

УДК 004.627

ДОПОЛНИТЕЛЬНОЕ СЖАТИЕ ФАЙЛОВ ФОРМАТА JPEG БЕЗ ПОТЕРЬ ПРИ ПОМОЩИ КОДА ХАФФМАНА С КОНТЕКСТНЫМ МОДЕЛИРОВАНИЕМ

Брильянтов В.А. (магистрант, Университет ИТМО)

Научный руководитель – Беляев Е.А.

(к.т.н., вед. н.с., Университет ИТМО)

В данной работе рассматривается алгоритм кодирования изображений формата JPEG, и предлагается алгоритм дополнительного сжатия данного формата без потерь.

Введение. В современном мире большое количество людей все чаще создают новые изображения, как правило, используя для этого смартфоны и переносные фотоаппараты, часто использующие для хранения изображений формат JPEG. С ростом числа изображений становится все более актуальной задача об эффективном хранении их в удаленном хранилище (облаке). При этом, формат JPEG был изобретен достаточно давно и не обладает должной степенью сжатия изображений. Поэтому, для минимизации затрат на содержание облачных хранилищ, файлы данного формата необходимо дополнительно сжимать без потери качества.

Основная часть. В алгоритме JPEG используется код Хаффмана для сжатия квантованных коэффициентов дискретного косинусного преобразования (ДКП). После применения алгоритма Хаффмана в выходной файл записывается таблица кодирования Хаффмана, а также закодированные в определенном порядке коэффициенты. Для декодирования изображений формата JPEG декодер читает таблицу Хаффмана, после чего с ее помощью декодирует коэффициенты, которые потом использует для дальнейшего декодирования изображения.

В работе была предложена оптимизация сжатия, существенно уменьшающая размер исходных изображений. Для этого входные изображения частично декодируются для получения коэффициентов ДКП. Данный промежуточный этап выбран ввиду того, что ДКП невозможно осуществить без потерь, а декодирование коэффициентов ДКП не вносит потерь. После этого можно применить к полученным коэффициентам потенциально любое более эффективное кодирование.

В данной работе было выбрано контекстное кодирование, основанное на алгоритме Хаффмана. В исходном алгоритме в качестве коэффициентов ДКП выступают коэффициенты DC и AC. Коэффициенты DC кодируются с помощью кода Хаффмана, примененного к разности текущего коэффициента DC и предыдущего. Из этого можно сделать предположение, что текущее значение коэффициента DC зависит от предыдущего, поэтому эти данные предлагается моделировать Марковским источником с памятью единичного объема. Коэффициенты DC принимают значения порядка $\pm 2^{16}$, поэтому возможных разностей может быть большое число (порядка $\pm 2^{17}$). Поэтому для эффективного контекстного кодирования было предложено укрупнить контексты, разбив разности на классы эквивалентности, зависящие от числа бит, необходимых для двоичного представления коэффициента DC равномерным кодом. Таких классов оказалось не более 11, что хорошо подошло для применения контекстного кодирования. Коэффициенты AC было предложено кодировать аналогичным образом.

Кроме этого, в работе предложено дополнительно сжимать заголовки файлов JPEG и полученные дополнительные таблицы Хаффмана, зависящие от контекста. Для этого была проведена частотная статистика по некоторому числу заголовков различных изображений, и

на ее основании была построена усредненная таблица Хаффмана, хорошо сжимающая большинство заголовков и дополнительных таблиц. Также преимуществом такого подхода является то, что такую таблицу можно вообще не хранить в результирующих сжатых данных т.к. она не меняется от входного изображения. Это допущение обусловлено тем, что размер таблицы для сжатия был бы сопоставим с размером заголовков, что могло бы негативно отразиться на результатах сжатия заголовков.

Выводы. Результаты проведенного исследования являются применимыми на практике и могут быть использованы для дополнительного сжатия изображений формата JPEG, хранящихся в облачных сервисах. Для тестовых изображений при выбранном качестве входных изображений в 30% удалось достичь дополнительного сжатия в среднем на 7.88%, а при выбранном исходном качестве входных изображений 80% – на 6.9%.

Брильянтов В.А. (автор)

Подпись

Беляев Е.А. (научный руководитель)

Подпись