

СПИСОК ЛИТЕРАТУРЫ

1. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. – Новосибирск: Изд-во НГТУ, 2000. – 168 с.
2. Harker M., O'Leary P. Computation of Homographies // Austria, Leoben: Tech. Rep., Dep. of Product Engineering, Univ. of Leoben, 2005. – 10 p.
3. Zhengyou Z. A Flexible New Technique for Camera Calibration // Microsoft Research Tech. Rep. MS-98-71. – 1998. – 21 p.
4. Shi J., Tomasi C. Good Features to Track // Proc. IEEE Conf. on Computer Vision and Pattern Recognition. – Seattle, WA, USA, 1994. – P. 593–600.
5. Lucas B.D., Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision // International Joint Conf. on Artificial Intelligence. – 1981. – P. 674–679.
6. Tomasi C., Kanade T. Detection and Tracking of Point Features // Carnegie Mellon Univ., Tech. Rep. CMU-CS-91-132. – 1991. – 20 p.
7. Jin H., Favaro P., Soatto S. Real-time tracking and outlier rejection with changes in illumination // Proc. IEEE Intern. Conf. on Computer Vision. – 2001. – P. 684–689.
8. Hartley R.I. Self-Calibration of Stationary Cameras // Intern. J. of Computer Vision. – 1997. – V. 22. – № 1. – P. 5–23.
9. Методы компьютерной обработки изображений / под ред. В.А. Сойфера. – М.: Физматлит, 2003. – 784 с.
10. Van Aken J.R. An Efficient Ellipse Drawing Algorithm // Computer Graphics & Artificial Intelligence. – 1984. – V. 4. – № 9. – P. 24–35.

Поступила 03.04.2010 г.

УДК 519.688

АЛГОРИТМ РАСПАРАЛЛЕЛИВАНИЯ ОБРАБОТКИ ЦИФРОВОГО ВИДЕОПОТОКА

А.М. Кох

Томский университет систем управления и радиоэлектроники
E-mail: alexander.koh@maiconcept.ru

Предложен новый алгоритм декодирования цифрового видеопотока, в котором использовано распараллеливание вычислений в сочетании с синхронизацией реконструкции опорных и предсказанных блоков видеoinформации. Алгоритм в два раза производительнее существующих аналогов для видеопотоков стандартного разрешения и на треть для видеопотоков высокой четкости.

Ключевые слова:

Цифровое видео, параллельные алгоритмы, компенсация движения, синхронизация, MPEG, VC-1.

Key words:

Digital video, parallel algorithms, motion compensation, synchronization, MPEG, VC-1.

Семейство MPEG-форматов, состоящее из MPEG-2, MPEG-4, H.264 (AVC) и VC-1, является наиболее распространенным, в связи с чем задача разработки модификаций, направленных на повышение производительности алгоритмов сжатия и декодирования, остается актуальной.

С увеличением разрешения и битрейта цифрового видеопотока (до 1920×1080 с битрейтом до 40 Мбит/с для видеопотока высокого разрешения) применение алгоритмов распараллеливания для обработки цифровой видеoinформации выходит на передний план, позволяя использовать все возможности и преимущества многопроцессорной техники.

Известно, что изображения в MPEG-последовательности подразделяются на следующие типы [1]:

- *I (intra)*, играющие роль опорных при восстановлении остальных изображений по их разностям;
- *P (predicted)*, содержащие разность текущего изображения с предыдущим *I* или *predicted* с учетом смещений отдельных фрагментов;

- *B (bidirectionally predicted)*, содержащие разность текущего изображения с предыдущим и последующим изображениями типов *I* или *P* с учетом смещений отдельных фрагментов.

Отдельные изображения состоят из макроблоков. Макроблок – основная структурная единица фрагментации изображения, он соответствует участку изображения размером 16×16 пикселей. Для них определяются векторы движения относительно *I*- или *P*-изображений. В свою очередь каждый макроблок состоит из нескольких блоков, количество которых определяется форматом сжатия, несущих информацию о яркости *Y*, цветовых *U*- и *V*-компонентах. Блоки являются базовыми структурными единицами, над которыми осуществляются основные операции кодирования, в том числе выполняется дискретное косинусное преобразование (DCT – Discrete Cosine Transform) и квантование полученных коэффициентов [2]. Векторы движения определяют расстояние между двумя фрагментами на экране, основываясь на количестве пикселей между этими областями.

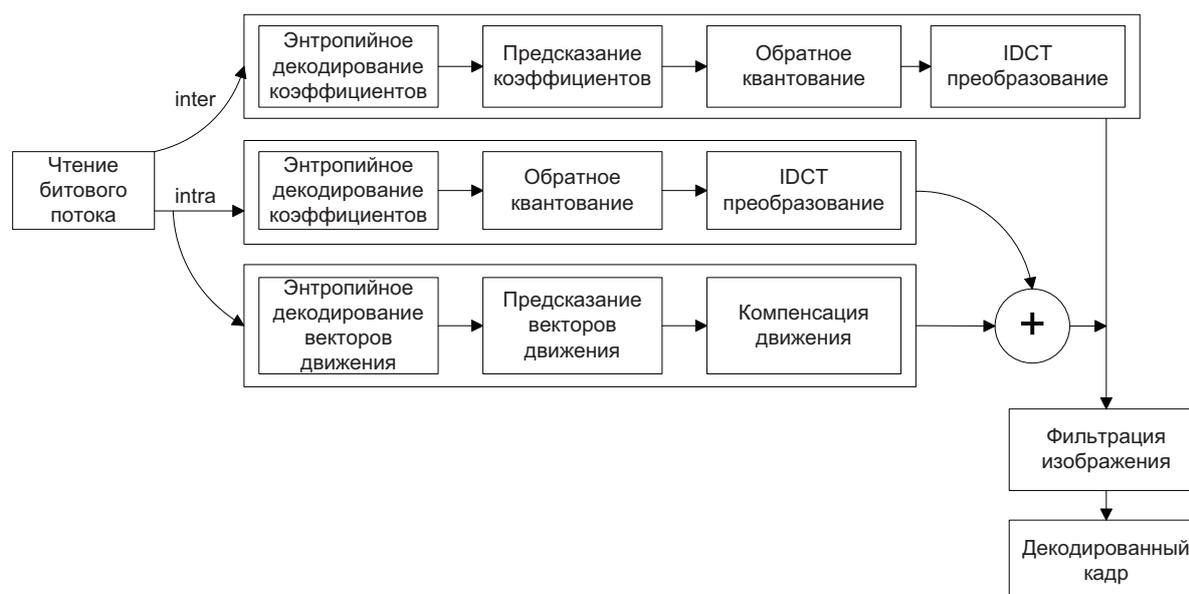


Рис. 1. Декодирование MPEG-последовательности

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

Существуют два основных способа распараллеливания процесса декодирования цифровой видеoinформации – 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 АС коэффициентов [3], фильтрации изображения с целью устранения блочности, необходимо наличие соседних блоков (левого, верхнего и правого верхнего), что приводит к созданию достаточно сложной схемы синхронизации. 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.

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 си-

стеме 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 декодирования видеопоследовательностей, состоящих исключительно из *intra*-кадров, их восстановление производится полностью независимо друг от друга.

Декодирование каждого кадра происходит в отдельном потоке, содержащем, помимо 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 предста-

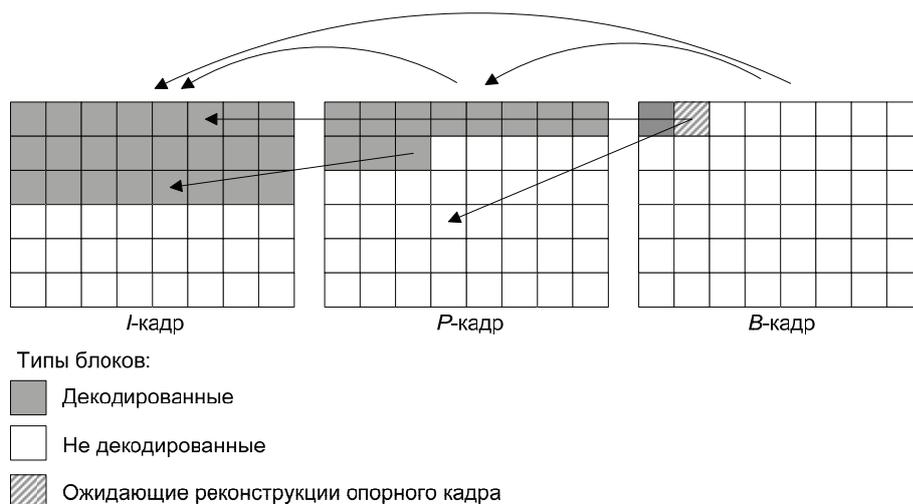


Рис. 2. Схема алгоритма

вленной 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. Пример данной ситуации изображен на рис. 3.

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-и процессоров. Замеры производились на системе с двумя четырехъядерными процессорами Intel Xe-

on 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-и процессоров соответственно. Результат в случае использования 2-х процессоров составил 1,21 для внутрикадрового и 1,91 для межкадрового параллелизма, для 4-х – 1,53 и 3,54, для 8-и – 1,61 и 5,68 соответственно, из чего можно сделать вывод о высокой степени масштабируемости предложенного метода.

Предложенный метод был реализован в программном компоненте MainConcept VC-1 Decoder, входящем в состав коммерческих продуктов MainConcept VC-1 и VC-1 Pro [4]. Ранее данный компонент использовал внутрикадровые методы распараллеливания.

Результаты сравнения производительности декодеров формата VC-1, использующих меж- и вну-

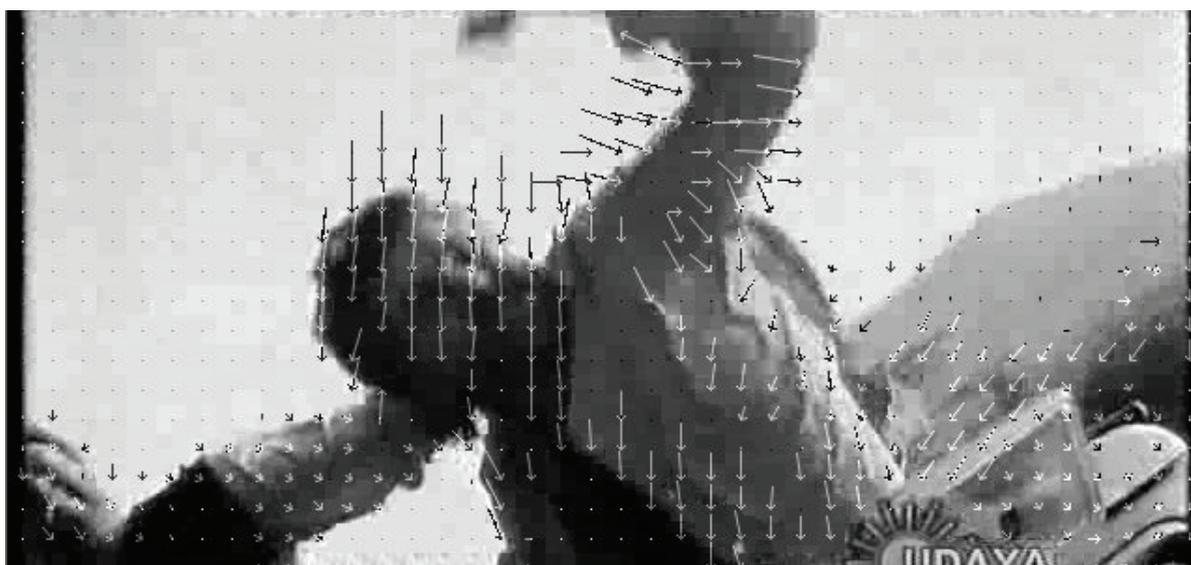


Рис. 3. Пример сцены с наибольшей потерей производительности (векторы движения направлены вниз)

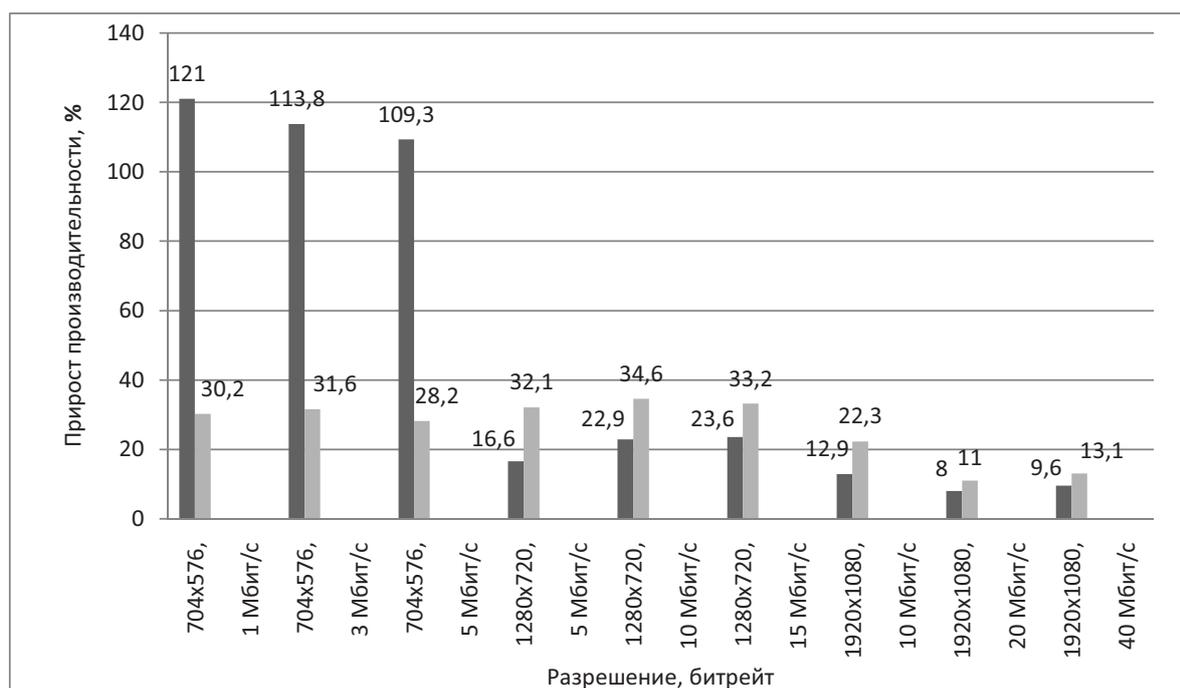


Рис 4. Прирост производительности по сравнению с аналогами, темным цветом – MS WM Video Decoder DMO, светлым – MainConcept VC-1 Decoder, использующий внутрикадровый параллелизм

трикадровые алгоритмы распараллеливания, приведены в таблице. Для тестирования 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, входящем в состав Windows Media 11. Тесты проводились на двухъядерном процессоре Pentium D820 с частотой 2,66 ГГц.

Таблица. 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

Анализ данных из таблицы и рис. 4 позволяет сделать вывод о том, что в сравнении с алгоритмами внутрикадрового распараллеливания предложенный метод дает прирост производительности 8...121 % в зависимости от разрешения видеопоследовательности и ее битрейта.

Выводы

Предложен новый алгоритм декодирования цифрового видеопотока, основная идея которого заключается в распараллеливании вычислений в сочетании с синхронизацией реконструкции опорных и предсказанных блоков видеoinформации. Преимуществами метода является использование синхронизации только на стадии компенсации движения и полностью независимое декодирование *intra*-кадров, что обеспечивает высокий коэффициент параллелизма, равный 1,91 для 2-х процессоров, 3,51 для 4-х и 5,68 для 8-и процессорных систем.

Применение предложенного подхода позволяет существенно повысить скорость декодирования цифрового видеопотока: метод практически в два раза производительнее известных аналогов (в частности, от Microsoft) для видеопотоков стандартного разрешения и на треть – для видеопотоков высокой четкости.

СПИСОК ЛИТЕРАТУРЫ

1. ISO/IEC 11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – P. 2: Video: англ. – ISO/IEC, 1993.
2. Lee J.-B., Kalva H. The VC-1 and H.264 video compression standards for broadband video services: англ. – N.Y.: Springer publishing, 2008. – 496 p.
3. SMPTE 421M-2006: VC-1 compressed video bitstream format and decoding process: англ. – SMPTE, 2006.
4. MainConcept VC-1 Pro // MainConcept: Information. 2009. URL: <http://www.mainconcept.com/site/prosumer-products-4/vc-1-pro-20250/information-20270.html> (дата обращения: 14.10.2009).

Поступила 12.02.2010 г.

УДК 65.012.122

ОПТИМИЗАЦИЯ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ С РЕЗЕРВНЫМ ПРИБОРОМ С УПРАВЛЕНИЕМ, ЗАВИСЯЩИМ ОТ ВРЕМЕНИ ОЖИДАНИЯ

Л.И. Самочернова

Томский политехнический университет
E-mail: am@am.tpu.ru

Изучена система обработки информации, функционирование которой описано математической моделью системы массового обслуживания с резервным прибором, управляемым по текущему времени ожидания заявки, находящейся первой в очереди. Проведена оптимизация системы при учете потерь на ожидание и амортизацию.

Ключевые слова:

Система, обслуживание, время ожидания, амортизация, оптимальный момент.

Key words:

System, service, queuing time, depreciation, optimal moment.

Введение

Значительное число работ [1–10] посвящено изучению управляемых систем массового обслуживания (УСМО). Это связано с тем, что именно УСМО удастся описывать функционирование многих реальных технических систем, в частности, систем связи. Во многих работах изучаются системы массового обслуживания (СМО), в которых интенсивность обслуживания, моменты включения и отключения резервного прибора зависят от числа заявок в системе или от длины очереди. Однако системы, в которых стратегия управления резервным прибором зависит от времени ожидания, остались изученными очень слабо.

Данная статья посвящена оптимизации системы обработки информации, функционирование которой описано математической моделью УСМО с резервным прибором, управляемым по текущему времени ожидания заявки, находящейся первой в очереди.

1. Описание системы

По некоторому каналу связи поступают сообщения, которые должны обрабатываться ЭВМ. Эти сообщения посылаются большим количеством источников информации, так что общий поток таких сообщений можно считать пуассоновским. После предварительной обработки эти сообщения поступают для дальнейшей обработки на специализиро-

ванную ЭВМ, которая в каждый момент времени может обрабатывать только одно сообщение. Оставшиеся сообщения записываются на диск и образуют очередь. Кроме основной имеется резервная ЭВМ, которая вводится, в основном, для повышения надежности всей системы. На обрабатываемые сообщения накладывается требование обработать их к определенному сроку. Поэтому, когда сообщений в очереди оказывается слишком много, и время ожидания сообщения, находящегося в очереди первым, достигает некоторой пороговой величины, резервная ЭВМ подключается к их обработке. Эту реальную систему опишем следующей математической моделью УСМО с резервным прибором.

Рассмотрим однолинейную систему массового обслуживания с простейшим входящим потоком интенсивности λ , к которой может подключаться резервный прибор. Обслуживание предполагается экспоненциальным с интенсивностями μ_1 и μ_2 соответственно для основного и резервного приборов. Если заявка, находящаяся в некоторый произвольный момент времени t первой в очереди, поступила в систему в момент времени t_0 , то величину $s(t)=t-t_0$ будем называть текущим временем ожидания заявки, находящейся первой в очереди. Дисциплина обслуживания резервным прибором следующая: как только s – текущее время ожидания заявки, находящейся в очереди первой, достигает величины s_0 ($s_0=\text{const}>0$), подключается резервный при-