

УДК 004.932

МЕТОД ОБРАБОТКИ P-КАДРОВ ДЛЯ ВИДЕОКОДЕКА НА ОСНОВЕ ДИСКРЕТНОГО ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ

Умняшкин С.В., д.ф.-м.н., профессор Национального исследовательского университета РФ «Московский государственный институт электронной техники (технический университет)», e-mail: vrintf@miee.ru
Табориский А.Е., аспирант Национального исследовательского университета РФ «Московский государственный институт электронной техники (технический университет)»

Ключевые слова: видеокодек, вейвлет-преобразование, компенсация движения, алгоритм, маска перекрытия, разностный кадр.

Введение

Одна из важных проблем разработки алгоритмов видеокомпрессии заключается в том, чтобы эффективно соединить этапы устранения *пространственной* (внутрикадровой) и *временной* (межкадровой) избыточности данных. В качестве преобразования, снижающего пространственную избыточность данных, часто выступает блочное дискретное косинусное преобразование (ДКП). Однако при использовании ДКП при высоких уровнях сжатия на декодируемом изображении начинает проявляться соответствующая блочная структура, ухудшающая субъективную оценку его качества. В отличие от ДКП, дискретное вейвлет-преобразование (ДВП) применяется ко всему изображению целиком, что позволяет избежать «разваливания» декодированного изображения на отдельные блочные фрагменты.

Снижение временной избыточности данных достигается, как правило, за счет использования некоторой процедуры предсказания кодируемого кадра на основе одного или нескольких ранее закодированных и восстановленных кадров. При этом необходимо построение прогнозного кадра, для чего чаще всего используется блочная компенсация движения в пространственной области. Далее полученная разница между кодируемым и прогнозируемым кадрами (т.н. разностный кадр), обрабатывается как статическое изображение с использованием для снижения внутрикадровой избыточности данных декоррелирующих преобразований (ДКП, ДВП и т.п.). Кадры, к которым применяется описанная выше процедура обработки с прогнозом, называются предсказанными кадрами, или *P-кадрами*. Кадры, к которым применяется только декоррелирующее преобразование и которым, таким образом, при декодировании не требуется дополнительной информации о других кадрах, называются ключевыми, или *I-кадрами*.

В данной работе рассматривается видеокодек на основе ДВП [1], эффективность которого в ряде случаев превышает эффективность MPEG-4 Layer 2 [2]. Нами предлагается модификация кодера [1], основанная на изменении способа обработки P-кадров и включающая:

- использование дробнопиксельной компенсации движения;
- использование оптимизированной маски перекрытия блоков при построении прогнозного кадра;
- учёт маски перекрытия блоков при определении наилучшего вектора перемещения в алгоритме компенсации движения;
- изменение схемы нормализации разностного кадра.

Предложен метод блочной компенсации движения для видеокодеков и рассмотрены варианты его реализации. Соответствующие алгоритмы ориентированы на совместное применение с дискретным вейвлет-преобразованием, но могут быть использованы и для кодеков на основе других преобразований.

Эффективность кодера оценивается в терминах пикового соотношения сигнал-шум (PSNR), производительность кодера оценивается по времени сжатия тестовых видеоизображений. Эксперименты были проведены для различных битовых затрат из диапазона 0,2-0,4 бита на пиксел, на примере двух известных тестовых видеопоследовательностей «Теннис» (Tennis) и «Цветочный сад» (Flower Garden).

Основные принципы работы видеокодека

Будем рассматривать вариант кодера, предназначенный для обработки полутоновых (монокроматических) видеоизображений. На рис. 1 показаны основные этапы алгоритма кодирования, реализованного в видеокодеке [1]. Кодек состоит из модуля компенсации движения, осуществляющего поиск векторов перемещений блоков кадра, модуля вычисления разностного изображения, модуля арифметического кодирования векторов перемещения и модуля сжатия статического изображения. В проводившихся нами экспериментах для сжатия статического изображения в кодеке использовался алгоритм SPIHT [3] с биортогональным ДВП CDF 9/7, принятом в стандарте JPEG 2000.

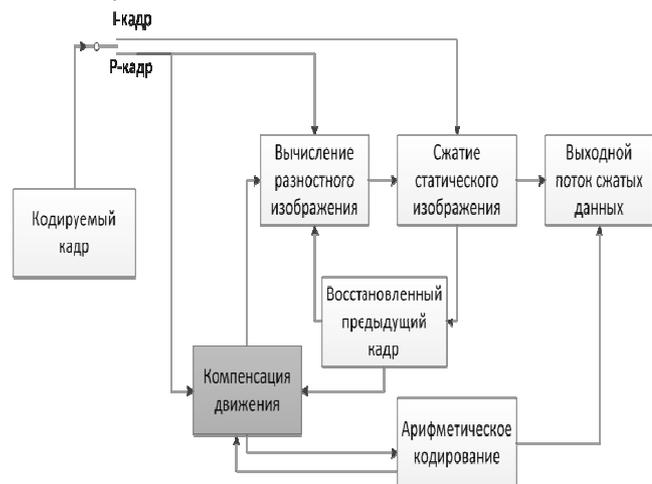


Рис. 1. Схема сжатия кодера на основе дискретного вейвлет-преобразования

I-кадр при кодировании подается непосредственно на модуль сжатия статического изображения. При обработке P-кадра используется восстановленный предыдущий кадр, см. рис. 2. Рассмотрим метод обработки P-кадра подробнее.

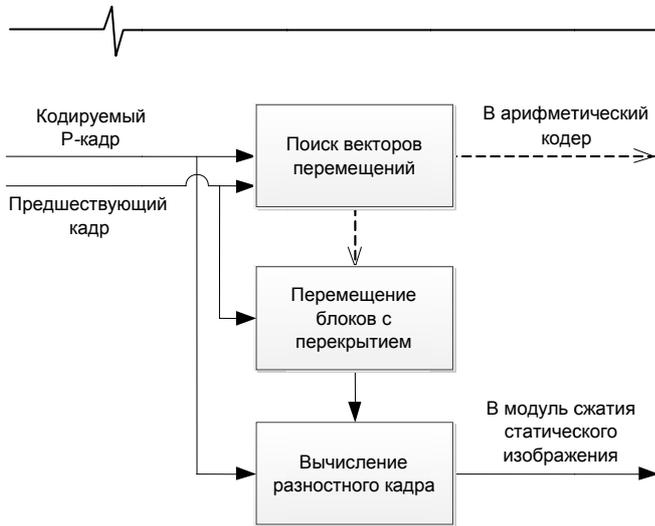


Рис. 2. Общая схема обработки P-кадров

Сначала кодируемый P-кадр и восстановленный предыдущий кадр вместе подаются в модуль компенсации движения. Данный модуль разбивает кодируемый кадр на блоки и для каждого из них находит вектор перемещения, соответствующий наиболее близкому приближению блока в восстановленном предыдущем кадре. Найденные векторы перемещений подаются в арифметический кодер, а затем в выходной поток. Результатом применения найденных векторов перемещений к восстановленному предыдущему кадру является прогнозный кадр (приближение кодируемого кадра), который строится из блоков предыдущего восстановленного кадра. Далее на основании кодируемого и прогнозного кадров строится разностный кадр, который подается в модуль сжатия статического изображения.

Рассмотрим базовый метод поиска векторов перемещений блоков кадра, следуя работе [1]. Каждый P-кадр разбивается на блоки 8x8 пикселей, блоки далее распределяются по трем слоям, см. рис. 3. Слои блоков обрабатываются последовательно, от 1-го к 3-му.

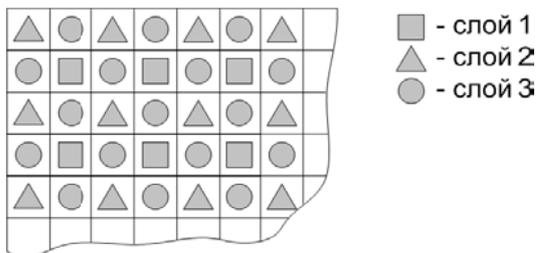


Рис. 3. Распределение блоков по слоям

Из первого слоя прореживанием выбираются 10...25% блоков, по которым при помощи медианной фильтрации координат векторов перемещений отыскивается вектор глобального перемещения всего кадра. Далее, для каждого блока изображения, последовательно по слоям, проводится поиск векторов перемещения по схеме, определяемой номером слоя. При поиске вектора перемещения очередного блока учитывается предсказание этого вектора, которое используется в качестве центра области поиска. Для блоков слоя 1 предсказанием является глобальное перемещение кадра. Для блоков слоя 2 предсказанием является результат медианной фильтрации векторов перемещений четырех соседних диагональных блоков слоя 1. Для блоков слоя 3 при построении прогноза для медианной фильтрации берутся найденные ранее векторы перемещений двух блоков слоя 1, двух блоков слоя 2 и двух блоков слоя 3,

которые окружают текущий обрабатываемый блок третьего слоя. В зависимости от номера слоя используются различные схемы поиска: чем больше номер слоя, тем более точными являются построенные прогнозы перемещений, и тем меньше выбирается область для поиска вектора перемещения обрабатываемого блока.

Исследуем возможность повышения характеристик описанного метода за счёт дополнительных модификаций.

Дробнопиксельная компенсация движения

Эффективность рассматриваемого метода компенсации движения можно повысить, если учесть тот факт, что блок за время перехода к следующему кадру может сместиться не на целое количество пикселей, а на дробное [4]. В настоящее время в системах видеосжатия использование дробнопиксельной компенсации является известным методом повышения эффективности видеосжатия.

Нами был реализован алгоритм дробнопиксельной компенсации, в котором после применения приведенного выше метода поиска векторов перемещений производится их уточнение до полупикселей, а затем и до четвертьпикселей. Для подобного уточнения восстановленное изображение предыдущего кадра увеличивается с использованием интерполяции, принятой в стандарте H.264 [5].

Уточнение вектора перемещений до полупикселей производится только для смещений, не превышающих h пикселей. Зона полупиксельного уточнения имеет размер $2h$ на $2h$. Аналогично, уточнение до четвертьпикселей производится только для смещений, не превышающих q пикселей, и зона четвертьпиксельного уточнения имеет размер $2q$ на $2q$. Таким образом, в разработанном алгоритме варьируются два параметра: h и q .

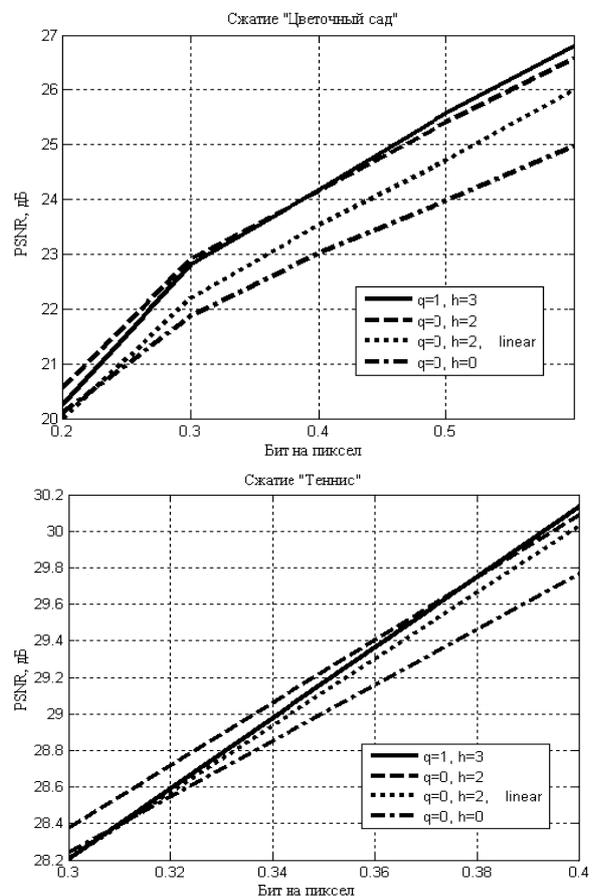


Рис. 4. Зависимость PSNR от битовых затрат при различных методах компенсации движения для последовательностей «Цветочный сад» и «Теннис»

Эффективность дробнопиксельной компенсации движения была проанализирована в виде зависимости от размера зоны перемещений, для которой производится дальнейшее дробнопиксельное уточнение. На рис. 4 приведены зависимости PSNR от битовых затрат, полученные для тестовых последовательностей для четырёх алгоритмов: четвертьпиксельная компенсация ($q=1, h=3$), полупиксельная компенсация ($q=0, h=2$), полупиксельная компенсация с использованием менее ресурсоемкой линейной (linear) интерполяции ($q=0, h=2$), отсутствие дробнопиксельной компенсации ($q=0, h=0$).

В табл. 1 приведены данные для оценки производительности приведённых методов. Измерялось суммарное время, затраченное на поиск векторов перемещений.

Таблица 1.

Оценка производительности методов

Алгоритм	Среднее время обработки последовательности, с	Дополнительные вычисл. затраты, %
Отсутствие дробнопиксельной компенсации ($q=0, h=0$);	2,28	0
Полупиксельная компенсация ($q=0, h=2$, линейная интерполяция)	2,71	19
Полупиксельная компенсация ($q=0, h=2$)	2,89	27
Четвертьпиксельная компенсация ($q=1, h=3$)	4,33	90

Приведённые данные показывают, что для битовых затрат 0,3 бита на пиксел оптимальным является применение блока полупиксельной компенсации при перемещении блока изображения относительно предсказанного вектора перемещения на расстояние, не превышающее 2 пиксела, т.е. $q=0, h=2$. Для битовых затрат 0,4 бита на пиксел оптимальной будет большая зона дробнопиксельной компенсации: $q=1, h=3$. Отметим, что в случае ограниченных вычислительных ресурсов, даже на более высоких битрейтах, следует отказаться от использования ресурсоемкой четвертьпиксельной компенсации ($q>0$) в пользу полупиксельной ($q=0, h=2$). Кроме того, в случае ограниченных вычислительных ресурсов вместо относительно сложной интерполяции, принятой в стандарте H.264, можно использовать более простую линейную интерполяцию, что позволяет существенно сократить время вычислений ценой некоторого уменьшения эффективности сжатия.

Оптимизация маски перекрытия

В кодеке [1] для блоков изображения используется весовая маска с перекрытием размера 16x16 пикселей, взятая из [6] То есть прогнозный кадр строится из перекрывающихся блоков 16x16 пикселей, взятых из предыдущего декодированного кадра и покомпонентно умноженных на рис. 5. Однако для практических целей использование столь большой маски перекрытия может оказаться нецелесообразным в силу высоких вычислительных затрат. Так, при использовании маски 12x12 вместо 16x16 количество умножений пикселей на

весовые коэффициенты уменьшается в $256/144 \approx 1,78$ раза, что существенно увеличивает быстродействие кодека. Перекрытие с использованием маски 12x12 показано на рис. 6.

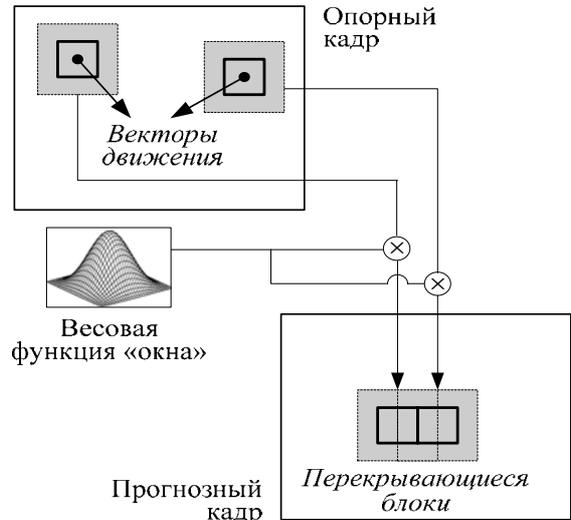


Рис. 5. Блочная компенсация движения с перекрытием с маской 16x16 пикселей

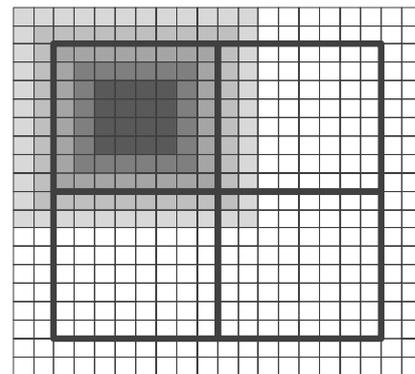
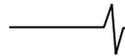


Рис. 6. Перекрытие соседних блоков при использовании маски 12x12 пикселей

Весовая маска должна удовлетворять следующим условиям. Во-первых, для каждого пиксела сумма соответствующих коэффициентов всех перекрывающихся в данном пикселе масок должна быть равна 1. Во-вторых, матрица должна быть симметрична относительно обеих осей. Такую маску 12x12 можно задать двумя параметрами – числами A, B из диапазона $[0, 1]$ – соответствующая матрица весов W будет иметь следующий общий вид:

$$W(A,B) = \begin{pmatrix} A \cdot A & A \cdot B & A \cdot B & A \cdot A & A & A & A & A & A \cdot A & A \cdot B & A \cdot B & A \cdot A \\ B \cdot A & B \cdot B & B \cdot B & B \cdot A & B & B & B & B & B \cdot A & B \cdot B & B \cdot B & B \cdot A \\ B \cdot A & B \cdot B & B \cdot B & B \cdot A & B & B & B & B & B \cdot A & B \cdot B & B \cdot B & B \cdot A \\ A \cdot A & A \cdot B & A \cdot B & A \cdot A & A & A & A & A & A \cdot A & A \cdot B & A \cdot B & A \cdot A \\ A & B & B & A & 1 & 1 & 1 & 1 & A & B & B & A \\ A & B & B & A & 1 & 1 & 1 & 1 & A & B & B & A \\ A & B & B & A & 1 & 1 & 1 & 1 & A & B & B & A \\ A & B & B & A & 1 & 1 & 1 & 1 & A & B & B & A \\ A \cdot A & A \cdot B & A \cdot B & A \cdot A & A & A & A & A & A \cdot A & A \cdot B & A \cdot B & A \cdot A \\ B \cdot A & B \cdot B & B \cdot B & B \cdot A & B & B & B & B & B \cdot A & B \cdot B & B \cdot B & B \cdot A \\ B \cdot A & B \cdot B & B \cdot B & B \cdot A & B & B & B & B & B \cdot A & B \cdot B & B \cdot B & B \cdot A \\ A \cdot A & A \cdot B & A \cdot B & A \cdot A & A & A & A & A & A \cdot A & A \cdot B & A \cdot B & A \cdot A \end{pmatrix}$$



где $A' = (1 - A)$, $B' = (1 - B)$.

Оптимальные значения коэффициентов A и B , дающие максимальную эффективность кодирования в терминах PSNR–битрейт, были определены экспериментально. Всего было проведено около 1400 различных измерений для двух тестовых последовательностей «Теннис» и «Цветочный сад», для ряда битрейтов [0,2, 0,25, 0,3, 0,35, 0,4]. В результате было получено, что оптимальными параметрами являются: $A=0,8$, $B=0,6$, что соответствует линейному спаду весов на границе маски.

На рис. 7 показана зависимость PSNR от используемой весовой маски блока. Замена оригинальной маски 16x16 на меньшую оптимальную маску 12x12 не привела к существенному падению PSNR и даже, напротив, дала некоторый выигрыш на малых битрейтах. За счёт того, что в маске 12x12 16 весовых коэффициентов являются единичными, умножение блока на такую маску требует $144-16=128$ умножений чисел – это в 2 раза меньше, чем при использовании маски 16x16. Поэтому предлагаемую модификацию весовой маски следует признать эффективной.

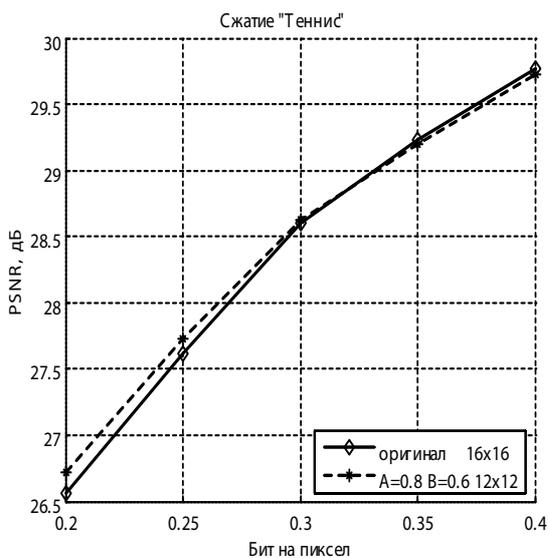
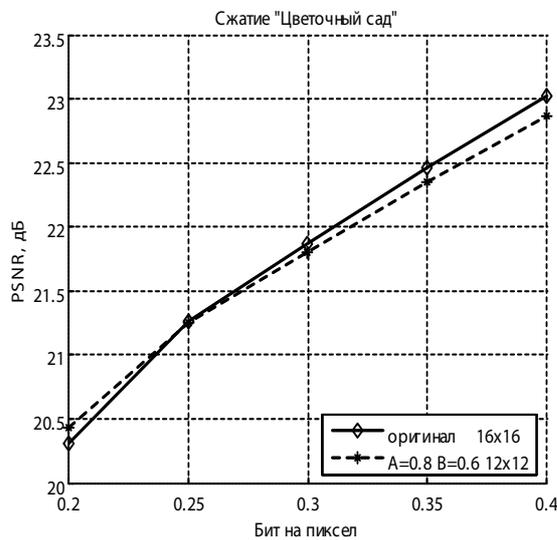


Рис. 7. Зависимость PSNR от используемой маски перемещения для последовательностей «Цветочный сад» и «Теннис»

Учёт маски перекрытия при поиске наилучшего вектора

Для поиска наилучшего вектора перемещений блока в [1] используется RD-критерий. Мера отклонения прогнозного блока $B' = B'(r)$ из предыдущего декодированного кадра от блока B текущего кодируемого кадра определяется как

$$J(r) = D(r) + \lambda R(r), \quad (1)$$

где $D = \sum_{y=1}^N \sum_{x=1}^N (p_{x,y} - p'_{x,y})^2$ – суммарная попиксельная

квадратичная разность блоков, $R = R(r)$ – количество бит, затрачиваемых на кодирование вектора перемещения r , λ – варьируемый параметр, $p_{x,y}$ – значение яркости

пиксела (x, y) блока B кодируемого кадра, $p'_{x,y}$ – значение яркости пиксела (x, y) блока B' предыдущего восстановленного кадра, N – размер блока (в наших экспериментах $N=8$). Наилучшим вектором признается тот, у которого величина J (1) минимальна.

Такой способ оценки векторов перемещений (1) позволяет учесть как отклонение блока восстановленного кадра от рассматриваемого блока, так и эффективность сжатия векторов перемещений при дальнейшем арифметическом кодировании. Однако, в (1) не учитывается тот факт, что при дальнейшем перемещении пикселей выбранного блока, необходимо для построения прогнозного кадра, имеет место перекрытие соседних блоков. То есть суммарная попиксельная квадратичная разность D неточно характеризует ошибку, реально вносимую при построении прогнозного кадра.

Предлагается заменить разность D в (1) на взвешенную разность

$$D' = \sum_{y=1}^M \sum_{x=1}^M ((p_{x,y} - p'_{x,y}) \cdot \mu_{x,y})^2, \quad (2)$$

где $\mu_{x,y}$ – весовой коэффициент маски перекрытия W для пиксела (x, y) , M – размер маски перекрытия в пикселах (в наших экспериментах $M=12$). Таким образом, модифицированная мера отклонения прогнозного блока может быть представлена как

$$J' = D' + \lambda R. \quad (3)$$

Модифицированный критерий J' , в отличие от J (1), учитывает влияние перекрытия соседних блоков (рис. 6) при дальнейшем вычислении прогнозного кадра.

Так как размер маски перекрытия M может быть больше размера блока N , и оценка вектора перемещения проводится чрезвычайно часто, применение модифицированного критерия J' может заметно замедлять работу алгоритма поиска перемещенного блока. Поэтому для выполнения видеокомпрессии на системах, чувствительных к быстродействию, также может иметь смысл использование и другого модифицированного (ускоренного) критерия

$$J'' = D'' + \lambda R, \quad (4)$$

$$D'' = \sum_{y=1}^N \sum_{x=1}^N ((p_{x,y} - p'_{x,y}) \cdot \mu_{x,y})^2.$$

Использование ускоренного критерия J'' , как показали результаты экспериментов, не позволяет достичь столь же высоких результатов, как использование J' (3), однако замедляет работу кодека незначительно.

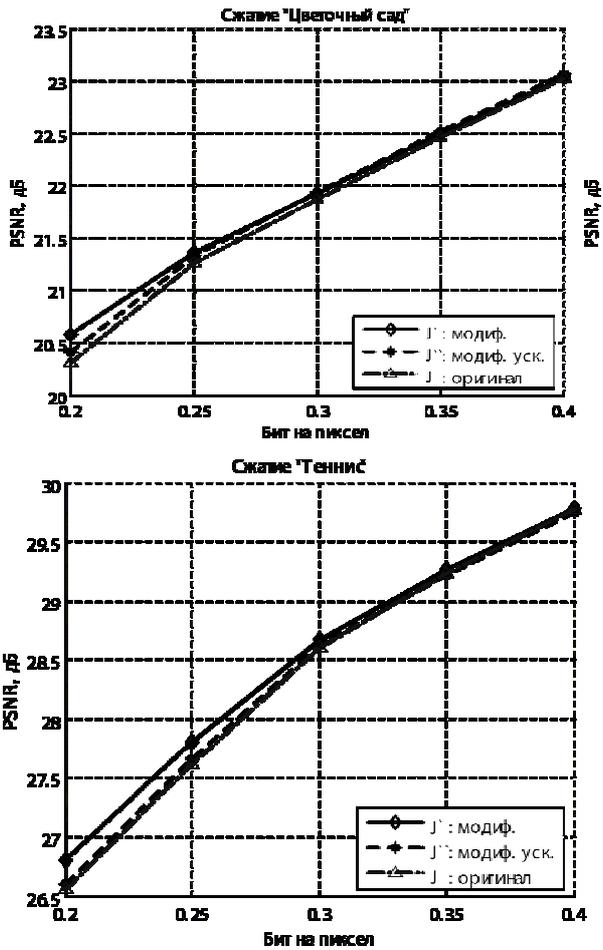


Рис. 8. Зависимость PSNR от используемого критерия оценки вектора перемещения

На рис. 8 приведены характеристики работы кодека при использовании различных критериев выбора наилучших векторов перемещений. Заметим, что по сравнению с использованием критерия (1) предложенные методы поиска по критериям (3) и (4) повышают эффективность кодирования во всем рассмотренном диапазоне битовых затрат. Ускоренный модифицированный критерий (4) целесообразно применять в системах, чувствительных к быстродействию, а критерий (3) – во всех остальных случаях.

Нормализация разностного кадра

При кодировании пиксела изображения целым числом из диапазона [0...255] (одним байтом) по-пиксельная разность прогнозного и кодируемого кадра в общем случае будет лежать в промежутке [-255...255]. Однако для обеспечения высокой про-изводительности кодека и эффективного использования памяти вычислительных устройств для раз-ностного изображения также удобнее использовать формат данных, использующий 1 байт на пиксел. Поэтому используемый нами модуль сжатия статического изображения был реализован для обработки кадров, представленных в таком формате. По этой причине перед кодированием разностного изображения попиксельная разность должна быть прежде отображена в диапазон [0...255], т.е. нормализована.

Простейший способ такой нормализации заключается в выполнении операции целочисленного деления на 2 и последующего прибавления числа 128.

Заметим, что операция нормализации, т.е. отображения [-255...255] → [0...255], может быть задана табличной функцией. Мы исследовали некоторые варианты такого отображения: на рис. 9а показаны оригинальная (соответствующая делению значений пикселей на 2), линейная и сглаженная нормализации. Сглаженная функция построена с учётом того факта, что пиксели в разностном кадре редко принимают большие по модулю значения.

На рис. 9б приведены соответствующие различным вариантам нормализации графики зависимостей ошибки PSNR от битовых затрат для видеопоследовательности «Цветочный сад». Лучшие результаты в среднем достигаются при выборе сглаженной таблицы перехода, что особенно заметно на низких битрейтах – аналогичные выводы дает и анализ результатов обработки последовательности «Теннис». Таким образом, в отсутствие дополнительных вычислительных затрат использование таблицы сглаженной нормализации позволяет повысить характеристики сжатия.

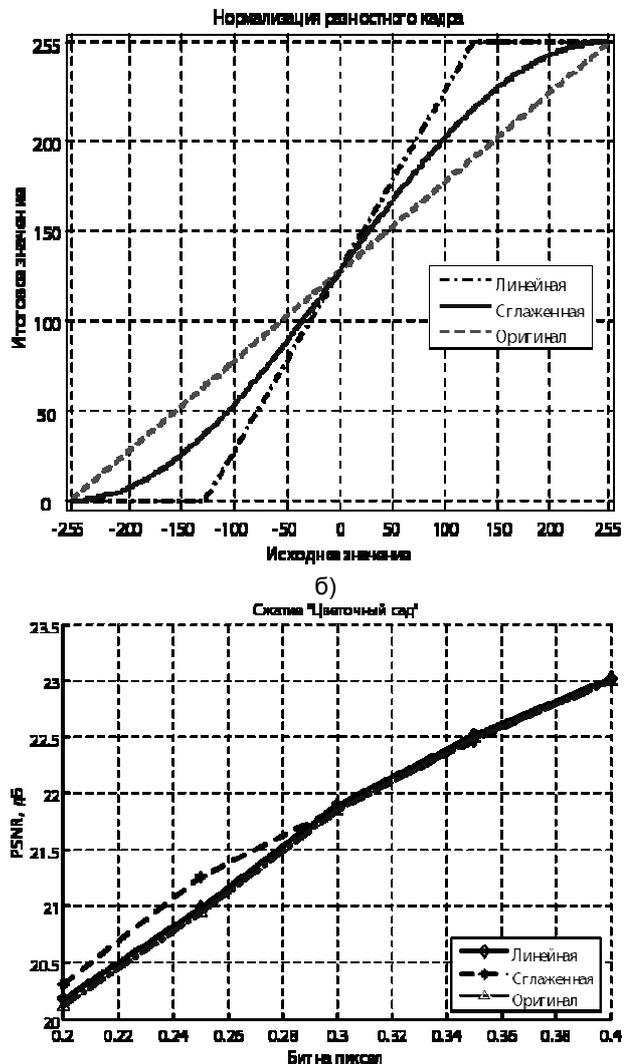


Рис. 9. Схемы нормализации: а) – функции нормализации; б) – зависимость PSNR от используемой схемы

Заключение

Все четыре предложенные способа повышения характеристик кодека могут быть использованы совместно. На рис. 10 приводится сравнение характеристик работы оригинальной [1] и итоговой версии кодека, использующей все описанные модификации. Получено



значительное увеличение итоговой величины PSNR (в диапазоне от 0,5 до 1,1 дБ) для всех представляющих интерес значений битовых затрат.

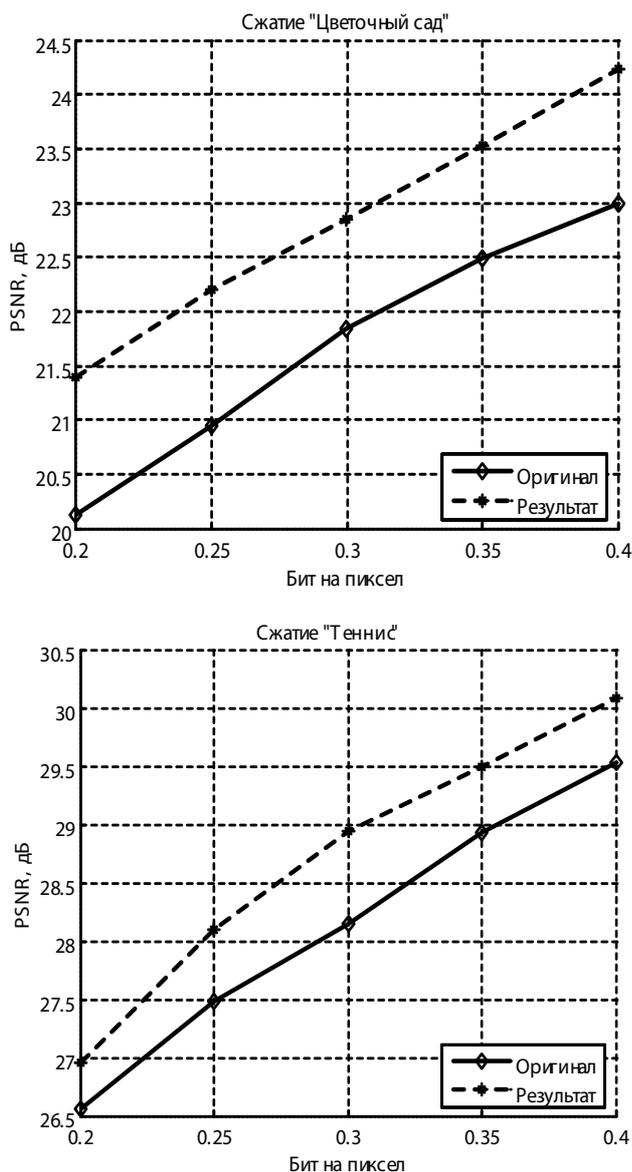


Рис. 10. Итоговые характеристики кодека

Из предложенных модификаций вычислительно сложным является использование дробнопиксельной компенсации, поэтому в системах видеосжатия реального времени, где быстродействие играет ключевую роль, использование данной опции кодека может оказаться нецелесообразным, либо следует использовать её ускоренный вариант. Кроме того, в подобных системах при поиске вектора перемещения также следует использовать ускоренный вариант критерия поиска (4), который практически не приводит к потере производительности.

Тем не менее, даже при использовании опций, дающих наименьшую производительность кодека (использование меры (2) для вычисления ошибки при поиске перемещения блока и включенная четвертьпиксельная компенсация), на компьютере с процессором Intel Core 2 Duo T8300 2,40 ГГц скорость кодирования полутоновой видеопоследовательности Flower

Garden с размером кадра 352×224 пикселя составила 22 кадра в секунду. При использовании опций кодека, обеспечивающих его максимальную производительность, скорость сжатия данной видеопоследовательности на компьютере с указанной конфигурацией возросла до 36 кадров в секунду. Таким образом, разработанная нами программная реализация описанного кодека обеспечивает обработку видео в реальном масштабе времени на вычислительных системах (персональных компьютерах) общего назначения.

Предложенный метод обработки Р-кадров и его составные части могут быть применены не только к видеокодеку [1], и не только к кодекам на основе вейвлет-преобразований, но и для построения других алгоритмов видеокомпрессии, которые используют блочную компенсацию движения.

Литература

1. Александров А.А., Коплович Е.А., Умняшкин С.В. Алгоритм видеокомпрессии на основе дискретного вейвлет-преобразования с трехслойной схемой кодирования векторов движения // Известия вузов. Электроника – №5. – 2008. – С. 69-73.
2. Umnyashkin S., Aleksandrov A., Koplovich E. Wavelet based video codec for telemedicine purposes // Proc. of the 5th Russian-Bavarian Conference on Biomedical Engineering (July 1-4, 2009, Munich, Germany). – 177-180 pp.
3. A. Said, W.A. Pearlman. A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees // IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp. 243-250, June 1996.
4. Тaborиский А.Е. Анализ эффективности дробнопиксельной компенсации перемещения для видеокодека на основе дискретного вейвлет-преобразования // Труды Российского научно-технического общества радиотехники, электроники и связи имени А.С. Попова. – Серия: Цифровая обработка сигналов и её применение. Выпуск XII-2. – С. 148-151.
5. Advanced video coding for generic audiovisual services. Recommendation ITU-T H.264, 03/2009.
6. Orchard M.T., Sullivan G.J. Overlapped block motion compensation: an estimation-theoretic approach // IEEE Transactions on Image Processing, Volume 3 Issue 5, Sep. 1994.

A method for P-frames wavelet-based video codecs

S. Umnyashkin, A. Taborisky

A method of block motion compensation for digital video compression is proposed, some of algorithmic implementations are described. The developed algorithms aim to wavelet based codecs applications. These algorithms can be applied to other transforms as well.