



УДК 004.932:777.3

**И. И. КАРПОВ, аспирант кафедры информационных систем и технологий;
Д. И. КИСЛИЦЫН, канд. техн. наук, доц., зав. кафедрой информационных
систем и технологий**

МЕТОДЫ СЖАТИЯ РАСТРОВЫХ ИЗОБРАЖЕНИЙ

ФГБОУ ВО «Нижегородский государственный архитектурно-строительный университет». Россия, 603952, г. Нижний Новгород, ул. Ильинская, д. 65.

Тел.: (919) 388-26-00; эл. почта: kii137@yandex.ru

Ключевые слова: растровые изображения, методы сжатия, сжатие без потерь, сжатие с потерями.

Рассматриваются два типа методов сжатия изображений: без потерь, обеспечивающие полное восстановление изображения (Хаффман, LZ77, Deflate, QOI, используемые в PNG и GIF), и с потерями, позволяющие существенно уменьшать размер за счет частичной потери информации (DCT и вейвлет-преобразования в JPEG и видеокодеках). Выполнено сравнение результатов сжатия трех типов изображений разными методами и выявлены достоинства и недостатки рассматриваемых методов.

Введение

Современные цифровые технологии создают огромные объемы графической информации, что требует эффективных способов хранения и передачи. Растровые изображения занимают значительную долю мультимедийных данных благодаря высокой детализации, но их объем быстро растет с увеличением разрешения и качества. Это создает высокие требования к вычислительным ресурсам и пропускной способности сетей.

Сжатие изображений позволяет уменьшить объем данных за счет устранения избыточности при сохранении визуального качества. Различают два основных типа сжатия: без потерь, при котором изображение полностью восстанавливается (используется, например, в PNG), и с потерями – обеспечивающее более высокую степень компрессии за счет удаления малозаметных деталей (как в JPEG) [1].

В графических системах применяются два типа изображений – растровые и векторные. Растровые формируются из пикселей, что позволяет точно передавать фотографии и сложные текстуры, но приводит к большим размерам файлов. Векторные изображения, основанные на математических примитивах, масштабируются без потери качества, однако плохо подходят для фотографий.

Растровая графика получила наибольшее распространение благодаря реалистичности и универсальности. Она используется в фотографии, веб-дизайне, мультимедиа, медицине и других сферах. Однако большие размеры файлов делают необходимым применение алгоритмов сжатия, позволяющих уменьшить объем без заметной потери качества и повысить эффективность хранения и передачи изображений. Существует два метода сжатия: с потерями и без потерь.

Методы сжатия без потерь

Сжатие без потерь основано на устранении избыточности данных без изменения исходной информации. После декодирования изображение полностью



восстанавливается в первоначальном виде, без каких-либо искажений. Такой метод применяется в случаях, когда критически важно сохранить точные данные. Классическими примерами алгоритмов без потерь являются методы Хаффмана, *LZ77*, *Deflate*, а также форматы *PNG* и *GIF* [2].

Метод Хаффмана относится к числу фундаментальных алгоритмов энтропийного кодирования и основан на построении двоичных кодов переменной длины, обратно пропорциональной частоте появления символов. Алгоритм начинается с анализа частотности элементов, после чего создается бинарное дерево, где каждому значению присваивается уникальный код (часто встречающиеся символы получают короткие комбинации, а редкие – более длинные). Это обеспечивает уменьшение среднего количества бит на символ и, следовательно, объема данных. Преимуществами метода являются простота, высокая эффективность при неравномерном распределении вероятностей и устойчивость к избыточности, однако при сложных распределениях вероятностей он уступает более современным подходам.

Алгоритм *LZ77* (*Lempel-Ziv*, 1977) представляет собой словарный метод сжатия без потерь, основанный на поиске повторяющихся последовательностей данных. Он использует скользящее окно, состоящее из области поиска (ранее обработанных данных) и буфера просмотра (текущих данных). Совпадения заменяются тройкой параметров: смещение, длина и следующий символ. Такой принцип позволяет эффективно сжимать изображения с повторяющимися структурами, но при высокой энтропии данных эффективность метода снижается, а хранение окна требует дополнительных вычислительных ресурсов [3].

В сжатии изображений метод *LZ77* часто используется в комбинации с энтропийным кодированием, что позволяет достичь более высокой степени компрессии, примером является алгоритм *Deflate*, применяемый в формате *PNG*. Здесь *LZ77* отвечает за устранение повторяющихся последовательностей, а метод Хаффмана – за эффективное кодирование полученных ссылок и оставшихся символов. Также к классическому алгоритму *Deflate* добавляют фильтрацию. Фильтры применяются к каждой строке байтов данных и всего существует пять видов: отсутствие фильтра (отсутствие фильтрации, данные подаются в *Deflate* как есть); *Sub* (заменяет каждый пиксель разницей между его значением и значением предыдущего в той же строке); *Up* (использует разницу между текущим пикселием и пикселием из предыдущей строки); *Average* (усредняет значения фильтров *Sub* и *Up*); *Paeth* (вычисляется на основе трех соседних пикселей слева, сверху и сверху-слева, выбирая ближайшее значение, и сохраняет разницу между реальным значением пикселя и вычисленным значением).

После применения фильтрации алгоритм *Deflate* показывает более сильное сжатие, данный алгоритм сжатия сам выбирает нужный фильтр и не всегда выбранные фильтры могут показать максимальный результат.

Проведем несколько пробных преобразований для оценки метода сжатия и сравнения его с другими известными алгоритмами. Для эксперимента были выбраны три изображения в формате *PNG*, представленные на рис. 1–3.

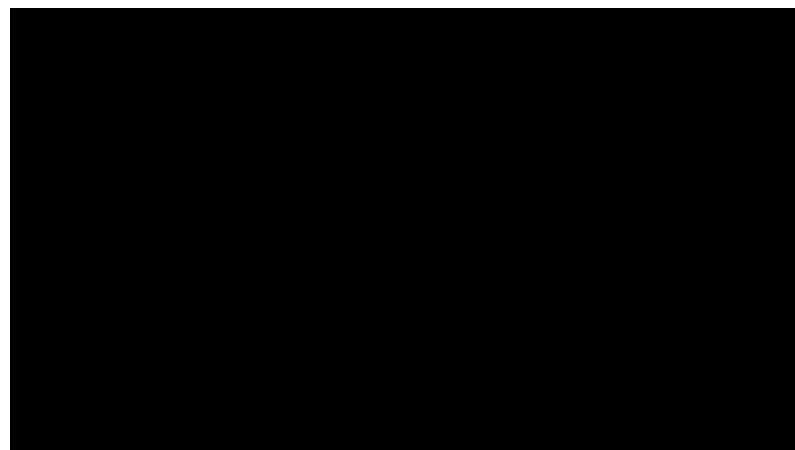


Рис. 1. Черный фон (наилучший исход)



Рис. 2. Изображение пейзажа (усредненный исход)

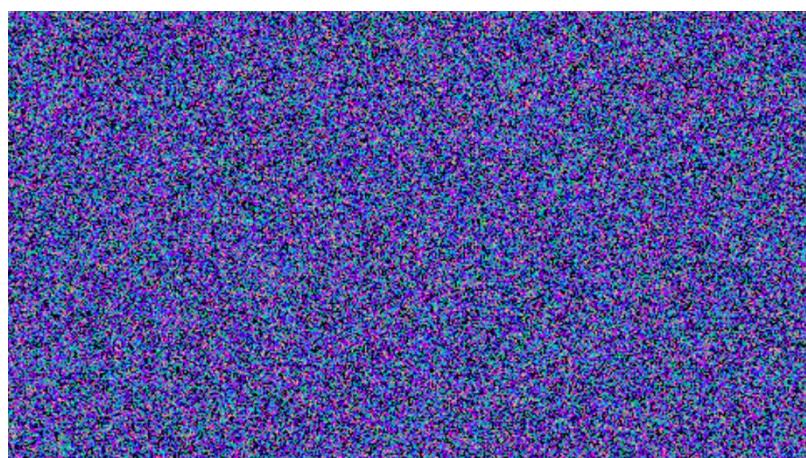


Рис. 3. Случайный шум (наихудший исход)



Для тестирования алгоритмов сжатия выбраны три типа изображений: черный фон (наилучший случай, все пиксели равны нулю), пейзаж (средний, типичный пример реальных изображений) и случайный шум (наихудший случай, данные полностью случайны). Все изображения имеют одинаковое разрешение 1920 на 1080 и три цветовых канала (*RGB*), что дает исходный размер 6 220 800 байт.

После применения алгоритма *Deflate* к данным трем изображениям получили следующие результаты: черный фон получилось сжать до 6 063 байт, коэффициент сжатия 1026,03; пейзаж до 4 693 533 байт, коэффициент получился 1,33; а шум сжался до 6 130 816 байт (коэффициент 1,01).

Одним из новых методов сжатия без потерь является *QOI. Quite OK Image format* – это формат представления растровых изображений с использованием сжатия без потерь, предложенный Домиником Шаблевски в 2021 году. Основная цель разработки заключалась в создании простого, быстрого и достаточно эффективного метода сжатия, ориентированного на практическое использование, а не на достижение максимального коэффициента компрессии. В отличие от *PNG*, который использует комбинацию фильтрации и алгоритма *Deflate*, *QOI* реализует более простой механизм кодирования, основанный на повторяющихся шаблонах в последовательности пикселей. В его основе лежат четыре ключевых техники: регистровое кодирование (использование массива из 64 слотов для хранения ранее встречавшихся пикселей); кодирование разности (сохранение небольших изменений относительно предыдущего пикселя); *RLE – Run-Length Encoding* (для последовательностей одинаковых значений); прямое кодирование (для нестандартных случаев). Каждый пиксель кодируется одним байтом-заголовком, определяющим тип записи, и сопровождается дополнительными данными. Главным достоинством *QOI* является высокая скорость кодирования и декодирования, что делает его привлекательным для приложений реального времени. В анонсирующем блоге сказано, что скорость кодирования в 20 раз выше, а декодирования в 3 раза выше чем аналогичные операции, проводимые с форматом *PNG*. Также преимуществом является простота реализации, данный алгоритм можно описать всего в несколько сотен строк кода без использования сторонних библиотек, при этом степень сжатия формата сопоставима с известным *PNG*. Ограничением *QOI* является отсутствие фильтров и адаптивных схем, что приводит к менее эффективному сжатию изображений со сложной структурой. Кроме того, формат не конкурирует с *WebP* или *JPEG XL* по степени компрессии, позиционируясь как «достаточно хороший» компромисс между скоростью и эффективностью и как удачное инженерное решение, в котором приоритет отдается простоте и скорости работы, а не максимальной степени сжатия [4].

Также проведем эксперименты работы данного алгоритма. После сжатия черного фона (рис. 1) итоговый размер файла составил 33 469 байт, что является уменьшением размера в 185,87 раз. Следующая картинка пейзажа (рис. 2) после обработки стала занимать 3 496 744 байт, а степень сжатия равна 1,78. Также после преобразования шума (рис. 3) итоговый размер составил 7 963 071 байт, что равняется 0,78 степени сжатия. Получается, после применения алгоритма случайный шум стал занимать даже больше места.

Методы сжатия с потерями

Вторым и противоположным типом сжатия изображений является сжатие с потерями, которое предполагает удаление информации, считающейся



малозначимой для человеческого восприятия. Наиболее часто исключаются высокочастотные компоненты, несущественные для визуальной оценки изображения, в результате достигается максимально высокая степень сжатия, но с необратимой потерей части данных. Такие методы нашли широкое применение там, где приоритетом является уменьшение размера файлов при сохранении субъективно приемлемого качества, например, при загрузке изображений на сайте, а наиболее известным примером является формат *JPEG* [5].

Дискретное косинусное преобразование (*DCT*) – один из ключевых методов сжатия изображений с потерями. Его суть заключается в переходе от пространственного представления изображения к частотному, что позволяет удалять малозначимые компоненты и сокращать объем данных. Изображение делится на блоки 8 на 8 пикселей, для каждого вычисляется двумерное *DCT*, формирующее матрицу коэффициентов. Левый верхний элемент (*DC*-компонент) отражает среднюю яркость блока, остальные (*AC*-компоненты) – частотные колебания разной амплитуды. После этого выполняется квантование, где коэффициенты делятся на значения из матрицы квантования с округлением, при этом высокочастотные данные упрощаются сильнее, обеспечивая основное сжатие. Большинство коэффициентов после квантования становятся нулями, поэтому для дополнительного уменьшения объема применяется энтропийное кодирование [6].

Основными недостатками являются артефакты (становятся заметны границы блоков) и невозможность восстановить исходное изображение. На практике (формат *JPEG*) изображение преобразуется из *RGB* в цветовое пространство *YCbCr*, яркостной канал (*Y*) сжимается точнее, а хроматические каналы (*Cb* и *Cr*) часто подвергаются субдискретизации (например, 4:2:0), так как глаз менее чувствителен к цветовым деталям [7].

Проведем преобразования на тех же изображениях. Результатом работы алгоритма после применения его к однородному фону (рис. 1), стало уменьшение размера в 187 раз, а итоговый размер составил 33 267 байт. При обработке пейзажа (рис. 2), итоговый размер составил 424 720 байт, полученный коэффициент сжатия 14,65. После проверки наихудшего исхода (рис. 3) размер изображения составил 905 178 байт, а степень сжатия при этом 6,87.

Вейвлет-преобразование: основано на использовании специальных функций (вейвлетов), обладающие одновременно локализацией и в пространстве для изображений, и в частотной области. Вейвлет-преобразование работает на нескольких масштабах и не ограничивается фиксированными блоками, что позволяет избежать характерных артефактов «блокинга». Процесс начинается с разложения изображения при помощи пары фильтров: низкочастотный фильтр выделяет медленно изменяющиеся компоненты (глобальную структуру изображения); высокочастотный фильтр извлекает детали и резкие переходы (границы, текстуры). После фильтрации выполняется *downsampling* – это выборка через один элемент, что уменьшает размер данных вдвое. Для двумерных изображений они применяются построчно и поколонично, в результате чего изображение делится на четыре области: *LL* (низкие частоты по строкам и столбцам) – аппроксимация, содержит основное содержание изображения; *LH* (низкие по строкам, высокие по столбцам) – горизонтальные детали; *HL* (высокие по строкам, низкие по столбцам) – вертикальные детали; *HH* (высокие по строкам и столбцам) – диагональные детали [8].



Процесс может повторяться рекурсивно для области LL , что формирует многоуровневое разложение. Для сжатия используется свойство: значительная часть изображения сосредоточена в области LL , тогда как коэффициенты в областях LH , HL и HH имеют малые значения. Эти коэффициенты подвергаются квантованию, часто приводящему их к нулю. Далее оставшиеся данные кодируются методами без потерь для дополнительного сжатия [9].

Вейвлет преобразование очень похоже на DCT , мы ожидаем примерно те же результаты. Черный фон (рис. 1) получилось сжать до все тех же 33 267 байт и степень сжатия 187. Размер изображения среднего исхода (рис. 2) уменьшился до 494 483 байт, что является уменьшением размера в 12,58 раз. И изображение случайного шума (рис. 3) получилось сжать до 1 149 436 байт, степень сжатия равна 5,41.

Для сравнения эффективности методов сжатия предлагается сводная таблица. В ней представлены результаты проведенных экспериментов для наглядного сравнения.

Таблица

Результаты использования методов сжатия изображений

Тип сжатия		Deflate	QOI	DCT	Wavelet
Лучший исход (Черный фон)	Исходный размер	6 220 800 байт			
	Итоговый размер	6 063 байт	33 469 байт	33 267 байт	33 267 байт
	Степень сжатия	1026,03	185,87	187,0	187,0
Среднее значение (Пейзаж)	Исходный размер	6 220 800 байт			
	Итоговый размер	4 693 533 байт	3 496 744 байт	424 720 байт	494 483 байт
	Степень сжатия	1,33	1,78	14,65	12,58
Худший исход (случайный шум)	Исходный размер	6 220 800 байт			
	Итоговый размер	6 130 816 байт	7 963 071 байт	905 178 байт	1 149 436 байт
	Степень сжатия	1,01	0,78	6,87	5,41

Заключение

Каждый метод обладает собственными преимуществами и ограничениями: одни лучше справляются с изображениями, содержащими большие однородные области, другие – с фотографиями и сложными текстурами, а некоторые рассчитаны на использование на большем количестве устройств, жертвуя степенью сжатия.

Классические подходы, основанные на статистических моделях и кодировании энтропии, достигли высокой эффективности, однако дальнейшее улучшение становится затруднительным. В связи с этим особое внимание исследователей все больше привлекают методы, основанные на искусственном



интеллекте. ИИ и глубокое обучение уже активно применяются в задачах обработки изображений и демонстрируют значительный потенциал в области сжатия. Нейросетевые модели позволяют находить более компактные и содержательные представления визуальных данных, чем традиционные алгоритмы [10]. Такие методы обеспечивают адаптивность под особенности изображения или целого класса данных. Благодаря этому, удается достигать лучшего баланса между степенью сжатия и качеством изображения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сэломон, Д. Сжатие данных, изображений и звука : перевод с английского В. В. Чепыкова / Д. Сэломон. – Москва : Техносфера, 2004. – 368 с. – ISBN 5-94836-027-X.
2. Алгоритм сжатия растровых изображений без потерь / М. С. Войтенко, А. С. Войтенко, Ю. Н. Феденко, А. И. Антонов. – Текст : электронный // Вестник ВГТУ. – 2024. – Том 20, № 2. – С. 25–28. – URL: <https://cyberleninka.ru/article/n/algoritm-szhatiya-rastrovyh-izobrazheniy-bez-poter>.
3. Андреев, В. Алгоритмы автоматической обработки и анализа аэрокосмических снимков // Оптический журнал. – 2007. – Том 74, № 5. – С. 12–30.
4. Krainyk, Y. Method of Image Compression Using Image Preprocessing, and Huffman and Quite Ok Image Algorithms / Y. Krainyk // Èlektronnoe modelirovanie. – 2024. – Volume 46, № 2. – P. 75–87.
5. Нгуен, В. Ч. Анализ методов интерполяции в задачах реконструкции изображений / В. Ч. Нгуен, А. А. Тропченко. – Текст : электронный // Наука и образование : электронный журнал. – 2017. – № 3. – С. 170–181. – URL: <https://cyberleninka.ru/article/n/analiz-metodov-interpolyatsii-v-zadachah-rekonstruktsii-izobrazheniy>.
6. Sawant, Y. Hybrid Image Compression Method using ANN and DWT / Y. Sawant, L. S. Admuthe // International Journal of Computer Applications (0975 – 8887). – 2014 June. – Volume 95, № 11. – P. 35–38.
7. Chen, D. A hybrid stochastic interpolation and compression method for kernel matrices / D. Chen // J Comput Phys. – 2023. – DOI 10.1016/j.jcp.2023.112491.
8. Alakuijala, J. JPEG XL next-generation image compression architecture and coding tools / J. Alakuijala, S. Boukortt, Z. Szabadka // Conference : applications of digital image processing XLII. – 2019 september. – DOI.org/10.1117/12.2529237.
9. ShiftLIC: lightweight learned image compression with spatial-channel shift operations / Bao Y., Tan W., Jia C., Li M. [et al]. – DOI 10.48550/arXiv.2503.23052.
10. Deep Learning Image Compression Method Based On Efficient Channel-Time Attention Module / Ji X., Yang X., Yue Z., Yang H.[et al] // Scientific reports. – 2025 may. – № 15 (1).

KARPOV Ilya Igorevich, postgraduate student of the chair of information system and technology; KISLITSYN Dmitry Igorevich, candidate of technical sciences, associate professor, holder of the chair of information systems and technologies

BITMAP IMAGE COMPRESSION METHODS

Nizhny Novgorod State University of Architecture and Civil Engineering.
65, Iljinskaya St., Nizhny Novgorod, 603952, Russia.

Tel.: (919) 388-26-00; e-mail: kii137@yandex.ru

Key words: bitmap images, compression methods, lossy compression, lossless compression.



Two types of image compression methods are considered: lossless methods, providing full image recovery (Huffman, LZ77, Deflate, QOI used in PNG and GIF), and lossy methods, allowing significant size reduction due to partial loss of information (DCT and wavelet transformations in JPEG and video codecs). The results of compression of three types of images by different methods are compared and the advantages and disadvantages of the considered methods are revealed.

REFERENCES

1. Salomon D. Szhatie dannykh, izobrazheniy i zvuka [Data, image, and sound compression]. Moscow, Tekhnosfera, 2004, 368 p.
2. Voytenko M. S., Voytenko A. S., Fedenko Yu. N., Antonov A. I. Algoritm szhatiya rastrovykh izobrazheniy bez poter [Lossless Raster Image Compression Algorithm]. Vestnik VGTU [Proceedings of the Voronezh State University of Engineering Technologies]. 2024, № 2. URL: <https://cyberleninka.ru/article/n/algoritm-szhatiya-rastrovyh-izobrazheniy-bez-poter>.
3. Andreev V. Algoritmy avtomaticheskoy obrabotki i analiza aehrokosmicheskikh snimkov [Algorithms for automatically processing and analyzing aerospace pictures]. Opticheskiy zhurnal [Journal of Optical Technology]. 2007, Vol 74, № 5, P. 12–30.
4. Krainyk Y. Method of Image Compression Using Image Preprocessing, and Huffman and Quite Ok Image Algorithms. Èlektronnoe modelirovaniye. 2024. Vol 46. № 2. P. 75–87.
5. Nguyen V. Ch., Tropchenko A. A. Analiz metodov interpolatsii v zadachakh rekonstruktsii izobrazheniy. [Analysis of Interpolation Methods in Image Reconstruction Tasks]. Nauka i obrazovanie [Science and Education]: elektronnii zhurnal. 2017, № 3, P. 170–181. URL: <https://cyberleninka.ru/article/n/analiz-metodov-interpolatsii-v-zadachah-rekonstruktsii-izobrazheniy>.
6. Sawant Y., Admuthe L. S. Hybrid Image Compression Method using ANN and DWT. International Journal of Computer Applications (0975 – 8887). 2014. June. Vol. 95. № 11. P. 35–38.
7. Chen D. A. hybrid stochastic interpolation and compression method for kernel matrices. J Comput Phys. 2023 Dec 1; 494:112491. doi: 10.1016/j.jcp.2023.112491. Epub 2023 Sep 12.
8. Alakuijala J., Boukortt S., Szabadka Z. JPEG XL next-generation image compression architecture and coding tools. Conference: Applications of Digital Image Processing XLII. 2019 September. DOI.org/10.11117/12.2529237.
9. Bao Y., Tan W., Jia C., Li M. [et al.] ShiftLIC: Lightweight Learned Image Compression with Spatial-Channel Shift Operations. DOI:10.48550/arXiv.2503.23052. License CC BY 4.0. 2025 March.
10. Ji X., Yang X., Yue Z., Yang H. [et al.] Deep Learning Image Compression Method Based On Efficient Channel-Time Attention Module. Scientific Reports. 2025 May. 15 (1).

© И. И. Карпов, Д. И. Кислицын, 2025

Получено: 09.11.2025 г.