

АЛГОРИТМ СЖАТИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ДИСКРЕТНОГО ПСЕВДОКОСИНУСНОГО ПРЕОБРАЗОВАНИЯ

Умняшкин С.В., д.ф.-м.н., профессор Московского государственного института электронной техники (технический университет)

Курина В.В., Московский государственный институт электронной техники (технический университет), valya_kurina@rambler.ru

Ключевые слова: сжатие изображений, алгоритм, дискретное преобразование, матрица, квантование, кодирование.

Введение

В настоящее время известно множество быстрых алгоритмов вычисления дискретного косинусного преобразования (ДКП), разработанных для различных задач сжатия изображений и видео. Они позволяют сократить количество операций умножения, необходимых для выполнения преобразований, но, тем не менее, эта операция не может быть полностью исключена [1]. Если требуется выполнять вычисления без операций умножения, можно воспользоваться преобразованиями, приближенными к ДКП, например [2, 3].

В данной работе мы предлагаем использовать для сжатия изображений дискретное псевдокосинусное преобразование [4] – более простую в вычислении альтернативу ДКП, не требующую операций умножения. Алгоритм сжатия изображений на основе такого преобразования может быть востребован в устройствах с ограниченными аппаратными ресурсами, например, карманных компьютерах или мобильных телефонах.

Дискретное псевдокосинусное преобразование

Матрица дискретного псевдокосинусного преобразования (ДПКП) \mathbf{W} может быть представлена в виде произведения двух матриц специального вида: $\mathbf{W} = \mathbf{D} \cdot \mathbf{C}$, где \mathbf{D} – диагональная нормировочная матрица, а \mathbf{C} – матрица, допускающая факторизованное представление в виде слабозаполненных матриц.

Так, матрица восьмиточечного ДПКП имеет вид [4]: $\mathbf{W}_8 = \mathbf{D}_8 \cdot \mathbf{C}_8$, где

$$\mathbf{D}_8 = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{5}} \right),$$

$$\mathbf{C}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 & 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 & 1 & -2 & 2 & -1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \end{bmatrix}.$$

При этом, за счет использования специального факторизованного представления, умножение на матрицу \mathbf{C}_8 сводится к выполнению 28 операций сложения чисел [4].

В общем случае, при других размерностях ДПКП, реализация умножения на матрицу \mathbf{C} также сводится только к

Предлагается алгоритм сжатия статических полутоновых изображений, который основан на быстром целочисленном преобразовании, полученном в результате модификации дискретного косинусного преобразования.

выполнению операций сложения. ДПКП, так же, как и ДКП, является ортогональным преобразованием.

Для оценки эффективности применения ортогональных преобразований с целью сжатия сигнала можно воспользоваться величиной *coding gain* (коэффициент эффективности кодирования) [5]:

$$C_g = 10 \lg \frac{\frac{1}{N} \sum_{k=0}^{N-1} \sigma_k^2}{\left(\prod_{k=0}^{N-1} \sigma_k^2 \right)^{\frac{1}{N}}}, \quad (1)$$

где σ_k^2 – дисперсия k -го коэффициента преобразования.

Значения коэффициента C_g для ДПУА – дискретного преобразования Уолша-Адамара, ДПКП, ДКП и ДПКЛ – дискретного преобразования Карунена-Лозва приведены в Таблице 1. Расчеты выполнены для входного сигнала ($N = 8$), имеющего статистику марковского процесса первого порядка с коэффициентом корреляции соседних отсчетов $\rho = 0,95$.

Таблица 1.

Преобразование	ДПУА	ДПКП	ДКП	ДПКЛ
C_g	7,9461	8,3251	8,8259	8,8462

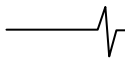
Таким образом, с одной стороны ДПКП не намного уступает ДКП по эффективности применения к сжатию данных, с другой – требует меньших вычислительных затрат, так как не использует операции умножения.

Базовый алгоритм сжатия

Предлагаемая нами базовая «JPEG-подобная» схема компрессии статического полутонного изображения приведена на рис 1. В процессе декомпрессии последовательно выполняются операции арифметического декодирования, скалярного деквантования и двумерного обратного ДПКП. Исходное изображение разбивается на блоки размера 8×8 пикселей. На этапе вычисления двумерного ДПКП сначала строки каждого блока изображения, затем столбцы полученной матрицы умножаются на матрицу \mathbf{C}_8 . Равномерное скалярное квантование спектрального коэффициента ДПКП w объединяется с умножением на матрицу $\mathbf{D}_8 = \{d_{i,j}\}$ благодаря введению специального шага квантования

$$q_{i,j} = \frac{q}{d_{i,i} d_{j,j}},$$

где индексы i и j определяются положением квантуемого коэффициента в блоке. Квантование выполняется в со-



ответствии с правилом:

$$\hat{w} = \text{Round}(w/q_{i,j}), \quad (2)$$

где функция Round производит округление до ближайшего целого числа. Деквантование значения \hat{w} выполняется по правилу:

$$\tilde{w} = q_{i,j} \hat{w}. \quad (3)$$

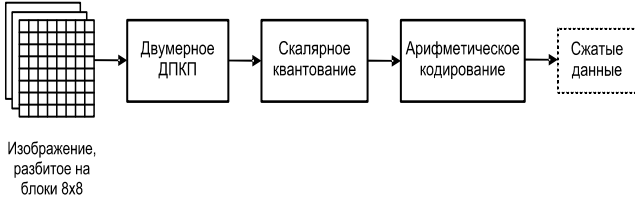


Рис. 1. Схема компрессии

Пусть $w_{i,j}^{(l)}$ – коэффициент с номером $l = 0, \dots, 63$ в блоке с координатами (i, j) (начало координат – блок в левом верхнем углу). Нумерация внутри блока – от левого верхнего угла по строкам. Поток данных из постоянных составляющих каждого блока (коэффициентов $w_{i,j}^{(0)}$) после квантования подвергается двумерной дифференциальной импульсно-кодовой модуляции (ДИКМ). Вычисление разности двумерной ДИКМ можно осуществить с помощью операций сложения и побитового сдвига в соответствии со следующим соотношением:

$$d_{i,j} = \hat{w}_{i,j}^{(0)} - \frac{1}{2} (2\hat{w}_{i-1,j}^{(0)} + 2\hat{w}_{i,j-1}^{(0)} - \hat{w}_{i-1,j-1}^{(0)} - \hat{w}_{i-1,j+1}^{(0)}). \quad (4)$$

Для реализации арифметического кодирования был выбран адаптивный кодер [6, 7]. Характерной особенностью этого кодера является использование целочисленной арифметики, однако в процессе вычислений требуется выполнение операций умножения и деления. В случае необходимости дополнительного сокращения (устранения) операций умножения, можно воспользоваться реализациями арифметических кодеров из работ [8, 9], в которых вычисления выполняются с помощью операций сложения и побитового сдвига, но при этом снижается эффективность кодирования [8].

Спектральные коэффициенты ДПКП в каждом блоке распределяются на восемь зон, кодируемых с помощью разных статистических моделей распределения вероятностей (рис. 2а). Зоны выбирались таким образом, чтобы среднеквадратичные отклонения (СКО) коэффициентов были близки, и наиболее коррелированные коэффициенты входили в одну зону. Данные о СКО спектральных коэффициентов приведены на рис. 2б. С целью сокращения объема алфавитов кодирования моделей предлагается кодировать модули коэффициентов, а знаки ненулевых коэффициентов записывать в отдельный поток с затратами 1 бит на символ.

Зону, в которой все коэффициенты равны нулю, будем называть *нулевой*. Для описания нулевых зон каждому блоку поставим в соответствие восьмибитовое число – *ключ блока*. Обозначим через $k_{i,j}$ значение ключа блока с координатами (i, j) , $k_{i,j}(r)$ – значение r -го бита числа $k_{i,j}$, $Z_{i,j}^r$ – множество проквантованных спектральных коэффициентов, образующих зону с номером r в блоке с координатами (i, j) , тогда

$$k_{i,j}(r) = \begin{cases} 0, & \text{если } \forall \hat{w}_{i,j}^{(l)} \in Z_{i,j}^r : \hat{w}_{i,j}^{(l)} = 0, \\ 1, & \text{если } \exists \hat{w}_{i,j}^{(l)} \in Z_{i,j}^r : \hat{w}_{i,j}^{(l)} \neq 0. \end{cases} \quad (5)$$

Таким образом, на этапе адаптивного арифметического кодирования формируются и кодируются независимо следующие потоки данных: поток ключей блоков (кодируемый с помощью 1 статистической модели), поток модулей разностей от двумерной ДИКМ постоянных составляющих блоков (1 модель), поток модулей спектральных коэффициентов из ненулевых зон (8 моделей).

Алгоритм обработки блока спектральных коэффициентов с координатами (i, j) на этапе арифметического кодирования состоит из следующих шагов:

1. Арифметическое кодирование постоянной составляющей. Вычисление разности $d_{i,j}$ по формуле (4). Абсолютное значение разности $|d_{i,j}|$ кодируется в отдельном потоке. Если $d_{i,j} \neq 0$, то знак $s_{i,j} = \text{sign}(d_{i,j})$ (1 бит) записывается в поток знаков.

	0	3	3	0	0	3	3
1	5	5	7	2	6	6	7
4	5	7	7	6	6	7	7
4	7	7	7	7	7	7	7
1	2	6	7	2	2	5	7
1	6	6	7	2	2	5	7
4	6	7	7	5	5	7	7
4	7	7	7	7	7	7	7

а)

	23,3	8,41	4,55	44,9	23,0	8,86	4,25
14,2	5,43	3,93	2,34	9,01	6,87	3,79	2,45
3,97	3,47	3,10	1,96	3,93	3,87	2,81	1,98
2,73	1,91	1,76	1,45	2,20	1,97	1,71	1,49
27,8	9,92	6,05	3,08	19,0	13,6	6,52	3,37
12,2	7,23	5,30	2,74	11,9	10,8	5,37	2,81
4,61	3,45	2,92	1,89	4,71	4,46	2,60	1,96
2,30	1,96	1,86	1,54	2,17	2,06	1,70	1,56

б)

Рис. 2. Расположение зон в блоке (а) и СКО коэффициентов блока двумерного ДПКП (б)



2. Формирование ключа блока $k_{i,j}$ по правилу (5) и его арифметическое кодирование.

3. Арифметическое кодирование спектральных коэффициентов из ненулевых зон. Просматриваем последовательно все коэффициенты блока (за исключением постоянной составляющей), $l=1, \dots, 63$. Пусть $\widehat{w}_{i,j}^{(l)} \in Z_{i,j}^r$, тогда, если $k_{i,j}(r)=1$, закодировать $|\widehat{w}_{i,j}^{(l)}|$, используя модель с номером r , и, если $\widehat{w}_{i,j}^{(l)} \neq 0$, то записать $s_{i,j}^{(l)} = \text{sign}(\widehat{w}_{i,j}^{(l)})$ в файл знаков. Если $k_{i,j}(r)=0$, то коэффициент $\widehat{w}_{i,j}^{(l)}$ не обрабатывается.

Модификация базового алгоритма

Квантование с нулевой зоной

В базовом варианте алгоритма сжатия выполнялось равномерное скалярное квантование с порогами $\{\dots, -2,5q, -1,5q, -0,5q, 0,5q, 1,5q, 2,5q, \dots\}$ и уровнями – в середине интервалов. То есть спектральные коэффициенты $w_{i,j}^{(l)}$, такие, что $-0,5q < w_{i,j}^{(l)} \leq 0,5q$, после квантования принимали значение, равное 0.

При квантовании с нулевой зоной (dead zone quantization) [10] изменяется интервал, соответствующий нулевому уровню. Пороги квантования принимают вид: $\{\dots -3kq, -2kq, -kq, kq, 2kq, 3kq, \dots\}$, а уровни квантования вычисляются как среднее арифметическое соседних порогов. Значение параметра k выбирается так, чтобы большая часть высокочастотных компонент квантовалась в 0. Такое квантование используется, например, в стандартах JPEG2000, MPEG-2, H.263.

Результаты обработки нескольких тестовых изображений для нашей схемы компрессии показали, что оптимальным является значение $k=1$. Таким образом, в окончательном варианте квантование выполнялось с порогами $\{\dots, -3q, -2q, -q, q, 2q, 3q, \dots\}$.

Многомодельное арифметическое кодирование

Как видно из рисунка 2б, в зонах коэффициентов, кодируемых при помощи одной и той же статистической модели, все же присутствуют коэффициенты, имеющие заметно различающиеся СКО. Для более точного моделирования распределения вероятностей коэффициентов каждой зоны можно использовать не одну, а несколько моделей, выбирая необходимую по «контексту», то есть на основании некоторого правила, использующего результаты предшествующей работы арифметического кодера.

В нашей схеме компрессии мы использовали по две модели для кодирования проквантованных коэффициентов каждой зоны рис. 2а. Правило выбора модели кодирования для текущего обрабатываемого коэффициента $\widehat{w}_{i,j}^{(l)}$ строилось следующим образом:

$$\text{Model}(\widehat{w}_{i,j}^{(l)}) = \begin{cases} 2r, & \text{Prog}(\widehat{w}_{i,j}^{(l)}) \leq t_r \\ 2r+1, & \text{Prog}(\widehat{w}_{i,j}^{(l)}) > t_r \end{cases} \quad (6)$$

Здесь r – номер зоны, в которой содержится коэффициент $\widehat{w}_{i,j}^{(l)}$, t_r – порог для выбора модели кодирования, соответствующий зоне r , $\text{Prog}(\widehat{w}_{i,j}^{(l)})$ – прогноз, представляющий собой некоторую функцию от уже обработанных «контекстных» коэффициентов ДПКП.

В алгоритмах сжатия изображений на базе ДКП возможны различные способы выбора контекста: на осно-

вании внутривекторной [11, 12] и межвекторной [13, 14] взаимной корреляции коэффициентов спектра. В алгоритме на основе ДПКП мы ограничились построением прогноза по коэффициентам из соседних с кодируемым блоком, которые соответствуют одинаковым базисным функциям:

$$\text{Prog}(\widehat{w}_{i,j}^{(l)}) = \frac{1}{6} \left(2|\widehat{w}_{i-1,j}^{(l)}| + 2|\widehat{w}_{i,j-1}^{(l)}| + |\widehat{w}_{i-1,j-1}^{(l)}| + |\widehat{w}_{i-1,j+1}^{(l)}| \right). \quad (7)$$

Весовые коэффициенты в данной формуле учитывают тот факт, что корреляция кодируемого коэффициента с соответствующими коэффициентами из блоков сверху и слева от текущего блока (i, j) больше, чем с коэффициентами из «диагональных» блоков $(i-1, j-1)$ и $(i-1, j+1)$. Заметим, что при реализации кодера деление на 6 можно опустить, увеличив в 6 раз значения порогов $\{t_r\}$ в решающем правиле (6). Таким образом, вычисление прогноза (7) вновь сводится только к выполнению операций сложения.

По результатам обработки широко распространенных тестовых изображений «Lena», «Barbara» и «Goldhill» для различных значений шага квантования q были эмпирически найдены оптимальные величины порогов $\{t_r\}$ и установлено, что их значения достаточно хорошо описываются формулой:

$$t_r = c_r / q, \quad (8)$$

где c_r – постоянные масштабные коэффициенты, определенные на основании экспериментальных данных по методу наименьших квадратов. Таким образом, пороги для правила (6) являются варьируемыми и выбираются по значению используемой величины шага равномерно квантования q при помощи формулы (8), а набор значений $\{c_r\}$ является фиксированным.

RD-оптимизация

Дополнительно повысить характеристики алгоритма сжатия можно, воспользовавшись идеями теории «битовые затраты - искажение» (rate-distortion theory, RD theory) [15].

Поставим следующую задачу RD-оптимизации: необходимо найти такой набор ключей блоков $\mathbf{k}^* = \{k_{11}^*, k_{12}^*, \dots, k_{nm}^*\}$ и такую величину шага квантования q^* , которые бы обеспечивали минимум функции Лагранжа:

$$L(\mathbf{k}^*, q^*) = \min_{\mathbf{k}, q} (L(\mathbf{k}, q) = D(\mathbf{k}, q) + \lambda R(\mathbf{k}, q)). \quad (9)$$

Здесь \mathbf{k} – всевозможные наборы ключей блоков; λ – задаваемый внешне неотрицательный параметр, который определяет соотношение качества изображения и битовых затрат на его кодирование;

$D = \sum_{i,j,l} (w_{i,j}^{(l)} - \widehat{w}_{i,j}^{(l)})^2$ – квадратичная ошибка, а R –

битовые затраты, которые связаны с кодированием изображения.

Найденные в процессе RD-оптимизации значения q^* и \mathbf{k}^* далее используются при квантовании и арифметическом кодировании. Схема компрессии с использованием RD-оптимизации приведена на рис. 3.

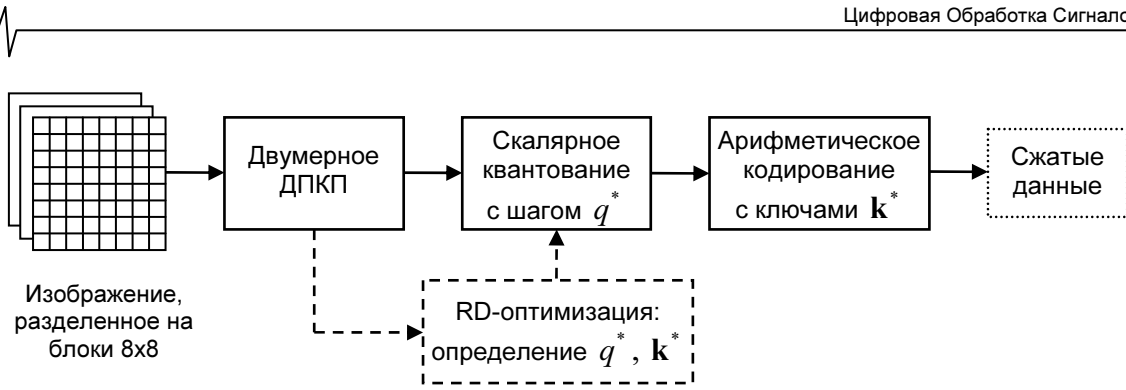


Рис. 3. Схема компрессии с RD-оптимизацией

Так как в алгоритме сжатия используется адаптивное арифметическое кодирование [6, 7], то для правильной оценки битовых затрат в процессе оптимизации требуется обновление внутренних гистограмм каждой модели без реального кодирования поступающих символов, то есть без непосредственного формирования выходного битового кода. Такую процедуру будем называть *виртуальным кодированием блока* (с заданным ключом $k_{i,j}$). Его алгоритм выглядит следующим образом:

1. Вычисление разности $d_{i,j}$ по формуле (4). Увеличение частоты символа $|d_{i,j}|$ в соответствующей модели.
2. Увеличение частоты символа $k_{i,j}$ в соответствующей модели.
3. Просматриваем последовательно все коэффициенты блока (за исключением постоянной составляющей), $l=1, \dots, 63$. Пусть $\tilde{w}_{i,j}^{(l)} \in Z_r^r$, тогда, если $k_{i,j}(r)=1$, увеличить частоту символа $|\tilde{w}_{i,j}^{(l)}|$ в модели, определяемой по правилу (6).

Тогда битовые затраты R_k на кодирование k -го символа алфавита можно оценить, используя соответствующую гистограмму (статистическую модель) арифметического кодера:

$$R_k = -\log_2 v_k,$$

где v_k – частота (то есть оценка вероятности) появления k -го символа алфавита.

Искать решение задачи (9) будем, проводя последовательную минимизацию:

$$L(\mathbf{k}^*, q^*) = \min_q \min_{\mathbf{k}} (L(\mathbf{k}, q) = D(\mathbf{k}, q) + \lambda R(\mathbf{k}, q)). \quad (10)$$

Пусть $L_{i,j}$ – значение функции Лагранжа для блока с координатами (i, j) , $D_{i,j}$ и $R_{i,j}$ – ошибка квантования и битовые затраты на кодирование блока с координатами (i, j) соответственно. Обозначим как $P_m(w)$ частоту появления символа w в модели с номером m , используемой для кодирования абсолютных значений коэффициентов ДПКП; $P_{dif}(u)$, $P_{key}(v)$ – частоты появления символов u и v в моделях кодирования абсолютных значений разностей ДИКМ постоянных составляющих и ключей блоков соответственно.

Поиск набора ключей блоков \mathbf{k}^* , минимизирующего функцию Лагранжа изображения

$$L = \sum_{i,j} L_{i,j} \text{ при заданных значениях } q \text{ и } \lambda,$$

выполняется по следующему алгоритму (последовательно для каждого блока):

1. Определить базовый ключ блока $k_{i,j}$ в соответствии с правилом (5). Положить текущее значение ключа

$k = k_{i,j}$, значение функции Лагранжа $L_{i,j}^* = +\infty$.

2. Вычислить функцию Лагранжа для блока при текущем значении ключа k :

$$L_{i,j} = D_{i,j} + \lambda R_{i,j},$$

$$D_{i,j} = \left(w_{i,j}^{(0)} - \tilde{w}_{i,j}^{(0)} \right)^2 + \sum_{l=1}^{63} \sum_{\tilde{w}_{i,j}^{(l)} \in Z_r^r, k(r)=1} \left(w_{i,j}^{(l)} - \tilde{w}_{i,j}^{(l)} \right)^2 +$$

$$+ \sum_{l=1}^{63} \sum_{\tilde{w}_{i,j}^{(l)} \in Z_r^r, k(r)=0} \left(w_{i,j}^{(l)} \right)^2$$

$$R_{i,j} = - \sum_{l=1}^{63} \sum_{\tilde{w}_{i,j}^{(l)} \in Z_r^r, k(r)=1} \log_2 P_m \left(\left| \tilde{w}_{i,j}^{(l)} \right| \right) -$$

$$- \log_2 P_{dif} \left(\left| d_{i,j} \right| \right) - \log_2 P_{key} (k) + a$$

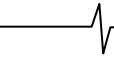
где $m = \text{Model}(\tilde{w}_{i,j}^{(l)})$ (6); a – количество ненулевых коэффициентов в блоке (считаем, что для кодирования знака каждого ненулевого коэффициента необходим 1 бит).

1. Если $L_{i,j} < L_{i,j}^*$, то положить $L_{i,j}^* = L_{i,j}$, $k_{i,j}^* = k$.
2. Выполнить полный перебор ключей $\{k\}$, которые могут быть получены из базового $k_{i,j}$ в результате всевозможных вариантов обнуления единичных битов базового ключа $k_{i,j}$: с каждым возможным текущим значением ключа k выполнить переход на шаг 2. Если перебор вариантов осуществлен, то перейти на шаг 5.

3. Выполнить виртуальное кодирование блока с ключом $k_{i,j}^*$ (по описанному выше алгоритму).

В результате работы алгоритма при заданных значениях параметра λ и шага квантования q будет получен набор оптимальных ключей блоков \mathbf{k}^* , в результате чего некоторая часть исходно ненулевых зон в блоках может быть впоследствии принудительно обнулена. Приведенную процедуру необходимо итерационно использовать при поиске оптимального значения шага квантования q^* , воспользовавшись одним из методов одномерной минимизации для функции $L(\mathbf{k}^*(q), q)$ по переменной q .

Количество итераций в описанной выше процедуре RD-оптимизации можно уменьшить за счет сокращения интервалов поиска оптимальных значений шага квантования и ключей. Так, например, эксперименты показали, что перебор всевозможных вариантов обнуления единичных битов на шаге 4 не требуется. Достаточно рассмотреть ключи, получаемые из базового заменой на каждой итерации одного бита, равного 1, на 0. То есть если число ненулевых битов в некотором ключе k_{ij} равно n , то вместо 2^n возможного коли-



чества вариантов обнуления зон достаточно рассмотреть n вариантов. Помимо этого, при реализации наиболее вычислительно емкого шага 2 процедуры RD-оптимизации возможно (и целесообразно) использование табличного вычисления квадратичной функции и функции логарифма.

Также было установлено, что связь между значением оптимального шага квантования q^* и параметра λ может быть приближенно описана соотношением: $q^* \approx C\sqrt{\lambda}$, где C – некоторая общая для различных изображений константа. При обработке изображений «Lena», «Barbara» и «Goldhill» с использованием метода наименьших квадратов было найдено следующее соотношение: $q^* \approx 2,21\sqrt{\lambda}$, поэтому поиск оптимального значения шага квантования q при минимизации $L(\mathbf{k}^*(q), q)$ выполняется в интервале $q \in (2\sqrt{\lambda}, 3\sqrt{\lambda})$.

Приведенный алгоритм RD-оптимизации вносит некоторую дополнительную вычислительную нагрузку в исходный алгоритм сжатия, но вместе с тем, существенно улучшает результаты базового алгоритма. Кроме того, усложнение алгоритма компрессии за счет введения процедуры RD-оптимизации относится только к стороне кодера, а объем операций, необходимых на стороне декодера, остается прежним.

Заключительный этап реализации схемы рис. 3 – много-модельное арифметическое кодирование каждого блока (i, j) с ключом $k_{i,j}^*$, найденным в результате RD-оптимизации – состоит из следующих шагов:

1. Арифметическое кодирование постоянной составляющей. Вычисление разности $d_{i,j}$ по формуле (4). Абсолютное значение разности $|d_{i,j}|$ кодируется в отдельном потоке. Если $d_{i,j} \neq 0$, то знак $s_{i,j} = \text{sign}(d_{i,j})$ (1 бит) записывается в поток знаков.

2. Арифметическое кодирование ключа $k_{i,j}^*$ в соответствующем потоке.

3. Арифметическое кодирование спектральных коэффициентов из ненулевых зон. Просматриваем последовательно все коэффициенты блока (за исключением постоянной составляющей), $l=1, \dots, 63$. Пусть $\hat{w}_{i,j}^{(l)} \in Z_{i,j}^r$, тогда, если $k_{i,j}^*(r) = 1$, закодировать $|\hat{w}_{i,j}^{(l)}|$ арифметическим кодером, используя модель, определяемую по правилу (6), и, если $\hat{w}_{i,j}^{(l)} \neq 0$, записать $s_{i,j}^{(l)} = \text{sign}(\hat{w}_{i,j}^{(l)})$ в файл знаков. Если $k_{i,j}^*(r) = 0$, то коэффициент $\hat{w}_{i,j}^{(l)}$ не обрабатывается (и будет восстановлен декодером как $\hat{w}_{i,j}^{(l)} = 0$).

Экспериментальные результаты и выводы

На графиках (рис. 4, 5 и 6) приведены результаты, полученные при сжатии изображений «Goldhill», «Barbara» и «Lena» базовым и модифицированным вариантами алгоритма, а также по методу JPEG. Сжатие по стандарту JPEG выполнялось с помощью программного обеспечения от Independent JPEG Group (версия 6b) [16] с использованием библиотеки арифметического кодирования [17]. Для корректности сравнения при расчете битовых затрат размер заголовка вычитался из общего размера файла JPEG.

Ошибка кодирования изображений (ось ординат) на рис. 4-6 измерялась по величине пикового отношения сигнала к шуму:

$$\text{PSNR} = 20 \lg \frac{255}{\sqrt{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{ij} - \tilde{x}_{ij})^2}} \quad (\text{дБ}),$$

где $\mathbf{X} = \{x_{ij}\}$ – исходное, $\tilde{\mathbf{X}} = \{\tilde{x}_{ij}\}$ – восстановленное изображение размера $M \times N$ пикселей. Величина сжатия данных (ось абсцисс) выражена в битовых затратах на кодирование одного пикселя изображения, bpp (bits per pixel).

Как показывают приведенные результаты для распространенных тестовых изображений, предложенный алгоритм компрессии изображений по эффективности не уступает JPEG на базе ДКП, более того, на мелкодетальных «высокочастотных» изображениях (типа «Barbara») показывает преимущества в сжатии при меньших вычислительных затратах, а следовательно, в приложениях, имеющих ограничения на аппаратные ресурсы, может заменить метод JPEG.

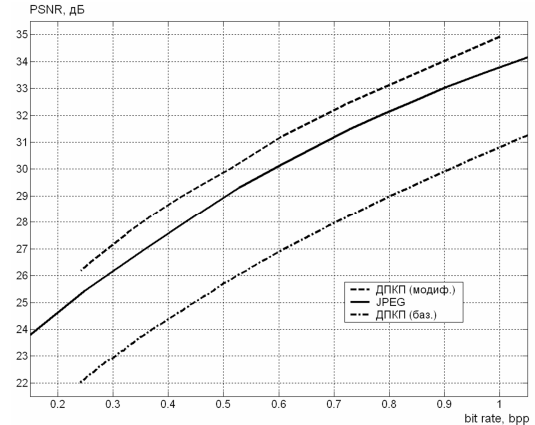


Рис. 4. График зависимости величины PSNR (дБ) от количества бит на пиксель изображения (bpp) («Barbara»)

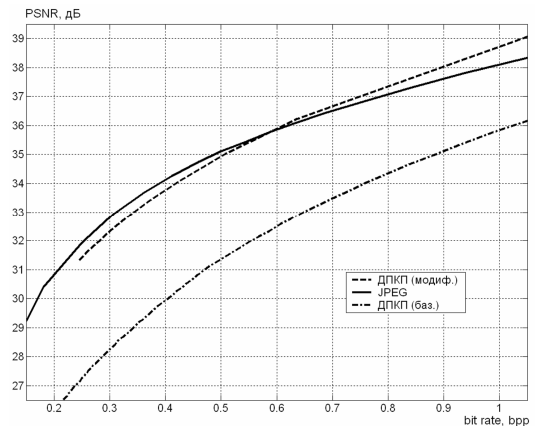


Рис. 5. График зависимости величины PSNR (дБ) от количества бит на пиксель изображения (bpp) («Lena»)

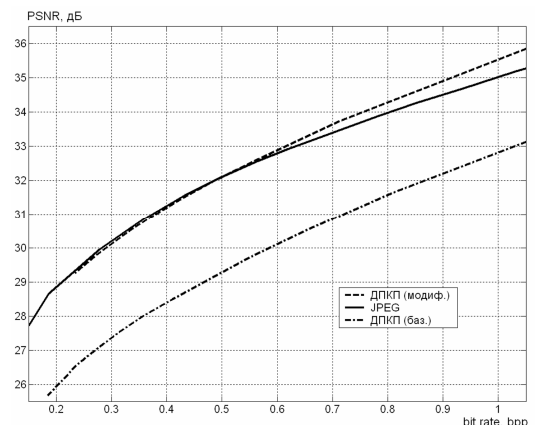


Рис. 6. График зависимости величины PSNR (дБ) от количества бит на пиксель изображения (bpp) («Goldhill»)

Литература

1. E. Feig, S. Winograd. On the multiplicative complexity of discrete cosine transform // IEEE transactions on information theory. – July 1992. – Vol. 38. – No. 4. – p. 1387-1391.
2. Jie Liang, Trac D. Tran. Fast multiplierless approximations of the DCT with the lifting scheme // IEEE transactions on signal processing. – December 2001. – Vol. 49. – No. 12. – p. 3032 – 3044.
3. Neri Merhav. Multiplication-free approximate algorithms for compressed domain linear operations on images // IEEE transactions on image processing. – February 1999. – Vol. 8. – No. 2. – p. 247-254.
4. С. В. Умняшкин. О модификации дискретного косинусного преобразования // Изв. Тул. гос. ун-та. Сер. Математика. Механика. Информатика. Тула: ТулГУ. – 1998. – Т. 4. – Вып. 1. – с. 143-147.
5. N. Jayant, P. Noll. Digital coding of waveforms. – Englewood Cliffs, NJ: Prentice Hall, 1984.
6. I. H. Witten, R. M. Neal, J. G. Cleary. Arithmetic coding for data compression // Communications of the ACM. – June 1987. – Vol. 30. – No. 6. – p. 520-540.
7. M. Nelson, J.-I. Gailly. The data compression book (second edition). – New York: M&T Books, 1995. – p. 541.
8. J. Rissanen. K. M. Mohiuddin. A multiplication-free multialphabet arithmetic code // IEEE transactions on communication. – February 1989. – Vol. 37. – p. 93-98.
9. М. В. Захаров. Об одной реализации арифметического кода. Электронный ресурс <http://www.maxime.net.ru/doc/mfarc.ps.gz>.
10. G. Davis, A. Nosratinia. Wavelet-based image coding: an overview // Applied and computational control, signals and circuits. – 1998. – Vol. 1. – No. 1. – p. 205-269.
11. С. В. Умняшкин. Использование контекстного арифметического кодирования для повышения сжатия данных по схеме JPEG // Известия вузов. Электроника. – 2001. – №3. – с. 96-99.
12. Ф. В. Стрелков, С. В. Умняшкин. Контекстное кодирование коэффициентов дискретного косинусного преобразования в JPEG-подобной схеме компрессии // Цифровая обработка сигналов. – 2003. – № 2. – с. 5-10.
13. X. Xue, W. Gao. Context-based statistical model for DCT-based image coder // Proceedings of picture coding symposium '99. Oregon State Univ. Corvallis, OR, USA. – April 1999. – p. 383-385.
14. С. В. Умняшкин, Е. А. Безуглова. Контекстное кодирование коэффициентов дискретного косинусного преобразования на основе межблочной корреляции в JPEG-подобной схеме компрессии // Цифровая обработка сигналов. – 2004. – № 2. – с. 13-17.
15. T. Berger. Rate distortion theory. – Englewood Cliffs, NJ: Prentice Hall, 1971.
16. Электронный ресурс <http://www.ijg.org>.
17. Электронный ресурс <http://sylvania.net/jpeg-ari/jpeg-ari.zip>.

A NEW STILL IMAGE COMPRESSION ALGORITHM BASED ON DCT MODIFICATION**Sergei V. Umnyashkin, Valentina V. Kurina.**

A new still image compression algorithm based on fast integer multiplication free DCT modification is proposed in this paper. The main features of the algorithm are RD-optimization, which is used to define optimal parameters of the algorithm, and arithmetic coding with several models. Developed algorithm is not inferior in performance to JPEG based on DCT and demands less computational costs.

Уважаемые коллеги!

Вы можете подписаться на наш журнал на первое полугодие 2010 года в любом отделении почтовой связи. Подписной индекс издания по каталогу ОАО «Роспечать» - 82185. Журнал издается ежеквартально.

Для тех, кто не успел оформить подписку на первое полугодие 2010 года через ОАО «Роспечать», сохраняется возможность приобретения журналов непосредственно в редакции по адресу: 107031, г. Москва, Рождественка, 6\9\20, стр. 1, Российское научно-техническое общество радиотехники, электроники и связи им. А.С. Попова, или оформить Заказ в соответствии с требованиями, выставленными на сайте журнала: www.dsra.ru.

Справки по телефонам: (495) 621-71-08, 621-06-10.

Факс: (495) 621-16-39.

E-mail: nto.popov@mtu-net.ru

tor@rgta.ryazan.ru