

УДК 004.62

**М. П. Шарабайко**, аспирант,  
**Н. Г. Марков**, д-р техн. наук, проф., зав. кафедрой,  
Томский политехнический университет,  
e-mail: sme\_box@tpu.ru

## Исследование эффективности кодирования цветных изображений с помощью фракталов\*

*Предложены и исследованы алгоритмы фрактального сжатия цветных изображений. Исследованы структуры распределения бит в файлах фрактально сжатых изображений. На основе полученных результатов увеличена степень сжатия одного из алгоритмов. Показаны направления дальнейших исследований по данной тематике.*

**Ключевые слова:** фрактальное сжатие цветных изображений, структура файла фрактального кода, цветовые системы, RGB, YUV

### Введение

Задача сжатия видеоинформации является актуальной и востребованной, применяемой как для хранения домашнего фотоархива, так и для длительного хранения космических, медицинских и иных статических изображений и видеопоследовательностей. При этом в одном случае важна высокая степень сжатия изображений при приемлемых, мало заметных для глаза потерях, а в другом — как можно меньше потерь или полное их отсутствие при хорошей степени сжатия.

В настоящее время решение такой задачи чаще всего осуществляется с помощью алгоритмов, основанных на дискретном косинусном преобразовании (ДКП). Для повседневного сжатия изображений часто используется алгоритм *JPEG* [1]. Сжатие видео строится на алгоритмах *MPEG-2* [2], *H.264* [3], *H.265* [4], по сути, являющихся все тем же сжатием на основе ДКП, с добавлением предсказаний, устранением избыточности хранения информации между соседними данными.

Фрактальное сжатие статических изображений строится на абсолютно ином принципе — на самоподобии частей изображения, при этом информация о значениях яркости пикселей в сжатом потоке

\* Работа выполнена при финансовой поддержке гранта УМНИК.

не хранится, а хранится лишь информация о структуре изображения, о взаимозависимостях его частей. Развитием работ по сжатию изображений являются исследования, нацеленные на создание быстроедействующих алгоритмов фрактального сжатия изображений в градациях серого, результаты которых изложены в статье [5].

В данной статье рассмотрен общий подход к представлению и сжатию цветных изображений. С учетом особенностей идеи фрактального сжатия предложены и исследованы несколько алгоритмов фрактального сжатия таких изображений.

### Задачи фрактального сжатия цветных изображений

#### **Особенности фрактального сжатия изображений.**

Фрактальное сжатие изображений выполняют в несколько этапов. Сначала проводят разбиение изображения на множество кодируемых (сжимаемых) блоков, исторически называемых ранговыми, и на множество доменных блоков, которые будут использоваться для сжатия, являясь чем-то вроде словаря кодирования.

На следующем этапе происходит кодирование каждого рангового блока. Оно основано на поиске такого доменного блока из множества сформированных, который с помощью преобразований поворота, изменения контраста и сдвига яркости будет максимально подобен текущему ранговому блоку. Более подробно процесс поиска соответствий изложен в работе [5].

Наконец, полученные коэффициенты преобразований и структура разбиения изображения сохраняются для его последующего восстановления декодером.

**Представление цветных изображений.** Обычно цветное изображение имеет три цветовых компоненты (плоскости, канала), вместо одной компоненты для изображений в градациях серого.

Самой распространенной цветовой моделью, используемой для цифрового представления цветных изображений и для передачи цвета в телевизорах и мониторах, является аддитивная цветовая модель *RGB*. В ней используется красный (*R*), зеленый (*G*) и синий (*B*) каналы для хранения информации о цвете (рис. 1, см. третью сторону обложки).

Система *RGB* имеет большой цветовой охват, но основной ее недостаток применительно к сжатию изображений — равнозначность каждого канала. Из рис. 1 видно, что каналы изображения имеют примерно одинаковую структуру, тем не менее,

потери в любом из каналов приводят к равноценным потерям в качестве цветного изображения.

Система  $YUV$  [6], с учетом особенностей восприятия цвета человеческим глазом, представляет цвет в виде канала яркости ( $Y$ ) и двух каналов цветности ( $U$ ,  $V$ ). При таком представлении яркостная составляющая имеет приоритет перед каналами цветности, поскольку цвет воспринимается человеческим глазом хуже яркости, и ошибки в каналах  $U$  и  $V$  менее заметны, нежели в канале  $Y$  (рис. 2, см. третью сторону обложки).

Этим свойством системы  $YUV$  пользуются при сжатии изображений, не только допуская больше потерь в плоскостях цветности, но и прореживая цветностные каналы изображения.

В системе  $RGB$  каждый пиксель изображения занимает 24 бита (по 8 бит на каждую компоненту). В системе  $YUV$  возможно представление цветного изображения теми же 24 битами на пиксель ( $YV24$ ). Однако преобразовав изображение из системы  $RGB$  в систему  $YUV$  [7], можно уменьшить плоскости цветности в 2 раза по ширине и высоте. В результате каждые 8 бит цветности обеих плоскостей соответствуют четырем яркостным пикселям изображения (рис. 3). Такая система обозначается  $YV12$  [6], поскольку каждый пиксель изображения представляется 12 битами (в среднем).

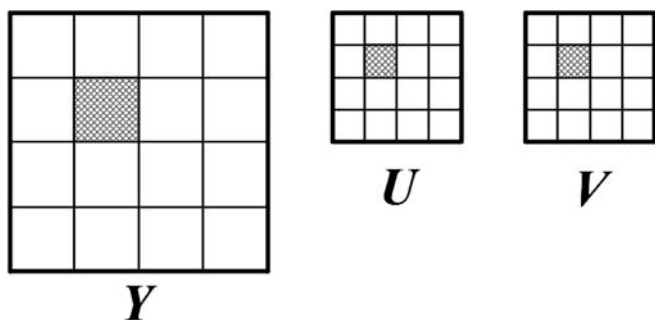


Рис. 3. Представление изображения в пространстве  $YV12$

### Фрактальное сжатие цветных изображений

Поскольку фрактальное сжатие изображений не хранит информацию о значениях пикселей изображения, а выявляет взаимозависимости отдельных его участков, нет необходимости использовать цветовую систему  $YV12$ . Незначительные потери, приносимые использованием этой системы, исказят изображение и изменят результат сжатия, никак не повлияв на степень сжатия.

При этом возможность кодирования каналов цветности с дополнительными потерями может оказаться достаточно полезной, поэтому при фрактальном сжатии цветного изображения целесообразно работать в цветовой системе  $YUV$ , 24 бита на пиксель ( $YV24$ ).

Самый простой способ фрактально сжать цветное изображение — закодировать каждую цвето-

вую плоскость отдельно, независимо от других. При таком подходе цветное изображение равнозначно трем изображениям в градациях серого. Плюсы данного подхода заключаются в простоте реализации и нахождении наилучших соответствий для ранговых блоков каждой плоскости, т. е. качество сжатия должно быть наилучшим. Отрицательной стороной подхода является игнорирование взаимозависимостей цветовых плоскостей изображения. Число блоков для перебора возрастает в 3 раза (в сравнении с независимым сжатием одной плоскости или сжатием изображения в градациях серого), и, следовательно, возрастает время сжатия.

Цветовые плоскости изображений обычно имеют похожую структуру (например, изображения на рис. 1, рис. 2, см. третью сторону обложки), т. е. имеют определенную корреляцию, которую можно более эффективно использовать при фрактальном сжатии цветных изображений. Например, найдя наиболее подходящий доменный блок для данного рангового блока одной из цветовых компонент, можно считать, что вероятнее всего наиболее подходящий доменный блок для ранговых блоков двух остальных цветовых плоскостей будет находиться в том же месте изображения.

В итоге, учитывая все изложенное, были предложены следующие алгоритмы фрактального сжатия цветных изображений (ниже описана суть этих алгоритмов).

*Алгоритм А.* Независимое сжатие каждого канала цветного изображения. При этом не используется никакая информация о корреляции значений яркости пикселей между каналами.

*Алгоритм Б.* Каждый макроблок изображения состоит из трех каналов. Для рангового блока каждого канала поиск доменного блока ведется независимо среди множества доменных блоков текущего канала. Решение о дополнительном разбиении макроблока принимается исходя из средней ошибки сопоставления блоков.

*Алгоритм В.* Поиск доменно-рангового сопоставления ведется для ранговых блоков яркостной составляющей. При нахождении удачного соответствия блоков далее ищутся коэффициенты преобразований для ранговых блоков цветоразностных компонент по доменным блокам, находящимся на одном участке изображения вместе с доменным блоком, выбранным для сжатия яркостного рангового блока.

### Исследование эффективности алгоритмов фрактального сжатия

Для оценки эффективности предложенных алгоритмов они были реализованы на языке программирования  $C++$  в среде *Microsoft Visual Studio 2010 Express*.

Для оценки потерь в декодированном (восстановленном) изображении (оценка качества изо-

бражения) используются метрики *SSIM* и *PSNR*, реализованные в работе [8].

Исследования проводили для библиотеки тестовых изображений группы фрактального кодирования и анализа из университета Ватерлоо (Канада) [9]. Использовались изображения *lenna* (рис. 4, а, см. третью сторону обложки) *peppers* (рис. 4, б) и *frymire* (рис. 4, в).

Численные эксперименты с использованием разработанных программных реализаций алгоритмов фрактального сжатия цветных изображений проводили на ПЭВМ со следующими значимыми характеристиками:

- процессор: *Intel Core i3 530* 2,93 ГГц,
- ОЗУ: 2 Гбайт *DDR3*,
- ОС *Windows 7* × 64.

В табл. 1 приведены результаты исследования указанных алгоритмов с ранговыми блоками размерами 4—8 пикселей в ширину и высоту, 7 бит — под коэффициенты изменения контраста, 5 бит — под коэффициенты сдвига яркости (обоснование выбора именно таких значений приведено в работе [10]). Здесь же для сравнения показаны результаты сжатия тестовых изображений с помощью алгоритма *JPEG*.

*Алгоритм А* позволяет сжимать цветовые плоскости изображений независимо друг от друга и является более адаптивным, поскольку удачное соответствие для разных цветовых компонент одного и того же участка изображения можно найти на разных уровнях разбиения. Это дает оптимальную степень сжатия и лучшее качество восстановленного изображения по сравнению с прочими алгоритмами.

*Алгоритм Б* дал самые плохие результаты. Независимый поиск блоков на одном уровне разбиения позволяет сократить лишь несколько бит за счет информации о разбиении каждого макроблока, при этом теряется возможность нахождения удач-

ного соответствия на более высоких уровнях разбиения методом квадродерева, что дало бы лучший коэффициент сжатия. Для этого алгоритма невысокие значения имеют качество, коэффициент сжатия и время сжатия.

*Алгоритм В* эффективнее остальных использует взаимосвязь цветовых плоскостей. При таком подходе найти удачное соответствие блоков сложнее, но вместо трех индексов доменных блоков можно хранить один общий (домены берутся с одного участка в соответствующих цветовых плоскостях). Поскольку коэффициенты доменных индексов занимают около 30 % объема файла сжатого изображения, то алгоритм позволил снизить размер файла при несущественных потерях качества.

Все предложенные и реализованные алгоритмы по соотношению степени сжатия к качеству декодированного изображения уступают алгоритму *JPEG*.

Для исследования возможностей повышения степени фрактального сжатия цветных изображений необходимо исследовать структуру файлов фрактально сжатых изображений. В экспериментах изучалось, какую часть закодированного изображения занимает индексация доменных блоков, биты яркости и контраста, биты коэффициентов преобразований поворота, при этом принималось во внимание число ранговых блоков на каждом уровне разбиения. Результаты в виде структуры файлов фрактально сжатых изображений алгоритмами *А*, *Б*, *В* представлены на рис. 5 (см. третью сторону обложки). Из него видно, что для алгоритмов *А* и *Б* существенную часть фрактального кода занимает индексация доменов, поскольку для блоков каждой компоненты хранится отдельный индекс. Вместе с тем алгоритм *В* за счет использования одной доменной области на все компоненты каждого рангового блока порождает файл фрактального кода

Таблица 1

Результаты работы алгоритмов фрактального сжатия цветных изображений и алгоритма *JPEG*

Имя алгоритма	Размер исходного изображения, Кбайт	Размер сжатого изображения, Кбайт	Время сжатия, с	Метрики качества	
				<i>SSIM</i>	<i>PSNR</i>
Изображение <i>lenna</i> (512 × 512)					
А	768	160	13,632	0,974	33,809
Б		161	13,755	0,974	33,809
В		114	5,857	0,970	33,087
<i>JPEG</i>		38	< 0,5	0,986	39,191
Изображение <i>peppers</i> (512 × 512)					
А	768	120	10,683	0,962	29,151
Б		148	14,130	0,962	29,135
В		112	5,739	0,957	28,558
<i>JPEG</i>		42	< 0,5	0,989	40,508
Изображение <i>frymire</i> (1024 × 1024)					
А	3570	564	275,471	0,955	22,678
Б		772	333,479	0,957	23,393
В		477	195,933	0,955	23,232
<i>JPEG</i>		676	< 0,5	0,999	41,177

Таблица 2

Результаты работы алгоритма *В* при различном числе бит, выделенных под коэффициенты цветности

Контраст цвета, бит	Яркость цвета, бит	Исходный размер, Кбайт	Размер сжатый, Кбайт	Время сжатия, с	Метрики качества		
					<i>SSIM</i>	<i>PSNR</i>	
Изображение <i>lenna</i> (512 × 512)							
6	4	768	106	6,398	0,966	32,542	
6	3		101	7,368	0,957	31,576	
5	4		101	6,498	0,966	32,480	
5	3		97	7,306	0,958	31,575	
6	4		105	6,529	0,952	28,567	
Изображение <i>peppers</i> (512 × 512)							
6	3	768	102	7,685	0,944	28,167	
5	4		101	6,513	0,953	28,451	
5	3		97	7,855	0,944	28,108	
6	4		3570	446	221,416	0,951	23,145
6	3			432	239,343	0,942	22,720
5	4	428		220,808	0,952	23,198	
5	3	413		240,020	0,942	22,782	
6	4	446		221,416	0,951	23,145	

с меньшим процентным соотношением битов индексации доменов.

Поскольку по времени сжатия и степени сжатия алгоритм *B* превосходит два других при не столь существенных потерях качества, то его будем модифицировать и исследовать далее. Действительно, повысить степень сжатия изображения с помощью алгоритма *B* можно за счет выделения меньшего числа бит под коэффициенты яркости и контраста на каналы цветности. Результаты таких исследований приведены в табл. 2. Видно, что снижение числа бит под коэффициенты контраста и яркости для компонент цветности уменьшило размер файла, однако не столь значительно, причем с привнесением дополнительных потерь в качестве изображения. Рациональным можно считать выделение 5 бит под контраст и 4 бит под яркость, так как при примерно одинаковой степени сжатия потери в декодированных изображениях заметно меньше.

### Заключение

Предложены и программно реализованы три быстродействующих алгоритма фрактального сжатия цветных изображений. В результате проведенных исследований показаны преимущества алгоритма *B* (основной поиск ведется по яркостной компоненте) по степени сжатия изображений и по потерям в декодированных изображениях в сравнении с другими предложенными алгоритмами. Для этого алгоритма изучено распределение бит в структуре фрактально сжатых файлов, на основе

чего предложен подход к увеличению степени сжатия изображений с помощью алгоритма *B*.

В качестве направлений дальнейших исследований можно изучать эффективность алгоритма *B* при расширении доменного пула блоками из соседних цветовых компонент. На наш взгляд, для дальнейшего повышения степени фрактального сжатия цветных изображений следует вести исследования, нацеленные на использование методов арифметического кодирования коэффициентов преобразований.

### Список литературы

1. **JPEG** File Interchange Format. Version 1.02. 1992 [Электронный ресурс]. URL: [www.w3.org/Graphics/JPEG/jfif3.pdf](http://www.w3.org/Graphics/JPEG/jfif3.pdf) (дата обращения 22.06.2011).
2. **ISO/IEC 13818**. Information technology. Generic coding of moving pictures and associated audio information. 1998.
3. **ITU-T H.264**: Advanced video coding for generic audiovisual services. 2010 [Электронный ресурс]. URL: <http://www.itu.int/rec/T-REC-H.264-201003-1/en> (дата обращения 12.07.2011).
4. **Bross B.** et al. High-Efficiency Video Coding (HEVC) text specification draft 8 // JCT-VC, Stockholm, SE, 11–20 July 2012.
5. **Шарабайко М. П., Осокин А. Н.** Быстродействующий алгоритм фрактального сжатия изображений // Известия Томского политехнического университета. 2011. Т. 318, № 5. С. 52–57.
6. **Recommended 8-Bit YUV Formats for Video Rendering** [Электронный ресурс]. URL: [http://msdn.microsoft.com/en-us/library/windows/desktop/dd206750 %28v = vs.85 %29.aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd206750%28v%3Dvs.85%29.aspx) (дата обращения 07.01.2012).
7. **Converting Between YUV and RGB**. URL: <http://msdn.microsoft.com/en-us/library/ms893078.aspx> (дата обращения 07.01.2012).
8. **Сидоров Д. В.** Программный продукт imq оценки качества изображений / Оценка качества изображений. 2011. URL: <http://imq.vt.tpu.ru/> (дата обращения 20.02.2011).
9. **Test image repository** / Fractal coding and analysis group. URL: <http://links.uwaterloo.ca/Repository.html> (дата обращения 20.02.2011).
10. **Fisher Y.** Fractal Image Compression — Theory and Application. — N. Y.: Springer-Verlag, 1994. 341 p.

УДК 004.93

**В. К. Гулаков**, канд. техн. наук., проф.,  
**С. Н. Огурцов**, аспирант,  
**А. О. Трубаков**, канд. техн. наук., доц.,  
Брянский государственный  
технический университет,  
e-mail: [Gulakov@tu\\_bryansk.ru](mailto:Gulakov@tu_bryansk.ru)

## Сегментация пейзажных изображений

*Рассмотрен вопрос сегментации изображений, дан анализ популярных алгоритмов и предложен модифицированный алгоритм для решения задачи сегментации пейзажных изображений. Также представлены результаты сравнительного анализа эффективности популярных алгоритмов и алгоритма, предложенного в статье.*

**Ключевые слова:** сегментация, алгоритмы сегментации, алгоритм водораздела, пирамидальная сегментация, контурная сегментация, *CBIR*

### Введение

Объем графической информации в сети Интернет очень велик и продолжает постоянно расти, поэтому обработка изображений и поиск отдельных объектов на них является актуальным направлением. Сегментация изображения является одним из основных этапов процесса обработки, и от того насколько качественно будет проделано выделение однородных объектов, зависит эффективность работы всей системы в целом. Известен ряд алгоритмов, которые отличаются по скорости выполнения и по качеству решения [1]. Однако идеального алгоритма для всех применений не существует, поэтому необходимо продолжать исследование особенностей разных типов изображений и разрабатывать новые методы и алгоритмы. В данной работе предложен новый алгоритм сегментации пейзажных изображений, имеющих свои особенности, а также проведено сравнение с наиболее распространенными