



Алгоритмы сжатия изображений

Презентацию выполнил: Шмаков А.О.

Алгоритмы сжатия изображений

Легко подсчитать, что несжатое полноцветное изображение, размером 2000×1000 пикселей будет иметь размер около 6 мегабайт. Если говорить об изображениях, получаемых с профессиональных камер или сканеров высокого разрешения, то их размер может быть ещё больше. Не смотря на быстрый рост ёмкости устройств хранения, по-прежнему весьма актуальными остаются различные алгоритмы сжатия изображений. Все существующие алгоритмы можно разделить на два больших класса:



▲ Алгоритмы сжатия без потерь

▲ Алгоритмы сжатия с потерями

▲ Алгоритмы сжатия без потерь

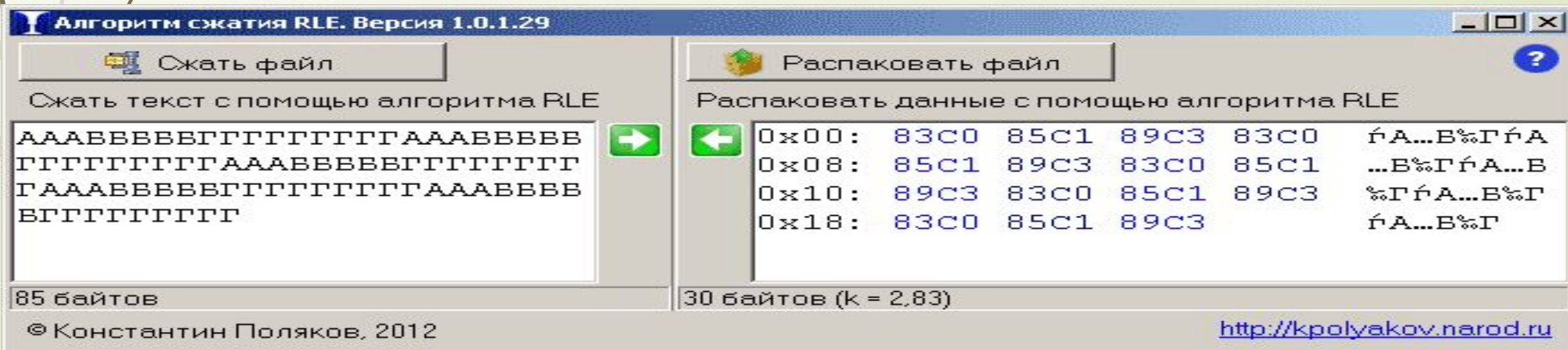
- Рассмотрим несколько разных алгоритмов сжатия изображения без потерь
(Восстанавливающие точное изображение)



Алгоритм RLE

Все алгоритмы серии RLE основаны на очень простой идее: повторяющиеся группы элементов заменяются на пару (количество повторов, повторяющийся элемент). Рассмотрим этот алгоритм на примере последовательности бит. В этой последовательности будут чередоваться группы нулей и единиц

Этот алгоритм наиболее эффективен для чёрно-белых изображений. Также он часто используется, как один из промежуточных этапов сжатия более сложных алгоритмов.



Словарные алгоритмы (сжатие без потерь)

- Идея, лежащая в основе словарных алгоритмов, заключается в том, что происходит кодирование цепочек элементов исходной последовательности. При этом кодировании используется специальный словарь, который получается на основе исходной последовательности.
- Существует целое семейство словарных алгоритмов, но мы рассмотрим наиболее распространённый алгоритм LZW, названный в честь его разработчиков Лепеля, Зива и Уэлча.
- Словарь в этом алгоритме представляет собой таблицу, которая заполняется цепочками кодирования по мере работы алгоритма. При декодировании сжатого кода словарь восстанавливается автоматически, поэтому нет необходимости передавать словарь вместе с сжатым кодом.

Данные	Символы на выходе	Строка, добавляемая в словарь и её код	
00001	К	К?	01011
00010	У	У?	01100
01011	КУ	КУ?	01101
...			

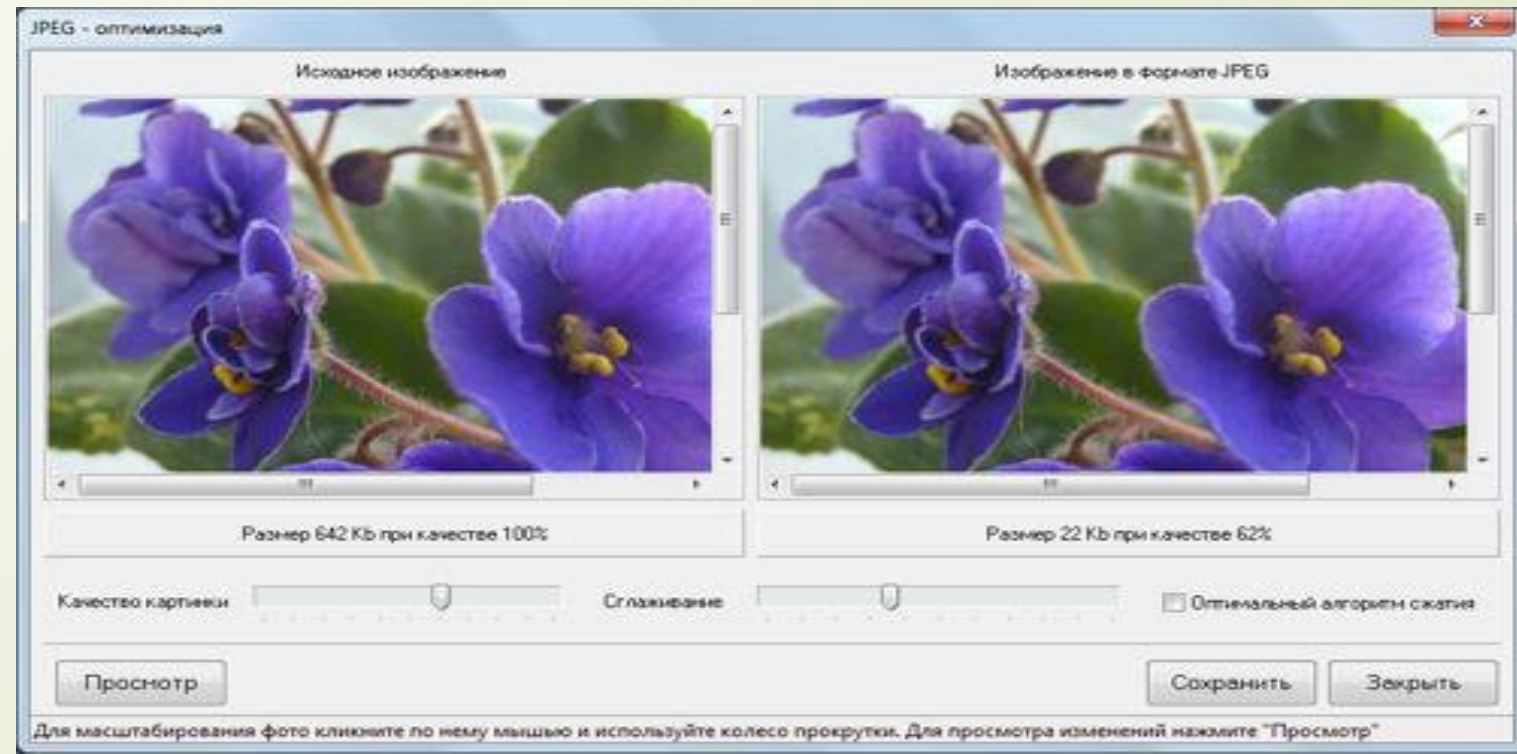
Символ	Код
К	00001
У	00010
Ш	00011
А	00100
О	00101
Н	00110
П	00111
И	01000
Л	01001
Ю	01010

Цепочка из исходной строки	Строка, добавляемая в словарь и её код		Сжатый код
К	КУ	01011	00001
У	УК	01100	00010
КУ	КУШ	01101	01011
Ш	ШК	01110	00011
К	КА	01111	00001
А	АК	10000	00100
КУ	КУК	10001	01011
КУШ	КУШО	10010	01101
О	ОН	10011	00101
Н	НК	10100	00110
КУК	КУКУ	10101	10001
У	УП	10110	00010
П	ПИ	10111	00111
И	ИЛ	11000	01000
Л	ЛА	11001	01001
АК	АКА	11010	10000
А	АП	11011	00100
П	ПЮ	11100	00111
Ю	ЮШ	11101	01010
Ш	ШО	11110	00011
ОН			10011

▲ Алгоритмы сжатия с потерями

- Рассмотрим несколько разных алгоритмов сжатия изображения с потерями

(Восстанавливающие неточное изображение)



Алгоритм сжатия JPEG

- JPEG на данный момент один из самых распространенных способов сжатия изображений с потерями. Опишем основные шаги, лежащие в основе этого алгоритма. Будем считать, что на вход алгоритма сжатия поступает изображение с глубиной цвета 24 бита на пиксел (изображение представлено в цветовой модели RGB).

Перевод в цветовое пространство YCbCr

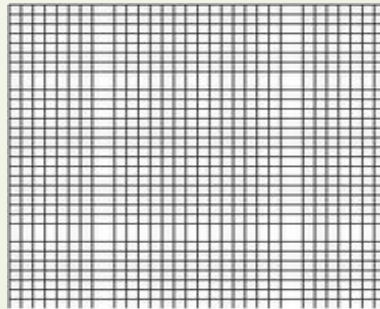
$$\begin{cases} Kg = 1 - Kr - Kb \\ Y = \min Y + (\max Y - \min Y) * (Kr * R + Kg * G + Kb * b) \\ Cb = \frac{\min C + \max C}{2} + \frac{\max C - \min C}{2} * \frac{1}{1 - Kb} * (-Kr * R - Kg * G + (1 - Kb) * B) \\ Cr = \frac{\min C + \max C}{2} + \frac{\max C - \min C}{2} * \frac{1}{1 - Kr} * ((1 - Kr) * R - Kg * G - Kb * B) \end{cases}$$

Фрактальное сжатие

□ Фрактальное сжатие — это относительно новая область. Фрактал — сложная геометрическая фигура, обладающая свойством самоподобия. Алгоритмы фрактального сжатия сейчас активно развиваются, но идеи, лежащие в их основе можно описать следующей последовательностью действий.

Процесс сжатия:

Разделение изображения на неперекрывающиеся области (домены). Набор доменов должен покрывать всё изображение полностью.



1



2



3



6



5



4