной области, но для удобства исследований в работе все изображения определяются в квадратных областях, т.е. $x \in [0; W]$, а $y \in [0; H]$, где W – ширина изображения, а H – высота изображения и $W=H$.

Все изображения можно подразделить на две группы: с палитрой и без неё. У изображений с палитрой в пикселе (одном из отчётов изображения – значение функции $I(x,y)$ для конкретного x_i и y_i) храниться число – индекс в некотором одномерном векторе цветов, называемом палитрой. Палитры обычно бывают 8, 16 и 256 – цветов[4].

Изображения без палитры обычно бывают в определенной системе цветопредставления или в градациях серого. В градациях серого значения каждого из пикселей определяется как яркость точки. Наиболее часто встречаются изображения с 2-мя, 16-ю и 256-ю уровнями серого.

Если изображение представлено в какой-то системе цветопредставления, то каждый её пиксель является структурой, описывающий компоненты цвета. Наиболее распространённой системой цветопредставления, используемой в электронных и компьютерных системах, является система RGB. В этой системе цвет определяется как комбинация красного, зелёного и синего цвета. И на каждую из составляющих приходится по одному байту. Существуют и другие системы цветопредставления, такие как CMYK, CIE, YUV и YCrCb [3,4].

Для того чтобы корректнее оценивать степень сжатия изображения, и применимости того или иного алгоритма сжатия к данному изображению вводится понятие класса изображения [4].

Под классом цифрового изображения понимается совокупность изображений, применение к которым, алгоритм сжатия даёт качественно одинаковый результат. Например, для одного класса алгоритм сжатия даёт превосходный коэффициент сжатия, а для другого класса изображений наоборот, увеличивает объём сжимаемого файла [3].

Условно можно выделить следующие классы изображений:

- изображения с небольшим количеством цветов и большими областями, заполненными одним цветом. В изображении отсутствуют плавные переходы цветов. К таким классам обычно относится деловая графика, научно-техническая, инженерная или плакатная графика;
- изображения с плавными переходами цветов, построенные на компьютере: графика презентаций и виртуальные модели;
- фотореалистичные изображения, полученные после цифровой фотосъёмки, сканирования, а также постобработка этих изображений.

Можно выделить и специфические классы изображений, такие как рентгеновские снимки, томографические изображения, радиолокационные планы местности и т.д. Но для сравнения алгоритмов сжатия изображений всегда необходимо определять класс изображений, с которыми они работает.

В процессе работы с изображениями приложения, осуществляющие обработку, предъявляют различные требования к алгоритмам сжатия изображений. Из-за специфики приложений такие требования иногда могут противоречить друг другу. В общем случае можно выделить следующие требования к алгоритмам сжатия изображений:

- высокая степень компрессии;
- высокое качество сжатого изображения (данное требование противоречит выполнению предыдущего требования, поэтому всегда приходится искать компромисс между степенью сжатия и качеством восстановленного изображения);
- высокая скорость компрессии (данное требование актуально для приложений, занимающихся кодированием изображений в реальном масштабе времени: цифровых фотоаппаратов, видеокамер);
- высокая скорость декомпрессии (данное требование актуально почти для всех приложений).
- возможность показать приблизительное изображение, не дожидаясь полной его загрузки (данное требование актуально для сетевых приложений и для приложений, занимающихся передачей больших изображений).
- учёт специфики изображения (данное требование реализуют алгоритмы сжатия, основанные на определении «области особого назначения» (ROI – regions of interest)).

1.2. Классификация методов сжатия. Основные характеристики

Все методы сжатия информации основаны на том простом предположении, что набор данных всегда содержит избыточные элементы. Сжатие достигается за счет поиска и кодирования избыточных элементов [3,7].

Поток данных об изображении имеет существенное количество излишней информации, которая может быть устранена практически без заметных для глаза искажений. При этом различают два типа избыточности.

Статистическая избыточность связана с корреляцией и предсказуемостью данных. Эта избыточность может быть устранена без потери информации, исходные данные при этом могут быть полностью восстановлены [3,4]. Наиболее известные методы эффективного кодирования символов основаны на знании частоты каждого символа присутствующего в сообщении. Зная эти частоты, строят таблицу кодов, обладающую следующими свойствами:

- различные коды могут иметь различное количество бит;
- коды символов с большей частотой встречаемости, имеют меньше бит, чем коды символов с меньшей частотой;
- хотя коды имеют различную битовую длину, они могут быть восстановлены единственным образом, т.е. коды строятся как префиксные.

Этими свойствами обладает известный алгоритм Хаффмана [4].

Визуальная (субъективная) избыточность, которую можно устранить с частичной потерей данных, мало влияющих на качество воспроизводимых изображений; это - информация, которую можно изъять из изображения, не нарушая визуально воспринимаемое качество изображений.

Устранение визуальной избыточности изображений является основным резервом сокращения передаваемой информации [3,7]. Для оптимизации процесса кодирования в целях обеспечения передачи наименьшего объема информации необходимо, с одной стороны, не передавать избыточную информацию, а с другой, - не допустить чрезмерной потери качества изображения.

До сих пор не существует простой и адекватной модели визуального восприятия изображений, пригодной для оптимизации их кодирования [7].

Задача сжатия изображения состоит из двух основных частей: кодирование и декодирование. Если декодированное изображение всегда в точности соответствует кодируемому изображению, то такой алгоритм кодирования-декодирования называется алгоритмом сжатия без потерь. Если декодированное изображение отличается от закодированного, то подобный алгоритм называют алгоритмом сжатия с потерями. Общая схема процесса сжатия изображения представлена на рисунке 1.2 [3].

Рассмотрим этапы процедуры сжатия данных в общем виде. Любой метод сжатия реализует три основных этапа (рис. 1.2):

- кодирование или первичное сжатие;
- вторичное сжатие;
- декодирование или восстановление изображения.

На первом этапе выполняется преобразование исходных данных из одной формы представления в другую. В частности, при сжатии изображений в зависимости от вида алгоритма сжатия может быть выполнен переход от исходного изображения к следующим видам представления (табл.1.1):

На втором этапе компоненты преобразования квантуются и приводятся к виду удобному для статистического кодирования, а затем кодируются. На этом этапе обеспечивается уплотнение информационного потока.

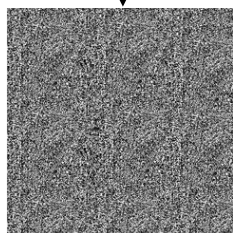
Существует несколько различных подходов к проблеме сжатия информации. Одни имеют весьма сложную теоретическую математическую базу, другие основаны на свойствах информационного потока и алгоритмически достаточно просты. Любой способ, реализующий сжатие данных, предназначен для снижения объема выходного потока информации помощи обратимого или необратимого преобразования. Поэтому все способы сжатия можно разделить на две категории: обратимое и необратимое сжатие [3,7]. Методы сжатия цифровых изображений можно классифицировать по их основным характеристикам, таким как: точность восстановления, симметричность основного преобразования и тип используемого преобразования [7].

Исходное изображение**Кодирование**

Дискретно-косинусное (JPEG),
преобразования Хартли Уолша-Адамара,
Вейвлет, Фрактальное
преобразование

Вторичное сжатие

Метод Хаффмана, RLE, ZIP, HA методы
сжатия

**Сжатое изображение****Декодирование**

Необязательная постобработка
восстановленного изображения

**Декодированное
изображение**

Рис.1.2. Основные этапы сжатия цифровых изображений

Таблица 1.1

Исходное изображение	Преобразованное изображение
Матрица пикселей (значения интенсивности)	Матрица компонент спектра (спектральные преобразования)
	Набор коэффициентов преобразования (фрактальное сжатие)
	Описание объектов изображения (сжатие с распознаванием)

Классификация наиболее распространенных методов сжатия приведена на рис.1.3.

Обратимое сжатие (сжатие без потерь). Обратимое сжатие всегда приводит к снижению объема выходного потока информации без изменения его информативности, т.е. без потери информационной структуры. Более того, из выходного потока, при помощи восстанавливающего алгоритма, можно получить входной [3].

Необратимое сжатие (сжатие с потерями). Под необратимым сжатием подразумевают такое преобразование входного потока данных, при котором выходной поток, основанный на определенном формате информации, представляет достаточно похожий по внешним характеристикам на входной поток объект, однако отличается от него объемом. Степень сходства входного и выходного потоков определяется степенью соответствия некоторых свойств объектов (т.е. сжатой и несжатой информацией в соответствии с некоторым определенным форматом данных), представляемого данным потоком информации.

Такие алгоритмы используются для сжатия, например данных растровых графических файлов с низкой степенью повторяемости байтов в потоке. При таком подходе используется свойство структуры формата графического файла и возможность представить графическую картинку приблизительно схожую по качеству отображения (для восприятия человеческим глазом) несколькими способами. Поэтому, кроме степени или величины сжатия, в таких алгоритмах возникает понятие качества, т.к. исходное изображение в процессе сжатия изменяется. Под качеством можно понимать степень соответствия исходного и результирующего изображения. Для графических файлов такое соответствие определяется визуально, хотя имеются и соответствующие формализованные методики и оценки [4]. Необратимое сжатие невозможно применять в областях, в которых необходимо иметь точное соответствие информационной структуры входного и выходного потоков.

Методы сжатия без потерь используются в основном в научных и медицинских приложениях, когда потеря информации недопустима или сами шумы изображения являются главной информацией, например в системах оценки качества оптико-электронных систем. Коэффициент сжатия,

достигаемый этими методами не более 1,5 для реальных сцен. Методы сжатия с потерями позволяют получить существенно большие коэффициенты сжатия.

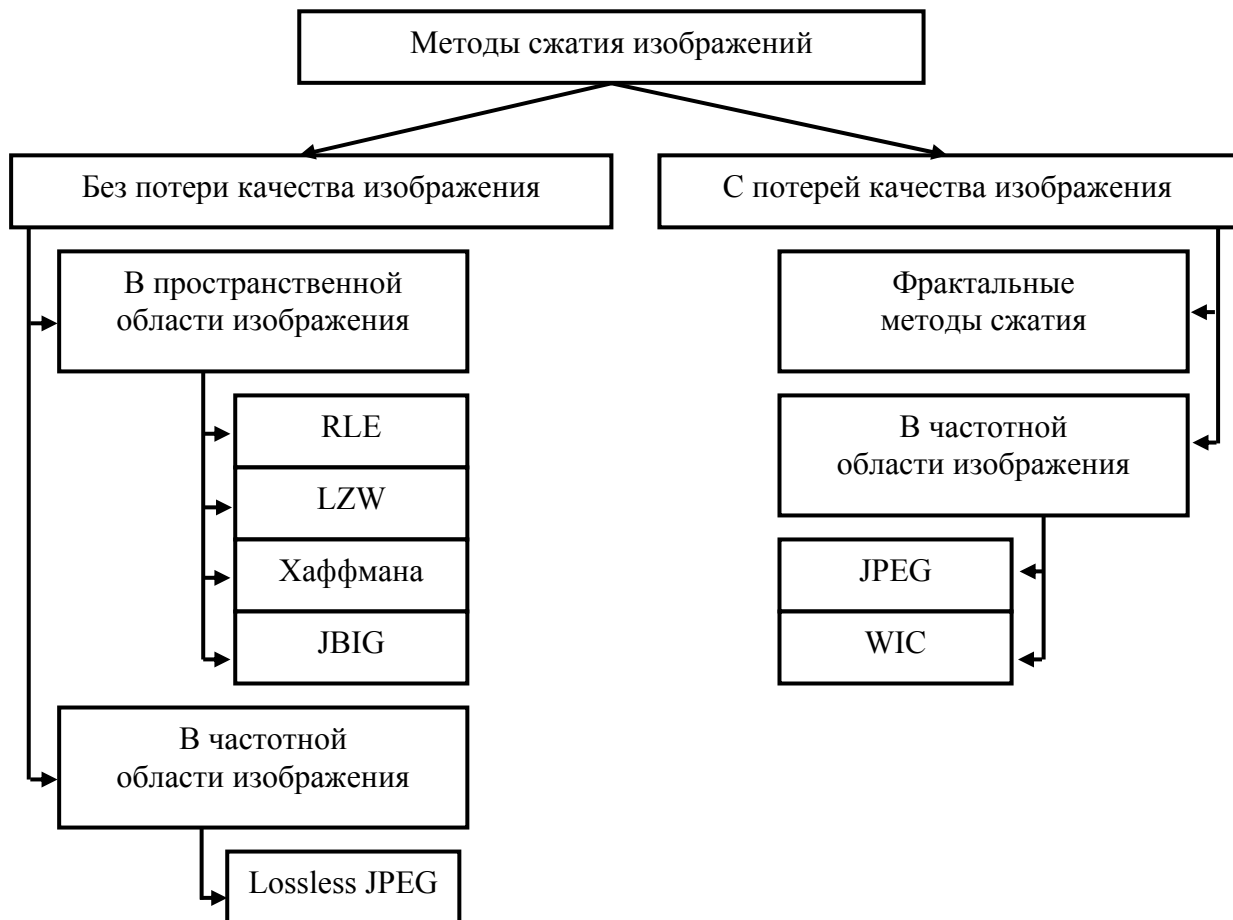


Рис.1.3. Классификация методов сжатия изображений

Однако при этом происходит искажение исходного изображения, ухудшение его качества. В связи с этим при сравнении различных методов сжатия помимо коэффициента сжатия нужно учитывать качество восстановления изображения.

Для симметричных методов сжатия процедуры сжатия и восстановления однотипны. Время сжатия и восстановления для таких методов сравнимы. Для несимметричных методов процедура сжатия отличается от процедуры восстановления и обычно занимает большее машинное время.

Определим основные величины, характеризующие метод сжатия.

1. Коэффициент сжатия ($K_{сж}$).

$$K_c = \frac{V_1}{V_2}$$

Этот параметр определяет во сколько раз файл, хранящий сжатое изображение, меньше файла, хранящего исходное изображение. Величины V1 и V2 выражаются в байтах. Ксж – величина безразмерная.

2. Оценка качества декодированного изображения.

Одна из проблем машинной графики заключается в том, что до сих пор не найден адекватный критерий оценки потерь качества изображения [4].

Качество теряется при оцифровке, при переводе в ограниченную палитру цветов или в другое цветовое пространство, а так же при сжатии изображений с потерями.

Пусть есть два изображения: $f(x, y)$ - оригинал, и $\hat{f}(x, y)$ - восстановленное изображение размером $M \times N$, тогда одним из простых критерием оценки потери качества является среднеквадратическое отклонение значений пикселей сжатого изображения от оригинала:

$$d(x, y) = \sqrt{\frac{\sum_{x,y}^{M,N} (f(x, y) - \hat{f}(x, y))^2}{M \cdot N}} \quad (1.1)$$

По этому критерию изображение будет сильно испорчено при изменении яркости всего на 5%. В тоже время изображение со снегом, резким изменением цвета отдельных точек будут признаны почти не изменившимися.

Другим критерием является максимальное отклонение от оригинала:

$$d(x, y) = \max_{x,y} |f(x, y) - \hat{f}(x, y)| \quad (1.2)$$

Данная мера крайне чувствительна к биению отдельных пикселей, т.е. в изображении может измениться только один пиксель, и данный критерий признает изображение сильно испорченным.

На практике используемой мерой качества изображения является критерий соотношения сигнал/шум (PSNR).

$$d(x, y) = 10 \log_{10} \frac{255^2 \cdot M \cdot N}{\sum_{x,y}^{M,N} (f(x, y) - \hat{f}(x, y))^2} \quad (1.3)$$

Эта мера аналогична среднеквадратическому отклонению, но пользоваться ей удобнее из-за логарифмического масштаба шкалы.

Лучше всего потери в качестве оценивает человеческий глаз. Сжатие изображения можно считать отличной, если на глаз невозможно отличить оригинал от сжатого изображения. Но на практике при сжатии с потерями в изображение всегда вносятся какие-либо искажения заметные при сравнении оригинала и сжатого изображения.

К другим наиболее употребляемым критериям оценки качества изображения относятся:

Средняя разность:

$$AD = \sum_{x=1}^M \sum_{y=1}^N \left[f(x, y) - \hat{f}(x, y) \right] / (MN) \quad (1.4)$$

Коэффициент кросс-корреляции:

$$K = \sum_{x=1}^M \sum_{y=1}^N f(x, y) \cdot \hat{f}(x, y) / \sum_{x=1}^M \sum_{y=1}^N [f(x, y)]^2 \quad (1.5)$$

Верность изображения (image fidelity):

$$AD = 1 - \left(\sum_{x=1}^M \sum_{y=1}^N \left[f(x, y) - \hat{f}(x, y) \right]^2 / \sum_{x=1}^M \sum_{y=1}^N [f(x, y)]^2 \right) \quad (1.6)$$

Среди большого числа критериев оценки качества изображения в работе для оценки качества восстановленного изображения были выбраны среднеквадратическое отклонение и соотношение сигнал/шум (как наиболее распространённые критерии), а для визуальной оценки используется разностное изображение.

3. Время преобразования

Различают две величины – время сжатия ($t_{сж}$) и время восстановления ($t_{восст}$).

Время сжатия состоит из времени работы основного преобразования ($t_{он}$) и времени упаковки (t_{yn}). Время восстановления состоит из времени распаковки ($t_{расп}$) и времени работы обратного преобразования ($t_{об}$):

$$\begin{aligned} t_{сж} &= t_{он} + t_{yn} \\ t_{восст} &= t_{расп} + t_{об} \end{aligned} \quad (1.7)$$

Для того, чтобы корректно оценивать алгоритмы сжатия.восстановления необходимо задать определенные критерии:

- худший, средний и лучший коэффициент сжатия. (Иначе говоря, разброс коэффициента сжатия, если исходные данные будут наихудшими; некий среднестатистический коэффициент для того класса изображений, на который ориентирован алгоритм; и, наконец, лучший коэффициент, причем последний имеет лишь теоретическое значение, поскольку показывает степень сжатия наилучшего изображения);
- класс изображений, на который ориентирован алгоритм (иногда указывают, почему на других классах изображений получаются худшие результаты);
- симметричность - характеризует ресурсоемкость процессов кодирования и декодирования (при этом наиболее важным является отношение времени кодирования ко времени декодирования);
- потери качества (у большинства алгоритмов сжатия с потерей информации существует возможность изменения коэффициента сжатия);
- характерные особенности алгоритма и изображений, к которым его применяют.

1.3. Алгоритмы сжатия изображений без потерь

1.3.1. Сжатие способом кодирования серий (RLE)

Наиболее известный и простой алгоритм сжатия информации обратимым путем - это кодирование серий последовательностей (Run Length Encoding - RLE) [3,7]. Суть данного подхода состоит в замене цепочек или серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений. Проблема всех аналогичных методов заключается лишь в определении способа, при помощи которого распаковывающий алгоритм мог бы отличить в результирующем потоке байтов кодированную серию от других - некодированных последовательностей байтов. Решение проблемы достигается обычно простановкой меток вначале кодированных цепочек. Такими метками могут быть, например, характерные значения битов в первом байте кодированной серии, значения первого байта кодированной серии и т.п. Лучший, средний и худший коэффициенты сжатия - $1/32$, $1/2$, $2/1$.

Данные методы, как правило, достаточно эффективны для сжатия растровых графических изображений (BMP, PCX, TIFF), т.к. последние содержат достаточно длинных серий повторяющихся последовательностей байтов.

Недостатком метода RLE является достаточно низкая степень сжатия или стоимость кодирования файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях. К положительным сторонам алгоритма, пожалуй, можно отнести только то, что он не требует дополнительной памяти при работе, и быстро выполняется. Интересная особенность группового кодирования в формате PCX заключается в том, что степень архивации для некоторых изображений может быть существенно повышена всего лишь за счет изменения порядка цветов в палитре изображений.

1.3.2. Сжатие по методу Хаффмана

Для текстовых файлов чаще других употребляется кодировка Хаффмана, заключающаяся в том, что символы текста заменяются цепочками бит разной длины. Методика Хаффмана гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом символов на букву [4].

Применительно к сжатию изображений в основе такого метода лежит учет частоты появления одинаковых байт в изображении. При этом пикселям исходного изображения, которые встречаются большее число раз, сопоставляется код меньшей длины, а встречающимся редко - код большей длины (т.е. формируется префиксный код переменной длины). Для сбора статистики требуется два прохода по файлу - один для просмотра и сбора

статистической информации, второй - для кодирования [3]. Коэффициенты сжатия: 1/8, 2/3, 1.

При использовании такого метода требуется запись в файл и таблицы соответствия кодируемых пикселей и кодирующих цепочек. Такое кодирование применяется в качестве последнего этапа архивации в JPEG. Методы Хаффмана дают достаточно высокую скорость и умеренно хорошее качество сжатия.

Основным недостатком данного метода является зависимость степени сжатия от близости вероятностей символов к величине 2^{-m} , поскольку каждый символ кодируется целым числом бит. Так, при кодировании данных с двухсимвольным алфавитом сжатие всегда отсутствует, т.к. несмотря на различные вероятности появления символов во входном потоке алгоритм фактически сводит их до 1/2. Такой алгоритм реализован в формате TIFF.

1.3.3. Алгоритм Лемпеля-Зива (LZ-compression)

Суть данного алгоритма состоит в следующем: упаковщик постоянно хранит некоторое количество последних обработанных символов в буфере. По мере обработки входного потока вновь поступившие символы попадают в конец буфера, сдвигая предшествующие символы и вытесняя самые старые. Размеры этого буфера, называемого также скользящим словарем, варьируются в разных реализациях кодирующих систем. Затем, после построения хеш-таблиц, выделяют (путем поиска в словаре) самую длинную начальную подстроку входного потока, совпадающую с одной из подстрок в словаре, и выдают на выход пару (length, distance), где length - длина найденной в словаре подстроки, а distance - расстояние от нее до входной подстроки (то есть фактически индекс подстроки в буфере, вычитенный из его размера). Если такая подстрока не найдена, в выходной поток просто копируется очередной символ входного потока [7].

Существует довольно большое семейство LZ-подобных алгоритмов, различающихся, например, методом поиска повторяющихся цепочек. Один из достаточно простых вариантов этого алгоритма, например, предполагает, что во входном потоке идет либо пара <счетчик, смещение относительно текущей позиции>, либо просто <счетчик> “пропускаемых” байт и сами значения байтов.

При разархивации для пары <счетчик, смещение> копируются <счетчик> байт из выходного массива, полученного в результате разархивации, на <смещение> байт раньше, а <счетчик> (т.е. число равное счетчику) значений “пропускаемых” байт просто копируются в выходной массив из входного потока.

Данный алгоритм является несимметричным по времени, поскольку требует полного перебора буфера при поиске одинаковых подстрок.

К достоинствам LZ можно отнести чрезвычайную простоту алгоритма декомпрессии.

1.3.4. Алгоритм Лемпеля-Зива-Велча (Lempel-Ziv-Welch -LZW)

Данный алгоритм отличают высокая скорость работы как при упаковке, так и при распаковке, достаточно скромные требования к памяти и простая аппаратная реализация. Недостаток - низкая степень сжатия по сравнению со схемой двухступенчатого кодирования. Алгоритм преобразует поток символов на входе в поток индексов ячеек словаря на выходе. Существует довольно большое семейство LZW - подобных алгоритмов, различающихся, например, методом поиска повторяющихся цепочек [7].

Коэффициенты сжатия: 1/1000, 1/4, 7/5. Коэффициент 1/1000 достигается только на одноцветных изображениях размером больше 4 Мб. Ситуация, когда алгоритм увеличивает изображение, встречается крайне редко. Сжатие обеспечивается за счет одинаковых подцепочек в потоке. Алгоритм является почти симметричным, при условии оптимальной реализации операции поиска строки в таблице.

LZW универсален - именно его варианты используются в обычных архиваторах. Он реализован в форматах GIF, TIFF и TGA.

1.3.5. Алгоритм JBIG

Алгоритм разработан группой экспертов ISO (Joint Bi-level Experts Group) специально для сжатия однобитных черно-белых изображений (например, для факсов или отсканированных документов). В принципе может применяться и к 2-х, и к 4-х битовым картинкам. При этом алгоритм разбивает их на отдельные битовые плоскости. JBIG позволяет управлять такими параметрами, как порядок разбиения изображения на битовые плоскости, ширина полос в изображении, уровни масштабирования. Последняя возможность позволяет легко ориентироваться в базе больших по размерам изображений, просматривая сначала их уменьшенные копии. Настраивая эти параметры, можно использовать интересный эффект при получении изображения по сети или по любому другому каналу, пропускная способность которого мала по сравнению с возможностями процессора. Распаковываться изображение на экране будет постепенно, как бы медленно "проявляясь". При этом человек начинает анализировать картинку задолго до конца процесса разархивации. Алгоритм построен на базе Q-кодировщика, патентом на который владеет IBM. Q-кодер также, как и алгоритм Хаффмана, использует для чаще появляющихся символов короткие цепочки, а для реже появляющихся длинные. Однако, в отличие от него, в алгоритме используются и последовательности символов. Характерной особенностью JBIG является резкое снижение степени сжатия при повышении уровня шумов исходного изображения [3,7].

1.3.6. Алгоритм Lossless JPEG

Этот алгоритм разработан группой экспертов в области фотографии (Joint Photographic Expert Group). В отличие от JBIG, Lossless JPEG ориентирован на полноцветные 24-битные изображения [3,4].

Коэффициенты сжатия: $1/20$, $1/2$, 1 .

Стандарт сжатия изображений JPEG включает два способа сжатия: первый предназначен для сжатия без потерь, второй – сжатия с потерей качества. Метод сжатия без потерь, используемый в стандарте lossless JPEG основан на методе разностного (дифференциального) кодирования. Основная идея дифференциального кодирования состоит в следующем. Обычно изображения характеризуются сильной корреляцией между точками изображения. Этот факт учитывается при разностном кодировании, а именно, вместо сжатия последовательности точек изображения x_1, x_2, \dots, x_N , сжатию подвергается последовательность разностей $y_i = x_i - x_{i-1}$, $i = 1, 2, \dots, N$, $x_0 = 0$. Числа y_i называют ошибками предсказания x_i . В стандарте lossless JPEG предусмотрено формирование ошибок предсказания с использованием предыдущих закодированных точек в текущей строке и/или в предыдущей строке.

Lossless JPEG рекомендуется применять в тех приложениях, где необходимо побитовое соответствие исходного и разархивированного изображений.

Попробуем сделать некоторые обобщения. С одной стороны, приведенные алгоритмы достаточно универсальны и покрывают все типы изображений, с другой - они, по сегодняшним меркам, обеспечивают слишком маленький коэффициент архивации. Используя один из алгоритмов без потерь, можно обеспечить коэффициент архивации изображения примерно в два раза. В то же время сжатия с потерями оперируют с коэффициентами 10-200 раз. Помимо возможности модификации изображения, одна из основных причин подобной разницы заключается в том, что традиционные алгоритмы ориентированы на работу с цепочкой. Они не учитывают так называемую "когерентность областей" в изображениях. Идея когерентности областей заключается в малом изменении цвета и структуры изображения на небольшом участке. Все алгоритмы, о которых речь пойдет ниже, были созданы позднее специально для сжатия графики и используют эту идею.

1.4. Алгоритмы сжатия с потерями

1.4.1. Метод усеченного блочного кодирования (УБК)

Название метода отражает тот факт, что изображение разбивается на небольшие прямоугольные куски одинакового размера, называемые блоками. Этот метод в отличие от большинства других подстраивает параметры кодирования не под некоторую усредненную характеристику всего изображения, а под локальные особенности в пределах каждого блока. Это позволяет сохранить мелкие детали изображений. Метод не приводит к

размыванию границ, что характерно для некоторых других алгоритмов. Метод УБК сопоставим с большинством других методов по эффективности сжатия данных и по объему вычислений, требуемых для кодирования, но не имеет конкурентов по простоте декодирования.

Базовый алгоритм УБК строится следующим образом [7]. Изображение, представленное $M \times N$ - матрицей $\|b_{ij}\|$ яркостей пикселей, разбивается на небольшие прямоугольные блоки $m \times n$ элементов. Каждый такой блок обрабатывается независимо от других, поэтому опишем алгоритм обработки одного блока.

Обработка блока начинается с вычисления порога и двух уровней квантования, затем проводится квантование блока на два уровня, после чего следует упаковка проквантованного блока. Для определения уровней квантования сначала вычисляются два первых выборочных момента - среднее значение C и средний квадрат E :

$$C = \frac{1}{m \times n} \sum_i \sum_j b_{ij}, \quad E = \frac{1}{m \times n} \sum_i \sum_j b_{ij}^2 \quad (1.8)$$

(где суммируются элементы изображения в пределах блока) и дисперсия

$$\sigma^2 = E - C^2. \quad (1.9)$$

Пороговая величина квантователя d полагается равной среднему C . Верхний a и нижний b уровни квантования вычисляются по следующим формулам:

$$a = C - \sigma \sqrt{q/(p-q)}, \quad b = C + \sigma \sqrt{(p-q)/q}, \quad (1.10)$$

где $p = m \times n$ - число элементов блока, q - число элементов, превышающих порог d .

Квантование проводится по правилу:

$$s_{ij} = \begin{cases} a, \text{ если } b_{ij} < d \\ b, \text{ если } b_{ij} \geq d \end{cases} \quad (1.11)$$

где s_{ij} - элементы изображения после квантования.

После квантования получается блок, содержащий только уровни a и b . Нетрудно показать, что среднее значение и средний квадрат исходного и проквантованного блоков совпадают. Практически для удобства последующей упаковки вместо a записывается нуль, вместо b - единица. Уровни a и b записываются отдельно.

Упаковка состоит в том, что блок, содержащий только нули и единицы, интерпретируется как двоичное число, имеющее $m \times n$ разрядов. Восстановление закодированного изображения также проводится поблочно и состоит в распаковке и обратной подстановке.

Из алгоритма видно, что степень сжатия непосредственно зависит от размеров блока. Наиболее удовлетворительные результаты, как по степени сжатия, так и по качеству восстановленного изображения были получены при использовании блоков размером 4×4 [3,7].

Описанный выше способ определения порога и уровней квантования не

является единственным. Существует ряд других критериев. Важно, чтобы критерий соответствовал целям последующей обработки изображения и ее конкретным особенностям.

1.4.2. Сжатие по стандарту JPEG

JPEG - один из самых распространенных и достаточно мощных алгоритмов, представляет собой метод сжатия изображений, реализуемый различными способами [4,7]. Работает он как на черно-белых, так и на полноцветных изображениях.

Стандарт JPEG (Joint Photographic Experts Group - Объединенная экспертная группа по фотографии) - формат хранения фотографических изображений, отличающийся хорошим качеством восстановленного изображения. Процесс сжатия изображения JPEG достаточно сложен и часто для достижения приемлемой производительности требует специальной аппаратуры. Схема процедуры сжатия изображений по стандарту JPEG приведена на рис.1.4.

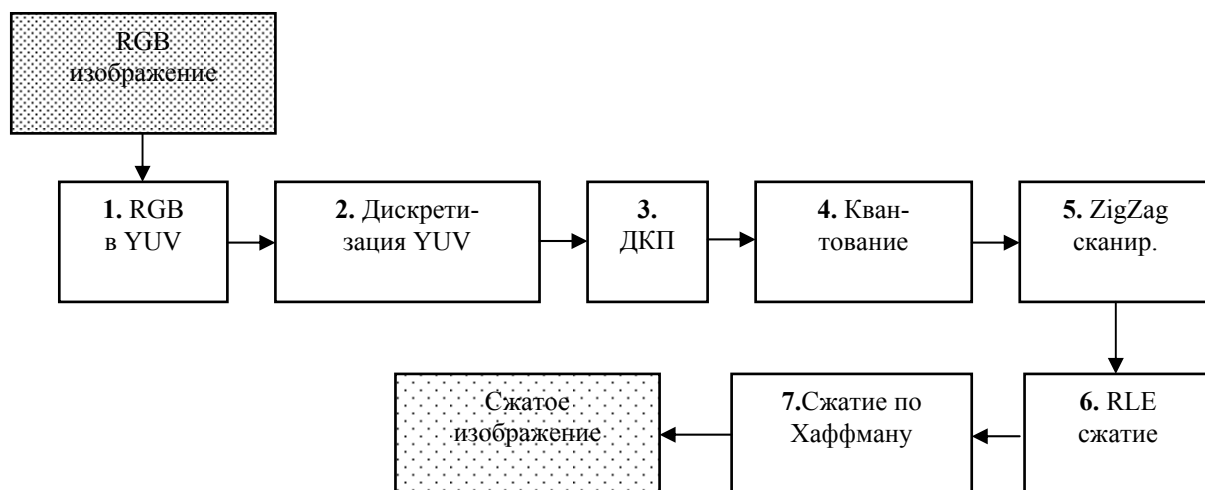


Рис.1.4. Основные этапы процедура сжатия по стандарту JPEG

Кодирование изображений по стандарту JPEG обычно начинается с преобразования цветового пространства из RGB в YUV (известное также под названием YCbCr). Цветное изображение традиционно может рассматриваться как результат сложения трех компонент:

$$X_{ijC} = a_1 X_{ij(R)} + a_2 X_{ij(G)} + a_3 X_{ij(B)} \quad (1.12)$$

В этом выражении a_1, a_2, a_3 - калориметрические коэффициенты.

В типичных изображениях в формате RGB имеется существенная корреляция между цветными компонентами и с точки зрения сжатия изображения этот формат является заведомо избыточным. Как известно, в

стандартах телевизионного вещания используется другое представление изображений, при котором также используются 3 компоненты сигнала, но при этом эти компоненты почти некоррелированы друг с другом. Компоненты R, G и B преобразуются в яркостную компоненту Y и две цветоразностных компоненты U и V , формата YUV .

Преобразование форматов выполняется по формулам:
для преобразования $RGB \rightarrow YUV$

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

и для преобразования $YUV \rightarrow RGB$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} * \begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

В формате YUV компоненты слабо коррелированы. Более того, так как большая часть информации сосредоточена в яркостной компоненте, то будет потеряно мало информации, если выполнить децимацию (прореживание) компонент U и V с коэффициентом 2. При таком прореживании 4 соседние точки (квадрат 2×2) описываются 4 значениями компонент Y и по одному значению компонент U и V .

Результатом является стандартный формат YUV 4:1:1, который, как правило, является входным для большинства видеокодеров. Таким образом, получается сжатие в 2 раза без сколько-нибудь заметного искажения изображения.

Стандарт не обязывает выполнять эту операцию, однако такой подход позволяет повысить эффективность сжатия.

Далее исходное изображение разбивается на матрицу клеток одинакового размера (чаще всего 8×8 пикселей). Такой размер выбран по следующим причинам [7]:

1) С точки зрения аппаратной и программной реализации размер блока 8×8 не накладывает существенных ограничений на размер требуемой памяти.

2) Вычислительная сложность ДКП для блока 8×8 также является приемлемой для большинства вычислительных платформ.

Следующий этап процедуры сжатия данных заключается в преобразовании небольших блоков изображения при помощи двумерного косинусного преобразования (ДКП). Обработка ведется блоками 8×8 пикселей. Выбор ДКП в качестве стандартного решения диктуется следующими причинами:

- Для изображений с сильно коррелированными отсчетами (коэффициент корреляции $> 0,7$) эффективность ДКП в смысле компактности представления данных близка к преобразованию Карунена-Лоэва (это преобразование является оптимальным в том смысле, что оно

ортонормированно и гарантирует некоррелированность коэффициентов преобразования – элементов Y).

- ДКП представляет собой ортогональное сепарабельное преобразование, независимое от изображения, поэтому его вычислительная сложность невелика.

Обработка каждой клетки выполняется независимо и заключается в выполнении ДКП по строкам и столбцам клетки, которое имеет вид:

$$F(u, v) = \frac{C(u, v)}{4} * \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) * \cos\left(\frac{2x+1}{16} * u * \pi\right) * \cos\left(\frac{2y+1}{16} * v * \pi\right) \quad (1.13)$$

В выражении (1.13) множитель $C(u, v)$ является нормирующим и равен $1/\sqrt{2}$ при $u=0$ или $v=0$ и равен единице для остальных значений индексов.

Общим недостатком дискретных ортогональных преобразований является их высокая вычислительная сложность. В связи с этим используются так называемые быстрые алгоритмы выполнения косинусного преобразования. Известные из литературы алгоритмы быстрых преобразований в базисах косинусных функций, хотя и отличаются меньшим числом операций умножения, но требуют дополнительных перекомпоновок после каждой итерации алгоритма. Одномерное косинусное преобразование может быть вычислено через одномерное преобразование Хартли [2]. При этом вначале производится перестановка элементов вектора исходных данных таким образом, что первую половину последовательности составляют нечетные элементы, а вторую половину последовательности - четные элементы в порядке возрастания номеров, а затем выполняется одномерное преобразование Хартли над модифицированным вектором [2]:

$$H_k = \sum_m X_m^{\wedge} \text{cas}[\pi(2\pi km / N)] \quad (1.14)$$

(где $\text{cas}[\dots] = \cos[\dots] + \sin[\dots]$), которое может быть вычислено по быстрому алгоритму; после чего выполняются дополнительные вычисления с элементами вектора результатов преобразования Хартли:

$$c_k^{\wedge} = \{H_k [\cos(\pi k / 2N) - \sin(\pi k / 2N)] + H_{N-k} [\cos(\pi k / 2N) + \sin(\pi k / 2N)]\} / 2 \quad (1.15)$$

где $k = [1, \dots, N-1]$, причем $c_0^{\wedge} = H_0$.

Процедура обратного косинусного преобразования отличается от прямого алгоритма другой последовательностью вычислений - вначале выполняются преобразования, подобные выражению (1.15), после чего выполняется преобразование Хартли и лишь затем производится перестановка элементов вектора результатов.

К неудобствам косинусного преобразования следует отнести:

- неразделимость ядра преобразования для двумерного варианта;
- негибкие алгоритмы для различного размера ядра;
- несимметричный алгоритм обратного косинусного преобразования.

Заметим, что строго определяемое двумерное косинусное преобразование не обладает разделимым по координатам ядром, поэтому выполнение двумерного косинусного преобразования в целях сокращения объема вычислений может быть выполнено как модифицированное преобразование строчно-столбцовым методом, подобно тому, как поступают при выполнении двумерного преобразования Хартли [2].

В результате выполнения ДКП формируется 64 частотных компоненты фрагмента (или, что тоже, коэффициентов ДКП). В результате исходный фрагмент представлен в области пространственных частот. Этот шаг еще не приводит к сжатию изображения. Однако, при его выполнении полагается, что в подавляющем большинстве изображений близкие по своим координатам пиксели имеют и близкие значения. Поэтому, при переходе от фрагмента к его частотному представлению большая часть энергии сигнала сосредотачивается в области низких частот, т.е. компоненты с меньшим значением индекса k в выражении (1.11) имеют большие значения (см. рис. 1.5).

При выполнении этой операции 64 исходных пикселей преобразуются в матрицу из 64 коэффициентов, которые характеризуют "энергию" исходных пикселей. Важнейшей особенностью этой матрицы коэффициентов является то, что первый коэффициент передает подавляющую часть "энергии", а количество "энергии", передаваемой остальными коэффициентами, очень быстро убывает. То есть, большая часть информации исходной матрицы 8×8 пикселей представляется первым элементом матрицы, преобразованной по способу ДКП.

На этом этапе происходит некоторая потеря информации, связанная с принципиальной невозможностью точного обратного преобразования (на этапе восстановления изображения). Однако эта потеря информации весьма незначительна по сравнению с потерями на следующем этапе.

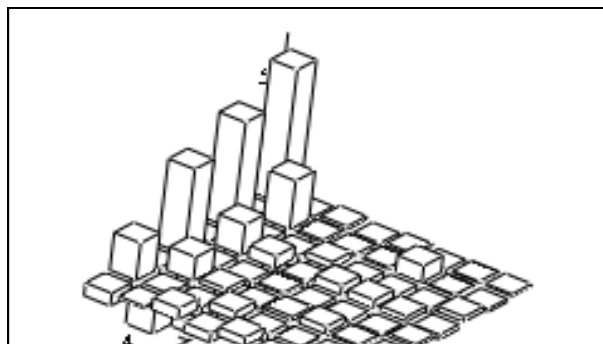


Рис.1.5. Спектр ДКП отдельного фрагмента изображения

Преобразованная матрица из 64 пикселей затем проходит операцию квантования, которая применяется для сокращения разрядности коэффициентов. Процесс квантования, который ведет к сжатию коэффициентов ДКП, выражается следующим образом [11]:

$$z_{kl} = \text{round}(y_{kl}/q_{kl}) = \lfloor (y_{kl} + \lfloor q_{kl}/2 \rfloor) / q_{kl} \rfloor, \quad k, l = 0, 1, \dots, 7, \quad (1.16)$$

где q_{kl} - весовой множитель матрицы квантования Q размера 8×8 с номером kl ($\lfloor x \rfloor$ обозначает наибольшее целое меньшее или равное x).

Иногда для формирования матрицы квантования может использоваться специальная весовая функция, позволяющая сформировать коэффициенты квантования, обращающие в 0 наибольшее число высоко и средне частотных коэффициентов. Согласно литературе [3,7], элементы матрицы Q линейно возрастают пропорционально сумме индексов элемента матрицы, например:

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

В результате квантования произошло обнуление многих коэффициентов y_{kl} . Выбор матрицы Q определяется требуемым коэффициентом сжатия.

Именно здесь происходит самая значительная потеря информации - отбрасываются малые изменения коэффициентов. Поэтому в процессе восстановления изображения после операции обратного ДКП получаются уже другие параметры пикселей. Квантование также обеспечивает возможность последующего эффективного сжатия данных при помощи любого способа сжатия без потерь.

Поскольку для U- и V- компонентов квантование может быть более грубым, чем для Y- компонента, на последнем этапе процесса сжатие U- и V-компонентов происходит в большей степени.

После квантования компоненты спектра всех обработанных фрагментов «вытягиваются» в последовательность чисел с помощью алгоритма диагонального сканирования. В основе такого сканирования лежит прием, позволяющий достичь большего уплотнения и основывающийся также на характерном виде спектра изображений реальных сцен.

Статистически доказано, что для реальных многоуровневых изображений двумерный квантованный спектр представляет собой матрицу треугольного вида. Большинство значений снизу и справа – нули. Построение элементов матрицы в цепочку производится так, как это показано на рис. 1.6. При этом в последовательность включаются только элементы от первого до последнего ненулевого. После него в последовательность включается специальный стоп – код. Это позволяет исключить из последовательности встречающиеся нули.

Далее обычно применяется метод однократного кодирования Хаффмана. Сначала анализируется вся последовательность символов. Часто повторяющимся сериям бит присваиваются короткие обозначения (маркеры). Различие размеров маркеров и представляемых ими битовых серий определяет достигаемую степень сжатия.

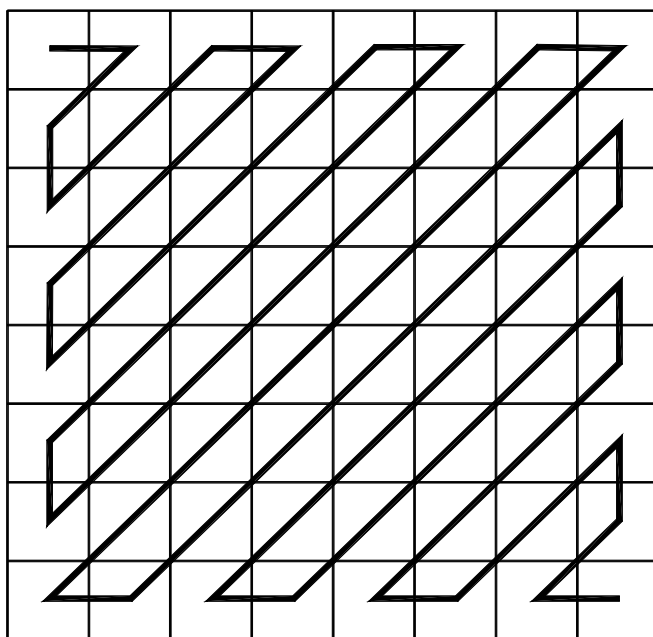


Рис.1.6. Диагональное «зиг-заг» сканирование спектральных компонент

Восстановление. При восстановлении изображений перечисленные выше шаги выполняются в обратном порядке. Декодирование начинается с восстановления

из полученного битового потока закодированных неравномерным кодом длин серий нулей и значащих элементов матрицы Z . Восстановление коэффициентов разложения \hat{Y} по квантованным значениям Z выполняется по формуле

$$\hat{y}_{kl} = z_{kl} q_{kl} \quad (1.16)$$

Далее выполняется обратное ДКП:

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

После этого цветовое пространство данных изображения можно преобразовать в исходный вид. Потери при обратном ДКП также не велики по сравнению с потерями квантования.

Коэффициент архивации в JPEG может изменяться в пределах от 2 до 200 раз (на практике коэффициент сжатия не превосходит 20-25) [11].

Как и любого другого алгоритма с потерями, у JPEG есть свои особенности. Наиболее известны "эффект Гиббса" и дробление изображения на квадраты 8×8 . Первый проявляется около резких границ предметов, образуя своеобразный "ореол" [4]. Разбиение на квадраты происходит, когда задается слишком большой процент архивации для данной конкретной картинки.

Недостатком метода 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 полосы изображения. По этой причине JPEG не рекомендуется использовать в полиграфии при высоких коэффициентах сжатия.

1.4.3. Сжатие по методу WIC (Wavelet Image Compression)

Под wavelet-преобразованием подразумевают множество базисных функций, которые формируют компактное описание видеосигнала. Анализ изображений, построенных на таких базисах, осуществляется по двум переменным - масштабу и сдвигу [1]. Это позволяет разделить крупные и мелкие детали изображений, одновременно локализуя их на временной шкале. Таким образом, учитывается важное для практики обстоятельство: протяженные объекты анализируемых данных лежат в низкочастотной области спектра, а короткие - в высокочастотной. Особенностью wavelet-преобразования является возможность сжатия и восстановления видеoinформации даже для слабоконтрастных изображений.

Несомненное преимущество применения этого преобразования - возможность представлять быстро изменяющиеся сигналы в компактной форме. Подобно широко используемому быстрому преобразованию Фурье (БПФ), wavelet-преобразование (DWT) обратимо и может служить инструментом анализа характеристик сигналов (спектральный анализ и др.). В отличие от БПФ (где базисными функциями служат синусы и косинусы) wavelet-преобразования формируют родительские функции более сложной формы. Их исходный набор обеспечивает получение бесконечного числа новых

форм функций. Поскольку wavelet-функции ограничены в пространстве, с их помощью можно локализовать пространственный объект с высокой степенью точности [1].

Непрерывное вейвлет-преобразование одномерного сигнала определяется следующим образом [1]:

$$W(a, x) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) w\left(\frac{t-x}{a}\right) dt. \quad (1.16)$$

Результат преобразования - функция, зависящая от двух переменных: от координаты - x и масштаба - a . Используемая здесь функция $w(x)$ является базисной. На каждом масштабе первоначальный базис w растягивается по горизонтали и сжимается по вертикали. Затем он сдвигается в точку x исследуемой функции и свертывается с ней. В литературе данный метод часто называют многомасштабным анализом [1].

Первоначальная функция $f(t)$ может быть восстановлена с помощью формулы:

$$f(t) = \frac{1}{C_\chi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} W(a, x) w\left(\frac{t-x}{a}\right) \frac{da dx}{a^2}, \quad (1.17)$$

где C_χ - норма базисной функции w

$$C_\chi = \int_0^{+\infty} \frac{\tilde{w}(v)^2}{v} dv. \quad (1.18)$$

Восстановление сигнала возможно, если норма определена.

На каждом шаге вейвлет-преобразования всплеск образует окно, вырезая из сигнала часть, попадающую в него. Не попавшие в окно значения сигнала будут слабо влиять на результат. Следовательно, появляется возможность локализовывать особенности сигнала.

Простейший всплеск – функция, на основе которой строится базис Хаара [1]:

$$f(x) = \begin{cases} 1 : x \in [-1, 0) \\ -1 : x \in [0, 1) \\ 0 : x \notin [-1, 1) \end{cases} \quad (1.19)$$

Непрерывный вейвлет-анализ позволяет получить большое количество информации о сигнале, но вместе с тем требует много вычислений. Действия, по вычислительным затратам эквивалентные преобразованию Фурье, выполняются для каждого элемента последовательности.

Рекурсивно волновое сжатие – wavelet сжатие – это сжатие с использованием всплесков. В отличие от Фурье анализа и ДКП всплески определены лишь на части области задания аргумента и могут рассматриваться как отличающиеся по масштабу и местоположению реплики единственной базовой функции, называемой порождающей или материнской функцией.

Самая простая идея алгоритма заключается в том, чтобы сохранять в файл разницу – число между средними значениями соседних блоков в изображении, которые обычно принимают значения близкие к 0.

Так два числа a_{2i} и a_{2i+1} можно представить в виде: $b_i^1 = (a_{2i} + a_{2i+1})/2$ и $b_i^2 = (a_{2i} - a_{2i+1})/2$, аналогично последовательность a_i может быть попарно переведена в последовательность $b_i^{1,2}$. Данное преобразование можно применять последовательно несколько раз к результатам предыдущего преобразования.

Для двух мерного случая (квадрата из 4-х точек с яркостями $a_{2i,2j}$, $a_{2i,2j+1}$, $a_{2i+1,2j}$, $a_{2i+1,2j+1}$) наипростейшим преобразованием может быть преобразование следующего вида:

$$\begin{aligned} b_{i,i}^1 &= (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1})/4 \\ b_{i,i}^2 &= (a_{2i,2j} + a_{2i+1,2j} - a_{2i,2j+1} - a_{2i+1,2j+1})/4 \\ b_{i,i}^3 &= (a_{2i,2j} - a_{2i+1,2j} + a_{2i,2j+1} - a_{2i+1,2j+1})/4 \\ b_{i,i}^4 &= (a_{2i,2j} - a_{2i+1,2j} - a_{2i,2j+1} + a_{2i+1,2j+1})/4 \end{aligned} \quad (1.20)$$

Таким образом, используя данное преобразование, исходное изображение преобразуется в 4 субизображения, к каждому из которых, в принципе, данное преобразование можно применить повторно.

Итак, основная идея вейвлетного преобразования сигнала состоит в иерархическом разложении входного сигнала на последовательности так называемых базовых компонент с последовательно уменьшающимся разрежением и связанных с ними компонент деталей. На каждом уровне разложения базовая компонента и компонента деталей содержат информацию, необходимую для восстановления базового сигнала на следующем уровне более с высоким разрешением.

При выполнении двумерного вейвлет-анализа с использованием преобразования Хаара (выражение 1.20) для разложения изображения вначале выполняется разложение по строкам, а затем по столбцам. Результатом разложения являются 4 матрицы $HH0$, $HL0$, $LH0$, $LL0$, соответствующие фильтрации фильтром $h_1(n)$ по строкам и столбцам, фильтром $h_1(n)$ по строкам и фильтром $h_0(n)$ по столбцам, фильтром $h_0(n)$ по строкам и фильтром $h_1(n)$ по столбцам, и фильтром $h_0(n)$ по строкам и столбцам. Далее низкочастотная матрица $LL0$ подвергается вейвлет разложению. Его результатом являются матрицы $HH1$, $HL1$, $LH1$, $LL1$. Такое разложение повторяется r раз, как показано на рис.1.8. Результатом разложения является набор из $3r+1$ матриц уменьшающейся размерности. Каждая матрица подвергается скалярному или векторному квантованию и последующему кодированию. Выбор числа уровней квантования или шага квантования производится исходя из требуемого сжатия и соответствующего распределения битов между матрицами.

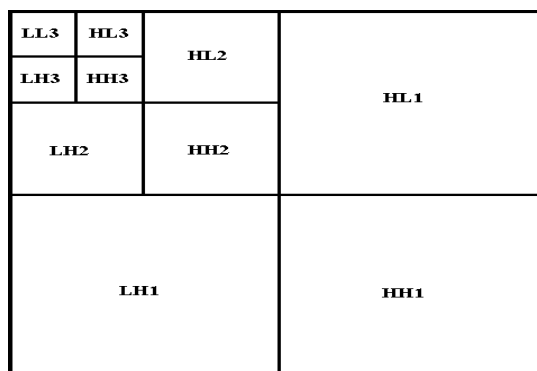


Рис. 1.8. Трёхуровневое вейвлет преобразования изображения

Высокочастотные матрицы (матрицы деталей), как правило, после квантования содержат большое число нулевых элементов. Одним из широко используемых методов кодирования матриц деталей является кодирование длин серий нулей и следующих за ними значений кодом Хаффмана. Низкочастотные матрицы непосредственно кодируются кодом Хаффмана.

При дискретном вейвлет-преобразовании (ДВП) анализируется сигнал с помощью блока анализа, направляющего входной сигнал x на два фильтра с импульсными откликами h (низкочастотный) и g (высокочастотный) соответственно. После прохождения фильтров сигналы прореживаются вдвое, т.е. сигнал, содержащий N элементов, преобразуется в $N/2$ – элементные выходы двух каналов. Блок анализа осуществляет один уровень прямого вейвлет преобразования.

Блок синтеза выполняет обратное преобразование. В оба канала этого блока после каждого входного отсчета вставляется нулевой отсчет.

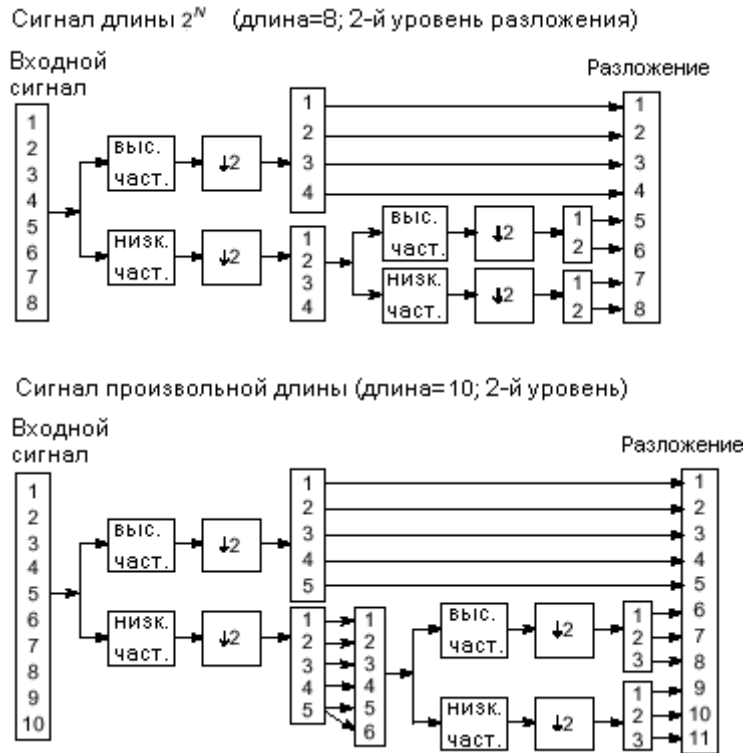


Рис.1.9. Схема ДВП разложения.

Удлиненные таким образом вдвое сигналы поступают на фильтры с импульсными откликами u и v , а затем складываются. Два канала, содержащие по $N/2$ элементов каждый, объединяется в один сигнал, состоящий из N элементов. Блок синтеза осуществляет один уровень обратного вейвлет преобразования. Низкочастотный фильтр задает скейлинг-функцию (масштабирующую функцию), а высокочастотный фильтр определяет всплеск.

Результатом M -кратного пропуска коэффициентов фильтра u/v через верхний канал блока синтеза будет вейвлет или скейлинг-функция на масштабном уровне M . Схема такого процесса представлена на рис.1.9.

Дискретное вейвлет-преобразование позволяет разложить и потом вновь собрать сигнал, но для практических приложений интерес представляют модификации вейвлет разложения, а не тождественные преобразования [19,24]. Например, для сжатия важно представить сигнал небольшим числом коэффициентов разложения [60-63].

Алгоритм быстрого пирамидального иерархического разложения сигнала в векторы данных половинной длины позволяет получить множество коэффициентов разложения (для разрешения $1/2$, $1/4$, $1/8$ и т. д.) [4]. Степень сжатия при wavelet-преобразовании зависит от необходимого разрешения.

Сегодня у специалистов не вызывает сомнения, что эти алгоритмы кодирования обладают рядом преимуществ по сравнению с алгоритмами, построенными на основе дискретного косинусного преобразования Фурье, которое используется в кодеках JPEG, MPEG-1, MPEG-2 [3]. В частности,

подобное разложение применяется в новых стандартах сжатия JPEG2000 и WIC [4,7,10].

1.4.4. Фрактальное сжатие изображений

Во многом недостатки в качестве восстановленного изображения по сжатому формату согласно JPEG связано с тем, что при сжатии никак не учитываются специфика изображения, т.е. не выявляется его структура, характерные участки и т.д. Именно учет специфики изображения лежит в основе фрактального метода, который предложил в 1988 году Мандельброт. Согласно Мандельброту, фрактал - это структура, выделенная при анализе изображения, и обладающая схожей формой независимо от ее размеров. Например, в изображении кроны дерева, фрактал - изображение листа. Поэтому изображение можно как бы собирать из фракталов, т.е. изображение в терминах фрактального подхода есть суперпозиция фракталов. В свою очередь, отдельный фрактал может быть описан некоторым стандартным образом.

Основу фрактального подхода составляет постулат о том, что изображения реального мира имеют аффинную избыточность [4]. Иначе говоря, существует набор аффинных коэффициентов, описывающих вращение, сжатие, расширение, искажение формы, сдвиг объектов изображения.

В начале 80-х годов Майкл Барнсли выдвинул идею получения заранее заданного изображения как аттрактора хаотического процесса. Барнсли пытался ответить на вопрос: возможно ли для данного изображения построить хаотическую систему, которая будет являться для него странным аттрактором. Он использовал систему итерируемых функций (Iterated Function System - IFS).

Наиболее распространённым примером фрактального изображения, сгенерированного с помощью IFS является изображение папоротника (рис.1.10), использованное для создания данного изображения, состоит из 4-х аффинных преобразований. Каждое преобразование кодируется считанными байтами, хотя исходное изображение может быть любого размера. Таким образом, можно заключить, что фрактальная компрессия – это поиск самоподобных областей и определение для них параметров аффинных преобразований.

Простая IFS, породившая изображения папоротника не пригодна для сжатия произвольных изображений, поскольку они обычно представляются в градациях серого, а не из значений 0 или 1 и т.к. простые IFS используются только для самоподобных изображений, т.е. когда изображения строятся из элементов, являющихся копией целого изображения.



Рис. 1.10. Изображение папоротника, сгенерированного с помощью IFS

В основе фрактального сжатия лежат несколько основных определений и теорем. Рассмотрим их вкратце.

Преобразование. Преобразование сопоставляет точке в одном пространстве точку в другом пространстве, возможно в том же самом пространстве. Преобразование называется отображением f и записывается как:

$f: X_1 \rightarrow X_2$, если оно переводит пространство X_1 в пространство X_2 .

Преобразование $f: X_1 \rightarrow X_2$ в метрическом пространстве (X, d) называется сжимающим, если существует константа s , $0 \leq s < 1$ такая что:

$$d(f(x_1), f(x_2)) \leq s \cdot d(x_1, x_2),$$

где $d(x_1, x_2)$ – расстояние от точки x_1 до точки x_2 в пространстве X .

Константа s называется коэффициентом сжатия отображения f .

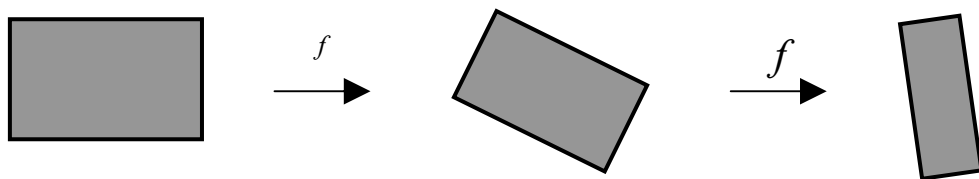


Рис. 1.11. Сжимающее отображение для точек в пространстве R^2

На рисунке 1.11 показан пример сжимающего отображения (R^2, d_2) , примененного к множеству точек в R^2 . Здесь данное преобразование применялось более одного раза: сначала $f(x)$ вычислялось для точки x , а затем преобразование f применялось к результату преобразования: $f(f(x))$. Такое последовательное, многократное применение преобразования называют итерациями и обозначают как: f^{on} , т.е. $f(f(\dots f(x)\dots))$, где f применяется n раз.

Теорема о сжимающих отображениях: пусть $f: X_1 \rightarrow X_2$ сжимающее отображение на полном метрическом пространстве. Тогда f имеет всего одну и только одну неподвижную точку $x_f \in X$ и для любого x из X последовательность $\{f^n(x): n=1, 2, \dots\}$ сходится к x_f . Эта теорема лежит в основе всех подходов к фрактальному сжатию изображений.

Пусть $\{w_1, w_2, \dots, w_n\}$ конечный набор сжимающих отображений в пространстве (X, d) с коэффициентами сжатия s_1, s_2, \dots, s_n , $0 \leq s_i < 1$. Определим отображение W , действующее на компактные множества точек B из X :

$$W(B) = w_1(B) \cup w_2(B) \cup \dots \cup w_n(B) = \bigcup_{n=1}^N w_n(B) \quad (1.21)$$

Таким образом, W осуществляет отображение в данном пространстве с коэффициентов сжатия s , где $s = \max\{s_1, s_2, \dots, s_n\}$.

Система итерирующих функций (IFS) состоит из полного метрического пространства (X, d) и конечного множества сжимающих отображений $w_n: X \rightarrow X$ с коэффициентами сжатия s_n . Таким образом, IFS можно обозначить следующим образом: $\{X, w_n: n=1, 2, \dots, N\}$, или если рассматривать пространство точек в R^2 , то просто $\{w_n\}$.

В теории фрактального сжатия доказывается теорема коллажа, которую можно сформулировать следующим образом.

Пусть есть двоичное изображение $L \subset R^2$ и пусть есть сжимающие отображения, такие что: $\bigcup_{n=1}^N w_n(L)$

Покрывают L почти точно. Можно считать такое $w_n(L)$ уменьшенной копией L . Тогда теорема коллажа утверждает, что аттрактор A (аттрактором IFS называется изображение, которое является единственной неподвижной точкой IFS) системы $\{w_n\}$ близок к L . «Коллажом» является набор областей $w_n(L)$.

Так как аттрактор A – это результат бесконечного числа итераций IFS, то он является по своей сути фракталом.

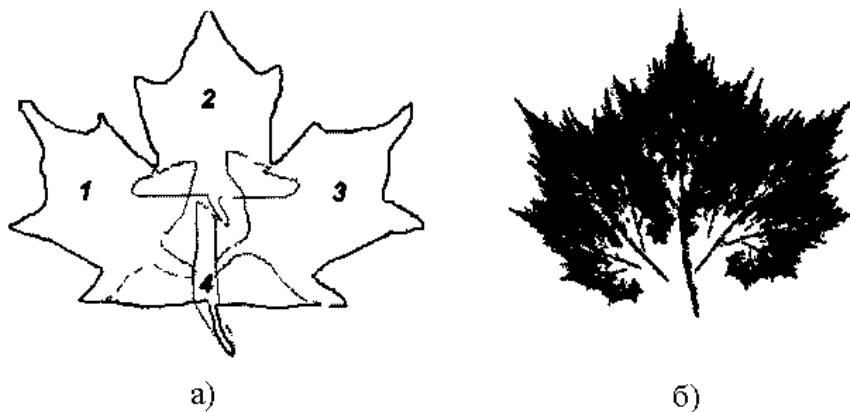


Рис. 1.12. Иллюстрация теоремы коллажа.

- (а) исходное изображение и четыре фрагмента изображения;
(б) изображение-аттрактор

Чтобы на практике применить теорему коллажа, необходимо выбрать

преобразования, которые будут являться сжимающими отображениями. Одним из таких преобразований являются аффинные преобразования: $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$:

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}, \quad (1.23)$$

где $a, b, c, d, e, f \in \mathbb{R}$. Аффинные преобразования могут осуществлять поворот, перемещение и масштабирование.

На рисунке 1.13 показано действие аффинного преобразования на множество точек в \mathbb{R}^2 .

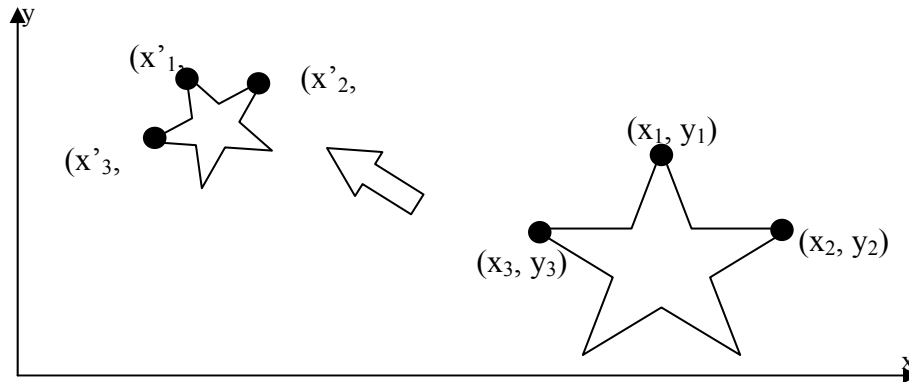


Рис. 1.13. Воздействие аффинного преобразования на точки в пространстве \mathbb{R}^2

Чтобы создать IFS изображение – нужно найти хотя бы некоторое приблизительное самоподобие в изображении. Затем нужно выделить точки и преобразование IFS для каждой из фигур подобия на изображении. Когда точки и преобразования для IFS определены, то вычисляются коэффициенты аффинного преобразования, используя систему уравнений, основанную на выражении (1.23).

Существует два алгоритма построения изображения-аттрактора с помощью IFS. Один из них это прямое применение теоремы о сжимающих отображениях, а другой – применение так называемой «игры хаоса».

Детерминированный алгоритм для построения изображения являющегося аттрактором IFS, к любому начальному изображению B применяет теорему о сжимающих отображениях. Алгоритм строит последовательность изображений A_n , многократно применяя IFS отображение $W = \{w_1, w_2, \dots, w_n\}$:

$$A_n = W^{on}(B) \quad (1.24)$$

Детерминированный алгоритм полезен с точки зрения обучения, поскольку позволяет видеть результат преобразования на каждом шаге итерации (рис 1.14).

Вероятностный алгоритм связывает с каждым аффинным преобразованием w_i в IFS вероятность p_i . Эти вероятности определяют, насколько плотно каждая часть изображения-аттрактора покрыта точками. Вероятностный алгоритм создаёт изображение - аттракторы высокого качества быстрее, чем детерминированный алгоритм. Это происходит не только из-за

того, что вероятностный алгоритм на каждом шаге итерации выполняет меньшую работу, но и сама выполненная работа даёт лучший результат.

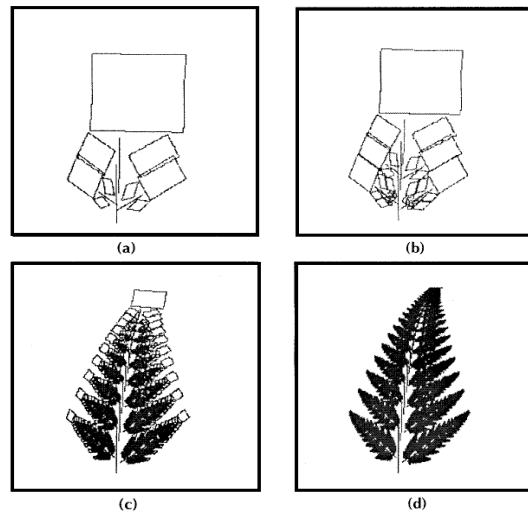


Рис. 1.14. Детерминированный алгоритм, применённый к IFS папоротника.
Вид изображения A_n после: а) 2-х, б) 3-х, с) 10-и, d) 30-и итераций.

Реальные изображения не обладают свойством глобального самоподобия, которое присуще изображениям IFS. К тому же реальные изображения представлены в градациях серого, а не в двоичных значениях, как изображения IFS. Поэтому при фрактальном кодировании реальных изображений пытаются найти множество сжимающих преобразований, которые отображают доменные блоки (которые могут перекрываться) в множество ранговых блоков, которые покрывают изображения. Ранговые блоки могут быть одинакового размера, но чаще используется адаптивное разбиение с переменным размером блока (одним из распространённых методов адаптивного разбиения является метод квадродерева).

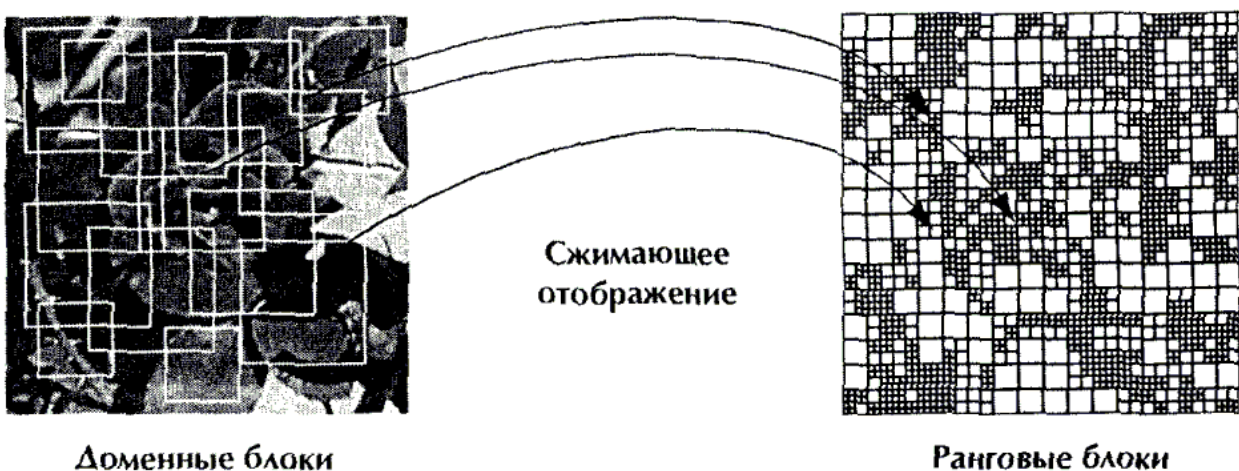


Рис. 1.15. Отображение доменных блоков в ранговые
при фрактальном сжатии изображения
Базовый алгоритм фрактального кодирования изображения выполняется

следующим образом (Рис 1.15):

1. Изображение f разбивается на не перекрывающиеся ранговые блоки $\{R_i\}$. В самом простом случае блоки могут представлять собой прямоугольники, но могут быть и другие формы.

2. Изображение покрывается последовательностью доменных блоков, которые могут пересекаться. Домены могут быть разного размера, и их количество может исчисляться сотнями и тысячами.

3. Для каждого рангового блока находят домен и соответствующее преобразование, которое наилучшим образом покрывает ранговый блок. Обычно это аффинное преобразование.

4. Если достаточно точного соответствия не получилось, то разбиваем ранговые блоки на меньшие ранговые блоки. Данный процесс продолжается до тех пор пока, не получают приемлемого соответствия или размер рангового блока достигает определённых значений.

Блок-схема такого алгоритма представлена на рис.1.16.

В настоящее время известны следующие разновидности такого алгоритма [7].

Алгоритм Фишера. Идея алгоритма заключается в том, чтобы некоторым образом классифицировать D-блоки и R-блоки, а поиск близкого D-блока производить в том же классе, к которому относится ранговая область. Делается это следующим образом [7].

Исходные блоки разбиваются на четыре части. Для каждой из частей подсчитывается среднее значение A_i и дисперсия V_i пикселей. Далее блоки классифицируются по следующему принципу. Определим три базовых типа блоков

тип 1: $A_1 \geq A_2 \geq A_3 \leq A_4$,

тип 2: $A_1 \geq A_2 \geq A_4 \geq A_3$,

тип 3: $A_1 \geq A_4 \geq A_2 \geq A_3$.

Понятно, что любой блок при помощи соответствующего аффинного преобразования квадрата в квадрат можно привести к виду, соответствующему одному из указанных типов. После того, как мы зафиксировали три основных класса, блоки классифицируются по дисперсии. Таким образом, в каждом из трех классов появляются 24 подкласса, итого 72 класса. Поиск близкого к R-блоку D-блока производится перебором в соответствующем классе.

Генетический алгоритм. Генетический алгоритм (ГА) представляет собой алгоритмический подход к решению экстремальных задач однокритериального выбора, основанный на моделировании основных факторов эволюционного развития популяции [7].

При использовании ГА для поиска оптимальных решений каждый элемент $x \in X$ пространства оптимизации должен быть представлен как вектор $b \in B$ из N символов двоичного алфавита $A = \{0,1\}$, где $B = A^N$. Необходимо также, чтобы пространство оптимизации X состояло из конечного числа элементов.

Популяцией $\Pi = (\chi^1, \chi^2, \dots, \chi^M)$ численности M считается вектор пространства V^M , координаты которого называются генотипами особей данной популяции.

Шагом ГА является переход от текущего поколения к следующему, т.е. получение новой популяции Π_{t+1} из Π_t . В построении очередной особи новой популяции участвуют операторы кроссинговера (скрещивания), мутации и случайный оператор отбора, $Select: V^M \rightarrow \{1, \dots, M\}$ действие которого состоит в выборе номера особи родителя при порождении очередного потомка.

Для определения ГА необходимо задать оператор кроссинговера $Cross: V \times V \rightarrow V \times V$ и оператор мутации $Mut: V \rightarrow V$.

Действие кроссинговера $(\chi', \tau') = Cross(\chi, \tau)$ заключается в выборе случайным образом некоторой позиции j , равномерно распределенной от 1 до $N-1$, после чего результат формируется в виде

$$\chi' = (\chi_1, \chi_2, \dots, \chi_j, \tau_{j+1}, \dots, \tau_N), \quad \tau' = (\tau_1, \tau_2, \dots, \tau_j, \chi_{j+1}, \dots, \chi_N).$$

Влияние кроссинговера регулируют с помощью вероятности P_{Cross} срабатывания этого оператора (в противном случае все остается без изменений).

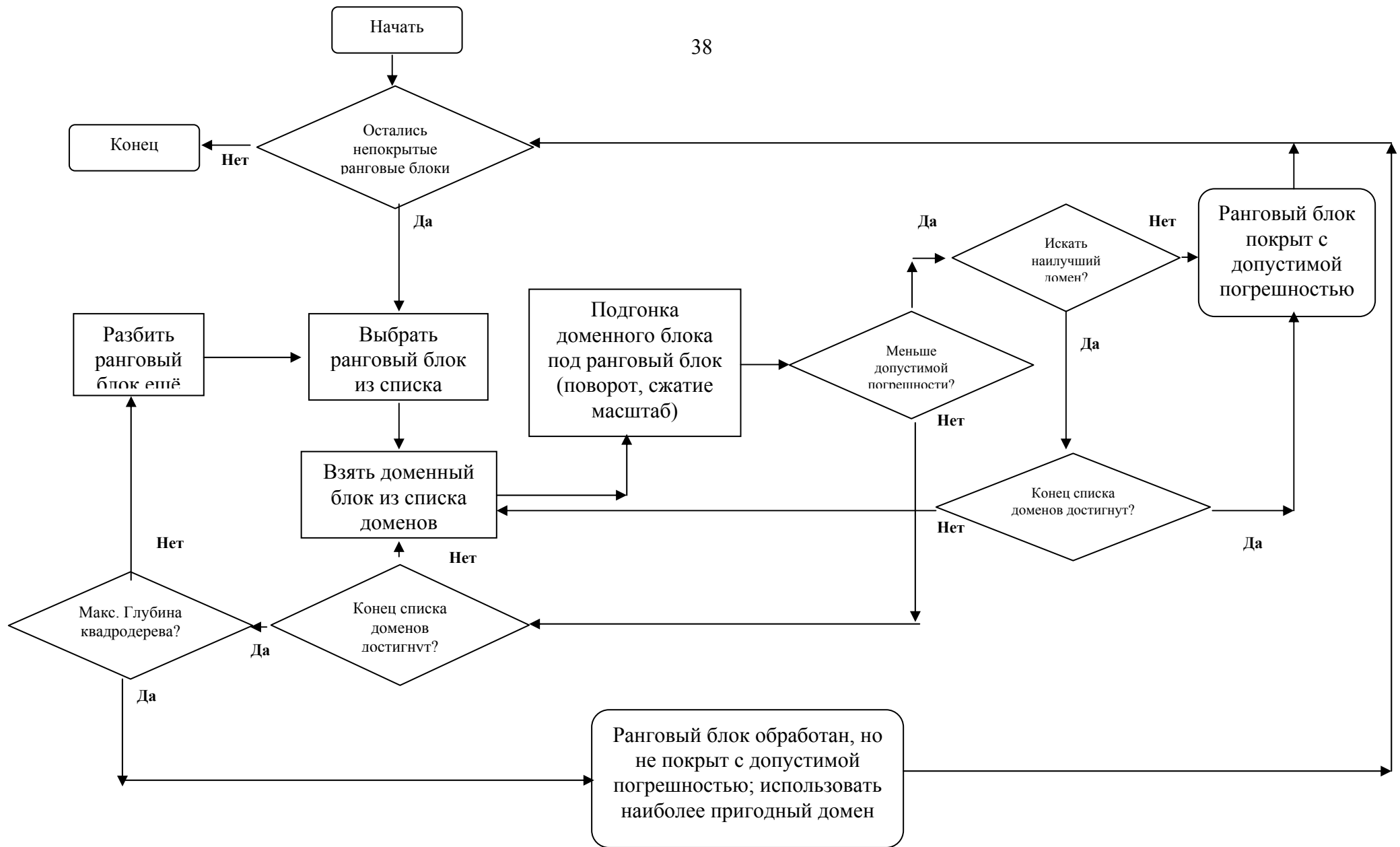


Рис. 1.16. Блок схема основных шагов фрактального кодирования изображения

Оператор мутации в каждой позиции аргумента с заданной вероятностью P_{mut} заменяет ее содержимое на случайный элемент двоичного алфавита A , выбранный в соответствии с равномерным распределением (в противном случае все остается без изменений).

Целевая функция исходной задачи, заменяется в ГА на неотрицательную функцию пригодности генотипа $\Phi(\chi)$, где $\chi \in B$.

Процесс работы алгоритма представляет собой последовательную смену поколений, на каждом шаге которой популяция Π_{t+1} наполняется парами потомков от особей популяции Π_t по формуле

$$(\chi_k^{t+1}, \chi_{k+1}^{t+1}) = Mut(Cross(\chi_{Select(\Pi_t)}^t, \chi_{Select(\Pi_t)}^t)),$$

где $(\chi_k^{t+1}, \chi_{k+1}^{t+1})$ - особи с наименьшей пригодностью популяции Π_t . То есть индивиды извлекаются попарно из Π_t и после кроссинговера и мутации помещаются в Π_{t+1} . Изменение вероятностей мутации и кроссинговера позволяет регулировать работу ГА и настраивать его на конкретные задачи.

Модифицированный генетический алгоритм. Опишем схему ГА в применении к задаче фрактального сжатия [7]. В качестве генотипа ГА удобно взять вектор, компонентами которого будут пиксельные координаты области $D_{j(i)}$ исходного изображения, определенного на тороидальной поверхности, и число кодирующее аффинное преобразование W_i . Имеется восемь способов аффинного преобразования квадрата в квадрат: поворот на четыре стороны или зеркальное отражение и поворот на четыре стороны. Следовательно, на кодировку этого преобразования достаточно трех бит. Функцию пригодности положим равной

$$\Phi = \frac{1}{1 + \sum ([f(\xi, \eta) - F_i(\xi, \eta)]^2 : (\xi, \eta) \in R_i \cap Z^2)},$$

где в нижней части под знаком суммы – евклидово расстояние между исходным и преобразованным блоком. Данная функция удовлетворяет требованиям ГА (неотрицательна) и адекватна для оператора рулеточной селекции, при которой каждый индивид $\chi^{i,t}$ популяции Π_t оказывается родителем при формировании очередной особи $\chi^{i,t+1}$ популяции Π_{t+1} с вероятностью

$$P_{select}(\chi^{i,t}) = \frac{\Phi(\chi^{i,t})}{\sum_j \Phi(\chi^{j,t})}.$$

При таком представлении хромосом, определяющих данный генотип, любой вектор пространства решений всегда допустим и имеет ненулевую пригодность.

Оператор мутации для данного алгоритма – стандартный, а оператор кроссинговера был модифицирован следующим образом:

- позиция кроссинговера может располагаться только в местах стыковки двоичного представления координат.

- выполняется только для близкородственных особей, т.е., если расстояние $\rho(\chi, \tau)$ мало, где ρ - некоторая метрика.

Увеличение быстродействия алгоритма. Недостатком данного метода сжатия является очень большой объем вычислений для кодирования изображений. Для программ реализующих данный метод, применяются как обычные методы увеличения скорости программ: целочисленная арифметика, ассемблерная реализация и другие, так и некоторые специфические приемы: использование аффинных преобразований только одного типа, различные варианты разбиения изображения на области R_i .

Приведем схемы двух алгоритмов, которые для некоторых классов изображений могут значительно уменьшить объем вычислений. Параметрами первого алгоритма служат уровень потерь при кодировании и минимальный размер областей R_i . Этот алгоритм обеспечивает равномерное качество кодирования всего изображения. Параметром второго алгоритма является количество областей R_i , используемых для кодирования изображения, что прямо влияет на объем вычислений, но он не обеспечивает достаточной точности кодирования отдельных фрагментов изображения [7].

Алгоритм 1.

- Выберем допустимый уровень потерь при кодировании ϵ .
- $R_1 = \Omega$ и пометим его как необработанный фрагмент.
- Пока есть необработанный фрагмент R_i выполнять:

1. Найти $D_{j(i)}$, W_i и F_i , которые наилучшим образом приближают R_i (на которых достигается минимум $\sigma_{R_i}^2(W_i, D_{j(i)}, F_i)$).

2. Если $\sigma_{R_i}^2(W_i, D_{j(i)}, F_i) < \epsilon$ или размер $R_i < \min$, то пометить R_i как обработанное. Иначе, разбить R_i на более мелкие фрагменты и пометить их как необработанные.

Алгоритм 2.

- Выберем максимальное число N фрагментов R_i .
- Добавим фрагмент $R_1 = \Omega$ в список преобразований и пометим его как необработанный.
- Пока есть необработанные фрагменты в списке выполнять:

1. Для каждого необработанного фрагмента найти соответствующие $D_{j(i)}$, W_i и F_i .

2. Найти в списке фрагмент R_i наибольшего размера и наибольшей метрикой $\sigma_{R_i}^2(W_i, D_{j(i)}, F_i)$.

3. Если число фрагментов в списке меньше N , тогда разбить фрагмент R_i на более мелкие и занести их в список как необработанные. Вычеркнуть из списка фрагмент R_i .

Алгоритм восстановления изображения:

1. Создается два изображения одинакового размера А и Б. Размер изображений может быть не равен размеру исходного изображения. Начальный рисунок областей А и Б не имеет значения. Это могут быть случайные данные.

2. Данные из области А преобразуются в область Б. Для этого сначала изображение Б делится на домены также как и на первой стадии процесса сжатия (расположение доменов описано в заголовке файла). Теперь для каждого домена области Б проводится соответствующее аффинное преобразование ранговых областей изображения А, описанное коэффициентами из сжатого файла. Результат помещается в область Б. На этой стадии создается совершенно новое изображение.

3. Данные из области Б преобразуются в область А. Этот шаг идентичен предыдущему, только изображения А и Б поменялись местами, т.е. теперь область А делится на блоки и ранговые области изображения Б отображаются на эти блоки.

4. Процедуры 2 и 3 повторяются до тех пор, пока изображения А и Б не станут неразличимыми.

Прямой и обратный ход (сжатие и восстановление) не эквивалентны по затратам. Прямое преобразование (сжатие) - значительно дольше, обратное преобразование (восстановление) - гораздо быстрее. Алгоритм фрактального сжатия - несимметричный алгоритм. Коэффициент симметричности (отношение времени архивации ко времени разархивации) колеблется в пределах 1000-10000.

К достоинствам фрактального метода можно отнести:

- Высокие коэффициенты сжатия.
- Высокую скорость обратного преобразования.
- Возможность дальнейшего структурного анализа изображения [4].

При этом фрактальный метод обладает следующими недостатками:

- Зависимостью результатов работы метода от принципов отбора базовых элементов и доменов.
- Коэффициент сжатия зависит от повторяемости базовых элементов.

Алгоритм ориентирован на полноцветные изображения и изображения в градациях серого цвета. Фрактальное сжатие реализовано в формате GIF.

В настоящее время существует достаточно много алгоритмов сжатия изображений. Основой любых методов сжатия данных является использование естественной избыточности исходной информации. Сжатие изображений осуществляется либо в пространственной либо в частотной областях изображения. Наиболее яркими примерами пространственного сжатия изображений являются алгоритмы PCX, GIF, а частотного сжатия - JPEG.

В пространственной области нельзя добиться достаточно большого коэффициента сжатия, поэтому в настоящее время большинство алгоритмов сжатия работают в частотной области изображения, поскольку наиболее явно выраженная избыточность исходных данных проявляется при их спектральном анализе, как результата разложения в некотором ортогональном базисе.

В подобных алгоритмах можно выделить три основных шага:

1. Применение обратимых дискретных ортогональных преобразований к изображению.
2. Выбор наиболее значимых частотных коэффициентов.
3. Вторичное сжатие выбранных коэффициентов, например арифметическим или статистическим алгоритмом сжатия.

2. АЛГОРИТМЫ СЖАТИЯ АУДИОСИГНАЛОВ

2.1. Параметры звукового сигнала

Звук – это упругое колебание среды, он распространяется в среде с помощью волн давления посредством колебания атомов и молекул. Как и любая волна, звук характеризуется скоростью, амплитудой и частотой [5].

Воздействуя на слух, звук вызывает раздражение, которое создает у человека субъективный эффект – ощущение.

Интенсивность звука I определяется как среднее количество звуковой энергии, проходящей в единицу времени через единицу поверхности:

$$I = \frac{P^2}{Q_0}, \text{ Вт/м}^2, \quad (2.1)$$

где Q_0 – удельное сопротивление воздуха, $Q_0 = 1,23 \text{ кг/м}^3$.

При исследовании восприятия изменения интенсивности звука было установлено, что одинаковые относительные изменения интенсивности вызывают одинаковые абсолютные изменения слухового ощущения, то есть слуховое ощущение E пропорционально логарифму раздражающей силы:

$$E = a \lg I + C. \quad (2.2)$$

На пороге слышимости $I = I_{\text{пс}}$ слуховое ощущение равно нулю, поэтому

$$C = -a \lg I_{\text{пс}}, \quad (2.3)$$

откуда

$$E = a \lg(I/I_{\text{пс}}). \quad (2.4)$$

Для оценки величины слухового ощущения была предложена единица под название "бел" ($a=1$). Эта единица соответствует десятикратному отношению интенсивностей, поэтому была введена более мелкая единица – "децибел" ($a=10$).

$$E = 10 \lg(I/I_{\text{пс}}), \text{ дБ} \quad (2.5)$$

Вследствие логарифмического закона восприятия звука и широкого диапазона слышимости звуков было введено понятие уровня интенсивности

$$N(I) = 10 \lg(I/I_0), \text{ дБ}. \quad (2.6)$$

За нулевой уровень условились принимать интенсивность I_0 близкую к порогу слышимости для нормального слуха на частоте 1 кГц ($I_0 = 10^{-12} \text{ Вт/м}^2$). Нулевой уровень по звуковому давлению P_0 равен $2,04 \cdot 10^{-5} \text{ Па}$. Интенсивность и звуковое давление связаны квадратичной зависимостью, поэтому уровень звукового давления N определяется равенством:

$$N = 20 \lg(P/P_0). \quad (2.7)$$

Порог слышимости – это минимальное звуковое давление, при котором еще существует слуховое ощущение. На рис. 2.1 показана зависимость порога слышимости от частоты, которая была выявлена опытным путем.

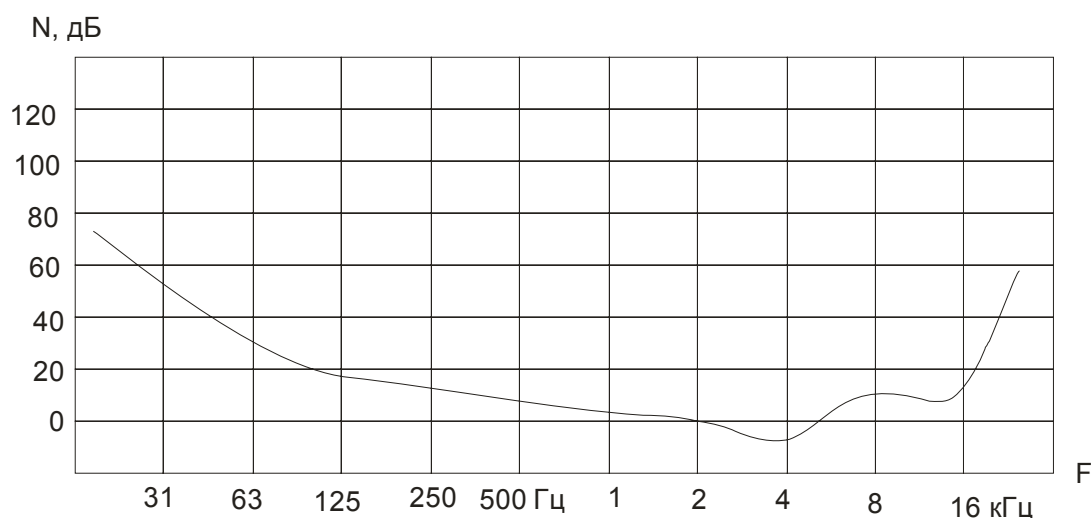


Рис. 2.1. Абсолютный порог слышимости

2.1.1. Критические полосы слуха

При восприятии звука слуховой аппарат человека разделяет его на частотные группы, называемые критическими полосами. Эта особенность является одним из важнейших свойств слуха. Стандартами MPEG установлены нижние границы критических полос слуха: 20, 100, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4440, 5300, 6400, 7700, 9500, 12000, 16000 Гц.

В области до 500 Гц ширина частотных групп не зависит от средней частоты шума и равна примерно 100 Гц. В области выше 500 Гц она увеличивается пропорционально средней частоте (рис. 2.2).

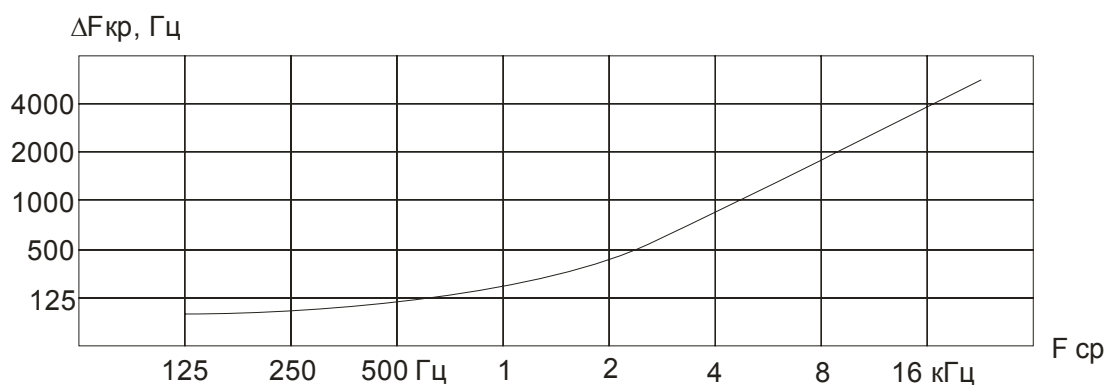


Рис. 2.2. Зависимость ширины критической полосы слуха от ее средней частоты

Критические полосы слуха оказалось удобно использовать в качестве единицы субъективной высоты тона, которую называли барк. В настоящее время в психоакустических моделях слуха вместо частотных шкал используются

шкалы высоты тона в барках. На рис. 2.3 приведен график перевода частотной шкалы высоты тона в октавах в шкалу высоты тона в барках.

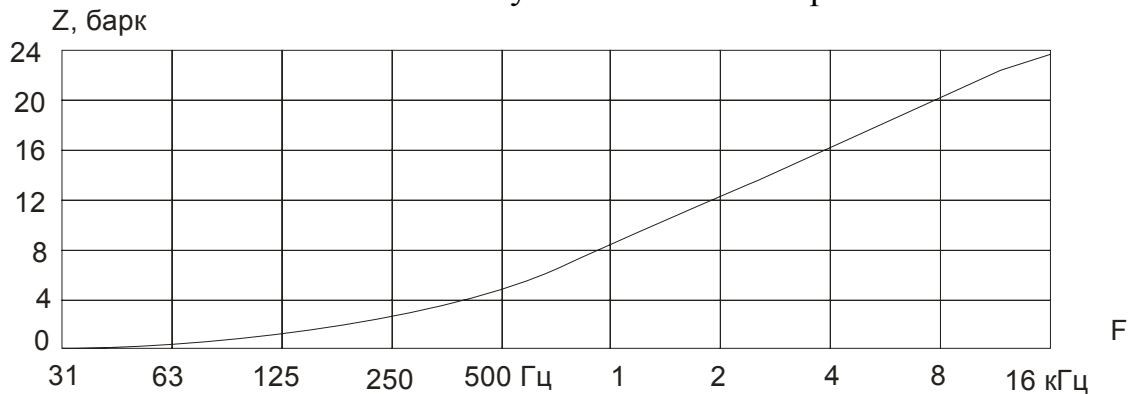


Рис. 2.3. График перехода от частотной шкалы высоты тона в октавах к шкале высоты тона в барках

2.1.2. Пороги слышимости при маскировке

Повышение порога слышимости одного тона (или сигнала) при одновременном воздействии другого тона (шума или сигнала) называют маскировкой. Различают частотное и временное маскирование.

На рис. 2.4 показаны кривые маскировки тона, когда мешающим звуком является тоже тон. По этим кривым определяют порог слышимости на конкретных частотах маскируемого тона, и рассчитывается коэффициент маскировки K_M , дБ

$$K_M = N_{ПС} - N_{АПС}. \quad (2.8)$$

Маскировка максимальна, когда частоты F_T и F_M близки, величина маскировки увеличивается по мере возрастания уровня мешающего тона N_M .

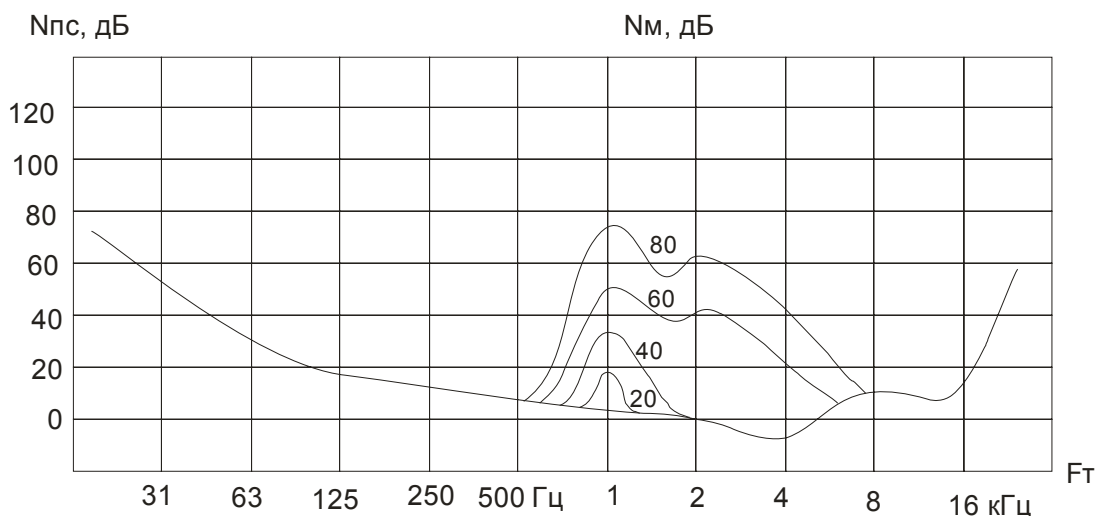


Рис. 2.4. Кривые порога слышимости тона с частотой F_T при маскировке тоном с частотой $F_M = 1$ кГц и уровнем N_M , дБ

На рис. 2.5 показаны кривые маскировки тона узкополосным шумом. Они примерно такие же, как при маскировке тоном с некоторыми отличиями: нет биений и провала коэффициента маскировки. Узкополосный шум маскирует не только тоны, но и широкополосные шумы квантования.

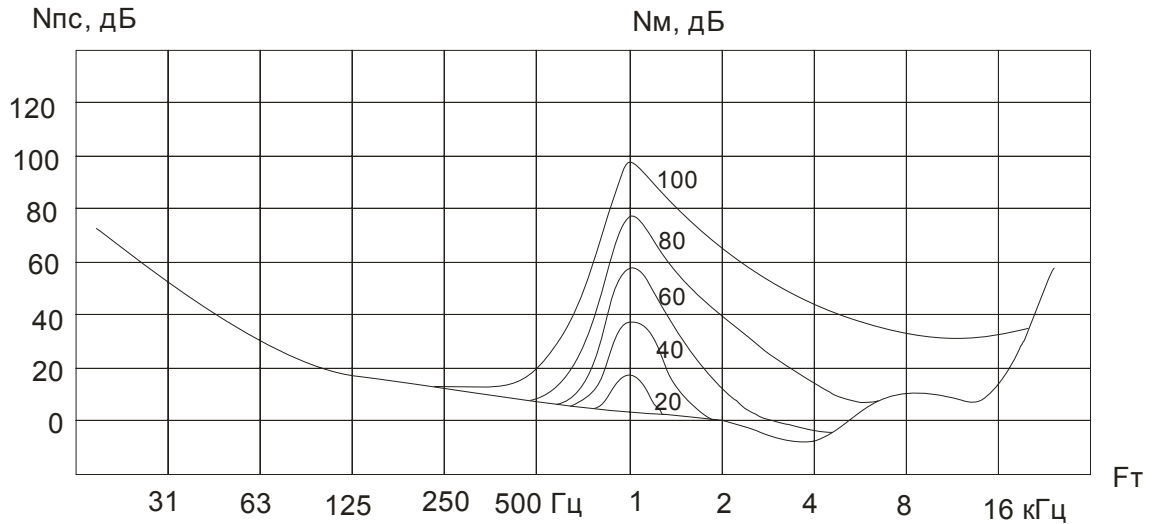


Рис. 2.5. Кривые порога слышимости тона с частотой F_T при маскировке узкополосным шумом с центральной частотой $F_M = 1$ кГц и уровнем N_M , дБ

На рис. 2.6 показан порог слышимости во временной области, при воздействии сигнала частотой f с уровнем интенсивности 60 дБ.

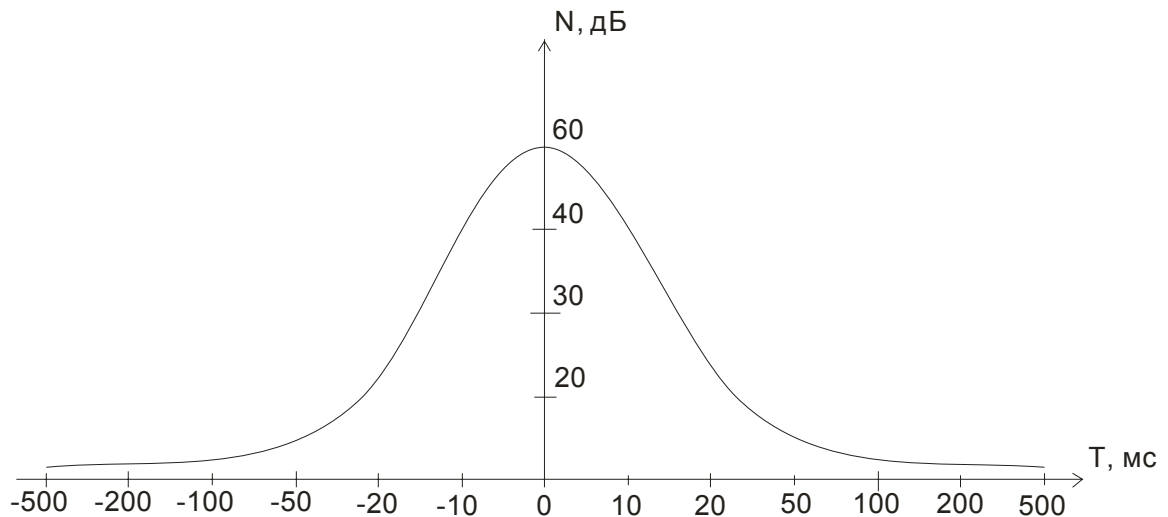


Рис. 2.6. Порог слышимости при временном маскировании

2.1.3. Цифровой звук

С записью звука в цифровом виде связано два действия: дискретизация и

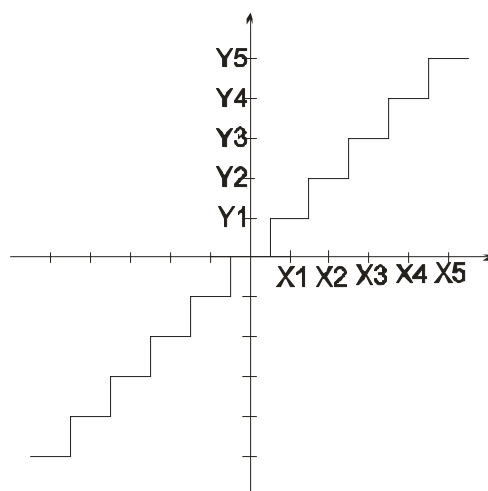
квантование. Дискретизация звука заключается в периодическом измерении аналогового сигнала и использовании полученных моментальных значений вместо исходной волны. Под квантованием понимается процесс получения предельно точных моментальных значений аналогового сигнала и последующего их округления.

Дискретизация. В соответствии с теоремой отсчетов В. А. Котельникова неискаженная передача непрерывного (аналогового) сигнала с полосой частот $0..F_{\max}$ дискретной последовательностью его отсчетов возможна, если частота дискретизации f_d связана с максимальной частотой F_{\max} исходного сигнала соотношением:

$$f_d > 2F_{\max}. \quad (2.9)$$

Если требуется передать синусоидальный сигнал с частотой 20 кГц, то требуемая частота его дискретизации должна быть более 40 кГц. Все сигналы, частота которых будет больше половины частоты дискретизации ($F > f_d/2$), при восстановлении будут интерпретированы как сигналы более низкой частоты.

Квантование. Квантование – это преобразование аналогового сигнала в ступенчатый сигнал с двоичным отсчетом уровней в квантах. При этой операции производится округление входного сигнала к принятой двоичной шкале квантователя. Процедуру квантования можно рассматривать как прохождение входного сигнала через устройство с амплитудной характеристикой ступенчатой формы (рис. 2.7), которая называется характеристикой (или шкалой) квантования. Если в пределах всей шкалы шаг квантования постоянен ($x_i - x_{i-1} = \Delta$ и $y_i - y_{i-1} = \Delta$), то квантование называют равномерным. Такой вид квантования удобен для начального представления звукового сигнала с целью последующей обработки¹.



¹ При использовании равномерного квантования получается сигнал в импульсно-кодовой модуляции (ИКМ)

Рис. 2.7. Характеристика квантования

Максимальное число уровней N_{KB} для каждой полярности звукового сигнала определяется числом разрядов m :

$$N_{KB} = 2^{(m-1)} - 1. \quad (2.10)$$

Чем больше уровней квантования, тем на большее число ступеней разбивается шкала квантования и тем с большей точностью производится аналого-цифровое преобразование. Но квантование сигналов неизбежно сопровождается погрешностью (или шумом квантования).

2.2. Основные виды алгоритмов сжатия аудиосигналов

К настоящему времени разработано множество алгоритмов кодирования общего назначения, которые выискивают повторяющиеся последовательности в двоичных данных. Если им удастся обнаружить такие последовательности, то они могут сжать данные, заменив последовательности кодами. Модуль, производящий декомпрессию (которому доступна информация о таких последовательностях) может произвести обратную операцию.

Эти алгоритмы относятся к алгоритмам сжатия без потерь, так как восстановленные данные будут с точностью до бита идентичны исходным. Они могут быть использованы для компрессии любых данных, но при сжатии аудио сигналов оказываются мало эффективными [7].

Например, в методе Лемпеля-Зива-Уэлча (Lempel-Ziv-Welch, LZW), дефляции и методе Берроуза-Уиллера (Burroughs-Wheeler) алгоритм заключается в поиске длинных последовательностей байтов, встречающихся в файле несколько раз. Однако, в звуковых файлах длинных повторяющихся последовательностей обычно не много: в основном это обуславливается наличием шумов.

В алгоритме Хаффмана (Huffman) и арифметическом кодировании ищутся байты с определенными значениями (или пары байтов), попадающие в файл чаще остальных. Как только удастся выделить такое значение, строится код, который тем короче, чем чаще встречается значение. Действительно, у звуковых файлов неравномерно распределены возможные значения байтов, поэтому алгоритмы данного типа могут достаточно хорошо обрабатывать звуковые файлы. Но так как звуковые файлы могут обладать очень большим объемом, эти методы также становятся неэффективными. Как правило, методы сжатия без потерь используются в современных стандартах сжатия на вторичном, дополнительном этапе сжатия с целью повышения коэффициента сжатия.

Еще одним недостатком алгоритмов сжатия без потерь является неоднородность компрессии, что не позволяет наложить ограничения на скорость передачи сжатых аудио потоков.

Учет основных требований к качеству речевого сегмента и временной

задержке, как правило, требует компромисса [5].

С увеличением сложности алгоритма для уменьшения требуемой полосы пропускания увеличивается время обработки в устройствах кодирования/декодирования.

Хотя качество сигнала в основном напрямую связано со скоростью цифрового потока, сложные алгоритмы кодирования способны достичь более высоких отношений качества и скорости.

Сжатие речевого сегмента (РС) может быть как без потерь (архивация), так и с потерями. Причем в последнем случае это кодирование можно подразделить на три вида [2,3]:

1. Кодирование непосредственно реализации РС (Wave Form Codec);
2. Измерение, кодирование и передача на приемную сторону параметров РС, по которым уже на приемной стороне производится синтез этого (искусственного) РС. Такие системы называют вокодерными (Source Codec);
3. Гибридные способы кодирования, т.е. сочетание первого и второго способов кодирования.

Под кодированием подразумевается преобразование РС в некоторый «другой» сигнал, который можно представить с меньшим числом разрядов, что в итоге повысит скорость передачи данных. Одним из видов такого кодирования является дифференциальная импульсно-кодовая модуляция (ДИКМ), о которой и пойдет речь в дальнейшем.

Кодеры формы волны аппроксимируют изменение сигнала во времени. Они требуют наибольших скоростей передачи, но имеют наилучшие показатели качества воспроизведенной речи.

При параметрическом компандировании моделируется процесс речеобразования человека. В кодере из речевого сигнала вычисляются определенные параметры, передаваемые к декодеру, в котором они применяются для восстановления формы исходного сигнала. Использование исключительно параметрических методов приводит к потере натуральности звучания голоса и большой чувствительности к фоновым шумам. Вокодерные преобразования отличаются наименьшими требованиями к полосе пропускания.

Один из способов повышения эффективности использования полосы пропускания состоит в применении *гибридных методов*, основанных на принципах линейного предсказания и объединяющих параметрическое компандирование и кодирование формы волны. Большинство гибридных кодеров используют замкнутое кодирование (метод «анализ через синтез») на передающей стороне, что позволяет подкорректировать определенные параметры посредством сравнения результата синтеза с оригиналом. Это, безусловно, увеличивает время обработки, но обеспечивает лучшие показатели при передаче.

На рис.2.8 - 2.10 отображены основные операции для различных методов кодирования.

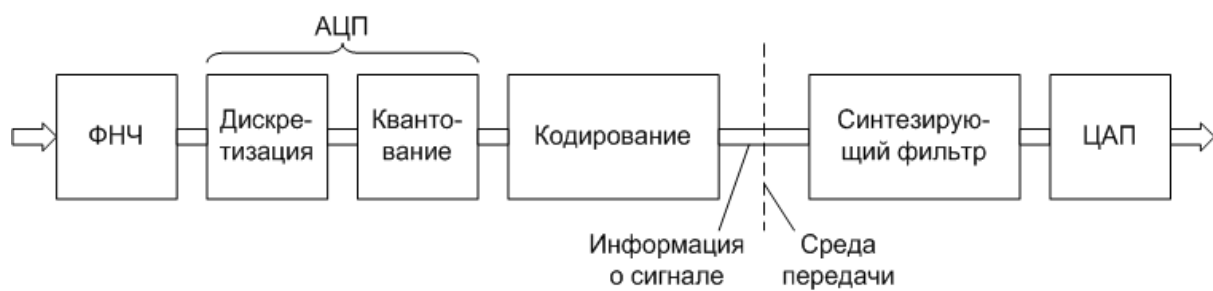


Рис.2.8. Основные операции кодирования формы волны

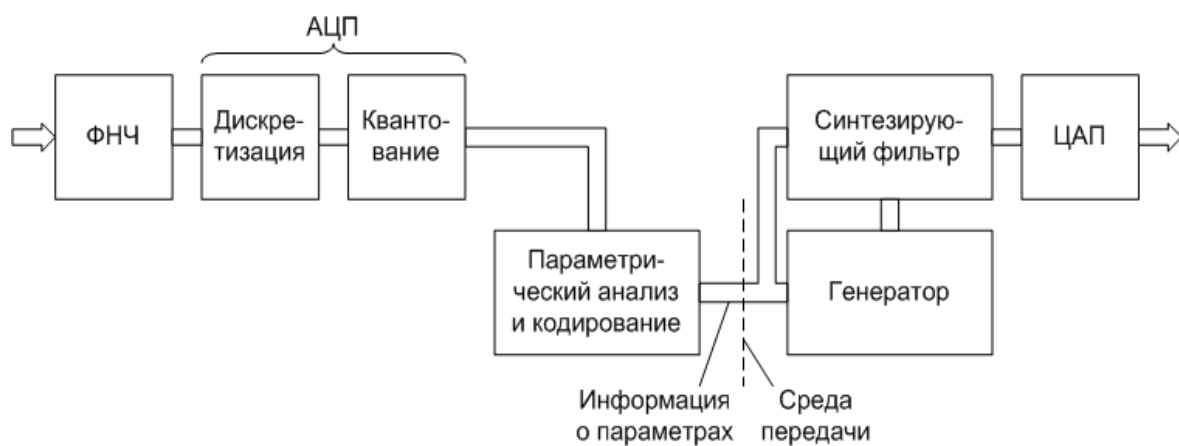


Рис.2.9. Основные операции параметрического компрессирования

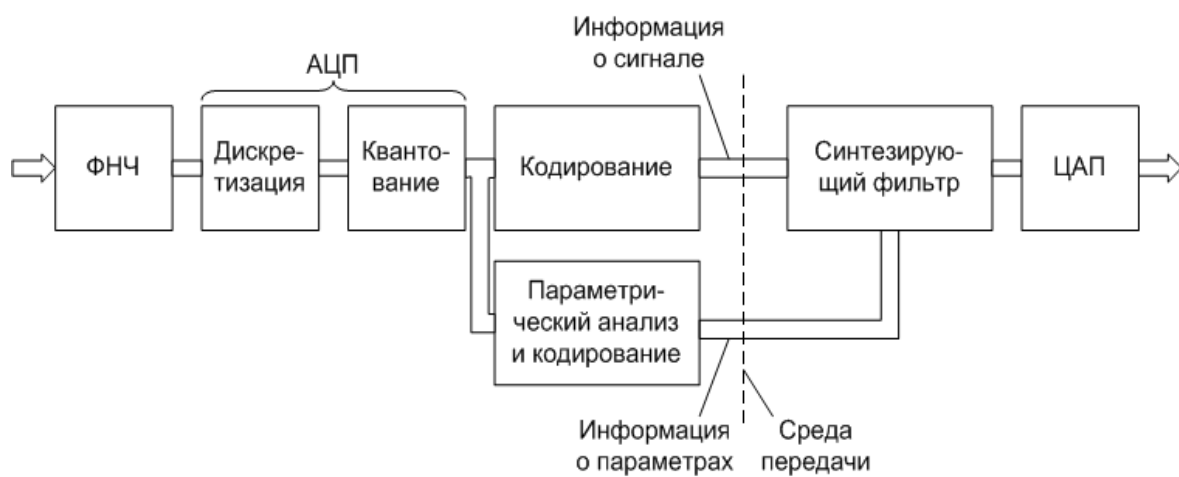


Рис.2.10. Основные операции кодирования гибридными методами

В табл.2.1 представлены наиболее распространенные алгоритмы и области их применения [5]. Алгоритмы указаны в порядке убывания битовой скорости потока.

Как правило, определяющими для выбора метода кодирования являются такие показатели, как:

- качество голоса по пятибалльной шкале экспертных оценок MOS (Mean Opinion Score, Рекомендация ITU-T P.800);
- задержка алгоритма;
- помехоустойчивость;
- степень ухудшения качества сигнала при квантовании QDU (Quantization Distortion Units);
- распространенность, поддержка производителями оборудования и др.

Таблица 2.1

Алгоритм	Скорость, кбит/ с	Стандарт	Год	Приложение
Аппроксимация формы речевой волны				
PCM ²	64, 56, 48	ITU-T G.711	1960	Общественные телефоны
SB-ADPCM ³	64, 56, 48	ITU-T G.722	1986	Передача широкополосных сигналов
ADPCM ⁴	32	ITU-T G.721	1984	Общественные телефоны

² PCM (Pulse Code Modulation) – импульсно-кодовая модуляция.

³ SB-ADPCM (Sub-Band Adaptive Differential Pulse Code Modulation) – адаптивная дифференциальная импульсно-кодовая модуляция с делением на поддиапазоны.

⁴ ADPCM (Adaptive Differential Pulse Code Modulation) – адаптивная дифференциальная импульсно-кодовая модуляция.

ADPCM	40, 32, 24, 16	ITU-T G.726	1984	Общественные и цифровые беспроводные телефоны
Гибридные методы кодирования				
LD-CELP ⁵	16	ITU-T G.728	1992	Общественные телефоны, видеотелефоны
RPE-LTP- LPC ⁶	13	ETSI GSM 06.10	1992	Европейские цифровые сотовые системы
CS-ACELP ⁷	11.8, 8, 6.4	ITU-T G.729, G.729 Annex	1997	Передача речи в сетях Frame Relay, ATM, в системах телесвязи Франции
MP-MLQ ⁸	6.3	ITU-T G.723.1	1996	Передача речи в видеотелефонии
VSELP ⁹	5.6	ETSI GSM 06.20		Европейские цифровые сотовые системы
ACELP ¹⁰	5.3	ITU-T G.723	1996	Передача речи в видеотелефонии
Вокодерные преобразования				
LPC-1011	2.4	ANSI		Специальные системы

В табл.2.2 приведены данные по соответствию качества речи, MOS, задержек передачи и типов каналов, удовлетворяющих предъявленным требованиям [5].

Таблица 2.2.2

Качество	Лучшее	Хорошее	Среднее	Плохое	Стандарт
MOS	> 4.5	4–4.5	3.5–4	3–3.5	ITU-T P.800, P.830
Задержка, мс	< 150	< 250	< 350	< 450	ETSI TS 101 329
	< 150	< 260	< 400	> 400	ITU-T G.114
Тип канала	ТфОП	Спутниковый	ТфОП + спутниковый	Допустимо для VoIP	

⁵ LD-CELP (Low Delay Code Excited Linear Prediction) – линейное предсказание с кодовым возбуждением и малой задержкой.

⁶ RPE-LTP-LPC (Regular Pulse Excitation Long Time Prediction Linear Predictive Coding) – кодирование на основе линейного предсказания с долговременным предсказанием с регулярным импульсным возбуждением.

⁷ CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction) – линейное предсказание сопряженной структуры с алгебраическим возбуждением.

⁸ MP-MLQ (Multi Pulse Maximum Likelihood Quantization) – метод квантования по максимуму правдоподобия.

⁹ VSELP (Vector Sum Excited Linear Prediction) – линейное предсказание с векторным возбуждением.

¹⁰ ACELP (Algebraic Code Excited Linear Prediction) – линейное предсказание с алгебраическим возбуждением.

¹¹ LPC (Linear Predictive Coding) – кодирование на основе линейного предсказания.

На рис.2.11 изображены сглаженные зависимости оценок MOS от требований к битовой скорости потока, построенные по усредненным результатам исследований ITU Study Group 15.

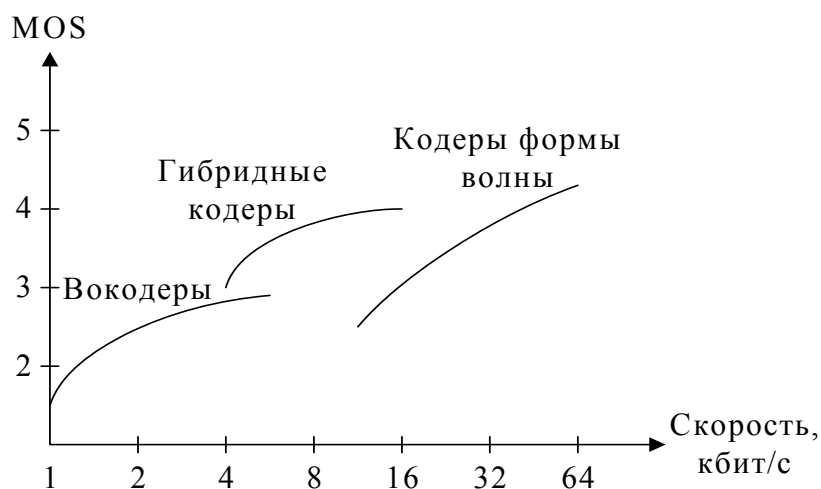


Рис.2.11. Зависимость оценок MOS от скорости потока для кодеров формы волны, вокодеров и гибридных кодеров

Следует отметить, что значения MOS можно встретить во многих информационных источниках, при этом отклонения составляют не более 0.4 балла, что допустимо, поскольку «хорошая» связь или «плохая» – это субъективная оценка, зависящая от ожиданий абонентов, их капиталовложений и других факторов.

Указанные в табл.1.2 значения задержек следует считать ориентировочными, можно встретить и другие.

На количественный показатель задержки оказывают воздействие [2]:

- алгоритмы кодирования/декодирования информации;
- сеть;
- операционная система;
- буфер устранения джиттера.

Следует отметить, что только среда передачи в среднем задерживает сигнал на 10–150 мс в зависимости от длины и типа каналов связи.

Как правило, более сложные алгоритмы кодирования обеспечивают лучшее сжатие при практически неизменном качестве речи. Становится очевидным, что для уменьшения битовой скорости, а, следовательно, и составляющих сетевой задержки, неизбежно увеличение задержки алгоритмов кодирования.

Безусловно, временная задержка кодирования зависит от быстродействия устройства, выполняющего преобразование. Поэтому представленные ниже графики, по мнению автора работы, правильнее рассматривать с точки зрения относительных значений, а не абсолютных.

На рис.2.12 изображены сглаженные зависимости общих задержек алгоритмов от битовой скорости потока. Численные значения задержек взяты из описаний рекомендаций ITU-T, ETSI и др., размещенных на сайтах: www.axenet.ru, www.vocal.com.

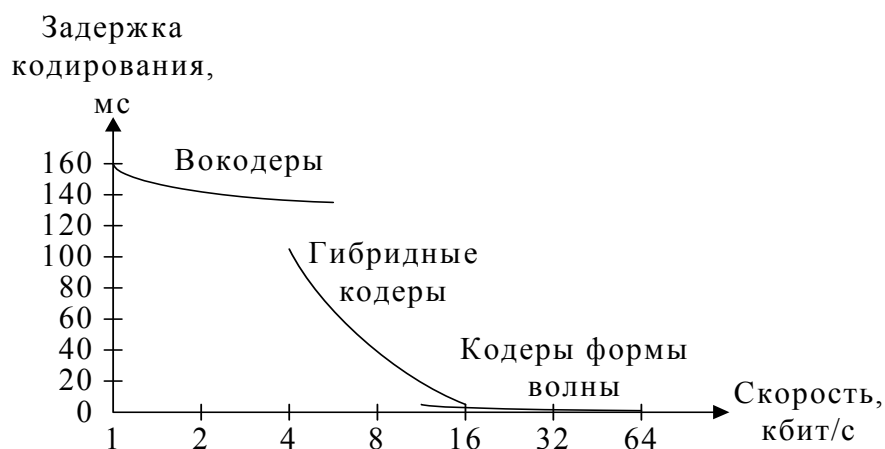


Рис.2.12. Зависимость задержки кодирования от скорости потока для кодеров формы волны, вокодеров и гибридных кодеров

Задержки декодирования могут существенно изменяться в зависимости от организации буфера устранения джиттера. В области телекоммуникаций джиттером называется первая производная задержки прохождения данных по времени.

2.3. Степень ухудшения качества сигнала при квантовании

Один QDU соответствует ухудшению качества при оцифровке с использованием стандартной процедуры PCM. Согласно рекомендациям ITU-T, для международных вызовов величина QDU не должна превышать 14. Следует отметить, что передача разговора по международным магистральным каналам ухудшает качество речи, как правило, на 4 QDU. При передаче разговора по национальным сетям должно теряться не более 5 QDU. Значения QDU для некоторых алгоритмов: ADPCM (32кбит/с), LD-CELP (16кбит/с) и CS-ACELP (8кбит/с) – 3.5, ADPCM (24кбит/с) – 7. Следовательно, для качественной передачи речи процедуру компрессии/декомпрессии желательно применять в сети только один раз. В некоторых странах это является обязательным требованием регулирующих органов, предъявляемым к сетям, подключенным к ТфОП.

В результате обзора услуг провайдеров IP-телефонии можно сделать вывод, что в настоящее время наибольшую популярность приобрели алгоритмы MP-MLQ и CS-ACELP, выполненные по стандартам ITU-T G.723.1 и G.729 Annex A, соответственно.

Анализ последних разработок показал, что в первую очередь учитываются скорость алгоритма и оценка качества речи, причем именно в указанной последовательности.

Минимизация скорости привела к появлению методов, основанных на интерполяции спектрально-временных алгоритмов параметрического компандирования [5,7].

Но большинство разработок ведется в области гибридных методов. В последних разработках кодеров применяются:

- алгоритмы долговременного и кратковременного предсказания;
- кодовые книги, хранящие различные виды сигналов возбуждения;
- подавление пауз, которые обычно занимают до 60% длительности разговора;
- переменная скорость кодирования, учитывающая:
 - разделение сегментов речевого сигнала на основе фонетической или энергетической классификации;
 - возможность применения различных систем кодирования на разных сегментах;
- настройка на говорящего абонента.

В составе кодеров можно выделить по крайней мере двух общих для большинства алгоритмов функциональных блоков:

- блок предсказания;
- блок спектрально-временного преобразования.

Последний блок представляет наибольший интерес, поскольку используется для решения различных задач. Спектральное представление сигнала позволяет:

- уменьшить объем передаваемых данных;
- осуществить частотную фильтрацию – селекцию желаемой полосы частот в обрабатываемом сигнале:
 - подавлять шумы обнулением компонент на нежелательных частотах;
 - выделить достаточную полосу частот для воспроизведения разборчивой речи и особенностей (тембра) говорящего, т.е. содержащую три первых формантных частоты (как правило, используется полоса частот от 300 до 3400Гц) [5];
 - выделять один период спектра для восстановления непрерывного сигнала;
- выполнить классификацию сегмента сигнала (вокализованный, невокализованный глухой, невокализованный фриктивный звук, шум) [5] и др.

Следует отметить, что в некоторых случаях операцию фильтрации целесообразнее выполнять путем свертки исходных данных с фильтрующей последовательностью, т.е. без проведения анализа в частотной области.

2.4. Алгоритмы сжатия аудиосигналов

2.4.1. Дифференциальная импульсно-кодовая модуляция

В обычной импульсно-кодовой модуляции каждый отсчет кодируется независимо от других. Однако у многих источников сигнала при стробировании с частотой Найквиста или быстрее проявляется значительная корреляция между последовательными отсчетами [5] (в частности, источник РС является квазистационарным источником и может относиться к рассматриваемым видам источников). Другими словами, изменение амплитуды между последовательными отсчетами в среднем относительно малы. Следовательно, схема кодирования, которая учитывает избыточность отсчетов, будет требовать более низкой битовой скорости.

При дифференциальной импульсно-кодовой модуляции (ДИКМ) по каналу связи передаются не значения отсчетов сообщения, а разность между отсчетами. Один из распространённых способов эффективной передачи непрерывных сообщений - способ передачи с предсказанием.

Последовательность коррелированных отсчётов исходного сигнала подаются на один из входов вычитающего устройства, на другой вход поступает сигнал предсказания, сформированный из предыдущих отсчётов. Полученный сигнал ошибки предсказания поступает в тракт передачи. Поскольку в сигнале ошибки как раз и содержатся новые сведения, представляющие разность между истинным и предсказанным значением, то такой способ передачи называется передачей с предсказанием [5].

Известно большое число вариантов реализации кодирования с предсказанием. Основное различие сводится к различию операций формирования сигнала ошибки: в одних системах сигнал ошибки формируется в аналоговой форме, а затем кодируется, в других - вначале кодируется исходный аналоговый сигнал, а затем формируется сигнал ошибки.

В системах с ДИКМ обычно применяют неравномерное квантование сигнала ошибки, так как наиболее вероятны малые ошибки. Поскольку мощность шума квантования составляет определённую долю мощности квантуемого процесса, а мощность ошибки предсказания существенно меньше мощности сообщения, шум квантования при ДИКМ меньше, чем при обычной ИКМ при том же числе уровней. ДИКМ обеспечивает одинаковое с ИКМ качество передачи при меньшем числе символов в кодовой комбинации. Корреляция между отсчётами возрастает по мере сокращения интервала между ними. Поэтому в пределе при большой частоте дискретизации число уровней квантования сигнала ошибки можно уменьшить до двух и перейти к одноразрядным системам.

Такой способ кодирования называют дельта-модуляцией (ДМ). Дельта модуляция (ДМ), отличается от ДИКМ тем, что передается только знак сигнала ошибки.

Предположим, что x_n означает текущий отсчет источника, и пусть \hat{x}_n

обозначает предсказанное значение (оценку) для x_n , определяемое как

$$\hat{x}_n = \sum_{k=1}^M a_k \cdot x_{n-k} \quad (2.11)$$

Таким образом, \hat{x}_n является взвешенной линейной комбинацией M отсчетов, а $\{a_k\}$ являются коэффициентами предсказания. Величины $\{a_k\}$ выбираются так, чтобы минимизировать некоторую функцию ошибки между x_n и \hat{x}_n .

Величина ошибки определяется как:

$$x_n - \hat{x}_n = e_n \quad (2.12)$$

Виды линейных предсказаний. Рассмотрим виды предсказания. «Линейное» предсказание означает, что \hat{x}_n является линейной функцией предыдущих отсчетов; при «нелинейном» предсказании – это нелинейная функция. Порядок предсказания определяется количеством используемых предыдущих отсчетов. То есть, предсказание нулевого и первого порядка является линейным, а второго и более высокого порядка – нелинейным. При линейном предсказании восстановить сигнал значительно проще, чем при нелинейном предсказании. Будем рассматривать только линейное предсказание.

Предсказание нулевого порядка. В этом случае для предсказания текущего отсчета используется только предыдущий отсчет РС, т.е.

$$\hat{x}_n = x_{n-1} \Rightarrow e_n = x_n - \hat{x}_n = x_n - x_{n-1} \quad (2.13)$$

Предсказание первого порядка (линейная экстраполяция). В этом случае для предсказания текущего отсчета используется не только предыдущий отсчет, но и разница между предпоследним и последним отсчетами, которая суммируется к общему результату:

$$\begin{aligned} \hat{x}_n &= x_{n-1} + \Delta x = x_{n-1} + (x_{n-1} - x_{n-2}) = 2 \cdot x_{n-1} - x_{n-2} \\ e_n &= x_n - \hat{x}_n = x_n - 2 \cdot x_{n-1} + x_{n-2} \end{aligned} \quad (2.14)$$

Реализация ДИКМ. Определив метод определения коэффициентов предсказания, рассмотрим блок-схему практической системы ДИКМ, показанную ниже (рис. 2.13).

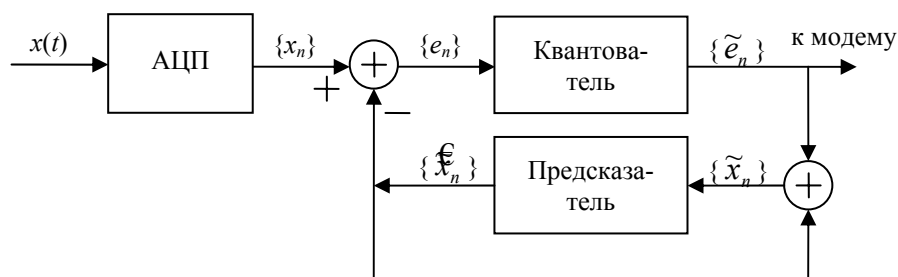


Рис 2.13. Блок-схема кодера ДИКМ

В этой схеме предсказатель стоит в цепи обратной связи, охватывающей квантователь. Вход предсказателя обозначен \tilde{x}_n . Он представляет собой

сигнальный отсчет x_n , искаженный в результате квантования сигнала ошибки. Выход предсказателя равен:

$$\mathbb{E}_n = \sum_{k=1}^M a_k \cdot \tilde{x}_{n-k}; \quad (2.15)$$

Разность $e_n = x_n - \mathbb{E}_n$ является входом квантователя, а \tilde{e}_n обозначает его выход. Величина квантованной ошибки предсказания \tilde{e}_n кодируется последовательностью двоичных символов и передается через канал в пункт приема. Квантованная ошибка \tilde{e}_n также суммируется с предсказанной величиной \mathbb{E}_n , чтобы получить \tilde{x}_n .

В месте приема используется такой же предсказатель, как на передаче, а его выход \mathbb{E}_n суммируется с \tilde{e}_n , чтобы получить \tilde{x}_n (см. рис. 2.14).

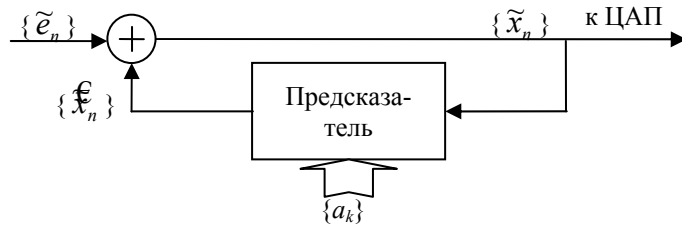


Рис.2.14. Блок-схема декодера ДИКМ

Сигнал \tilde{x}_n является входным воздействием для предсказателя и в то же время образует входную последовательность, по которой с помощью ЦАП восстанавливается сигнал $x(t)$. Использование обратной связи вокруг квантователя обеспечивает то, что ошибка в \tilde{x}_n - просто ошибка квантования $q_n = \tilde{e}_n - e_n$ и что здесь нет накопления предыдущих ошибок квантования при декодировании. Имеем

$$q_n = \tilde{e}_n - e_n = \tilde{e}_n - (x_n - \mathbb{E}_n) = \tilde{x}_n - x_n \quad (2.16)$$

Следовательно, $\tilde{x}_n = x_n + q_n$. Это означает, что квантованный отсчет \tilde{x}_n отличается от входа x_n ошибкой квантования q_n независимо от использования предсказателя. Значит, ошибки квантования не накапливаются.

В рассмотренной выше системе ДИКМ оценка или предсказанная величина \mathbb{E}_n отсчета сигнала x_n получается посредством линейной комбинации предыдущих значений \tilde{x}_{n-k} , $k = 1, 2, \dots, M$, как показано в формуле (2.15). Улучшение качества оценки можно получить включением в оценку линейно отфильтрованных последних значений квантованной ошибки.

Конкретно, оценку \mathbb{E}_n можно выразить так:

$$\mathbb{E}_n = \sum_{k=1}^m a_k \cdot \tilde{x}_{n-k} + \sum_{k=1}^l b_k \cdot \tilde{e}_{n-k},$$

где $\{b_k\}$ – коэффициенты фильтра для квантованной последовательности ошибок \tilde{e}_n . Блок-схемы кодера на передаче и декодера на приеме приведены

ниже на рис. 2.15.

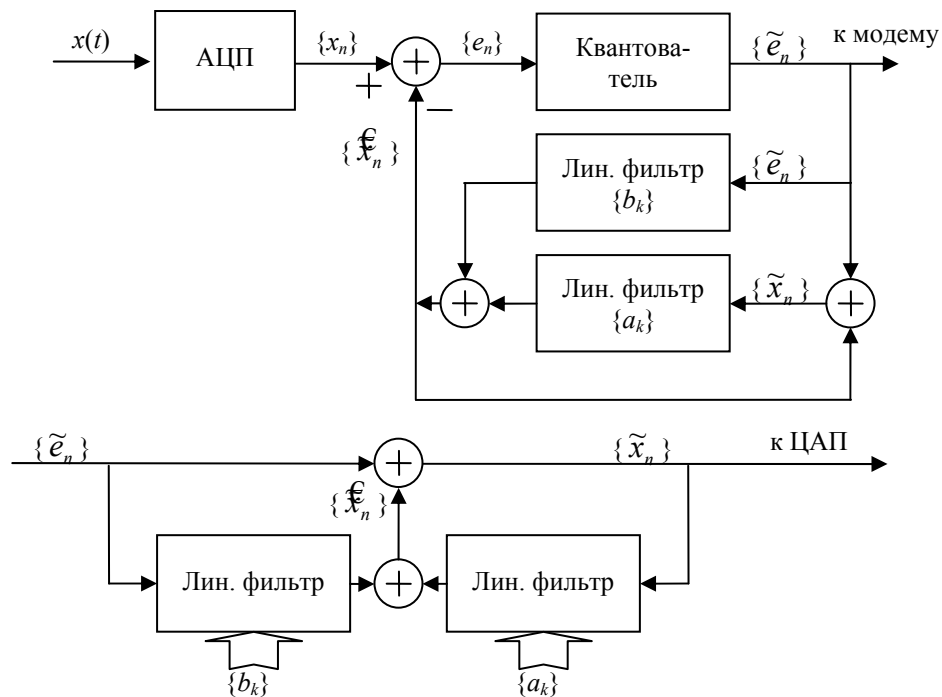


Рис.2.15. Блок-схема кодера и декодера ДИКМ с линейным фильтром

Здесь два ряда коэффициентов $\{a_k\}$ и $\{b_k\}$ выбираются так, чтобы минимизировать некоторую функцию ошибки $e_n = x_n - \hat{x}_n$, например, среднеквадратическую ошибку.

Наряду с равномерным (линейным) квантованием, при котором шаг квантования постоянен в пределах всего динамического диапазона, существует возможность согласования величины шага квантования с амплитудой сигнала. Для малых сигналов величина шага квантования мала, а для больших — возрастает. При этом с ростом амплитуды сигнала возрастает и мощность шумов квантования. Если использовать логарифмическую зависимость между выходным и входным сигналами, то можно поддерживать соотношение сигнал/шум квантования ниже заданной границы, как для малых, так и для больших по уровню сигналов (рис 2.16).

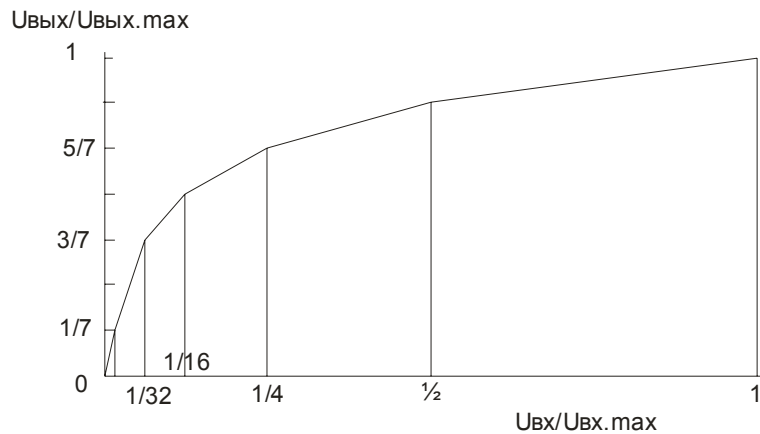


Рис. 2.16. Кусочно-линейная аппроксимация при адаптивном квантовании.

При использовании такого метода используется больше бит для слабых сигналов, где шум квантования наиболее заметен. Широко распространенный формат, использующий мю-функцию ($\mu - Law$), часто характеризуют как формат, сжимающий 12-битные отсчеты в 8-битные.

Адаптивная дифференциальная импульсно-кодовая модуляция. Многие реальные источники (например, источники РС), как уже было сказано выше, являются квазистационарными по своей природе. Одно из свойств квазистационарности характеристик случайного выхода источника заключается в том, что его дисперсия и автокорреляционная функция медленно меняются со временем. Кодеры ИКМ и ДИКМ, однако, проектируются в предположении, что выход источника стационарен. Эффективность и рабочие характеристики таких кодеров могут быть улучшены, если они будут адаптироваться к медленно меняющейся во времени статистике источника [5]. Как в ИКМ, так и в ДИКМ ошибка квантования q_n , возникающая в равномерном квантователе, работающем с квазистационарным входным сигналом, будет иметь меняющуюся во времени дисперсию (мощность шума квантования).

Одно улучшение, которое уменьшает динамический диапазон шума квантования, - это использование адаптивного квантователя. Другое - сделать адаптивным предсказатель в ДИКМ. При этом коэффициенты предсказателя могут время от времени меняться, чтобы отразить меняющуюся статистику источника сигнала. Определенные таким образом коэффициенты предсказателя могут быть вместе с ошибкой квантования \tilde{e}_n переданы приемнику, который использует такой же предсказатель. К сожалению, передача коэффициентов предсказателя приводит к увеличению необходимой битовой скорости, частично компенсируя снижение скорости, достигнутое посредством квантователя с немногими битами (немногими уровнями квантования) для уменьшения динамического диапазона ошибки e_n , получаемой при адаптивном предсказании.

В качестве альтернативы предсказатель приемника может вычислить свои собственные коэффициенты предсказания через \tilde{e}_n и \tilde{x}_n , где

$$\tilde{x}_n = \tilde{e}_n + \sum_{k=1}^M a_k \cdot \tilde{x}_{n-k} ; \quad (2.17)$$

Если пренебречь шумом квантования, \tilde{x}_n эквивалентно x_n . Следовательно, \tilde{x}_n можно использовать для оценки АКФ в приемнике.

Основное преимущество такого метода адаптации – это отказ от решения СЛАУ, что значительно уменьшает вычислительные затраты.

2.4.2. Кодирование в частотных поддиапазонах

Одним из факторов, снижающих эффективность дифференциального кодирования, является частота. При кодировании звуков низкой частоты обычно получается множество небольших приращений, в то время как при кодировании звуков высокой частоты – множество приращений большой величины. Один из способов повышения эффективности дифференциального кодирования заключается в делении частотного диапазона звука на несколько частей или выделении частотных поддиапазонов и последующей компрессии каждого из них по отдельности [5,7].

Знания о том, как устроен слух человека, помогут еще более эффективно использовать этот метод компрессии. Так как человек в одних диапазонах частот слышит лучше, чем в других, можно использовать различные параметры сжатия для разных поддиапазонов. Поддиапазоны, расположенные вблизи области слышимости, остаются практически без изменений, в то время как менее различимые поддиапазоны подвергаются большему сжатию, либо вообще не учитываются. Методы поддиапазонного кодирования основаны на математическом аппарате, используемом для выделения поддиапазонов, и на тщательном изучении слуха человека, необходимом для разработки рекомендаций по обработке каждого из диапазонов. Такое кодирование позволяет сжать ИКМ аудиоданные в 10-20 раз.

2.4.3. Аудио MPEG-1

Стандарт MPEG-1 сжатия видеофильмов состоит из двух основных частей: сжатия видео и сжатия звука.

Стандарт, описывающий MPEG-1, имеет нормативный и описательный разделы. Нормативный раздел содержит спецификации стандарта: таблицы с различными параметрами и кодами Хаффмана, которые используются в стандарте MPEG. Описательный раздел иллюстрирует выбранные концепции, объясняет причины выбора того или иного подхода, содержит необходимые базовые сведения, например, алгоритм задающий психоакустическую модель.

Основной принцип сжатия аудиоданных основан на сжатии с потерями. Общий принцип таков: в исходном WAV файле, как известно, хранится полная информация об исходном звуке, оцифрованном и проквантованном с частотой

44 кГц. Именно эта информация и хранится на обычных аудио-CD. В соответствии с теоремой Котельникова, этой информации абсолютно достаточно для воспроизведения всех частот исходного сигнала, меньших половины частоты квантования. Т.е. все частоты до 22кГц включительно воспроизводятся так же, как они звучали при оцифровке.

Для сжатия аудио данных самое удачное решение было найдено, разработано и запатентовано учеными из немецкого университета имени Фраунгофера. Формат файлов, (и стандарт) который они разработали, носил название MPEG Layer-3 (сокращенно MP3). Благодаря тому, что лицензия позволяла неограниченное и бесплатное некоммерческое использование формата, он приобрел широкое распространение и популярность и является доминирующим форматом сжатия до сих пор.

При кодировании в MP3 исходный звуковой файл режется на фрагменты, длительностью по 50 миллисекунд, каждый из которых анализируется отдельно. При анализе фрагмент раскладывается на гармоники по методу Фурье, из которых в соответствии с теорией восприятия звука человеческим ухом выбрасываются те гармоники, которые человек не воспринимает или воспринимает хуже на фоне остальных. Кроме того, выбрасываются звуки, замаскированные вследствие инертности слуха. Информация об оставшихся после фильтрации гармониках и записывается в MP3 файл, который в результате получается гораздо меньше по размеру, чем исходный WAV.

При воспроизведении производится обратное преобразование, при котором оставшиеся гармоники вновь преобразуются в звуковую волну. Получающийся звук не совпадает с исходным, но поскольку отбрасывались малозначимые и не слышимые звуки, отличить сигнал от исходного для человеческого уха достаточно сложно.

Смысл понятия "битрейт". Битрейт (англ. *bitrate*) — буквально, скорость прохождения битов информации. Битрейт принято использовать при измерении эффективной скорости передачи информации по каналу, то есть скорости передачи «полезной информации» (помимо таковой по каналу может передаваться служебная информация, например, стартовые и стоповые символы при асинхронной передаче, или контрольные символы при избыточном кодировании). Термин **битрейт** используется в двух основных значениях [7]:

1. Характеристика канала или устройства – максимальное количество бит, которое можно передать в единицу времени
2. Величина потока данных передаваемого в реальном времени (минимальный размер канала, который сможет пропустить этот поток без задержек). Частный случай – битрейт сжатого звука или видео.

Чем меньше битрейт, тем меньший размер имеют файлы с одинаковой по времени длине. С другой стороны, чем он меньше, тем большее количество "лишних" гармоник приходится отбрасывать. При очень низких битрейтах (16-24 килобита/с) музыку сложно воспринимать, а голос, хотя и остается разборчивым, приобретает весьма "психоделическую" окраску.

Переменный битрейт позволяет уменьшить размер файла при таком же

качестве за счет устранения избыточности. Иначе говоря, нет необходимости кодировать тишину с битрейтом 256 кбит/с, если ее с точно таким же качеством можно воспроизвести с битрейтом 64 кбит/с.

Качество аудиотреков в MPEG-1 может варьироваться в очень больших пределах – от высококачественного до крайне низкого. Окончательно все форматы сжатия аудиоданных были стандартизированы в 1992 году европейской комиссией по стандартам ISO. В зависимости от используемого кодера и степени сжатия аудиоинформация видеоролика может быть представлена в следующем виде: моно, dual mono, стерео, интенсивное стерео (стереосигналы, чьи частоты превышают 2 кГц объединяются в моно), m/s стерео (один канал - сумма сигналов, другой - разница) и по частоте дискретизации могут быть: 48, 44.1 и 32 кГц.

2.5. Общая схема аудиокодера и аудиодекодера MPEG

Алгоритм сжатия MPEG, как и любой другой алгоритм сжатия, можно разделить на три этапа:

- предварительная обработка;
- основное преобразование;
- кодирование и упаковка компонент преобразования.

На этапе предварительной обработки производится, в общем случае, подготовка исходного потока аудио данных к выполнению процедуры основного преобразования. В частности, можно выделить два вида такой подготовки: разбиение на блоки и фильтрация шумов.

На втором этапе с помощью дискретного преобразования Фурье (ДПФ) входные отсчеты ИКМ преобразуются в 512 спектральных составляющих. Таким образом осуществляется переход от временного представления сигнала к частотному [5].

На этапе кодирования и упаковки компонент производится анализ частотной области психоакустической моделью, которая отбрасывает неслышимые компоненты спектра и вычисляет шаг квантования, при котором шум квантования будет не слышен. Также на этом этапе осуществляется само квантование оставшихся спектральных отсчетов и, далее, они подвергаются кодированию по методу Хаффмана.

Алгоритм восстановления сигнала гораздо проще и состоит из двух этапов [5]:

- восстановление отсчетов;
- обратное преобразование.

На этапе восстановления отсчетов происходит декодирование спектральных компонент.

На втором этапе с помощью обратного ДПФ производится переход к временному представлению сигнала.

На рис. 2.17 приведены структурные схемы кодера и декодера MPEG.

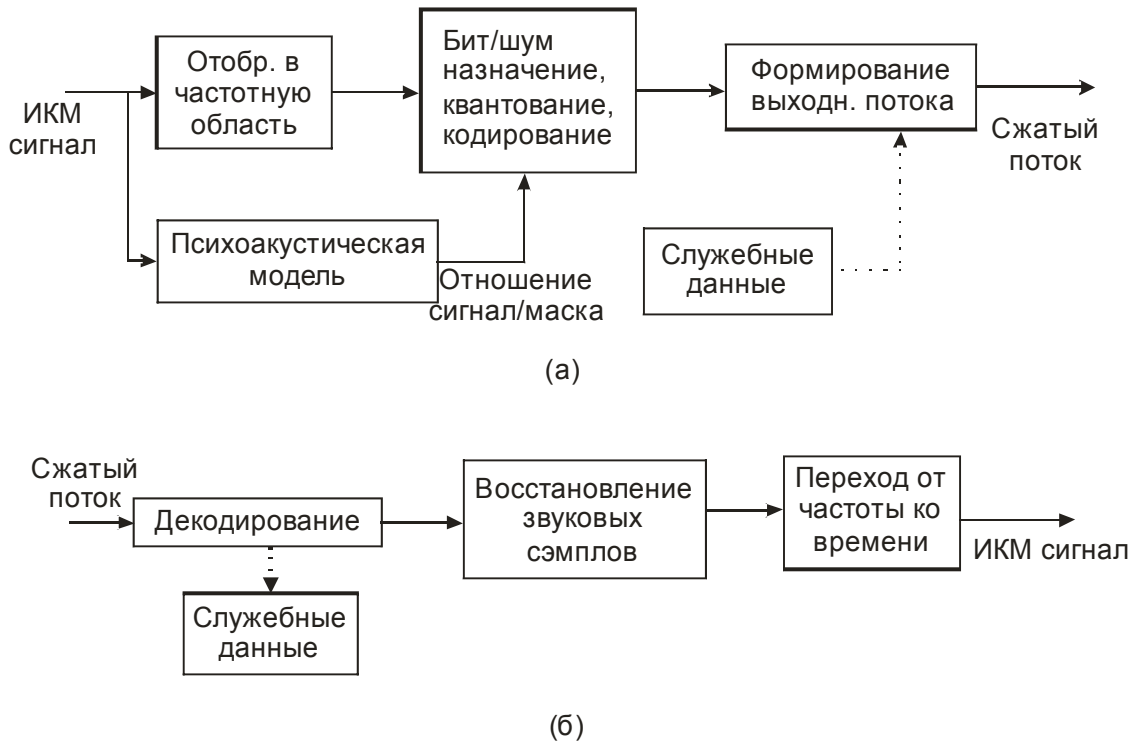


Рис. 2.17. Кодер (а) и декодер (б) MPEG.

Отображение в частотную область. Первый шаг кодирования аудио сигнала заключается в преобразовании его в частотную область. Для этого используют дискретное преобразование Фурье (ДПФ) [7]:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)nk}, k = 0, 1, \dots, N-1, \quad (2.18)$$

где $X(k)$ – отсчеты спектральной плотности, $x(n)$ – отсчеты исходного сигнала.

Так как аудио данные могут быть достаточно велики, то для вычисления спектра используется "окно" сканирования: спектр находится только для отсчетов попавших в "окно". После вычисления спектра "окно" сдвигается вправо (рис. 2.19).

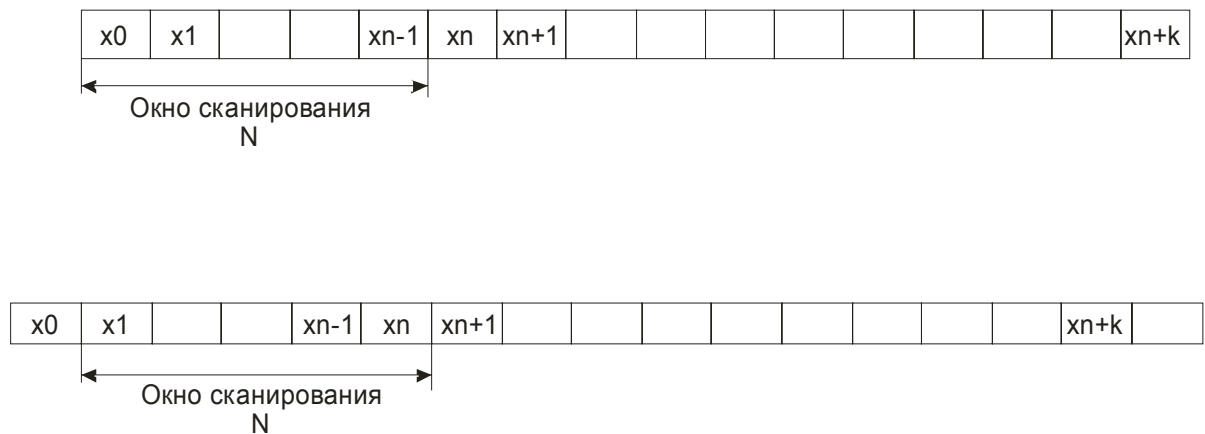


Рис. 2.19. Сканирование аудио потока "окном" N

Сдвиг может производиться как на один отсчет сигнала, так и сразу на несколько. В кодеке MPEG сдвиг "окна" производится на 32 отсчета, при размере "окна" 512 или 1024. Весь спектр делится на 32 частотные полосы равной ширины (частотные поддиапазоны). На следующем этапе происходит устранение избыточности с помощью анализа спектра сигнала в психоакустической модели для каждой из частотных полос. Заметим, что подобные идеи используются в AAC и других современных стандартах при кодировании частотных диапазонов (кодировании спектрограмм или сонограмм). Под сонограммой понимается график зависимости амплитуды от частоты и от времени. На рис.2.20 приведена для примера временная диаграмма колебаний струны, а на рис. 2.21 – соответствующая ей сонограмма.

При анализе сонограмм (рис 2.21) видно, что спектр мало изменяется во времени, меньшие изменения в области низких частот и чуть большие в области верхних частот.

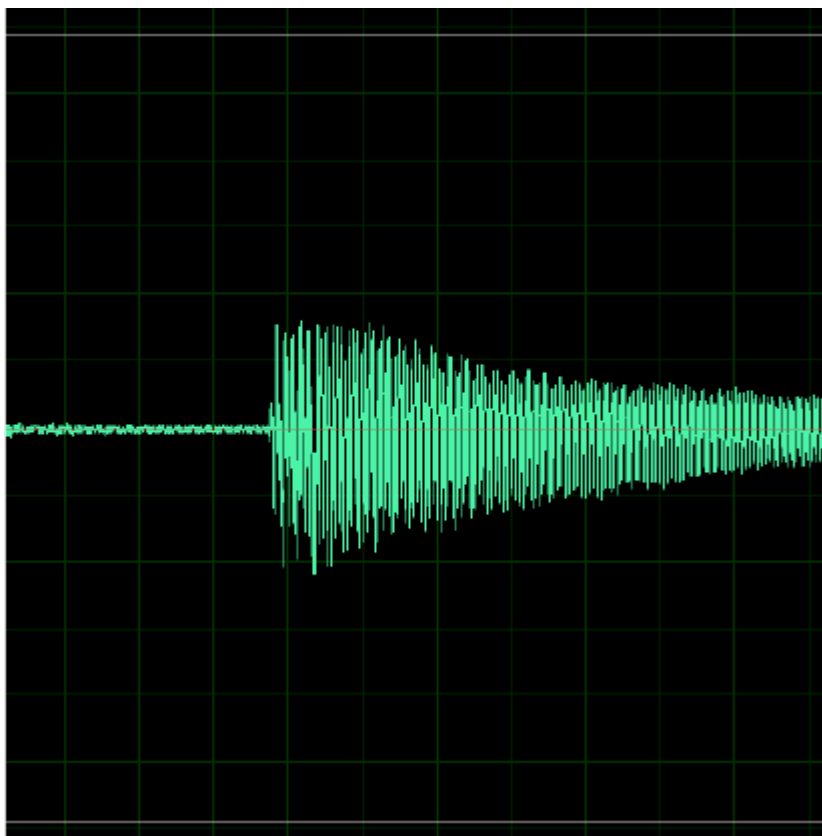


Рис. 2.20. Временная диаграмма колебания струны

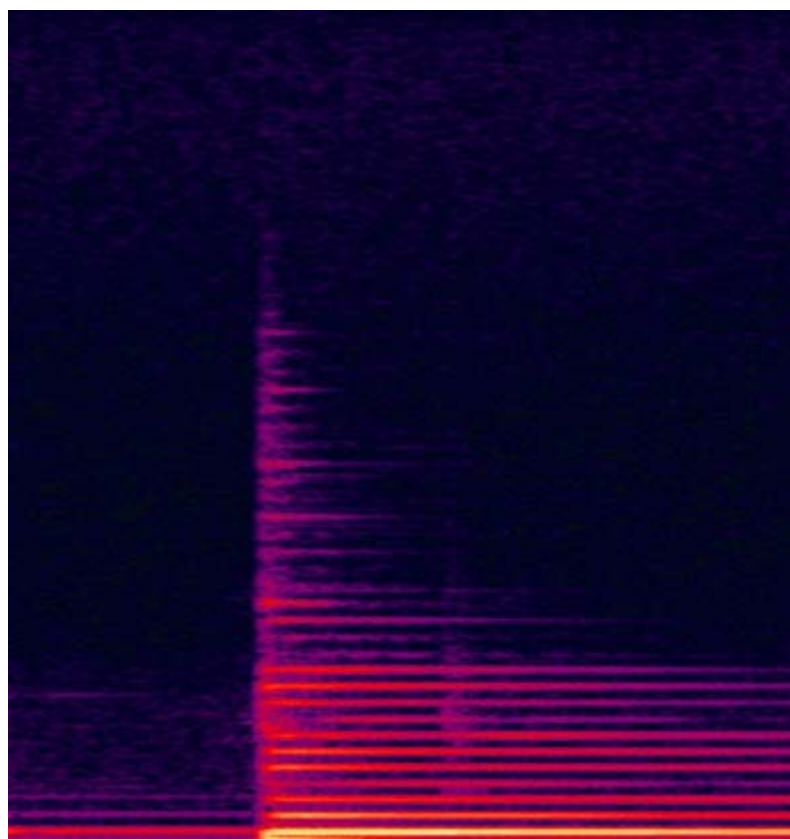


Рис. 2.21. Сонограмма колебания струны.

Следовательно, отличия в соседних спектрах соответствующих отсчетов будут незначительны, а иногда их совсем не будет. Поэтому эффективнее кодировать разность между соседними спектральными отсчетами, а не сами отсчеты.

Дискретное косинусное преобразование. В большинстве современных стандартах сжатия звука (в том числе, в стандарте MPEG) применяется дискретное косинусное преобразование (ДКП) выглядит следующим образом:

$$X(k) = Ck \sum_{n=0}^{N-1} x(n) \cos\{(2\pi + 1)nk/16\}, n, k = 0, 1, \dots, N-1. \quad (2.19)$$

В выражении (1.20) множитель Ck является нормирующим и равен $1/\sqrt{2}$ при $k=0$ и равен единице для остальных значений индексов. Если производить вычисления по этой формуле, то вычислительная сложность составит $Q_{\text{ДПФ}} = N^2$ базовых операций (БО). Но принимая во внимания особенности матрицы преобразования, можно сократить вычислительные затраты [5].

Психоакустическая модель. Психоакустическая модель позволяет сократить спектр сигнала без изменения восприятия сжатого аудио сигнала, так как она учитывает особенности человеческого слуха и отбрасывает неслышимые частоты, которые маскируются более громкими звуками.

Психоакустическая модель дает возможность кодеру определить порог допустимого шума квантования для каждого поддиапазона. Эта информация будет использована алгоритмом назначения битов, что в сочетании с количеством имеющихся битов задаст число уровней квантования для каждой подполосы.

Стандарт сжатия звука MPEG разрешает значительную свободу при реализации моделей. Изошренность этой реализации в конкретном кодере зависит от требуемой степени сжатия. В приложениях широкого потребления, в которых не требуется высокий фактор сжатия, психоакустическая модель может вовсе отсутствовать. В этом случае алгоритм назначения битов не использует соотношение сигнал/маскирование (SMR – signal to mask ratio).

Основные шаги модели состоят в следующем:

1. Спектральные значения частотных полос разделяются на тональные (подобные синусоиде) и нетональные (шумоподобные) компоненты. Компонента спектра $X(k)$ считается тональной¹², если:

$$\begin{aligned} X(k) - X(k+j) &\geq 7\text{дБ} \\ j = -2, +2; 2 < k < 63 \\ j = -3, -2, +2, +3; 63 \leq k < 127 \\ j = -6, \dots, -2, +2, \dots, +6; 127 \leq k < 255 \\ j = -12, \dots, -2, +2, \dots, +12; 255 \leq k < 511 \end{aligned} \quad (2.20)$$

Полученные компоненты исключаются из исходного спектра, оставшиеся компоненты являются нетональными.

¹² Окно сканирования 512

2. Прореживается спектр тональных компонент. Исключаются компоненты, лежащие ниже абсолютного порога слышимости. Оставшиеся компоненты прореживаются с помощью "окна" шириной 0,5 Барк.

3. В каждом поддиапазоне вычисляется порог маскирования для тональных и не тональных компонент. Маскируемые компоненты отбрасываются.

4. Вычисляется общее отношение сигнал/маскирование (SMR) для каждой субполосы кодирования.

Сжатие коэффициентов. На следующем этапе происходит квантование и сжатие оставшихся спектральных компонент. Количество уровней квантования выбираются на основании соотношения сигнал/маскирование, полученного на предыдущем этапе.

Сжатие коэффициентов осуществляется методами сжатия без потерь, например алгоритмом Хаффмана.

Декодер. Декодер MPEG проще кодера и состоит всего из двух этапов:

- восстановление отсчетов спектральной плотности;
- выполнение обратного преобразования Фурье.

Восстановление отсчетов спектральной плотности происходит из данных сжатых алгоритмом без потерь, в основном используется алгоритм Хаффмана.

После выполнения обратного ПФ получается сигнал в ИКМ, готовый для воспроизведения или записи на носитель.

AAC (Advanced Audio Coding). AAC - формат аудио-файла с меньшей потерей качества при кодировании, чем MP3 при одинаковых размерах. На 2005 год распространён существенно меньше, чем MP3 и другие альтернативные решения. Этот стандарт изначально создавался как преемник MP3 с улучшенным качеством кодирования. Формат AAC, официально известный как ISO/IEC 13818-7, вышел в свет в 1997 как новая, седьмая, часть стандарта MPEG-2. Существует также формат AAC, известный как MPEG-4 Часть 3 [7].

AAC представляет собой широкополосный алгоритм кодирования аудио, в котором использует два основных принципа кодирования для сильного сжатия:

- удаляются не воспринимаемые слухом составляющие сигнала в соответствии с психоакустической моделью;
- удаляется избыточность в кодированном аудио сигнале, а затем сигнал обрабатывается MDCT согласно его сложности.

Семейство алгоритмов аудио кодирования MPEG-4 охватывает диапазон от кодирования низкогокачественной речи (до 2 кбит/с) до высококачественного аудио (от 64 кбит/с на канал и выше). AAC имеет частоту сэмплов от 8 Гц до 96 кГц и количество каналов от 1 до 48.

В отличие от гибридного набора фильтров MP3, AAC использует MDCT вместе с увеличенным размером окна до 2048 отсчетов. AAC более подходит для кодирования аудио с потоком сложных импульсов и прямоугольных сигналов чем MP3 или Musicam. AAC может динамически переключаться между блоками MDCT от 2048 отсчетов до 256 отсчетов. При кратковременном

анализе используется малое окно в 256 отсчетов для лучшего разрешения. По умолчанию используется большое окно в 2048 отсчетов для улучшения эффективности кодирования.

2.6. Анализ возможностей алгоритмов сжатия аудиосигналов

Для того чтобы грамотно выбрать речевой кодек, достаточно представления об используемом в нем методе (на котором базируется алгоритм кодирования) и о процессе согласования сигнала, полученного после цифровой обработки, с цифровым каналом связи.

Поскольку рассматриваемые методы кодирования являются методами сжатия звука с потерями, то при восстановлении (декодировании) звукового сигнала наблюдаются искажения сигнала.

В качестве примера подобных искажений на рис. 2.22. приведен фрагмент исходного звукового сигнала, а на рис. 2.23 и рис.2.24 – соответствующие ему фрагменты после восстановления кодеками MPEG AAC и GSM 6.10.

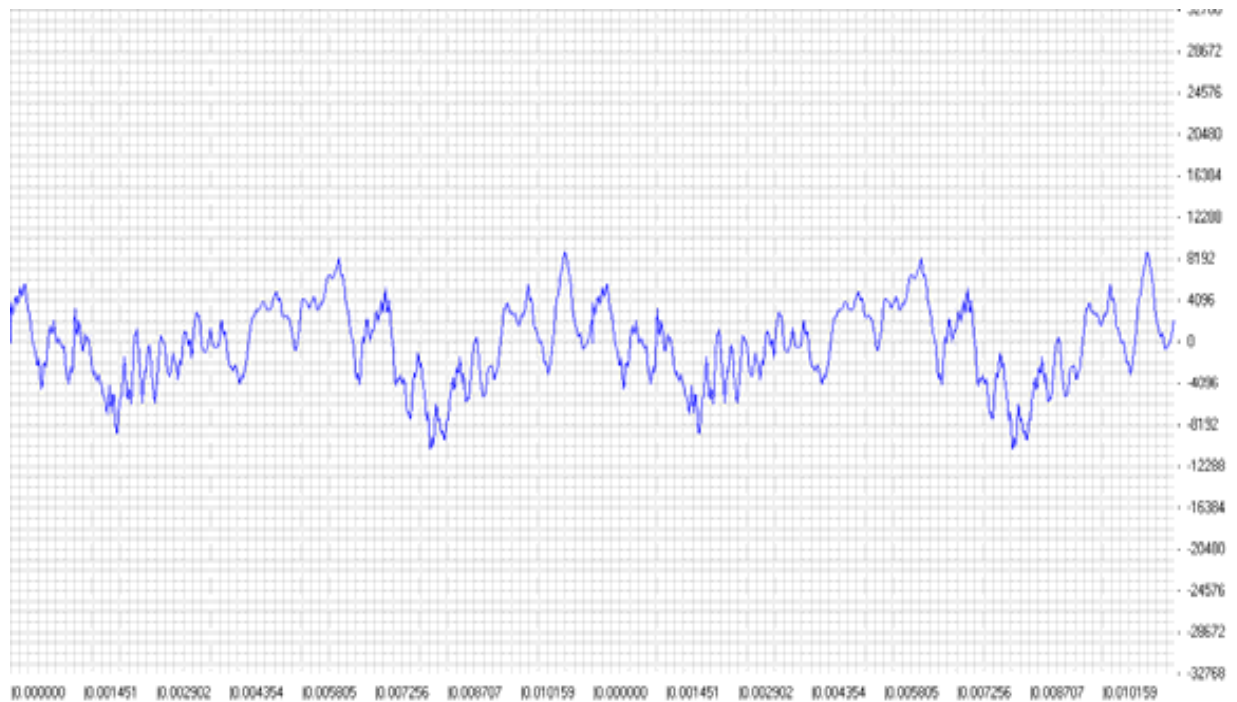


Рис.2.22. Фрагмент звукового сигнала
(по оси X – время, мс, по оси Y – амплитуда)

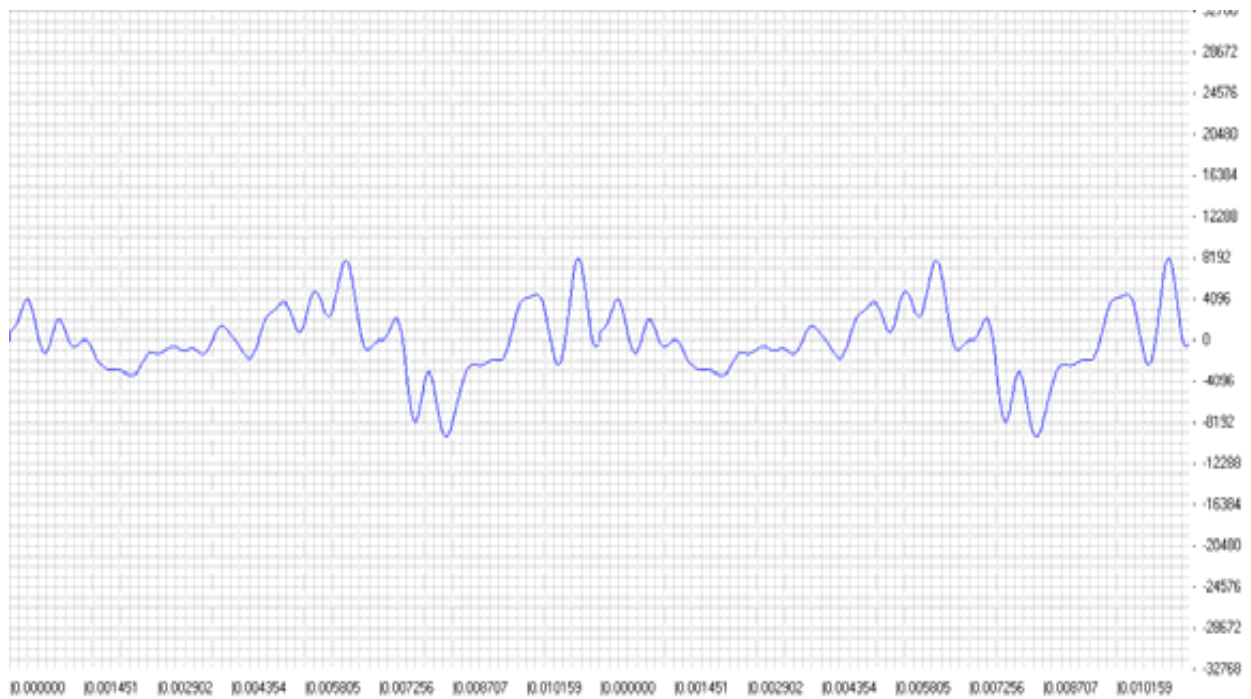


Рис.2.23. Фрагмент того же сигнала после декодирования кодеком MPEG AAC

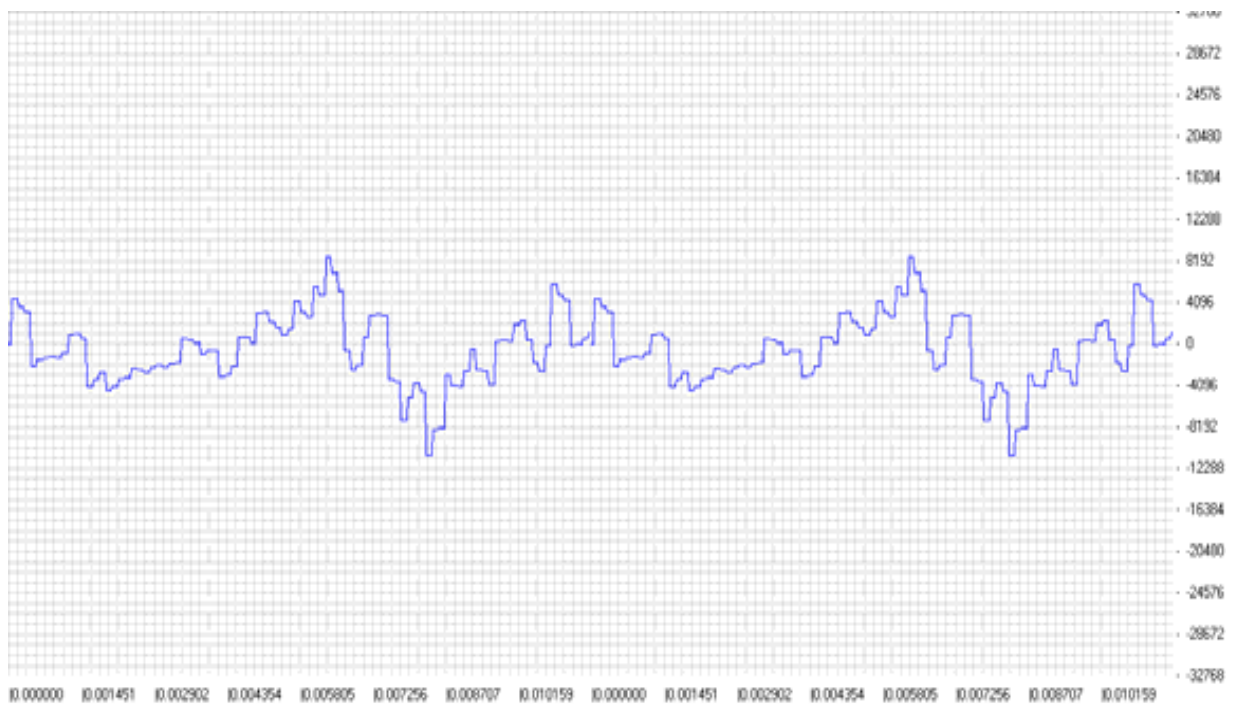


Рис.2.24. Фрагмент того же сигнала после декодирования кодеком GSM 6/10

Из рис.2.24 следует, что применение кодека MPEG AAC приводит к устранению высокочастотных компонент сигнала (Эффект сглаживания, а кодек GSM 6.10 вносит характерную ступенчатую структуру формы сигнала что связано с существенно меньшим размером базового фрейма при сжатии и отсутствием перекрытий между блоками.

Качественные показатели методов кодирования речи.

Зависимость качества звучания речи (в баллах по шкале MOS) от скорости передачи при использовании различных методов кодирования показана на рис.2.25.

По эффективности (т.е. соотношению битрейта к заданному качеству) методы кодирования можно разделить на три группы. К первой группе - низкоэффективных кодеров - относятся ИКМ и АДМ (адаптивная дельта-модуляция), которые обеспечивают удовлетворительное качество звука только при скоростях передачи выше 24 кбит/с. Кодеры второй группы - ОПА, МПК, ЛПКВ, АДИКМ - позволяют реализовать удовлетворительное и отличное качество звучания при скоростях 8-32 кбит/с. В третью группу входят ЛПКВ- и ЛПМВ-кодеры, обеспечивающие отличное качество при низких скоростях.

В большинстве алгоритмов эффективного кодирования речи вероятность однократной ошибки составляет 10^{-4} - 10^{-3} на символ. По зависимости помехоустойчивости от скорости передачи (рис. 2.26) кодеры можно разделить на две группы: в первую (ИКМ, АДМ, АДИКМ) входят алгоритмы с сильной зависимостью, во вторую (МПК, ЛПКВ, ОПА, ЛПМВ) - со слабой.

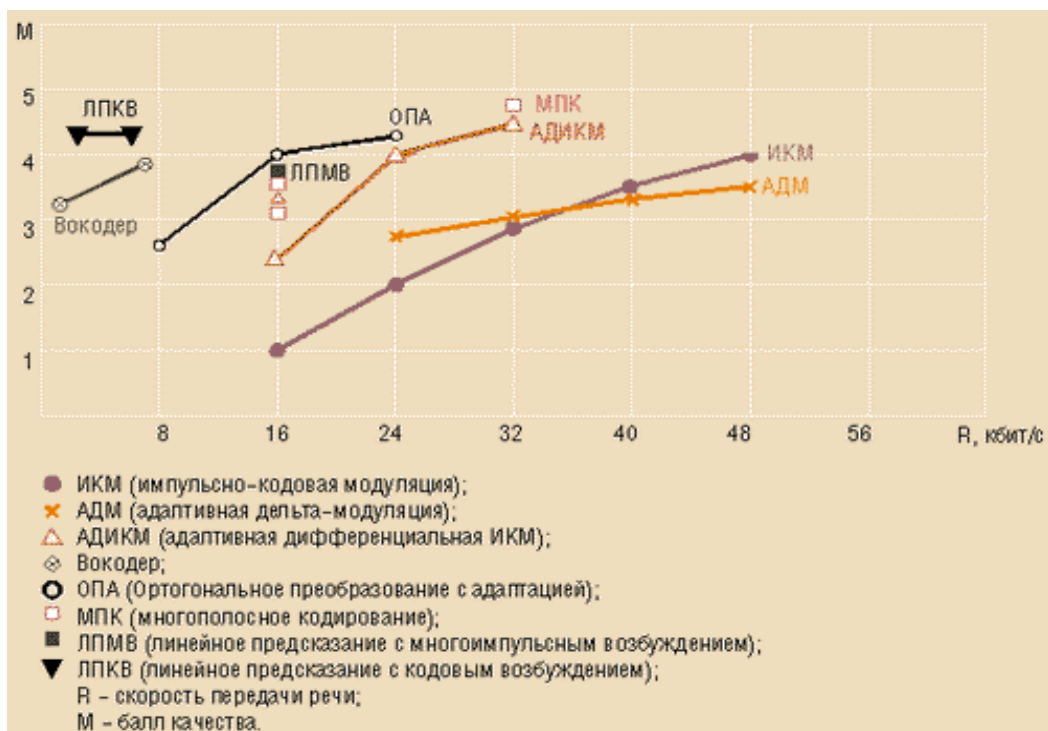


Рис. 2.25. Зависимость качества восстановления от битрейта R для различных методов кодирования

Под помехоустойчивостью подразумевается максимальная вероятность ошибки, при которой качество звучания сигнала (отношение сигнал/шум) снижается не более чем на 10% (в испытаниях по методу парных сравнений различия между сигналами составляют не более 20%).

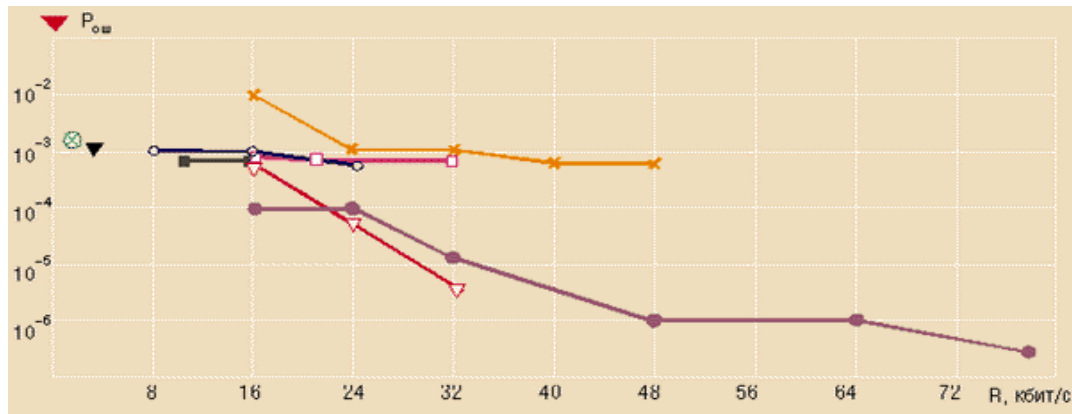


Рис. 2.26. Зависимость вероятности появления одиночной ошибки при декодировании (помехоустойчивости) кодеров от битрейта R (условные обозначения те же, что на рис. 1.6)

Необходимо отметить, что большинство кодеров предназначены для высококачественных цифровых каналов, но, несмотря на это, допустимая и максимально удерживаемая вероятности ошибки являются стандартными характеристиками всех эффективных кодеров.

Выводы. Рассмотрение существующих кодеков позволяет сформулировать ряд основных тенденций:

1. Доминирующее положение при построении низкоскоростных кодеков речи метода кодирования на основе линейного предсказания.
2. Возрастание доли адаптивных процедур обработки сигналов в современных системах кодирования речи.
3. Однозначная связь качества синтезированной речи на низких скоростях кодирования со степенью адаптации соответствующих кодеков речевых сигналов.
4. Наиболее перспективными являются алгоритмы сжатия типа CELP и его вариаций, а также векторного кодирования HVXC, положенные в основу методов MPEG AAC, и более современных стандартов MPEG.
5. Использование более сложных алгоритмов энтропийного кодирования (SABAS – контекстно-адаптивного арифметического кодирования)

Таким образом, анализ степени адаптации современного парка кодеков указывает на перспективность перехода систем кодирования речи к многопараметрической адаптации в условиях априорной и текущей неопределенности в описании моделей речевого сигнала и внешней среды функционирования кодека.

Наибольшая компрессия достигается в методах, которые учитывают особенности человеческого слуха, использующих разбиение на поддиапазоны и последующего проведения в них анализа. Некоторые примеры таких методов приведены в табл. 2.3. Кодеки MPEG являются наиболее распространенными и используют наиболее перспективный метод сжатия.

Таблица 1.3.

Наименование алгоритма компрессии аудиоданных	Метод компрессии	Скорость передачи, кбит/с на канал	Величина компрессии	Области применения
ASPEC	Кодирование с преобразованием	64-192	1:6	ISDN
ATRAC	Субполосное кодирование с преобразованием	256	1:5	MiniDisk
MUSCAM	Субполосное кодирование	128-256	1:4	DAB (Digital Audio Broadcasting)
MUSICAM	Субполосное кодирование	128-256	1:4	DAB
MPEG-1, Layer 1 и Layer 2	Субполосное кодирование (MUSICAM)	32-448 (Layer 1) 32-384 (Layer 2)	1:4 (Layer 1) 1:6 (Layer 2)	DAB (Layer 2, 128-256 кбит/с), DBS (Direct Broadcast Satellite, Layer 2, 224 кбит/с), DCC (Digital Compact Cassete, Layer 1, 384 кбит/с)
MPEG-1, Layer 3	Субполосное кодирование с преобразованием	32-320	1:9	Internet-вещание
MPEG-2	Субполосное кодирование/субполосное кодирование с преобразованием	32-384	> 1:9	Многоканальное стереофоническое вещание
MPEG-2 AAC	Субполосное кодирование с преобразованием	16-384	1:15	Многоканальное стереофоническое вещание
MPEG-4	Субполосное кодирование с преобразованием/параметрическое кодирование	2-64	-	Мультимедиа приложения
Dolby AC-3	Кодирование с преобразованием	32-384	1:13	Кинематограф, HDTV, спутниковое вещание
Гибридное	Субполосное кодирование с преобразованием/	32-64	1:(15-20)	Радиовещание, хранение информации

	параметрическое кодирование			
--	--------------------------------	--	--	--

3. СТАНДАРТЫ СЖАТИЯ ВИДЕОДАННЫХ

Передача цифрового видео от источника (видеокамера или записанный видеоролик) к получателю (видеодисплей) вовлекает в разработку целую цепь различных компонентов и процессов. Ключевыми звеньями этой цепи являются процесс компрессии (кодирования) и декомпрессии (декодирования), при которых несжатый цифровой видеосигнал сокращается до размеров, подходящих для его передачи и хранения, а затем восстанавливается для отображения на видеоэкране.

Продуманная разработка процессов компрессии и декомпрессии может дать существенное коммерческое и техническое преимущество продукта, обеспечив лучшее качество видеоизображения, большую надежность и гибкую приспособляемость по сравнению с конкурирующими решениями.

Таким образом, имеется живая заинтересованность в развитии и улучшении методов компрессии и декомпрессии видео [3,6,8].

Недавние успехи цифровой видеоиндустрии (прежде всего широковещательного цифрового видео и DVD-видео) базировались на международном стандарте ISO/IEC 13818, широко известном под аббревиатурой MPEG-2 (назван так по имени рабочей группы экспертов по движущимся изображениям, разработавшей этот стандарт, - Moving Picture Experts Group). Предвосхищение насущной необходимости лучшего сжатия породило разработку дальнейших стандартов видеосжатия, известных под названиями ISO/IEC 14496 Part 2 (MPEG-4 Visual) и Рекомендация организации ITU-E H.264/ISO/IEC 14496 Part 10 (сокращенно H.264). Стандарты MPEG-4 Visual и H.264 имеют общее происхождение и многие общие черты. Они оба были разработаны на основе более ранних стандартов сжатия. Однако они развивают старые стандарты в существенно различных направлениях. Стандарт MPEG-4 Visual удаляется от прямоугольного видеокадра, предлагает гибкий и открытый взгляд на визуальные коммуникации и использует высокоэффективное видеосжатие и объектно-ориентированную обработку данных. Стандарт H.264 имеет более прагматичный взгляд. Он стремится выполнять те же действия, что и предыдущие стандарты (обеспечивая механизм сжатия для прямоугольных кадров), но с большей эффективностью и устойчивостью, обеспечивая совместимость со всеми широко распространенными типами приложений, такими как широкое телевидение, хранение визуальной информации и передача потокового видео.

3.1. Особенности сжатия видеоданных

Скорости передачи данных в сетях, а также емкости жестких дисков, флэш-памяти и оптических накопителей постоянно растут. Имея в виду снижение цены передачи и хранения бита информации, не сразу становится очевидным, почему необходимо видеосжатие и его улучшение. Видеосжатие имеет два важных преимущества. Во-первых, оно дает возможность

использовать цифровое видео в среде передачи и хранения видеоконтента, которая не поддерживает несжатое («сырое») видео. Например, пропускная способность современного Интернета недостаточна для обращения с несжатым видео в реальном масштабе времени даже при низкой частоте кадра и малом его размере. Цифровой многослойный видеодиск DVD может вместить всего несколько секунд несжатого видео с разрешением и частотой кадров, обеспечивающими обычное телевизионное качество, поэтому использование DVD было бы абсолютно непрактичным без применения аудио и видеосжатия. Во-вторых, видеосжатие делает более эффективным использование ресурсов при передаче и хранении видеоданных. Если доступен высокоскоростной канал, то более привлекательным представляется решение, позволяющее передавать сжатое видео высокого разрешения вместо несжатого видео низкого разрешения. Несмотря на постоянный рост емкости устройств хранения информации и пропускной способности каналов передачи данных, представляется весьма вероятным, что сжатие видео будет оставаться существенным компонентом мультимедийных сервисов еще многие годы.

Сигнал, несущий определенную информацию, можно сжать путем удаления из него имеющейся избыточности. Избыточность – это компоненты данных, без которых можно обойтись для верного изображения исходной информации. Многие типы данных содержат в себе статистическую избыточность. Такие данные можно эффективно сжимать, используя компрессию без потерь. К сожалению, сжатие без потерь применительно к видео дает относительно небольшой выигрыш. Поэтому для достижения высокой эффективности сжатия приходится применять сжатие с потерями. При сжатии видео с потерями используется несколько типов избыточности:

- когерентность областей изображения - малое изменение цвета изображения в соседних пикселях (свойство, которое эксплуатируют все алгоритмы сжатия изображений с потерями);
- избыточность в цветовых плоскостях - используется большая важность яркости изображения для восприятия;
- подобие между кадрами - использование того факта, что на скорости 25 кадров в секунду, как правило, соседние кадры изменяются незначительно.

Использование подобия между кадрами в самом простом и наиболее часто используемом случае означает кодирование не самого нового кадра, а его разности с предыдущим кадром. Для видео типа "говорящая голова" (передача новостей, видеотелефоны) большая часть кадра остается неизменной и даже такой простой метод позволяет значительно уменьшить поток данных. Более сложный метод заключается в нахождении для каждого блока в сжимаемом кадре наименее отличающегося от него блока в кадре, используемом в качестве базового. Далее кодируется разница между этими блоками. Этот метод существенно более ресурсоемкий.

Требования приложений к алгоритму кодирования (сжатия) [3,6]:

1. Произвольный доступ - подразумевает возможность найти и показать любой кадр за ограниченное время. Обеспечивается наличием в потоке данных так называемых точек входа - кадров, сжатых независимо (т. е. как обычное статическое изображение). Приемлемым временем поиска произвольного кадра считается 1/2 с.

2. Быстрый поиск вперед/назад - подразумевает быстрый показ кадров, не следующих друг за другом в исходном потоке. Требует наличия дополнительной информации в потоке. Эта возможность активно используется всевозможными проигрывателями.

3. Показ кадров фильма в обратном направлении. Редко требуется в приложениях. При жестких ограничениях на время показа очередного кадра выполнение этого требования может резко уменьшить степень сжатия.

4. Аудиовизуальная синхронизация - самое серьезное требование. Данные, необходимые для того, чтобы добиться синхронности аудио и видео дорожек, существенно увеличивают размер видеоданных. Для видеосистемы это означает, что если не успеваем достать и показать в нужный момент времени некий кадр, то необходимо корректно показать кадр, следующий за ним.

5. Устойчивость к ошибкам - требование, обусловленное тем, что большинство каналов связи ненадежны. Испорченное помехой изображение должно быстро восстанавливаться. Требование достаточно легко удовлетворяется необходимым числом независимых кадров в потоке. При этом также уменьшается степень сжатия, так как на экране 2-3 с (50-75 кадров) может быть одно и то же изображение, но мы будем вынуждены нагружать поток независимыми кадрами.

6. Время кодирования/декодирования. Во многих системах (например, видеотелефонах) общая задержка на кодирование-передачу-декодирование должна составлять не более 150 мс. Кроме того, в приложениях, где необходимо редактирование, нормальная интерактивная работа невозможна, если время реакции системы составляет более 1 с.

7. Редактируемость. Под редактируемостью понимается возможность изменять все кадры так же легко, как если бы они были записаны независимо.

8. Масштабируемость - простота реализации концепции "видео в окне". Необходимо быстро изменять высоту и ширину изображения в пикселах. Масштабирование способно породить неприятные эффекты в алгоритмах, основанных на дискретном косинусном преобразовании. Корректно реализовать эту возможность для MPEG на данный момент можно, пожалуй, лишь при достаточно сложных аппаратных реализациях, только тогда алгоритмы масштабирования не будут существенно увеличивать время декодирования. Интересно, что масштабирование достаточно легко осуществляется в так называемых фрактальных алгоритмах. В них, даже при увеличении изображения в несколько раз, оно не распадается на квадраты, т. е. отсутствует эффект "зернистости". Если необходимо уменьшать изображение (что хоть и редко, но бывает нужно), то с такой задачей хорошо справляются

алгоритмы, основанные на wavelet-преобразовании

9. Небольшая стоимость аппаратной реализации. При разработке хотя бы приблизительно должна оцениваться и учитываться конечная стоимость. Если эта стоимость велика, то даже при использовании алгоритма в международных стандартах производители будут предлагать свои, более конкурентоспособные алгоритмы и решения. На практике это требование означает, что алгоритм должен реализовываться небольшим набором микросхем.

Описанные требования к алгоритму противоречивы. Очевидно, что высокая степень сжатия подразумевает архивацию каждого последующего кадра с использованием предыдущего. В то же время требования на аудиовизуальную синхронизацию и произвольный доступ к любому кадру за ограниченное время не дают возможности вытянуть все кадры в цепочку. И, тем не менее, можно попытаться прийти к некоторому компромиссу. Сбалансированная реализация, учитывающая систему противоречивых требований, может достигаться на практике за счет настроек компрессора при сжатии конкретного видеопотока [3].

3.2. Основные процедуры сжатия видеоданных

Видеокодек кодирует исходную видеопоследовательность в сжатой форме, а также декодирует сжатую видеопоследовательность, производя цифровую видеокопию, которая или совпадает, или близка к исходной видеопоследовательности.

Кодек преобразует исходный видеоряд с помощью определенной модели. Модель кодирования – это эффективное кодированное представление видеоданных, с помощью которого можно реконструировать эти данные с определенной степенью точности. В идеале модель должна представлять последовательность с наименьшим числом бит и наибольшей возможной точностью. Эти две цели (высокое качество и эффективность сжатия) обычно противоречат друг другу, так как высокая степень сжатия видеоданных предполагает существенное снижение качества на выходе декодера.

Видеокодек состоит из следующих основных функциональных блоков:

- блока преобразования цветового пространства;
- блока устранения временной статистической взаимосвязи (схожести соседних по времени кадров видеопотока между собой);
- блока устранения пространственной статистической взаимосвязи между соседними пикселями на кадре;
- блока энтропийного кодера.

3.2.1. Цветовые пространства и их преобразование

RGB. В цветовом пространстве RGB пиксели цветного изображения представляются с помощью трех чисел, указывающих относительное соотношение красного (Red), зеленого (Green) и голубого (Blue) цветов (три основные компоненты видимого света). Любой цвет можно получить с помощью комбинации красного, зеленого и голубого цветов в соответствующей пропорции. Пространство RGB хорошо приспособлено для фиксирования и показа цветных изображений. Цветные электроннолучевые трубки CRTs (Cathode Ray Tubes) и жидкокристаллические дисплеи отображают RGB-изображения, отдельно освещая красные, зеленые и голубые компоненты каждого пикселя в соответствии с интенсивностью каждого из них. Если смотреть на экран с расстояния обычного зрителя, то различные компоненты сливаются в единый «правильный цвет». В силу этого, цветовое пространство RGB применяется в компьютерной технике, однако при таком подходе практически нет возможности эффективного сжатия изображений, т.к. для правильного отображения, все три компоненты должны быть представлены с одинаковым разрешением. Следующее цветовое пространство предлагает выход из этой ситуации.

YCbCr. Известно, что органы зрения человека менее чувствительны к цвету предметов, чем к их яркости (светимости). 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 светимость от цветовой информации и представив ее с большим разрешением, чем цвет.

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

$$y = k_r R + k_g G + k_b B, \quad (3.1)$$

где k обозначает соответствующий весовой множитель.

Цветовая информация может быть представлена компонентами цветовых разностей, т.е. каждая из этих компонент представляет собой разность между компонентами R, G и B и компонентой светимости Y. Таким образом:

$$\begin{aligned} Cb &= B - Y \\ Cr &= R - Y \\ Cg &= G - Y \end{aligned} \quad (3.2)$$

Таким образом, цветное изображение полностью описывается компонентой светимости Y и тремя хроматическими составляющими. Возникает резонный вопрос – до преобразования имелось три компоненты составляющие изображение – стало четыре. Однако, зная две из трех хроматических составляющих, можно легко вычислить четвертую, так как сумма $Cb + Cr + Cg$ является постоянной. Для описания изображения выбираются составляющие Cb и Cr. Преимущества такого способа представления изображений состоит в том, что можно не сжимая компоненту

светимости Y , сжать световые составляющие, представив их с меньшим разрешением, что и осуществляется в алгоритме JPEG на втором шаге сжатия.

Перед тем, как отображать картинку на экране, требуется произвести обратное преобразование из YCbCr в RGB.

Формулы для прямого и обратного преобразования выглядят следующим образом [4,6,9]:

$$\begin{aligned}
 Y &= k_r R + (1 - k_b - k_r)G + k_b B \\
 C_b &= \frac{0.5}{1 - k_b} (B - Y) \\
 C_r &= \frac{0.5}{1 - k_r} (R - Y) \\
 R &= Y + \frac{1 - k_r}{0.5} C_r \\
 G &= Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} C_b - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} C_r \\
 B &= Y + \frac{1 - k_b}{0.5} C_b
 \end{aligned}$$

Рекомендация ITU-T с идентификатором BT.601 предлагает коэффициенты $k_b = 0.114$ и $k_r = 0.229$. С этими коэффициентами получаем следующие формулы:

$$\begin{aligned}
 Y_{601} &= 0.299R' + 0.587G' + 0.114B' \\
 C_b &= -0.172R' - 0.339G' + 0.511B' + 128 \\
 C_r &= 0.511R' - 0.428G' - 0.083B' + 128
 \end{aligned}$$

Данные формулы используются для кодирования восьмибитного сигнала RGB с диапазоном возможных значений от 16 до 235, то есть 16 соответствует полностью белому, а 236 – полностью черному цвету. Это сделано в целях улучшения передачи изображений и видеопоследовательностей по линиям передачи (телевидение). При таких значениях промежутки от 0 до 15 и от 237 до 256 содержит шум, который при преобразовании отбрасывается, что позволяет улучшить шумовые характеристики изображения.

При использовании в компьютерной технике необходимость в этом отпадает, и диапазон значений сигнала RGB является полным – от 0 до 256. В этом случае используются следующие формулы преобразования:

$$\begin{aligned}
 Y_{601} &= 0.257R' + 0.504G' + 0.098B' + 16 \\
 C_b &= -0.148R' - 0.291G' + 0.439B' + 128 \\
 C_r &= 0.439R' - 0.368G' - 0.071B' + 128
 \end{aligned}$$

В этом случае полностью белый цвет – 0, полностью черный – 256. В контексте использования данного кодирования в телевидении говорят о “суперчерном” и “супербелом” цветах.

Данный вариант преобразования YCbCr имеет название YCbCr: SDTV (Soft Definition Television). Существует также еще один вариант преобразования.

Недавно появившийся стандарт телевидения HDTV (High Definition Television) использует несколько иные формулы перехода из RGB в YCbCr при диапазоне RGB от 16 до 236:

$$\begin{aligned} Y_{709} &= 0.213R' + 0.715G' + 0.072B' \\ Cb &= -0.117R' - 0.394G' + 0.511B' + 128 \\ Cr &= 0.511R' - 0.464G' - 0.047B' + 128 \end{aligned}$$

и при полном диапазоне от 0 до 256

$$\begin{aligned} Y_{709} &= 0.183R' + 0.614G' + 0.062B' + 16 \\ Cb &= -0.101R' - 0.338G' + 0.439B' + 128 \\ Cr &= 0.439R' - 0.399G' - 0.040B' + 128 \end{aligned}$$

Другие цветовые пространства. Однако, цветовое пространство YCbCr – не единственное, использующее для передачи изображения компоненту светимости и две компоненты цветоразности. На данный момент существуют следующие цветовые пространства.

1. YUV, используемое стандартами PAL (Phase Alternation Line), NTSC (National Television System Committee) и SECAM (Sequential Color with Memory). Цветовое пространство YCbCr было разработано в рамках рекомендации ITU-R BT.601 на основе именно этого цветового пространства. Формулы перехода из RGB в YUV выглядят следующим образом:

$$\begin{aligned} 2. \quad Y' &= 0.299R' + 0.587G' + 0.114B' \\ 3. \quad U &= -0.147R' - 0.289G' + 0.436B' \\ 4. \quad V &= 0.615R' - 0.515G' - 0.100B' \end{aligned}$$

2. YIQ – это цветовое пространство также разработано на основе YUV и опционально используется в NTSC. Здесь “I” – “inphase” – синфазный сигнал, “Q” – “quadrature” – квадратура. Формулы перехода из RGB:

$$\begin{aligned} 3. \quad Y' &= 0.299R' + 0.587G' + 0.114B' \\ 4. \quad I &= -0.596R' - 0.275G' - 0.321B' \\ 5. \quad Q &= 0.212R' - 0.523G' - 0.311B' \end{aligned}$$

6. Photo YCC (торговая марка Eastman Kodak Company) – было разработано для кодирования изображения на носителях Photo CD. Целью было создания цветового пространства, независимого от устройства отображения. Формулы перехода из RGB:

$$\begin{aligned} 7. \quad Y &= 0.213R' + 0.419G' + 0.081B' \\ 8. \quad C1 &= -0.131R' - 0.256G' + 0.387B' + 156 \\ 9. \quad C2 &= 0.373R' - 0.312G' - 0.061B' + 137 \end{aligned}$$

3.2.2. Форматы семплирования

Формат 4:4:4 подразумевает, что все три компоненты (Y, Cb и Cr) имеют одинаковое разрешение и, следовательно, сэмплы (отсчеты) 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 (рис.3.1).



Рис.3.1. Семплирование 4:4:4(сохраняется по 4 пиксела яркостной и цветоразностных компонент)

При сэмплировании по формуле 4:2:2 (этот формат иногда обозначается YUY2) хроматические компоненты по вертикали имеют одинаковое разрешение с яркостью, а по горизонтали они имеют половину от разрешения яркости (рис.3.2). Числа 4:2:2 означают, что на каждые четыре сэмпла яркости Y по горизонтали отбирается только две компоненты Cb и две компоненты Cr. Формат 4:2:2 используется для высококачественного цветного видео.



Рис.3.2. Семплирование 4:2:2 (сохраняется 4 пиксела яркости и по две – цветоразных)

В популярном формате семплирования 4:2:0 (YV12) каждая компонента Cb и Cr имеет и по вертикали и по горизонтали половину разрешения по сравнению с Y(рис. 3.3)

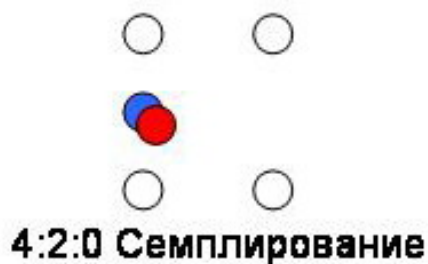


Рис.3.3. Семплирование 4:2:0 (Сохраняется 4 пиксела яркости и по одному – для цветоразностных компонент)

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 диски DVD. Поскольку хроматические компоненты отбираются в четыре раза реже компонент яркости, то пространство 4:2:0 YCbCr требует в два раза меньше сэмплов по сравнению с форматом видео 4:4:4 (или RGB).

Для того, чтобы провести анализ эффективности различных цветовых пространств и форматов сэмплирования было выполнено сжатие и последующее восстановление ряда различных по своему информационному смыслу изображений [9]. Тестирование проводилось на изображениях в формате “*.bmp” размером 1024 пикселей по длине и 768 пикселей по высоте. Реализовано несколько вариантов преобразования:

- цветовые пространства, в которые преобразовывалось исходное изображение - YUV, YCbCr, YCbCr (HDTV), Photo YCC и YIQ;
- режимы сэмплирования: режим 4:4:4 – при этом достигаются минимальные потери качества при меньшем коэффициенте сжатия и режим 4:2:0, при котором компоненты C_b и C_r берутся вдвое реже, чем яркостная составляющая, то есть через строку и через столбец, что обеспечивает больший коэффициент сжатия при худшем качестве.

Коэффициент качества, который используется в стандарте JPEG как входной параметр для формирования матрицы квантования (с помощью этого параметра происходит управление степенью потери качества сжатого изображения и коэффициентом сжатия), принимал значения от 1 до 10.

В результате сжатия и последующего восстановления изображений для указанных случаев определялось отношение сигнала к шуму по трем цветовым компонентам – R, G и B и максимальное отклонение значений пикселей.

На рис.3.4 показано изменение сигнала к шуму в зависимости от коэффициента качества, который используется для формирования матрицы квантования и с помощью которого происходит управление степенью потери качества сжатого изображения. На графике отображены зависимости для всех рассматриваемых в исследовании цветовых пространств.

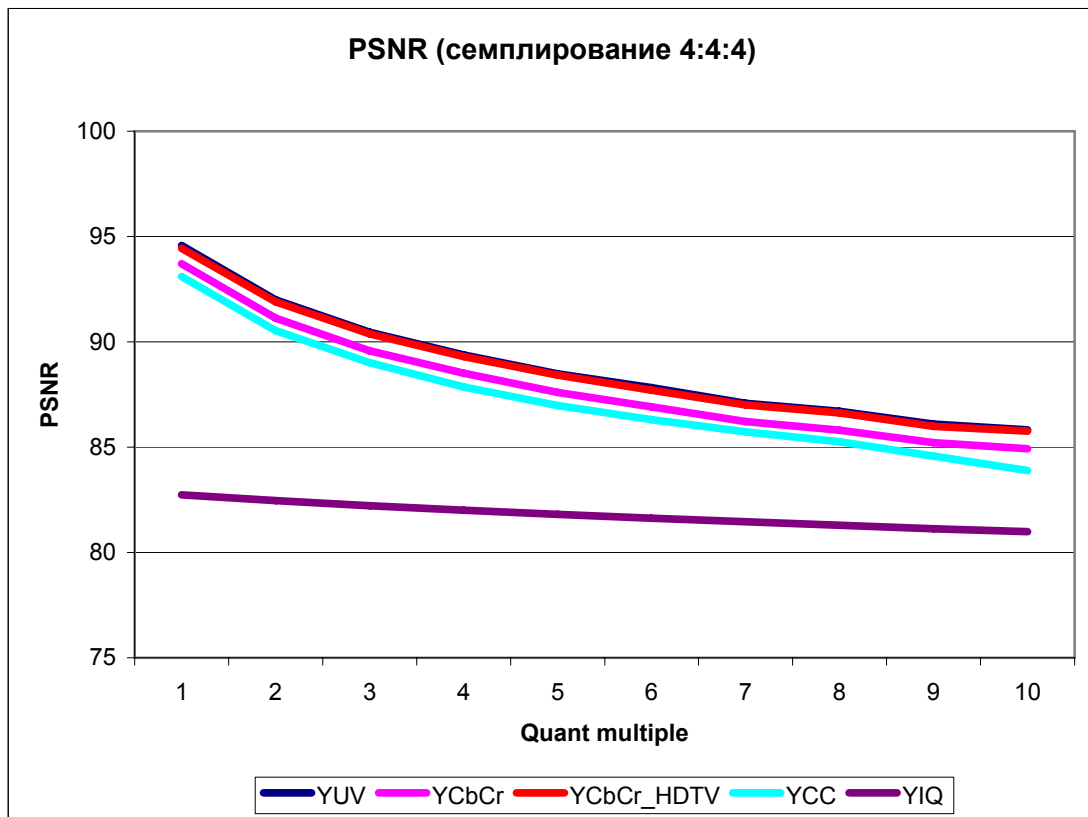


Рис. 3.4. Результаты сжатия при использовании семплирования 4:4:4.

Как видно из рис.3.4, при использовании семплирования 4:4:4, практически нет разницы между использованием цветовых пространств YUV и YCbCr (HDTV), их графики практически сливаются. Цветовое пространство YCbCr, используемое в алгоритме “JPEG” по умолчанию, показывает средние результаты, примерно на 1-2 децибела меньше, чем и YCbCr (HDTV). Далее по эффективности идет YCC. Наименьшие значения по значению отношения сигнала к шуму показало цветовое пространство YIQ, вероятно, это вызвано тем, что оно создавалось для специализированного применения, что и послужило причиной таких низких показателей, разрыв по отношению к другим пространствам составил в среднем около 8-10 децибел. Однако, значение максимального отклонения значений пикселей осталось в норме, оно даже не максимально по отношению к другим.

Однако, данный режим семплирования (4:4:4) используется лишь когда необходимо добиться минимальной потери данных, в ущерб коэффициенту сжатия. Гораздо чаще используется другой режим – 4:2:0 (YUV12), при использовании которого каждая компонента Cb и Cr имеет и по вертикали и по горизонтали половину разрешения по сравнению с Y.

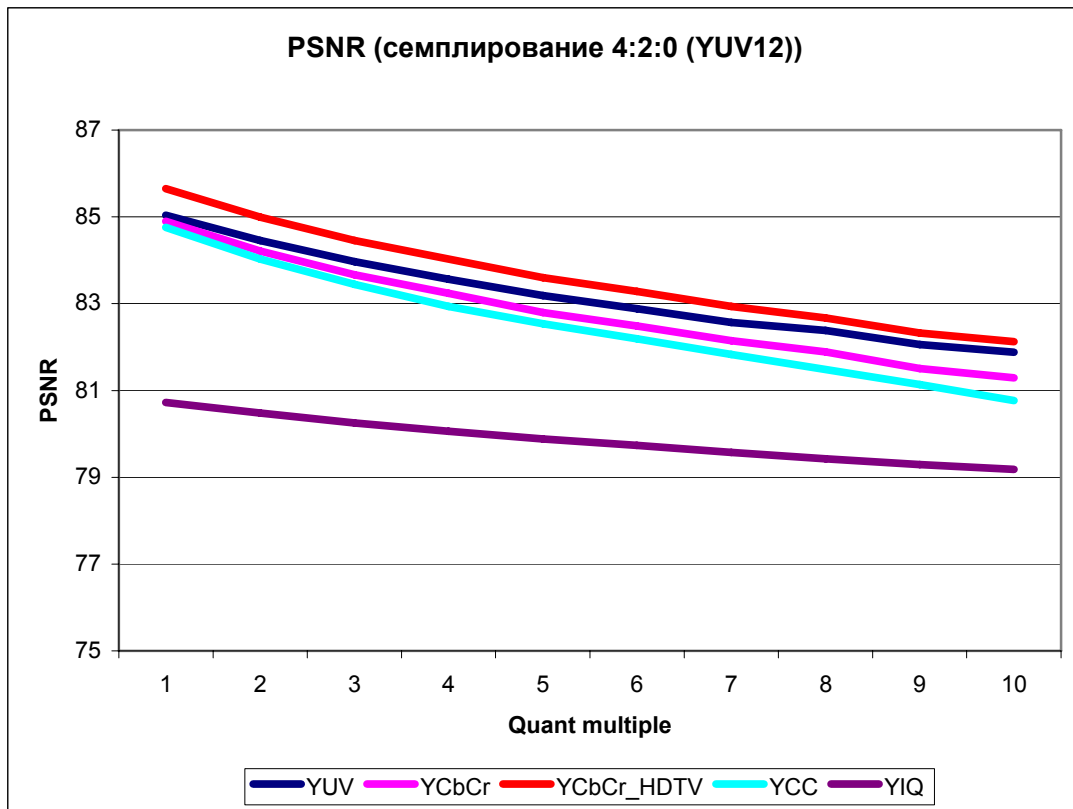


Рис. 3.5. Результаты сжатия при использовании семплирования 4:2:0.

В этом режиме семплирования наблюдается следующая картина (рис.3.5). Гораздо более хорошие результаты показало цветовое пространство YCbCr HDTV, которое в этом режиме опередило цветовое пространство YUV. Остальные цветовые пространства не изменили своих позиций, YCbCr, также находится на третьем месте.

Проанализировав их, следует отметить, что цветовое пространство YCbCr HDTV, ставшее лидером по общему значению отношения сигнала к шуму, показывает более плохие результаты по отношению к YUV по красной и синей цветовым компонентам, при этом резко вырываясь вперед по зеленой, особенно это ярко выражается в варианте семплирования 4:2:0, где отрыв от YUV составляет три децибела.

Значение максимального значения отклонений пикселей в режиме семплирования 4:4:4 наименьшее у цветового пространства YCbCr HDTV, на втором месте YCbCr, и наибольшее значение у цветового пространства YUV.

Однако, следует отметить, что максимальное отклонение значений пикселей изменяется крайне неравномерно. Особенно ярко это выражено в режиме семплирования 4:2:0.

На основе представленных результатов можно сделать следующие заключения относительно использования цветовых пространств в алгоритме JPEG.

В режиме семплирования 4:4:4 наилучшие результаты показали использование цветового пространства YUV и YCbCr (HDTV), которое дали примерно одинаковые, самые большие показатели отношения сигнала к шуму.

В режиме 4:2:0 использование цветового пространства YCbCr HDTV показало наилучшие результаты по значению отношения сигнала к шуму. Вместе с тем, использование YCbCr HDTV показало наименьшее отношение максимального отклонения значений пикселей.

3.2.3. Устранение пространственной статистической избыточности

Общая схема устранения пространственной статистической зависимости для стандартов MPEG и H.263/264 определяет процесс сжатия отдельного кадра видеопотока и совпадает с процедурой сжатия статического полноцветного изображения по стандарту JPEG (рис.1.4.) [6].

В начале изображение переводится из цветового пространства RGB в цветовое пространство YCrCb согласно описанной в предыдущем разделе модели преобразования цветовых пространств.

Затем изображение разбивается на блоки размером 8x8. Такие блоки создаются для каждой из цветовой составляющей. Для компонент Cr и Cb формирование блока происходит через строчку и через столбец, т.е. из исходного блока 16x16 получается один рабочий блок 8x8. В дальнейшем к каждому из блоков применяется дискретно косинусное преобразование, которое описывается выражением (1.13).

Для уменьшения хранимой информации производится квантование полученной матрицы частотных коэффициентов. Квантование представляет собой деление матрицы частотных коэффициентов на матрицу квантования. Для каждой из цветовых составляющих обычно используется своя матрица квантования $q[u,v]$:

$$Yq[u,v] = \text{IntegerRound} \left(\frac{Y[u,v]}{q[u,v]} \right) \quad (3.3)$$

Иногда для формирования матрицы квантования может использоваться специальная весовая функция, позволяющая сформировать коэффициенты квантования, обращающие в 0 наибольшее число высоко и средние частотных коэффициентов.

Этап квантования – это этап, на котором происходят самые большие потери в качестве изображения и этап, на котором осуществляется управление степенью сжатия изображения. При сильном квантовании низкочастотные коэффициенты могут настолько измениться, что изображение распадется на визуально различимые квадраты. Потери в высокочастотных коэффициентах приводят к появлению «нимба» вокруг контуров с резким переходом цвета.

Квантователь отображает числовой сигнал с областью значений X в квантованный сигнал области Y с уменьшенным числом значений. Это дает возможность представить квантованные величины с меньшим числом бит по сравнению с исходными неквантованными величинами.

Скалярный квантователь отображает один исходный сэмпл в одно квантованное значение на выходе. Простейшим примером такого квантователя может метод округления дробного числа до ближайшего целого. Это процесс с частичной потерей информации, так как не возможно установить точное значение исходного дробного числа по округленному целому. Квантование можно использовать для понижения точности изображения после применения кодирующего преобразования. Тем самым будет удалена часть незначимых величин и коэффициентов, оставляя малое число существенных ненулевых коэффициентов.

Векторный квантователь отображает группу исходных сэмплов в группу квантованных величин – кодовое слово, а декодер каждому кодовому слову сопоставляет некоторое приближение к исходному множеству данных. Множество векторов хранится кодером и декодером в специальной кодовой книге.

Проквантованная матрица затем преобразуется в вектор с помощью ZigZag сканирования (Рис.2.5).

После ZigZag сканирования низкочастотные элементы оказываются в «голове» вектора, а высокочастотные коэффициенты в «хвосте» вектора.

Далее выполняется так называемое вторичное сжатие с использованием энтропийного кодера.

Энтропийный кодер преобразует элементы последовательности символов, представляющих элементы видеопоследовательности, в сжатый поток битов, который можно сохранить в файле или передать по сетям связи. Входными символами могут являться квантованные коэффициенты преобразования, векторы движения, маркеры, хедеры и дополнительную информацию – данные, несущественные для корректного декодирования. Векторы движения обычно представляются в виде координат вектор перемещения всех блоков с компенсацией движения с целым или дробным пиксельным разрешением. Маркеры – это коды, обозначающие точки синхронизации в видеопоследовательности. Хедеры – заголовки макроблоков, изображений, последовательностей и других объектов.

Полученный вектор затем кодируется методом RLE. А полученные пары кодируются по Хаффману с фиксированной таблицей.

Процесс восстановления изображения полностью симметричен процессу сжатия изображения.

3.3. Модель DPCM/DCT видеокодека

Большинство современных стандартов видеокодирования основаны на единой модели видео кодека, которая включает модуль оценки движения и компенсации высоких частот (обычно обозначается сокращением DPCM), модуль преобразования (чаще всего косинусного – DCT) и модуль энтропийного кодирования. По схожему принципу строятся кодеки , совместимые со стандартами H.261, H.263, MPEG-1, MPEG-2? MPEG-4 и H.264.

Однако внутри указанных модулей имеется существенное различие по реализуемым функциям.

На рис. 3.6 приведена общая схема подобного кодера DPCM/DCT. Видеокодер обрабатывает кадр F_n и производит закодированный (сжатый) видеопоток [6].

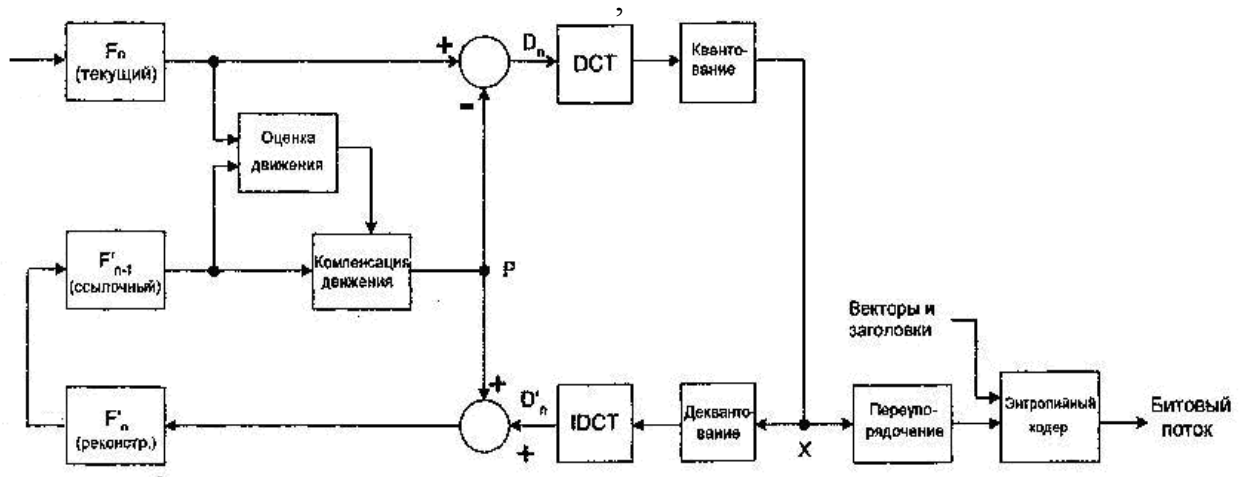


Рис.3.6. Кодер DPCM/DCT

Процесс кодирования состоит в следующем.

1. Входной кадр F_n подается на вход кодера и обрабатывается там макроблоками (например, размером 16x16 или 8x8 отсчетов).

2. Кадр F_n сравнивается с ссылочным кадром, например с ранее закодированным кадром F_{n-1} . Функция оценки движения находит в F_{n-1} область, соответствующую макроблоку, которая больше всего схожа с текущим макроблоком в F_n . Смещение текущего макроблока по отношению к положению выбранного ссылочного макроблока записывается в вектор движения MV.

3. На основе выбранного вектора движения MV строится прогноз компенсированного движения P, т.е. область, где может оказаться макроблок.

4. Макроблок P вычитается из текущего макроблока и их разность D является остаточным макроблоком.

5. Макроблок D преобразуется с помощью DCT. Обычно макроблок D делится на подблоки размером 8x8 или 4x4 пиксела и каждый подблок преобразуется отдельно.

6. Каждый подблок квантуется.

7. Квантованные коэффициенты DCT всех подблоков переупорядочиваются и затем с использованием RLE и LZW кодирования.

8. Коэффициенты, векторы движения и связанные с ними заголовки каждого макроблока кодируются энтропийным кодером с использованием оптимальных кодов переменной длины Хаффмана или арифметическим кодированием для получения выходного сжатого битового потока.

На рис. 3.7 приведена схема DPCM/DCT декодера [6].

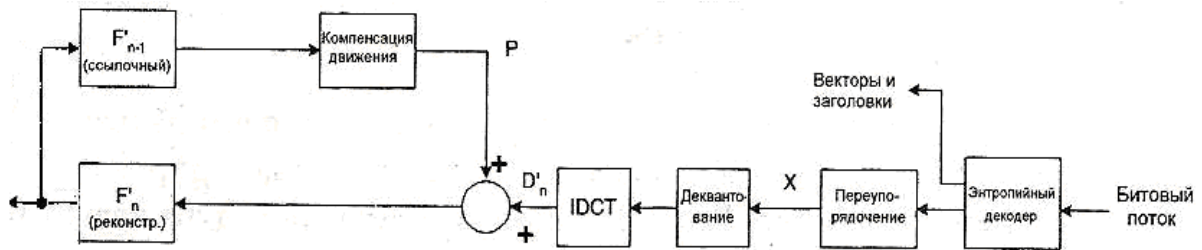


Рис.3.7 Декодер DPCM/DCT

Процесс декодирования сводится к следующему.

1. Сжатый видеопоток битов декодируется энтропийным декодером для извлечения из него коэффициентов DCT, векторов движения и заголовков каждого из макроблоков.

2. Обращается RLE и LZW кодирование и делается обратное упорядочение коэффициентов.

3 Макроблок деквантуется и к результату применяется обратное DCT, что порождает остаточный макроблок **D**.

4. Декодированный вектор движения используется для нахождения местоположения области размером 16x16 на предыдущем (ссылочном) кадре F_{n-1} , копия которого хранится декодером. Эта область становится прогнозом с компенсацией движения **P**.

5. Макроблок **P** прибавляется к **D** и получается восстановленный макроблок, который сохраняется на своей позиции на декодированном кадре F_n .

После полного декодирования такой кадр готов для воспроизведения и, кроме того, он может сохраняться в качестве ссылочного для декодирования следующего кадра.

Для того чтобы удовлетворить противоречивым требованиям и увеличить гибкость алгоритма, рассматривается 4 типа кадров:

- I-кадры - кадры, сжатые независимо от других кадров (I - Intra pictures);
- P-кадры- сжатые с использованием ссылки на одно изображение (P - Predicted);
- В-кадры- сжатые с использованием ссылки на два изображения (B - Bidirection), что, однако, возможно лишь при двухпроходном кодировании;
- DC-кадры - независимо сжатые с большой потерей качества (используются только при быстром поиске).

I-кадры обеспечивают возможность произвольного доступа к любому кадру, являясь своеобразными входными точками в поток данных для декодера.

P-кадры используют при архивации ссылку на один I- или P-кадр, повышая тем самым степень сжатия фильма в целом.

В-кадры, используя ссылки на два кадра, находящиеся впереди и позади, обеспечивают наивысшую степень сжатия. Сами в качестве ссылки использоваться не могут.

Последовательность кадров в фильме может быть, например, такой: как

показано на рис. 3.8.

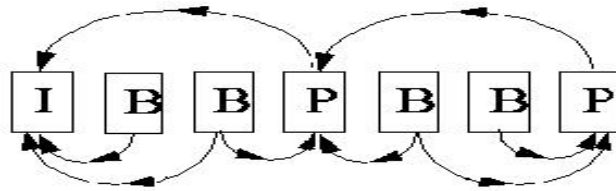


Рис.3.8. Последовательность кадров разного типа во времени

Поскольку изображение на соседних кадрах обычно сдвинуто, применяется компенсация движения, то есть кодируется разность от некоторого сдвинутого опорного изображения. Кодирование выполняется по макроблокам (16 x 16 яркость, 8 x 8 цветность), для каждого макроблока находится свой вектор движения.

Частота I-кадров выбирается в зависимости от требований на время произвольного доступа и надежности потока при передаче через канал с ошибками. Соотношение P- и B-кадров подбирается, исходя из требований к величине компрессии и ограничений декодеру. Как правило, декодирование B-кадров требует больше вычислительных мощностей, однако позволяет повысить степень сжатия. Именно варьирование частоты кадров разных типов обеспечивает алгоритму необходимую гибкость и возможность расширения. Понятно, что для того, чтобы распаковать B-кадр, мы должны уже распаковать те кадры, на которые он ссылается. Поэтому для последовательности IBVRBVRBVRBVRBVRBVR кадры в фильме будут записаны так: 0**312645..., где цифры - номера кадров, а звездочкам соответствуют либо B-кадры с номерами -1 и -2, если мы находимся в середине потока, либо пустые кадры (ничего), если мы в начале фильма. Подобный формат обладает достаточно большой гибкостью и способен удовлетворять самым различным наборам требований.

На практике широко используется метод компенсации движения, который компенсирует перемещение прямоугольных областей текущего кадра. При этом для каждого фиксированного блока, состоящего из $M \times N$ пикселей обрабатываемого кадра, выполняется следующая процедура [6]:

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 блоками $M \times N$ области поиска. Область поиска обычно представляет собой некий регион с центром в середине этого выбранного блока. Популярным критерием схожести блоков является энергия остатка, получаемая вычитанием блока-кандидата из фиксированного $M \times N$ блока, то есть 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 необходимо вычитать вычесть из этого блока для получения остаточного $M*N$ -блока.

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

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

Преимущества компенсации движения на основе блоков:

1. Метод достаточно прост и легко поддается программной реализации.
2. Метод хорошо вписывается в прямоугольный формат кадров, а также в процедуры преобразования изображений на основе блоков.
3. Обеспечивается достаточно приемлемая временная модель для многих видеопоследовательностей.

Недостатки компенсации движения на основе блоков:

1. Реальные объекты редко имеют четкие прямоугольные границы.
2. Объект часто перемещаются на нецелое число пиксельных позиций между кадрами.
3. Многие методы типы перемещений трудно скомпенсировать с помощью методов на основе блоков. К таким типам относятся деформация объектов, изменение масштаба, сложные движения, такие как клубы дыма.

Несмотря на перечисленные недостатки, компенсация движения на основе блоков является базой временных моделей во всех современных стандартах цифрового кодирования.

Макроблок, соответствующий области $16*16$ пикселей в кадре, является базисной единицей для прогноза с компенсацией движения во многих важных визуальных кодовых стандартах.

Оценка движения макроблока состоит из поиска области пикселей $16*16$ на ссылочном кадре, который похож на текущий макроблок. Ссылочным кадром может является уже декодированный кадр, который хронологически может идти раньше или позже текущего. Область поиска в 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.

Выбранный наилучший макроблок на ссылочном кадре вычитается из текущего блока, и их разность, называемая остаточным макроблоком, кодируется и передается вместе с вектором движения, задающим позицию наилучшего макроблока по отношению к текущему макроблоку.

Движущиеся объекты на видеосцене редко следуют строго вдоль границ блоков пикселей $16*16$, поэтому более эффективным представляется использование переменных размеров блоков для оценки и компенсации движения. Объекты могут перемещаться на нецелое число пикселей между кадрами. Поэтому лучший прогноз можно построить интерполяцией ссылочного кадра по позициям подпикселей перед поиском наилучшего

положения прогноза.

Результат компенсации движения улучшается при уменьшении размера блока компенсации. Однако уменьшение этого размера влечет за собой увеличение сложности 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 изображения, то есть можно выбирать большие размеры блоков для равных 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 размера блоков компенсации движения.

Объекты, перемещающиеся в естественных видеосценах, редко бывают выровнены по границам прямоугольных блоков. Они могут находиться в любом месте кадра, и иногда менять свою форму от кадра к кадру. Можно было бы добиться лучших результатов с помощью компенсации движения на основе областей произвольной формы. Однако при таком подходе возникают новые практические трудности, такие, как необходимость точного и достоверного описания границ области, сегментации и кодирования контура границы объекта для декодера, кодирование остатка после компенсации движения и многие другие. Стандарт MPEG-4 включает в себя ряд инструментов, реализующих компенсацию движения на основе областей.

3.4. Анализ возможностей современных методов сжатия видеоданных

Для различных сфер использования цифрового видео выдвигались разные требования к сжатию, которые привели к формированию ряда стандартов сжатия с различными областями применения:

- необходимость сжатия для видеоконференций привело к возникновению стандартов ITU H.261 для ISDN-видеоконференций, для видеоконференций в телефонных сетях и H.263 для видеоконференций в сетях ATM и по широкополосным каналам;
- необходимость сжатия видеопоследовательностей для хранения на CD-ROM (с условием обеспечения 1.2 Мбит/с для видео-потока и 256 кбит/с для аудио) привело к возникновению первоначального стандарта ISO MPEG-1;
- для вещания и хранения на DVD, с битрейтом от 2 до 15 Мбит/с для видео и аудио, был разработан стандарт ISO MPEG-2;
- необходимость кодирования отдельных аудио-визуальных объектов как естественного происхождения, так и синтезированных, привело к созданию ISO MPEG-4. Этот стандарт включает в себя несколько частей, в которых рассматривается, помимо кодирования видео, аудиокодирование, кодирование объектов и т.д. К видео относятся части 2 (ISO 14496-2 или MPEG-4 Part 2) и 10 (ISO 14496-10 или MPEG-4 Part 10);
- необходимость кодирования метаданных, описывающих свойства

мультимедийных данных, привело к появлению MPEG-7;

Детальное рассмотрение современных стандартов приводится в [3,6,8]. Остановимся на части 10 стандарта MPEG-4. У нее есть другие названия - H.264 и AVC – Advanced Video Coding. Подробно MPEG-4 Part 10 и H.264 рассматриваются в [6]. Часть 10 стандарта MPEG-4 определяет один из самых современных и технически совершенных методов видео кодирования. Стандарт AVC/H.264 был разработан Joint Video Team (JVT), которая включает экспертов из MPEG и VCEG (Video Coding Experts Group). Под этим названием формат уже широко известен. «Официальное» наименование нового стандарта, Advanced Video Coding (AVC), было выбрано MPEG как видео дополнение к аудио формату Advanced Audio Coding (AAC).

В AVC/H.264 определены следующие профили: базовый (baseline), основной (main), расширенный (extended). Позднее расширенный профиль был дополнен профилем для видео высокого разрешения (high profile), профилем high 10, профилем high 4:2:2 и профилем high 4:4:4.

- базовый профиль нацелен на кодирование и декодирование в реальном времени для мобильных устройств. Он поддерживает прогрессивную развертку, использует I- и P-кадры, а также энтропийное кодирование по методу CAVLC;

- основной профиль предназначен в основном для использования в ширококовещании. Он поддерживает чересстрочную и прогрессивную развертки, использует I-, P-, B-кадры, весовое предсказание (weighted prediction), а также энтропийное кодирование по методам CAVLC и CABAC;

- расширенный профиль предназначен для использования в средствах передачи, подверженных ошибкам – например, в мобильных коммуникациях. Использует I-, P-, B-, SP-, SI-кадры, поддерживает как чересстрочную, так и прогрессивную развертку, позволяет использовать только метод CAVLC для энтропийного кодирования;

- профиль для видео высокого разрешения предназначен для эффективного кодирования HDV (high definition video). Он использует адаптивный размер блока (8x8 или 4x4) и позволяет применять контекстно-зависимые матрицы квантования;

- профиль high 10 является расширением предыдущего профиля для 10 бит на отсчет компоненты изображения;

- профиль high 4:2:2 поддерживает формат YUV 4:2:2 и до 10 бит на отсчет для цветоразностных компонент изображения;

- профиль high 4:4:4 поддерживает формат YUV 4:4:4 и до 12 бит на отсчет цветоразностных компонент изображения. Помимо этого он позволяет использовать режим кодирования без потерь и прямое кодирование RGB сигнала. Этот профиль предназначен для кодирования видео студийного качества.

Кодирование осуществляется поблочно. При этом сначала производится предсказание отсчетов яркостной компоненты и цветоразностных компонент в пространственной и временной областях. Затем разность между предсказанными значениями и реальными подвергается целочисленному

преобразованию и квантуется. После этого результат сжимается энтропийным кодером. Обработка каждого кадра ведется в пространстве YUV по блокам размером 16x16 для яркостной компоненты (luma) и по 8x8 (для YUV 4:2:2) для цветоразностных компонент (chroma).

Целочисленное преобразование осуществляется над блоками размером 4x4. Это преобразование обладает схожими свойствами с дискретным косинусным преобразованием, но отличается тем, что в нем используется целочисленная арифметика. Это позволяет добиться повышения скорости работы кодера и декодера. Полученные коэффициенты с помощью зигзаг-сканирования выстраиваются в вектор, который квантуется и подвергается энтропийному сжатию. Помимо этого преобразования в стандарте заложена возможность использования wavelet-преобразований.

AVC/H.264 определяет два инструмента для энтропийного кодирования битового потока – контекстное адаптивное кодирование с переменной длиной (Context-Adaptive Variable Length Coding, CAVLC) и контекстное адаптивное бинарное арифметическое кодирование (Context-Adaptive Binary Arithmetic Coding, CABAC). В CAVLC количество коэффициентов, не равных 0, кодируется отдельно от их значения и положения в векторе. После зигзаг-сканирования коэффициентов преобразования большие значения коэффициентов, соответствующие низкочастотной составляющей, располагаются в начале вектора, а меньшие (соответствующие высокочастотной части) – в конце. В [6] работа метода CAVLC рассмотрена на следующем типичном векторе коэффициентов:

7 6 -2 0 -1 0 0 1 0 0 0 0 0 0 0

Для передачи информации об этом векторе используются следующие данные:

- Количество ненулевых коэффициентов (N) и количество коэффициентов, равных единице по модулю, располагающихся после последнего ненулевого элемента, отлично по модулю от единицы (T1). Для данного примера $T1 = 2$ (коэффициенты -1 и 1), а $N = 5$ (коэффициенты 7, 6, -2, -1, 1).
- 2) Закодированные значения коэффициентов. При этом для ненулевых элементов, равных по модулю 1, необходимо сохранять только их знак. Коэффициенты кодируются в обратном порядке, т.е. первым закодированным коэффициентом в данном случае будет -2. Для кодирования используются 6 таблиц VLC (Variable Length Codes, коды переменной длины) Голомба. Для кодирования первого коэффициента используется стартовая таблица. Контекстная адаптивность алгоритма заключается в возможности сменить таблицу для кодирования каждого следующего коэффициента в зависимости от предыдущих коэффициентов.
- 3) Информация о знаках. Для кодирования знака используется 1 бит. Для кодирования единичных коэффициентов кроме этого бита никакой другой информации не требуется, а для других коэффициентов знаковый бит включается в коды Голомба. Местонахождение каждого ненулевого

коэффициента кодируется путем указания позиций нулей перед последним ненулевым коэффициентом. Эта информация разбивается на 2 части:

- 4) Общее количество нулей. Это число определяет количество нулей между последним ненулевым коэффициентом вектора и его начальным элементом. Для данного примера это число равно 3. Так как уже известно, что число ненулевых коэффициентов в векторе (N) равно 5, то это число должно быть в пределах [0,11]. Для N в пределах от 1 до 15 доступно 15 таблиц (???). N, равное 16, означает, что в векторе нет нулевых коэффициентов.
- 5) Расположение нулей в векторе. В данном примере нужно обозначить положение 3 нулей. Сначала кодируется количество нулей перед последним ненулевым коэффициентом (в примере – 2). Это число должно находиться в пределах [0,3], поэтому используется соответствующая таблица VLC. Осталось закодировать положение последнего нуля. Количество нулей перед предпоследним ненулевым коэффициентом должно быть в пределах [0,1]. В примере это число равно 1. Больше нулей нет, поэтому кодирование заканчивается.

Эффективность энтропийного кодирования повышена путем использования контекстно-адаптивного двоичного арифметического кодирования (СABAC). Использование арифметического кодирования допускает использование нецелого количества бит на символ алфавита, что особенно хорошо для символов с вероятностью больше 0,5. Важной особенностью САВАС является контекстное моделирование. Статистические данные об уже закодированных символах используются для оценки вероятностей символов. Эти данные используются для переключения между несколькими моделями для оценки вероятности. В H.264/AVC арифметический кодер представляет собой совокупность процедур низкой сложности, в которых отсутствуют операции умножения. Процедуры включают в себя сдвиги и обращения к таблицам. Использование САВАС позволяет уменьшить в среднем битрейт на 10-15%. Наибольший выигрыш получается обычно при обработке чересстрочных сигналов ТВ. В табл. 3.3 приводится сравнение упомянутых выше стандартов сжатия и используемых в них алгоритмах сжатия [8].

Таблица 3.3

Название стандарта	Область применения	Первичный алгоритм	Вторичный алгоритм	Разрешения	Поток	Недостатки и достоинства
H.261	ISDN-	DCT	VLC	352x288	0,04-2	Достоинства: прост в

	видеоконференции, аппаратные кодеки	(блоки 8x8) и квантование межкадровой разности	(метод Хаффмана)	x 30 176x144 x 30	Мбит/с (px64 Кбит/с, где p от 1 до 30)	апп. реализации. Недостатки: низкая степень сжатия, плохая компенсация движения (алгоритм рассчитан на видео с небольшим количеством движения, напр. голова собеседника)
H.263	Для видеоконференций в сетях ATM и по широкополосным каналам	DCT (блоки 8x8) и квантование	Арифметическое кодирование	Sub-QCIF, QCIF, CIF, 4CIF, 16CIF, отдельно настраиваемые разрешения	0,04-2 Мбит/с (px64 Кбит/с, где p от 1 до 30)	Достоинства: улучшен алгоритм компенсации движения по сравнению с H.261, более эффективный вторичный алгоритм. Недостатки: находится между MPEG-2 и MPEG-4 по количеству заложенных идей
ISO MPEG-1	хранение видеопоследовательностей на CD-ROM	DCT (блоки 8x8) и квантование	Коды переменной длины (Хаффман)	352x240 x 30 352x288 x 25	1,5 Мбит/с	Достоинства: прост в апп. реализации. Недостатки: низкая степень сжатия, недостаточная гибкость формата.
ISO MPEG-2	хранения на DVD			Универсальный	3-15 Мбит/с	Достоинства: сравнительная простота апп. реализации. Недостатки: недостаточная степень сжатия, малая гибкость.
MPEG-4 Part 10 (AVC, H.264)	кодирование отдельных аудио-визуальных объектов, как естественного происхождения, так и синтезированных	Целочисленное DCT (блоки 4x4), либо DWT и квантование межкадровой разности	CAVLC или CABAC	Универсальный	0,0048-20 Мбит/с	Достоинства: высокая степень универсальности, объектно-ориентированная работа с потоком данных. Недостатки: высокая сложность реализации

MPEG-4, будучи последним по времени появления стандартом, использует наиболее эффективные алгоритмы сжатия данных, например алгоритм арифметического кодирования для энтропийного кодирования и вейвлет преобразования в качестве первичного алгоритма. Наиболее перспективными сейчас представляются следующие направления развития:

- использование альтернативных методов сжатия без потерь, такие как различные реализации арифметического кодирования (напр., CABAC), цепные коды и т.п.

- использование альтернативных преобразований: вейвлет-преобразований (DWT), фрактальных преобразований и выделения объектов с их последующим

описанием.

Важнейшими характеристиками при сравнении различных методов кодирования видео сигналов является выходной битрейт и оценки качества восстановленного после декодирования видео потока. Сведения о битрейте выходного потока для различных методов кодирования приведены в таблице 2.1.

Одним из простых критерием оценки потери качества является среднеквадратическое отклонение значений пикселей сжатого изображения от оригинала.

По этому критерию изображение будет сильно испорчено при изменении яркости всего на 5%. В тоже время изображение со снегом, резким изменением цвета отдельных точек будут признаны почти не изменившимися.

Другим критерием является максимальное отклонение от оригинала.. Даная мера крайне чувствительна к биению отдельных пикселей, т.е. в изображении может измениться только один пиксель, и данный критерий признает изображение сильно испорченным.

На практике используемой мерой качества изображения является критерий соотношения сигнал/шум (PSNR). Эта мера аналогична среднеквадратическому отклонению, но пользоваться ей удобнее из-за логарифмического масштаба шкалы.

Лучше всего потери в качестве оценивает человеческий глаз. Сжатие изображение можно считать отличной, если на глаз невозможно отличить оригинал от сжатого изображения. Но на практике при сжатии с потерями в изображение всегда вносятся какие-либо искажения заметные при сравнении оригинала и сжатого изображения.

Среди большого числа критериев оценки качества восстановленного изображения наибольшее распространение получили среднеквадратическое отклонение и соотношение сигнал/шум, а для визуальной оценки используется разностное изображение.

На практике, видео изображение со значениями PSNR порядка 40-43 дБ и выше является изображением очень высокого качества. Сжатое изображение с уровнем PSNR свыше 43 дБ в связи с особенностями человеческого зрения неотлично от оригинала, с уровнем от 40 до 43 дБ показывает отличное качество изображения, с уровнем PSNR от 35 до 40 дБ – хорошее, от 30 до 35 дБ - приемлемое качество изображения, с уровнем PSNR ниже 30 дБ – неприемлемое качество изображения [3].

Основные задачи сравнительного анализа алгоритмов и программных моделей кодирования потокового видео:

- сравнение результатов работы кодека, работающего на основе стандарта Mpeg-4 Part 10, при разных значениях параметров, связанных с временной моделью;

- сравнение результаты работы кодеков, работающих на основе стандартов Mpeg-4 Part 2 и Mpeg-4 Part 10, а также более раннего стандарта H.263+.

Сложность и дороговизна субъективных методов оценивания качества делают привлекательным автоматическое измерение качества с помощью определенных алгоритмов [3]. Для количественной характеристики качества существуют различные методы, среди которых наибольшее распространение получили точные с математической точки зрения критерии оценки качества изображения на основе оценки отношения энергии сигнала и искажений (см. раздел 1.5).

В качестве объекта для изучения влияния параметров на сжатие был выбран кодек на основе стандарта Mpeg-4 Part 10, так как этот стандарт является более совершенным по сравнению с предыдущими стандартами, и реализует самые последние наработки в области сжатия прямоугольных кадров видеоизображений.

Тестирование состояло в сравнении качества сжатия по различным стандартам. При этом:

- Сжатие кодеком x264 производилось с помощью программы MeGUI Modern Media Encoder 0.2.6.1040;
- Сжатие в других форматах производилось с помощью программы MediaCoder 0.6.0 входящим в ее состав кодеком ffmpeg;
- Исходные видеопоследовательности создавались с помощью программы VirtualDub 1.7.6;
- Все настройки, кроме битрейта, брались по умолчанию;
- Кодекам задавались следующие значения битрейта (Кбит/сек): 200, 300, 700, 1000.
- Подсчет PSNR производился с помощью программы MSU Video Quality Measurement Tool 1.52.

В качестве исходного видеоматериала использовались несжатые видеопоследовательности. Для тестирования были выбраны четыре сцены из фильма «Космическая одиссея 2001». Из них две сцены – с очень быстрым движением, одна – со средним и одна – с малым, так как именно искажения при кодирования движения являются наиболее значимым фактором, влияющим на качество сжатого видео. Также были выбраны четыре битрейта – 100, 300, 700 и 1000 Кбит/сек. 100 и 300 Кбит/сек – типичные битрейты для сжатия потокового видео, 700 и 1000 Кбит/сек – фильм со средним и высоким качеством на CD.

Последовательность №1. Flight

Разрешение	640x272
Цветовое пространство	RGB
Частота кадров	25

Источник	несжатое видео
Количество кадров	1449
Особенности	Последовательность является отрывком из фильма “Космическая одиссея 2001”. В кадре происходит постоянное быстрое поступательное движение камеры сквозь непрерывно изменяющиеся световые потоки. Цвет и узор световых потоков быстро изменяется. Также изменяется яркость. При этом в сцене отсутствуют мелкие детали, что несколько облегчает задачу сжатия.

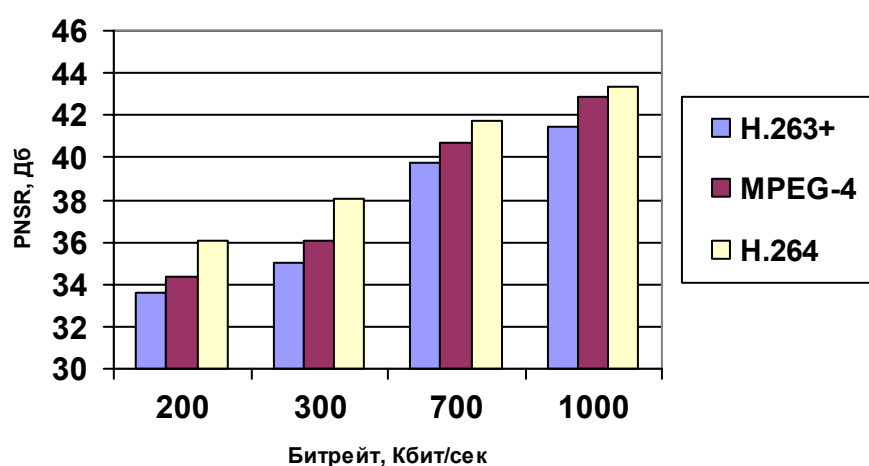


Рис. 3.9. Зависимость качества передачи яркости от битрейта

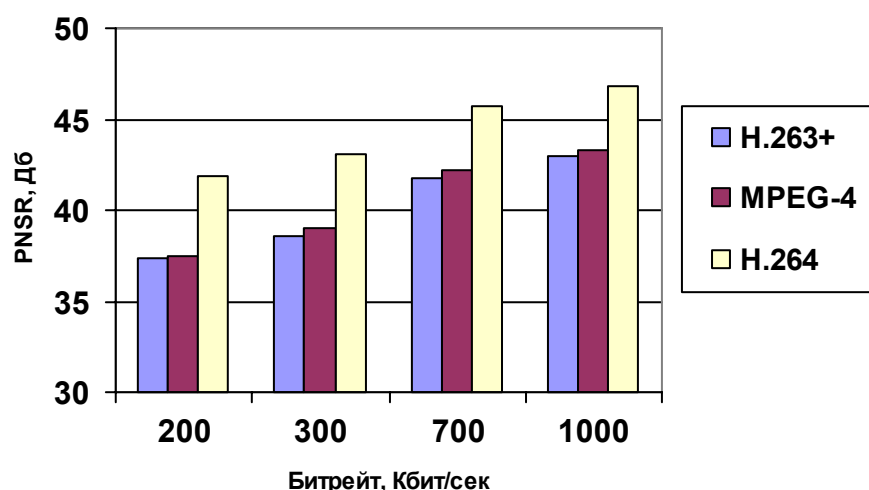


Рис. 3.10. Зависимость качества цветопередачи от битрейта

Сравнение показало превосходство кодека H.264 на всех тестах. Превосходство особенно ярко проявляется на низких значениях битрейта. При этом превосходство по качеству цветопередачи значительно превосходит превосходства по передаче яркости. Кодек на основе более раннего стандарта H.263+ ожидаемо проигрывает своим конкурентам на всех битрейтах.

Последовательность №2. Monkeys

Разрешение	640x272
Цветовое пространство	RGB
Частота кадров	25
Источник	несжатое видео
Длительность	1 мин 0,4 сек
Количество кадров	1447
Особенности	Последовательность является отрывком из фильма “Космическая одиссея 2001”. Очень сложная для компенсации движения сцена. В кадре присутствует множество мелких деталей и постоянно движущихся небольших объектов. Также присутствует небольшое вращение камеры.

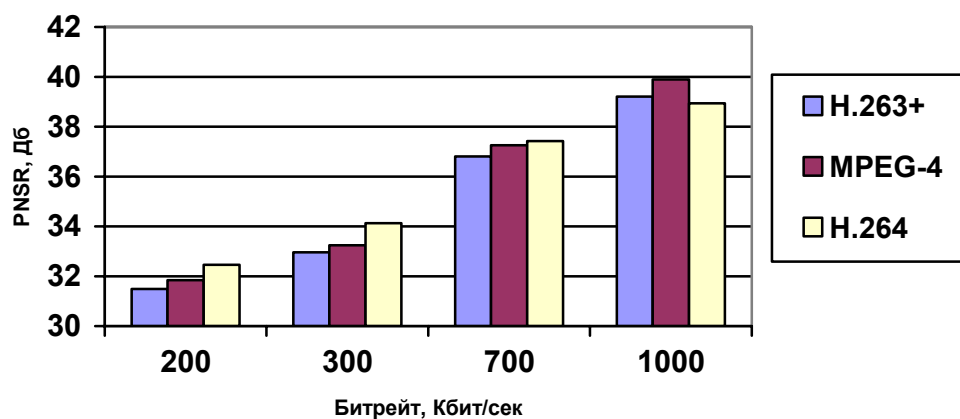


Рис. 3.11. Зависимость качества передачи яркости от битрейта

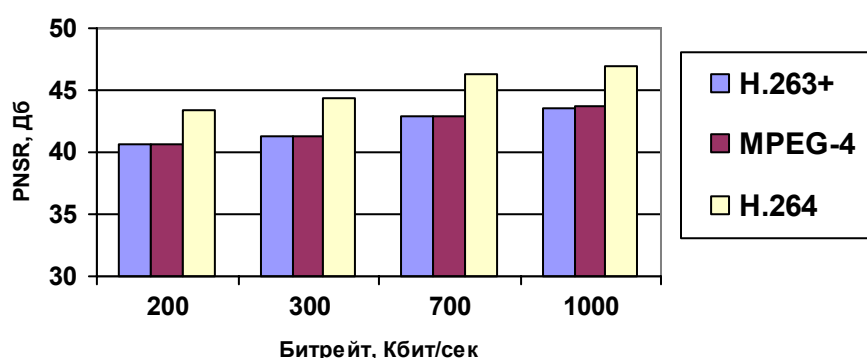


Рис. 3.12. Зависимость качества цветопередачи от битрейта

Кодек H.264 по-прежнему удерживает абсолютное лидерство на низких битрейтах и в качестве цветопередаче, но неожиданно уступил другим кодекам при битрейте 1000 Кбит/сек в передаче яркости.

Последовательность №3. Discussion

Разрешение	640x272
Цветовое пространство	RGB
Частота кадров	25
Источник	несжатое видео
Длительность	1 мин
Количество кадров	1438
Особенности	Последовательность является отрывком из фильма “Космическая одиссея 2001”. Последовательность простая для сжатия – сцена почти статическая, за исключением лиц и рук говорящих персонажей. Яркость также не изменяется.

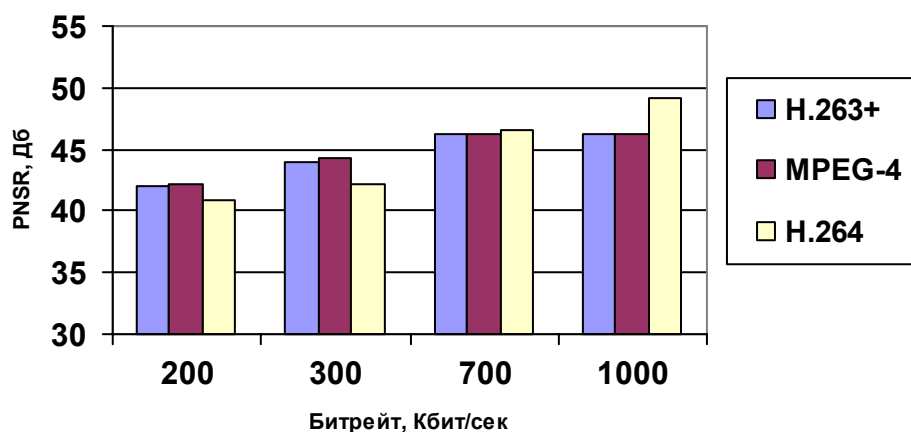


Рис.3.13. Зависимость качества передачи яркости от битрейта

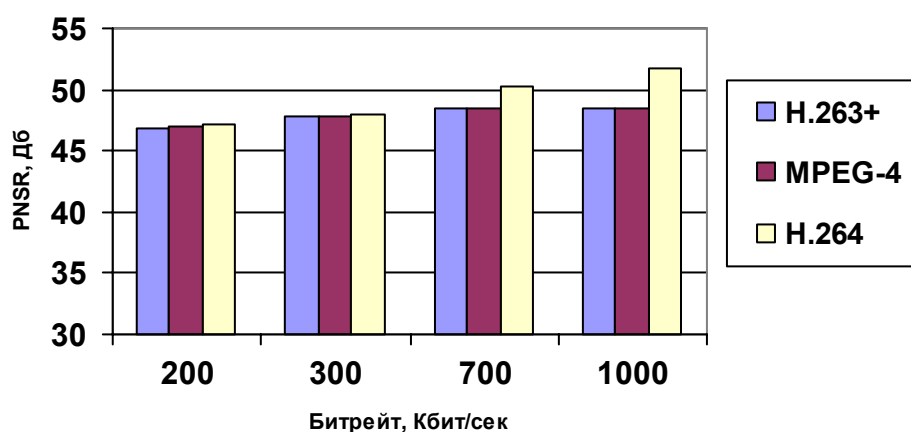


Рис.3.14. Зависимость качества цветопередачи от битрейта

Кодек H.264 уступил другим на низких битрейтах, но превзошел на высоких. В целом, кодеки показали приблизительно равные результаты. Это объясняется статичностью данной сцены – для ее эффективного сжатия нет необходимости использования сложных алгоритмов.

Последовательность №4. Space ship

Разрешение	640x272
Цветовое пространство	RGB
Частота кадров	25
Источник	несжатое видео
Длительность	51 сек
Количество кадров	1224
Особенности	Последовательность является отрывком из фильма “Космическая одиссея 2001”. Средняя по сложности компенсации движения последовательность. В кадре присутствуют плавные медленные движения объектов и камеры. Перепадов яркости нет. В сцене присутствует множество мелких деталей.

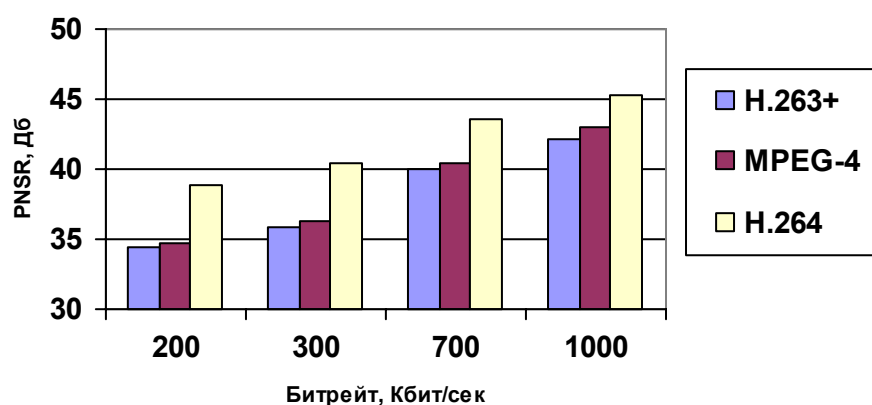


Рис.3.15. Зависимость качества передачи яркости от битрейта

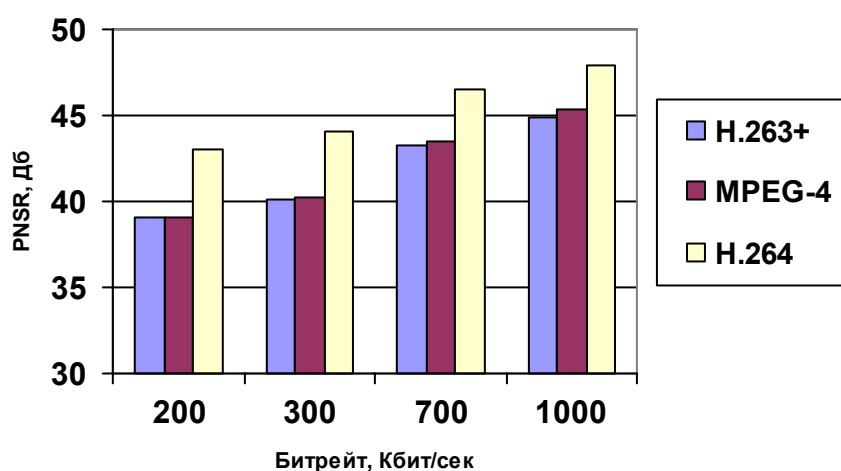
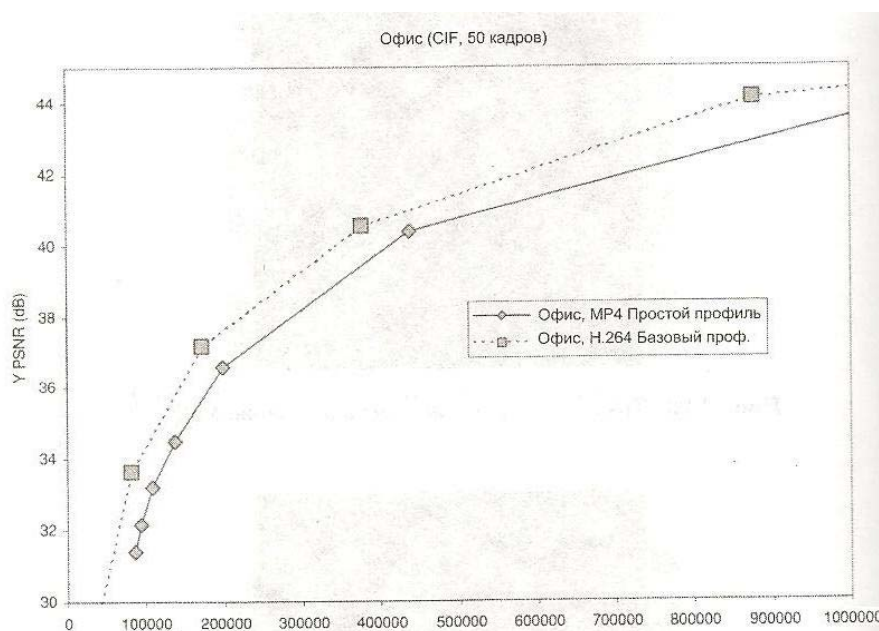


Рис.3.16 Зависимость качества цветопередачи от битрейта

Тесты показали превосходство кодера на основе стандарта H.264 на всех битрейтах. Кодер на основе MPEG-4 занял второе место, лишь немного опередив кодек H.263+. На рис. 3.17, а) и рис. 3.17, б) приведены зависимости, характеризующие производительность стандартов MPEG-4 (Part2) (простой профиль) и H.264 (базовый профиль) при одном ссылочном кадре для видеопоследовательностей «офис» (аналогичен **Discussion**) и «травы» (пейзажный видеоряд) соответственно [6].

а)



б)

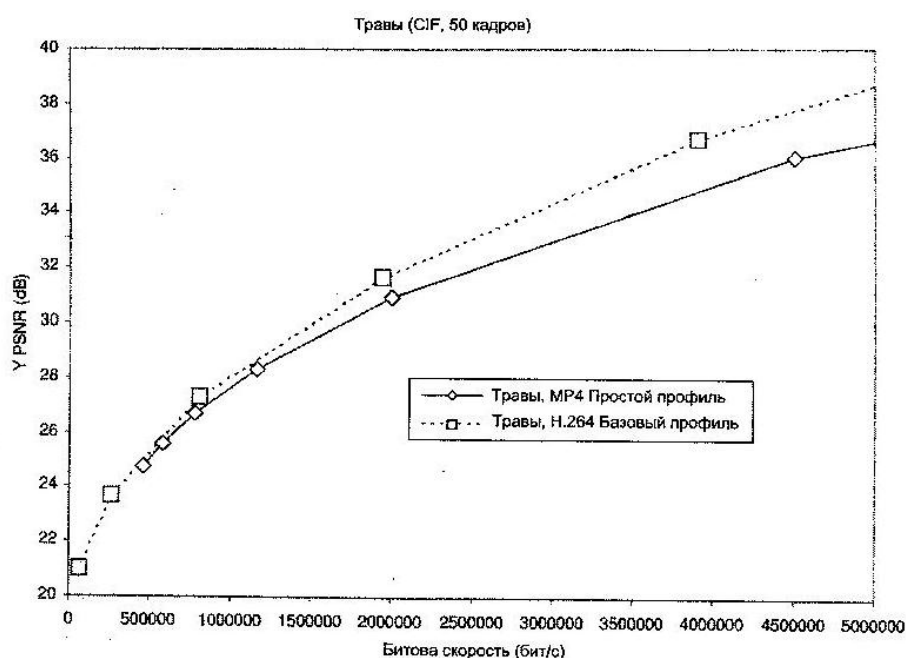


Рис. 3.17. Зависимость уровня искажений от битрейта, размер кадра CIF (352x288 пикселей яркости, семплирование 4:2:0)

Выводы. Как видно из графиков, на разных последовательностях кодеки

показывают разные результаты. Поскольку подобраны разные последовательности, это позволяет оценить эффективность работы кодеков с различным исходным материалом.

Результат сравнения кодеков разных стандартов показал явное преимущество кодека на основе стандарта H.264 над остальными кодеками. Особенно это преимущество заметно на низких битретах и при оценке качества цветопередачи. Второе место принадлежит кодеку на основе стандарта MPEG-4, а третье – кодеку на основе устаревшего стандарта H.263+.

Такие результаты объясняются следующим. Стандарт H.264 был разработан для эффективной и помехоустойчивой передачи видео прямоугольного формата. Его первоначальной задачей было обеспечение функциональности схожей с H.263+ и MPEG-4 Part-2, но с лучшим сжатием и с более надежным механизмом передачи видеоданных. Целевые приложения включают двухсторонние видеокоммуникации (видеоконференции и видеотелефония), кодирование сетевого широко вещания, 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. Существенный вклад в победу стандарта H.264 внесли усовершенствованные инструменты, устраняющие временную избыточность. Наиболее перспективными сейчас представляются следующие направления развития:

- использование альтернативных методов сжатия без потерь, такие как различные реализации арифметического кодирования (например, CABAC), цепные коды;
- использование альтернативных преобразований типа вейвлет-преобразований (DWT) и выделения объектов с их последующим описанием.

ЛИТЕРАТУРА

1. Блаттер К. Вейвлет-анализ. Основы теории.// – М. – Техносфера. - 2006, - 279 с.
2. Брейсуэлл Р. Преобразование Хартли. Теория и приложения.// - М. - Мир, - 1990. – 192 с.
3. Ватолин Д., Ратушняк А., Смирнов М. Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео// -М. – Диалог-МИФИ. -2003. - 384 с.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений.// Пер. с англ.- Москва.- Техносфера. – 2006. -1072 с.
5. Ковалгин Ю.А., Вологодин Э.И. Цифровое кодирование звуковых сигналов// – СПб. - КОРОНА-принт, 2004. – 240 с.
6. Ричардсон Я. Видеокодирование H.264 и MPEG-4 – стандарты нового поколения.// - М. – Техносфера. – 2005. - 368 с.
7. Сэломон Д. Сжатие данных, изображений и звука// – М. – Техносфера. - 2004. – 368 с.
8. Тропченко А.Ю., Курносенков И.Н. Анализ современных стандартов сжатия видеоданных// Научно-технический вестник СПбГУ ИТМО.- Вып. 32. - СПб. - СПбГУ ИТМО. – 2006. - с.17-21.
9. Тропченко А.А., Молчанов В.А. Особенности сжатия цветных изображений JPEG-подобными алгоритмами// Научно-технический вестник СПбГУ ИТМО/ - Вып. 32. - СПб. - СПбГУ ИТМО. – 2006. - с.22-26.



В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

О кафедре

Кафедра ВТ СПбГУ ИТМО создана в 1937 году и является одной из старейших и авторитетнейших научно-педагогических школ России.

Первоначально кафедра называлась кафедрой математических и счетно-решающих приборов и устройств и занималась разработкой электромеханических вычислительных устройств и приборов управления. Свое нынешнее название кафедра получила в 1963 году.

Кафедра вычислительной техники является одной из крупнейших в университете, на которой работают высококвалифицированные специалисты, в том числе 8 профессоров и 15 доцентов, обучающие около 500 студентов и 30 аспирантов.

Кафедра имеет 4 компьютерных класса, объединяющих более 70 компьютеров в локальную вычислительную сеть кафедры и обеспечивающих доступ студентов ко всем информационным ресурсам кафедры и выход в Интернет. Кроме того, на кафедре имеются учебные и научно-исследовательские лаборатории по вычислительной технике, в которых работают студенты кафедры.

Чему мы учим

Традиционно на кафедре ВТ основной упор в подготовке специалистов делается на фундаментальную базовую подготовку в рамках общепрофессиональных и специальных дисциплин, охватывающих наиболее важные разделы вычислительной техники: функциональная схемотехника и микропроцессорная техника, алгоритмизация и программирование, информационные системы и базы данных, мультимедиа-технологии, вычислительные сети и средства телекоммуникации, защита информации и информационная безопасность. В то же время, кафедра предоставляет студентам старших курсов возможность специализироваться в более узких профессиональных областях в соответствии с их интересами.

Специализации на выбор

Кафедра ВТ ИТМО предлагает в рамках инженерной и магистерской подготовки студентам на выбор по 3 специализации.

1. Специализация в области информационно-управляющих систем направлена на подготовку специалистов, умеющих проектировать и разрабатывать управляющие системы реального времени на основе средств микропроцессорной техники. При этом студентам, обучающимся по этой специализации, предоставляется уникальная возможность участвовать в конкретных разработках реального оборудования, изучая все этапы проектирования и производства, вплоть до получения конечного продукта. Для этого на кафедре организована специальная учебно-производственная лаборатория, оснащенная самым современным оборудованием. Следует

отметить, что в последнее время, в связи с подъемом отечественной промышленности, специалисты в области разработки и проектирования информационно-управляющих систем становятся все более востребованными, причем не только в России, но и за рубежом.

2. Кафедра вычислительной техники - одна из первых, начавшая в свое время подготовку специалистов в области открытых информационно-вычислительных систем. Сегодня студентам, специализирующимся в этой области, предоставляется уникальная возможность изучать и осваивать одно из самых мощных средств создания больших информационных систем - систему управления базами данных Oracle. При этом повышенные требования, предъявляемые к вычислительным ресурсам, с помощью которых реализуются базы данных в среде Oracle, удовлетворяются за счет организации на кафедре специализированного компьютерного класса, оснащенного мощными компьютерами фирмы SUN, связанными в локальную сеть кафедры. В то же время, студенты, специализирующиеся в данной области, получают хорошую базовую подготовку в области информационных систем, что позволяет им по завершению обучения успешно разрабатывать базы данных и знаний не только в среде Oracle, но и на основе любых других систем управления базами данных.

3. И, конечно же, кафедра не могла остаться в стороне от бурного натиска вычислительных сетей и средств телекоммуникаций в сфере компьютерных технологий. Наличие высокопрофессиональных кадров в данной области и соответствующей технической базы на кафедре (две локальные вычислительные сети, объединяющие около 80 компьютеров и предоставляющие возможность работы в разных операционных средах - Windows, Unix, Solaris), позволило организовать подготовку специалистов по данному направлению, включая изучение вопросов компьютерной безопасности, администрирования, оптимизации и проектирования вычислительных сетей.