



УДК 681.323 (075)

© С. В. Сай, Е. С. Перегуда, 2006

## МЕТОДЫ СОКРАЩЕНИЯ ОБЪЕМА ВЫЧИСЛЕНИЙ В АЛГОРИТМАХ ФРАКТАЛЬНОГО СЖАТИЯ ИЗОБРАЖЕНИЙ

*Сай С. В.* – завкафедрой «Вычислительная техника» д-р техн. наук, проф.;  
*Перегуда Е. С.* – аспирант кафедры «Вычислительная техника» (ТОГУ)

Рассматривается проблема ускорения алгоритма фрактального сжатия, приводится анализ методов сокращения объема вычислений с использованием дискретного косинусного преобразования (DCT).

In the paper, the problem of fractal compression algorithm is considered. The analysis of calculating reduction methods with DCT using is resulted.

### Введение

Разработка эффективных алгоритмов сжатия изображений является актуальной задачей при архивации или передаче видеоданных по низкоскоростным каналам связи. Такие алгоритмы могут быть применены для хранения больших объемов информации, организации видеоконференций, создания видеотелефонного сервиса, а также для разработки новых систем цифрового телевидения.

Методы сжатия изображений основываются на устранении тех или иных форм избыточности, в частности, фрактальные методы рассматривают самоподобие как источник избыточности. Считается, что самоподобие является свойством почти всех природных объектов и их изображений и, следовательно, устранение этой формы избыточности может значительно уменьшить объем данных, необходимых для описания природного объекта или его изображения.

Алгоритм фрактального сжатия основан на поиске локального самоподобия отдельных элементов анализируемого изображения, при этом изображение разбивается на отдельные блоки, и среди них выполняется поиск наиболее совпадающих блоков. В процессе поиска необходимо выполнять следующие операции: аффинные преобразования и смещение яркости.

Основным аффинным преобразованием является сжатие блока. Общее операторное выражение следующее



$$F \approx G(F) + H,$$

где  $G$  – оператор аффинного преобразования над всем множеством блоков изображения  $F$ ;  $H$  – смещение яркости.

С учетом того, что в большинстве случаев это уравнение не имеет точного решения, его можно представить в более строгом виде

$$F = G(F) + H + \varepsilon,$$

где  $\varepsilon$  – погрешность изображения. Пусть  $\tilde{F}$  – точное решение, тогда

$$F - \tilde{F} = G(F) + H + \varepsilon - G(\tilde{F}) - H = G(F - \tilde{F}) + \varepsilon,$$

откуда через неравенство Коши – Буняковского следует

$$\|F - \tilde{F}\| \leq \|G\| \cdot \|F - \tilde{F}\| + \|\varepsilon\| \quad \text{или} \quad \|F - \tilde{F}\| \leq \frac{\|\varepsilon\|}{1 - \|G\|}.$$

Это выражение имеет смысл только при условии  $\|G\| < 1$  – что говорит о том, что преобразование должно быть сжимающим. Заметим, что при этом из условия выпадает смещение яркости  $H$ .

Общий алгоритм поиска заключается в минимизации выражения

$$\sum_i \sum_j (s \cdot d_{ij} + h - r_{ij})^2, \quad (1)$$

где  $d_{ij}$  и  $r_{ij}$  – пиксели двух сравниваемых блоков изображения, называемых доменами и рангами;  $s$  – масштабирующий фактор яркости;  $h$  – смещение яркости в блоке. При этом размеры домена  $d$  превышают размер ранга  $r$ , таким образом сжимающее условие выполняется.

Целью работы является поиск методов, позволяющих сократить объем вычислений (1) для заданных размеров доменов и рангов при ограниченном количестве аффинных преобразований.

### Методы сокращения объема вычислений

При реализации алгоритма, прежде всего, ставится вопрос об оптимальном разбиении изображения на ранги и домены. Разбиение на квадродерево можно считать достаточно удобным, но глобальное разбиение приводит к значительному увеличению времени вычислений. Конечно, если изображение имеет большие области с равномерным распределением энергии, то такой алгоритм будет эффективен. Но такие изображения встречаются редко и носят скорее искусственный характер.

В связи с этим на начальном этапе исследований было принято решение отказаться от разбиения квадродеревом и ограничиться только размерами 8 на 8 пикселей для рангов и 16 на 16 пикселей для доменов.



Следующая проблема состояла в ограниченности значений яркостей, которые лежат в узком диапазоне значений от 0 до 255, следовательно, необходима проверка аппроксимации на условие  $0 \leq (s \cdot d_y + h) \leq 255$ . Проверка этого условия при реализации алгоритма приводит к невозможности конвейеризации алгоритма процессором, что побудило к переходу анализа от целочисленного к вещественному значению методом DST.

Следующим шагом был анализ количества операций, необходимых на обработку одного пикселя. Для этого потребуется две операции умножения для 63 точек во всех 8 аффинных преобразованиях, т.е. 1008 операций умножения на одну проверку сходства. Если в изображении имеется 1000 доменов и 1000 рангов, то для полного анализа потребуется более одного миллиарда операций умножения. Это слишком расточительно. Но как указывалось выше, сходимость изображения к аттрактору определяется только аффинным преобразованием  $G$ , но не яркостью. Поэтому можно поставить под сомнение необходимость операции  $s \cdot d$  при анализе, хотя значение  $s$  необходимо при восстановлении. Так как  $s$  является масштабирующим коэффициентом яркости, то вышеуказанный алгоритм сравнивает как количественные, так и качественные характеристики. Отказавшись от масштабирования, можно анализировать только качественные характеристики.

В работе [1] представлено доказательство существования минимизации Евклидовой метрики для нормированных векторов. Таким образом, задача заключается в нахождении минимального угла между нормированными векторами. При такой реализации число операций умножения сокращается вдвое. При применении аффинных преобразований к DST этот процесс приводит либо к умножению на (-1), либо к транспонированию матриц коэффициентов DST, и можно составить таблицу преобразований [2] (табл. 1).

Из анализа таблицы видно, что при отражении по горизонтали, вертикали или одновременно все четные гармоники всегда умножаются на единицу, т.е. если сравнивать отраженные изображения, то все их четные гармоники совпадут.

Таким образом, нет необходимости перебирать все восемь преобразований. Достаточно сравнивать только четные гармоники прямой и транспонированной матрицы и принимать решение о том, потребуется ли вращать изображение. При этом число операций умножения можно сократить еще в два раза или в четыре раза, если разложить изображение только по четным коэффициентам DST.

Таблица 1

Таблица аффинных преобразований

	Пространственный домен	DCT домен
0	равенство	$i_0(F(u, v)) = F(u, v)$
1	отражение от средней вертикальной оси	$i_1(F(u, v)) = (-1)^v F(u, v)$
2	отражение от средней горизонтальной оси	$i_2(F(u, v)) = (-1)^u F(u, v)$
3	отражение от первой диагонали	$i_3(F(u, v)) = F(v, u)$
4	отражение от второй диагонали	$i_4(F(u, v)) = (-1)^{u+v} F(v, u)$
5	поворот на $+90^\circ$	$i_5(F(u, v)) = (-1)^u F(v, u)$
6	поворот на $+180^\circ$	$i_6(F(u, v)) = (-1)^{u+v} F(u, v)$
7	поворот на $-90^\circ$	$i_7(F(u, v)) = (-1)^v F(v, u)$

При кодировании цветных изображений используется кодирование трех составляющих: яркости и двух цветоразностных составляющих красного и синего цветов. В частности, такое кодирование используется в алгоритмах JPEG и вейвлет сжатия. Два этих алгоритма основываются на низкочастотной фильтрации изображений и фильтруют каждую из трех составляющих по отдельности.

В отличие от JPEG, алгоритм фрактального сжатия основан на анализе структуры изображения. Так как структура составляющей яркости включает в себя структуры красного, синего и зеленого цветов очевидно, что закодированная структура фрактала яркостной составляющей будет близкой к структуре фрактала цветоразностных составляющих. Это позволяет пропустить этап поиска наиболее оптимального домена для каждой из трех составляющих и произвести расчет коэффициентов фрактала на основе одной яркостной составляющей. При этом время полного кодирования изображения увеличивается не в три раза, как можно ожидать для цветного изображения, а только на 30 %.

### Заключение

Результаты моделирования предложенных методов показали значительное ускорение анализа и значительное сокращение времени вычислений по сравнению с известными методами фрактального сжатия.

В частности, тестовое изображение «LENA» (рисунок) кодируется за 0,25 с, при этом степень сжатия составляет 14 раз. Качество изображения достаточно высокое, хотя при разработке алгоритма упор делал



ся на сокращение времени расчета и одновременно на сохранение опознаваемости. Крупные цветные изображения размером 1024x768 кодируются за время от 15 до 25 с.



Пример тестового изображения

Ресурсы данного алгоритма еще достаточно значительны. Для DST был использован алгоритм разложения по обычному базису, что сократило скорость только в 2 раза, а не в 4, как это было бы возможно при разложении по четному базису DST. Дополнительно скорость можно увеличить, если взять для анализа только первый минор матрицы DST [3], что эквивалентно уменьшению масштаба изображения и позволяет сократить время анализа еще в четыре раза.

Алгоритм реализован на высокопроизводительном языке программирования C++, но перевод алгоритма на язык Ассемблера с использованием мультимедийных технологий, таких как MMX, SSE, 3DNow, Nupur Tread и т.п. мог бы также значительно увеличить скорость выполнения алгоритма. И конечно на современных многоядерных процессорах фрактальный алгоритм мог бы показать значительную скорость, что позволило бы обрабатывать изображение в 625 строк с высоким качеством в режиме реального времени. При этом появится воз-



возможность масштабировать изображение на приемной стороне как фрактальную структуру, приближая качество к телевидению высокой четкости в 1250 строк со скоростью потока данных близкой к стандарту MPEG-2 в 625 строк.

В заключении стоит указать на тот факт, что выражение для расчета элемента изображения  $(s \cdot d_{ij} + h)$  является уравнением линейного искусственного нейрона с весовым коэффициентом  $s$  и собственным возбуждением  $h$ , и одновременно производится трассировка связей между элементами – нейронами (поиск подобия между элементами). Таким образом, на основе фракталов возможно построение различных экспертных систем искусственного интеллекта.

### Библиографические ссылки

1. *Dietmar Saupe, Raouf Hamzaoui, Hannes Hartenstein* Fractal Image Compression. An Introductory Overview // [www.compression.ru](http://www.compression.ru), 1998.
2. *G. Davis* Implicit Image Models in Fractal Image Compression / Proc. of SPIE Conf. on Wavelet Applications in Signal and Image Processing IV, Denver, / [www.compression.ru](http://www.compression.ru), 1998.
3. *Yao Zhao, Baozong Yuan* A Hybrid Compression Scheme Combining Block-based Fractal Coding and DCT // [www.compression.ru](http://www.compression.ru), 1998.