

## МЕТОДЫ ОПТИМИЗАЦИИ ФРАКТАЛЬНОГО СЖАТИЯ ИЗОБРАЖЕНИЙ

Джура М.Е.

Российский университет дружбы народов, Москва, Россия, m.dzhura@gmail.com

*В работе представлены способы увеличения производительности фрактального сжатия изображений, основывающиеся на анализе блоков изображения и их предварительной обработке.*

Ключевые слова: фрактальное сжатие изображений, классификация.

### Введение

Одним из аналогов широко распространенным методам сжатия цифровых изображений JPEG и JPEG2000 является фрактальное сжатие. Этот метод основывается на системе итерируемых функций (СИФ), которая является набором аффинных операторов.

Процесс сжатия состоит в создании сходящейся СИФ, чья неподвижная точка является искомым изображением. Процесс восстановления состоит в последовательном применении СИФ к произвольному изображению. Утверждается, что результирующее изображение будет похоже на искомое[1]. Другими словами, результирующее изображение не будет тождественно исходному, то есть при сжатии произойдет потеря части информации.

### Постановка задачи сжатия

Ключевой задачей является создание сходящейся СИФ для произвольного искомого изображения. Рассмотрим эту задачу более подробно. Пусть изображение разбито на непересекающиеся квадратные блоки — множество рангов  $R$ . Множество доменов — набор квадратных блоков изображения, размер которых больше ранговых. Множество доменов может быть увеличено, если в него также включим блоки, полученные из доменов путем применения восьми преобразований  $w$ , состоящие из комбинации поворотов (на  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ ) и отражения (к примеру, сверху вниз, слева направо или вдоль одной из диагоналей). Путём усреднения доменных блоков по соседним пикселям, их размер может быть уменьшен до размера ранговых. Таким образом, получим набор блоков  $D$ , называющихся таблицей кодов (codebook).

Задача сжатия в простейшем виде состоит в следующем: для каждого рангового блока  $r$  из  $R$  надо найти такое преобразование  $w$  и  $d$  из  $D$ , что  $\|r - w(d)\| \rightarrow \min$ , где норма изначально задана. Для поиска блока  $w(d)$ , который наилучшим образом аппроксимирует ранговую область — сегмент изображения, используется простой перебор доменных областей и их преобразований. То есть, если число доменных блоков равно  $N$ , то для поиска наилучшей пары домен-ранг потребуется  $O(N)$  сравнений. На практике этот процесс имеет высокую вычислительную сложность [5].

По этой причине актуальной является задача снижения вычислительных затрат. Для этой цели предлагается использовать классификацию блоков изображения. При этом поиск наилучшей пары домен-ранг необходимо производить только внутри одного класса, что уменьшает множество альтернатив для перебора и снижает вычислительные затраты.

### Классификация блоков изображения

Разобьем каждый блок на 4 квадранта: верхний левый, верхний правый, нижний правый и нижний левый. В каждом квадранте вычисляется среднее значение яркости  $A_i$  и среднеквадратичное отклонение  $V_i$ , где  $i = 1, 2, 3, 4$ . Можно преобразовать блок так, что средние значения яркостей упорядочатся одним из трёх способов, что соответствует определенному классу:

1.  $A_1 > A_2 > A_4 > A_3$

$$2. \quad A_1 > A_2 > A_3 > A_4$$

$$3. \quad A_1 > A_3 > A_2 > A_4$$

Так как преобразование блока фиксировано, то каждый из 3 классов имеет 24 подкласса, соответствующие расположению  $V_i$ . Поэтому, множество ранговых и доменных областей можно разбить на 72 класса.

Зная класс, к которому принадлежат ранговый и доменный блоки, и их преобразования в каноническую форму, легко найти преобразование домена  $w$ , которое минимизирует расстояние  $\|g - w(d)\|$  [2].

### Сохранение часто используемых параметров

Для оценки сходства двух блоков, вычисляется квадратичная разность светимости пикселей.[3] Для её вычисления требуется знать суммы яркостей и суммы квадратов яркостей пикселей двух блоков. Очевидно, что многократный расчет сумм одних и тех же блоков является излишним, достаточно предварительно их вычислить и сохранить.

При адаптивном разбиении, размер ранга не является фиксированным. Если поиск не дал удовлетворяющего соотношения домен-ранг, то ранг разбивается на квадранты — ранги меньшего размера. Для этих новых блоков необходима собственная таблица кодов, которая, фактически, содержит уменьшенные домены из старой таблицы. При этом потребуются перерасчёт сумм яркостей пикселей, что является трудозатратой операции. На практике, кодовая книга, созданная из изображения Лены [4] с шагом в 8 пикселей и размером 32x32 пикселя, может состоять из 12228, 86016, 77028 блоков, в зависимости от заданных параметров. В данном примере, для создания новой таблицы кодов, состоящей из блоков 16x16 пикселей, нужно вычислить  $2 \cdot 12228 = 24456$  сумм, как минимум. Для оптимизации перерасчёта сумм яркостей пикселей, предлагается вычислять среднее значение и среднеквадратичное отклонение яркости блока, из которых легко получить необходимые суммы аналитически.

### Выводы

В ходе эксперимента было выявлено, что реализация сжатия с использованием классификации и с предварительным расчетом сумм уменьшает время работы без визуальной потери качества восстановленного изображения.

### Литература

1. *Jacquin A.E.* Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. IEEE Trans. Image Processing 1 (1992) 18-30
2. *Fisher Y.* Fractal Image Compression: Theory and Application.// Springer-Verlag, 1995.
3. *Saupe D., Hamzaoui R., Hartenstein H.* Fractal Image Compression: An Introductory Overview.
4. *Munson, David C.*, A Note on Lena. IEEE Transactions on Image Processing (archive) 5 (1). January 1996.
5. *Шарабайко М.* Обзор фрактального кодека Артёма Петрова. [http://fic.bos.ru/articles/MLovic\\_APetrovCodecTest.php](http://fic.bos.ru/articles/MLovic_APetrovCodecTest.php)

## METHODS FOR OPTIMIZATION OF FRACTAL IMAGE COMPRESSION

*Dzhura M.*

*Peoples' Friendship University of Russia, m.dzhura@gmail.com*

***The paper presents ways to increase productivity of fractal image compression, based on the analysis of image blocks and their pretreatment.***

Key words: fractal image compression, classification.