

УДК 519.688

А.М. Кох

Межкадровый параллельный алгоритм обработки цифрового видеопотока

Предложен новый алгоритм декодирования цифровых видеоданных, в котором использован параллелизм на уровне кадров в сочетании с синхронизацией реконструкции опорных и предсказанных блоков видеоинформации. Алгоритм в два раза производительнее существующих аналогов для видео стандартного разрешения и на треть для видео высокой четкости.

Ключевые слова: цифровое видео, параллельные алгоритмы, компенсация движения, синхронизация, MPEG, VC-1.

Введение

Семейство MPEG-форматов, состоящее из MPEG-2, MPEG-4, H.264 (AVC) и VC-1, является наиболее распространенным, в связи с чем задача повышения производительности алгоритмов сжатия и декодирования, используемых в этих форматах, остается актуальной.

В настоящее время прогресс в области микроэлектронных компонент сталкивается с ограничениями, связанными с фундаментальными законами природы. Вряд ли можно надеяться на то, что в ближайшее время основной прогресс в быстродействии электронно-вычислительных машин будет достигнут лишь за счет совершенствования их элементной базы. Переход на качественно новый уровень производительности потребовал от разработчиков ЭВМ и новых архитектурных решений, поэтому дальнейшее уменьшение времени переключения электронных схем не позволит существенно повысить производительность ЭВМ. В этих условиях требования по дальнейшему повышению быстродействия ЭВМ могут быть удовлетворены только путем распространения принципа параллелизма на сами устройства обработки информации и создания многомашинных и многопроцессорных вычислительных систем.

С увеличением разрешения и битрейта цифрового видео (до 1920×1080 с битрейтом до 40 Мбит/с для видео высокого разрешения [1]) применение алгоритмов распараллеливания для обработки цифровой видеоинформации в настоящее время выходит на передний план, позволяя использовать все возможности и преимущества многопроцессорной техники.

Изображения в MPEG-последовательности подразделяются на два типа: *intra* (I-кадры), являющиеся опорными, и *inter* (P- и B-кадры), содержащие разность текущего изображения с предыдущим опорным кадром [2].

Отдельные изображения состоят из макроблоков. Макроблок – основная структурная единица фрагментации изображения, он соответствует участку изображения размером 16×16 пикселей. Каждый макроблок состоит из нескольких блоков, количество которых определяется форматом сжатия. Блоки являются базовыми структурными единицами, над которыми осуществляются основные операции кодирования [3]. Группы блоков объединяются в прямоугольные области, называемые слайсами. Отдельные слайсы, формирующие изображение, могут быть декодированы независимо друг от друга. Векторы движения определяют расстояние между двумя фрагментами на экране, основываясь на количестве пикселей между этими областями.

Общая схема декодирования цифровой видеоинформации для большинства современных MPEG-форматов представлена на рис. 1.

Параллельные алгоритмы, используемые в обработке цифрового видео

Существуют два основных способа распараллеливания процесса декодирования цифровой видеоинформации – меж- и внутрикадровый, каждый из которых обладает своими преимуществами и недостатками.

Недостатком внутрикадрового параллелизма является необходимость полного предварительного восстановления всей сжатой информации, относящейся к кадру, т.к. сам процесс энтропийного декодирования не подразумевает распараллеливания. Помимо этого, для многих стадий, таких как предсказание AC коэффициентов [4], фильтрации изображения с целью устранения блочности, необходимо наличие соседних блоков (левого, верхнего и правого верхнего), что приводит к созданию достаточно сложной схемы син-

хронизации. Как правило, при наличии нескольких процессоров сначала выполняется энтропийное декодирование нескольких кадров, после чего реконструкция каждого кадра выполняется диагональным методом, когда декодирование очередной строки макроблоков начинается после восстановления правого верхнего соседа первого блока в текущей строке с последующей синхронизацией с верхней строкой.

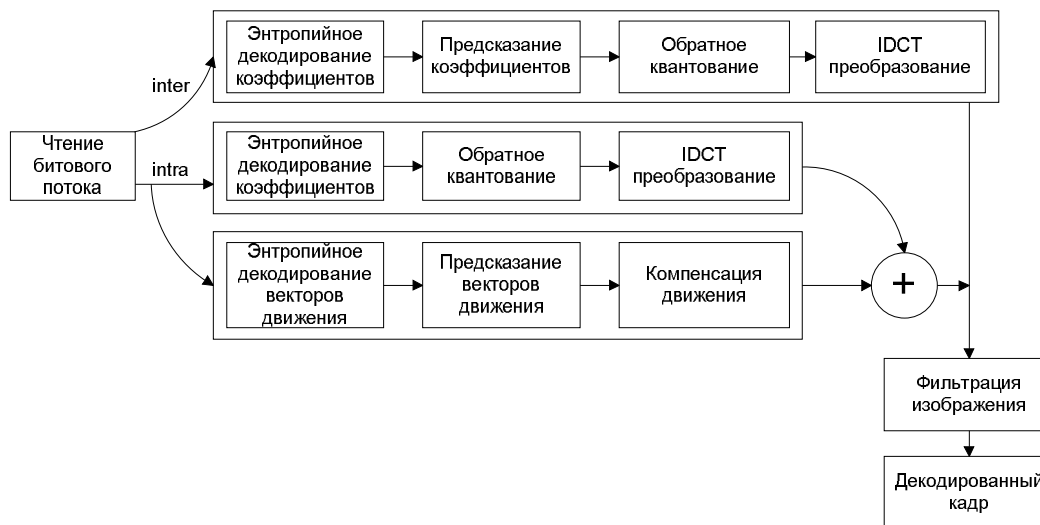


Рис. 1. Декодирование MPEG-последовательности

Еще одной разновидностью внутрикадровых методов является параллелизм на уровне слайсов, когда каждый слайс изображения обрабатывается отдельным потоком. Серьезным недостатком данного метода является высокая зависимость от структуры видеопотока, т.к. зачастую кадры в видеопоследовательности состоят из одного слайса, в этом случае данный метод не дает прироста производительности.

Основной проблемой межкадрового распараллеливания является зависимость между *intra*- и *inter*-кадрами, т.е. для декодирования *P* и *B* изображений необходимо предварительно реконструировать соответствующие им опорные кадры.

В предложенном методе выполняется одновременное декодирование нескольких кадров, число которых зависит от количества имеющихся в системе процессоров. Одним из достоинств данного метода является использование синхронизации только на стадии компенсации движения, все остальные стадии выполняются независимо. Синхронизация необходима в случае, когда векторы движения в *inter*-кадре указывают на еще не восстановленную область соответствующего опорного кадра. В случае же декодирования видеопоследовательностей, состоящих исключительно из *intra*-кадров, их восстановление производится полностью независимо друг от друга.

Декодирование каждого кадра происходит в отдельном потоке, содержащем, помимо всего прочего, массив системных примитивов синхронизации (событий в случае платформы Win32), каждый элемент которого соответствует строке макроблоков в кадре. При завершении декодирования строки соответствующее ей событие помечается как активное, что означает ее пригодность для восстановления *inter*-кадров, векторы движения которых указывают на данную область изображения. В случае если вектор движения в *P* или *B* кадре указывает на еще не реконструированную область опорного кадра, производится ожидание объекта-события соответствующей строки макроблоков в опорном кадре. На рис. 2 приведен пример параллельного декодирования трех кадров – *I*, *P* и *B* с использованием предложенного алгоритма, стрелками сверху обозначены зависимости между *inter*- и *intra*-кадрами, стрелками между блоками – векторы движения.

Один из векторов движения во втором блоке *B*-кадра указывает на еще не восстановленную строку в опорном *P*-кадре, в связи с чем декодирование данного блока будет приостановлено до тех пор, пока не будет реконструирована соответствующая строка в *P* кадре. Как видно из представленной схемы, наибольшая потеря производительности, связанная с обеспечением синхронизации, будет в случае, когда движение сцены в кадре будет происходить снизу вверх, т.е. векторы движения в *inter*-кадрах будут указывать вниз на еще не восстановленные области в соответствующих им опорных кадрах.

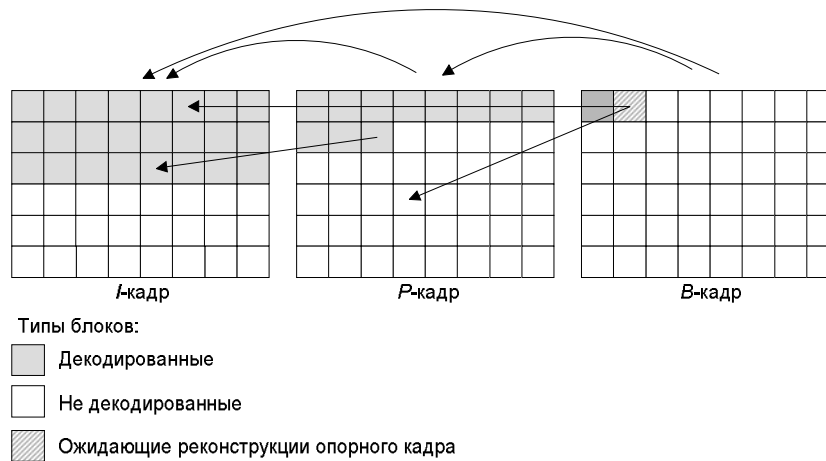


Рис. 2. Схема алгоритма

Тестирование, результаты

Помимо более высокой скорости декодирования по сравнению с внутрикадровым распараллеливанием, данный способ является алгоритмически более простым, т.к. не требует модификации существующих алгоритмов реконструкции и фильтрации блоков. Также преимуществом предложенного метода является высокий коэффициент масштабирования, полученный путем сравнения производительности алгоритма при использовании 1, 2, 4 и 8 процессоров. Замеры производились на системе с двумя четырехъядерными процессорами Intel Xeon E5420 с частотой 2,5 ГГц в операционной системе Windows XP Professional 32-bit, для получения точных данных операционная система запускалась с поддержкой 1, 2, 4 и 8 процессоров соответственно. Результат в случае использования 2 процессоров составил 1,21 для внутрикадрового и 1,91 для межкадрового параллелизма, для 4 – 1,53 и 3,54, для 8 – 1,61 и 5,68 соответственно, из чего можно сделать вывод о высокой степени масштабируемости предложенного метода.

Также для оценки эффективности меж- и внутрикадровых алгоритмов, используемых при декодировании цифровых видеопотоков, были разработаны программные модули, реализующие математические модели данных методов с учетом задержек, связанных с синхронизацией потоков. Входными данными для моделей являются два массива, содержащих значения, соответствующие времени энтропийного кодирования и реконструкции каждого блока, а также количество потоков, используемых алгоритмом. Модели позволяют оценить пиковую теоретическую эффективность каждого из алгоритмов, т.к. единственным учитываемым критерием, помимо непосредственно самого времени, затрачиваемого на декодирование блоков, является использование примитивов синхронизации. Главным критерием оценки эффективности каждого метода считался коэффициент параллелизма. Времена для моделей были получены с использованием программного декодера MainConcept VC-1 Video Decoder [5] на системе с процессором Intel Core 2 Quad Q9300, для точных измерений использовалась ассемблерная инструкция платформы $\times 86$ *rdtsc* [6], читающая счётчик TSC (Time Stamp Counter) и возвращающая в регистрах EDX:EAX 64-битное количество тактов с момента последнего сброса процессора. Для получения статистики в декодере перед выполнением функций энтропийного декодирования и реконструкции макроблока сохранялось значение, возвращаемое инструкцией *rdtsc*, также получалось значение счетчика тактов после выполнения данных функций, временем выполнения функций декодирования и реконструкции является разность полученных значений.

С помощью разработанных моделей были получены коэффициенты параллелизма для обоих методов с использованием двух и четырех потоков. Среднее значение для двух потоков в случае использования внутрикадрового параллелизма составило 1,91, для межкадрового параллелизма – 1,98, для четырех потоков – 3,71 и 3,92 соответственно. Более высокий коэффициент параллелизма в случае использования межкадрового метода распараллеливания подтверждает высокую эффективность данного метода.

Предложенный метод был реализован в программном компоненте MainConcept VC-1 Decoder, входящем в состав коммерческих продуктов MainConcept VC-1 и VC-1 Pro. Ранее данный компонент использовал внутрикадровые методы распараллеливания.

Результаты сравнения производительности декодеров формата VC-1, использующих меж- и внутрикадровые алгоритмы распараллеливания, приведены в таблице. Для тести-

рования использовались три видеопоследовательности с разрешением 704×576, 1280×1280 и 1920×1080 пикселей, закодированные с различными битрейтами. Помимо сравнения компонентов MainConcept VC-1 Decoder с различными методами распараллеливания было проведено их сравнительное тестирование с программным компонентом Microsoft WM Video Decoder DMO, входящим в состав Windows Media 11. Тесты проводились на двухъядерном процессоре Pentium D820 с частотой 2,66 ГГц.

Производительность сравниваемых меж- и внутрикадровых алгоритмов распараллеливания, кадров в секунду

| Разрешение, битрейт | MS WM Video Decoder DMO | Параллелизм | |
|----------------------|-------------------------|----------------|-------------|
| | | Внутрикадровый | Межкадровый |
| 704×576, 1 Мбит/с | 85,4 | 145,0 | 188,7 |
| 704×576, 3 Мбит/с | 76,2 | 123,8 | 162,9 |
| 704×576, 5 Мбит/с | 71,2 | 116,3 | 149 |
| 1280×720, 5 Мбит/с | 66,4 | 58,6 | 77,4 |
| 1280×720, 10 Мбит/с | 57,4 | 52,4 | 70,5 |
| 1280×720, 15 Мбит/с | 52,7 | 48,9 | 65,1 |
| 1920×1080, 10 Мбит/с | 28,7 | 26,5 | 32,4 |
| 1920×1080, 20 Мбит/с | 25,3 | 24,6 | 27,3 |
| 1920×1080, 40 Мбит/с | 22,1 | 21,4 | 24,2 |

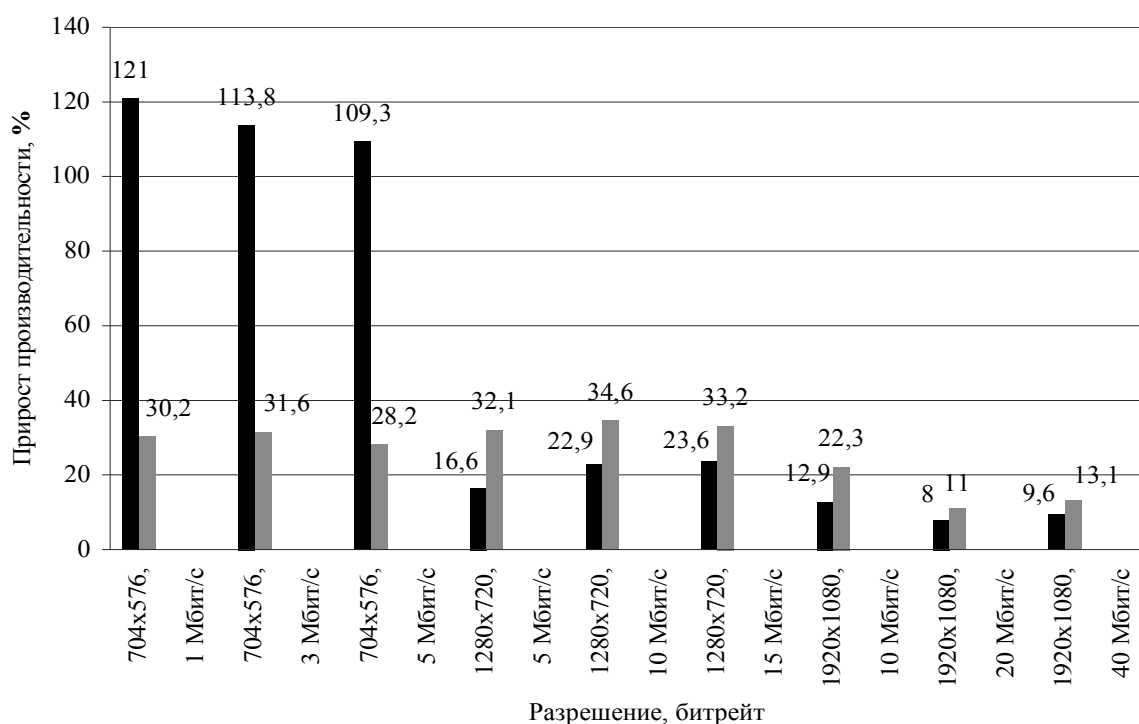


Рис. 3. Прирост производительности по сравнению с аналогами, темным цветом показан MS DMO Video decoder, светлым – MainConcept VC-1 Decoder, использующий внутрикадровый параллелизм

Анализ данных из таблицы и рис. 3 позволяет сделать вывод о том, что в сравнении с алгоритмами внутрикадрового распараллеливания предложенный метод дает прирост производительности 8–121% в зависимости от разрешения видеопоследовательности, а также от ее битрейта.

Заключение

Предложен новый алгоритм декодирования цифрового видеопотока, основная идея которого заключается в применении распараллеливания вычислений в сочетании с синхронизацией реконструкции опорных и предсказанных блоков видеoinформации. Преимуществами данного метода являются использование синхронизации только на стадии компенсации движения и полностью независимое декодирование *intra*-кадров, что обес-

печивает высокий коэффициент параллелизма, равный 1,91 для 2 процессоров, 3,51 для 4 и 5,68 для 8-процессорных систем, разработанные математические модели параллельных алгоритмов подтвердили высокий коэффициент параллелизма предложенного метода.

Применение предложенного подхода позволяет существенно повысить скорость декодирования цифрового видеопотока. Проведенный анализ показал, что данный метод практически в два раза производительнее известных аналогов для видео стандартного разрешения и на треть для видео высокой четкости.

Литература

1. Blu-Ray FAQ [Электронный ресурс]. – Режим доступа: <http://www.blu-ray.com/faq/>, свободный (дата обращения: 21.10.2009).
2. ISO/IEC 11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video. Международный стандарт, 1993.
3. Lee Jae-Beom. The VC-1 and H.264 video compression standards for broadband video services / Jae-Beom Lee, Hari Kalva. – New York: Springer, 2008. – 496 p.
4. SMPTE 421M-2006: VC-1 compressed video bitstream format and decoding process. – SMPTE, 2006.
5. MainConcept VC-1 Pro [Электронный ресурс]. – Режим доступа: <http://www.mainconcept.com/site/prosumer-products-4/vc-1-pro-20250/information-20270.html>, свободный (дата обращения: 07.02.2010).
6. Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 2B: Instruction Set Reference, N-Z [Электронный ресурс]. – Режим доступа: <http://developer.intel.com/Assets/PDF/manual/253667.pdf>, свободный (дата обращения: 18.12.2009).

Кох Александр Михайлович

Аспирант каф. экономической математики, информатики и статистики ТУСУРа

Тел.: 8-913-8249273

Эл. почта: alexander.koh@mainconcept.ru

Koh A.M.

Parallel algorithm for digital video stream decoding

Modern techniques of the parallel digital video decoding are reviewed in the paper. The author's parallel decoding method and the developed software complex are described.

Keywords: digital video, parallel algorithms, MPEG, VC-1.
