МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)

Институт информационных технологий, математики и механики
Кафедра алгебры, геометрии и дискретной математики

Направление подготовки:
«Фундаментальная информатика и информационные технологии»
Профиль подготовки: «Инженерия программного обеспечения»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА

Тема:
«Эволюция алгоритмов сжатия изображений: от RLE до JPEG»

Выполнил:
студент группы: 3821Б1ФИ2
Казанцев Евгений Александрович

_____

Научный руководитель:
к.ф.-м.н., доцент каф. АГДМ
Смирнова Татьяна Геннадьевна

_____

Нижний Новгород
2025

Аннотация

Сжатие изображений — применение алгоритмов сжатия данных к изображениям, хранящимся в цифровом виде. В результате сжатия уменьшается размер изображения, из-за чего уменьшается время передачи изображения по сети и экономится пространство для хранения.

Сжатие изображений подразделяют на сжатие с потерями качества и сжатие без потерь.

Преимущество методов сжатия с потерями над методами сжатия без потерь состоит в том, что первые дают возможность большей степени сжатия, при этом, продолжая удовлетворять поставленным требованиям, а именно — искажения должны быть в допустимых пределах чувствительности органов человеческого восприятия.

Методы сжатия с потерями используются для сжатия аналоговых данных — чаще всего звука или изображений.

В таких случаях распакованный файл может сильно отличаться от оригинала по размеру, но практически неотличим для человека «на слух» и «на глаз» в большинстве применений.

Множество методов фокусируются на физических особенностях органов чувств человека. К примеру психоакустические модели определяют то, как сильно звук может быть сжат без ухудшения воспринимаемого человеком качества звука.

При использовании сжатия с потерями необходимо учитывать, что повторное сжатие обычно приводит к деградации качества. Однако, если повторное сжатие выполняется без каких-либо изменений сжимаемых данных, качество не меняется. Так например, сжатие изображения методом JPEG, восстановление его и повторное сжатие с теми же самыми параметрами не приведет к снижению качества.

ОГЛАВЛЕНИЕ

Введение

Сжатие информации с потерями — это алгоритмическое преобразование данных, направленное на уменьшение их объёма за счёт допустимых потерь качества. Необходимость сокращения объёма данных актуальна с появлением цифровых технологий и остается актуальной по сей день. Несмотря на развитие технологий хранения и передачи данных, ограничения по объёму памяти и скорости передачи информации по-прежнему требуют эффективных методов сжатия. Объёмы данных продолжают расти, их структура усложняется, а качество — повышается. Это особенно актуально для мультимедийных данных — изображений, видео и звука.

В качестве основных литературных источников данной работы используются: книга Д. Сэломона «Сжатие данных, изображений и звука» [1], труды П. Пенна [2], а также две книги Т. Рафгардена из серии «Совершенный алгоритм» [3,4]. В книге [1] содержится формальное описание алгоритмов сжатия с потерями, в частности алгоритма JPEG, включая его математическую основу — дискретное косинусное преобразование (DCT). Труды П. Пенна [2] дают детализированное объяснение реализации алгоритма JPEG на программном уровне. Книги [3,4] содержат информацию о базовых алгоритмических принципах, используемых в методах сжатия. В ходе работы также использовалась официальная документация библиотеки OpenCV [5] и языка программирования Python 3 [6], а так же библиотеки для создания графического интерфейса PyQt [7].

Вернуться и доработать

Существует огромное множество методов сжатия данных, основанных на различных предположениях о структуре сжимаемой информации. Все алгоритмы сжатия можно разделить на две основные группы: Сжатие без потерь — предполагает возможность точного восстановления исходных данных после сжатия. Сжатие с потерями — допускает потерю части данных ради значительного сокращения объема. Этот метод часто применяется к изображениям, видео и аудиоданным, где некоторая потеря качества незаметна для восприятия.

В данной работе рассматривается алгоритм сжатия с потерями на примере широко используемого метода JPEG. Алгоритм JPEG основан на использовании дискретного косинусного преобразования (DCT) для преобразования изображения, последующем квантовании и энтропийном кодировании. В рамках данной реализации из рассмотрения исключаются этапы энтропийного кодирования, в частности кодирование Хаффмана, а также альтернативные подходы, такие как вейвлет-преобразования. Этот метод обеспечивает высокую степень сжатия при сохранении приемлемого качества изображения. JPEG стал важным этапом в развитии технологий сжатия данных, объединив достижения математических и инженерных исследований своего времени, и дал мощный толчок развитию в области сжатия изображений.

Основной задачей работы является исследование алгоритма JPEG в контексте его математической основы и алгоритмической реализации. Предполагается рассмотреть влияние различных параметров сжатия на качество изображения, а также провести сравнение производительности различных реализаций алгоритма. В качестве возможных модификаций рассматриваются методы адаптивного квантования и постобработки изображения после декодирования.

Целью работы является изучение развития алгоритмов сжатия с потерями, начиная с фундаментальных математических открытий (например, дискретного косинусного преобразования), простых алгоритмов сжатия без потерь, и заканчивая созданием аналогичного алгоритма JPEG. В этой работе реализуется собственный алгоритм, основанный на ключевых этапах стандарта JPEG. Его структура почти аналогична базовому варианту. В дальнейшем по тексту он будет обозначаться как “модификация алгоритма JPEG” или просто “JPEG”. Так же планируется разработать программу с интерфейсом, позволяющую проводить эксперименты по сжатию изображений с использованием модификации алгоритма JPEG. Программа должна обладать следующими функциями:

- кодирование изображения методом JPEG с возможностью настройки параметров квантования и качества;
- визуализация результатов кодирования и декодирования;
- сравнение качества изображения до и после сжатия с использованием метрик SSIM и PSNR;
- возможность включения адаптивного квантования для улучшения качества при низких битрейтах.

Для достижения поставленной цели сформулированы следующие задачи:

- изучить теоретическую основу алгоритма JPEG, включая дискретное косинусное преобразование и квантование;
- реализовать алгоритм JPEG на языке программирования Python с использованием библиотеки OpenCV;
- разработать интерфейс для управления параметрами алгоритма и отображения результатов;
- провести эксперименты на различных изображениях, оценить влияние параметров на степень сжатия и качество изображения;
- проанализировать результаты экспериментов и сделать выводы о целесообразности использования различных модификаций алгоритма.

v

1 Основные понятия и определения

Перед тем как углубиться в описание алгоритма JPEG, важно рассмотреть основные понятия, лежащие в основе цифровой обработки и сжатия изображений.

Цифровое изображение — двумерный массив, где каждая ячейка представляет собой пиксель.

Таблица 1. Упрощенная схема цифрового изображения как двумерного массива пикселей

P(0,0)	P(0,1)	P(0,2)	P(0,3)
P(1,0)	P(1,1)	P(1,2)	P(1,3)
P(2,0)	P(2,1)	P(2,2)	P(2,3)
P(3,0)	P(3,1)	P(3,2)	P(3,3)

Пиксель (pixel) — наименьшая единица изображения, содержащая информацию о цвете или яркости.

Таблица 2. Представление пикселей с помощью RGB-значений и их цветов

RGB значения	**Цвет**
RGB(255,0,0)	
RGB(0,255,0)	
RGB(0,0,255)	

Наиболее распространённой моделью представления цветных изображений является RGB.

RGB (Red, Green, Blue) — аддитивная цветовая модель, в которой каждый цвет формируется путём смешивания трёх базовых компонентов: красного, зелёного и синего. Эта модель близка к принципу работы матриц дисплеев и сенсоров цифровых камер, поэтому RGB часто используется как исходный формат цветовых изображений.

В зависимости от глубины цвета, каждый пиксель кодируется определённым числом бит. Например, в изображении с глубиной 24 бита (8 бит на каждый из каналов RGB) один пиксель может представлять более 16 миллионов оттенков. С увеличением разрешения и глубины цвета возрастает и общий объём изображения, что создаёт необходимость в эффективном сжатии данных.

Однако при сжатии изображений RGB-пространство не является наиболее эффективным. Это связано с тем, что RGB не разделяет яркость и цветовую информацию, а человеческий глаз по-разному чувствителен к этим компонентам. В связи с этим перед сжатием изображения, как правило, преобразуются в цветовое пространство **YCbCr**, где **Y** отражает яркость (luminance), а **Cb** и **Cr** — цветовые отклонения от серого (chrominance).

Также необходимо упомянуть несколько ключевых понятий, тесно связанных с задачами кодирования изображений:

Разрешение изображения — количество пикселей по горизонтали и вертикали.

Избыточность — повторяющаяся или избыточная информация, которую можно исключить без потери качества или с минимальными потерями.

Энтропия — мера неопределённости или информационной насыщенности данных, определяющая нижнюю границу возможной степени сжатия.

Формат **JPEG** представляет собой стандарт кодирования изображений с потерями, определяющий последовательность преобразований, квантования и кодирования данных. Разработка формата началась в 1986 году с формированием международной рабочей группы под названием *Joint Photographic Experts Group*. На тот момент существовало множество несовместимых форматов для хранения графики, что затрудняло обмен изображениями между системами. JPEG стал первым по-настоящему универсальным и широко распространённым стандартом сжатия фотоизображений. Официальная спецификация была опубликована в 1992 году и используется до сих пор.

2 Алгоритмы и структуры данных

2.1 Преобразование цветового пространства (RGB → YCbCr)

Перед сжатием изображений в формате JPEG зачастую используется преобразование из цветового пространства RGB в YCbCr. Это обусловлено как особенностями восприятия цвета человеком, так и требованиями алгоритмов сжатия.

Такое разделение позволяет применить хрома-субдискретизацию — уменьшение разрешения цветовых компонент — без заметного ухудшения качества изображения. Для RGB с диапазоном значений от 0 до 255, преобразование в YCbCr выполняется по следующим формулам:

$$Y = 0,299R + 0.587G + 0,114B;$$

$$Cb = -0,168736R - 0,331264G + 0,5B + 128;$$

$$Cr = 0,5R - 0,418688G - 0,081312B + 128.$$

Здесь значения R, G, B принимаются в диапазоне $[0, 255]$. Смещение на 128 единиц в формулах для Cb и Cr необходимо для корректного представления как положительных, так и отрицательных значений в беззнаковом формате (unsigned int).

Поскольку компоненты Cb и Cr менее значимы для восприятия, их можно хранить с пониженным разрешением. Наиболее распространённый формат — 4:2:0, при котором на каждые 4 пикселя хранится 4 значения яркости Y и по 1 значению Cb и Cr. Это даёт значительное

уменьшение объёма данных без существенного ущерба для качества изображения.

2.2 Преимущества YCbCr

- Обеспечивает эффективное сжатие без видимых потерь качества;
- Разделение яркости и цвета позволяет применять различные методы компрессии к компонентам Y и CbCr;
- Поддерживается большинством стандартов обработки изображений и видео.

Таким образом, преобразование RGB $\rightarrow$ YCbCr является ключевым шагом в JPEG и других алгоритмах сжатия, позволяющим использовать особенности восприятия цвета человеком для достижения более высокой степени сжатия.

2.3 Субдискретизация

На втором этапе сжатия изображения применяется методика, известная как субдискретизация цветности (англ. chroma subsampling). Её ключевая идея заключается в сокращении объёма данных, содержащих информацию о цвете, путём уменьшения пространственного разрешения цветовых компонентов. Это возможно благодаря физиологической особенности зрительной системы человека: она в гораздо большей степени чувствительна к деталям яркости (компонент Y — luminance), чем к изменениям цветовых оттенков (компоненты Cb и Cr — chrominance).

В практической реализации данный метод означает, что цветовая информация (Cr и Cb) кодируется с меньшей точностью по сравнению с яркостной. Существует несколько вариантов схем хрома-субдискретизации, которые различаются по степени уменьшения разрешения цветовых компонентов:

4:4:4 — нет сжатия, все компоненты Y, Cb и Cr сохраняют оригинальное разрешение.

4:2:2 — разрешение Cb и Cr уменьшается в два раза по сравнению с Y.

4:2:0 — стандарт для большинства сжатых форматов, где разрешение Cb и Cr уменьшается в два раза как по вертикали, так и по горизонтали.

Схема 4:2:0 является наиболее распространенной для видео и изображений, так как она дает хороший баланс между качеством и сжатием. Она предполагает, что на каждые четыре пикселя яркости (расположенных в виде блока 2×2) приходится всего один усреднённый пиксель Cr и один усреднённый пиксель Cb. То есть цветовые значения вычисляются как средние значения для группы пикселей, и это значение применяется ко всей группе. В результате разрешение цветовых компонентов уменьшается в четыре раза (на 75%) по сравнению с яркостной компонентой.

Для визуализации можно представить исходное изображение как состоящее из трёх равнозначных слоёв:

Y (яркость),

Cb (синий цветовой компонент),

Cr (красный цветовой компонент).

До субдискретизации каждый из этих компонентов имел одинаковое разрешение и соответственно одинаковый вклад в общий объём данных:

$$Y + Cb + Cr = 1 + 1 + 1 = 3 \text{ единицы информации.}$$

После применения схемы 4:2:0, каждая из цветовых компонент уменьшается до $\frac{1}{4}$ исходного размера, а яркостная остаётся без изменений:

$$Y + \frac{1}{4}Cb + \frac{1}{4}Cr = 1 + 0.25 + 0.25 = 1.5 \text{ единицы информации.}$$

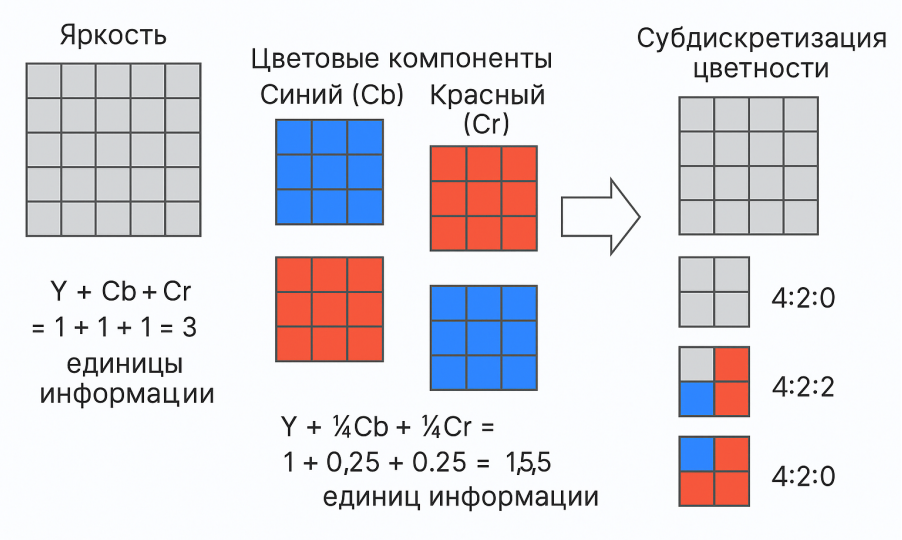

Рисунок 1. Как происходит преборазование

Таким образом, достигается двукратное уменьшение общего объёма данных, подлежащих хранению или передаче.

Что важно, такое преобразование происходит практически без визуальных потерь качества. Несмотря на то, что цветовые данные редуцируются, человеческий глаз практически не воспринимает различие между оригинальным и сжатым изображением. Именно поэтому субдискретизация цветности широко применяется в большинстве алгоритмов сжатия изображений и видео (JPEG, MPEG, H.264), где критически важно достичь компромисса между качеством и эффективностью хранения данных.

Литература

1. Сэломон, Д. Сжатие данных, изображений и звука / Д. Сэломон. – М.: Техносфера, 2006. – 368 с.

2. Томас, Кормен. Алгоритмы Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К, Штайн - 2-е изд., - Москва: Вильямс, 2011. – 1296 с.

3. Официальная документация OpenCV. – Режим доступа: -URL: https://docs.opencv.org/4.x/index.html (дата обращения: 19.02.2025). – Текст: электронный.

4. Земцов, А. Н. Сравнительный анализ эффективности методов сжатия изображений на основе дискретного косинусного преобразования и фрактального кодирования (начало) / И. В. Петрова, Н. Г. Мамаев. – Волгоград: Издательство Волгоградского государственного технического университета, 2015. – 8 с.

5. Гарсия, Г. Обработка изображений с помощью OpenCV / Г. Гарсия, О. Суарес, Х. Аранда, Х. Терсеро, И. Грасиа, Н. Энано. – Москва: ДМК, 2016. – 210 с.

6. Официальная документация PyQt5. – Режим доступа: -URL: https://www.riverbankcomputing.com/ static/Docs/PyQt5/. (дата обращения: 05.01.2025). – Текст: электронный.

7. Официальная документация языка Python. – В режиме доступа: -URL: https://docs.python.org/3/ (дата обращения: 05.01.2025). – Текст: электронный.