

Потапов Павел Вячеславович

АЛГОРИТМЫ И ПРОГРАММНЫЕ СРЕДСТВА ПОИСКА
ВЕКТОРОВ ПОХОЖЕСТИ ДЛЯ СЖАТИЯ ВИДЕОДАННЫХ

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание учёной степени
кандидата технических наук

Томск – 2008

Работа выполнена на кафедре Автоматизированных систем управления ГОУ ВПО «Томский государственный университет систем управления и радиоэлектроники».

Научный руководитель: Заслуженный деятель науки РФ,
доктор технических наук, профессор
Кориков Анатолий Михайлович

Официальные оппоненты: Доктор технических наук, профессор
Погребной Владимир Кириллович

Кандидат технических наук, доцент
Поляков Алексей Юрьевич

Ведущая организация: Новосибирский государственный
технический университет

Защита состоится « 29 » декабря 2008 г. в 14³⁰ на заседании совета по защите докторских и кандидатских диссертаций Д 212.269.06 при Томском политехническом университете по адресу: 634034, г. Томск, ул. Советская, 84, институт «Кибернетический центр» ТПУ.

С диссертацией можно ознакомиться в библиотеке Томского политехнического университета по адресу: 634034, г. Томск, ул. Белинского, 55.

Автореферат разослан « 28 » ноября 2008 г.

Ученый секретарь
Совета
к.т.н., доцент

М.А. Сонькин

Общая характеристика работы

Актуальность темы: Пропускная способность компьютерных сетей продолжает расти, высокоскоростное соединение с домашним компьютером стало обычным явлением. Вместительность жёстких дисков, флэш-памяти и оптических устройств хранения данных достигла внушительных объемов. Стоимость передачи и хранения информации становится всё меньше и меньше, однако проблема сжатия видеоданных не теряет своей актуальности. Алгоритмы сжатия видеоданных постоянно совершенствуются, создаются новые стандарты видеокомпрессии. Компрессия видеоданных предоставляет два важных преимущества. Во-первых, сжатие позволяет использовать цифровые видеоданные в таких средах хранения и передачи информации, в которых невозможно использовать видеоданные без компрессии. Во-вторых, сжатие видеоданных позволяет более эффективно использовать ресурсы среды передачи или хранения информации.

Основополагающие результаты в области теории информации принадлежат ведущим отечественным (В.А. Котельников, А.Я. Хинчин, А.Н. Колмогоров, А.А. Харкевич, Л.М. Финк) и зарубежным (К. Шеннон, С. Голдман, Д. Хоффман, Р.М. Грей) ученым.

В настоящее время огромное внимание уделяется разработке алгоритмов сжатия видеоданных, использующих компенсацию движения. Начиная с 1980 года, организации *ITU* и *MPEG* последовательно выпускают ряд стандартов сжатия, использующих блочную компенсацию движения: *H.261*, *MPEG-1*, *MPEG-2/H.262*, *H.263*, *MPEG-4*, *H.264 (MPEG-4 part-10)*. Использование алгоритмов компенсации движения при сжатии видеоданных позволяет существенно увеличить степень компрессии при том же соотношении сигнал/шум результирующего видео. Параллельно с разработкой новых стандартов сжатия происходит совершенствование алгоритмов поиска векторов похожести. Алгоритм поиска векторов похожести выполняет важнейшую роль в системах сжатия видеоданных, использующих компенсацию движения. От выбора алгоритма поиска векторов похожести напрямую зависит степень компрессии и вычислительная сложность компрессора. Поиск векторов похожести является одним из самых ресурсоемких этапов компрессии видеоданных. Однако сам алгоритм поиска векторов похожести обычно не фиксируется в стандарте сжатия. Таким образом, модуль поиска векторов похожести можно свободно подвергать алгоритмической оптимизации в рамках стандарта сжатия. Использование более эффективного алгоритма поиска векторов похожести даёт значительное конкурентное преимущество над другими компрессорами, реализующими тот же стандарт сжатия. По этой причине разработка алгоритмов поиска векторов похожести является актуальной задачей.

Актуальным также является исследование оценочных функций, обладающих низкой вычислительной сложностью, но позволяющих с высокой точностью оценить качество найденного вектора похожести с точки зрения компрессии текущего блока.

Цель диссертационной работы состоит в разработке алгоритмического и программного обеспечения поиска векторов похожести, позволяющего улучшить производительность компрессоров и повысить степень компрессии видеоданных.

Для достижения поставленной цели необходимо решить следующие **задачи**:

- выполнить анализ существующих алгоритмов поиска векторов похожести;
- разработать программные средства сравнения алгоритмов поиска векторов похожести;
- реализовать и сравнить существующие алгоритмы поиска векторов похожести;
- разработать новую оценочную функцию;
- разработать и реализовать более эффективный по сравнению с аналогами алгоритм поиска векторов похожести.

Объектом исследования является компрессия цифровых видеоданных. **Предметом исследования** являются алгоритмы поиска векторов похожести, используемые при сжатии видеоданных.

Методы исследования: Результаты диссертации получены с помощью методов математического и имитационного моделирования, методов оптимизации и вычислительной математики. В процессе программной реализации теоретических результатов использовались методы теории алгоритмов, теории структур данных и объектно-ориентированного программирования.

Научная новизна:

1. Разработана программная система «*MEFramework*», позволяющая оценивать вычислительную сложность и эффективность алгоритмов поиска векторов похожести. Отличительными особенностями данной программной системы является возможность сравнивать степень компрессии и количество вызовов оценочной функции при сжатии с использованием различных алгоритмов поиска векторов похожести, а также возможность отдельно задавать оценочную функцию и алгоритм поиска.

2. Предложен оригинальный алгоритм поиска векторов похожести, отличающийся от известных использованием метода градиентного спуска для локального поиска, применением обратного прохода для уточнения найденных векторов похожести и использованием метода сканирования для векторов с плохим начальным приближением.

3. Предложена новая оценочная функция, используемая при поиске векторов похожести. В отличие от известных, предложенная оценочная функция позволяет учитывать не только сложность компрессии невязки блоков, но и сложность кодирования векторов похожести.

Практическая ценность:

– Реализована программная система «MEFramework» в виде фильтра *DirectShow*. Программная система «MEFramework» позволяет оценивать вычислительную сложность и эффективность алгоритмов поиска векторов похожести. Отличительной особенностью данной программной системы является возможность отдельно задавать оценочную функцию и алгоритм поиска векторов похожести. Это позволяет оценивать эффективность алгоритмов поиска векторов похожести при использовании различных оценочных функций.

– В качестве модулей системы «MEFramework» программно реализованы алгоритмы локального поиска, а также наиболее эффективные из известных алгоритмов поиска векторов похожести.

– Программно реализован предложенный алгоритм поиска векторов похожести. Для предложенного алгоритма реализована новая оценочная функция. При помощи программной системы «MEFramework» произведено сравнение с современными аналогами. Экспериментальные результаты продемонстрировали преимущество разработанного алгоритма над аналогами.

– Разработанный алгоритм поиска векторов похожести был внедрен компанией ООО «МэйнКонцепт-Дивикс» в составе коммерческого продукта «Mainconcept MPEG-2 Video Encoder».

Основные положения, выносимые на защиту:

1. Разработанная программная система «MEFramework» позволяет количественно оценивать вычислительную сложность и степень компрессии, достигаемую при использовании различных алгоритмов поиска векторов похожести.

2. Использование оригинального алгоритма поиска векторов похожести, использующего метод градиентного спуска для локальной оптимизации, позволяет уменьшить количество вычислений оценочной функции и увеличить степень компрессии, достигаемую при сжатии видеоданных.

3. Использование новой оценочной функции, учитывающей сложность кодирования векторов, позволяет увеличить степень компрессии, достигаемую при сжатии видеоданных.

Внедрение результатов:

Алгоритм поиска векторов похожести, разработанный в рамках данной диссертационной работы, внедрен в коммерческий продукт компании ООО «МэйнКонцепт-Дивикс», *Mainconcept MPEG-2 Video Encoder*. Программная система «MEFramework» используется компанией ООО «МэйнКонцепт-Дивикс» для проведения внутренних испытаний алгоритмов поиска векторов похожести.

Апробация работы. Основные положения работы докладывались на следующих семинарах и конференциях:

1. XLIII международная научная студенческая конференция «Студент и научно-технический прогресс» - Новосибирск, 2005. Доклад отмечен дипломом третьей степени.

2. XI Всероссийская научная конференция студентов-физиков и молодых учёных - Екатеринбург, 2005.

3. Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых «Научная сессия ТУСУР-2005» - Томск, 2005. Доклад отмечен почётной грамотой «за лучший доклад».

4. Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых «Научная сессия ТУСУР – 2007» - Томск, 2007.

5. Научно-практическая конференция «Электронные средства и системы управления: итоги реализации программы развития электроники и IT-технологий в Томской области» - Томск, 2008. Доклад отмечен дипломом первой степени.

6. Научно-технические семинары кафедры автоматизированных систем управления, ТУСУР.

Публикации. По теме диссертации автором опубликованы 10 работ, в том числе 4 статьи в научных периодических журналах, из них 1 в журналах из перечня ВАК [7], 5 докладов опубликовано в материалах научных конференций и семинаров, зарегистрирована программа в Отраслевом фонде алгоритмов и программ (ОФАП) [10].

Структура и объем диссертации: Диссертационная работа состоит из введения, трех глав и трех приложений. Содержит 142 страницы, 11 таблиц, 48 рисунков и 3 приложения. Список цитируемой литературы содержит 98 наименований.

Содержание работы

Во введении обоснована актуальность темы диссертации, определены цель и задачи исследования, научная новизна и практическая ценность результатов диссертации, сформулированы основные положения, выносимые на защиту.

В первой главе рассмотрены вопросы сжатия видеоданных. Представлена история развития видеокомпрессии, используемые в современных видеокомпрессорах методы сжатия видеоданных.

Компрессия – это процесс сжатия данных с целью представления этих данных меньшим количеством бит. Сжатие предполагает наличие пары систем: компрессор и декомпрессор (енкодер, декодер). Компрессор преобразует исходные данные в сжатую форму (занимающую меньшее количество бит) для передачи или хранения данных. Декомпрессор преобразует данные из сжатой формы обратно к первоначальному виду.

При сжатии видеопоследовательности кадры могут быть сжаты как отдельные изображения, путём устранения пространственной избыточности, подавления мелких деталей, представления в виде оптимального математического кода и т.д. Степень компрессии можно существенно увеличить, если учесть тот факт, что в пределах коротких интервалов времени (промежутков между кадрами) видеоизображение обычно изменяется незначительно. Неподвижную часть кадра можно кодировать как разницу с предыдущим кадром, а движущуюся часть описать при

помощи векторов похожести. Вектором похожести (*motion vector*) называют смещение между координатами блока, используемого для предсказания, и координатами текущего блока. Процесс устранения временной избыточности с использованием векторов похожести называется компенсацией движения (*motion compensation*). При сжатии с применением метода компенсации движения для нахождения векторов похожести используются алгоритмы поиска векторов похожести, обеспечивающие поиск векторов, позволяющих достичь наибольшей степени компрессии.

Векторы похожести можно искать для различных по форме и размеру участков кадра. Поиск может осуществляться для целого кадра, для каждого пикселя отдельно, либо для участков произвольной формы. На практике чаще всего используют метод согласования блоков. Этот метод подразумевает разбиение кадра на большое число блоков (обычно прямоугольных). Каждый блок одного кадра (блок поиска) сравнивается с блоками предыдущего кадра, и оценивается вероятность того, что они являются фрагментами одного и того же участка изображения. Затем вычисляются компоненты смещения блока из исходного положения до положения, которое наилучшим образом согласуется с блоком поиска, что дает величину и направление вектора похожести. Точность оценки вектора находится на уровне одного элемента, поскольку обнаружение согласования блоков основано на сравнении значений элементов изображения. Размеры области поиска определяют максимальную величину вектора похожести, которая может быть измерена, поскольку этот параметр ограничивает максимальную скорость перемещения объекта, которую возможно зафиксировать.

Формально задача поиска векторов похожести представляет собой задачу минимизации оценочной функции. Для поиска векторов похожести общепринято использовать следующие оценочные функции:

1. $M(V_x, V_y)$ – для определения значения этой функции производится сжатие блока с использованием текущего вектора похожести. Значением оценочной функции является размер сжатого блока в битах. При использовании данной оценочной функции производится непосредственная минимизация размеров сжатых блоков кадра.

2. Оценочные функции, основанные на минимизации энергии невязки текущего и предсказанного блоков. Для блока размером $N \times N$, имеющего координаты (x, y) оценочные функции вычисляются следующим образом:

$$SAD(V_x, V_y) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} |I_t(x+m, y+n) - I_{t-1}(x+V_x+m, y+V_y+n)|,$$

$$SSE(V_x, V_y) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} (I_t(x+m, y+n) - I_{t-1}(x+V_x+m, y+V_y+n))^2,$$

где (V_x, V_y) – значения вектора похожести, I_t, I_{t-1} – значения яркости пикселей текущего и предыдущего кадров соответственно.

В первой главе приведены описания следующих алгоритмов поиска векторов похожести: алгоритма полного перебора, алгоритма поиска по алмазу, *NTTS*, *CBA*, *ADZS*, *PMVFAST*, *EPSZ*.

Во второй главе обосновывается выбор критериев оценки эффективности алгоритмов поиска векторов похожести: размер S сжатой видеопоследовательности и количество вычислений K оценочной функции. Сжатие видеопоследовательности производится согласно стандарту *MPEG-2*. При сжатии используется постоянный коэффициент квантования для всех блоков видеопоследовательности. Это обеспечивает одинаковый уровень вносимых при компрессии искажений для сравниваемых алгоритмов.

Для выполнения сравнительного анализа алгоритмов поиска векторов похожести реализована программная система «*MEFramework*» в виде фильтра *DirectShow*. Состав программной системы представлен на рис. 1.

Для сравнения алгоритмов поиска векторов похожести и оценочных функций с помощью «*MEFramework*» алгоритмы и оценочные функции должны быть реализованы в качестве динамических библиотек с оговоренным программным интерфейсом. Для оценки эффективности того или иного алгоритма либо оценочной функции пользователю необходимо выбрать соответствующий файл динамической библиотеки.

На вход «*MEFramework*» подаются несжатые видеоданные. «*MEFramework*» производит сжатие входных видеоданных согласно стандарту *MPEG-2*. Компенсация движения производится с использованием выбранного алгоритма поиска векторов похожести и оценочной функции (блок *MC* (*motion compensation*) на рис. 1). Далее видеоданные подвергаются дискретному косинусному преобразованию (блок *DCT* – *discrete cosine transform*), квантованию (блок *Q* – *quantization*) с постоянным значением коэффициента квантования и арифметическому кодированию при помощи кодов переменной длины (блок *VLC* – *variable length coding*). Для получения восстановленного кадра, использующегося в качестве опорного при компрессии следующего кадра, производится декодирование текущего кадра. Для этих целей после операции квантования кадр подвергается деквантованию (в блоке Q^{-1} – *dequantization, inverse quantization*) и обратному дискретному косинусному преобразованию (в блоке *IDCT* – *inverse discrete cosine transform*). На выходе *MEFramework* имеем сжатый в соответствии со стандартом *MPEG-2* видеопоток. Декодирование видеопотока может быть осуществлено любым *MPEG-2* декодером.

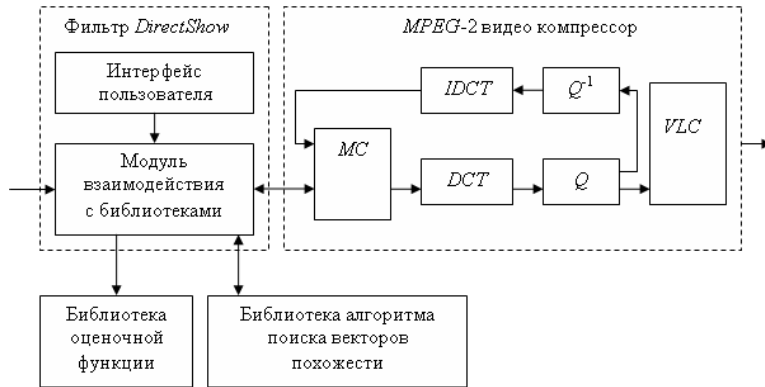


Рис. 1. Состав разработанной программной системы

Результатом работы «MEFramework» являются две численные оценки, позволяющие сделать вывод об эффективности и вычислительной сложности алгоритма на заданной исходной видеопоследовательности.

Преимущества разработанной программной системы состоят в следующем:

- выходом программной системы является сжатая с применением стандарта MPEG-2 и указанного алгоритма поиска векторов похожести входная видеопоследовательность. Компрессия производится с постоянным коэффициентом квантования, что позволяет оценивать качество найденных векторов похожести непосредственно по степени сжатия видеопоследовательности;
- оценка вычислительной сложности алгоритма поиска векторов похожести производится по количеству вызовов оценочной функции. Такой подход позволяет получить оценку вычислительной сложности алгоритма, не зависящую от производительности вычислительной системы, на которой производилось тестирование;
- использование системы динамических библиотек позволяет разделить код программной системы и код алгоритмов поиска векторов похожести;
- реализации оценочной функции и алгоритма поиска векторов похожести располагаются в различных подгружаемых библиотеках. Это позволяет оценить эффективность алгоритма поиска векторов похожести в сочетании с различными оценочными мерами не внося изменений в сам алгоритм;
- «MEFramework» позволяет производить поиск векторов похожести с точностью до половины пикселя.

Данная точность выбрана исходя из ограничений стандарта MPEG-2.

Во второй главе также приведены описания тестовых видеопоследовательностей. Для сравнения алгоритмов поиска векторов похожести были использованы десять видеопоследовательностей, рекомендованных организацией ITU для проверки цифровых кодеков.

С использованием разработанной программной системы был произведён сравнительный анализ оценочных функций M , SAD и SSE . При проведении сравнения в качестве алгоритма поиска векторов похожести использовался алгоритм полного перебора. Размеры сжатых с использованием сравниваемых оценочных функций для пяти видеопоследовательностей представлены на рис. 2.

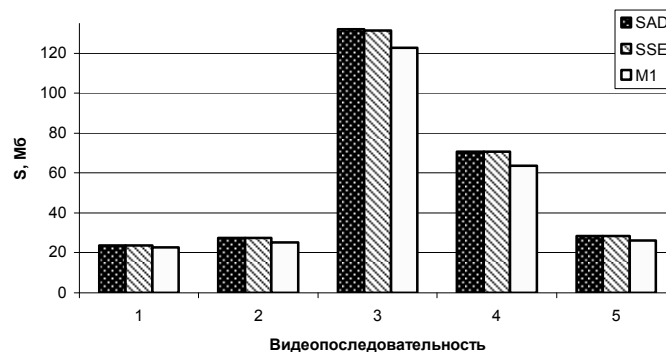


Рис. 2. Значения меры S для сравниваемых оценочных функций

По результатам сравнения сделаны следующие выводы:

1. Использование оценочной функции M позволило достичь наибольшей степени компрессии. Однако оценочная функция M обладает высокой вычислительной сложностью и редко используется на практике при поиске векторов похожести.

2. Использование оценочных функций SAD и SSE позволило достичь схожей степени компрессии. Вычислительная сложность меры SSE выше, чем у меры SAD , так как при вычислении SSE используются операция умножения, а при вычислении SAD – нет. В связи с этим был сделан вывод о том, что использование меры SAD более эффективно.

В третьей главе предлагается новый алгоритм поиска векторов похожести. Выдвигаются требования к разрабатываемому алгоритму. Для формализации процесса разработки алгоритм поиска векторов похожести был раз-

бит на следующие части:

- 1) поиск начального приближения вектора похожести;
- 2) уточняющий поиск;
- 3) завершающая обработка найденных векторов похожести.

Этап поиска начального приближения вектора похожести очень важен. Так как оценочная функция не является выпуклой и имеет множество оврагов и локальных минимумов, применение метода локальной оптимизации будет не эффективно, если начальная точка поиска выбрана неудачно. Для определения начальной точки поиска применяется алгоритм поиска начального приближения вектора похожести.

Для определения точки начального приближения мы можем использовать найденные векторы похожести соседних блоков в текущем кадре, а так же векторы похожести из предыдущего кадра. Использование этой априорной информации и есть основная идея метода поиска начального приближения вектора похожести. На рис. 3 представлены блоки, векторы похожести которых используются для поиска начального приближения.



Рис. 3. Блоки, используемые для вычисления медианного вектора

Алгоритм нахождения вектора начального приближения:

Шаг 1. Векторы соседних блоков MB_1 , MB_2 , MB_3 (рис. 3) используются для определения так называемого медианного вектора. Вычисляем значение оценочной функции для медианного вектора. Сравниваем это значение с пороговой величиной T_1 , если полученное значение меньше чем T_1 переходим к уточняющему поиску, в противном случае переходим к шагу 2. Пороговая величина T_1 вычисляется исходя из значений оценочных функций соседних блоков по формуле:

$$T_1 = a_k \min(SAD_1, SAD_2, SAD_3, SAD_4) + b_k,$$

где $SAD_1, SAD_2, SAD_3, SAD_4$ – меры SAD , рассчитанные для блоков MB_1, MB_2, MB_3 соответственно.

Шаг 2. Составляем список векторов-кандидатов. В этот список добавляем найденные векторы соседних блоков в текущем и предыдущем кадрах. Так же в список добавляются интерполированные векторы: если в предыдущем кадре вектор похожести какого либо из блоков указывал в текущий блок, то этот вектор так же добавляется в список.

Шаг 3. Из списка векторов-кандидатов удаляются близкие векторы. Для всех векторов попарно рассчитывается величина $d(V^1, V^2) = |V_x^1 - V_x^2| + |V_y^1 - V_y^2|$. Если $d(V^n, V^m) < T_2$, то вектор V^m исключается из списка векторов-кандидатов. Экспериментально было получено оптимальное значение параметра T_2 , равное 4.

Шаг 4. Вычисление оценочной функции производится для всех оставшихся векторов-кандидатов. Вектор-кандидат с наименьшим значением оценочной функции принимается за вектор начального приближения.

Шаг 5. Если значение оценочной функции в точке начального приближения больше чем $T_3 = 2T_1$, то принимаем решение о том, что точка начального приближения найдена неудачно и производим её адаптивный поиск методом сканирования в зависимости от размеров поискового окна и разрешения изображения.

В результате работы данного алгоритма мы получаем точку начального приближения. На следующем этапе происходит уточнение найденного векторы похожести.

Для выбора метода локальной оптимизации при помощи программной системы «MEFramework» был проведён сравнительный анализ следующих методов: симплексный метод, Гаусса-Зейделя (покоординатного спуска), Хука-Дживса, градиентный метод и метод поиска по алмазу. По результатам сравнительного анализа был сделан вывод о том, что сравниваемые методы локальной оптимизации позволяют достичь близкой степени компрессии. Однако градиентный метод требует меньшего количества вызовов оценочной функции.

Градиентный метод сводится к следующему итерационному процессу:

Шаг 1. В точке V_k вычисляется градиент оценочной функции. Вычисление градиента оценочной функции в зависимости от ситуации производится методом прямой (упреждающей), либо обратной (отстающей) конечной разности по формулам:

$$\nabla f(x, y) = \{f(x+1, y) - f(x, y), f(x, y+1) - f(x, y)\},$$

$$\nabla f(x, y) = \{f(x, y) - f(x-1, y), f(x, y) - f(x, y-1)\}.$$

Шаг 2. Принимаем $V^{k+1} = V^k - \lambda \frac{\nabla f(V^k)}{\|\nabla f(V^k)\|}$, вычисляем $f(V^{k+1})$.

Шаг 3. Если $f(V^{k+1}) < f(V^k)$, принимаем V^{k+1} за текущую точку и переходим к шагу 1, иначе уменьшаем λ на единицу и переходим к шагу 4.

Шаг 4. Если параметр $\lambda > 1$, переходим к шагу 2, иначе заканчиваем поиск, результатом поиска является вектор похоти V^k .

Автором предложено производить уточняющий обратный проход после нахождения векторов похоти для всех блоков кадра. На этом этапе проходим все блоки кадра в обратном порядке. По сравнению с первым проходом появляется информация о векторах похоти блоков расположенных ниже и правее текущего блока. Проверяем значения оценочных функций для этих векторов. Если значение оценочной функции меньше чем для найденного на первом проходе вектора похоти, проводим уточняющий поиск для вектора с наименьшим значением оценочной функции. Использование обратного прохода позволяет увеличить степень компрессии за счёт нахождения более точных значений векторов похоти.

Для проведения сравнительного анализа с помощью программной системы «MEFramework» автором были реализованы следующие алгоритмы поиска векторов похоти: *NTTS*, *CBA*, *ADZS*, *PMVFAST*, *EPZS*. По совокупности критериев алгоритм *EPZS* оказался наиболее эффективным алгоритмом поиска векторов похоти. Использование этого алгоритма позволяет достичь высокой степени компрессии при низком количестве вычислений оценочной функции.

Произведено сравнение предложенного алгоритма с алгоритмом *EPZS*. Результаты сравнения, алгоритмов представлены на рис. 4, 5.

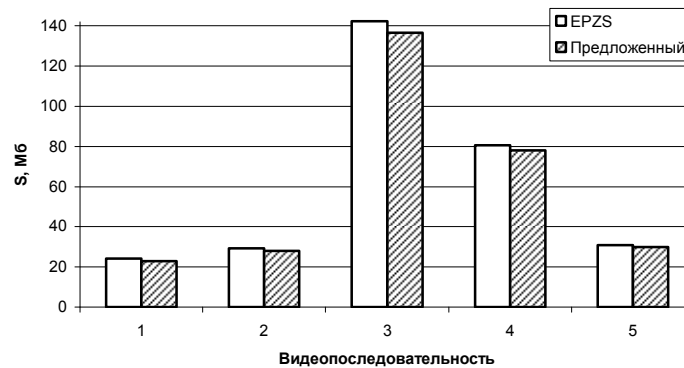


Рис. 4. Значения меры S для тестируемых алгоритмов, Мбайт

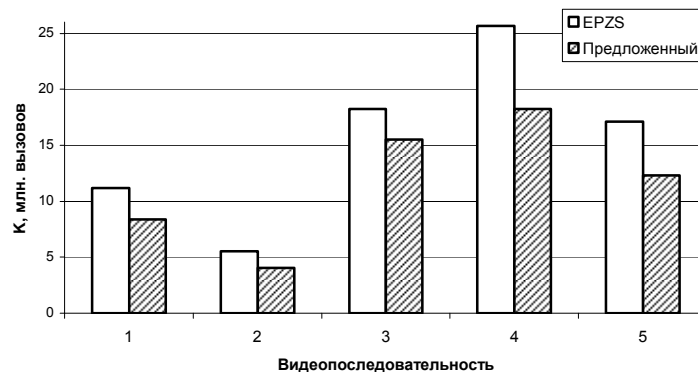


Рис. 5. Значения меры K для тестируемых алгоритмов, млн. вызовов

По результатам сравнения алгоритмов сделаны следующие выводы: использование предложенного алгоритма поиска векторов похоти позволяет достичь увеличения степени компрессии по сравнению с алгоритмом *EPZS* на 1–2%. При этом, предложенный алгоритм требует на 15–25% меньшего количества вычислений оценочной функции чем алгоритм *EPZS*. Таким образом, предложенный алгоритм поиска векторов похоти имеет меньшую вычислительную сложность и его использование позволит увеличить производительность компрессора при сохранении степени сжатия.

Использование оценочных функции *SAD* не позволяет учесть количество бит, требуемое для кодирования векторов похоти. При сжатии с большими коэффициентами квантования количество бит, требуемое для кодирования векторов, становится сравнимо с количеством бит, требуемым для кодирования невязки пикселей блока. Для таких случаев очень важно чтобы оценочная функция позволяла учесть не только энергию невязки между опорным и текущим блоком, но и количество бит, требуемое для кодирования вектора похоти. Использование такой оценочной функции позволит увеличить степень компрессии при сжатии с большим коэффициентом квантования.

В стандарте сжатия *MPEG-2* кодированию подвергаются не сами значения векторов похоти, а их разница с вектором похоти предыдущего блока. Используя в качестве функции оценки энергии невязки блока функцию *SAD*, запишем новую оценочную функцию в виде:

$$Cost(V_x^i, V_y^i) = SAD(V_x^i, V_y^i) + k_t I(|V_x^i - V_x^{i-1}| + |V_y^i - V_y^{i-1}|),$$

где (V_x^i, V_y^i) – вектор похожести текущего блока, (V_x^{i-1}, V_y^{i-1}) – вектор похожести предыдущего блока, k – коэффициент для учёта степени компрессии, достигаемой для видеопоследовательности, l – коэффициент, соответствующий среднему количеству бит, требуемому для кодирования единицы длины вектора похожести. Экспериментально было получено значение $l=0.3$.

Так как разные видеопоследовательности сжимаются с разной степенью компрессии, к тому же при сжатии могут использоваться различные коэффициенты квантования, использовать фиксированный коэффициент k для всех случаев невозможно. Поэтому коэффициент k рассчитывается адаптивно. Исходя из вида зависимости количества бит s , требуемых для кодирования блоков, от величины оценочной функции SAD , предложено использовать линейную аппроксимацию зависимости $s(SAD)$. В этом случае коэффициент k для t -го кадра вычисляется, используя результаты компрессии предыдущего кадра согласно формуле:

$$k_t = \frac{\sum_{i=1}^M SAD_{t-1}^i}{\sum_{i=1}^M s_{t-1}^i},$$

где M – количество блоков в кадре, s_{t-1}^i – количество бит, потребовавшееся для кодирования i -го блока в $(t-1)$ -м кадре, SAD_{t-1}^i – значения меры SAD , рассчитанной для i -го блока в $(t-1)$ -м кадре.

Произведено сравнение предложенной оценочной функции $Cost$ с оценочной функцией SAD на тестовых видеопоследовательностях. В эксперименте использовались различные значения коэффициентов квантования. Результаты сравнения для пяти тестовых видеопоследовательностей представлены в табл. 1.

Таблица 1. Размер видеопоследовательностей, сжатых с применением сравниваемых оценочных функций, Мбайт

№	Оценочная функция	Коэффициент квантования			
		3	10	20	30
1	$Cost$	23,379	7,364	3,7	2,698
	SAD	23,616	7,671	4,025	3,066
2	$Cost$	27,204	9,924	4,479	2,798
	SAD	27,445	10,338	4,863	3,144
3	$Cost$	132,24	40,814	16,904	10,942
	SAD	132,248	41,974	17,689	11,413
4	$Cost$	70,01	19,779	8,477	5,958
	SAD	70,602	20,708	9,323	6,76
5	$Cost$	28,079	7,105	3,603	2,839
	SAD	28,383	7,612	3,982	3,204

Анализ данных из табл. 1 позволяет сделать вывод о том, что в сравнении с функцией SAD использование предложенной оценочной функции увеличивает степень компрессии на 3-5% для коэффициента квантования 10, на 5-9% для параметра коэффициента 20 и на 5-12% для коэффициента квантования 30.

Разработанный алгоритм поиска векторов похожести и новая оценочная функция были встроены в реальное приложение сжатия видеоданных, компрессор «MainConcept MPEG-2 Video Encoder». Для проведения сравнительного анализа была реализована версия компрессора, использующего алгоритм $EPZS$. В качестве оценочной функции для алгоритма $EPZS$ использовалась функция SAD . При проведении сравнения эффективности алгоритмов поиска векторов похожести тестовые видеопоследовательности сжимались с постоянным битрейтом. Для оценки качества найденных векторов использовалась общепризнанная мера оценки качества сжатых видеоданных $PSNR$, рассчитанная для декодированного и оригинального кадров. Для оценки вычислительной сложности алгоритмов сравнивалось время, требуемое для сжатия видеопоследовательностей. Результаты сравнения представлены на рис. 6, 7.

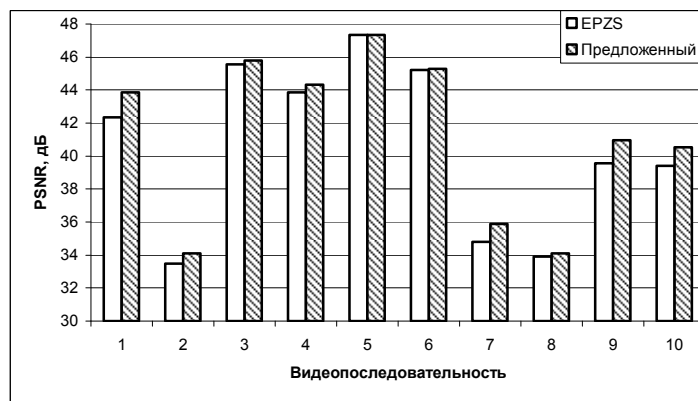


Рис. 6. Значения меры $PSNR$ для сравниваемых алгоритмов поиска векторов похожести, дБ

В результате анализа данных, представленных на рис. 10, рис. 11 сделаны следующие выводы:

1) использование предложенного алгоритма поиска векторов схожести и новой оценочной функции при сжатии видеоданных позволяет увеличить значение меры качества $PSNR$ по сравнению со значениями, достигаемыми при использовании алгоритма $EPZS$;

2) для видеопоследовательностей с низким разрешением (1, 2, 3) время, затраченное на сжатие с применением предложенного алгоритма и алгоритма $EPZS$ практически одинаковое. Это объясняется тем, что эти видеопоследовательности не содержат длинных векторов схожести, и поиск часто останавливается на предварительном этапе;

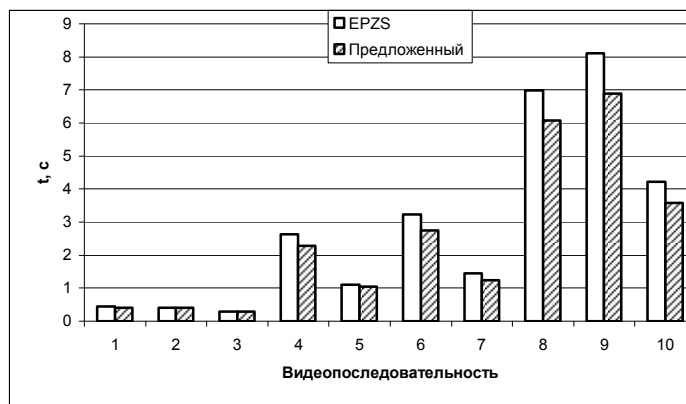


Рис. 7. Время, требуемое для сжатия с применением сравниваемых алгоритмов, с

3) для видеопоследовательностей 4–10 использование предложенного алгоритма поиска векторов схожести позволяет уменьшить общее время компрессии на 5–15%, что является ощутимым достижением оптимизации компрессора видеоданных.

В заключении перечислены основные результаты, полученные в диссертационной работе.

1. Разработана и реализована программная система «MEFramework», позволяющая количественно оценивать эффективность алгоритма поиска векторов схожести на основе предложенных критериев.

2. Произведён анализ оценочных функций SAD , SSD , $M1$. Степень компрессии с использованием функции $M1$ оказалась самой высокой, однако вычислительная сложность данной функции не позволяет использовать её в реальных приложениях сжатия видеоданных. Функции SAD и SSD позволили достичь схожей степени компрессии, однако в силу того, что функция SSD является вычислительно более сложной, в дальнейших исследованиях было решено использовать функцию SAD .

3. Проведен сравнительный анализ эффективности известных алгоритмов поиска векторов схожести, для сжатия видеоданных. По результатам экспериментов был сделан вывод о том, что лучшим среди доступных аналогов по совокупности критериев является алгоритм $EPZS$. По этой причине $EPZS$ использовался для сравнения с разработанным алгоритмом.

4. Проведён сравнительный анализ пяти алгоритмов поиска экстремума: симплексный метод, Гаусса-Зейделя, Хука-Дживса, градиентный метод, метод поиска по алмазу. Лучшим среди тестируемых методов оказался градиентный метод.

5. Разработан оригинальный алгоритм поиска векторов схожести. Для нахождения точки начального приближения алгоритм осуществляет поиск среди расширенного набора векторов-кандидатов. В случае если на предыдущем этапе точка начального приближения найдена неудачно, используется метод сканирования. Для уточнения точки начального приближения применяется градиентный метод локальной оптимизации. По окончании поиска векторов схожести для всего кадра производится обратный проход по всем блокам с целью уточнения найденных векторов схожести с учетом новых векторов-кандидатов.

6. Произведено сравнение разработанного алгоритма поиска векторов схожести с алгоритмом $EPZS$. Использование предложенного алгоритма при сжатии видеоданных позволяет уменьшить количество вычислений оценочной функции по сравнению с алгоритмом $EPZS$ на 15-25% при той же степени компрессии.

7. Предложена новая оценочная функция. В отличие от широко используемых функций SAD и SSD предложенная функция позволяет учитывать не только энергию невязки блоков, но и количество бит, требуемое для кодирования векторов схожести. Использование новой оценочной функции позволяет увеличить степень компрессии видеоданных.

8. Произведено сравнение разработанного алгоритма поиска векторов схожести, использующего новую оценочную функцию с алгоритмом $EPZS$ в реальном приложении сжатия видеоданных. Использование предложенного алгоритма поиска векторов схожести позволяет уменьшить общее время компрессии по сравнению с результатами алгоритма $EPZS$, на 5-15%, при незначительном улучшении качества.

9. Алгоритм поиска векторов схожести, разработанный в рамках данной диссертационной работы, внедрен в коммерческий продукт компании Mainconcept AG, Mainconcept MPEG-2 Video Encoder, что подтверждается актом о внедрении.

10. Программная система «MEFramework» зарегистрирована в Отраслевом фонде алгоритмов и программ (ОФАП) и используется компанией ООО «МэйнКонцепт-Дивикс» для проведения внутренних испытаний алгоритмов поиска векторов схожести.

В приложениях к диссертационной работе представлен список используемых обозначений и сокращений, описание интерфейса программной системы на языке «C++», а также акт о внедрении.

Публикации по основным результатам диссертации:

1. Потапов П.В. Оптимизация распределения ресурсов при кодировании видеоданных // Информационные системы. – Томск: ТУСУР, 2004. – № 3. – С. 129 – 134.
2. Потапов П.В. Оптимальное распределение ресурсов при сжатии видеоданных с использованием двухпроходных алгоритмов // Материалы всероссийской научной конференции студентов-физиков. – Екатеринбург: АСФ, 2005. – С. 251 – 254.
3. Потапов П.В. Оптимальное распределение ресурсов при сжатии видеоданных с использованием двухпроходного алгоритма // Материалы XLIII международной научной студенческой конференции «студент и научно-технический прогресс». – Новосибирск: НГУ, 2005. – С. 45 – 46.
4. Потапов П.В. Применение двухпроходного алгоритма для распределения ресурсов при сжатии видеоданных // Материалы всероссийской научно-технической конференции студентов и молодых ученых «Научная сессия ТУСУР – 2005». – Томск: ТУСУР, 2005. – С. 65 – 68.
5. Потапов П.В. Двухпроходный режим компрессии видеоданных // Доклады Томского государственного университета систем управления и радиоэлектроники. – Томск, 2005. – Т. 11. – № 3. – С. 77 – 81.
6. Потапов П.В. Двухпроходные алгоритмы распределения ресурсов при сжатии видеоданных // Информационные системы. – Томск: ТУСУР, 2006. – № 4. – С. 10 – 18.
7. Корилов А.М., Потапов П.В. Программная среда для разработки и исследования алгоритмов оценки движения при сжатии видеоданных // Вычислительные технологии. – Томск, 2007. – Т. 12. – Спец выпуск 1. – С. 42 – 50.
8. Потапов П.В. Разработка программной среды для сравнения алгоритмов поиска видеоданных. // Материалы докладов Всероссийской Научно-Технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР-2007». – Томск, 2007. – С. 189 – 192.
9. Потапов П.В. Использование методов локальной оптимизации для поиска векторов схожести при сжатии видеоданных // Сб. науч. докл. Международной науч.-практ. конф. «Электронные системы и средства управления». – Томск, 2008. – С. 43 – 47.
10. Потапов П.В. Программная система моделирования поиска векторов схожести при сжатии видеоданных «MEFramework» // Отраслевой фонд алгоритмов и программ. Свидетельство об отраслевой регистрации разработки № 11657. Номер государственной регистрации: 50200802145. – 2008.