

УДК 004.627

СЖАТИЕ ИЗОБРАЖЕНИЙ С ПОТЕРЯМИ НА ОСНОВЕ АДАПТИВНОЙ СЕГМЕНТАЦИИ

А.Ю. Тропченко, Ю.В. Лужков

В статье излагается суть алгоритма сжатия с помощью выделения локальных областей и рассматривается возможность его применения для сжатия видеопоследовательностей.

Ключевые слова: сжатие изображений, адаптивная сегментация, квантование, аппроксимирующая функция, видеопоследовательность, MPEG.

Введение

Распространение технологий цифрового вещания и цифрового видео привело к необходимости разработки эффективных методов сжатия видеопоследовательностей. Сжатие видео основано на двух важных принципах: пространственной избыточности, присущей каждому кадру видеоряда, и временной избыточности, т.е. похожести предыдущего кадра на следующий. Таким образом, типичный метод сжатия заключается в кодировании первого кадра с помощью некоторого алгоритма сжатия изображений и последующем кодировании разности первого кадра и последующих [1, 4]. Для кодирования первого кадра предлагается использовать алгоритм сжатия с помощью выделения локальных областей. На последующих кадрах производится поиск областей, выде-

ленных на первом кадре, и вычитание их из изображения. Разностное изображение может кодироваться с помощью алгоритма JPEG, либо сохраняется среднее значение яркости фона изображения.

Описание алгоритма сжатия

Рассматриваемый алгоритм сжатия относится к фрактальным алгоритмам, которые основываются на повторяемости объектов в реальных изображениях. Фрактальные алгоритмы сжимают изображение путем преобразования доменных областей изображения в ранговые области. Ранговая область представляет собой характерный объект изображения, который повторяется в разных видах в домене. Сжатое изображение представляет собой множество ранговых областей и данные о преобразованиях, с помощью которых из каждой ранговой области можно получить доменные области и собрать из них исходное изображение. Изображение можно рассматривать как набор локальных областей (пятен близкой интенсивности), что может быть записано как

$$I(X, Y) \approx \sum_{s=1}^{N_1} f_s(x, y), \quad (1)$$

где $I(X, Y)$ – исходное изображение; $f_s(x, y)$ – функция, аппроксимирующая локальную область D_s с размерами l_{xs} , l_{ys} с центром $\{x_{0s}; y_{0s}\}$; s – порядковый номер области; N_1 – число областей.

Функция f_s , которая задает пятно, должна быть гладкой, чтобы избежать разрывов при аппроксимации больших пятен. Применение аппроксимирующей функции позволяет сохранить информацию о пятне в виде координат центра, амплитуды и размеров фрагмента используемой гармонической функции. К аппроксимирующим функциям предъявляются следующие требования:

- простота аналитического описания аппроксимирующей функции;
- наличие выраженного максимума в центре и монотонность убывания к периферии;
- осесимметричность;
- плавное сопряжение с фоновым изображением;
- простота масштабного преобразования.

К числу аппроксимирующих функций, наиболее полно удовлетворяющих указанным требованиям, могут быть отнесены, в частности, двумерные гармонические функции вида [2]

$$f_s(x, y) = \begin{cases} C_s [\cos[\omega_s(x - x_{0s})] + 1][\cos[\omega_s(y - y_{0s})] + 1]; x, y \in D_s, \\ 0; x, y \notin D_s. \end{cases} \quad (2)$$

Рассмотрим аппроксимацию выделенных локальных областей фрагментами двумерных гармонических функций вида (1). Такая аппроксимация приводит к погрешности (потерям) при построении изображения из функций, аппроксимирующих пятна, но при этом существенно упрощается задача описания пятен. Для этого нужно определить параметры: C – максимальную амплитуду области; x_0 , y_0 – координаты центра области; l_x , l_y – размеры области, однозначно определяющие частоты ω_k .

Если область имеет эллиптическую форму, то необходимо также определить его ориентацию на изображении. В этом случае при аппроксимации объекта необходимо выполнять дополнительное аффинное преобразование аргументов функции f_s , соответствующее ее развороту на угол φ в пространстве изображения. Для аппроксимации областей используется следующая функция:

$$F = \frac{\cos(\omega x + 1) \cos(\omega y + 1)}{4}. \quad (3)$$

После выделения на изображении локальных областей и их аппроксимации значение аппроксимирующей функции вычитается из изображения. Вычитание проводится до тех пор, пока в изображении не останется объектов, амплитуда которых превышает некоторый порог, отделяющий фон от объектов.

В данной работе в качестве объектов сжатия рассматриваются полутоновые цветные изображения. При этом перед самым сжатием изображения преобразуются из цветового пространства RGB в пространство YUV [3–5]. После преобразования производится прореживание цветковых плоскостей U и V так, как это делается в алгоритме стандарта JPEG (YUV 4:1:1) [3, 4, 6].

Для каждой из цветковых плоскостей общая схема алгоритма для предлагаемого метода состоит из следующих блоков.

1. Определение точки максимальной яркости (ТМЯ) на исходном изображении.
2. Проверка условия завершения работы алгоритма: если яркость ТМЯ не превышает уровень яркости фона $I_{background}$, то завершить процесс сжатия.
3. Определение по алгоритму сегментации всех точек объекта, которому принадлежит ТМЯ.
4. Определение параметров функции f_s для выбранного объекта ($C, x_0, y_0, l_x, l_y, \varphi$).
5. Вычитание значения функции f_s из исходного изображения.
6. Переход к п. 1.

Сжатие производится для кадров в пространстве YUV. Алгоритм применяется для каждой плоскости, при этом плоскости могут обрабатываться независимо друг от друга. После кодирования первого кадра видеопоследовательности для каждого следующего кадра выполняются шаги:

1. сжатие первого I-кадра в соответствии с описанным выше алгоритмом;
2. поиск выделенных областей на следующем кадре с учетом компенсации движения;
3. вычитание найденных областей из кадра и сохранение их координат;
4. если найдено менее половины областей, то этот кадр считается началом новой последовательности, переход к п. 1;
5. кодирование полученного разностного изображения;
6. переход к следующему кадру.

Рассмотрим подробнее стадии алгоритма.

Сжатие первого кадра видеопоследовательности

Выделение локальных областей

Для выделения области используется алгоритм пороговой сегментации. Суть алгоритма сводится к тому, что все точки, яркость которых находится в заданных пределах, считаются принадлежащими одному объекту. В данной работе на выбор точек накладывается еще несколько ограничений, необходимых для того, чтобы выделенные области были выпуклыми. Это требование обусловлено тем, что выпуклые области наиболее точно аппроксимируются выбранными функциями.

На каждой итерации алгоритма сжатия формируется маска области, которая представляет собой двумерный массив отметок, размер которого равен размеру изображения, и выбираются пороги I_{top} и I_{bottom} . Если точка удовлетворяет условиям принадлежности к выделяемой области, то соответствующий элемент массива отметок принимает ненулевое значение.

Сначала производится поиск точки максимальной яркости (ТМЯ) на изображении. Яркость ТМЯ является верхним пределом (I_{top}). Нижний предел определяется из соотношения

$$I_{bottom} = I_{top} \times \xi, \quad (4)$$

где $\xi = 0,25$ для яркостной компоненты (Y-плоскость) и $\xi = 0,125$ для цветоразностных компонент (плоскости U и V).

Различные значения для разных компонент обусловлены тем, что при выделении областей на плоскостях U и V алгоритм должен обладать большей чувствительностью, чем при выделении областей на плоскости Y. Это связано с тем, что цветоразностные компоненты несут меньше информации, чем Y (что позволяет производить их прореживание без заметного ухудшения качества восстановленного изображения). При этом области на плоскостях U и V не так ярко выражены, как на Y-плоскости (яркость ТМЯ ближе к яркости фона, чем на плоскости Y, т.е. плоскости U и V менее контрастны). Если использовать такое же значение ξ для этих плоскостей, как и для Y, то при выделении областей будет вноситься существенная ошибка.

Затем осуществляется проверка условия завершения алгоритма сжатия. Яркость ТМЯ сравнивается с яркостью фона $I_{background}$. Яркость фона определяется из соотношения

$$I_{background} = I_{max} \cdot \varepsilon, \quad (5)$$

где I_{max} – максимальное значение яркости в сжимаемом изображении; ε – заданный пользователем коэффициент, изменяется в пределах (0,1).

Если яркость ТМЯ не превышает $I_{background}$, то процесс сжатия завершается. Коэффициент ε позволяет регулировать точность восстановления сжатого изображения. За счет уменьшения этого коэффициента достигается увеличение точности при уменьшении коэффициента сжатия $K_{сж}$, т.е. при увеличении размера сжатого файла.

Определение яркости фона относительно самой яркой точки изображения сообщает алгоритму гибкость при работе с изображениями с разной средней яркостью.

ТМЯ также является первой точкой в выделяемой области. Анализ яркости остальных точек изображения происходит в соответствии с алгоритмом заполнения с затравкой. При этом выполняются следующие шаги.

1. Затравочная точка помещается в стек. В данном случае такой точкой будет ТМЯ.
2. Точка извлекается из стека.
3. Проверяются на принадлежность области 8 соседних точек. Точки обходятся по часовой стрелке, как показано на рис. 1. Если точка принадлежит области, то она помещается в стек. Если точка не удовлетворяет условиям принадлежности к области, то она пропускается.

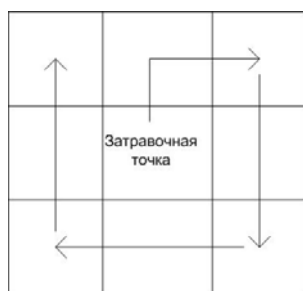


Рис. 1. Схема обхода точек

4. Если стек не пуст, осуществляется переход к п.2. В противном случае выделение области завершается.

Рассмотрим подробнее условия принадлежности области.

1. Любая из соседних точек должна быть отмечена. Выполнение этого условия гарантируется порядком рассмотрения точек в алгоритме заполнения с затравкой.
2. Яркость точки должна удовлетворять условию $I_{top} \leq I \leq I_{bottom}$. Если это условие не выполняется, осуществляется переход к следующей точке.

3. Проверяются тройки соседних точек. Если они являются отмеченными, то проверяемая точка отмечается в массиве маски, и осуществляется переход к следующей точке.

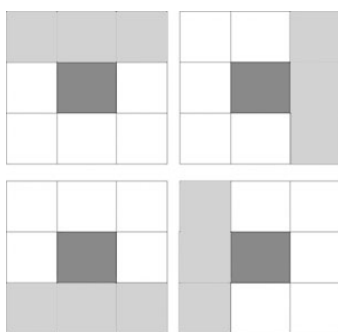


Рис. 2. Проверка троек соседних точек

На рис. 2 темно-серым помечена рассматриваемая точка, а светло-серым – уже отмеченные точки. Если имеет место одна из этих четырех ситуаций, то условие 3 считается выполненным. Это условие служит для ускорения процесса сегментации, так как позволяет быстро добавлять к области точки, которые не нарушат выпуклость области.

4. Линия, проведенная из любой точки границы области к проверяемой точке, не должна пересекать точки, яркость которых не принадлежит диапазону $[I_{bottom}; I_{top}]$. При этом данные точки могут быть не отмеченными – предполагается, что они будут присоединены к области на последующих итерациях алгоритма выделения области. Данное условие служит для обеспечения выпуклости выделяемой области.

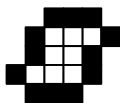


Рис. 3. Добавление новых точек на k -ом шаге в выделяемую область

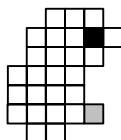


Рис. 4. Исключение точки из области при разметке

На рис. 3 белым отмечены точки, которые были отмечены на предыдущих шагах, а черным – точки, которые могут быть отмечены на k -ом шаге. На рис. 4 черным помечена точка максимальной яркости, белым – точки, которые были отмечены на предыдущих шагах, и серым – точка, которая не может быть отмечена на текущем (k -м) шаге. Условия 3 и 4 призваны обеспечить выпуклость выделенной области.

В результате работы алгоритма выделения областей получаем массив, в котором отмечены точки, входящие в выпуклую область. Далее эта область аппроксимируется заданной функцией. В том случае, если выделенная область состоит всего из одной точки, то она не аппроксимируется функцией. Выделенная точка вычитается из изображения. При восстановлении изображения яркость такой точки полагается равной среднему арифметическому 4 соседних точек.

Определение параметров аппроксимирующей функции

Для аппроксимации выделенной области с помощью гладкой функции находятся параметры функции. Определение параметров происходит в следующем порядке.

1. Определяется центр области x_0, y_0 .

2. Определяется ориентация области φ .
3. Определяются размеры области l_x, l_y .
4. Определяется амплитуда C .

Рассмотрим подробнее эти шаги. Для определения центра области x_0, y_0 считаются предварительные размеры (l_{x1}, l_{y1}) области без учета ее ориентации и ее границы $x_{\min}, y_{\min}, x_{\max}, y_{\max}$. Координаты центра вычисляются следующим образом:

$$\begin{aligned} y_0 &= y_{\min} + \frac{l_{y1}}{2}, \\ x_0 &= x_{\min} + \frac{l_{x1}}{2}. \end{aligned} \quad (6)$$

Затем определяется ориентация области φ .

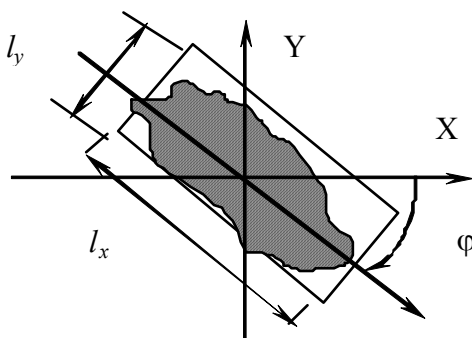


Рис. 5. Определение ориентации локальной области эллиптической формы

Для всех точек области выполняются следующие действия. Для всех направлений $i = 0, \dots, 7$ выполняется обратное преобразование:

$$\begin{aligned} x &= a_{ci}(x' - x_0) + a_{si}(y' - y_0) \\ y &= -a_{si}(x' - x_0) + a_{ci}(y' - y_0) \end{aligned} \quad (7)$$

Для каждого направления для всех точек определяются $x_{\min i}, y_{\min i}, x_{\max i}, y_{\max i}$, т.е. границы прямоугольной области, описывающей объект. Зная количество точек исходного объекта Cnt , определяем $K_{\varphi i}$ – коэффициент заполнения прямоугольной области для каждого направления:

$$K_{\varphi i} = \frac{Cnt}{(x_{\max i} - x_{\min i} + 1)(y_{\max i} - y_{\min i} + 1)}. \quad (8)$$

Выбирая $\max(K_{\varphi i})$, определяем направление (поворот) i , размеры l_x и l_y :

$$\begin{aligned} l_x &= x_{\max i} - x_{\min i} + 1 \\ l_y &= y_{\max i} - y_{\min i} + 1 \end{aligned} \quad (9)$$

Размеры l_x и l_y не должны превышать 255, так как для их хранения используются 2 байта. В случае, если размер выделенной области по любой из координат будет больше 255, то область уменьшается до максимального разрешенного размера.

Новый центр объекта (x_0^*, y_0^*) для аппроксимирующей функции f_s определяется соотношениями

$$\begin{aligned} x_0^* &= \frac{(x_{\min i} - x_{\max i})}{2}, \\ y_0^* &= \frac{(y_{\min i} - y_{\max i})}{2}. \end{aligned} \quad (10)$$

Выполняя прямое аффинное преобразование над (x_0^*, y_0^*) , получаем (x_0, y_0) в пространстве изображения:

$$\begin{aligned}x_0' &= a_{ci}x_0^* - a_{si}y_0^* + x_0, \\y_0' &= a_{si}x_0^* + a_{ci}y_0^* + y_0.\end{aligned}\tag{11}$$

Значения l_x , l_y , x_0 , y_0 округляются до целых. Таким образом, параметры φ , l_x , l_y , x_0 , y_0 функции f определены. Яркость ТМЯ данной области будет составлять амплитуду C .

Определенные таким образом параметры аппроксимирующей функции сохраняются. Значения аппроксимирующей функции вычитаются из исходного изображения.

Поиск выделенных областей на следующем кадре с учетом компенсации движения

Следующий шаг алгоритма заключается в поиске выделенных областей на следующем кадре. Для поиска используются области до их аппроксимации. Поиск происходит в окрестности выделенной области (области поиска). Размеры окрестности определяются следующим образом:

$$\begin{aligned}L_x &= 2 \Delta x, \\L_y &= 2 \Delta y,\end{aligned}\tag{12}$$

где Δx , Δy – ширина и высота прямоугольника, в который вписана выделенная область. При этом центром окрестности будет являться точка с координатами $(a + \Delta x/2, b + \Delta y/2)$, где a и b – координаты левого верхнего угла прямоугольника, в который вписана область. Для всех возможных положений прямоугольника с выделенной областью внутри области поиска вычисляется разность пикселей, принадлежащих выделенной области, и соответствующих пикселей в области поиска. Положение, при котором средняя разность пикселей (остаточная энергия изображения) минимальна, используется для построения вектора движения области. Пороговое значение остаточной энергии, выше которого области считаются несовпадающими, устанавливается экспериментально.

Вычитание найденных областей из кадра и сохранение их координат

Найденная область вычитается из кадра, при этом сохраняется вектор движения, описывающий перемещение области относительно ее положения на I -кадре, и идентификатор области.

Переход к следующему I -кадру

Если найдено менее половины областей, то этот кадр считается началом новой последовательности. Следующий кадр рассматривается как новый I -кадр, т.е. сжимается независимо от предыдущих кадров.

Кодирование полученного разностного изображения

Разностное изображение кодируется в соответствии с алгоритмом jpeg и сохраняется наряду с векторами движения и соответствующими идентификаторами областей I -кадра. После этого осуществляется переход к следующему кадру.

Восстановление видеопоследовательности

Для первого I -кадра выделенные области хранятся в явном виде. Они используются для восстановления этого кадра и всех последующих. Последующие сжатые кадры представляют собой набор координат и идентификаторов областей I -кадра, а также разностное изображение.

Восстановление кадра сводится к вычислению функций $f(x, y)$ на каждой итерации и алгебраическому суммированию. Вычислительные затраты процедуры восстановления минимальны.

После восстановления плоскостей в них производится поиск одиночных точек с нулевой яркостью. Эти точки в подавляющем большинстве случаев являются результатом отбрасывания одноточечных областей на этапе сжатия. Яркость этих точек рассчитывается как среднее арифметическое 4 соседних точек.

Заключение

Для кодирования видео предлагается использовать метод, основанный на выделении локальных областей и параметрическом их описании. Выделение областей производится на первом кадре видеопоследовательности (*I*-кадр), затем производится поиск выделенных областей на последующих кадрах.

Литература

1. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с.
2. Ожиганов А.А., Тропченко А.А., Тропченко А.Ю. Модифицированный фрактальный метод сжатия многоуровневых изображений // Информационные технологии. – 2003. – № 3.
3. Сэломон Д. Сжатие данных, изображений и звука. – М.: Техносфера, 2004 – 368 с.
4. Ричардсон Я. Видеокодирование. H.264 и MPEG-4 – стандарты нового поколения. – М.: Техносфера, 2005. – 368 с.
5. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), 2003
6. Image and Video Coding – Emerging Standards and Beyond // IEEE Transactions on Circuits and Systems for Video Technology. – V. 8. – № 7. – November 1997. – P. 814–837.

Тропченко Александр Ювенальевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, tau@d1.ifmo.ru

Лужков Юрий Валерьевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, магистр техники и технологий, аспирант, luzhkov@inbox.ru