

ОПТИМИЗАЦИЯ ПРОЦЕССА КОМПРЕССИИ ВИДЕОДАНЫХ МЕТОДОМ КРОСС-КАДРОВОЙ ИНТЕРПОЛЯЦИИ

В.А. Главнов, А.В. Крапивенко

В статье рассмотрены два подхода к ускорению процесса компрессии видеоданных методом кросс-кадровой интерполяции. Даны определения основных терминов используемых в рамках изложения процесса компрессии и его оптимизации. Кратко рассмотрены классические методы поиска векторов смещения блоков изображения. Проведена оценка алгоритмической сложности процесса компрессии классического и с применением подходов к ускорению.

Введение

В связи с интенсивным использованием аудио- и видеоданных в современных компьютерных системах и сетях закономерно возникают проблемы эффективного представления таких данных и быстрого доступа к ним. Для разрешения этих проблем создан ряд методов компрессии с некоторой потерей качества оригинала, допустимой в связи с особенностями работы органов чувств и субъективностью восприятия человека. Это большая группа JPEG и MPEG алгоритмов, методы фрактальной компрессии, волнового (wavelet) сжатия.

Наиболее распространенными методами сжатия видеоданных являются алгоритмы M-JPEG, MPEG 2 и MPEG 4. Основой алгоритмов является разбиение каждого кадра на квадратные области одинакового размера с целью применения дискретного косинус-преобразования и метода компенсации движения [1]. Алгоритмы обеспечивают высокую степень компрессии видеоданных и приемлемое для просмотра художественных и научно-популярных фильмов качество. Однако многие часто встречающиеся сцены, особенно насыщенные мелкими деталями и плавными цветовыми переходами, являются сложными для алгоритмов. В таких случаях приходится жертвовать либо степенью компрессии, либо качеством изображения, что нередко приводит к ухудшению восприятия декомпрессированного видео.

Кроме того, использование дискретного разложения вышеупомянутых квадратных областей изображения в гармонические ряды и квантование коэффициентов приводит к потере мелких деталей изображения, к “смазыванию” граней и появлению эффекта Гиббса - блочных артефактов правильной квадратной формы [2].

Перспективное развитие получили методы компрессии, основанные на т.н. *вейвлет* (wavelet) разложениях. Это принципиально отличный от алгоритмов JPEG/MPEG подход к сжатию видеoinформации, основанный на разложении сигнала по частотным диапазонам, дальнейшем огрублении (квантовании, а иногда и просто замены нулями интенсивности) малозначительных составляющих и рациональном представлении полученных данных [2]. Методы демонстрируют на практике сравнимую с MPEG степень компрессии, свободны от большинства недостатков последнего, но обладают своими. Одним из таких недостатков является квантование четкости представления мелких деталей, что можно показать на примере лестничного дефекта. Перила лестницы, уходящей далеко от места съемки, вдалеке будут сливаться, но в некотором месте произойдет резкий переход на их отчетливую видимость. Такие дефекты приводят к потере чувства пространства в двумерном изображении.

Таким образом, возникает проблема создания способа компрессии, отличного от описанных алгоритмов, позволяющего получить оптимальное качество с наименьшими потерями в степени компрессии. Одним из таких способов является метод кросс-кадровой интерполяции, который и является основным предметом рассмотрения.

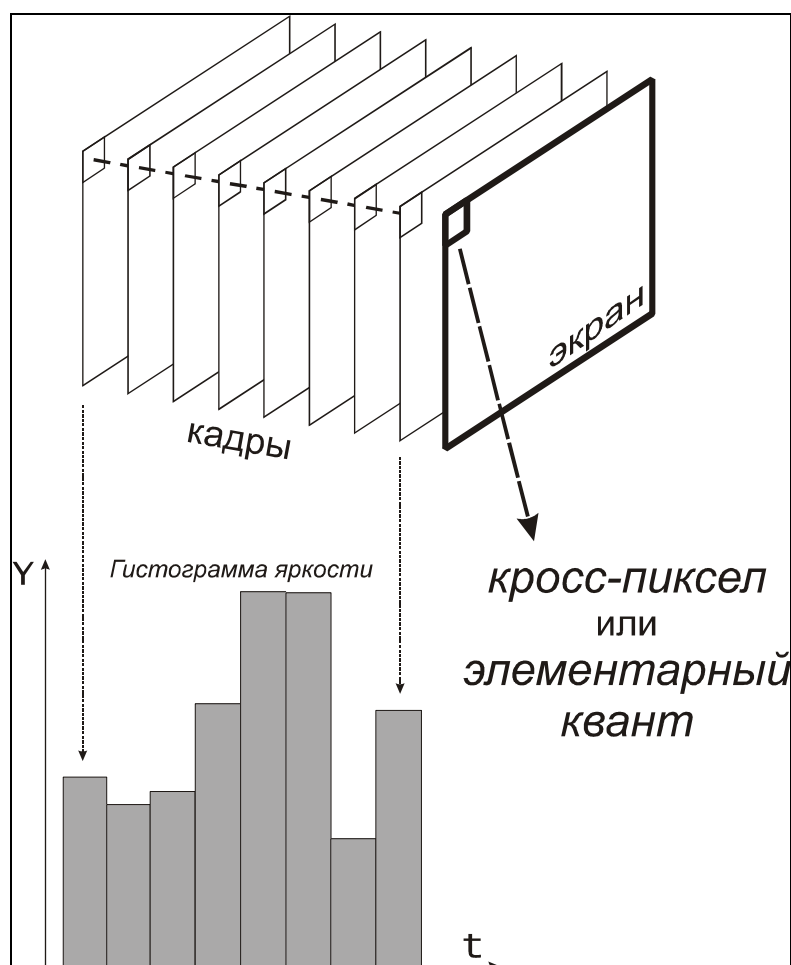
Большинство современных методов направлены на поиск закономерностей в пределах одного кадра. Одной из отличительных особенностей метода кросс-кадровой интерполяции является отведение приоритетного внимания поиску временных закономерностей поведения каждого пиксела [2]. Метод, обладая несколько меньшей степенью компрессии видео по сравнению с MPEG, свободен от большинства его недостатков. Одним из преимуществ метода является возможность упорядочивания данных по важности с точки зрения восприятия человеческим глазом и возможности передачи лишь самой важной части этих данных. При снижении пропускной способности канала связи динамичность смены кадров на экране не меняется, а ухудшается только качество изображения. Это свойство метода допускает его применение в области вещания по каналам с сильно меняющейся пропускной способностью, например, при Internet вещании.

Одним из существенных недостатков метода кросс-кадровой интерполяции является его высокая асимметричность (соотношение времени, потраченного на компрессию/декомпрессию), процесс компрессии видеоданных ресурсоемок – предъявляются высокие требования к производительности процессора и объему оперативной памяти. Материал данной статьи освещает поиск путей устранения этих недостатков.

Определения и терминология

Кросс-пикселом или элементарным квантом видеопоследовательности называется источник элементарного временного MultiMedia сигнала.

Иными словами, квантом является область экрана, соответствующая по размерам одному пикселу воспроизводимой видеопоследовательности, как проиллюстрировано на рисунке ниже [3].



Траекториями будем называть функции (гистограммы) яркости Y или функции цветоразностных составляющих квантов, заданные на некотором отрезке времени.

Под расстоянием между двумя траекториями с дельтой δ будем понимать количество точек (кадров), на которых модуль разности значений функций больше δ :

$$d(f_i, f_j)_{[t, t+\Delta t]} = \text{count}\{t : |f_i(t) - f_j(t)| > \delta, t \in [t, t + \Delta t]\} \quad (1)$$

Классом траекторий одинакового поведения с дельтой δ , радиусом r и опорной траекторией f_{on} на сегменте $[t, t+\Delta t]$ будем называть множество траекторий f_i для которых выполняется:

$$d_\delta(f_{on}, f_i)_{[t, t+\Delta t]} \leq r, \quad (2)$$

где f_{on} - опорная траектория класса.

Замечание: опорная траектория класса может быть произвольной функцией, не обязательно из множества траекторий.

Оптимальным коэффициентом подобия (или просто коэффициентом подобия) траектории f_i относительно траектории f_j на сегменте $[t, t+\Delta t]$ будем называть такое число k :

$$k = \arg \min_k (d_\delta(f_j, kf_i)_{[t, t+\Delta t]}) \quad (3)$$

Замечание: чисел k , удовлетворяющих условию (3), может быть больше одного, множество таких чисел k будем называть множеством оптимальных коэффициентов подобия (или просто множеством коэффициентов подобия) траектории f_i относительно траектории f_j .

Классом траекторий подобного поведения с дельтой δ , радиусом r и опорной траекторией f_{on} на сегменте $[t, t+\Delta t]$ будем называть множество траекторий f_i для которых выполняется:

$$d_{\delta}(f_{on}, k_i f_i)_{[t, t+\Delta t]} \leq r, \quad (4)$$

где f_{on} - опорная траектория класса, k_i - оптимальные коэффициенты подобия траекторий f_i относительно траектории f_{on} на сегменте $[t, t+\Delta t]$.

Замечание: в дальнейшем, говоря о коэффициентах подобия, классах траекторий одинакового и подобного поведения, будем всегда подразумевать, что они рассматриваются на некотором сегменте (фрагменте видеопоследовательности), даже если это явно не указано.

Схема метода кросс-кадровой интерполяции

Рассмотрим видеоряд в дискретном пространстве яркостей Y (уровней серого), $Y=0\div 255$. Без ограничения общности метод можно распространить на полноцветные видеоряды.

Основными этапами кросс-кадровой интерполяции являются [2]:

- 1) Исходный видеоряд по ключевым кадрам разбивается на временные сегменты (смысловые сцены);
- 2) Для каждого кванта строится функция яркости Y от времени (траектория от номера кадра) в пределах рассматриваемого сегмента;
- 3) Все траектории специальным образом группируются в классы траекторий одинакового поведения сегмента так, что в пределах одного класса кванты ведут себя примерно одинаково;
- 4) Кванты, не отнесенные ни к одному из классов подобия, трактуются как одиночный незначительный шум, и заменяются средневзвешенными значениями соседей по кадру;
- 5) В каждом классе ищется опорная (базисная) траектория, имеющая минимальное расстояние с остальными. Определяются места и величины отклонений (т.н. *выбросов*) остальных траекторий от опорной;
- 6) Формируется так называемый прообраз сегмента – некая весьма длинная ломаная, поставленная в однозначное соответствие опорным траекториям всех классов траекторий подобного поведения;
- 7) Выбросы между опорной траекторией и остальными в местах единичных несоответствий подвергаются дискретному косинусоидальному преобразованию (ДКП), коэффициенты ДКП квантуются;
- 8) В компрессированный файл посегментно заносится принадлежность траекторий квантов к классам траекторий подобного поведения, коэффициенты разложения прообраза сегмента,

коэффициенты подобия траекторий, места и величины отклонений траекторий класса от опорных траекторий;

9) Для устранения избыточности полученный файл подвергается сжатию без потерь информации.

Процесс вычисления оптимального коэффициента подобия

Рассмотрим процесс вычисления коэффициента подобия траекторий при наличии произвольного числа выбросов (местах отличия поведения одних траекторий пикселей от других), необходимый для выполнения шага 5 метода кросс-кадровой интерполяции. Алгоритм вычисления следующий:

1. Вычисление наименьшего и наибольшего допустимых значений коэффициента подобия для траекторий с учетом ограничения траекторий δ -трубкой для каждого сечения по времени (кадра) в сегменте.
2. Сортировка массивов нижних и верхних границ коэффициента подобия.
3. Инициализация нулем значений текущего и оптимального показателей качества. Установка указателей текущей позиции массива нижних границ и массива верхних границ на первые элементы массивов.
4. Сравнение текущих значений в массивах нижних и верхних границ коэффициента. Если текущее значение массива верхних границ больше или равно текущему значению массива нижних границ, то увеличиваем значение текущего показателя качества и передвигаем текущую позицию массива нижних границ на единицу вперед. В противном случае уменьшаем значение текущего показателя качества и передвигаем текущую позицию массива верхних границ на единицу вперед.
5. Сравнение значения текущего показателя качества и оптимального качества. Если текущий показатель качества окажется лучше, чем оптимальный, то присваиваем оптимальному показателю качества значение текущего, а предыдущее значение в массиве нижних границ и текущее значение в массиве верхних границ запоминаем как границы отрезка пересечения наибольшего числа отрезков.
6. Проверка: если указатель текущей позиции в массиве нижних границ указывает на позицию за пределами массива, то поиск завершен. В противном случае совершается переход к четвертому шагу и поиск продолжается.

В этом алгоритме первый шаг направлен на получение множества отрезков значений коэффициента подобия, а остальные – на нахождение пересечения наибольшего числа таких отрезков. При этом полученное пересечение – это множество коэффициентов подобия траекторий,

удовлетворяющих формуле (3), а число отрезков, которые не содержат это пересечение – это ни что иное, как число выбросов второй траектории за δ -трубку первой, то есть расстояние между траекториями с дельтой δ (это число – дополнение вычисленного значения оптимального показателя качества до полного числа сечений в сегменте).

Подсчитаем сложность выполнения формирования первого класса траекторий подобного поведения классическим методом (метод перебора «каждый с каждым») для сегмента с высотой M пикселей, шириной N пикселей и временной продолжительностью T кадров. Для поиска наилучшей траектории понадобится $(M*N)^2$ операций поиска коэффициентов. При этом сложность операции поиска коэффициентов, в случае применения быстрой сортировки Хоара, в среднем составляет:

- $4*T$ операций сравнения и перемещения, $4*T$ операций сложения и деления для вычисления верхних и нижних границ коэффициентов;
- $2*T*\log_2(T)$ операций сравнения и $T*\log_2(T)$ операций перестановок элементов для сортировки двух массивов;
- не менее T операций сравнения для поиска интервала и количества выбросов.

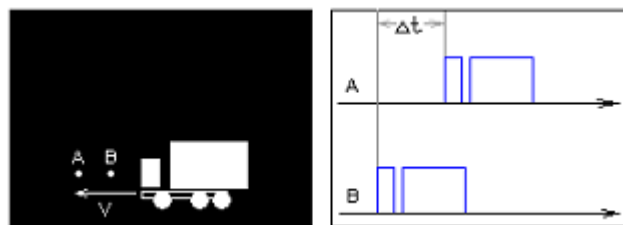
Итого получается сложность: $(M*N)^2*(12*T+4*T*\log_2(T)+T)$ машинных операций (операция перестановки элементов выполняется двумя машинными операциями).

Таким образом, если нужно сформировать первый класс траекторий подобного поведения классическим методом для сегмента размером 512x384x30 (одна секунда фильма в формате MPEG4), нам понадобится порядка 75 секунд времени при условии реализации метода на языке ассемблера и выполнения процессором с частотой 1ГГц. Если при этом учесть то, что большинство современных процессоров работают с кэш-памятью на половинной частоте, и для реализации алгоритмов помимо команд сравнения, переноса и деления выполняется примерно столько же команд проверки условий и переходов, то оценка скорости в 75 секунд компрессии на секунду исходного видео вырастает в 4-8 раз. В связи с этим возникают потребности в более быстрой сборке первых классов траекторий подобного поведения.

Рассмотрим два основных подхода к быстрой сборке траекторий в классы подобного поведения.

Метод блочного формирования классов

Метод основан на взаимосвязи поведения траекторий и перемещения блоков изображения во времени. Если рассмотреть равномерно движущийся белый автомобиль на черном фоне, то не трудно заметить, что траектории функций яркости соседних квантов весьма схожи, и смещены относительно друг друга на некоторый промежуток времени:



Этот промежуток времени обратно пропорционален скорости движения объекта. Таким образом, если какой-либо блок изображения получается из предыдущего кадра путем смещения изображения на постоянный вектор на протяжении всего временного сегмента, то можно предположить, что траектории блока имеют схожее поведение с траекториями блока смещенного на тот самый постоянный вектор смещения. Таких векторов смещения для блока может не существовать, а может быть несколько.

Существует особое значение постоянного вектора смещения - это значение – $(0,0)$. Особенность блоков изображения, получаемых из предыдущих кадров путем смещения на вектор $(0,0)$ заключается в том, что вспышки яркости по всей площади блока, приводящие только к пропорциональному изменению яркостей пикселей данного блока, не влияют на подобие поведений траекторий внутри данного блока. Таким образом, если какой-либо блок изображения получается из предыдущего кадра путем смещения на вектор $(0,0)$ и масштабирования значений яркостей пикселей в блоке, то можно сделать заключение о том, что траектории данного блока имеют подобное поведение.

Преимуществом данного метода является то, что он легко обобщается на случай наличия незначительного числа выбросов на траекториях.

Преобладающим вектором смещения будем называть заранее выбранный по каким-либо причинам вектор смещения. Например, преобладающим вектором смещения мы можем выбрать тот вектор смещения, который чаще всего применим для получения блока изображения путем смещения на него изображения предыдущего кадра на протяжении всего сегмента.

В местах выбросов среди возможных векторов смещения не будет преобладающего вектора смещения.

Для поиска возможных векторов смещения можно воспользоваться различными хорошо себя зарекомендовавшими алгоритмами. Вот два из них:

- метод корреляции фаз;
- метод прямого сравнения блоков изображения.

Метод корреляции фаз

Метод основан на теореме о смещении преобразования Фурье [4]. Если одна функция получается из другой путем смещения начала отсчета на вектор (x_0, y_0) : $f_1(x, y) = f_2(x + x_0, y + y_0)$, то по теореме о смещении изображения функций связаны формулой:

$$\begin{aligned} F_1(\xi, \eta) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(x, y) e^{-i(x\xi + y\eta)} dx dy = \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_2(x + x_0, y + y_0) e^{-i((x+x_0-x_0)\xi + (y+y_0-y_0)\eta)} d(x + x_0) d(y + y_0) = \\ &= e^{i(x_0\xi + y_0\eta)} F_2(\xi, \eta) \end{aligned}$$

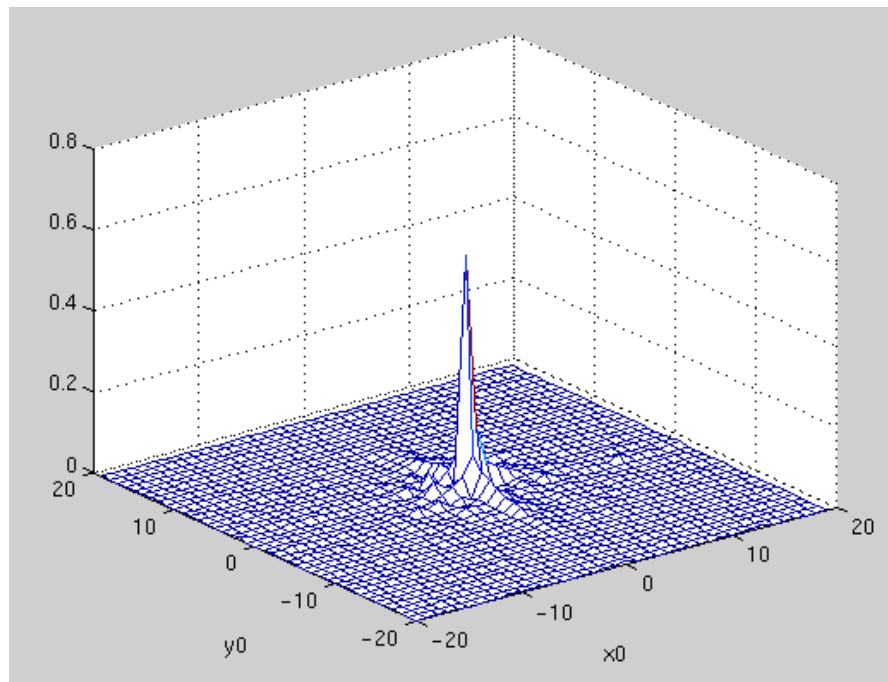
Теперь домножим левую и правую части на $F_2^*(\xi, \eta)$:

$$\begin{aligned} F_1(\xi, \eta) * F_2^*(\xi, \eta) &= e^{i(x_0\xi + y_0\eta)} F_2(\xi, \eta) * F_2^*(\xi, \eta) = e^{i(x_0\xi + y_0\eta)} \|F_2(\xi, \eta)\|^2 = \\ &= e^{i(x_0\xi + y_0\eta)} \|F_2(\xi, \eta) * e^{-i(x_0\xi + y_0\eta)} F_1(\xi, \eta)\| = e^{i(x_0\xi + y_0\eta)} \|F_2(\xi, \eta) * F_1(\xi, \eta)\| * \|e^{-i(x_0\xi + y_0\eta)}\| = \\ &= e^{i(x_0\xi + y_0\eta)} \|F_2(\xi, \eta) * F_1(\xi, \eta)\| \end{aligned}$$

Отсюда:

$$\frac{F_1(\xi, \eta) * F_2^*(\xi, \eta)}{\|F_2(\xi, \eta) * F_1(\xi, \eta)\|} = e^{i(x_0\xi + y_0\eta)}$$

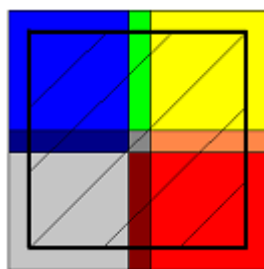
Левая часть - это не что иное, как нормализованная совместная спектральная плотность двух сигналов: функции f_1 и функции f_2 . Теперь нас будет интересовать обратное преобразование Фурье от левой части. Оно представляет собой совместную корреляционную функцию f_1 и f_2 и имеет максимум в точке (x_0, y_0) :



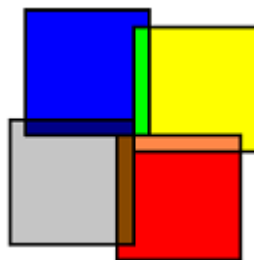
Теперь, если вместо f_1 подставить функцию яркости пикселей в блоке некоторого кадра, вместо f_2 подставить функцию яркости пикселей в таком же по размерам блоке предыдущего кадра, то (x_0, y_0) будет вектором смещения первого блока относительно второго.

Достоинства метода: метод алгоритмически прост, применяя быстрое преобразование Фурье для вычисления изображений функций и корреляционной функции можно добиться высокой производительности, метод позволяет эффективно разбивать видеопоследовательности на сегменты, применяя метод совместно с методом иерархического сопряжения блоков можно быстро находить направление и скорость перемещения больших объектов.

Вместе с тем, метод имеет массу недостатков. Самый большой из них – это то, что при вычислении корреляционной функции для двух блоков изображения, математическая природа метода периодически дополняет функции яркости квантов. В связи с этим метод применим только для нахождения небольших смещений больших участков изображения друг относительно друга. Для нахождения же больших смещений маленьких участков изображения необходимо получить покрытие области поиска вектора фрагментами размером с искомую область с достаточным коэффициентом перекрытия. Например, для поиска смещения участка изображения квадратной формы 8×8 пикселей в области $[-6, 6] \times [-6, 6]$ пикселей покрытием области поиска векторов с наименьшим перекрытием покрытие будет состоять из четырех квадратов 8×8 :



Возможно, покрытие вида:



может показаться более эффективным, но это не так, оно не обеспечивает разрешение неопределенностей в знаках координат на границах блоков, покрывающих область поиска векторов смещений.

В предыдущем примере приведено покрытие с минимальным перекрытием, реально же приходится пользоваться более мощными перекрытиями из-за существенных искажений корреляционной функции вблизи границ блоков, покрывающих область поиска векторов

смещений. Как следствие этого, требуются большие объемы памяти и большие вычислительные мощности для использования метода корреляции фаз.

Метод прямого сравнения блоков изображения

Метод заключается в вычислении норм разности между заданным блоком и его смещением в другом кадре на каждый вектор из области поиска векторов смещения.

К достоинствам метода относятся: его несомненная простота, более высокая производительность при поиске больших смещений маленьких областей или равномерно движущихся больших областей изображения, меньшие требования к памяти. В случае больших областей изображения метод можно ускорить путем применения свойства полуаддитивности нормы: проводить вычисление нормы разности не сразу для целых блоков, а только для некоторого подмножества точек блока. В большинстве случаев такой прием приводит к существенному ускорению поиска по причине малости множества возможных векторов смещения по сравнению с множеством поиска.

Недостатками метода являются: низкая эффективность при поиске малых смещений больших областей изображения, сложность реализации и обобщения метода на случай поиска не эквивалентных друг другу блоков, а подобных по значениям яркости пикселей блока. В этом случае приходится пользоваться разностью, не являющейся нормой и не обладающей свойством полуаддитивности. Расстояние между траекториями здесь понимается не в смысле формулы (1), поэтому после сборки требуется дополнительная работа по фильтрации траекторий и добавлению к классу траекторий, не попавших в него при сборке.

Метод является вспомогательным для определения стержневых массивов траекторий и позволяет определить только точно попадающие в класс подобного поведения траектории. Остальные траектории необходимо относить к классам подобного поведения каким-либо другим методом.

Подсчитать алгоритмическую сложность метода весьма затруднительно, поскольку структура иерархии сопряженных блоков может весьма различаться для разных видеопоследовательностей. Подсчитаем сложность самого «плохого» случая, когда совершается разбиение кадра на блоки размером 8×8 каждый, и не будем рассматривать возможность сдвига траекторий относительно друг друга.

Сложность = [сложность вычисления коэффициента пропорциональности блоков соседних кадров] + [сложность объединения блоков] + [сложность отнесения оставшихся траекторий к сформированному классу] + [сложность фильтрации траекторий класса].

Дано $(M \cdot N / 64)$ блоков. Коэффициент пропорциональности для каждого блока – $8 \cdot 8 \cdot T$ – операций деления, $8 \cdot 8 \cdot T$ операций сравнения. Объединение ведется только для блоков, внутри

которых есть подобие траекторий сложностью не выше $(M*N)^2/4096$ операций поиска коэффициента подобия. Сложность отнесения оставшихся траекторий и фильтрации траекторий класса - $(M*N)$ операций поиска коэффициента подобия.

Итого алгоритмическая сложность метода:

$3*M*N + M*N*(1 + (M*N)/4096)*(12*T + 4*T*\log_2(T) + T)$, что значительно ниже, чем для классического метода.

Метод каскадной сборки классов траекторий подобного поведения

Метод основан на следующем подходе к визуальному восприятию разности поведений траекторий: чем ближе находятся друг к другу кванты – тем заметней разница поведений траекторий. Идея метода состоит в разбиении всей площади кадра на отдельные участки и сборки траекторий в классы подобия внутри этих участков. Далее, объединяя эти участки кадра в бóльшие участки, проводится сборка в классы подобного поведения опорных траекторий классов, полученных на предыдущих стадиях. Процесс продолжается далее, пока не будет проведена сборка по кадру в целом.

Метод допускает несколько модификаций по разбиению кадра на области и по параметрам классов, собираемых на каждом этапе.

Первый и наиболее простой подход к разбиению кадра на области заключается в априорном разбиении кадра на заранее известные области для каждого этапа сборки. Второй подход сложнее, и заключается в том, что разбиение кадра на каждом последующем этапе зависит от числа опорных траекторий, получившихся на предыдущем этапе сборки. Например, можно ограничить число траекторий или опорных траекторий в области для сборки.

В способах задания параметров собираемых классов так же можно выделить два подхода. Первый подход основан на принципе «собрать траектории, наверняка попадающие в класс». Например, для трехэтапной сборки в классы с дельтой 15 и радиусом 30% от длины сегмента, в качестве параметров для траекторий классов подобного поведения на каждом этапе можно выбрать дельту равную 5 и радиус класса 10% от длины сегмента. Таким образом, после сборки на третьем заключительном этапе мы получим классы с дельтой не более 15 и радиусом не более 30% от длины сегмента. Правда, после такой сборки может возникнуть явно избыточное число классов траекторий подобного поведения, что потребует объединения некоторых классов для снижения их избыточности.

Другой подход заключается в сборке по принципу: «не забыть ни одну траекторию из класса». Например, для предыдущего случая трехэтапной сборки в классы подобия с дельтой 15 и радиусом 30% от длины сегмента, параметры классов на каждом этапе будут такими же: дельта – 15, радиус – 30%. Таким образом, в результате сборки, получим классы с траекториями, которые

могли бы попасть в класс с требуемыми параметрами при однокаскадной сборке. В этом случае после сборки в классы траекторий подобного поведения следует сделать дополнительную проверку на принадлежность траекторий классу и, возможно, исключить некоторые траектории.

К достоинствам подхода следует отнести относительную простоту реализации, легкость обобщения на случай масштабирования траекторий в одном классе, возможность полной сборки траекторий в классы, незначительное понижение коэффициента компрессии по сравнению с однокаскадной сборкой по принципу наилучших опорных траекторий.

Из недостатков можно отметить необходимость разбиения видеопоследовательности на сегменты до сборки в классы подобного поведения.

Процесс подсчета алгоритмической сложности существенно зависит от выбранного подхода к разбиению. Если следовать правилу, что сложность сборки в классы подобия на каждом следующем этапе не должна превышать сложность сборки на предыдущем, то можем получить оценку сверху. Для первого этапа сборки при разбиении кадра на блоки 32x32 имеем сложность: $(32)^4$ операций поиска коэффициента подобия для каждого блока. Число этапов сборки при таком разбиении, дельте 10, радиусе 25% от продолжительности сегмента и кадре 512x384 редко превышает 4. Таким образом, итоговая сложность составит $M*N*(32)^2*(12*T+4*T*\log_2(T)+T)*4$ элементарных операций.

Заключение

Применение вышеописанных алгоритмов для сборки квантов в классы подобного поведения существенно ускоряет процесс компрессии видеоданных методом кросс-кадровой интерполяции и снижает требования к объему оперативной памяти компрессора. При этом практика показывает, что степень компрессии видеоданных падает незначительно – в пределах 5%. Если при использовании классического метода для компрессии одной секунды несжатого видео большого формата уходило до двух часов времени, то теперь это время сократилось до 3-15 минут в зависимости от характера компрессируемой видеопоследовательности. Внесенные улучшения в виде использования классов траекторий подобного поведения вместо классов траекторий одинакового поведения, как это было в классическом методе, делают возможным применение метода кросс-кадровой интерполяции для компрессии крупноформатных видеопоследовательностей, что существенно расширяет его функциональные возможности.

Список литературы

1. Graham R. Martin. Block-matching motion compensation. - <http://www.dcs.warwick.ac.uk/mcg/bmmc/> (10.10.2001)
2. Крапивенко А.В. Кросс-кадровая интерполяция видеоданных. – М.: Вузовская книга, 2000. – 77с.
3. Крапивенко А.В. Метод кросс-кадровой фрактальной интерполяции многомерных multimedia данных.: дисс. к. ф. - м. н. // МАИ. - М.: 2001. – 116 с.
4. Lyndon Hill. Phase Correlation. - <http://www.ee.surrey.ac.uk/Personal/L.Hill/pc.html> (12.10.2001)

Сведения об авторах

1. *Крапивенко Андрей Викторович, ассистент кафедры вычислительной математики и программирования Московского государственного авиационного института (технического университета), кандидат физико-математических наук.*

E-mail: kav@elias.ru

2. *Главнов Владимир Александрович, студент шестого курса факультета прикладной математики и физики Московского государственного авиационного института (технического университета).*

E-mail: jascher@mail.ru.