МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра информационных систем и технологий

На правах рукописи

УДК 004.832.3

Сахибназарова Виктория Бахтиёровна

ИССЛЕДОВАНИЕ алгоритмов фрактального сжатия изображений

Автореферат

выпускной квалификационной работы магистра

по направлению подготовки 09.04.01 «Информатика и вычислительная техника»

профиль «Программное обеспечение мобильных устройств»

Самара – 2018 год

Работа выполнена в Самарском национальном исследовательском университете имени академика С. П. Королева.

Научный руководитель: к.т.н., доцент, Кудрина М.А.

Научные консультанты:

Рецензент:

Общая характеристика работы

**Актуальность темы**

Одной из главных характеристик любого канала связи является скорость передачи информации. Необходимо передавать как можно больше информации в сообщении наименьшего размера. В случае передачи графической информации для уменьшения объема передаваемых данных используются различные методы сжатия изображений. Все алгоритмы сжатия изображений обобщенно можно разделить на две группы: сжатие без потерь информации и сжатие с частичной потерей информации. Как правило, алгоритмы сжатия с потерей информации имеют более высокие коэффициенты сжатия по сравнению с алгоритмами сжатия без потерь. При этом качество изображения ухудшается не сильно, порой изменения даже незаметны для человеческого глаза. Кроме того, при сжатии изображений для передачи их по каналам связи повышается надежность передачи [1]. Надежность растет с уменьшением трафика, поскольку время передачи сокращается, и тем самым увеличивается объем передаваемой информации за единицу времени без увеличения ширины канала.

Одним из алгоритмов сжатия изображений с частичной потерей информации является фрактальное сжатие. Образно процесс фрактального сжатия можно описать как поиск самоподобных областей на изображении. Данный метод основан на использовании системы итерируемых функций Iterated Function System (IFS) [2]. Так как метод базируется на самоподобии элементов изображения, он заключается в моделировании рисунка несколькими меньшими фрагментами его самого. Специальные уравнения позволяют переносить, поворачивать и изменять масштаб участков изображения; таким образом, эти участки служат компоновочными блоками остальной части картины[3].

Сама система итерируемых функций представляет собой набор трехмерных аффинных преобразований, переводящих одно изображение в другое. Преобразованию подвергаются точки в трехмерном пространстве (х\_координата, у\_координата, яркость). С учетом использования IFS для осуществления фрактального сжатия (или фрактальной компрессии) исходное изображение делится на подобласти квадратной формы, называемые *ранговыми блоками*. Ранговые блоки пересекаться не могут. Также на исходном изображении выделяют *доменные блоки* (домены), являющиеся совокупностью 4-х соседних ранговых блоков. Домены могут пересекаться. Все ранговые блоки и домены – это квадраты со сторонами, параллельными сторонам исходного изображения. В ходе работы алгоритма компрессии для каждого рангового блока производится поиск домена, который после аффинных преобразований (сжатие доменного блока в 4 раза, поворот и/или отражение) и изменения яркости наименее всего отличается от рангового блока.

Достоинствами фрактальной компрессии являются степень сжатия на уровне JPEG при сравнительно одинаковом качестве, быстрый процесс декодирования и независимость восстанавливаемого изображения от разрешения (хранится структура изображения, а не данные о пикселях). Недостатками являются большие временные затраты сжатия и невозможность гарантировать ту или иную степень потерь (качество декодированного изображения зависит от самоподобия сжимаемого).

Таким образом, выбранная тема является актуальной, так как объем графической информации, которым мы оперируем ежедневно, чрезвычайно велик и поэтому изучение методов сжатия изображений имеет важное практическое значение. Также представляется интересным исследовать возможность ускорения выполнения фрактального сжатия изображений.

**Целью работы** является исследование вариантов реализации алгоритма фрактального сжатия изображений, а также изучение подходов, позволяющих ускорить выполнение данного алгоритма.

В соответствии с поставленной целью в выпускной квалификационной работе магистра решаются следующие **задачи**

**исследования:**

1. Изучение основного алгоритма фрактального сжатия и вариантов его реализации;
2. Изучение методов повышения скорости выполнения фрактального сжатия изображений;
3. Разработка и реализация программного и информационного обеспечения, позволяющего применять изученные алгоритмы и методы;
4. Исследование зависимости скорости сжатия исходного изображения и качества декодируемого изображения от использованного алгоритма.

**Методы исследования**, используемые в выпускной квалификационной работе магистра, основаны на положениях теории вероятностей и математической статистики.

**Научная новизна** работы заключается в исследовании времени выполнения фрактального сжатия изображения и качества декодируемого изображения, в зависимости от выбранного варианта реализации, используемого метода ускорения и параметров сжатия.

**Практическая ценность** работы заключается в разработке программного обеспечения автоматизированной системы, позволяющего решать следующие задачи:

1. Выбор изображения для проведения исследования.
2. Выбор варианта реализации алгоритма, метода ускорения и параметров сжатия.
3. Получение данных о времени сжатия, времени декомпрессии, предоставление изображения, полученного в результате декодирования и оценки его качества.

**Положения, выносимые на защиту:**

1. Исследование вариантов реализации алгоритма фрактального сжатия и методов его ускорения.
2. Автоматизированная система фрактального сжатия изображения, позволяющая выбрать сжимаемого изображение, вариант реализации алгоритма и метод ускорения и предоставляющая информацию о времени декомпрессии и качестве декодированного изображения.

**Апробация работы**

Основные положения и результаты работы представлялись, докладывались и обсуждались на международной конференции «Перспективные информационные технологии - 2018» (Самара, 2018), всероссийской заочной научно-технической конференции «Инновационные технологии в проектировании и производстве» (Пенза, 2018).

**Публикации**

Работы по теме выпускной работы опубликованы в 2 сборниках трудов конференции, индексируемых в РИНЦ, и 1 статья опубликована в сборнике материалов студенческой международной научно-практической конференции.

сОдержание работы

**Во введении** приведены основные определения и понятия, показана актуальность темы выпускной квалификационной работы.

**В первой главе** приведена постановка задачи, математическое обоснование фрактального сжатия изображений и приведены формулы, используемые алгоритмом для преобразования блоков изображений, приведены классический алгоритм сжатия и алгоритм декомпрессии.

Фрактальное сжатие осуществляется за счет использования так называемых *сжимающих* отображений, для которых справедливо следующее утверждение:

Если к какому-то изображению *F0* мы начнём многократно применять отображение *W* таким образом, что

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 *F0*:

Это конечное изображение *F* называют *аттрактором*, или *неподвижной точкой отображения W*.

Как будет описано далее, процесс декодирования изображения будет осуществляться именно путем многократно применения отображения *W* к базовому изображению. Именно наличие аттрактора у отображения *W* позволяет при декомпрессии в качестве базового использовать любой изображение.

Отображение *W* является объединением отображений *wi*, применяемых к отдельным блокам изображения. Такими отображениями являются аффинные и яркостные преобразования.

Аффинное преобразование − отображение плоскости или пространства в себя, при котором параллельные прямые переходят в параллельные прямые, пересекающиеся − в пересекающиеся, скрещивающиеся − в скрещивающиеся [4].

В общем виде аффинные преобразования на плоскости описываются следующими формулами:

где A, B, C, D, E, F – некие константы, (x,y) - координаты точки на плоскости до преобразования, (X,Y) – координаты точки на плоскости после преобразования.

Аффинным преобразованием, применяемым мной в данной работе является поворот на градусов. Такое преобразование выражается формулами:

Каждый из приведенных далее алгоритмов фрактального сжатия включает применение 8 аффинных преобразований к доменными блоками.

Примеры этих преобразований можно увидеть в таблице 1.

Таблица 1 – Аффинные преобразования доменного блока

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Название | Формулы | Пример | № | Название | Формулы | Пример |
| 1 | Поворот на 0о |  |  | 5 | Отражение относительно оси Y |  |  |
| 2 | Поворот на 90о |  |  | 6 | Отражение относительно оси Х |  |  |
| 3 | Поворот на 180о |  |  | 7 | Поворот на 90о и отражение относительно оси Y |  |  |
| 4 | Поворот на 270о |  |  | 8 | Поворот на 90о и отражение относительно оси Х |  |  |

При поиске подходящего доменного блока для достижения максимального соответствия блоков часто нам необходимо не только поворачивать доменный блок, но также и изменять его яркость и контрастность.

Для таких изменений блока используют *контрастность s* и *яркость o* – яркостные характеристики преобразования доменного блока к ранговому блоку.

Оптимальные контрастность и яркость минимизируют выражение

в котором и это соответственно значения цветовых компонент пискелей ранговой и доменной областей, а – длина стороны рангового и доменного блоков. Сами яркость и контрастность вычисляются по формулам

где

и – размер стороны рангового (доменного) блока.

По своей сути, фрактальное сжатие (или фрактальная компрессия) − это процесс поиска самоподобных областей изображения и определения для них параметров аффинных и яркостых преобразований.

Для реализации алгоритма компрессии необходимо выполнить следующие шаги:

1) исходное изображение разбивается на подобласти, которые представляют из себя квадраты, называемые *ранговыми блоками*. Ранговые блоки пересекаться не могут;

2) на исходном изображении выделяются *домены* – совокупности четырех ранговых блоков. Домены могут пересекаться. Все ранговые блоки и домены – это квадраты со сторонами, параллельными изображению;

3) для каждого рангового блока производится попытка найти на изображении домен, такой чтобы этот домен можно было преобразовать в ранговый блок при помощи аффинных преобразований;

4) перевод домена в ранговый блок производится с помощью поворота домена на 0°, 90°, 180°, 270° и с помощью вертикального и горизонтального зеркальных преобразований;

5) при переводе доменной области в ранговую, ее линейный размер уменьшается в 2 раза;

6) изменение яркости производится кратно некоторому коэффициенту;

7) совпадение преобразованного домена с ранговым блоком может производиться при помощи среднеквадратичного отклонения:

где – точка в домене; – точка в блоке; – пороговое значение «похожести»;

8) если же для некоторого рангового блока не было найдено ни одного удовлетворяющего среднеквадратичному отклонению домена, то ранговый блок разбивается на 4 подобласти, и для каждой из них ищутся домены.

Для осуществления декомпрессии необходимо задать базовое изображение, k – количество итераций декодирования, фрактальный код.

Алгоритм фрактальной декомпрессии включает следующие шаги:

1. задаем значение исходных данных;
2. из фрактального кода выделяем параметры преобразований для рангового блока;
3. на базовом изображении выделяем соответствующий параметрам доменный блок;
4. уменьшаем его;
5. применяем аффинное преобразование;
6. применяем яркостное преобразование;
7. копируем преобразованный доменный блок на место текущего рангового блока в базовом изображении;
8. если декодировали все ранговые блоки, то переходим к п.9, иначе – к п.2;
9. повторяем данный выполнение данного алгоритма k раз.

**Во второй главе** описаны варианты реализации алгоритма фрактального сжатия и методы его ускорения.

Общий алгоритм фрактального сжатия включает следующие шаги:

1. исходное изображение разбивается на ранговые блоки;
2. для каждого рангового блока:
3. ищется доменный блок, соответствующий ранговому блоку;
4. сохраняются параметры подошедшего доменного блока;
5. полученные соответствия между ранговыми и доменными блоками (фрактальный код) сохраняются для дальнейшего использования.

Степень схожести рангового и доменного блока вычисляется как среднеквадратическое отклонение (СКО):

где – длина стороны рангового и доменного блоков.

С учетом яркостных преобразований, условие (1) принимает следующий вид:

Подходящий доменный блок может выбираться несколькими способами:

1. До первого найденного доменного блока, удовлетворяющего условию (1). Если ни один доменный блок не удовлетворяет условию (1):
   1. Берем доменный блок с минимальным СКО (алгоритм А1);
   2. Разбиваем ранговый блок на 4 блока и для каждого из них ищем подходящий доменный блок (алгоритм А2).
2. Доменный блок с минимальным СКО, который ищется с помощью полного перебора всех возможных доменных блоков (алгоритм Б).

Для ускорения процесса сжатия в работе были рассмотрены следующие подходы:

1. предварительная классификация блоков;
2. метод эталонного блока.

В случае использования предварительной классификации блоков [5], прежде чем приступить к выполнению компрессии, каждому ранговому и доменному блоку, выделяемому на изображении, присваивается определенный класс; в дальнейшем, поиск подходящего доменного блока осуществляется только среди доменных блоков, имеющих тот же класс, что и ранговый блок.

В данной работе для классификации блоков использовались следующие подходы:

1. классификация по значению центра массы блока;
2. классификация по разнице граничных яркостных значений блока.

Центр масс блока рассчитывается следующей формуле:

где – количество пикселей в блоке, – яркость пикселей блока.

Разница граничных яркостных значений блока вычисляется по формуле:

Метод эталонного блока основан на том, что доменные блоки сравниваются не с каждым ранговым блоком, а единожды – с эталонным, и на основе СКО между ними делается вывод о степени «похожести» доменного блока и рангового. Схема алгоритма данного метода приведена на рисунке 1.

Да

Нет

Расчет domensSKOs[j][s]

(СКО между эталонным блоком и аффинным преобразованием s доменного блока)

etalon – эталонный блок

Исходное

изображение

etalon

Расчет N (количество ранговых блоков)

Расчет M (количество доменных блоков)

i

s

s

s *>* 8

i

i *>* M

i

i

i *>* N

Выделение рангового блока i

Расчет rangSKO

(СКО между эталонным блоком и ранговым блоком)

min = Min(|domensSKOs[j][s] - rangSKO |),

i = 1..M,

s = 1..8

min < ε

Сохранение параметров преобразования доменного блока, соответствующего min

Выполнение алгоритма А1 (А2 или Б) для рангового блока i

s = s + 1

*i*  = *i* + 1

*i*  = *i* + 1

Конец

Начало

Рисунок 1 − Схема метода эталонного блока

**В третьей главе** приведен логический проект разработанной системы, описанный с использованием нотации UML, описан выбор языка программирования и средств разработки, а так же приведена информация о программной реализации разработанной системы.

Методология UML является мощным средством проектирования, устранившим недостатки более ранних методологий, в том числе и основной недостаток SADT-методологии – отсутствие объектно-ориентированного представления моделей сложных систем. Основная задача, которая стояла при создании проекта – отобразить функциональность системы.

Разработка проекта системы выполнена в бесплатной среде UML-моделирования StarUML 5.0. На рисунке 2 приведена диаграмма вариантов использования разработанной системы.

Для разработки программной системы была выбрана операционная система Microsoft Windows 7, среда разработки Microsoft Visual Studio 2015 и язык C#. Интерфейс разработанной системы представлен на рисунке 3.



Рисунок 2 – Диаграмма вариантов использования системы

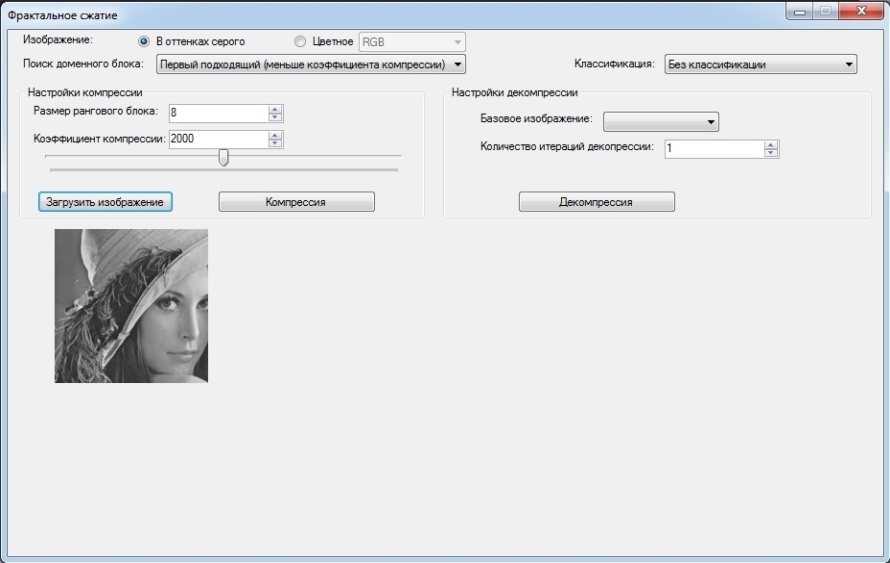


Рисунок 3 – Интерфейс разработанной системы

**В четвертой главе** исследована зависимость времени выполнения фрактального сжатия изображения и качества декодируемого изображения, от выбранного варианта реализации, используемого метода ускорения и параметров сжатия.

Время сжатия и декомпрессии замеряется программной системой. Качество декомпрессии оценивается как СКО декодируемого изображения от исходного. В таблицах, представленных ниже R – размер рангового блока, ЦМ – классификация центром масс, РГЗ – классификация разницей граничных значений.

Для оценки качества декодируемого изображения используется SSIM – величина, изменяющаяся от 0 до 1. Чем SSIM ближе к 1, тем более декодируемое изображение походит на сжимаемое.

В таблице 2 представлены результаты исследования зависимости параметров компрессии и декомпрессии от выбранного подхода, параметров сжатия и содержимого для изображений в оттенках серого, размером 160×160 пикселей.

Из таблицы 2 можно увидеть, что содержимое изображения влияет на коэффициенты компрессии (такие как размер рангового блока, или ε). Для каждого типа изображение использование классификаций (центром масс, или разницей граничных значений) приводит к уменьшению времени выполнения алгоритма.

Таблица 2 – Сводная таблица результатов исследований

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм поиска доменного блока | Классификация | Тип изображения | Наиболее эффективный метод | Размер рангового блока | ε | tкомп, сек | tдекомп, сек | Степень сжатия | SSIM |
| А1  (первый подходящий доменный блок без разбиения) | - | Портрет |  | 4 | 150 | 9,21 | 1,89 | 4,534096 | 0,980774 |
| Мало деталей |  | 4 | 300 | 2,12 | 2,50 | 4,37765 | 0,989341 |
| Много деталей |  | 4 | 50 | 131,24 | 2,15 | 4,391942 | 0,980996 |
| Текст |  | 4 | 200 | 148,96 | 1,93 | 4,524096 | 0,928961 |
| Центр масс | Портрет |  | 4 | 150 | 4,06 | 2,01 | 4,497911 | 0,976509 |
| Мало деталей |  | 4 | 300 | 1,04 | 2,09 | 4,392911 | 0,990205 |
| Много деталей |  | 4 | 50 | 69,36 | 2,41 | 4,391813 | 0,979381 |
| Текст |  | 4 | 200 | 64,17 | 2,44 | 4,551515 | 0,904461 |
| Разница граничных значений | Портрет | + | 4 | 150 | 2,95 | 2,07 | 4,525048 | 0,977291 |
| Мало деталей | + | 4 | 300 | 0,94 | 2,52 | 4,381127 | 0,989054 |
| Много деталей |  | 4 | 50 | 56,09 | 2,15 | 4,392115 | 0,988772 |
| Текст |  | 4 | 200 | 43,26 | 2,35 | 4,547281 | 0,912796 |
| А2  (первый подходящий доменный блок с разбиением) | - | Портрет |  | 16 | 5 | 45,08 | 1,83 | 8,795074 | 0,979343 |
| Мало деталей |  | 16 | 5 | 27,50 | 2,01 | 21,82031 | 0,988581 |
| Много деталей |  | 16 | 5 | 109,67 | 1,76 | 4,551515 | 0,975073 |
| Текст |  | 16 | 5 | 120,94 | 1,72 | 5,364286 | 0,915178 |
| Центр масс | Портрет |  | 16 | 5 | 16,84 | 1,91 | 8,427301 | 0,979660 |
| Мало деталей |  | 16 | 5 | 8,84 | 2,38 | 19,94933 | 0,990315 |
| Много деталей |  | 16 | 5 | 43,27 | 1,81 | 4,549565 | 0,978722 |
| Текст |  | 16 | 5 | 71,21 | 1,80 | 5,380271 | 0,901211 |
| Разница граничных значений | Портрет |  | 16 | 5 | 15,86 | 2,45 | 8,45464 | 0,980494 |
| Мало деталей |  | 16 | 5 | 8,90 | 2,17 | 20,19 | 0,992567 |
| Много деталей |  | 16 | 5 | 37,42 | 2,13 | 4,544096 | 0,975133 |
| Текст |  | 16 | 5 | 52,24 | 1,91 | 5,215278 | 0,916037 |
| Б  (доменный блок с минимальным СКО) | - | Портрет |  | 8 | - | 35,72 | 2,15 | 17,67223 | 0,972241 |
| Мало деталей |  | 8 | - | 46,89 | 2,49 | 17,51587 | 0,992094 |
| Много деталей |  | 4 | - | 225,11 | 2,31 | 4,417647 | 0,988125 |
| Текст |  | 4 | - | 136,17 | 2,10 | 4,443787 | 0,906616 |
| Центр масс | Портрет |  | 8 | - | 15,75 | 2,13 | 17,71316 | 0,967713 |
| Мало деталей |  | 8 | - | 14,34 | 2,56 | 17,60504 | 0,988661 |
| Много деталей |  | 4 | - | 57,06 | 2,05 | 4,391813 | 0,977484 |
| Текст |  | 4 | - | 96,77 | 2,34 | 16,8 | 0,901921 |
| Разница граничных значений | Портрет |  | 8 | - | 13,43 | 2,21 | 17,71525 | 0,971010 |
| Мало деталей |  | 8 | - | 30,45 | 2,32 | 17,55206 | 0,989017 |
| Много деталей | + | 4 | - | 31,31 | 2,48 | 4,402117 | 0,972327 |
| Текст | + | 4 | - | 37,19 | 2,21 | 4,470238 | 0,901411 |
| Метод эталонного блока | - | Портрет |  | 4 | 300 | 35,86 | 2,42 | 4,814103 | 0,977680 |
| - | Мало деталей |  | 4 | 300 | 35,03 | 2,29 | 4,690897 | 0,991809 |
| - | Много деталей |  | 4 | 100 | 103,77 | 2,35 | 4,366279 | 0,972875 |
| - | Текст |  | 4 | 200 | 119,01 | 2,24 | 4,524096 | 0,925923 |

Также исследования проводились над набором изображений в оттенках серого, размером, превышающим 160×160 пикселей и над изображениями не квадратной формы.

По данным таблиц 3 и 4 можно видеть, что в случае изменения пропорций изображения, или его размера, тенденция уменьшения времени выполнения алгоритмов А1, А2 и Б, при использовании классификаций сохраняется. При этом эффективность классификаций зависит от содержимого изображения: нельзя однозначно выявить преобладание классификации центром масс над классификацией разницей граничных значений (или наоборот). Наименьшее время сжатия во всех трех случаях дает выбор первого подходящего доменного блока без разбиения (алгоритм А1). Метод эталонного блока дает преимущество по времени, относительно примененного без классификаций алгоритма А2 (который можно назвать классической реализацией выбора доменного блока для алгоритма фрактального сжатия).

Таблица 3 *–* Зависимость параметров компрессии и декомпрессии от выбранного подхода и параметров сжатия для изображения размером 120×200 пикселей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм выбора доменного блока | Метод классификации | R | ε | tкомп, сек | tдекомп, сек | Степень сжатия | SSIM |
| А1 | - | 4 | 150 | 7,71 | 1,81 | 4,36646 | 0,99042 |
| ЦМ | 4 | 150 | 5,81 | 1,97 | 4,39375 | 0,98321 |
| РГЗ | 4 | 150 | 2,59 | 1,72 | 4,39375 | 0,99063 |
| А2 | - | 16 | 5 | 142,17 | 1,68 | 6,509259 | 0,98436 |
| ЦМ | 16 | 5 | 25,87 | 1,61 | 6,390909 | 0,96865 |
| РГЗ | 16 | 5 | 24,69 | 1,62 | 6,276786 | 0,97109 |
| Б | - | 8 | - | 31,34 | 1,62 | 17,575 | 0,98054 |
| ЦМ | 8 | - | 8,67 | 1,61 | 17,575 | 0,97648 |
| РГЗ | 8 | - | 5,28 | 1,58 | 17,575 | 0,97597 |
| Метод эталонного блока | - | 4 | 300 | 47,42 | 1,61 | 4,535484 | 0,98702 |

Таблица 4 *–* Зависимость параметров компрессии и декомпрессии от выбранного подхода и параметров сжатия для изображения размером 304×304 пикселей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм выбора доменного блока | Метод классификации | R | ε | tкомп, сек | tдекомп, сек | Степень сжатия | SSIM |
| А1 | - | 4 | 150 | 68,42 | 7,01 | 4,383117 | 0,98257 |
| ЦМ | 4 | 150 | 33,32 | 8,56 | 4,347826 | 0,98936 |
| РГЗ | 4 | 150 | 18,02 | 6,38 | 4,376013 | 0,98937 |
| А2 | - | 16 | 5 | 506,02 | 7,64 | 7,541899 | 0,98001 |
| ЦМ | 16 | 5 | 189,61 | 6,26 | 7,2 | 0,98661 |
| РГЗ | 16 | 5 | 152,18 | 6,55 | 7,238606 | 0,98691 |
| Б | - | 8 | - | 361,05 | 7,48 | 17,53247 | 0,98222 |
| ЦМ | 8 | - | 127,12 | 8,14 | 17,53247 | 0,97945 |
| РГЗ | 8 | - | 109,15 | 7,71 | 17,41935 | 0,97941 |
| Метод эталонного блока | - | 4 | 300 | 399,98 | 7,53 | 4,728546 | 0,98734 |

В таблице 5 приведены результаты исследования зависимости времени сжатия от типа изображения (цветное или в оттенках серого) и используемого алгоритма.

Таблица 5 – Зависимость параметров компрессии и декомпрессии от используемого алгоритма и типа изображения

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм выбора доменного блока | Тип изображения | R | ε | tкомп, сек | tдекомп, сек | Степень сжатия | SSIM |
| А1 | В оттенках серого | 4 | 150 | 9,21 | 1,89 | 4,534096 | 0,980774 |
| RGB | 4 | 300 | 18,99 | 7,78 | 1,461089 | 0,979791 |
| IYQ | 4 | 200 | 96,38 | 12,76 | 1,452081 | 0,912761 |
| А2 | В оттенках серого | 16 | 5 | 45,08 | 1,83 | 8,795074 | 0,979343 |
| RGB | 16 | 10 | 154,74 | 8,51 | 2,282675 | 0,972539 |
| IYQ | 16 | 10 | 220,31 | 10,42 | 4,607362 | 0,903127 |
| Б | В оттенках серого | 8 | - | 35,72 | 2,15 | 17,67223 | 0,972241 |
| RGB | 8 | - | 55,91 | 6,61 | 5,867188 | 0,972118 |
| IYQ | 8 | - | 200,37 | 8,47 | 5,437176 | 0,913239 |

Данные из таблицы 5 указывают на то, что время сжатия цветного изображения ожидаемо больше времени сжатия изображения в оттенках серого, однако время обработки возрастает не в 3 раза, как этого можно было бы ожидать, а меньше. Это достигается за счет того, что для всех трех цветовых компонент рангового блока запускается единый цикл перебора доменных блоков, а не отдельный цикл для каждой компоненты. и для изображения в оттенках серого и для цветных изображений обеих моделей, при условии одинакового содержимого изображения, наилучший результат по времени предоставляет использование алгоритма А1, а наихудший – А2. Также можно видеть, что для цветного изображения увеличивается время декомпрессии.

По результатам проведенных исследований можно сделать вывод, что для всех трех реализаций алгоритма фрактального сжатия (первый подходящий доменный блок без разбиения, первый подходящий доменный блок с разбиением и доменный блок с минимальным СКО), использование предварительной классификации блоков позволяет уменьшить время сжатия. При этом, эффективность классификаций зависит от содержимого изображения: нельзя однозначно выявить преобладание классификации центром масс над классификацией разницей граничных значений (или наоборот). Метод эталонного блока дает заметное ускорение для изображений типа «портрет», для остальных типов – практически не дает преимущества перед алгоритмом А2, который можно считать классической реализацией фрактального сжатия.

**В заключении** сформулированы основные выводы, перечислены полученные в работе результаты.

основные выводы и результаты

1. Разработана автоматизированная система фрактального сжатия изображения, позволяющая выбрать сжимаемого изображение, вариант реализации алгоритма и метод ускорения и предоставляющая информацию о времени декомпрессии и качестве декодированного изображения
2. Проведены исследования зависимости времени фрактального сжатия и декомпрессии от параметров компрессии, содержимого изображения и его типа.
3. Для исследованных наборов изображений определены наиболее эффективные (с точки зрения времени сжатия и качества декодируемого изображения) параметры компрессии, используемый вариант реализации и метод ускорения.

Основные публикации по теме диссертации

**Работы, опубликованные в ведущих рецензируемых научных журналах и изданиях, определенных Высшей аттестационной комиссией:**

* 1. ? так как нет – пропустить этот пункт вообще?

**Тезисы докладов:**

* 1. Ссылка на надежность и качество
  2. Сахибназарова, В.Б. Исследование алгоритмов фрактального сжатия изображений [Текст]/В.Б. Сахибназарова, М.А. Кудрина//Перспективные информационные технологии (ПИТ 2017): труды Международной научно-технической конференции / под ред. С.А. Прохорова. – Самара: Издательство Самарского научного центра РАН, 2018. – С. 128-132
  3. Сахибназарова, В.Б. Применение предварительной классификации данных для повышения скорости фрактального сжатия изображений [Текст]/В.Б. Сахибназарова //Научное сообщество студентов XXI столетия: сб. науч. тр. по мат-лам Междунар. заоч. науч-практ. конф. апрель 2018 г.: в 3 частях. Часть 2/Тамбов: Изд-во АНС «СибАК», 2018. – 272 с.

**Патент:**

* 1. Свидетельство о государственной регистрации программы для ЭВМ № 2014613802 ? у нас же ничего такого тоже не делалось