

УДК 004.9

ББК 32.97

## **РЕЗУЛЬТАТЫ ФРАКТАЛЬНОГО СЖАТИЯ ИЗОБРАЖЕНИЙ ПРИ РАЗЛИЧНЫХ ФОРМАХ РАНГОВЫХ ОБЛАСТЕЙ**

В.Г. Колебошин, А.В. Крапивенко

### **Аннотация**

В статье изложен метод фрактального сжатия изображений с помощью системы итерируемых функций. Рассматриваются основные способы разбиения изображения на непересекающиеся ранговые области - триангулярное разбиение с ограничениями, где в качестве границ выступает контур изображения, квадродерево и предложенное расширение квадродерева. Приводится сравнение качества сжатых изображений и степени сжатия.

### **Ключевые слова**

фрактальное сжатие изображений; ранговые области; доменные области; триангулярное разбиение; квадродерево, оценка качества изображений.

### **1. Введение**

Задача эффективного сжатия данных всегда была одной из самых актуальных в информатике, а с развитием передачи изображений по сети Интернет она приобрела еще большее значение. Так, с начала 1980-х гг. было разработано множество алгоритмов, направленных на эффективное сжатие цифровых изображений для их передачи и хранения в компрессированном виде.

Развитие теории фракталов и соответствующего математического аппарата в конце XX в. повлекло разработку принципиально нового алгоритма сжатия изображений — фрактального, который дает высокие коэффициенты сжатия при приемлемом качестве. Идея фрактального метода сжатия применительно к изображениям была сформулирована

Майклом Барнсли в 1990 году, и довольно быстро получила резонанс в научных кругах. Большинство научных статей, издаваемых на эту тему, имеют целью устранить главный недостаток алгоритма – большое время сжатия, путем классификации и оптимизации огромного числа перебираемых подобластей изображения.

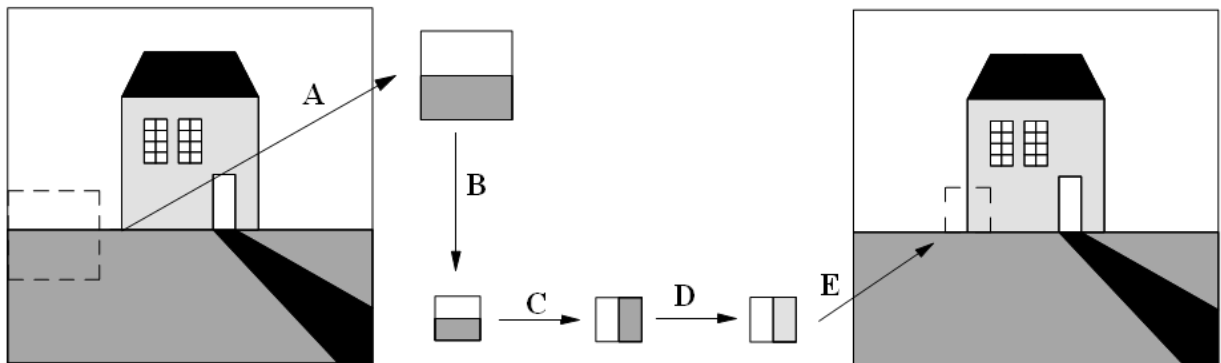
Целью данной статьи является изучение возможности модификации классического фрактального алгоритма для улучшения качества декодированного изображения и сравнение результатов со стандартным подходом. Модификация заключается в изменении формы подобласти и включение в разбиение на подобласти дополнительной информации о кодируемом изображении. В работе сравнивается обычный подход с использованием квадродерева и две модификации: *триангулярное разбиение с ограничениями* (в качестве ограничений выступает предварительно выделенная граница изображения); и *расширение структуры квадродерева*, позволяющее в некоторых случаях существенно сократить количество ранговых областей и увеличить коэффициент сжатия.

## 2. Идея метода фрактального сжатия

Основа метода фрактального кодирования – это обнаружение самоподобных участков в изображении, на основе которых в сжатый файл заносятся параметры геометрических преобразований, позволяющие восстановить большие участки изображения из подобных участков меньшего размера. В соответствии с данным методом, изображение одновременно разбивается на два множества: неперекрывающихся ранговых блоков (range subimages) и перекрывающихся доменных блоков (domain subimages). Для каждого рангового блока алгоритм кодирования находит наиболее подходящий доменный блок и аффинное преобразование, которое переводит этот доменный блок в ранговый блок. Иными словами, структура изображения отображается в систему ранговых блоков, доменных блоков и преобразований.

Такая система преобразований называется *системой итеративных функций* (Iterated Function System – IFS). Метод IFS применительно к построению фрактальных самоподобных изображений был изобретён Майклом Барнсли (Michael Barnsley) в конце 80-х годов прошлого века. Однако у него отсутствовали какие-либо сведения о решении обратной задачи: как по заданному изображению найти аффинные преобразования. В статье Барнсли было показано несколько реалистичных фрактальных изображений, но все они были созданы вручную.

В общем случае, необходимо уметь находить для любого изображения систему аффинных преобразований (IFS), воспроизводящую изображение с заданной точностью [1]:



**Рис 1. Схема фрактального кодирования с помощью IFS**

На рис. 1. буквами обозначены следующие действия: А - берем очередной доменный блок; В - сжимаем до размера рангового блока; С - применяем аффинное преобразование; D - подбираем оптимальные коэффициенты яркости и контрастности; Е - сравниваем с ранговым блоком.

Алгоритм фрактального сжатия существенно асимметричен по времени. Время кодирования изображения может превосходить аналогичное время затрачиваемое алгоритмом JPEG в 1000 раз, при этом декодирование происходит намного быстрее. Тем не менее, степень сжатия может теоретически достигать 1:2000, поэтому фрактальный алгоритм кодирования имеет специфические области применения. Например, доступ к библиотекам готовых изображений через Internet, когда более важен объем передаваемого файла и время разархивации. Также особенностью фрактального алгоритма является возможность увеличения размера изображения в 2 – 4 раза без появления блочного эффекта.

### 3. Математическое обоснование алгоритма фрактального сжатия

Есть отображение  $W: \Delta \rightarrow \Delta$ , где  $\Delta$  – множество всех возможных изображений.  $W$  является объединением отображений  $w_i$ :

$$W(R) = \bigcup_i w_i(d_i) \quad (1)$$

где  $R$  – изображение, а  $d_i$  – какие-либо (возможно, перекрывающиеся) области изображения

D. Каждое преобразование  $w_i$  переводит  $d_i$  в  $r_i$ . Таким образом:

$$W(R) = \bigcup_i r_i \quad (2)$$

Будет логично представить изображение в виде функции двух переменных  $f(x, y)$ . На множестве всех таких функций введём метрику (расстояние между изображениями), например, таким образом:

$$\delta(f, g) = \max_{x, y} (|f(x, y) - g(x, y)|) \quad (3)$$

Согласно теореме Банаха, существует определённый класс отображений, для которых существует константа  $c < 1$  такая, что для любых изображений  $f$  и  $g$  выполняется неравенство:

$$\delta(W(f), W(g)) \leq c \cdot \delta(f, g) \quad (4)$$

Такие отображения называются *сжимающими*, и для них справедливо следующее утверждение: Если к какому-то изображению  $F_0$  мы начнём многократно применять отображение  $W$  таким образом, что

$$\begin{aligned} F_1 &= W(F_0) \\ F_i &= W(F_{i-1}) \end{aligned} \quad (5)$$

то в пределе, при  $i$ , стремящемся к бесконечности, мы получим одно и то же изображение вне зависимости от того, какое изображение мы взяли в качестве  $F_0$ :

$$\lim_{i \rightarrow \infty} F_i = F \quad (6)$$

Это конечное изображение  $F$  называют *аттрактором*, или неподвижной точкой отображения  $W$ . Также известно, что если преобразования  $w_i$  являются сжимающими, то их объединение  $W$  тоже является сжимающим. Таким образом, любое начальное изображение можно перевести в искомое (точнее, в сколь угодно близкое к нему) за относительно небольшое количество итераций.

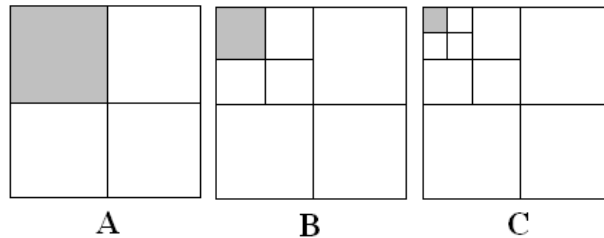
#### 4. Способы разбиения изображения на ранговые области

Рассмотрим два основных способа разбиения изображения на ранговые области: *квадродерево* и *триангуляция*.

##### 4.1. Квадродерево

Способ пространственной индексации точек плоскости – *квадродерево* – было применено М. Брансли в его первых статьях по фрактальному сжатию. При построении *квадродерева* [2] вначале производится грубое разбиение изображения, например на четыре прямоугольника. Для каждого полученного рангового блока алгоритм пытается подобрать соответствующий доменный блок и коэффициенты отображения, которые наилучшим образом покрывают ранговый блок. Если получилось покрытие в пределах допустимой

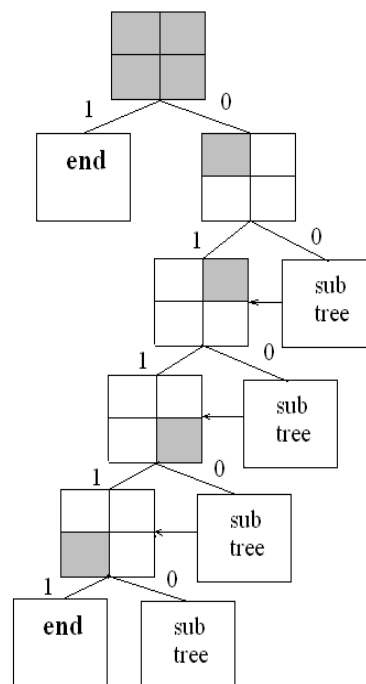
погрешности, то этот ранговый блок отмечается как обработанный и берется следующий. В случае, если покрытие не найдено, блок аналогично разбивается на 4 части. Процедура разбиения проводится до тех пор, пока не будет достигнута максимальная заданная глубина вложенности.



**Рис 2. Индексация областей в квадродереве**

Простейшая нумерация областей в квадродереве следующая: каждая область представляется в виде списка из 4х векторов, длина каждого вектора равна максимальному уровню вложенности. Значение первого вектора показывает номер прямоугольника, отсчитываемый по часовой стрелке и полученный на первом шаге разбиения. Остальные вектора определяются аналогично. Например, для закрашенных областей на рис. 2 индексация будет следующая: для А-100, для В-110 и для С-111.

В предлагаемом в данной статье варианте реализовано другое представление индекса. Все квадродерево представляется в виде обычного дерева:



**Рис. 3. Альтернативная индексация квадродерева**

При начале работы алгоритма указатель установлен на корень дерева, индекс корня — 000, т.е. ранговый блок изначально равен всему изображению. В зависимости от результата покрытия (сравнения с помощью какой-либо метрики визуального сходства блоков), возвращается значение 1 – покрытие удалось, или 0 – покрытие не удалось. Таким образом, мы переходим по одной из ветвей дерева. Блок «sub tree» устанавливает указатель на вновь сгенерированное дерево. Это дерево перебирает подобласти в пределах того блока, который его вызвал. По мере прохождения поддерева, блок, вызвавший поддерево, окажется покрытым (или будет осуществлен проход до максимального уровня вложенности), а указатель устанавливается по ветви «1» вызывающего блока.

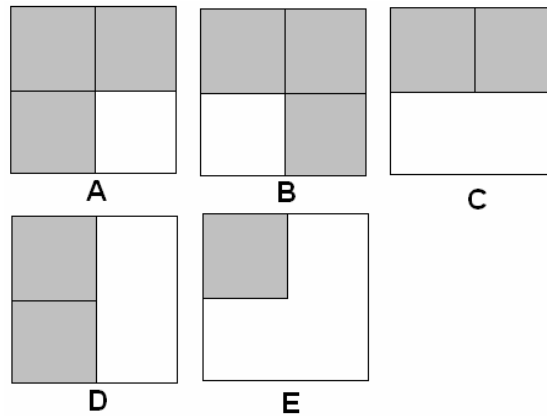
Затраты на хранение индекса у второго способа индексации существенно меньше. Например, для всех блоков обозначенных на рис 2.С размер индекса первого способа  $S=10(\text{блоков}) \cdot 2(\text{бита на 1 вектор}) \cdot 2 \cdot 2=80$  бит, для второго случая индекс в который входят результаты перехода по ветвям дерева - 000111111111111 — 15 бит. Таким образом, второй способ индексации существенно экономичней и позволяет передавать последовательность ранговых областей отдельно от коэффициентов преобразований каждой области.

#### 4.2 Расширенное квадродерево

Напомним, что при стандартном подходе разбиения изображения способом квадродерева в случае, если для рангового блока не найдено покрытие доменным блоком, то он разбивается на 4 подблока. При этом возникает множество случаев, когда ранговый блок достаточно было бы разбить всего лишь на 2 подблока для осуществления искомого покрытия. Чтобы избавиться от избыточности ранговых блоков, Ю. Фишер [4] предложил использовать так называемое HV-разбиение. При этом разбиении ранговый блок делится на 2 подблока, и линия разбиения блока проводится с учетом границ изображения. Ф Девойн (F. Davoine) для уменьшения числа ранговых блоков и улучшения аппроксимации структуры изображения предложил использовать смешанное триангулярно-квадратичное разбиение.

Здесь мы предлагаем другое расширение идеи квадродерева, где учтена возможность подразбиения непокрытого рангового блока не только на 2 подблока, но и возможно на 3, при этом сохраняется простота индексации квадродерева, описанная выше.

При разбиении рангового блока в целях сравнения сначала выдаются подблоки А, В, С, D – содержащие всевозможные сочетания четвертей (рис. 4). Если ни одно из них не удалось покрыть доменным блоком, то тогда алгоритм перебора запускается уже для первой четверти (рис. 4.Е).

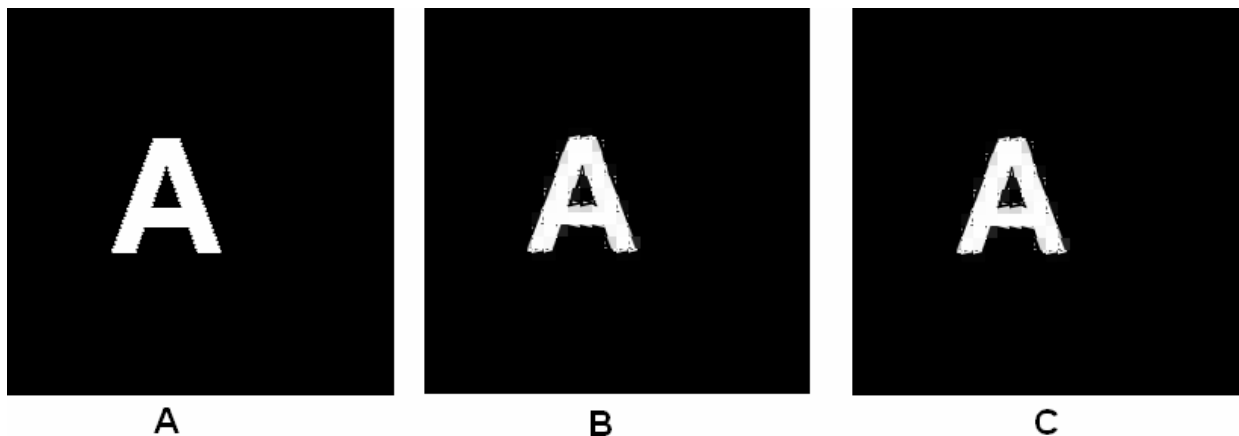


**Рис.4. Возможные сочетания подблоков**

В случае, если удалось найти приемлемое покрытие для одного из сочетаний подблоков, то оставшаяся непокрытая часть дробится аналогично, и перебираются все подразбиения, начиная с частей с максимальной площадью.

По сравнению с обычным квадродеревом, расширенный способ в большинстве случаев дает выигрыш в сжатии порядка 10-15%, эффективность от его применения также возрастает при увеличении допустимой ошибки визуального сходства блоков.

На некоторых специфических изображениях коэффициент сжатия  $K$  может возрасти до 43%, при этом значение PSNR практически не изменилось (рис. 5):



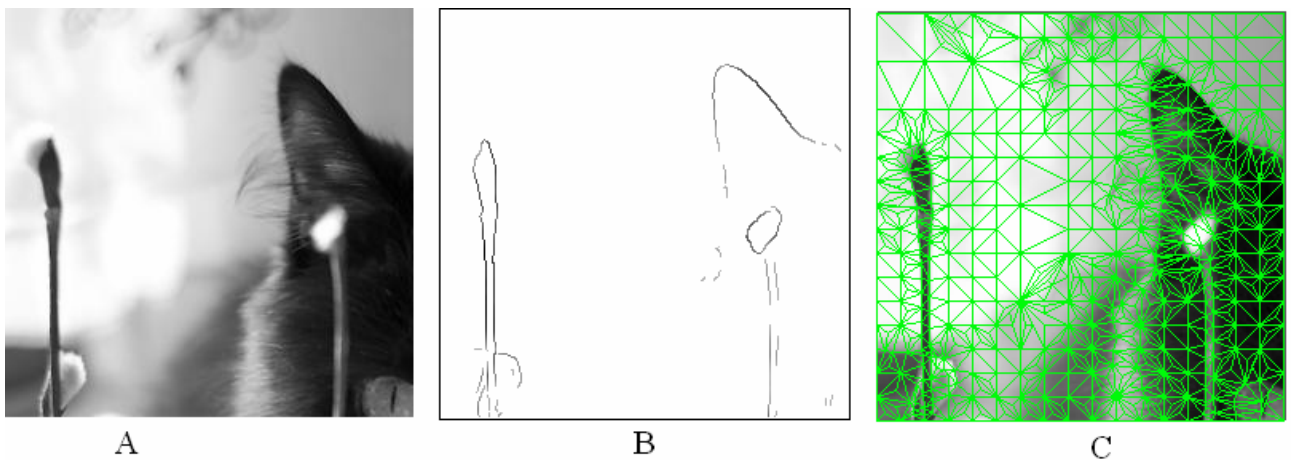
**Рис. 5. А-оригинал, В-разбиение с помощью квадродерева ( $K=76$ , PSNR=28.5), С-разбиение с помощью расширенного квадродерева ( $K=109$ , PSNR=28.5)**

### 4.3. Триангуляция

При триангуляции исходное изображение разбивается на непересекающиеся треугольники. Затраты на такое разбиение и хранение индекса изображения существенно

выше, чем у квадродерева. Это происходит за счет нерегулярности структуры разбиения. Плюсом является то, что нерегулярность структуры позволяет включить в разбиение дополнительную информацию об изображении. Например, разбиение можно строить с учетом локализованных границ изображения.

Метод триангуляции был реализован в программе следующим образом: сначала методом Кенни выделяется контур изображения, затем делается обход контура и аппроксимация его отрезками заданной длины. Далее мы накладываем точечную сетку с заданным шагом, и производим триангуляцию с ограничениями. В качестве ограничений выступают аппроксимированные линии контуров. Затем при работе алгоритма фрактального сжатия мы перебираем полученные треугольники и пытаемся покрыть их доменными областями. Если покрытие не удалось, то разбиваем этот треугольник на 2 или 3 «подтреугольника» (в зависимости от вытянутости треугольника добавляем дополнительную точку в треугольник на середину стороны максимальной длины или в точку центра масс). Критерием возможности дальнейшего подразбиения, является то, что площадь полученного треугольника должна быть больше минимального заданного значения.



**Рис. 6. А-оригинал, В-выделенный контур, С-конечное разбиение на треугольники**

Результатом реализованного подхода обеспечивается разбиение на ранговые области, достаточно точно учитывающее структурные особенности изображения.

При хранении разбиения в сжатом файле достаточно в заголовке файла иметь список всех точек, упорядоченный таким образом, чтобы впоследствии по нему можно было восстановить последовательность треугольников.



## 5. Сравнение качества изображений, получаемых при различных формах ранговых областей

Рассмотрим результаты сжатия изображений, полученные при различных способах разбиения на доменные области. Для начала определим формулы, по которым будем вычислять коэффициент сжатия.

Пусть  $N_1$  – количество ранговых областей,  $N_2$  – количество доменных областей,  $N_t$  – количество точек, полученных при триангуляции,  $N_{qt}$  – размер индекса квадродерева,  $W$  и  $H$  – высота и ширина исходного изображения. Для хранения коэффициентов преобразования доменной области достаточно  $R_1=20$  бит [2], аффинное преобразование кодируется  $R_2=8$  битами. Тогда размер закодированного файла при использовании структуры «квадродерево» будет следующим:

$$R = N_{qt} + N_1 * (R_1 + R_2 + [\text{Log}_2 N_2]).$$

При триангуляции:

$$R = N_t * ([\text{Log}_2(H)] + [\text{Log}_2(W)]) + N_1 * (R_1 + R_2 + [\text{Log}_2 N_2]),$$

где  $[X]$  – минимальное целое число, большее  $X$ . Тогда теоретический коэффициент сжатия будет равен  $K=W*H/R$ .

В подтверждение полученных формул проведем экспериментальное тестирование на типовом изображении *LENA.bmp* размером 256x256, часто используемом для оценки работы методов сжатия:

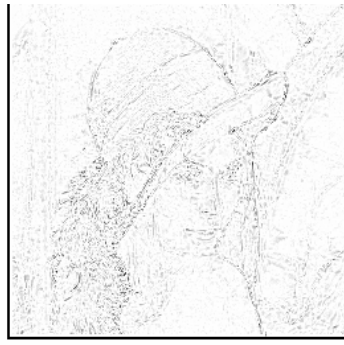


Рис. 7. Изображение – оригинал.

Далее приводятся примеры декодированных изображений и их разности с оригиналом, при одинаковых значениях коэффициентов сжатия. Использовались следующие оценки: **MSE** - *среднеквадратичное отклонение декодированного изображения от оригинала*, **PSNR** - *отношение сигнал/шум*, **k** - *расчетный коэффициент сжатия*.



A

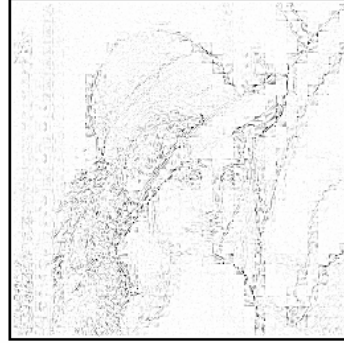


B

MSE=90.5  
PSNR=28.56  
k=15.3



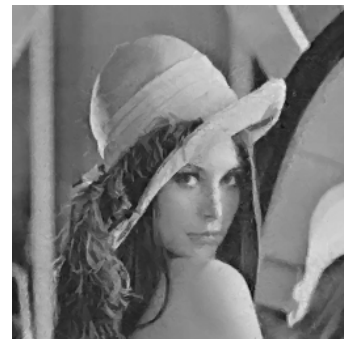
C



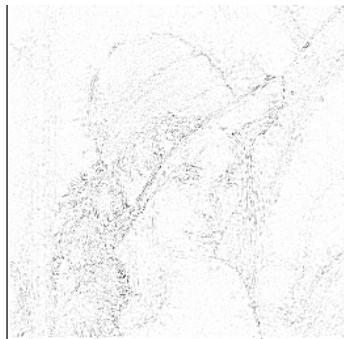
D

MSE=135.6  
PSNR=26.8  
k=16

**Рис.7. А,В – триангулярное разбиение, С,Д – разбиение с помощью квадродерева. Для средних коэффициентов сжатия.**



A



B

MSE=67.85  
PSNR=29.8  
k=7.6



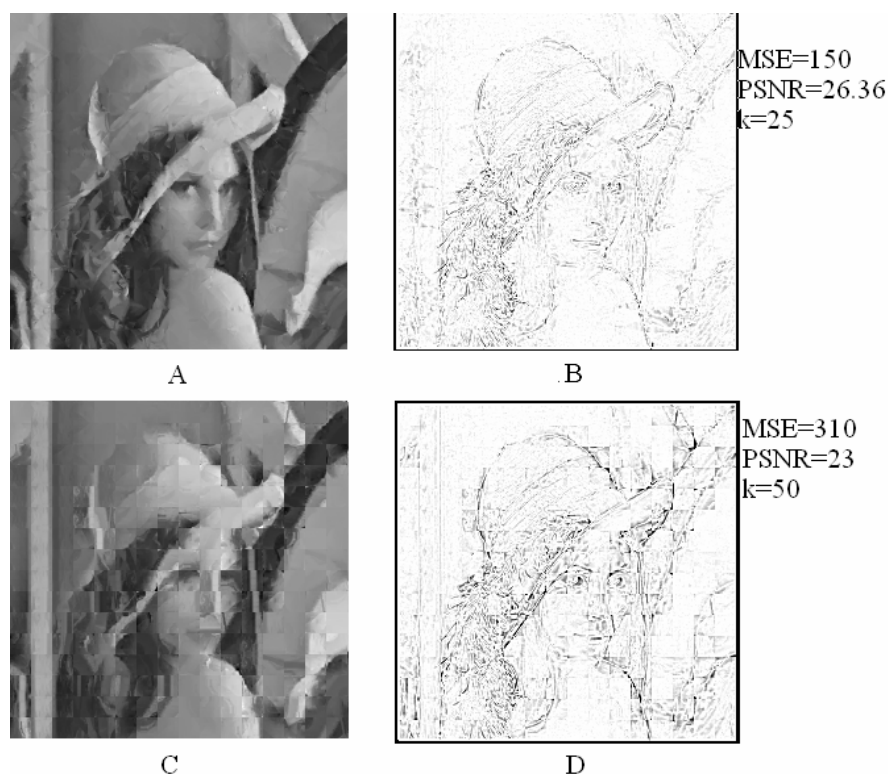
C



D

MSE=54.8  
PSNR=30.7  
k=7.7

**Рис.8. А,В – триангулярное разбиение, С,Д – разбиение с помощью квадродерева. Для маленьких коэффициентов сжатия.**



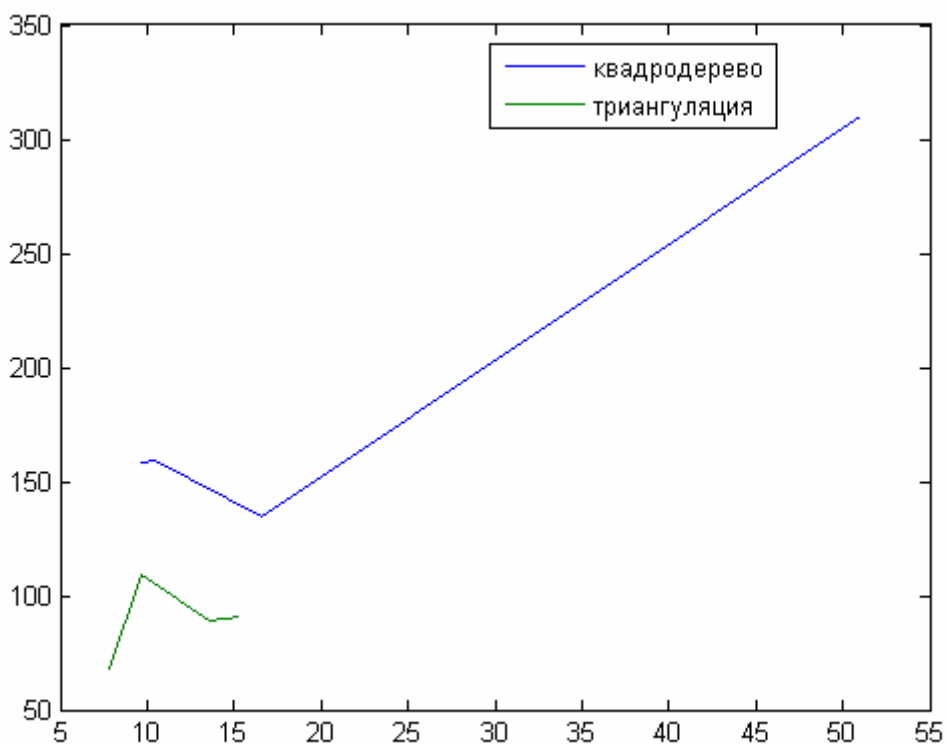
**Рис.9. А,В – триангулярное разбиение, С,Д – разбиение с помощью квадродерева. Для высоких коэффициентов сжатия.**

На приведенных парах изображений, хорошо виден эффект блочности на резких границах при разбиении с помощью квадродерева. При триангуляции, даже при высоких коэффициентах сжатия, изображение визуально выглядит более четко. Алгоритм тестировался на относительно маленьком числе доменных областей, что не дает возможности сравнить его с распространенными алгоритмами сжатия, такими как JPG или вейвлет-сжатие.

Рассмотрим график зависимости MSE от коэффициента сжатия изображения. Данные получены на основе серии экспериментов:

Тип разбиения	Число доменных областей	Число ранговых областей	Допустимая ошибка покрытия	Коэффициент сжатия	Среднеквадратичное отклонение (MSE)	Время кодирования (мин)
Квадр.д.	1024	985	5	16,6	135.2	21
Квадр.д.	1024	256	5	51	310	12
Квадр.д.	1024	1024	0	9.7	158.8	17
Трианг.	1024	1120	5	13,6	88.17	27
Трианг.	1024	2001	5	7,58	67,85	42
Трианг.	1024	997	5	15,3	90,5	20
Трианг.	1024	1051	5	14,5	89,6	22

И в графической форме:



**Рис. 10. Зависимость MSE от коэффициента сжатия изображения.**

## 6. Выводы и рекомендации

На основе предложенных подходов к выделению ранговых областей создана и апробирована программная модель фрактального сжатия. Алгоритм был протестирован на относительно низком числе доменных областей. Однако модификация формы доменной области и включение в форму разбиения дополнительной информации о границах изображения позволяет добиться уменьшения эффекта блочности и улучшение субъективного визуального качества изображения при приблизительно одинаковых значениях MSE и коэффициента сжатия. При высоких коэффициентах сжатия треугольное разбиение позволяет добиться более низкого значения MSE. Время кодирования в обоих случаях примерно одинаковое и зависит от параметров сжатия. Однако в целом, накопленная статистика позволяет сделать вывод о потенциально более качественных результатах треугольного разбиения по сравнению с использованием квадродерева.

## Библиографический список

1. *Stanislav A.* Fractal research methodology // Nature Neurosci. 1999. 1. 10-140.
2. *Уэлстид С.* Фракталы и вейвлеты для сжатия изображений в действии. Учебное пособ. – М.: Издательство Триумф, 2003 – 320 с.:ил.
3. *Guorui Jiang, Yuzhuo Zhong* Fast fractal compression based on HV partition. Department of computer science, Hebei Teacher's University, Shijiazhuang 050016
4. *Fisher Y.* Fractal Image Compression: Theory and Application, Springer-Verlag, New York, 1995.

## Сведения об авторах

**Колешин Владимир Геннадьевич**, Московский авиационный институт (государственный технический университет)

студент шестого курса. [neovovik@list.ru](mailto:neovovik@list.ru)

**Крапивенко Андрей Викторович**, Московский авиационный институт (государственный технический университет)

доцент кафедры «Вычислительная математика и программирование», к.ф.-м.н. 8-985-364-75-58, [kav@elias.ru](mailto:kav@elias.ru)