Обзор алгоритмов, используемых при сжатии и восстановлении изображений, сравнение их качественных характеристик

Батурин Игорь
СПбГЭТУ (ЛЭТИ)
РФ, г. Санкт-Петербург
Email: Baturin29082000@mail.ru

Аннотация **- в статье описаны следующие алгоритмы сжатия и восстановления изображений: алгоритм Хаффмана [1][7], групповое сжатие LRE (Run-Length Encoding) [2][6][7], метод LZW (Lempel-Ziv-Welch) [3][7], метод JPEG (Joint Photographic Experts Group) [4][7], фрактальный алгоритм [5][7]. Сравнение данных алгоритмов было выполнено на основе требований к сжатию и восстановлению изображений методом сингулярного разложения. В рамках данной статьи было выявлено несколько аналогов, подходящих для обработки изображений, в зависимости от качества сжатия, формата и класса изображений. Фрактальный алгоритм используется в случае, когда размер 24-х битного изображения нужно максимально уменьшить. Метод LZW, в отличие от фрактального алгоритма, работает с 1-8 битными изображениями и сжимает без потерь.**

Ключевые слова – ***сжатие, восстановление, коэффициент сжатия, формат изображений, сжатие без потерь, класс изображений*****.**

I. ВВЕДЕНИЕ

На сегодняшний день практически у каждого жителя планеты имеется смартфон с возможностью выхода в мировую сеть Интернет. Многие пользователи обмениваются фотографиями и картинками, размеры которых могут достигать до нескольких гигабайт памяти, что обходится очень дорого при оплате мобильного интернета. Так, например, самым объёмным снимком на данный момент является снимок Луны, склеенный их 10581 фотографии, занимаемый 950 гигабайт памяти. Средний размер фотографий обычных пользователей, конечно, меньше и занимает от 1 до 20 Мб, но всё же, при отправке большого количества фотографий, трафик быстро расходуется, что является неэффективным решением. Для того, чтобы минимизировать потери памяти для хранения фотографий на смартфонах и облачных хранилищах, а также передачи другим пользователям, используются алгоритмы сжатия изображений. Целью данного исследования является нахождение алгоритмов, которые максимально эффективно смогут обрабатывать изображения для сжатия с возможностью восстановления с минимальными потерями данных. Для достижения поставленной цели, анализа и оценки эффективности, необходимо решить следующие задачи:

1. Изучить существующие алгоритмы сжатия и восстановления изображений.
2. Выявить качественные характеристики каждого из методов сжатия.
3. Провести сравнительный анализ изученных алгоритмов сжатия.
4. Выбор наиболее подходящего/ подходящих методов, позволяющих максимально эффективно и с минимальным количеством потерь обрабатывать различные форматы изображений.

II. ОБЗОР АЛГОРИТМОВ, ИСПОЛЬЗУЕМЫХ ПРИ СЖАТИИ И ВОССТАНОВЛЕНИИ ИЗОБРАЖЕНИЙ

Принцип отбора аналогов

Одной из причин использования метода решения СЛАУ с использованием сингулярного разложения матрицы системы является преобразование изображений. В основе выбора аналогов лежат алгоритмы, позволяющие сжимать и восстанавливать изображения. Примеры поисковых запросов: математические алгоритмы сжатия и

восстановления изображений, технологии jpeg, компьютерная обработка изображений, сжатие изображений методом сингулярного разложения, алгоритм SVD-сжатия, математические алгоритмы сжатия изображений.

Алгоритмы

1. *Алгоритм Хаффмана [1][7]*

Один из классических алгоритмов шифрования и восстановления данных, который используется для обработки информации, в частности, изображений. Алгоритм разработан в 60-х годах Дэвидом Хаффманом. Использует только частоту появления одинаковых байт в изображении. Данный метод кодирования состоит из двух этапов: построение оптимального кодового дерева и построение отображения код-символа на основе построения дерева. Классический алгоритм Хаффмана требует записи в файл таблицы соответствия кодируемых символов и кодирующих цепочек. Для того, чтобы не проходиться по изображению 2 раза используется либо готовая постоянная таблица, либо таблица стоится в процессе архивации/разархивации.

2. *Групповое сжатие (RLE) [2][6][7]*

Один из самых старых и простых способов архивации графики. При использовании данного алгоритма, изображение вытягивается в цепочку байт по строкам растра. Сжатие достигается за счёт того, что в исходном изображении встречаются цепочки одинаковых байт. Такие цепочки заменяются на пары <Кол-во повторений, Символ>, что уменьшается избыточность данных. Данный алгоритм предназначен в основном для деловой графики, в изображениях которой встречаются часто повторяющиеся символы.

3. *Метод LZW [3][7]*

Универсальный алгоритм сжатия данных без потерь, который был создан в 1984 году. Использует дерево для представления и хранения цепочек, т.к. выгодно сжимать даже цепочки, состоящие из 2-х байт. Процесс сжатия выглядит следующим образом: последовательно считываются символы входного потока и проверяется, есть ли в созданной ранее таблице строк такая строка. В случае, если строка уже есть, считывается следующий символ, если строки нет, то в поток заносится код для предыдущей найденной строки. Метод LZW, как и RLE, лучше действует на изображениях, содержащих однородных, свободных от шума участки цветов. При этом он действует гораздо лучше, чем RLE, при сжатии произвольных графических данных, но процесс кодирования и распаковки происходит медленнее.

4. *Метод JPEG [4][7]*

Один из самый новых и мощных алгоритмов сжатия данных. Оперирует областями 8x8 пикселов, на которых цвет и яркость меняются сравнительно плавно. Сжатие JPEG осуществляется за счет плавности изменения цветов в изображении. Алгоритм основан на дискретном косинусоидальном преобразовании (ДКП), которое используется для получения матрицы коэффициентов. ДКП раскладывает изображение по амплитудам некоторых частот и впоследствии получается матрица, в которой многие коэффициенты либо близки, либо равны нулю. Также можно аппроксимировать коэффициенты, что не сильно ухудшит качество изображения. В результате преобразования выборки 8 на 8 получаем выборку спектра 8 на 8. При этом низкие частоты содержатся в верхнем левом углу спектра, а высокие в правом нижнем. Высокие частоты можно обнулить и не хранить.

5. *Фрактальный алгоритм [5][7]*

Фрактальная архивация основана на том, что изображение представляется в более компактной форме - с помощью коэффициентов системы итерирующих функций (СИФ). СИФ представляется собой набор трёхмерных афинных преобразований, которые преобразуют одно изображение в

другое. Строго говоря, при архивации изображений СИФ задаёт фрактал. Для фрактального алгоритма компрессии, как и для других алгоритмов сжатия с потерями, очень важны механизмы, с помощью которых можно будет регулировать степень сжатия и степень потерь: ограничение количества аффинных преобразований, запрет за дробление фрагментов размером меньше некоторого количества точек.

Критерии сравнения аналогов

1. *Коэффициент сжатия (лучший)*

Чем лучше сжимается изображение, тем лучше и эффективнее работает алгоритм. Одним из важных критериев является коэффициент сжатия в лучшем случае.

2. *Коэффициент сжатия (средний)*

Во сколько раз в среднем уменьшается размер изображения.

3. *Коэффициент сжатия (худший)*

Во сколько раз уменьшается размер изображения в худшем случае.

4. *Формат*

Для использования вышеперечисленных алгоритмов, нужны определенные расширения изображений. Следовательно, формат изображения является одним из важных критериев.

5. *Принцип*

Для сравнительного анализа также нужно учесть тот факт, теряются ли какие-либо данные при архивации.

6. *Класс изображений*

Используется для того, чтобы сравнить сколько цветов может содержать изображение для использования алгоритма.

Результат сравнения алгоритмов представлен в табл. 1.

ТАБЛИЦА 1. СРАВНЕНИЕ АЛГОРИТМОВ ПО КРИТЕРИЯМ

	Алгоритм Хаффмана	Групповое сжатие (RLE)	Метод LZW	Метод JPEG	Фрактальный алгоритм
Коэффициент сжатия (лучший)	8.0	32.0	1000.0	200.0	2000.0
Коэффициент сжатия (средний)	1.5	2.0	4.0	101.0	1001.0
Коэффициент сжатия (худший)	1.0	0.5	0.7	2.0	2.0
Формат	GIF, TIFF	BMP, GIF, TIFF, TGA	GIF, TIFF	Quick Time, PostScript Level 2, Tiff 6.0, JPEG	FIF
Принцип	Без потерь	Без потерь	Без потерь	С потерями	С потерями
Класс изображений	8 битные	3, 4-х битные	1-8 битные	24-битные	24-битные

III. ВЫВОДЫ ПО ИТОГАМ СРАВНЕНИЯ

В результате обзора аналогов и сравнения их по выделенным критериям можно сделать вывод, что среди рассмотренных алгоритмов нет такого, что позволил бы в среднем сжимать изображения с более высоким коэффициентом и восстанавливать без каких-либо потерь для большого количества форматов изображений. Например, алгоритм Хаффмана не подходит, т.к. маленький коэффициент сжатия и используется только для 8-битных изображений. При использовании RLE изображение может стать только больше из-за того, что коэффициент сжатия в худшем случает составляет 0.5. Метод LZW лучше предыдущих, если сравнивать коэффициент сжатия в лучшем случае и классы изображений, но коэффициент сжатия в худшем случае меньше единицы, что не всегда может быть эффективным. Метод JPEG и фрактальный алгоритм сжимают 24-битные изображения с большим коэффициентом, но сжатие происходит с потерями данных.

IV. ВЫБОР МЕТОДА РЕШЕНИЙ

Предоставленный обзор и анализ алгоритмов сжатия данных необходим для того, чтобы определить, какие алгоритмы в каких случаях использовать. Были описаны

достоинства и недостатки каждого из алгоритмов и составлена таблица сравнения по критериям. В итоге, после реализации ПО, позволяющего находить сингулярные числа при решении системы алгебраических уравнений и сжимать изображения, будет проведено сравнение с методом сжатия JPEG и фрактальным алгоритмом по всем критериям; будет предоставлен вывод результатов в виде графиков, иллюстрирующих выходную информацию.

ВЫВОДЫ

В ходе проведения исследования была достигнута основная цель, а именно были найдены алгоритмы, которые максимально эффективно обрабатывают изображения для сжатия с возможностью восстановления с минимальными потерями данных. Фрактальный алгоритм и сжатие методом JPEG сжимает с большим коэффициентом, но при этом теряются данные, в то время, как LZW позволяет сжать с коэффициентом до 1000 в лучшем случае без потерь данных. Недостаток этого алгоритма заключается в том, что в худшем случае размер файла наоборот может увеличиться в размере. Поэтому, было принято решение программно реализовать метод решения СЛАУ с использованием сингулярного разложения матрицы системы для дальнейшего сжатия изображений. В дальнейшем планируется провести сравнение алгоритмов LZW, JPEG и фрактального алгоритма с SVD-сжатием изображений.

ИСТОЧНИКИ

[1] Wikipedia / URL: https://en.wikipedia.org/ wiki/Huffman_coding (Дата обращения: 20.12.2020)

[2] Habr / URL: https://habr.com/ru/post/141827/ (Дата обращения: 20.12.2020)

[3] Wikipedia / URL: https://en.wikipedia.org/ wiki/Lempel%E2%80%93Ziv%E2%80%93Welch (Дата обращения: 20.12.2020)

[4] ОмГУ / URL: http://www.univer.omsk.su /omsk-old/Edu/infpro/1/jpeg/algcz2.htm (Дата обращения: 20.12.2020)

[5] Habr / URL: http://mf.grsu.by/UchProc /livak/po/comprsite/theory_fractal.html (Дата обращения: 20.12.2020)

[6] ITMO / URL: http://aco.ifmo.ru/el_books /image_processing/3_04.html (Дата обращения: 20.12.2020)

[7] Библиотека Максима Мошкова / URL: http://www.lib.ru/TECHBOOKS/ALGO/VATOLIN/algcomp.htm (Дата обращения: 20.12.2020)

[8] Электронный ресурс учебного научно-методического комплекса «Математическая физика» и «Численные методы» / URL: http://www.lvg.kubsu.ru/files/Lezhnev_Markovskiy_Image_compression.pdf (Дата обращения: 20.12.2020)