УДК 004. 921

О. Ю. Софина, к. т. н., доц.; А. В. Лозун

РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА ФРАКТАЛЬНОГО КОДИРОВАНИЯ

В статье предложена модификация алгоритма фрактального сжатия путем использования дискретного косинусного преобразования, вейвлет-преобразования Хаара и параллельных вычислений. Целью разработанного алгоритма является уменьшение вычислительной сложности алгоритма фрактального кодирования и повышение быстродействия процесса сжатия изображений.

Ключевые слова: *сжатие изображений, фрактальное кодирование, дискретное косинусное преобразование, вейвлет-преобразование, параллельные вычисления.*

Введение

На сегодняшний день благодаря значительному распространению электронно-вычислительной техники хранение и обработка данных разнообразного типа осуществляется преимущественно в цифровом виде. Все большую популярность приобретают мультимедийные типы данных: видео-, аудиозаписи и цифровые изображения. Исходя с вышеописанного, одной из наиболее актуальных проблем современных информационных технологий является разработка эффективных методов компрессии мультимедийных данных, в частности графической информации.

К мультимедийной информации чаще всего применяют методы сжатия с потерями. Такие применения являются справедливыми, поскольку для мультимедийных объектов, обычно, можна отказаться от сохранения некоторых их особенностей (например, мелких деталей на изображении), что позволит повысить степень сжатия. Но существуют распространенные алгоритмы сжатия без потерь, такие как: FLAC для звуковых файлов или PNG для цифровых изображений. При использовании этих форматов следует принимать во внимание, что их проектировали как универсальные для своего типа данных – и в результате они оказались непригодными для учета особенностей конкретного файла, подлежащего сжатию. Как следствие происходило значительное ухудшение коэффициента сжатия по сравнению с аналогичными алгоритмами компрессии с потерями.

Для цифровых изображений как класса мультимедийных данных наиболее распространенным форматом сжатия с потерями является JPEG. Он стал широко использоваться благодаря цифровым фотоаппаратам, сканерам и т. п. С учетом количества представленных в формате JPEG изображений становятся очевидными значительные потери, связанные с хранением, передачей и обработкой, возможно, неоптимально (по качеству и степени компрессии) сжатой информации.

Следовательно, приобретают актуальность исследования методов сжатия, базирующиеся на других представлениях изображений, к которым относится фрактальное кодирование. Его особенностью является свойство самоподобия изображения. Однако существующие методы фрактального сжатия изображений требуют существенного развития вместе с одновременным учетом многих критериев (в частности, скорости работы, степени сжатия, качества при декомпрессии) для того, чтобы их можна было рассматривать в качестве реальной альтернативы JPEG для многих классов изображений, использующихся в научно-технической и повседневной сферах жизнедеятельности человека.

Решению проблем и задач фрактального кодирования посвящено большое количество исследований отечественных (Д. С. Ватолин, В. В. Сергеев, В. А. Сойфер, В. В. Александров, Н. Д. Горский) и зарубежных (М. Барнсли, А. Жакен, Ю. Фишер, Д. Заупе) ученых [1 – 3]. Однако и сегодня есть куда двигаться в этом направлении.

Целью этой статьи является построение модифицированного алгоритма фрактального

кодирования,что позволит повысить быстродействие процесса сжатия изображений, а также достичь приемлемого коэффициента сжатия путем сочетания известных технологий ДКП, вейвлет-преобразования и параллельных вычислений.

Основными задачами, поставленными для достижения цели, являются разработка и исследование оптимального способа разбития изображения на доменные и ранговые блоки, разработка индексной и дескрипторной методики сопоставления доменных и ранговых блоков, а также сочетание разработанных методик для создания целостного алгоритма.

Разбитие изображения на доменные и ранговые блоки

Изображение разбивают на ранговые блоки одинакового размера r и на всевозможные доменные блоки размером 2r. Данное множество доменов называют основным. Далее формируют дополнительное множество доменов следующим образом: для каждого из «основных» доменов строят семь трансформаций, а именно: повороты блока на 90°, 180° и 270°, отображение блока относительно вертикальной оси симметрии и повороты полученного отраженного блока также на 90°, 180° і 270°. В результате сочетания основного и дополнительного множества доменов формируют расширенное множество доменных блоков. Для каждого рангового блока осуществляют поиск среди расширенного множества доменов. Для этого каждому ранговому и доменному блоку присваивают индекс и дескриптор.

Приведенный способ разбития изображения на доменные и ранговые блоки является оптимальним, поскольку каждый блок может быть однозначно идентифицирован его порядковым номером (при предварительно заданном фиксированном способе нумерации): для хранения номера блока необходимо меньшее количество бит, чем, например, для координат его верхнего кута. Еще одним преимуществом является отсутствие необходимости указывать размер каждого блока в сжатом файле: достаточно один раз в заголовке файла указать размер рангового блока, а также коэффициент пропорциональности для определения размера доменного блока.

Расчет индексов для доменных и ранговых блоков

С целью вычисления индекса для каждого блока b рассчитывают матрицу B его двумерного дискретного косинусного преобразования (ДКП). Исходя из анализа работы [4], предлагаем следующий алгоритм расчета индекса с использованием полученных ДКП-матриц:

– рассматривают «верхний левый угол» матрицы B, соответствующий текущему блоку b (подматрица $\widehat{B} = \{B_{ij}\}_{i,j=\overline{1,3}}$).

– соответственно с порядком выбора элементов для вычисления индекса, приведенного на рис. 1, оценивают элементы этой подматрицы с 1-го по *t*-ый, где *t* – выбранная размерность индекса, $t \in [1, 8]$. Если текущий элемент меньше нуля, соответствующему разряду индекса в двоичном представлении, начиная слева, присваивают значение 0, если больше либо равен нулю – значение 1.

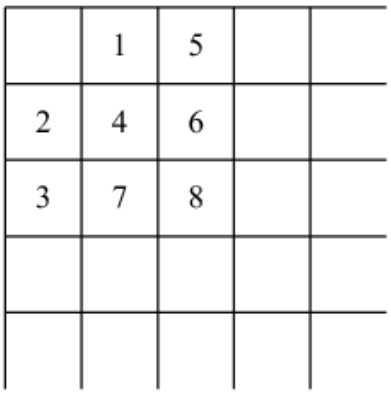

Рис. 1. Порядок выбора элементов для вычисления индекса

При применении к блоку b поворотов или отображения правила изменения, ДКП-матрицы B задают такими равенствами:

$$\begin{cases} F_{ij}^{'} = (-1)^{i+1} F_{ij}, f_{ij}^{'} = f_{N-i,j}, \\ F_{ij}^{'} = (-1)^{j+1} F_{ij}, f_{ij}^{'} = f_{i,N-j}, \\ F_{ij}^{'} = F_{ij}, f_{ij}^{'} = f_{i,j}, \end{cases} \quad (1)$$

где $i, j = \overline{1, N}$.

Отсюда следует, что ДКП-матрицы (соответственно, и индексы) «дополнительных» доменов можно найти довольно просто на основе предварительно вычисленных ДКП-матриц «основных» доменов.

Расчет дескрипторов для доменных и ранговых блоков

Для вычисления дескрипторов используют двумерное дискретное вейвлет-преобразование Хаара [5]. Предлагаем следующий алгоритм формирования дескриптора (для блока b размером p×p пикселей, где p=2q, q∈N):

– вычисляют коэффициент e_1:

$$e_1 = \frac{1}{p^2} \sum_{i=1}^{p} \sum_{j=1}^{p} b_{ij}. \quad (2)$$

– блок разбивают на четыре равных подблока b_2, b_3, b_4 и b_5.

– последовательно вычисляют коэффициенты

$$\begin{cases} e_2 = e_1 - \frac{1}{q^2} \sum_{i=1}^{q} \sum_{j=1}^{q} b_{ij} \\ e_3 = e_1 - \frac{1}{q^2} \sum_{i=1}^{q} \sum_{j=q+1}^{q} b_{ij} \\ e_4 = e_1 - \frac{1}{q^2} \sum_{i=q+1}^{q} \sum_{j=1}^{q} b_{ij} \\ e_5 = e_1 - \frac{1}{q^2} \sum_{i=q+1}^{q} \sum_{j=q+1}^{q} b_{ij} \end{cases} \quad (3)$$

что соответствуют подблокам b_2, b_3, b_4 і b_5.

– формируют вектор e = $[e_2 e_3 e_4 e_5]$, который и будет искомым дескриптором.

Как и вычисление индекса, формирование дескриптора для дополнительного домена может быть упрощено, а именно: вместо расчета по вышеприведенному алгоритму можна осуществить простую перестановку компонент дескриптора основного домена в соответствии с конкретной трансформацией блока.

Обобщающий алгоритм поиска оптимальных доменов

1. Рассматривают первый ранговий блок r_1.

2. Формируют подмножество D_{r_1} расширенного множества D доменов путем выборки всех доменов с полностью либо частично совпадающим значением индекса методом иерархического поиска (домены, что принадлежат D_{r_1}, названы обладающими «первичным сходством» с ранговым блоком, который рассматривают, либо просто «первичными»).

3. Дескриптор блока r_1 по очереди сравнивают с дескрипторами каждого из принадлежащих D_{r_1} доменов.

4. Формируют подмножество D_1 множества D_{r_1} путем выбора некого количества доменов, дескрипторы которых обладают максимальным сходством с дескриптором рангового блокаr_1 в значении эвклидового расстояния (домены, которые принадлежат D_1, называются обладающими «вторичным сходством» с ранговым блоком, который рассматривают, либо просто «вторичными»).

5. Среди найденных «вторичных» доменов ищут домен d_1, а также соответствующие паре $r_1 - d_1$ коэффициенты яркости и контрастности s_1 и o_1, исходя из условий минимизации СКО: $d_1 = argmin(\text{СКО}(r_1, \tilde{\tilde{d}}_1))$, где $\tilde{\tilde{d}}_1 = s\tilde{d}_1 + oE$, $\tilde{d}_1$– сжатый до размера блока r_1 блок d_1, а E – единичная матрица размера блока r_1. Доменный блок, для которого будет выполняться приведенное условие минимизации СКО, называется оптимальным или наилучшим.

6. Если найденный на предыдущем шаге домен d_1 принадлежит «основному» множеству доменов, фиксируют его порядковый номер v_1, если же «дополнительному» – фиксируют номер трансформации τ_1, путем которой он был получен из «начального» домена, а также номер текущего «начального» домена $\tilde{v}_1$.

7. Рассматривают второй ранговый блок r_2, для которого повторяются шаги 2 – 6, а потом ранговый блок r_3 и так далее, пока не будет найден оптимальный домен d_N для последнего рангового блока r_{N_r}, где N_r – количество ранговых блоков.

На рис. 2 приведена схема предложенного алгоритма поиска оптимальных доменов, которая учитывает распараллеливание вычислений на уровне ранговых блоков, так как для каждого из них необходимо выполнять определенное количество независимых операций.

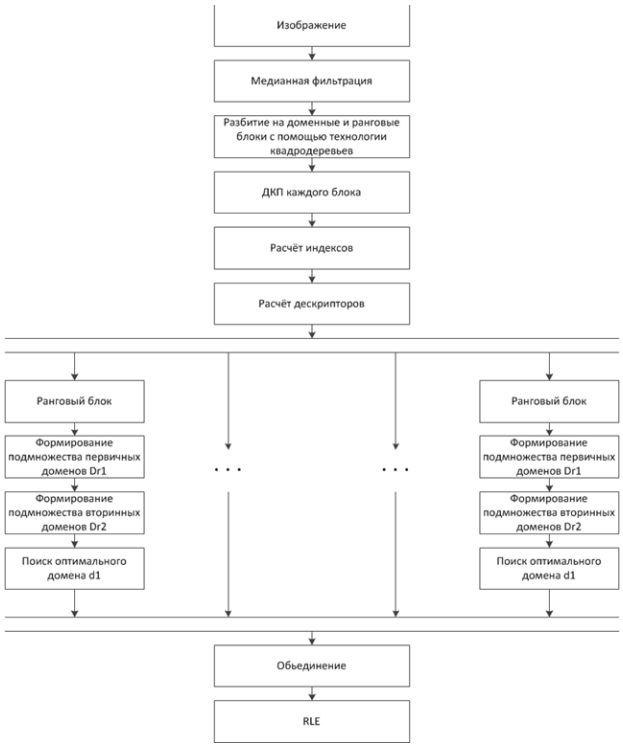

Рис. 2. Алгоритм поиска оптимальных доменов

Модификация формулы расчета среднеквадратического отклонения

Модификация формулы расчета СКО делают таким образом, чтобы уменьшить вычислительную нагрузку на систему.

Общая формула преобразования значений пикселей доменного блока выглядит следующим образом:

$$D_i^* = sD_i + o, \tag{4}$$

где D_i^* и D_i – преобразованный и начальный i-ый доменный блок соответственно; s – коэффициент изменения контраста; o – коэффициент сдвига по яркости.

Для оценки расхождения (расстояния) между преобразованным доменными и текущим ранговым блоками необходимо ввести соответствующую метрику. Обычно используют функцию среднеквадратического отклонения (СКО):

$$Q = \sum_{i=1}^{N}(D_i^* - R_i)^2 = \sum_{i=1}^{N}\left((sD_i + o) - R_i\right)^2, \tag{5}$$

где R_i – i-ый ранговый блок; D_i^*и D_i – преобразованный и начальный i-ый доменный блок соответственно; N – количество пикселей в ранговом блоке.

Очевидно, что чем меньше расстояние между блоками, тем более они сходственны.

Коэффициентыsиo можна найти из формулы (5), взяв частичные производные по этим переменным. Раскроем квадрат в выражении (5):

$$Q = \sum_{i=1}^{N} s^2 D_i^2 + 2soD_i + o^2 - 2sR_iD_i - 2oR_i + R_i^2. \tag{6}$$

Имеем следующее условие:

$$\begin{cases} \frac{\partial Q}{\partial s} = \sum_{i=1}^{N}(2sD_i^2 + 2D_io - 2R_iD_i) = s\sum_{i=1}^{N} D_i^2 + o\sum_{i=1}^{N} D - \sum_{i=1}^{N} R_iD_i = 0 \\ \frac{\partial Q}{\partial o} = \frac{1}{N}\sum_{i=1}^{N} R_i - \frac{1}{N}s\sum_{i=1}^{N} D_i \end{cases} \tag{7}$$

Выразим сдвиг по яркости:

$$o = \frac{1}{N}\sum_{i=1}^{N} R_i - \frac{1}{N}s\sum_{i=1}^{N} D_i \tag{8}$$

Подставим (8) в уравнение частичной производной по s (7) и получим следующие формулы для нахождения коэффициентов:

$$\begin{cases} s = \frac{N\sum_{i=1}^{N} R_iD_i - \sum_{i=1}^{N} R_i\sum_{i=1}^{N} D_i}{\sum_{i=1}^{N} D_i^2 - \left(\sum_{i=1}^{N} D_i\right)^2} \\ o = \frac{1}{N}\left(\sum_{i=1}^{N} R_i - s\sum_{i=1}^{N} D_i\right) \end{cases} \tag{9}$$

Преобразовав формулу (3), получим выражение для нахождения расстояния:

$$Q = s^2\sum_{i=1}^{N} D_i^2 + No^2 + \sum_{i=1}^{N} R_i^2 - 2s\sum_{i=1}^{N} R_iD_i + 2so\sum_{i=1}^{N} D_i - 2o\sum_{i=1}^{N} R_i \tag{10}$$

Формулы (6, 7) позволяют упростить вычислительную нагрузку, поскольку суммы $\sum_{i=1}^{N} R_i$, $\sum_{i=1}^{N} R_i^2$, $\sum_{i=1}^{N} D_i$, $\sum_{i=1}^{N} D_i^2$ можна посчитать еще до начала перебора, когда уже сформированы множества ранговых и доменных блоков. Тогда на этапе сопоставления необходимо посчитать только сумму $\sum_{i=1}^{N} R_iD_i$ и найти коэффициенты.

Экспериментальные результаты тестирования разработанного метода

Проведем сначала исследование классического алгоритма фрактального сжатия. Для этого была сформирована тестовая выборка, содержащая разные по своему содержанию и структуре типы изображений, которые встречаются чаще всего:

- наземные изображения сцен искусственных объектов;
- наземные изображения природных объектов;
- аэрокосмические изображения;

– астрономические изображения;
– изображения неструктурированных объектов, таких как: облака или дым.

На рис. 3 приведены примеры изображений этих классов, которые далее будут использоваться с целью исследования.

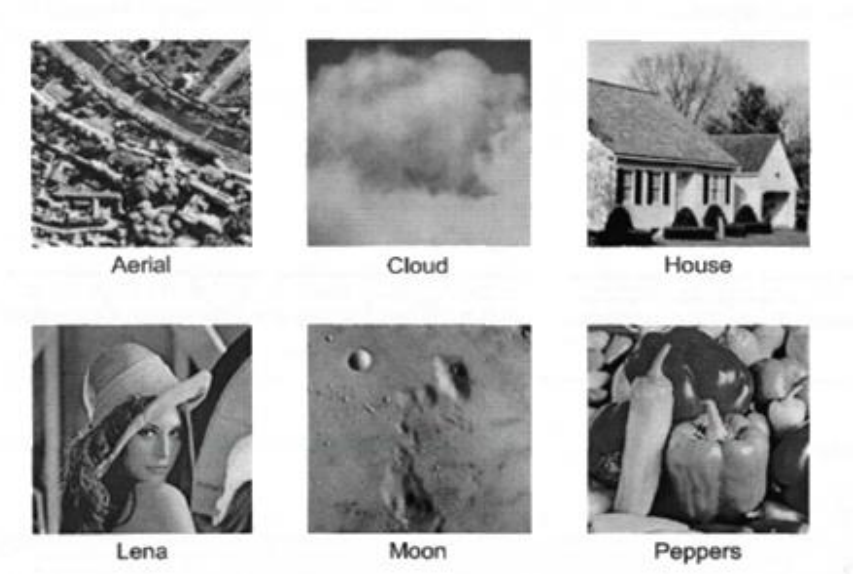

Рис. 3. Тестовые изображения

Рассмотрим показатели эффективности, достигнутые классическим алгоритмом фрактального сжатия. В табл. 1 приведены значения времени работы кодера, среднеквадратического отклонения восстановленного изображения после декомпрессии от исходного, соотношения сигнал-шум, а также компоненты критерия минимальной длины описания (МДО) и его итогового значения для изображений разного размера – 60, 120, 180 и 240 пикселей, сжатых при фиксированном размере рангового блока, что составляет 6 пикселей. Дадим определение критерия МДО. Принцип минимальной длины описания сформируем следующим образом: среди множества моделей необходимо выбрать ту, которая позволит минимизировать сумму длины описания модели (в битах) и длину данных, описанных путем применения этой модели (в битах). Под моделью подразумевают определенную информационную структуру, которая представляет собой закодированное изображение. В качестве длины описания модели будем рассматривать объем L_{img} сжатого изображения в битах. Под длинной данных, описанных путем использования модели, имеют в виду ту часть данных, которая не вошла в саму модель. Другими словами, это объем потерь L_{loss} в следствии сжатия.

Таким образом, критерий МДО можна рассчитать по формуле:

$$L = L_{loss} + L_{img} \tag{11}$$

Рассмотрим условные обозначения, которые используют в табл. 1. Тут N – размер изображения, t – время вычислений в секундах, σ – среднеквадратическое отклонение (СКО) восстановленного изображения от оригинального, PSNR – соотношение сигнал-шум, L_{loss} – оценка объема потерь информации при сжатии в битах, L_{img} – оценка длины сжатого файла в битах, L – сумма значений L_{loss} и L_{img} – минимальная длина описания сжатого изображения. Минимизация этой длины осуществляется по всем возможным соответствиям доменных и ранговых блоков при их фиксированных размерах.

Таблица 1

Результаты тестирования классического алгоритма для изображений разного размера

N	t	σ	PSNR	L_{img}	L_{loss}	L
Aerial						
60	0,82	32,365	16,685	3022	25233	28255
120	16,04	21,336	21,547	13014	92574	105588
180	91,10	13,840	25,307	30421	187285	217706
240	287,56	12,329	26,311	55485	322719	378204
Cloud						
60	0,82	3,343	35,133	3023	12913	15936
120	15,97	1,720	40,903	13014	38951	51965
180	87,24	1,146	44,429	30421	69578	99999
240	286,70	1,121	44,665	55485	124105	179590
House						
60	0,82	21,403	21,520	3023	21725	24748
120	16,03	13,743	25,367	13014	78847	91861
180	86,64	9,088	28,959	30421	159497	189918
240	301,86	8,765	29,138	55485	281237	336722
Lena						
60	0,82	14,308	24,121	3023	20531	23554
120	15,94	8,918	28,561	13014	69354	82368
180	86,44	6,216	31,731	30421	135884	166305
240	288,86	6,086	31,916	55485	238511	293996
Moon						
60	0,82	8,141	29,916	3023	16859	19882
120	16,04	5,685	32,652	13014	62723	75737
180	86,43	4,100	35,806	30421	127659	158080
240	298,76	4,414	35,196	55485	235437	290922
Peppers						
60	0,82	16,944	22,345	3023	21298	24321
120	15,97	9,946	27,757	13014	71099	84113
180	86,34	6,019	32,538	30421	134361	164782
240	284,15	5,334	32,767	55485	228856	284341

Рассмотрим, насколько высоким является качество сжатия с использованием классического фрактального алгоритма. Поскольку достаточно хорошим качеством восстановленного изображения обычно считают величину PSNR~30, в табл. 2 приведены оценки коэффициентов сжатия в случае восстановления изображения с таким качеством путем деления исходного количества бит в изображении на значение L_{img}. Данное значение соответствует наиболее близкому к 30 значению PSNR в табл. 1. Кроме того в табл. 2 также приведены коэффициенты сжатия для формата JPEG при настройках, которые соответствуют PSNR~30.

Таблица 2

Коэффициенты сжатия при PSNR~30

Изображение	Коэффициент сжатия	
	Фрактальный алгоритм	JPEG
Aerial	3,6	4,4
Cloud	59,4	47,3
House	8,2	8,0
Lena	14,7	15,8
Moon	33,3	24,2
Peppers	14,8	15,1

Стоит отметить, что при использовании стандартных программных средств коэффициент сжатия JPEG может быть намного худшим через неоптимальные настройки и запись большого объема дополнительной информации в файл.

Как видно из таблицы, коэффициенты сжатия существенно отличаются для разных изображений.

Протестируем теперь разработанный модифицированный алгоритм фрактального кодирования и сравним результаты. Так как основным недостатком классического алгоритма является быстрый рост времени работы при увеличении размеров изображений, очевидной является неперспективность его использования на практике. В данной статье были описаны разработанные методы поиска соответствий доменных и ранговых блоков в фрактальном кодировании. Осуществим экспериментальную проверку разработанного алгоритма и определим потери качества изображения. Поскольку повышение быстродействия не влияет на размеры сжатого файла, достаточно будет рассмотреть значения СКО и PSNR для восстановленного изображения.

В табл. 3 приведены характеристики качества восстановленного изображения, полученного после сжатия модифицированным алгоритмом, а также время его работы на изображениях разных размеров.

Таблица 3

Результаты тестирования модифицированного алгоритма

N	t	σ	PSNR	L_{img}	L_{loss}	L
Aerial						
60	0,09	35,010	16,120	3022	25626	28648
120	0,48	24,425	20,373	13014	95273	108287
180	1,68	16,261	23,906	30421	194581	225002
240	3,93	14,431	24,943	55485	335408	390893
Cloud						
60	0,09	4,021	33,532	3023	13659	16682
120	0,50	1,992	39,628	13014	41788	54802
180	1,74	1,312	43,251	30421	75539	105960
240	4,01	1,235	43,780	55485	131567	187052
House						
60	0,09	25,520	19,992	3023	22599	25622
120	0,47	17,643	23,198	13014	82894	95908
180	1,60	11,260	27,098	30421	167840	198261
240	4,14	15,148	24,522	55485	300273	355758
Lena						
60	0,09	17,702	22,158	3023	21563	24586
120	0,53	10,83	27,429	13014	73525	86539
180	1,66	7,389	30,374	30421	143374	173795
240	4,10	7,076	30,497	55485	248954	304439
Moon						
60	0,09	9,981	27,097	3023	17621	20644
120	0,45	6,564	31,785	13014	65029	78043
180	1,62	4,755	34,342	30421	133436	163857
240	4,05	5,000	33,909	55485	244204	299689
Peppers						
60	0,09	22,30	20,072	3023	22770	25793
120	0,48	12,073	26,181	13014	75027	88041
180	1,60	7,388	30,303	30421	142956	173377
240	4,04	6,416	31,671	55485	241339	296824

Как видно из табл. 3, скорость работы не только выше по сравнению с классическим

алгоритмом (в 30 раз на изображениях размером 120 пикселей), но и имеет другую зависимость от размера изображения. В следствии этого на изображениях размером 60 пикселей получаем 8-кратный выигрыш, а на изображениях размером 240 пикселей – уже 70-75-кратный выигрыш.

Путем сравнения таблиц 1 и 3 можно сделать вывод, что критерий качества минимальной длины описания (МДО) в результате использования оптимизированного алгоритма ухудшается в среднем приблизительно на 4%. При этом для части изображений наблюдаем постоянный рост ухудшения (например, с 1,5% до 3,5% для Aerial) при увеличении размера изображения, тогда как для других изображений может происходить даже уменьшение ухудшения. Таким образом, выигрыш в быстродействии по сравнению с классическим алгоритмом значительно возрастает при увеличении размера изображения, тогда как проигрыш в коэффициенте сжатия от размера почти не зависит либо не зависит совсем.

В данной работе также было предложено использование параллельных вычислений. Отметим, что распараллеливание вычислений применялось к уже оптимизированному алгоритму фрактального кодирования, который был описан выше. На рис. 4 приведен график, отображающий результаты сравнения последовательного и параллельного алгоритмов. Так как распараллеливание вычислений независит от структуры изображения, а также никак не влияет на качество восстановленного изображения, для тестирования было выбрано 4 изображения разных размеров. Следовательно, рассмотрим зависимость времени сжатия от размера изображения. Работа параллельного алгоритма в данном случае реализована для 16-процессорной системы.

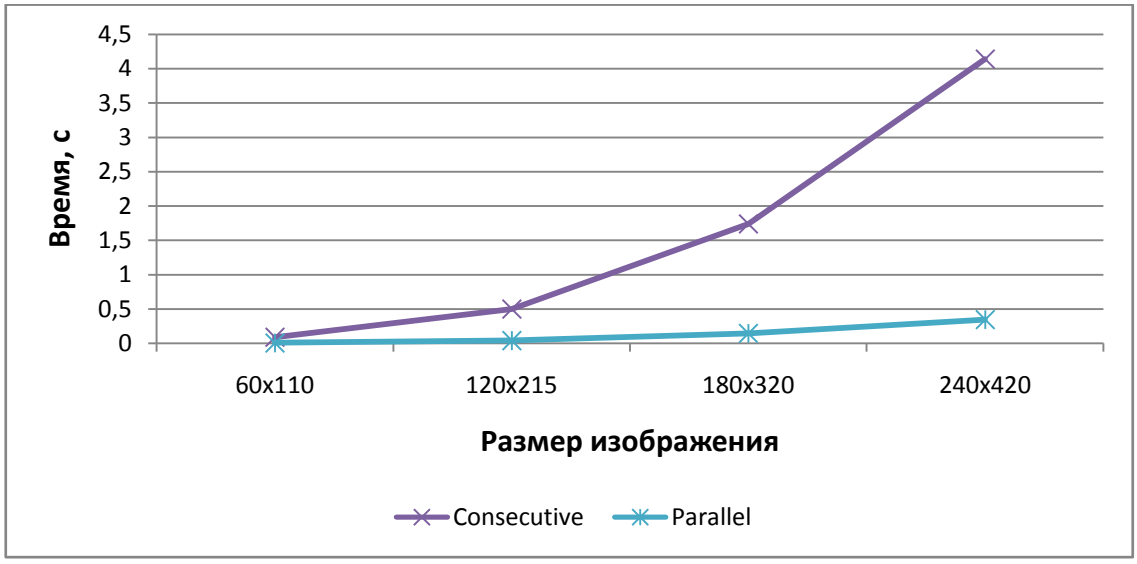

Рис. 4. Зависимость времени выполнения от размера изображения

Как видим, скорость обработки изображений по параллельному алгоритму гораздо выше, чем время работы последовательного алгоритма. Было установлено, что использование каждого дополнительного процессора приводит к повышению быстродействия на 60 – 75%.

Выводы

Разработанный алгоритм поиска оптимальных доменов, который непосредственно являет собой модифицированный алгоритм фрактального кодирования, позволяет повысить быстродействие обработки изображения за счет использования ДКП, вейвлет-преобразования Хаара и параллельных вычислений, а также позволяет уменьшить вычислительную нагрузку на систему.

Так как фрактальное кодирование относится к классу NP-полных задач, допустимое решение таких задач достигается только при использовании эффективных предметно-зависимых эвристик. Большинство работ по фрактальному сжатию посвящены именно разработке подобного типа эвристик.

Для субоптимального выполнения фрактального сжатия необходимо ограничить перебор доменных блоков, ранговых блоков и соответствий между ними, что и было предложено в данной статье. Эвристики должны отсекать неперспективные способы разбития изображения на ранговые блоки без выполнения последующих шагов сжатия, а также при фиксированном разбитии изображения на ранговые блоки – отсекать неперспективные подмножества доменных блоков для конкретного рангового блока. В данной работе формирование предметно-специфических эвристик осуществляли с помощью наиболее распространенных методов, которыми сейчас являются коэффициенты ДКП и вейвлет-разложение соответственных блоков. Эти преобразования позволяет построить признаки, несущественно искажающие попиксельную меру сходства блоков.

Уменьшение вычислительной сложности достигают путем уменьшения попиксельных сравнений за счет использования индексов и дескрипторов. Попиксельные сравнения осуществляют только при оценке соответствия вторичных доменов ранговому блоку. Также на уменьшение вычислительной нагрузки влияет модифицированная формула расчета среднеквадратического отклонения, поскольку значения сумм и квадратов сумм доменных и ранговых блоков можна вычислить еще до начала перебора.

Исходя из вышеописанного, предложен модифицированный алгоритм фрактального сжатия, эффективность которого подтверждена на практике.

СПИСОК ЛИТЕРАТУРЫ

1. Ватолин Д. С. Тенденции развития алгоритмов архивации графики / Д. С. Ватолин // Открытые системы. – 2010. – № 2. – С. 15 – 24.

2. Fisher Y. Fractal image compression using iterated transform / Y. Fisher, E.W. Jacobs, R. D. Boss // Naval Ocean Systems Center: NOSC Technical Report. – 1991. – № 1408. – P. 1122 – 1128.

3. Jacquin A. Image coding based on a fractal theory of iterated contractive image transformations / A. Jacquin // IEEE Transactionson Image Processing. – 1992. – No 1. – P. 18 – 30.

4. Ватолин Д. С. Использование ДКП для ускорения фрактального сжатия изображений / Д. С. Ватолин // Программирование. – 1999. – № 3. – С. 51 – 57.

5. Дискретное вейвлет-преобразование [Електронний ресурс]: – Режим доступу: https://ru.wikipedia.org/wiki/Дискретное_вейвлет-преобразование. – Назва з екрану.

Софина Ольга Юрьевна – к. т. н., доцент кафедры автоматики и информационно-вычислительной техники.

Лозун Алла Васильевна – магистрант кафедры автоматики и информационно-вычислительной техники.

Винницкий национальный технический университет.