УСТРОЙСТВО ФОРМАТА JPEG

Коннов Михаил Александрович

студент, кафедры информационно вычислительные комплексы, Ульяновский Государственный Технический Университет РФ, г. Ульяновск

Зотов Глеб Владиславович

студент, кафедры информационно вычислительные комплексы, Ульяновский Государственный Технический Университет РФ, г. Ульяновск

Игнатьев Дмитрий Сергеевич

студент, кафедры информационно вычислительные комплексы, Ульяновский Государственный Технический Университет РФ, г. Ульяновск

Кадырова Динара Руслановна

студент, кафедры информационно вычислительные комплексы, Ульяновский Государственный Технический Университет РФ, г. Ульяновск

Лушников Леонид Леонидович

студент, кафедры информационно вычислительные комплексы, Ульяновский Государственный Технический Университет РФ, г. Ульяновск

Тамьярова Майя Владиславовна

научный руководитель, канд. техн. наук, Ульяновский Государственный Технический Университет, РФ, г. Ульяновск

Предположим, у нас есть снимок с разрешением в 12,2 мегапикселя, если быть точным 4032 на 3024 точки. Но сколько он должен весить?

Как известно, обычный цвет кодируется тремя базовыми цветами: красным, зеленым и синим.

Каждая из этих составляющих кодируется числом 8 бит. Таким образом, если перемножить, получим: 8 бит умножить на три цвета да на 4032 точки по горизонтали и еще на 3024 точки по вертикали и получится 292 миллиона бит, то есть почти 300 Мб. Но в реальности эта фотка весит всего 7,36 Мб, то есть в 40 раз меньше. Но как так получается?

Все благодаря легендарному формату для хранения изображений – JPEG.

Конечно, присутствует небольшая потеря качества, но по сравнению с тем, что размер файла был уменьшен в десятки раз, это уже не играет такой большой роли. И да, новые форматы тоже появляются, тот же HEIF, но JPEG все еще остается королем индустрии.

К началу восьмидесятых годов компьютеры уже умели хранить и показывают цифровые изображения, но делали это все по-разному. Существовало слишком много стандартов.

Банально нельзя было просто взять и отправить картинку с одного компьютера на другой.

УСТРОЙСТВО ФОРМАТА JPEG // Студенческий форум: электрон. научн. журн. Коннов М.А. [и др.]. 2023. № 5(228). URL: https://nauchforum.ru/journal/stud/228/123499 (дата обращения: 06.09.2025).

Для решения этой проблемы в 1986 году был собран комитет экспертов со всего мира под названием Объединенная Группа Экспертов по Фотографии (JPEG).

И этот коллектив в 1992 году создал стандарт сжатия цифровых изображений, который спасает нас до сих пор. А как вообще возможно сжимать изображение так, чтобы наши глаза этого не замечали?

JPEG удаляет информацию, которую человеческий глаз плохо воспринимает. И формат, действительно, здорово использует особенности нашего зрения. Как известно, в глазу человека есть два типа восприимчивых к свету клеток - палочки и колбочки. Палочки в основном отвечают за свет и часто нужны при низкой освещенности, а за распознавания цвета уже нужны колбочки. У них есть соответствующие рецепторы. Именно они обеспечивают глазам цветное зрение. Но фишка в том, что их относительно мало: 6 - 7 миллионов штук, в то время как палочек намного больше: 100 - 120 миллионов. Как следствие, глаз намного сильнее восприимчив к изменению яркости изображения, чем к изменению его цвета. Поэтому, при сжатии изображений можно использовать следующую хитрость. Во-первых, мы отделяем цвет от яркости, а во-вторых, ту информацию, которая отвечает за цвет, мы можем немножечко ужать, и, скорее всего, взгляд ничего не заметит.

Именно на этом принципе основан первый важный этап сжатия. Он называется - цветовая субдискредитация.

Но как именно нам отделить яркость от цвета? Каждый пиксель кодируется тремя компонентами RGB, по одной переменной для каждого цвета, которые могут принимать значения в диапазоне от 0 до 255. Но суть в том, что эту же информацию мы можем сохранить при помощи других трех переменных: Y-Cb-Cr. Поэтому, мы можем без потерь преобразовать изображение из формата RGB в формат Y-Cb-Cr. Здесь Y отвечает за яркость, а Cb и Cr - это цветовые каналы. Таким образом, вычислив эти переменные, мы получим три новых изображения. Теперь будем использовать особенность нашего зрения, которая замечает яркости, но не так сильно замечает цвета. В обоих цветовых каналах - Cb и Cr, каждые 4 пикселя объединяем в один, по сути, снижаем разрешение этих каналов. Обе цветовые составляющие уменьшаются в 4 раза по сравнению с исходным размером, а яркость не трогаем.

Таким образом, на этом шаге размер изображения уменьшился вдвое, поскольку две из трех компонент уменьшились в четыре раза. Но здесь мы сжали в два раза, а как вы помните надо в 40. Поэтому, переходим к самому сложному и умному этапу. Для этого, изображение разбивается на квадратные блоки 8 на 8 пикселей. И теперь, алгоритмы должны каким-то образом понять, насколько много деталей в каждом из таких блоков имеется. Если деталей мало, то можно закодировать меньшим количеством бит, уменьшая размер файла, а если деталей много, то наоборот.

Иными словами, если бы картины в музеях хранились в JPEG, то любая картина ван Гога занимала бы намного больше места, чем черный квадрат Малевича. Этот анализ происходит при помощи штуки со страшным названием - дискретное косинусное преобразование. Главная философия вот такая: может быть, в детстве вы играли в игру со стеклянными стеклышками, накладывая одно на другое, можно получить третий цвет. Такой же подход используется и в JPEG, только вместо цветов – паттерны. Вот наш блок 8 на 8 пикселей. Он содержит 64 точки, и, оказывается, что любое простое монохромное изображение мы можем представить в виде комбинации базовых картинок. Всего 64 штуки. То есть, если накладывать их друг на друга с разной степенью прозрачности, можно получить любую простую картинку разрешением 8 на 8, примерно, как с цветными стеклышками.

Именно таким образом мы можем перестроить наше изображение. Единственное, что нужно знать - это в какой пропорции смешивать эти паттерны. Каждый из узоров умножается на число, которое говорит насколько интенсивно такой узор должен присутствовать в картинке. И суть алгоритма со страшным названием - вычисление этих чисел. И на выходе он выдает матрицу 8 на 8, которая описывает вклад каждого рисунка. Но, на этом не вся магия. Оказывается, коэффициенты, находящиеся в левой верхней части матрицы, отвечают за плавные переходы и более общие черты рисунка, а коэффициенты, лежащие в правой нижней

части - за частые переходы, то есть более тонкие детали и линии.

Если детали в картинке мало, то коэффициенты в правой нижней части будут близки к нулю. И, напомню, что на данном этапе мы пока не отбросили никакой информации, а просто разложили нашу картинку на слои. Но главное, мы поняли от каких из этих слоёв потенциально можно избавиться. Как же это сделать?

Делим числа исходной матрицы на числа из матрицы квантования, и здесь, как можно догадаться, выживает сильнейший. Если коэффициент какого-то из узоров слишком маленький, его в итоговый JPEG попросту не возьмут. То есть, данные, отвечающие за четкие детали, отбрасываются прежде всего. Им нужно быть очень выразительными, чтобы сохраниться. В общем, в результате деления чисел между матрицами, получается, что большая часть коэффициентов становится равна нулю после округления. Именно они отбрасываются, а всё остальное остается в картинке. Как раз на этом этапе происходит самое сильное сжатие JPEG.

Дальше, для записи этих чисел в строку, используется интересный механизм в виде змейки. Здесь тоже есть несколько ухищрений, например, вместо того, чтобы писать все нули, можно просто в соответствующих местах указывать количество нулей. Это тоже будет сжатием. И, примерно здесь, основные этапы сжатия JPEG закончены. В итоге, получилась длинная последовательность цифр. Но как же теперь вывести её на экран? Нужно сделать все то же самое, только в обратном порядке. Во-первых, декодировать, с помощью алгоритмов сжатия цифр, заполнить матрицу 8 на 8, умножить каждое число на соответствующее в таблице квантования, подложить слои с узорами и получившиеся переменные Y-Cb-Cr преобразовать в RGB.

Вот такой механизм действий проделывает ваш смартфон каждый раз, когда вы просматриваете картинку в сообщениях. И, в итоге, получается создать алгоритм, не зря потратив 6 лет. В нем учитывается куча аспектов: особенности нашего зрения, сложная математика, преобразование матриц и расположение цветов в переменные.

Круто, но конечно же, у JPEG куча недостатков. Например, он не очень эффективно сжимает текст или дает кучу артефактов при повторном сжатии. Но, несмотря на почтенный возраст, до сих пор остается основным форматом в интернете. И это явно говорит о качестве заложенных подходов и технологий.

Список литературы:

1. https://ru.wikipedia.org/wiki/JPEG

2. https://hi-tech.mail.ru/news/chem-otlichayutsya-formaty-izobrazhenij/

3. https://filesreview.com/ru/info/jpeg

4. https://www.photoconverter.ru/convert/jpeg.html