Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

Семич Д. Ф.

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ И РЕАЛИЗАЦИЯ ТЕХНОЛОГИИ ВЫСОКОКАЧЕСТВЕННОЙ НИЗКОБИТРЕЙТНОЙ ЦИФРОВОЙ ВИДЕОКОМПРЕСИИ

Препринт
86

Новосибирск 2001

В работе описаны математическая модель и пример риализации низкобитрейтного видеокодека. Эта модель отличается от существующих тем, что при битрейте в 1 Мбит/с качество получаемого видео сравнимо с выдаваемым MPEG–2 при битрейте 3.5 Мбит/с.

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

Dmitry F. Semich

DEVELOPMENT OF THE MATHEMATICAL MODEL AND REALIZATION OF LOWBITRATE HIGH-PERFOMANCE DIGITAL VIDEO-COMPRESSION TECHNOLOGY.

Preprint
86

Novosibirsk 2001

This article describes the mathematical model and realization example of lowbitrate high-performance videocodec. Positive difference of this model from other similar models is quality of compressed video. This model can produce quality of MPEG-2 3.5Mbit videostream within 1 Mbit bitrate.

1. ВВЕДЕНИЕ

В последнее время широкое распространение получает спутниковое цифровое теле- и радиовещание. С тех пор как в 80–х гг. произошел бум, связанный с появлением компакт-диска, где высококачественный звук был записан с применением цифровых технологий, усилия многих компаний были направлены на создание стандарта, который позволил бы записывать и передавать видеоизображение в цифровом виде. При передаче изображения оказалось, что для уменьшения потока информации достаточно передавать не каждый кадр в отдельности, а лишь ИЗМЕНЕНИЯ, происходящие в каждом последующем кадре по сравнению с предыдущим. Так как таких изменений обычно не так уж много, то экономится объем передаваемой информации.

Стандартом стали разработки Motion Picture Experts Group (MPEG). Вначале 80-х гг. был разработан стандарт MPEG–1. Этот стандарт, известный по VideoCD, имел недостатки. Например, однотонная поверхность всегда оказывалась нарисованной из рассыпающихся квадратиков. Также квадраты появлялись на сценах, содержащих много действий. Известно, что при издании VideoCD версий фильмов приходилось урезать в несколько раз те сцены, где много движения, взрывов и т.п. Таким образом, не было никакого преимущества по сравнению с VHS, и сейчас MPEG–1 уже практически не используется.

Впоследствии был создан стандарт MPEG–2, явившийся огромным шагом вперед. Теперь изображение стало всегда кристально чистым, с повышенной четкостью и отличной цветопередачей, звук же сравним с качеством компакт-диска и даже лучше (обычно применяется частота сэмплирования 48kHz вместо 44.1kHz, характерных для CD). На базе этого MPEG–2 и построены стандарты DVD (Digital Video Disk) и стандарты цифрового телевещания. Ныне используются подстандарты DBS/DSS и DVB, несовместимые между собой. DBS и DSS используются только в США для вещания платных пакетов, поэтому это нам неинтересно, и далее речь пойдет только про DVB (Digital Video Broadcasting), которое используется в Европе, Азии, Африке, Австралии, а теперь уже и в Америке. Главные преимущества, которые нам дают цифровые технологии — это отличное изображение и звук, а также увеличение числа передаваемых каналов с расширением сервиса. Дело в том, что теперь на спутниковом пердатчике, предназначенном для обычного аналогового канала, можно передавать от 8

до 10 каналов цифрового телевидения вместе со стереозвуком (это называется "пакет каналов"). В результате выбор каналов заметно вырос. Со спутника теперь могут вещать каналы, которые раньше не могли себе это позволить, так как аренда "места" стала в 8 раз дешевле. Появились также и новые ранее недоступные удобства. Вместе с цифровым пакетом можно передавать много сопутствующей информации.

Дальнейшее совершенствование алгоритмов кодирования видеоизображений позволит достичь увеличения числа каналов спутникого телевидения в несколько раз при неизменной пропускной способности канала. Кроме того, используя новые алгоритмы цифрового представления видеоинформации, можно увеличить количество минут видеозаписи на цифровом носителе, не меняя самого носителя, что очень важно для дальнейшего развития видеотехники.

2. ОПИСАНИЕ СТАНДАРТОВ MPEG

Слово **MPEG** является сокращением от Moving Picture Expert Group — названия экспертной группы ISO, разрабатывающей стандарты кодирования и сжатия видео- и аудиоданных. Официальное название группы — ISO/IEC JTC1 SC29 WG11. Часто аббревиатуру MPEG используют для ссылки на стандарты, разработанные этой группой. На сегодняшний день известны следующие стандарты:

2.1. MPEG-1

MPEG-1 предназначен для записи синхронизированных видеоизображений (обычно в формате SIF, 288 x 358) и звукового сопровождения на CD-ROM с учетом максимальной скорости считывания около 3 Мбит/с. Качественные параметры видеоданных, обработанных MPEG-1, во многом аналогичны обычному VHS-видео, поэтому этот формат применяется в первую очередь там, где неудобно или непрактично использовать стандартные аналоговые видеоносители. В нем используется стандарт развертки с четкостью в 4 раза меньшей, чем в вещательном телевидении: 288 активных строк в ТВ-кадре и 352 отсчета в активной части ТВ- строки. Субъективная оценка качества ТВ-изображения в зависимости от скорости передачи данных показывает, что стандарт MPEG-1 можно эффективно использовать при кодировании видеоданных до скорости 3,5 Мбит/с, так как в интервале скоростей от 1,5 до 3,5 Мбит/с увеличение скорости передачи видеоданных сопровождается адекватным улучшением качества ТВ-

изображения. Однако дальшейшее повышение скорости передачи уже не ведет к заметному улучшению качества, и при скорости передачи видеоданных выше 3,5 Мбит/с лучшее качество изображения получается при кодировании по стандарту MPEG-2. Метод сжатия MPEG-1 освнован на том, что полностью записывается лишь один кадр из группы примерно в 10 кадров. Это опорный или I-кадр. Он сворачивается методами внутрикадрового сжатия. Следующие кадры сравниваются при кодировании, и вычисляются векторы движения. Для этого кадр подразделяется на макроблоки размером 16x16 пикселов, и измеряется движение фрагмента при переходе к следующему кадру. Для некоторого предсказанного кадра (P-кадра) производится сравнение с реальным, и определяется ошибка предсказания. При помощи векторов движения и разностных данных кодируются и остальные кадры. Их называют двунаправленными (B-кадрами), поскольку для их декодирования необходим I- или P-кадр до и после данного B-кадра. Последовательности I-, P-, B-адров объединяются в фиксированные по длине и структуре группы, называемые GOP (Group of Pictures). Каждая такая группа обязательно начинается с I-адра и с определенной периодичностью содержит P-адры. Ее структуру описывают как M/N, где M — общее число кадров в группе, а N — интервал между P–кадрами. Для кадров разных типов применяется различный уровень компрессии. Меньше вчего сжимаются I–кадры. P–кадр составляет по размеру примерно третью часть от I, а B — восьмую от обьёма, занимаемого I–кадром. Поэтому суммарный поток данных в значительной степени зависит от состава GOP. В зависимости от назначения и требуемого качества записи — видеофильм, мультимедиа–продукция, фильм для демонстрации в Internet и т.д. используется различный состав GOP. Так, типичная для VideoCD IPB группа 15/3 имеет следующий вид: IBBPBBPBBPBBPBB. Программы для записи MPEG обычно позволяют пользователю регулировать состав группы. Теоретически MPEG позволяет описывать кадры большого размера, но в связи с ограничением числа макроблоков на картинку реальный размер кадра составляет 352x240, 30 кадров/с или 352x288, 25 кадров/с, формат 4:2:0, 8 бит на точку.

2.2. MPEG-2

MPEG-2 предназначен для обработки видеоизображения, соизмеримого по качеству с телевизионным при пропускной способности системы передачи данных в пределах от 3 до 15 Мбит/с, в профессиональной аппаратуре используют потоки скоростью до 50 Мбит/с. Стандарт MPEG-2 не регла-

ментирует методы сжатия видеосигнала, а только определяет, как должен выглядеть битовый поток кодированного видеосигнала, поэтому конкретные алгоритмы являются коммерческой тайной фирм-производителей оборудования. Однако существуют общие принципы, и процесс сжатия цифрового видеосигнала может быть разбит на ряд последовательных операций: преобразование аналогового сигнала в цифровую форму, предварительная обработка, дискретное косинусное преобразование, квантование, кодирование.

2.3. MPEG-3

MPEG-3 предназначался для использования в системах телевидения высокой четкости (high-definition television, HDTV) со скоростью потока данных 20-40 Мбит/с, но позже стал частью стандарта MPEG-2 и в литературе отдельно не упоминается.

2.4. MPEG-4

MPEG-4 задает принципы работы с цифровым представлением медиаданных для трех областей: интерактивного мультимедиа (включая продукты, распространяемые на оптических дисках и через сеть), графических приложений (синтетического контента) и цифрового телевидения. MPEG-4 не только стандарт, фактически он задает правила организации среды, причем среды объектно-ориентированной. Он имеет дело не просто с потоками и массивами медиа-данных, а с медиа–объектами — это ключевое понятие стандарта. Объекты могут быть аудио–, видео–, аудиовизуальными, графическими (плоскими и трехмерными), текстовыми. Они могут быть как “естественными” (записанными, отснятыми, отсканированными и т. п.), так и синтетическими (т. е. искусственно сгенерированными). Примерами объектов могут служить неподвижный фон, видеоперсонажи отдельно от фона (на прозрачном фоне), синтезированная на основе текста речь, музыкальные фрагменты, трехмерная модель, которую можно двигать и вращать в кадре, анимированный спрайт. Медиа-объекты могут быть потоковыми. Каждый медиа-объект имеет связанный с ним набор дескрипторов, где задаются все его свойства, операции, необходимые для декодирования ассоциированных с ним потоковых данных, размещения в сцене, а также поведение и допустимые реакции на воздействия пользователя. Из объектов строятся сцены. Сцена имеет свою систему координат, в соответствии с которой размещаются объекты. Звуковые объекты также могут иметь (и менять во времени) координаты в пространстве сцены, благодаря чему дос-

тигаются стерео- и “окружающие” (surround) эффекты. Объекты могут быть элементарными (primitive) и составными (compound), т. е. представляющими ту или иную композицию элементарных объектов. Стандарт задает правила кодирования различных объектов, их иерархии и способы композиции при построении сцены, а также методы взаимодействия пользователя с отдельными объектами внутри сцены. Каждый объект имеет свою локальную систему координат: с ее помощью объект управляется в пространстве и во времени. При помещении объекта в сцену происходит преобразование его локальной системы координат в систему координат старшего по иерархии объекта или глобальную систему координат сцены. Объекты и сцена могут обладать поведением, контролируемым уровнем композиции при визуализации сцены (характер звука, цвет объекта и т. п.). Сцена описывается с помощью иерархической структуры; узлами этой структуры являются объекты, и она динамически перестраивается по мере того, как узлы-объекты добавляются, удаляются или заменяются. В MPEG–4 определен двоичный язык описания объектов, классов объектов и сцен BIFS, который характеризуют как “расширение Си++”. С помощью команд BIFS можно анимировать объекты, менять их координаты, размеры, свойства, задавать поведение, реакции на воздействия пользователя, менять свойства среды, изменять и обновлять сцену, выполнять 2D–геометрические построения и т. п. Поскольку язык двоичный, он весьма компактен и быстр в интерпретации. Согласно заявлениям разработчиков, многие концепции BIFS позаимствованы у VRML, и сейчас MPEG и Web 3D Consortium продолжают работу по сближению MPEG–4 и VRML.

2.5. Краткое описание схемы кодирования MPEG

Базовым объектом кодирования в стандарте MPEG является кадр телевизионного изображения. Поскольку в большинстве фрагментов фон изображения остается достаточно стабильным, а действие происходит только на переднем плане, сжатие начинается с создания *исходного кадра*. **Исходные** (*Intra*) кадры кодируются только с применением внутрикадрового сжатия по алгоритмам, аналогичным используемым в *JPEG*. Кадр разбивается на блоки 8x8 пикселов. Над каждым блоком производится **дискретно-косинусное преобразование** (ДКП) с последующим квантованием полученных коэффициентов. Вследствии высокой пространственной корелляции яркости между соседними пикселами изображения, ДКП приводит к концентрации сигнала в низкочастотной части спектра, который после квантования эффективно сжимается с использованием кодирования кодами

переменной длины. Обработка **предсказуемых** (*Predicted*) кадров производится с использованием предсказания вперед по предшествующим исходным или предсказуемым кадрам. Кадр разбивается на макроблоки 16x16 пикселов, каждому макроблоку ставится в соответствие наиболее похожий участок изображения из опорного кадра, сдвинутый на *вектор перемещения*. Эта процедура называется анализом и **компенсацией движения**. Допустимая степень сжатия для предсказуемых кадров превышает возможную для исходных в 3 раза. В зависимости от характера видеоизображения кадры **двунаправленной интерполяции** (*Bi-directional Interpolated*) кодируются одним из четырех способов: предсказание вперед; обратное предсказание с компенсацией движения используется, когда в кодируемом кадре появляются новые объекты изображения; двунаправленное предсказание с компенсацией движения; внутрикадровое предсказание — при резкой смене сюжета или при высокой скорости перемещения злементов изображения. С двунаправленными кадрами связано наиболее глубокое сжатие видеоданных, но, поскольку высокая степень сжатия снижает точность восстановления исходного изображения, двунаправленние кадры не используются в качестве опорных. Если бы коэффициенты ДКП передавались точно, восстановленное изображение полностью совпадало бы с исходным. Однако ошибки восстановления коэффициентов ДКП, связванные с квантованием, приводят к искажениям изображения. Чем грубее производится квнтование, тем меньший объем занимают коэффициенты и тем сильнее сжатие сигнала, но и тем больше визуальных искажений.

3. ОПИСАНИЕ РАБОТЫ ВИДЕОКОДЕКА

Представленный в данной работе кодек построен по прототипу MPEG. Он использует компенсацию движения (motion compensation) и коррекцию движения (motion estimation). Этапы работы кодека показаны на схеме “Блок-схема видеокодека” (см. п. 5).

Первым этапом кодирования последовательности изображений является чтение определённого заранее заданного количества изображений и вычисления их характеристик. На первом этапе характеристики, которые нас интересуют — это общая яркость изображения, контрастность и степень дробления представленных на изображении элементов. Также нас интересуют вектора смещения объектов, которые вычисляются для каждого изображения относительно предыдущего (Рис. 1). Характеристики, вычисленные единожды, нигде не вычисляются повторно. У каждой характеристики

есть время жизни, определяемое алгоритмом обработки последовательности изображений. Далее вектора смещения объектов пропускаются через итерационный алгоритм коррекции, число итераций этого алгоритма задаётся извне. На выходе этого алгоритма мы получаем скорректированные “идеальные” вектора смещения объектов. Таким образом, скорректированные вектора движения объектов являются конечным продуктом алгоритма коррекции движения. Эти вектора отсылаются на упаковку различным методам компрессии, построенным в основном на принципах словарных наборов с некоторыми вариациями. Упаковка при этом идёт параллельно, и в итоге записывается лучшее значение, полученное при использовании одного из алгоритмов.

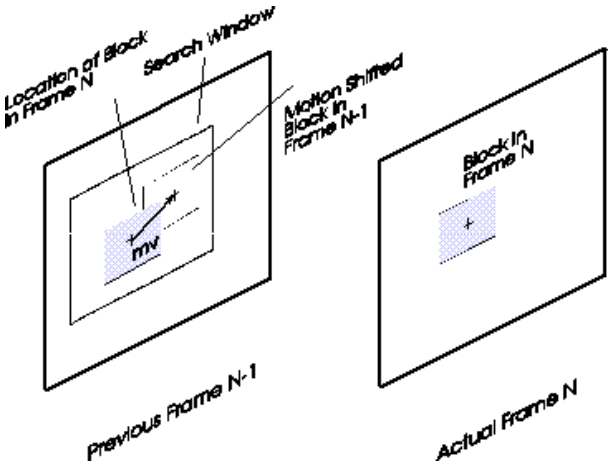

Рис. 1. Схема поиска MV

Вторым этапом упаковки последовательности изображений является определение типа фрейма для каждого отдельно взятого изображения последовательности. Опеределение типа фрейма производится на основе свойств изображения, определённых на предыдущем шаге. Последовательность фрэймов всегда начинается с I-фрейма — такого изображения, которое сжимается без применения методов коррекции движения, тоесть такими, как DWT, DCT, XC.

Делее за I-фрэймом следуют наборы B-фрэймов, разделенные P-фрэймами, критерием присвоения фрэёму того или иного типа типа является некая величина, которая получается из анализа остаточного изображения

(Рис. 2). Остаточное изображение анализируется на гладкость, зернистость и разнородность. В конце второго этапа мы получаем типизированную последовательность изображений, которая в дальшейшем упаковывается.

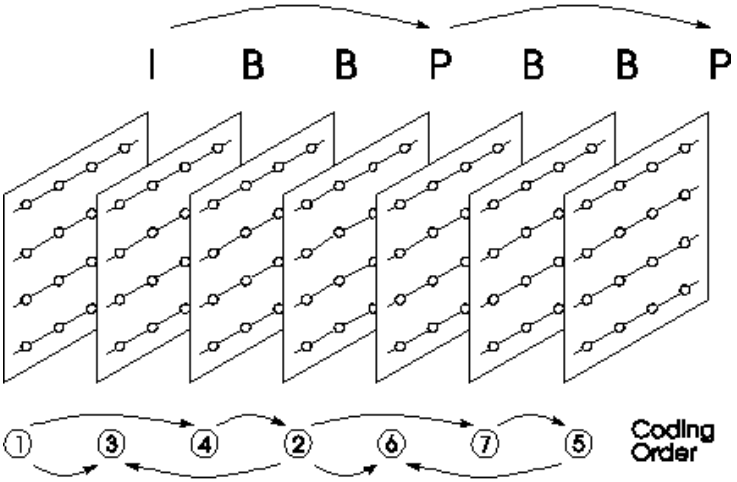

Рис. 2. Типы фрэймов

Третий этап заключается в построении последовательности остаточных изображений по исходной последовательности типизированных изображений и её упаковке. Построение остаточного изображения осуществляется путём прямого копирования содержания блока, расположенного по вектору смещения на следующем кадре, с вычитанием текущего блока на текущем кадре (см. рис. 3). Таким образом мы получаем изображение, которое учитывает в себе характеристики предыдущего и текущего кадров, и по некоторым критериям является минимально возможным. Именно его минимальность даёт нам возможность хорошей упаковки. На рис. 3 приведён процесс получения остаточного изображения.

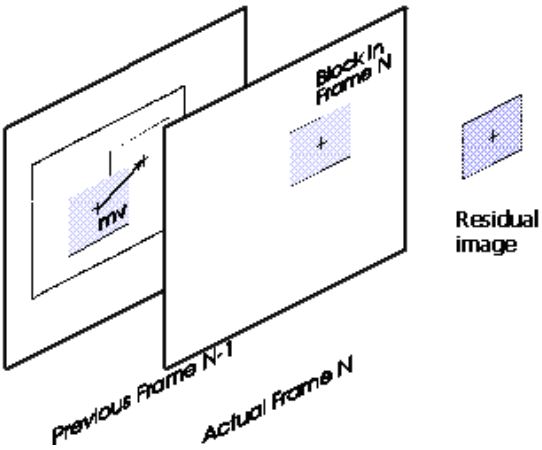

Рис. 3. Построение остаточного изображения

Упаковка остаточных изображений также осуществляется параллельно несколькими алгоритмами, лучший разультат записывается. При упаковке на всех этапах используется алгоритм обработки виртуального буфера (Рис. 4). Этот алгоритм является важной частью механизма распределения битрейта. Виртуальный буфер имеет две функции: взять место и отдать место. Основная цель алгоритма упаковки — сохранить качество при максимально большем сжантии. Если качество отвечает норме, но место для записи информации ещё осталось, оно отдаётся буферу, буфер также может быть заполнен и не примет его. В таком случае оставшееся место запишется дополнительная информация о текущем кадре. Размер буфера задаётся извне при процессе кодирования и влияет на время прекэширования. С помощью подобного алгоритма можно перераспределять место для записи кадров с медленным и быстрым движением. При медленном движении место экономится и задействуется на сценах с быстрым движением, тем самым достигается максимальное качество изображения.

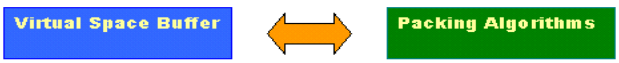

Рис. 4. Обработка виртуального буфера

Именно такой подход к упаковке последовательности изображений позволяет достичь наилучших результатов. При битрейте в 1 Мбит/с достигается качество, доступное при битрейте в 3 Мбит/с на других кодеках.

4. ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

4.1. Преобразования RGB→YUV и YUV→RGB

Преобразование изображения из RGB представления в YUV осуществляется в соответствии со следующими формулами RGB→YUV:

$$Y = 0.114*B + 0.587*G + 0.299*R$$
$$U = 0.5000*B + 0.3313*G + 0.1687*R$$
$$V = 0.0813*B + 0.4187*G + 0.5000*R.$$

Дробные коэффициенты представляются в виде дробей:

$$0.1145 = 3735/32768$$
$$0.2989 = 9798/32768$$
$$0.5866 = 19235/32768$$

$$0.5 = 16384/32768$$
$$0.3313 = 10855/32768$$
$$0.1687 = 5529/32768$$

$$0.0813 = 2264/32768$$
$$0.4187 = 13720/32768$$
$$0.5 = 16384/32768$$

Деление производится с коррекцией, для чего к делимому добавляется половина делителя. Сама операция деления заменена операцией сдвига вправо.

Обратное преобразование (YUV→RGB) цветовых координат вычисляется по стандартным формулам.

$$R = Y + 1.402*(V-128)$$
$$G = Y - 0.3441*(U-128) - 0.7141*(V-128)$$
$$B = Y + 1.772*(U-128)$$

Для упрощения вычислений используются следующие приближения:

0.1145 = 3735/32768
0.2989 = 9798/32768
0.5866 = 19235/32768

0.5 = 16384/32768
0.3313 = 10855/32768
0.1687 = 5529/32768

0.0813 = 2264/32768
0.4187 = 13720/32768
0.5 = 16384/32768

1.772 = 14516/8192
1.402 = 11485/8192
0.3441 = 2819/8192
0.7141 = 5850/8192

4.2. Векторная квантизация (VQ-ME)

Основная идея векторной квантизации состоит в том, чтобы делить изображение на блоки. Как правило некоторые блоки подобны другим блокам, хотя обычно они не идентичны. Кодер идентифицирует класс подобных блоков и заменяет их на "универсальный" блочный представитель класса подобных блоков. Затем он составляет поисковую таблицу, которая отображает короткие двоичные коды к "универсальным" блокам. Самые короткие двоичные коды представляют наиболее общие классы блоков в изображении. При векторной квантизации декодер использует поисковую таблицу, чтобы транслировать приблизительное изображение, составленное из "универсальных" блоков в поисковой таблице. Потери при сжатии неизбежны, потому что фактические блоки заменены универсальным блоком, который является "достаточно хорошим" приближением к первоначальному блоку. Процесс кодирования происходит медленно и в вычислительном отношении интенсивен, кодер должен накопить статистику по частоте блоков и вычислить подобие блоков, чтобы сформировать поисковую таблицу. Процесс декодирования очень быстр, он основывается на уже созданной поисковой таблице. В векторной квантизации поисковая таблица называется книгой шифров. Двоичные коды, которые индексируют в таблицу, могут

называться ключевыми словами. Более высокое сжатие достижимо при уменьшении поисковой таблицы, но и качество воспроизведенного приблизительного изображения ухудшается, поскольку поисковая таблица становится меньшей.

4.3. Дискретное косинусное преобразование (FDCT, IDCT)

Дискретное косинусное преобразование широко используется при сжатии изображений. Стандарт сжатия неподвижных изображений JPEG, стандарты видеоконференций H.261 (p*64) и H.263, а также стандарты цифрового видео MPEG (MPEG-1, MPEG-2 и MPEG-4) используют DCT. В этих стандартах, двумерное DCT применяется к блокам изображения 8*8 пикселей. Аббревиатура FDCT означает Forward Discrete Cosine Transform, IDCT соответственно — Inverse Discrete Cosine Transform.

DCT задается с помощью некоторой матрицы B и ее транспонированной матрицы B^T. Для блока размером 8×8 матрица B имеет вид:

$$B(i, j) = 0.5\,C\cos\{i\pi(2j+1)/16\},$$

где строки и столбцы имеют номера *i* и *j* соответственно, меняющиеся от 0 до 7 и

$$C = \begin{cases} 1/\sqrt{2} & if\ i = 0, \\ 1 & otherwise. \end{cases}$$

Заметим, что DCT является ортогональным преобразованием, и поэтому имеет место равенство $B^{-1} = B^T$.

Человеческие глаза менее чувствительны к высоким компонентам частоты изображения, представленного более высокими коэффициентами DCT. Больший коэффициент квантования применяется к этим более высоким компонентам частоты. Фактическая стандартная матрица квантования (матрица 64 коэффициентов квантования, один для каждого из 64 коэффициентов DCT) в JPEG стандарте имеет более высокие коэффициенты квантования для более высокой частоты коэффициенты DCT.

$$\begin{pmatrix} 1 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{pmatrix}$$

После квантизации обычно применяется обход зигзагом и метод RLE, результат передается алгоритму арифметического кодирования или кодирования по Хаффману, также ограничивается длина зигзага.

4.4. Дискретное вэйвлет преобразование (DWT)

Дискретное вэйвлет преобразование по существу состоит из прохождения сигнала через два фильтра: ФВЧ и ФНЧ. Перед вводом на фильтры сигнал разбивается на два. Таким образом, в этой точке разделенные сигналы совпадают с исходным сигналом. Далее эти сигналы уменьшаются вдвое. Параметры из двух фильтров выбраны так, чтобы, когда сигнал с ФНЧ был добавлен к сигналу с ФВЧ, первоначальный сигнал был бы воспроизведен. Вывод ФВЧ может тогда быть подан в другую пару фильтров для повторного процесса. Простым примером дискретного вэйвлет преобразования волны является преобразование Хаара (Haar).

Имеется входной дискретный сигнал — x [n], ряд выборок, индексированных n.

Haar НПФ (среднее число двух последовательных выборок):

$$G[n] = \frac{1}{2}(x[n] + x[n+1])$$

Haar ВПФ (различие двух последовательных выборок):

$$H[n] = \frac{1}{2}(x[n+1] - x[n])$$

Заметим, что:

$$X[n] + g[n] - h[n]$$
$$X[n+1] = g[n] + h[n]$$

Последовательности вывода g [n] и h [n] содержат избыточную информацию. Их можно безопасно разложить на две последовательности, т.е. опустить четные или нечетные члены, и последовательность все еще воспроизводит первоначальный входной сигнал x [n]. Обычно нечетные выборки опускаются.

$$g[0], g[2], g[4], \ldots$$
$$h[0], h[2], h[4], \ldots$$

Законченный входной сигнал x [n] может быть воспроизведен следующим образом:

$$X[0] = g[0] - h[0]$$
$$X[1] = g[0] + h[0]$$
$$X[2] = g[2] - h[2]$$
$$X[3] = g[2] + h[2]$$

И т.д.

Вывод низкого фильтра — грубое приближение первоначального входного сигнала. Когда входной сигнал является изображением, выходной представляет собой версию первоначального изображения с низкой разрешающей способностью. Вывод высокого фильтра — информация о детализации или сигналы различия. При объединении с грубым приближением первоначальный входной сигнал точно воспроизводится. Грубое приближение называется основным уровнем, а добавление деталей — уровнем расширения. Вывод ФВЧ — сигнал h[n], может быть подан в другую пару фильтров, повторяя процесс. Дискретное вэйвлет преобразование, используемое в сжатии изображения и видео, выполняет итерации процесса сжатия много раз, до тех пор пока не достигается необходимый коэффициент сжатия, или процесс не будет прерван. Преобразование производит то же число бит, что входной сигнал. Результаты на выходе называются коэффициентами преобразования. Преобразование Харра используется прежде всего для иллюстративных целей. Опытным путём установленно, что наи-

более предпочтительным является использование преобразования “7/9”, так как при его применении достигаются наилучшие результаты.

4.5. Древовидная квантизация

На первом шаге к исходному изображению применяется Wavelet Transform, DCT или VQ. Полученное изображение разбивается на четыре равные части. Обозначим их $P_i, 1 \le i \le 4$. Далее вычисляется энергия $E(P_i)$ каждой из частей.

Если существует такое i, что $E(P_i) > 2E(P_j)$ для всех $j \ne i$, то ко всем блокам P_j применяется Wavelet Transform. Далее они аналогично разбиваются на четыре части и т.д. В итоге возникает древообразная рекурсивная процедура. Процесс прерывается, если один из размеров изображения (длина или высота) является нечетным.

4.6. Битовая квантизация

После применения процедуры, описанной в предыдущем пункте, изображение имеет вид матрицы $S(i,j), 0 \le i < N, 0 \le j < M$, каждый элемент которой представлен в виде байта $S(i,j) = s_{ij}^7 s_{ij}^6 s_{ij}^5 s_{ij}^4 s_{ij}^3 s_{ij}^2 s_{ij}^1 s_{ij}^0$, где $s_{ij}^k \ 0 \le k \le 7$ являются соответствующими битами.

Естественным образом возникают восемь битовых матриц $S^k(i,j) = (s_{ij}^k), 0 \le i < N, 0 \le j < M, 0 \le k \le 7$. Процедура Bit Quantization состоит в следующем. Сначала берутся самые старшие разряды, т.е. матрица $S^7(i,j)$. Она передается процедуре арифметического кодирования. Результат кодирования записывается в некоторый буфер. Потом берется матрица $S^6(i,j)$. Она также передается процедуре арифметического кодирования.

Если Bit-rate позволяет, то результат кодирования дописывается в буфер. Во втором случае переходим к более младшим разрядам. Если предоставляемый Bit-rate маленький, то процесс может прерваться на некотором шаге, либо все восемь битовых матриц будут закодированы с помощью арифметического кодирования и помещены в буфер.

4.7. Метод поиска векторов смещения блоков изображения

Фрейм, обрабатываемый в данный момент, разбивается на блоки размером 8×8. Рассмотрим один из них и обозначим его B. На предыдущем фрейме имеется соответствующий ему блок B' размером 8×8, т.е. блок, состоящий из пикселей, имеющих те же координаты.

Далее на предыдущем фрейме рассматривается блок $\overline{B}$ большего размера (он называется макроблоком) такой, что $B' \subset \overline{B}$, и B' находится в центре $\overline{B}$. Внутри макроблока отыскивается блок B'' размером 8×8 такой, что энергия разности $E(B - B'')$ является минимальной.

Если один и тот же минимум достигается для нескольких блоков, то берется такой, что расстояние от него до центра макроблока минимально. Иначе говоря, длина вектора смещения является минимальной. Таких векторов может быть несколько. Реально эта неопределенность разрешается, так как фиксируется процесс обхода области поиска. Например, против часовой стрелки, по возрастающим радиусам. Одновременно считаем энергию разности. После обхода всего макроблока берем блок с наименьшей энергией разности, который первый встречается в этом списке, в случае, если имеется несколько блоков, с данной минимальной энергией разности.

Для того чтобы повысить точность передачи движения в области поиска, вводятся полупиксели, т.е. переходим к сетке, имеющей ячейки в два раза меньше. Значения функции S во вновь введенных узлах (полупикселях) вычисляются посредством линейной интерполяции значений в целых пикселях. Аналогично ищется блок с наименьшей энергией разности и соответствующий ему вектор смещения, но уже на более мелкой сетке.

В последних реализациях алгоритма наиболее подходящий блок, т.е. с минимальной энергией разности, ищется на исходном изображении без префильтрации крестообразным фильтром и на префильтрованном изображении. Из двух найденных блоков выбирается лучший из них по критерию минимимальности энергии разности. Используя один бит, можно запомнить, из какого изображения взят блок.

Аналогично, при рассмотрении B-фреймов рассматриваем два варианта блока на предыдущем не B-фрейме (префильтрованный и не префильтрованный) и два варианта блока на последующем B-фрейме.

4.8. Критерий “новизны” отдельного блока изображения

Принимая во внимание обозначения предыдущего раздела, мы имеем блок B на текущем фрейме и блок B'' на предыдущем фрейме, находящийся в определенной области, наиболее близкий к нему в энергетической норме.

Далее проверяем неравенство. Если $E(B) \le E(B - B'')$, то блок B считаем новым, не имеющим прототипа на предыдущем фрейме. В этом случае к нему применяется дискретное косинусное преобразование DCT.

Если имеет место противоположное неравенство $E(B) > E(B - B'')$, то дискретное косинусное преобразование DCT применяется к разности $R = B - B''$.

4.9. Интерполяция векторов движения блоков к векторам перемещения точек изображения

Приведённый ниже алгоритм позволяет сопоставить вектора движения точкам изображения. Вектора движения точек могут быть получены из векторов движения блоков путём аппроксимации последних с помощью поверхности второго порядка. Запишем уравнение поверхности в виде:

$$F(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$$

Введём такую систему координат, в которой центр текущего блока совпадает с центром координат, соответственно границы блока параллельны осям координат и пересекают их в точках (+1) и (-1).

Для того чтобы вектора для блоков не поменяли свою длину при преобразовании, введём нормирующее условие:

$$\int_{x=-1}^{x=1} \int_{y=-1}^{y=1} f(x, y)dxdy = M(0,0)$$, *где M(0,0) — вектор перемещения центрального блока*

Следующее выражение даёт нам сглаженность векторов на полуширине прямоугольников:

$$F(i,j)=\frac{M(0,0)+M(i,j)}{2}$$

Таким образом мы можем построить четыре условия на точки.Следующим условием, которое также распадается на четыре случая, будет то, что углы прямоугольника гладко соединены со своими соседями.

$$F(i,j)=\frac{M(0,0)+M(i,0)+M(i,j)+M(0,j)}{4}$$

Итак, мы получили 9 уравнений с 6 неизвестными, в матричной форме это выглядит так:

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \frac{M_{-1,0}+M_{0,0}}{2} \\ \frac{M_{1,0}+M_{0,0}}{2} \\ \frac{M_{0,-1}+M_{0,0}}{2} \\ \frac{M_{0,1}+M_{0,0}}{2} \\ \frac{M_{-1,0}+M_{-1,-1}+M_{0,-1}+M_{0,0}}{4} \\ \frac{M_{-1,0}+M_{-1,1}+M_{0,1}+M_{0,0}}{4} \\ \frac{M_{1,0}+M_{1,1}+M_{0,1}+M_{0,0}}{4} \\ \frac{M_{1,0}+M_{1,-1}+M_{0,-1}+M_{0,0}}{4} \\ M_{0,0} \end{bmatrix}$$

Это уравнение можно решить взяв псевдо обратную матрицу.

$$\begin{bmatrix} \frac{5}{53} & \frac{5}{53} & \frac{-43}{106} & \frac{-43}{106} & \frac{12}{53} & \frac{12}{53} & \frac{12}{53} & \frac{12}{53} & \frac{-15}{53} \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{-1}{4} & \frac{1}{4} & \frac{-1}{4} & 0 \\ \frac{-43}{106} & \frac{-43}{106} & \frac{5}{53} & \frac{5}{53} & \frac{12}{53} & \frac{12}{53} & \frac{12}{53} & \frac{12}{53} & \frac{-15}{53} \\ \frac{-1}{6} & \frac{1}{6} & 0 & 0 & \frac{-1}{6} & \frac{-1}{6} & \frac{1}{6} & \frac{1}{6} & 0 \\ 0 & 0 & \frac{-1}{6} & \frac{1}{6} & \frac{-1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{-1}{6} & 0 \\ \frac{35}{106} & \frac{35}{106} & \frac{35}{106} & \frac{35}{106} & \frac{-11}{53} & \frac{-11}{53} & \frac{-11}{53} & \frac{-11}{53} & \frac{27}{53} \end{bmatrix}$$

Теперь осталось выполнить обратную подстановку, чтобы определить коэффициенты функции поверхности $F(i,j)$. На краях изображения или там, где вектора смещения принадлежат отличному от рассматриваемого физическому объекту, в качестве угловой точки может быть положена другая угловая точка прямоугольника или его центральная точка. Можно построить и более "умные" алгоритмы, основываясь на этом принципе.

4.10. Фильтр "Демоскито" (Demosquito)

Стандартный алгоритм "Демоскито" состоит из нескольких шагов, приводимых ниже. "Демоскито" убирает эффект появления лишних пикселов около контрастных границ (эффект Гиббса).

1. Копируем блок 8×8 с окружением в специальный массив *ARR1*. При этом, если блок находится на краю, пиксели окружения дополняются копированием существующих.
2. Определяем для блока максимум и минимум яркости.
3. Вычисляем величину *thr=(max+min+1)/2*.
4. Вычисляем величину *range=max-min.*
5. Корректируем *thr*:

$$if\ (range<16)\ then\ thr = 0;$$

6. Для блока 10x10 (8×8 и один ряд — пиксели окружения) запишем индексный массив *ARR2*:

$$bin(h,v) = \begin{cases} 1, & if\ \ pix(h,v) \geq thr, \\ 0, & otherwise. \end{cases}$$

7. Для пикселя с координатами *(h,v)* блока 8×8 вычисляется сумма 9 элементов массива *ARR2*:

Sum(h,v) = ARR2(h — 1,v — 1)+2*ARR2(h — 1,v)+4*ARR2(h — 1,v+1)+ 8*ARR2(h,v — 1) + 16*ARR2(h,v)+32*ARR2(h,v+1)+ 64*ARR2(h+1,v –1) + 128*ARR2(h+1,v) + 256*ARR2(h+1,v+1).

Если полученная сумма *Sum(h,v)* равна 0 или 511, пиксель фильтруется в соответствии с п. 7.1, 7.1a или 7.1b, если равна 1, 4, 64 или 256, то в соот-

ветствии с п.7.2, а если 2, 8, 32 или 128, то в соответствии с п.7.3. В остальных случаях пиксель остается неизменным, и мы переходим к п.8.

7.1. Вычисляем

*f'(h,v)=(ARR1(h — 1,v — 1)+2*ARR1(h — 1,v)+ARR1(h — 1,v+1)+2*ARR1(h,v — 1)+4*ARR1(h,v)+2*ARR1(h,v+1)+ARR1(h+1,v — 1)+2*ARR1(h+1,v)+ARR1(h+1,v+1)+8)/16*,

после чего переходим к п. 7.4.

7.1a. Альтернативный вариант фильтра:

f'(h,v)=(32+5(ARR1(h — 1,v — 1)+7*ARR1(h — 1,v) +5*ARR1(h — 1,v+1)+7*ARR1(h,v — 1)+16*ARR1(h,v) +7*ARR1(h,v+1)+5*ARR1(h+1,v — 1)+7*ARR1(h+1,v) +5*ARR1(h+1,v+1))/64*,

после чего переходим к п. 7.4.

7.1b. Альтернативный вариант фильтра (работает "мощнее" первых двух)

f'(h,v)=5(32+ARR1(h — 1,v — 1)+8*ARR1(h — 1,v) +5*ARR1(h — 1,v+1)+8*ARR1(h,v — 1)+12*ARR1(h,v) +8*ARR1(h+1,v+1)+5*ARR1(h+1,v)+8*ARR1(h+1,v) +5*ARR1(h+1,v+1))/64*,

после чего переходим к п. 7.4.

7.2. Вычисляем

*f'(h,v)=(16+5*ARR1(h,v — 1)+5*ARR1(h — 1,v)+12*ARR1(h,v) +5*ARR1(h+1,v) +5*ARR1(h,v+1))/32*,

после чего переходим к п. 7.4

7.3. Вычисляем

*f'(h,v)=(16+5*ARR1(h — 1,v — 1)+5*ARR1(h+1,v — 1)+12*ARR1(h,v) +5*ARR1(h — 1,v+1)+5*ARR1(h+1,v+1))/32*,

после чего переходим к п. 7.4.

7.4. Корректируем *f'(h,v)*, получая фильтрованный пиксель *f(h,v)* по следующему правилу:

```
if( flt'(h,v) — rec(h,v) > max_diff )
        flt(h,v) = rec(h,v) + max_diff
else if( flt'(h,v) — rec(h,v) < — max_diff)
        flt(h,v) = rec(h,v) — max_diff
else
        flt(h,v) = flt'(h,v),
```

где *max_diff* входной параметр.

7.6. Вписываем профильтрованный пиксель в выходное изображение и переходим к следующему пикселю, пока не профильтруем все 64.

4.11. Фильтр “Деблокинг” (Deblocking)

Стандартная процедура деблокинга получает указатель на загруженный в память bitmap, длину строки и число строк изображения, а также параметр квантования *QP*, рабочее значение которого лежит в диапазоне от 10 до 25.

Параметр *QP* упоминается в документации по MPEG-4, из которого взято описание алгоритма фильтрации. Обозначим *AB|CD* пиксели на вертикальной или горизонтальной границах блока 8×8, где “|” — граница между блоками.

Краевые значения *B* и *C* замещаются на *B1*, *C1* соответственно:

B1 = B + d1, C1 = C — d1,

где d1 = sign (d)(MAX(0,|d| — MAX(0,2*|d| — QP))),*

d = (3A — 8B+8C — 3D)/16.

4.12. Фильтр “Крестообразный”

$$\begin{array}{ccccccccc} & & & 1 & & & \\ & & & 1 & & & \\ & & & 1 & & & \\ 1 & 1 & 1 & 4 & 1 & 1 & 1 \\ & & & 1 & & & \\ & & & 1 & & & \\ & & & 1 & & & \end{array}$$

Данный фильтр даёт хорошие результаты по сглаживанию и прост в реализации. Так как нормирующий коэффициент равен 16=(1+1+1+1+1+1+1+1+1+1+1+1+4), то это позволяет использовать операции сдвига вместо деления.

4.13. Зумминг с единовременной фильтрацией

Постановка задачи: по 8 пикселям окружения текущего пикселя построить распределение яркости в блоке 2×2, полученном увеличением текущего пикселя вдвое по горизонтали и вертикали.

Решение (не слишком строгое): выпишем соотношения для FDCT и IDCT, которые в случае блока 2×2 выглядят особенно просто:

FDCT 2×2:

$$F(0,0)= 1/4 * [(f(0,0) + f(0,1)) + (f(1,0) + f(1,1))]$$
$$F(0,1)= 1/4 * [(f(0,0) + f(1,0)) — (f(0,1) + f(1,1))]$$
$$F(1,0)= 1/4 * [(f(0,0) + f(0,1)) — (f(1,0) + f(1,1))]$$
$$F(1,1)= 1/4 * [(f(0,0) + f(1,1)) — (f(1,0) + f(0,1))]$$

IDCT 2×2:

$$f(0,0) = 1/4 * [F(0,0) + F(0,1) + F(1,0) + F(1,1)]$$
$$f(0,1) = 1/4 * [F(0,0) — F(0,1) + F(1,0) — F(1,1)]$$
$$f(1,0) = 1/4 * [F(0,0) + F(0,1) — F(1,0) — F(1,1)]$$
$$f(1,1) = 1/4 * [F(0,0) — F(0,1) — F(1,0) + F(1,1)]$$

Не зная распределения яркости внутри блока 2×2, вычислить коэффициенты DCT *F(i,j)* (за исключением *F(0,0)*, представляющего собой учетверенную среднюю яркость блока 2×2, равную яркости исходного, т.е. немасштабированного пикселя) в точности невозможно, но можно использовать их простую аппроксимацию. Заметим, что *F(0,1)* представляет собой полуразность левой и правой колонок блока 2×2. Очевидно, что в первом приближении *F(0,1) = K1*(P(L) - P(R)),* где *P(L)* и *P(R)* — яркости соответственно левого и правого пикселов (немасштабированных), окружающих текущий пиксель. Так же точно коэффициент *F(1,0) = K2*(P(T) - P(B))*, где *P(T)* и *P(B)* яркости соответственно верхнего и нижнего пикселов окружения. *F(1,1)* вычисляем по разности пикселов окружения, лежащих на продолжении главной и побочной диагоналей: *F(1,1)=K3*((P(LT) + P(RB))-(P(RT)+ P(LB)).* Таким образом, задача сводится к нахождению неизвестных коэффициентов *K1, K2, K3*.

Для нахождения этих коэффициентов попробуем к решению исходной задачи подойти иначе: запишем уравнение покрывающей блок 2x2 поверхности:

$$Z(x,y)k1*x2 + k2*y2 + k3c*x*y + k4*x + k5*y + k6*f$$

Граничные условия связывают известные яркости текущего пикселя и пикселей окружения, что позволяет построить систему линейных уравнений относительно *k1 - k6*. Решая систему, получаем требуемые соотношения для *f(i,j)*:

*f(0,0) = 1/4 * [4*P + 1/2*(P(L) — P(R)) + 1/2*(P(T) — P(B)) + 1/16*((P(LT) + P(RB))- (P(RT)+ P(LB))]*
*f(0,1) = 1/4 * [4*P — 1/2*(P(L) — P(R)) + 1/2*(P(T)-P(B)) — 1/16*((P(LT) + P(RB)) — (P(RT)+ P(LB))]*
*f(1,0) = 1/4 * [4*P + 1/2*(P(L) — P(R)) — 1/2*(P(T) — P(B)) — 1/16*((P(LT) + P(RB)) — (P(RT)+ P(LB))]*
*f(1,1) = 1/4 * [4*P — 1/2*(P(L) — P(R)) — 1/2*(P(T) — P(B)) + 1/16*((P(LT) + P(RB)) — (P(RT)+ P(LB))]*

Сравнение полученных соотношений даёт: *K1=K2=1/2, K3=1/16*.

В более удобном для программирования виде расписанные соотношения для *f(i,j)* выглядят так:

*f(0,0) = 1/64 * [32+64*P + 8*(P(L) — P(R)) + 8*(P(T) — P(B)) + ((P(LT) + P(RB)) — (P(RT)+ P(LB))]*
*f(0,1) = 1/64 * [32+64*P — 8*(P(L) — P(R)) + 8*(P(T) — P(B)) — ((P(LT) + P(RB)) — (P(RT)+ P(LB))]*
*f(1,0) = 1/64 * [32+64*P + 8*(P(L) — P(R)) — 8*(P(T) — P(B))– ((P(LT) + P(RB)) — (P(RT)+ P(LB))]*
*f(1,1) = 1/64 * [32+64*P — 8*(P(L) — P(R)) — 8*(P(T) — P(B))+ ((P(LT) + P(RB)) — (P(RT)+ P(LB))]*

Уточним: *f(0,0)* — верхний левый пиксель блока 2×2, полученного удвоением исходного по вертикали и по горизонтали, f(0,1) — верхний правый пиксель, *f(1,0)* — нижний левый пиксель, *f(1,1)* — нижний правый пиксель. Деление на 64 производится целочисленно. Добавка 32 необходима для правильного округления.

Эксперименты показали, что приведенные выше соотношения могут быть упрощены, если отказаться от использования диагональных элементов матрицы окружения *((P(LT) + P(RB)) — (P(RT)+ P(LB)),* причем упрощение это практически не сказывается на качестве изображения, но заметно повышает скорость масштабирования:

*f(0,0) = 1/8 * [4+8*P + (P(L) — P(R)) + (P(T) — P(B))]*
*f(0,1) = 1/8 * [4+8*P — (P(L) — P(R)) + (P(T) — P(B))]*
*f(1,0) = 1/8 * [4+8*P + (P(L) — P(R)) — (P(T) — P(B))]*
*f(1,1) = 1/8 * [4+8*P - (P(L) — P(R)) — (P(T) — P(B))]*

Иногда бывает полезно дополнительно сгладить масштабированное изображение. Сглаживающая фильтрация может быть выполнена ***одновремен-***

но с масштабированием (т.е. в один проход). При этом приведенные выше упрощенные соотношения выглядят несколько иначе:

*f(0,0) = 1/16 * [8 + 12*P + (P(L) + P(R) + P(T) + P(B)) + 2*(P(L) — P(R)) + (P(T)-P(B))]*
*f(0,1) = 1/16 * [8 + 12*P + (P(L) + P(R) + P(T) + P(B)) — 2*(P(L) — P(R)) + (P(T)-P(B))]*
*f(1,0) = 1/16 * [8 + 12*P + (P(L) + P(R) + P(T) + P(B)) + 2*(P(L) — P(R)) — (P(T) — P(B))]*
*f(1,1) = 1/16 * [8 + 12*P + (P(L) + P(R) + P(T) + P(B)) — 2*(P(L) — P(R)) — (P(T) — -P(B))]*

5. СЛОВАРЬ ТЕРМИНОВ

Степень сжатия (англ. rate) — отношение размера входного (некодированного) файла к размеру выходного (кодированного) файла. Например, степень сжатия 11:1 означает, что закодированный файл в 11 раз меньше оригинала.

Битрейт (англ. bitrate) — количество бит, отведенное для записи или передачи в единицу времени. Измеряют обычно в Кбит/с, т.е. килобит в секунду (англ. Kb/s или Kbps). Степень сжатия (следовательно и качество), определяется шириной потока (bitrate) при кодировании сигнала. Термин битрейт в общем случае обозначает общую величину потока, количество передаваемой за единицу времени информации, и поэтому не связан с внутренними тонкостями строения потока. Его смысл не зависит от того, содержит ли поток моно или стерео, или пятиканальное аудио с текстом на разных языках, или что-либо еще. Bitrate может варьировать в широких пределах.

FEC (Forward Error Correction) — коэффициент, который показывает избыточность информации для данного пакета (избыточность используется для восстановления информации в случае ошибки). FEC может принимать стандартные значения в 1/2, 2/3, 3/4, 5/6, 7/8, 1. FEC 7/8 означает, что на каждые семь битов информации передается один избыточный бит для коррекции ошибок.

ДКП — дискретное косинусное-преобразование.

Кодек (Codec) — система для кодирования и декодирования информации для обеспечения сжатия файла или документа.

AVI (Audio Video Interleaved) — файловый формат, введенный фирмой Microsoft для использования систем работы с видеоизображениями в среде

Windows. Этот формат чередует секторы с видеоданными вместе с секторами со звуковыми данными таким образом, что видеоплейер мог бы поддерживать минимальную буферизацию данных.

BMP — аппаратно независимый формат файлов для хранения графических данных, разработанный для работы в среде Windows. Формат поддерживает изображения от 1 до 24 битов и использует RLE сжатие.

JPEG — стандарт для кодирования цифровых изображений на основе использования дискретных косинусных преобразований и кодирования энтропии. Этот стандарт эффективен при квантовании данных, поэтому его считают методом кодирования с большими потерями.

Keyframe — ключевой кадр. Кадр файла цифрового видеофрагмента, содержащий полное изображение.

MPEG — стандарт для кодирования цифрового кино, включающий некоторые исходные кадры, прогнозирование промежуточных кадров и интерполяцию дополнительных кадров между исходными и промежуточными.

RGB (Red, Green, Blue) — формат представления цвета.

NTSC (National Television System Committee) — система цветного телевидения США.

PAL (Phase Alternating Line) — строки с переменной фазой, система цветного телевидения ФРГ.

SECAM (Systeme Sequentiel Couleurs a Memoire) — последовательная передача цветов с запоминанием, система цветного телевидения СЕКАМ ряда европейских стран и России.

РЕСУРСЫ ИНТЕРНЕТ-САЙТОВ

```
http://www.mpeg.org
http://www.wavelet.org
http://drogo.cselt.stet.it/mpeg/faq/faq_mpeg-4.htm
http://lib.sarbc.ru/win/TXT/dv-faq.txt
http://www.fssr.ru/icccs/kunegin/ref/avi/algor.htm
http://audioservis.narod.ru/index.htm
http://misc.info.kuzbass.net/sound/hardware/AnotheraboutDVD.htm
http://www.optima-m.ru/hi-technews.htm
http://mpeg.boom.ru/mpeg1.htm
http://audioservis.narod.ru/pages/mpeg2.htm
http://codecs.nm.ru/
```

Семич Д. Ф

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ И РЕАЛИЗАЦИЯ ТЕХНОЛОГИИ ВЫСОКОКАЧЕСТВЕННОЙ НИЗКОБИТРЕЙТНОЙ ЦИФРОВОЙ ВИДЕОКОМПРЕСИИ

Препринт
86

Рукопись поступила в редакцию 20.04.01
Рецензент Е. В. Ворожцов
Редактор З. В. Скок

Подписано в печать 28.01.01
Формат бумаги 60 × 84 1/16 Объем 2.7 уч.-изд.л., 3 п.л.
Тираж 50 экз.

НФ ООО ИПО “Эмари” РИЦ, 630090, г. Новосибирск, пр. Акад. Лаврентьева, 6