

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ
II Всероссийской молодежной
научной конференции
«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ,
ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 16–17 мая 2014 г.

*Под общей редакцией
кандидата технических наук И.С. Шмырина*

Томск
Издательский Дом Томского государственного университета
2014

зуюем оценку схожести двух видеокладров через анализ рангового распределения мощностей кластеров [7].

Программная реализация

Программная реализация данного проекта разбивается на следующие этапы:

1. Захват текущего кадра из видео потока
2. Сегментация кадра
3. Проведение оценки рангового распределения мощностей кластеров.
4. Сравнение полученной оценки со следующим видеокладром.

На первом этапе захват кадра из потока видео и представление в виде растрового рисунка происходит с помощью класса `RenderTargetBitmap`, который является частью пространства имен `Windows.Media.Imaging` и его метода `Render`. Для получения кадра из видео, без вывода кадра на экран в качестве изображения, необходимо выделить отдельный поток для функции, который будет реализовывать захват кадра.

Заключение

1. Исследованы методы поиска ключевых кадров.
2. Программно реализован простой метод поиска ключевых кадров на основе оценки средней яркости кадров.
3. Программно реализован метод k -средних для сегментации видеокладров.
4. Проводятся вычислительные эксперименты для выявления узких мест.

Работа выполняется в рамках государственного задания «Наука».

ЛИТЕРАТУРА

1. Гориченко А. В., Дёмин А. Ю. Обработка видеoinформации средствами Visual Studio // Технологии Microsoft в теории и практике программирования : сборник трудов X Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 19–20 марта 2013 г. – Томск : Изд-во ТПУ, 2013. – С. 237–239.
2. Лапин Е. В. Подготовка и запись DVD всех типов. Краткое руководство. – М. : Вильямс, 2006. – С. 320.
3. D. Feng, W. Siu, H. Zhang Multimedia information retrieval and management: Technological Fundamentals and Applications. Springer, 2003. pp. 44.
4. Байгарова Н. С., Бухитаб Ю. А., Евтеева Н. Н. Современная технология содержательного поиска в электронных коллекциях изображений. URL: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2001/part4/BBE> (дата обращения: 16.04.2014).
5. Демин А. Ю. Основы компьютерной графики : учеб. пособие – Томск : Изд-во Томского политехнического университета, 2011. – 191 с.
6. Дж. Ту, Р. Гонсалес Принципы распознавания образов. М. : Мир, 1978. С. 109–112.
7. Кудрин Б. И. Математика ценозов: видовое, ранговидовое, ранговое по параметру гиперболические H -распределения и законы Лотки, Ципфа, Парето, Мандельброта // Техногенная самоорганизация. Вып. 25 : Ценологические исследования. М. : Центр системных исследований, 2004. 248 с. URL: <http://mens.by/style/shine/470-mustache-sideburns> (дата обращения: 16.04.2014).

АЛГОРИТМ ОЦЕНКИ ДВИЖЕНИЯ, АДАПТИРОВАННЫЙ ДЛЯ ОБРАБОТКИ ЭКРАННОГО ВИДЕО

Д. В. Дружинин

Томский государственный университет
E-mail: dendru@rambler.ru

Введение

Камерное видео – это видео, снятое видеокамерой. Экранное видео – это видео происходящего на экране пользователя. Экранное видео формируется путём периодического снятия скриншотов экрана.

Часто сжатие и запись на жёсткий диск этого типа видеоданных требуется производить в режиме реального времени. Так как экранное видео – это видео высокого разре-

шения, требуется применять эффективные алгоритмы сжатия этого типа видеоданных. Поэтому актуальной является задача оптимизации по времени алгоритмов сжатия экранного видео. Использование алгоритма оценки движения позволяет существенно увеличить коэффициент сжатия, хотя и требует значительных вычислительных ресурсов.

Поскольку экранное видео в значительной степени отличается от камерного видео (см. Раздел 1), то для использования алгоритма оценки движения для обработки экранного видео его требуется адаптировать для этих целей. Проблема оптимизации по времени алгоритма оценки движения для обработки экранного видео является малоизученной. Удалось найти только одну работу [1], где представлен алгоритм оценки движения, адаптированный для обработки экранного видео. Но этот алгоритм выполняется слишком медленно (см. Раздел 3), что не позволяет использовать его для сжатия экранного видео в режиме реального времени.

В этой работе представлена модификация алгоритма оценки движения, рассмотренного в [1], позволяющая значительно повысить скорость кодирования видеоданных. Ускорение процесса кодирования достигается за счёт выявления не всех типов движений, а только некоторых из них.

1. Сравнение экранного и камерного видео

Экранное видео в значительной степени отличается от видео, снятого видеокамерой. Камерное видео запечатлевает объекты реального мира с определённой точки обзора при определённом освещении. Для этого типа видеоданных характерны плавные (непрерывно-тоновые) цветовые переходы между соседними пикселями.

Это свойство камерного видео используется во многих алгоритмах сжатия графических данных (например, JPEG [2, с. 182]). Экранное видео запечатлевает искусственным образом созданные объекты, для которых характерны резкие (дискретно-тоновые) цветовые переходы между соседними пикселями.

Разумно использовать информацию об особенностях экранного видео, чтобы адаптировать алгоритм оценки движения для более эффективного выявления объектов, перемещённых на некоторое расстояние относительно предыдущего кадра.

В [1] представлен алгоритм оценки движения, адаптированный для обработки экранного видео. Рассмотрим более подробно отличия экранного видео от камерного видео, которые наиболее ярко проявляются при использовании алгоритма оценки движения, а также особенности алгоритма [1], использующие эти свойства.

1. Для экранного видео характерны низкая степень совпадения для большинства векторов движения и полное (или почти полное) совпадение для единственного вектора в то время как для камерного видео часто существуют несколько векторов движения, которые обеспечивают высокую степень сходства соответствующих блоков по заданному критерию. Поэтому в [1] используется расстояние Хэмминга для измерения степени сходства блоков, то есть рассчитывается количество пар соответствующих пикселей одного цвета.

2. В экранном видео объекты могут перемещаться на значительное расстояние за временной отрезок, разделяющий два соседних кадра, в то время как в видео, снятом видеокамерой, обычно происходит плавное движение объектов от кадра к кадру. Поэтому в [1] используется более обширная область поиска (вплоть до поиска по всему кадру), чем при сжатии камерного видео.

3. Градиентный метод поиска оптимального вектора движения активно применяется при сжатии камерного видео [3–5]. Но в случае экранного видео в силу дискретно-тоновой природы таких видеоданных выбор из нескольких ближайших векторов движения в приоритетном порядке в качестве текущего вектора v_1 , который обеспечивает минимальное расхождение (если оно не равно 0), зачастую неэффективен. Это обусловлено тем обстоятельством, что вероятность того, что вектор движения, обеспечивающий наилучшее соответствие блоков, находится в непосредственной близости от

вектора v_1 не выше, чем при проверке любого другого вектора движения. Поэтому в [1] не используется градиентный метод.

4. В экранном видео обычно значительная часть соответствующих пикселей соседних кадров совпадают. Поэтому в [1] предлагается сначала провести попиксельное сравнение текущего и предыдущего кадров на равенство. Затем оценка движения проводится только для блоков текущего кадра, изменившихся относительно предыдущего кадра.

5. В экранном видео велика вероятность соседства нескольких блоков, имеющих один и тот же вектор движения (например, при движении окна), в отличие от камерного видео, где одинаковый вектор движения для соседних блоков встречается несколько реже за счёт изменения формы объекта и угла обзора при движении этого объекта.

2. Описание алгоритма оценки движения, адаптированного для экранного видео

Выявление всех движений в экранном видео оказалось слишком трудоёмкой задачей для выполнения в режиме реального времени (см. Раздел 3). В такой ситуации логичной выглядит идея провести некоторую классификацию типов движений в экранном видео и разработать алгоритмы, выявляющие некоторые из этих типов движений, и работающие при этом значительно быстрее алгоритма, выявляющего все типы движений.

Можно выделить 3 основных вида движений в экранном видео:

1. Движения по вертикали и горизонтали (потенциально на большие расстояния). Это движения, осуществляемые вследствие вертикального и горизонтального скроллинга, нажатия пользователем на клавиши вниз, вверх, вправо, влево, Pg Up, Pg Dn и прочие.

2. Движения в произвольном направлении на небольшие расстояния. Это движения, осуществляемые вследствие, например, достаточно плавного перетаскивания пользователем окна.

3. Прочие движения.

Как правило, подавляющее большинство движений различных объектов на экране, возникающих в ходе работы пользователя относятся к (1) либо (2) категории. Используя приведённую классификацию движений в экранном видео, удалось разработать следующую схему кодирования на основе алгоритма оценки движения, описанного в [1].

Последовательно выполняются две модификации алгоритма оценки движения:

1. Алгоритм, осуществляющий поиск блока, соответствующего текущему блоку, только по вертикали и по горизонтали. То есть этот алгоритм не рассматривает векторы движения с одновременно ненулевыми сдвигами по оси x и y .

2. Алгоритм, осуществляющий поиск блока, соответствующего текущему блоку, во всех направлениях, но только в ближайшей окрестности текущего блока.

Таким образом, выполнение алгоритма, осуществляющего полный перебор возможных векторов движения (предложенный в [1]), заменяется на выполнение двух менее трудоёмких алгоритмов.

Для ускорения работы алгоритма оценки движения при обработке экранного видео предлагается использовать технику предварительного сравнения диагональных элементов. Только в случае, когда все диагональные элементы попарно равны между собой, происходит сравнение всех элементов этих блоков. Такая техника приводит к ускорению выполнения алгоритма оценки движения, так как она одновременно учитывает и горизонтальную и вертикальную корреляцию пикселей движущегося объекта.

Вначале производится вычисление минимального прямоугольника, охватывающего все изменившиеся относительно предыдущего кадра области текущего кадра. Именно в пределах этого прямоугольника в дальнейшем будет осуществляться поиск соответствия для текущего блока.

3. Результаты тестирования

При тестировании был выбран размер блока $16*16$ пикселей. Максимальные отклонения по оси x и y для алгоритмов-участников тестирования (5) и (6) рассчитывались по

формуле $n \cdot blockSize$, где $blockSize$ – это размер блока. При этом было выбрано значение n , равное 3. При тестировании каждый кадр имел разрешение 1024×768 и глубину цвета в 32 бита. Тестирование проводилось на платформе со следующими характеристиками: процессор Intel Core 2 Duo E6750 2,66 ГГц, оперативная память DDR2 2Гб, операционная система Windows XP.

Замечание 1. Замеряемый параметр “Количество блоков” означает количество блоков, для которых было найдено соответствие данной реализацией. При этом суммарное количество блоков равно $1024 \times 768 / (16 \times 16) = 3072$. При этом надо учитывать, что часть блоков не изменились по сравнению с предыдущим кадром.

Замечание 2. запись $3 \rightarrow 5$ означает, что текущий кадр сначала обрабатывается реализацией (3) алгоритма оценки движения, а затем реализацией (5).

Рассмотрим результаты тестирования при двух типах движений.

Т а б л и ц а 1

Движения первого типа (для возникновения движения использовался скроллинг)

Алгоритм / Параметр	Время выполнения, мс	Количество блоков, шт
1. Алгоритм оценки движения, представленный в [1]	10 049	1 436
2. Алгоритм оценки движения, представленный в [1], с добавленной начальной проверкой диагональных элементов	4 012	1 436
3. Модификация (1) алгоритма оценки движения, представленная в этой работе (поиск блока, соответствующего текущему блоку, осуществляется только по вертикали и по горизонтали)	24	1 436
4. Модификация (1) алгоритма оценки движения, представленная в этой работе, без начальной проверки диагональных элементов	65	1 436
5. Модификация (2) алгоритма оценки движения, представленная в этой работе (поиск блока, соответствующего текущему блоку, осуществляется во всех направлениях, но только в ближайшей окрестности текущего блока)	510	697
6. Модификация (2) алгоритма оценки движения, представленная в этой работе, без начальной проверки диагональных элементов	1 094	697
$3 \rightarrow 5$	$24 \rightarrow 10$	$1436 \rightarrow 0$
$5 \rightarrow 3$	$510 \rightarrow 15$	$396 \rightarrow 830$

Т а б л и ц а 2

Движения второго типа (для возникновения движения использовалось плавное перетаскивание окна)

Алгоритм / Параметр	Время выполнения, мс	Количество блоков, шт
1. Алгоритм оценки движения, представленный в [1]	3561	1384
2. Алгоритм оценки движения, представленный в [1] с добавленной начальной проверкой диагональных элементов	2422	1384
3. Модификация (1) алгоритма оценки движения, представленная в этой работе (поиск блока, соответствующего текущему блоку, осуществляется только по вертикали и по горизонтали)	63	450
4. Модификация (1) алгоритма оценки движения, представленная в этой работе, без начальной проверки диагональных элементов	140	450
5. Модификация (2) алгоритма оценки движения, представленная в этой работе (поиск блока, соответствующего текущему блоку, осуществляется во всех направлениях, но только в ближайшей окрестности текущего блока)	47	1384
6. Модификация (2) алгоритма оценки движения, представленная в этой работе, без начальной проверки диагональных элементов	62	1384
$3 \rightarrow 5$	$62 \rightarrow 47$	$463 \rightarrow 962$
$5 \rightarrow 3$	$47 \rightarrow 16$	$1379 \rightarrow 38$

Быстрее всего выполняется реализация (3), что позволяет использовать её при сжатии экранного видео в режиме реального времени. Но реализация (3) не в состоянии

выявить движения второго типа, поэтому имеет смысл рассмотреть комбинации реализаций (3) и (5). Наиболее стабильное время выполнения демонстрирует последовательность (3) → (5). Время работы этой последовательности не превышает 109 мс. При этом последовательно выполняемые реализации (3) и (5) находят соответствие для практически такого же количества блоков, что и реализация (1).

Заключение

Проведение классификации движений в экранном видео сделало возможным поиск заданных типов движений. Затем был разработан алгоритм оценки движения, использующий информацию о том, что одни типы движений встречаются в экранном видео чаще, чем другие. При этом просматриваются лишь области, соответствующие типам движений, которые встречаются чаще всего. Такое сужение области поиска позволяет, как показали результаты тестирования, значительно ускорить выполнение оценки движения по сравнению с алгоритмом, выявляющим все типы движений, что в свою очередь позволит в перспективе применить представленный алгоритм для сжатия экранного видео в режиме реального времени.

Предложенная схема сжатия позволяет ускорить выполнение оценки движения десятикратно при незначительных потерях в количестве распознанных движений, так как большая часть движений в экранном видео относятся именно к (1) или (2) типам. Поэтому такой алгоритм оценки движения может быть использован на практике.

ЛИТЕРАТУРА

1. *Motion estimation/compensation for screen capture video* / FreePatentsOnline.com. URL: <http://www.freepatentsonline.com/7224731.html>
2. *Сэлмон Д.* Сжатие данных, изображений и звука / перевод с англ. В. В. Чепьжова. – М.: Техносфера, 2006. – 365 с. – (Мир программирования).
3. *Timoner J. S., Freeman D. M.* Multi-Image Gradient-Based Algorithms for Motion Estimation. URL: http://people.csail.mit.edu/samson/papers/Timoner_Freeman_MultiImageGradient.pdf
4. *Keller Y., Averbuch A.* Fast motion estimation using bidirectional gradient methods. URL: http://www.eng.biu.ac.il/~kellery1/publications/pdf/optical_flow_jeec_final.pdf
5. *Movement estimation system for video signals using a recursive gradient method* / FreePatentsOnline.com. URL: <http://www.freepatentsonline.com/4695882.html>

ОБНАРУЖЕНИЕ ЛОЖНЫХ ПУТЕЙ В СХЕМАХ С ПАМЯТЬЮ

Е. П. Ирдынеева

Томский Государственный Университет
E-mail: Kateird@mail.ru

Введение

В последнее время отчетливо просматривается тенденция к значительному ужесточению требований к надежности различного рода дискретных устройств (ДУ). Это объясняется важностью и ответственностью функций, возлагаемых на них в различных областях, включая атомную энергетику, химическое производство, авиацию и т.п. Одним из способов обеспечения высокой надежности является применение, как в процессе производства, так и на этапе эксплуатации, средств и методов технической диагностики, позволяющих осуществлять контроль, локализацию неисправностей и восстановление аппаратуры.

В связи с ростом уровня интеграции схем, работающих на высоких частотах и при низких напряжениях питания, возникает необходимость в тестировании неисправностей задержек путей. Различают робастные и не робастные неисправности задержек путей. Противоположным перепадам значений сигналов на рассматриваемом пути в общем случае сопоставляются различные задержки. Каждая из них обнаруживается специальной парой тестовых наборов. Если для каждого из перепадов пути отсутствует