

Алгоритм JPEG-IT, повышающий сжатие изображений

Р.В. Голованов, А.А. Прокофьев

Национальный исследовательский университет «МИЭТ»

JPEG-IT Algorithm for Increasing Image Compression Ratio

R.V. Golovanov, A.A. Prokofiev

National Research University of Electronic Technology

Для стандартного алгоритма JPEG с постоянной таблицей Хаффмана построена модификация JPEG-IT, изменяющая этап энтропийного кодирования коэффициентов дискретного косинусного преобразования каждого блока 8×8 пикселей. На основе набора 24 оригинальных изображений из базы Kodak проведено сравнение стандартного алгоритма JPEG и модификации JPEG-IT. Как показали результаты, усредненные по всем изображениям, JPEG-IT дает выигрыш в сжатии до 20 % по сравнению со стандартным алгоритмом JPEG.

Ключевые слова: сжатие изображений; алгоритм JPEG.

The modification JPEG-IT has been developed for the baseline JPEG with fixed standard Huffman table. The proposed modification changes the stage of the entropy encoding DCT coefficients for each 8x8 pixels block. Based on a set of 24 original images from the KODAK database the comparison of a standard JPEG algorithm and the JPEG-IT modification has been performed. As the results, averaged for all images, have shown, JPEG-IT increases the image compression up to 20% compared to baseline JPEG.

Keywords: image compression, JPEG algorithm.

Общепризнанным и широко распространенным при сжатии изображений считается стандарт JPEG, принятый около 20 лет назад [1]. Проблема улучшения данного алгоритма и в настоящее время актуальна [2, 3]. Известны различные модификации кодека JPEG, которые снижают его трудоемкость, сохраняя характеристики алгоритма по получаемому качеству и сжатию изображения [4, 5]. При этом значительных сдвигов в повышении сжатия при том же качестве не наблюдается.

В стандарте JPEG описаны различные режимы работы алгоритма: baseline, sequential, progressive, optimized, arithmetic, lossless и др. Baseline – базовая версия алгоритма JPEG, получившая наибольшее распространение на практике.

В настоящей работе представлена модификация JPEG-IT (initial triangle), улучшающая характеристики алгоритма baseline JPEG, основанная на изменении схемы энтропийного кодирования низкочастотных коэффициентов дискретного косинусного преобразования (ДКП). Она позволяет повысить сжатие изображений и не требует дополнительных вычислительных ресурсов.

Baseline JPEG. Сжатие в базовом алгоритме JPEG выполняется согласно схеме на рис.1. Предварительно изображение разбивается на блоки $b_{m,n}$ размером 8×8 пикселей,

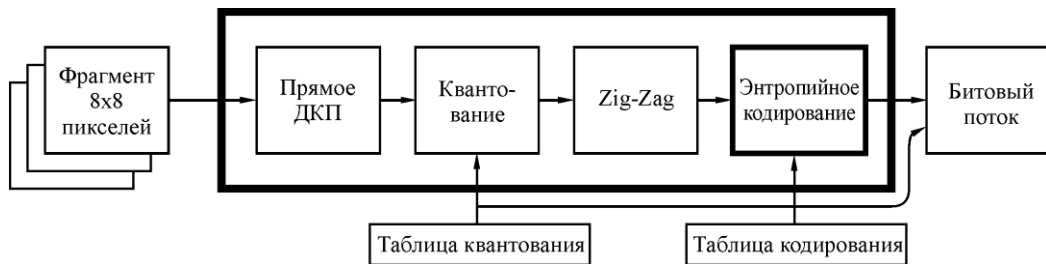


Рис.1. Схема сжатия изображения по стандарту JPEG

где $m \geq 0$ – номер строки; $n \geq 0$ – номер столбца, в которых расположен блок. Каждый блок образует матрицу $X : \{x_{i,j}\}_{i,j=0}^8$ ($x_{i,j}$ – яркость пикселя с индексами i и j внутри текущего блока), которая обрабатывается в четыре этапа.

Этап 1. Прямое ДКП. Блоки обрабатываются построчно слева направо от левого верхнего угла изображения. Внутри каждого блока выполняется ДКП и получается новая матрица Y :

$$y_{i,j} = \text{ДКП}(x_{i,j}). \quad (1)$$

Этап 2. Квантование (огрубление) коэффициентов ДКП. Цель процедуры – в обнулении или уменьшении высокочастотных коэффициентов ДКП, слабо влияющих на качество изображения, воспринимаемое человеком. Для этого используется специальная матрица квантования $Q : \{q_{i,j}\}_{i,j=0}^8$, общая для всех блоков. Матрица Y поэлементно делится на Q , что дает новую матрицу \tilde{Y} :

$$\tilde{y}_{i,j} = \text{round}\left(\frac{y_{i,j}}{pq_{i,j}}\right), \quad (2)$$

где функция round означает округление до ближайшего целого числа; p – параметр, задаваемый пользователем. В baseline JPEG используется встроенная матрица квантования, а параметр p задается пользователем перед сжатием. Он формально определяет качество получаемого изображения.

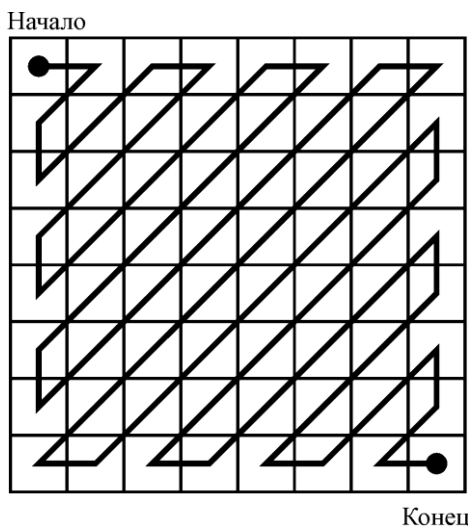


Рис.2. Схема «зигзаг» упорядочивания

Этап 3. «Зигзаг» упорядочивания. Целочисленная матрица \tilde{Y} состоит из элементов, убывающих с ростом каждого индекса. Элементы матрицы упорядочиваются в одномерный массив по правилу «зигзаг», как показано на рис.2. Сначала выписывается угловой элемент $\tilde{y}_{0,0}$, потом два элемента первой побочной диагонали, а затем элементы следующей побочной диагонали и т.д.

Закон убывания коэффициентов ДКП [6], процедура квантования (2) и специальная схема упоря-

дочивания «зигзаг» дают, как правило, невозрастающий вектор $V = \{\tilde{y}_k\}$, $\tilde{y}_k \geq \tilde{y}_{k+1}$, $0 \leq k \leq 63$.

Для элементов $\tilde{y}_0^{m,n}$ каждого блока $b_{m,n}$ проводится дифференциальная импульсно-кодированная модуляция. В текущем блоке $b_{m,n}$ вместо истинного значения $\tilde{y}_0^{m,n}$ хранится разность с \tilde{y}_0 предыдущего блока. Для соседних блоков внутри строки с номером m вычисляется разность:

$$\Delta\tilde{y}_0^{m,n} = \tilde{y}_0^{m,n} - \tilde{y}_0^{m,n-1}, m \geq 0, n > 0, \quad (3)$$

для блоков в начале строки предыдущим считается начальный блок предыдущей строки:

$$\Delta\tilde{y}_0^{m,n} = \tilde{y}_0^{m,n} - \tilde{y}_0^{m-1,n}, m \geq 0, n = 0, \quad (4)$$

а первый элемент первого блока $\tilde{y}_0^{0,0}$ остается неизменным.

Разность средних яркостей соседних блоков слабо меняется в естественных изображениях, поэтому хранить абсолютные значения \tilde{y}_0 не выгодно. Таким образом, дифференциальная импульсно-кодированная модуляция обеспечивает более экономное представление элементов \tilde{y}_0 .

Этап 4. Энтропийное кодирование. Это заключительный этап, представляющий собой кодирование векторов $V_{m,n}$ каждого блока $b_{m,n}$ в битовые последовательности и их запись в выходной поток. Блоки обрабатывают отдельно друг от друга, но при этом считается, что они подчиняются одной статистической модели. Поэтому для их кодирования используются общие кодовые таблицы. Важно отметить, что кодируются не сами элементы векторов, а их особая запись.

Первый элемент вектора $\Delta\tilde{y}_0$ кодируется по принципу VLI (variable length integer) [7]. Таким образом, получается минимальный битовый код

$$\Delta\tilde{y}_0 \xrightarrow{\text{VLI}} R_0, \quad (5)$$

который не является префиксным, т.е. может быть началом любого другого кода. Поэтому дополнительно записывается информация о длине этого кода:

$$L_0 = \text{length}(R_0), \quad (6)$$

где length – функция, подсчитывающая количество бит в коде R_0 . Эта информация необходима для корректного считывания кодов из потока данных на этапе декодирования. Число L_0 отдельно представляется префиксным кодом

$$L_0 \xrightarrow{\text{code table}} L_0^{H1}, \quad (7)$$

который берется из специально подготовленной таблицы кодовых слов $H1$. Среди оставшихся элементов y_k , $1 \leq k \leq 63$ кодируются только ненулевые так же, как и в (5):

$$\tilde{y}_k \xrightarrow{\text{VLI}} R_k, \quad (8)$$

а информация о количестве предшествующих нулевых элементов записывается в байт-пару $(Z_k, \text{length}(R_k))$:

$$P_k = Z_k \ll 4 + \text{length}(R_k), \quad (9)$$

где Z_k – число нулевых элементов, предшествующих y_k ; length – функция из (6); операция \ll – битовый сдвиг в сторону старших разрядов. Такие пары кодируются префиксными кодами аналогично (7):

$$P_k \xrightarrow{\text{code table}} P_k^{H2} \quad (10)$$

с помощью другой таблицы кодовых слов $H2$.

На практике Z_k может быть более 15, следовательно, 4 бит может оказаться недостаточно для записи этого числа. Поэтому последовательности из 15 нулевых элементов кодируются специальной парой (15,0). Если после какого-то y_k все оставшиеся элементы $y_{k+1} \dots y_{63}$ равны нулю, то они кодируются другой специальной парой (0,0) – признаком конца текущего блока $b_{m,n}$. В выходной поток записываются коды

$$b_{m,n} \xrightarrow{\text{entropy coding}} (L_0^{H1} R_0) (P_{k_1}^{H2} R_{k_1}) \dots (P_{k_d}^{H2} R_{k_d}), \quad (11)$$

где k_1, \dots, k_d – индексы ненулевых \tilde{y}_k .

В baseline JPEG на этапе энтропийного кодирования используется алгоритм Хаффмана. Существуют и другие альтернативы кодирования, например арифметическое кодирование, но они более ресурсоемки, хотя в некоторых случаях дают большее сжатие.

Возможны два варианта кодирования – с использованием фиксированных кодовых таблиц или оптимальных. Кодовые таблицы $H1$ и $H2$ из (7) и (10) формируются на основе статистического анализа сжимаемых данных. Как показывает практика, такой анализ может занимать до половины времени всего кодирования [8]. Поэтому пользуются встроенными таблицами Хаффмана, оптимизированными для конкретного уровня качества p из (2), что значительно экономит временные ресурсы. Однако для других значений p эти таблицы оказываются неоптимальными, и, как следствие, получается не лучшее сжатие.

В стандарте JPEG предлагается использовать встроенные таблицы Хаффмана для (7) и (10). На примере этих таблиц строится модификация JPEG-IT.

Алгоритм JPEG-IT. В ходе исследования найден способ улучшения характеристики базового кодека JPEG, использующего встроенные таблицы Хаффмана. Если пользователь требует высокого качества от сжатого изображения, то квантование (2) будет слабым и большинство элементов вектора $V_{m,n}$ во всех блоках $b_{m,n}$ будут ненулевыми. Почти всегда несколько первых элементов этого вектора будут ненулевыми, поэтому стандартная схема кодирования (8)–(10) будет избыточной.

Предлагается изменить схему кодирования так, чтобы несколько первых элементов $\tilde{y}_k \neq 0, 1 \leq k < t \leq 63$ кодировались так же, как и $\Delta \tilde{y}_0$ согласно (5)–(7). В этом случае в выходной поток, по аналогии с (11), записываются следующие коды:

$$b_{m,n} \xrightarrow{\text{entropy coding}} (L_0^{H1} R_0) (L_1^{H1} R_1) \dots (L_t^{H1} R_t) (P_{k_1}^{H2} R_{k_1}) \dots (P_{k_d}^{H2} R_{k_d}), \quad (12)$$

где $0 < t < k_1$ – индексы первых ненулевых элементов вектора $V_{m,n}$.

Первые ненулевые элементы \tilde{y}_k образуют начальный треугольник исходной матрицы \tilde{Y} (см. рис.2). Отсюда и предложено название модификации JPEG-IT – initial triangle. Такое изменение позволяет экономить до нескольких бит на блок.

При сильном квантовании (при низком качестве получаемого изображения) встречаются блоки, в которых вектор $V_{m,n}$ состоит только из нулевых элементов, быть может

за исключением $\Delta\tilde{y}_0$. В этом случае в выходной поток записывается код Хаффмана для специальной пары (0,0) – признака конца блока. Стандартная таблица Хаффмана кодирует эту пару четырьмя битами [1010]. Если $\forall k, 1 \leq k \leq 63: \tilde{y}_k = 0$, то в выходной поток записываются коды

$$b_{m,n} \xrightarrow{\text{entropy coding}} (L_0^{H1} R_0) [1010], \quad (13)$$

в противном случае, когда встречается несколько ненулевых \tilde{y}_k , записываются

$$b_{m,n} \xrightarrow{\text{entropy coding}} (L_0^{H1} R_0) (P_{k_1}^{H2} R_{k_1}) \dots (P_{k_d}^{H2} R_{k_d}) [1010], \quad (14)$$

где k_1, \dots, k_d – индексы ненулевых \tilde{y}_k .

Предлагается использовать только 1 бит вместо 4 для обозначения конца блока. Если $\tilde{y}_k = 0, 1 \leq k \leq 63$, то записывается «0»:

$$b_{m,n} \xrightarrow{\text{entropy coding}} (L_0^{H1} R_0) [0], \quad (15)$$

в противном случае – «1», а оставшиеся элементы кодируются по стандартной схеме:

$$b_{m,n} \xrightarrow{\text{entropy coding}} (L_0^{H1} R_0) [1] (P_{k_1}^{H2} R_{k_1}) \dots (P_{k_d}^{H2} R_{k_d}) [1010]. \quad (16)$$

Схема кодирования (15) и (16) позволяет экономить до 3 бит на блок по сравнению с (13) и (14).

Оба предложения (12) и (15), (16) дополняют друг друга и являются эффективной модификацией базового кодека JPEG.

Методика сравнения. Для проведения расчетов выбрана общедоступная реализация алгоритма JPEG [9] на языке C++. Описанные модификации реализованы в этом коде. Модификация JPEG-IT затрагивает только этап энтропийного кодирования кодека baseline JPEG, что не влияет на качество сжатого изображения. Поэтому изображения, получаемые кодеками baseline JPEG и JPEG-IT, идентичны по качеству, но отличаются степенью сжатия.

Другие режимы работы кодека JPEG (progressive, optimized, arithmetic, lossless) являются надстройками над baseline JPEG. Поэтому при сравнении этих режимов работы кодека с JPEG-IT будет получен такой же результат, как и для baseline JPEG, но с учетом выигрыша в сжатии, который получен в ходе экспериментов.

Сравнение проводилось на 24 разнородных черно-белых изображениях из базы Kodak [10]. Каждое изображение базы представлено в четырех размерах: 0,4; 0,6; 1,6; 6,3 Мп. Для оценки качества сжатых изображений использованы специальные метрики, адекватные человеческому восприятию. Известны примерно 20 альтернативных математических критериев оценки качества искаженного изображения, в том числе и широко распространенный критерий пиковое отношение сигнал-шум, не учитывающий особенностей человеческого зрения. Как следует из результатов предыдущих работ [11], для оценки качества изображения наилучшим является математический критерий VIF [12].

В ходе эксперимента проведено кодирование тестовых изображений кодеками baseline JPEG и JPEG-IT. Пользовательский параметр качества p менялся во всем диапазоне допустимых значений. Для каждого тестового изображения вычислялись степень сжатия, как отношение размера исходного файла к сжатому, и качество по критерию VIF.

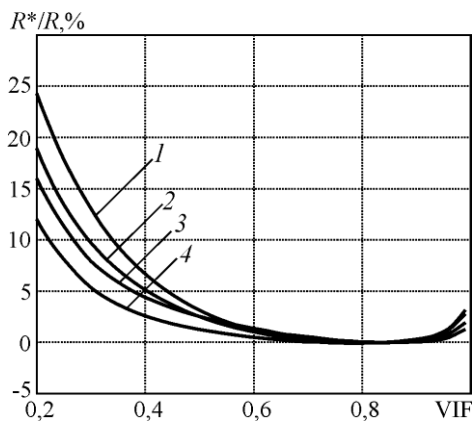


Рис.3. Средний выигрыш в сжатии JPEG-IT по 24 изображениям: кривые 1–4 соответствуют размерам изображений, равным 6,3; 1,6; 0,4; 0,2 Мп

Анализ результатов. Основные результаты приведены на рис.3. По оси абсцисс отложено качество изображения по критерию VIF в линейном масштабе. Значения критерия VIF соответствуют условным субъективным оценкам качества: плохое [0;0,4), посредственное [0,4; 0,6), хорошее [0,6; 0,8) и отличное [0,8; 1,0]. На рис.4 приведены примеры сжатых изображений для каждого диапазона качества изображений. На рис.3 эти диапазоны отделены вертикальными пунктирами. По оси ординат отложен выигрыш в сжатии R^* алгоритма JPEG-IT по сравнению с сжатием R стандартного JPEG. Каждая кривая соответствует среднему выигрышу в сжатии изображений одинакового размера.

Результат сжатия изображения, представленный на рис.4, отображен в таблице. Для каждого уровня качества приведены битовые затраты на пиксель при использовании кодека JPEG и JPEG-IT. В последней строке таблицы приведен процентный прирост в сжатии предложенной модификации против стандартного кодека. Выигрыш меняется от ~0,4 до 20,4% в зависимости от уровня качества сжатого изображения.



Рис.4. Изображение из базы Kodak после компрессии для разных уровней качества VIF (построчно слева направо : 0,11; 0,21; 0,31; 0,46; 0,68; 0,99)

Битовые затраты на пиксель изображений (см. рис.4)

Название кодека	Уровень качества VIF					
	0,11	0,21	0,31	0,46	0,68	0,99
Baseline JPEG	0,159	0,218	0,349	0,541	0,939	3,46
JPEG-IT	0,132	0,197	0,332	0,529	0,935	3,30
Выигрыш, %	20,4	10,6	5,12	2,26	0,427	4,85

Исходя из данных рис.3 и таблицы можно сделать следующие выводы:

- в случае применения предложенных модификаций выигрыш в сжатии имеется во всем диапазоне качества. При хорошем качестве он составляет ~0,5%, при высоком – до ~4,5%, а при среднем и плохом возрастает до ~20% в зависимости от размера изображения;

- при работе с изображениями размером от 0,2 до 6,3 Мп выигрыш в сжатии меняется от 2 до 4% в диапазоне среднего качества, от 5 до 20% в диапазоне плохого качества соответственно;

- при использовании других таблиц Хаффмана, оптимизированных для другого уровня качества, форма кривых на рис.3 сохранится. При этом минимум сместится в область качества, для которой проводилась оптимизация.

Таким образом, модификация JPEG-IT дает выигрыш в сжатии до 20% по сравнению с базовым кодеком JPEG в зависимости от требуемого качества и исходного размера изображения. Модификация дает выигрыш в сжатии как на изображениях с мелкими деталями, так и на фоновых изображениях.

Модификация не увеличивает трудоемкость базового алгоритма JPEG при компрессии и декомпрессии изображений. Поэтому данное изменение можно внедрять в существующие программные и аппаратные модули сжатия изображения на основе JPEG.

Возможен перенос модификации JPEG-IT в видеокodeки на основе JPEG, например Motion-JPEG.

Авторы выражают благодарность Н.Н. Калиткину за ценные замечания и советы при обсуждении работы.

Работа выполнена при финансовой поддержке РФФИ (грант №14 – 01 – 00161).

Литература

1. **Wallace G.K.** The JPEG algorithm for image compression standard // Communications of the ACM. – 1991. – Vol. 34, N 4. – P. 30–44.
2. **Умняшкин С.В., Космач М.В.** Оптимизация кодирования цифровых изображений по методу JPEG // Изв. вузов. Электроника. – 2000. – №4–5. – С. 139–141.
3. **Умняшкин С. В.** Использование контекстного арифметического кодирования для повышения сжатия данных по схеме JPEG // Изв. вузов. Электроника. – 2001. – №3. – С. 96–99.
4. **Умняшкин С.В., Курина В.В.** Алгоритм сжатия изображений на основе дискретного псевдокосинусного преобразования // Цифровая обработка сигналов. – 2009. – №3. – С. 2–7.
5. **Jie Liang, Trac D.** Fast multiplierless approximations of the DCT with the lifting scheme // IEEE Transactions on Signal Processing. – 2001. – Vol. 49, N 12. – P. 3032–3044.
6. **Голованов Р.В., Калиткин Н.Н.** Продолжения повышенной гладкости для фурье-аппроксимации непериодической функции // Доклады Академии наук. – 2012. – Т. 446, № 1. – С. 25–29.
7. **Fenwick P.** Punctured elias codes for variable-length coding of the integers // Technical Report 137. Department of Computer Science. The University of Auckland. – 1996.
8. **Беляев И.А.** СФ-блок кодирования Хаффмана для сжатия изображений по стандарту JPEG // Проблемы разработки перспективных микро- и наноэлектронных систем – 2010: сб. трудов / Под общ. ред. А.Л. Стемповского. – М.:ИППМ РАН, 2010. – С. 332–335.
9. JPEG C++ source code // URL: <http://www.infai.org/jpeg/> (дата обращения: 09.02.2014).
10. Kodak lossless true color image suite // URL: <http://r0k.us/graphics/kodak/> (дата обращения: 7.05.2010).

11. *Голованов Р.В., Калиткин Н.Н.* Статистическая недостоверность распространенных критериев оценки качества искаженного изображения // *Цифровая обработка сигналов.* – 2013. – №3. – С. 74–79.

12. *Sheikh H. R., Bovik A. C.* Image information and visual quality // *IEEE Transactions on Image Processing.* – 2006. – Vol. 15. – P. 430–444.

Статья поступила
после доработки 2 июня 2014 г.

Голованов Роман Вячеславович – аспирант кафедры высшей математики–1 МИЭТ. *Область научных интересов:* цифровая обработка изображений, численные методы. **E-mail:** golovanovrv@gmail.com

Прокофьев Александр Александрович – доктор педагогических наук, кандидат физико-математических наук, доцент, заведующий кафедрой высшей математики–1 МИЭТ. *Область научных интересов:* цифровая обработка сигналов и изображений, математическое моделирование.

**Информация для читателей журнала
«Известия высших учебных заведений. Электроника»**

С тематическими указателями статей за 1996 - 2014 гг., аннотациями и содержанием последних номеров на русском и английском языках можно ознакомиться на нашем сайте:

<http://www.miet.ru>