

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**РАЗРАБОТКА И РЕАЛИЗАЦИЯ ОПТИМАЛЬНОГО АЛГОРИТМА
ФРАКТАЛЬНОГО СЖАТИЯ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ
АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ**

студента 2 курса 273 группы
направления 01.04.02 – Прикладная математика и информатика факультета
компьютерных наук и информационных технологий
Батталова Аббаса Илдаровича

Научный руководитель:

Доцент

Е. В. Кудрина

подпись, дата

Зав. кафедрой:

Доцент, к.ф.-м.н.

М.В. Огнева

подпись, дата

2018

ВВЕДЕНИЕ

Актуальность темы. В связи с бурным развитием вычислительной техники и интернета, тесно связанных с мультимедийными приложениями, особую популярность приобрели цифровые изображения. В настоящее время количество изображений во всемирной паутине увеличивается в разы, за одни сутки на такие ресурсные сервисы как Facebook и Instagram загружается более 200 миллионов фотографий. Для хранения такого количества изображений в цифровой форме требуются значительные объёмы электронной памяти. Сохранение изображений в меньших объёмах памяти приводит к сокращению расходов на хранение информации и уменьшению времени передачи изображений по современным каналам связи, поэтому в наши дни проблема сжатия изображений приобрела особую актуальность.

Способы сжатия изображений можно разделить на две категории: сжатие без потерь не позволяет добиться высоких коэффициентов сжатия (в среднем не более чем в 3 раза, в некоторых случаях размер сжатого изображения может превысить размер исходного за счет избыточности информации), но восстановленное после сжатия изображение будет до пиксела соответствовать оригиналу; сжатие с потерями позволяет добиться высоких коэффициентов сжатия (до 50-ти раз) при несущественном ухудшении качества изображения, но реализуется только при значительных временных затратах.

Из всех современных алгоритмов сжатия, фрактальное сжатие потенциально способно обеспечить наилучшее соотношение степени сжатия и качества. Однако, фрактальное сжатие не лишено недостатков, основной из которых – это значительные временные затраты при его реализации на ЭВМ, обоснование этого факта приведено в данной работе. С развитием вычислительной техники и информационных технологий, мощность машин возрастает, что позволяет оптимизировать затраты времени на процесс сжатия. В связи с этим разработка алгоритмов быстрого фрактального сжатия представляет собой актуальную научную задачу, решение которой позволило

бы построить на его основе специальные форматы сжатия для применения в узких областях, например, в здравоохранении, где предъявляются особые требования к качеству изображений.

Следует отметить, что магистратская работа является логическим продолжением бакалаврской работы, в которой был изучен и исследован алгоритм фрактального сжатия для изображений с применением некоторых способов оптимизации [1]. Данная работа посвящена дополнительному исследованию фрактального сжатия цветных изображений с **целью** разработки улучшенного алгоритма фрактального сжатия цветных изображений и созданием приложения пригодного для практического применения.

Поставленная цель определила следующие **задачи**:

1. Исследовать алгоритмы фрактального сжатия цветных изображений и осуществить выбор наиболее перспективных направлений для разработки алгоритма быстрого фрактального сжатия.

2. Реализовать способы ускорения и оптимизации фрактального сжатия цветных изображений.

3. Разработать программное обеспечение, позволяющее пользователю сжимать и восстанавливать цветные изображения формата bmp с использованием фрактального алгоритма сжатия в режиме реального времени.

Методологические основы фрактального сжатия изображения представлены в трудах М.П. Шарабайко, А.Н. Осокина [5], С. Уэлстида [6], А. С. Сибирякова [9], А.С. Касаткин, Е.А. Шанцын, В.В. Мусихин [10], С.В. Илюшин [12], А.Н. Зыков [13], С.С. Кулбаев [15].

Практическая значимость бакалаврской работы. Все основные научные результаты, программные реализации и выводы, изложенные в дипломной работе, получены автором самостоятельно. Разработанное приложение ColorFractus2.0 является, программным обеспечением позволяющим пользователю использовать его для сжатия и восстановления одного или нескольких изображений в реальном времени и управлять процессами устанавливая значения параметров сжатия/восстановления.

Структура и объём работы. Дипломная работа состоит из введения, четырех разделов, заключения, списка использованных источников и приложений, одно из которых представлено на цифровом носителе. Общий объем работы – 66 страниц, из них 55 страниц – основное содержание, включая 16 рисунков и 21 таблицу, цифровой носитель в качестве приложения, список использованных источников – 31 наименование.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы фрактального сжатия изображений»

Рассмотрено представление изображения в памяти ЭВМ системой RGB, в которой цвет представлен значениями интенсивности красного (R), зеленого (G) и синего (B) компонента [2]. Описана цель и основные общие требования к архиваторам изображений [3]. Далее говорится о фундаменте фрактального сжатия: теоретическая основа [4] и общая идея фрактального сжатия [5-10]. Описана подготовка изображения перед сжатием: разбиение на ранговые и доменные блоки, сжатия доменных блоков до размеров рангового для дальнейшего сравнения их с помощью «дистанции», представляющей собой функцию среднеквадратического отклонения. Применение данной идеи для сжатия цветных изображений. Представлены блок-схемы базовых алгоритмов сжатия и восстановления.

Далее рассматриваются способы усовершенствования базового алгоритма сжатия/восстановления изображений.

Один из способов оптимизации заключается в уменьшении временной сложности алгоритма, основывающегося на переборе блоков. Сокращение количества перебираемых блоков возможно за счет использования погрешности (числовой величины, определяющей степень «похожести» рангового и доменного блоков). Был проведен ряд тестов и выявлены наиболее оптимальные значения для используемых размеров рангового блока.

Другой способ оптимизации – это классифицирование. Один из наиболее эффективных путей сокращения времени сжатия является классификация

блоков по одному или нескольким признакам [12-13]. В работе представлены примеры блоков из одной категории, а также принцип, по которому блоки распределяются в соответствующие категории.

Кэширование – один из способов, с помощью которых можно увеличить эффективность приложения, обеспечивая использование соответствующего уровня кэширования повторно используемых данных, для создания которого требуются большие средства [14-15]. В работе представлено решение, в котором каждый блок получает свой ключ, используемой для кэширования уже полученных результатов сравнения.

Четвертый способ — это его распараллеливание [16-17]. Данная технология использовалась: при разложении изображения на три составляющих, при независимом поиске схожих доменных блоков и для восстановления изображения. После реализации всех выше перечисленных способов оптимизации в работе представлены результаты профилировщика для демонстрации положительных результатов. После описан следующий способ усовершенствования — это изменения метода записи структуры данных в конечный файл, в основу принципе легла идея: первоочередная запись всех, самых распространённых при сохранении, доменных блоков и последующая запись ранговых блоков со ссылками на записанные доменные блоки.

Второй раздел «Обзор инструментальных средств и технологий, используемых при реализации алгоритма фрактального сжатия цветных изображений» посвящен инструментам и технологиями, которые непосредственно использовались в реализации поставленных.

Третий раздел «Программная реализация фрактального сжатия изображения» посвящен архитектуре приложения, а также визуальному представлению интерфейса пользователя.

При разработке приложения за основу взята многоуровневая архитектура, в которой выделены три уровня: уровень представления, уровень логики, уровень данных. В архитектуре выделены функциональные слои. Они слабо связаны, и между ними осуществляется явный обмен данными [18]. Для

программной реализации был использован язык С# – язык разработки приложений для платформы Microsoft .NET Framework. Для создания интерфейса использовался Windows Forms, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework а также проанализированы лучшие практики по заданию интерфейса [19-23].

Представлена демонстрация работы приложения, изложены этапы использования приложения ColorFractus2.0, шаги проиллюстрированы в виде скриншотов рабочего приложения.

Четвертый раздел «Экспериментальное исследование и оценка эффективности программной реализации фрактального сжатия/восстановления цветного изображения» посвящен экспериментальной части, оценки эффективности приложения ColorFractus будет происходить принципов сравнения с другими современными и распространёнными архиваторами. Такие как WinRar [29-30] и Jpeg [11].

При сжатии программой ColorFractus2.0 в скобках указывается размер рангового блока и выделенная погрешность. Программа WinRar использовалась в двух режимах: обычный и быстрая скорость. JPEG использовался в двух режимах качества 100 и 80.

Далее описаны колонки таблиц-тестов, каждая из которых отсортирована по последней колонке по возрастанию значения, в результате первая колонка «тип сжатия» получилась отсортирована от более приоритетного в использовании типа сжатия применимому к этому изображению к менее приоритетному.

Полученные результаты сжатия и восстановления изображений были занесены в соответствующие таблицы. Основным заключением экспериментальной части является вывод, в котором говорится о основной проблеме построения алгоритма фрактального сжатия. О зависимости размера рангового блока, время выполнения и качестве восстановленного изображения. Показаны как положительные, так и отрицательные стороны фрактального сжатия.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы магистратуры были решены все поставленные задачи. Результатом работы стало программное обеспечение ColorFractus 2.0, позволяющее пользователю сжимать и восстанавливать цветные изображения формата bmp с использованием фрактального алгоритма сжатия в режиме реального времени.

ColorFractus2.0 имеет не плохой коэффициент сжатия, но недостаточное время выполнения и качество восстановленного изображения, для того чтобы конкурировать с алгоритмом JPEG. WinRar, как говорилось ранее, является архиватором сжатия без потерь, его участие в эксперименте было аргументированно лишь его популярностью.

По тематике магистерской работы были подготовлены и опубликованы следующие работы:

1. Батгалов А.И. Фрактальное сжатие цветных изображений//Научные исследования студентов Саратовского государственного университета: материалы итоговой студенческой научной конференции. – Саратов: Изд-во Сарат.ун-та, 2016. – с.31-32.
2. Батгалов А.И., Кудрина Е.В. Разработка и реализация быстрого алгоритма сжатия цветных изображений//Информационные технологии в образовании: материалы VIII Международ. Научно-практ. конф. – Саратов: ООО «Издательство центр «Наука»», 2016. –с. 21-25.
3. Батгалов А.И., Батгалова Н.И. Использование средства профилирования Visual Studio для оптимизации программного обеспечения // Научное обозрение. Технические науки. – 2018. – № 1. – С. 5-9.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Батгалов А.И. [Электронный ресурс] // Фрактальное сжатие цветных изображений. Автореферат выпускной квалификационной работы бакалавра. СГУ. Саратов, 2016. 10 с. URL: http://library.sgu.ru/cgi-bin/irbis64r_17/cgiirbis_64.exe?LNG=&C21COM=2&I21DBN=VKR&P21DBN

=VKR&Z21ID=&IMAGE_FILE_DOWNLOAD=1&IMAGE_FILE_OCC=1&IMAGE_FILE_MFN=2368 (дата обращения: 05.2018)

2. RGB [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/RGB> (дата обращения: 04.2018).
3. Всё о сжатии данных, изображений и видео «Методы сжатия данных: Сжатие изображений» [Электронный ресурс]. URL: http://www.compression.ru/book/part2/part2__1.htm (дата обращения: 04.2018).
4. Фрактал [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Фрактал> (дата обращения: 04.2018).
5. М.П. Шарабайко, А.Н. Осокин, Статья «Быстродействующий алгоритм фрактального сжатия изображений» [Электронный ресурс]. URL: <http://fic.bos.ru/articles/> (дата обращения: 04.2018).
6. С.Уэлстид. Фракталы и вейвлеты для сжатия изображения. Учебное пособие. – М.; Издательство Триумф, 2003. – 320с
7. Статья «Фрактальное сжатие изображений» [Электронный ресурс]. URL: <http://www.compression.ru/arctest/descript/fract-comp.htm> (дата обращения: 04.2018).
8. Занятия «Алгоритмические основы растровой графики: Информация» [Электронный ресурс]. URL: <http://www.intuit.ru/studies/courses/993/163/info> (дата обращения: 04.2018).
9. Сибиряков А. С. «Фрактальное сжатие изображений» [Электронный ресурс]. URL: <http://cs.usu.edu.ru/study/fractal/#11> (дата обращения: 04.2018).
10. А.С. Касаткин, Е.А. Шанцын, В.В. Мусихин «Фрактальное кодирование цветных изображений и полутоновых видеопоследовательностей». Тематический выпуск № 2-2006 год.
11. Всё о сжатии данных, изображений и видео [Электронный ресурс]. URL: http://www.compression.ru/book/part2/part2__3.htm (дата обращения: 04.2018).
12. С.В. Илюшин «Ускорение фрактального сжатия изображений путем классификации блоков по полярному углу их центров масс»// Т-Comm: Телекоммуникации и транспорт. 2011. Т. 5. № 4. С. 43-47.

13. А.Н. Зыков «Применение оператора локальных бинарных шаблонов в задаче фрактального кодирования изображений»// Arctic Environmental Research. 2015. № 4. С. 138-145.
14. Кэширование в основе эффективности приложений [Электронный ресурс]. URL: [https://msdn.microsoft.com/ru-ru/library/ff953179\(v=pandp.50\).aspx](https://msdn.microsoft.com/ru-ru/library/ff953179(v=pandp.50).aspx) (дата обращения: 05.2018)
15. С.С. Кулбаев «Реализация параллельного алгоритма сжатия изображений на основе фракталов»// Электронные средства и системы управления. 2011. № 1. С. 141-145.
16. System.Threading [Электронный ресурс]. URL: [https://msdn.microsoft.com/ru-ru/library/system.threading\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.threading(v=vs.110).aspx) (дата обращения: 05.2018).
17. Оператор lock [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/lock-statement> (дата обращения: 13.10.2017).
18. Сомасегару С. Руководство Microsoft по проектированию архитектуры приложений/ С. Сомасегару, С. Гатри. – Корпорация Майкрософт. – 2010. – 462с.
19. Windows Forms [Электронный ресурс]// Портал Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms> (дата обращения: 05.2018).
20. Сулова Е. Графический интерфейс пользователя [Электронный ресурс]. – URL: http://cultlook.org/es_gui (дата обращения: 15.03.2018).
21. User Interface Design – Some Of The Best Practices To Keep In Mind [Электронный ресурс]. URL: <https://visualhierarchy.co/blog/user-interface-design-best-practices/> (дата обращения: 05.2018).
22. Купер А. Об интерфейсе: Основы проектирования взаимодействия / А. Купер, Р. М. Рейманн, Д. Кронин. – Москва: Изд-во «Символ-Плюс». – 2009. – 688 с.

23. Фетзгеральд А. The Guide to Design Consistency: Best Practices for UI and UX Designers [Электронный ресурс]. URL: http://cultlook.org/es_gui (дата обращения: 0.5.2018).
24. Application.DoEvents [Электронный ресурс] // Портал Microsoft. – URL: [https://msdn.microsoft.com/library/system.windows.forms.application.doevents\(v=vs.90\).aspx](https://msdn.microsoft.com/library/system.windows.forms.application.doevents(v=vs.90).aspx) (дата обращения: 05.2018).
25. Практическое руководство. Осуществление потокобезопасных вызовов элементов управления Windows Forms. [Электронный ресурс] // Портал Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/how-to-make-thread-safe-calls-to-windows-forms-controls> (дата обращения: 05.2018).
26. Многопоточность в элементах управления Windows Forms [Электронный ресурс] // Портал Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/multithreading-in-windows-forms-controls> (дата обращения: 05.2018)
27. Класс Task [Электронный ресурс] // Портал Microsoft. – URL: [https://msdn.microsoft.com/ru-ru/library/system.threading.tasks.task\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.threading.tasks.task(v=vs.110).aspx) (дата обращения: 15.03.2018)
28. Вагнер Б. Асинхронное программирование с использованием ключевых слов Async и Await (C#) [Электронный ресурс] // Портал Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/async/> (дата обращения: 05.2018).
29. RAR [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/RAR> (дата обращения: 29.05.2016).
30. Сравнение 64-битных архиваторов [Электронный ресурс]. URL: <http://compress.ru/article.aspx?id=23664> (дата обращения: 29.05.2016)
31. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2006. 1072 с