

Компрессия цифровых изображений на основе кодирования древовидных структур вейвлет-коэффициентов с прогнозированием статистических моделей

С.В.Умняшкин,

Московский государственный институт электронной техники

Предложен вычислительно эффективный алгоритм компрессии статических цифровых изображений, использующий многомодельное арифметическое кодирование спектров вейвлет-преобразований с RD-оптимизированным выделением структур нулевых деревьев. Для тестового изображения Lena при уровне сжатия 0.5 бит на пиксель отношение пикового значения сигнала к шуму составило величину 37,66 дБ.

1. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Базисы функциональных систем вейвлетов (часто называемых также всплесками) [1] на протяжении последних лет привлекают огромное внимание исследователей. Важное свойство, которое отличает вейвлеты, состоит в возможности построения ортогональных систем функций, обладающих переменным разрешением как во временной, так и в частотной областях, что предопределяет широкое применение вейвлет-преобразований для сжатия нестационарных (имеющих локальные особенности) сигналов, примером которых являются фотографические изображения естественного характера. В последнее десятилетие XX века появилось большое количество работ, посвященных изучению и разработке алгоритмов цифровой компрессии изображений с использованием вейвлет-преобразований, среди которых можно выделить и публикации, уже ставшие классическими (хотя бы по признаку частоты упоминания). К их числу относятся и работы [2,3], общие идеи которых послужили отправной точкой для построения предлагаемого алгоритма вейвлет-компрессии изображений.

Пусть $\mathbf{X} = \{x_{k,m}\}$ ($k=0,\dots,N-1$; $m=0,\dots,M-1$) - матрица отсчетов яркости, или дискретное изображение, а $\mathbf{W} = \{w_{k,m}\}$ - матрица той же размерности, представляющая собой результат двумерного n -шагового вейвлет-преобразования [1] матрицы \mathbf{X} , $\mathbf{W}=F(\mathbf{X})$. Матрицу \mathbf{W} будем называть также вейвлет-спектром. Количество шагов вейвлет-преобразования определяет число уровней частотного разрешения в дискретном спектре (матрице \mathbf{W}), для n шагов имеем $n+1$ уровень. При этом коэффициенты вейвлет-разложения можно упорядочить в виде совокупности деревьев, корнями которых являются элементы, лежащие в самом низкочастотном диапазоне спектра (*саббэнде*), см. пример для спектра трехшагового преобразования ($n=3$), приведенный на рис. 1. Корневой узел, соответствующий масштабирующей функции (единственная функция в дереве, не являющаяся вейвлетом), имеет трех потомков. Прочие узлы-родители соответствуют вейвлет-функциям и имеют по четыре потомка, причем базисные функции узлов-потомков имеют носитель, не выходящий за границы носителя функции-родителя. Переход от верхнего уровня дерева к нижнему (в направлении стрелок) соответствует повышению

итерационно просматривать узлы дерева, и для каждого узла i принимать решение о подрезании или сохранении выходящей из него ветви по значению специальным образом вычисленного RD-критерия. Предложенный в [3] алгоритм показывает достаточно высокие характеристики сжатия, однако является вычислительно сложным. Кроме того, при кодировании не попавших в нулевые ветви вейвлет-коэффициентов используется простейший вариант арифметического кодирования [4], когда все коэффициенты обрабатываются единым потоком данных с использованием единственной статистической модели распределения вероятностей. В настоящей работе изучается возможность дальнейшего развития алгоритмов, предложенных в [2,3].

2. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

При изложении будем использовать далее следующие обозначения. Дерево, включающее в себя все свои спектральные коэффициенты, обозначим T . Строчными буквами i будем обозначать номера узлов дерева, каждый из которых соответствует спектральному коэффициенту (начало нумерации с нуля, последовательность нумерации соответствует схеме рис. 2). Соответственно под T_i понимаем дерево, начинающееся из узла i и включающее в себя этот узел. Таким образом, T_i – это определенная ветвь полного дерева T . (Полное дерево можно также обозначать T_0 , но для краткости нулевой индекс опускаем.) *Подрезанным* деревом S_i называется любое поддереву дерева T_i ($S_i \subseteq T_i$), содержащее узел i . Аналогично, S – это дерево, полученное в результате подрезания ветвей полного дерева T , а все возможные варианты топологии дерева характеризуются множеством $\{S | S \subseteq T\}$. *Деревом остатка* U_i будем называть множество всех прямых потомков узла i , т.е. $U_i = T_i \setminus \{i\}$. Множество непосредственных потомков узла i обозначаем через C_i . Тогда можем записать также, что $U_i = \bigcup_{j \in C_i} T_j$. *Листьями* будем называть те

узлы дерева (подрезанного или неподрезанного), у которых нет потомков, т.е. для которых $C_i = U_i = \emptyset$. В исходном неподрезанном дереве T листьями являются узлы, соответствующие трем саббэндам самого высокого разрешения. Отметим, что среди прочих узлов только узел $i=0$ (начало дерева) имеет трех непосредственных потомков, остальные узлы имеют по четыре потомка (см. также рис. 1 и 2). *Уровнями дерева* называем горизонтальные слои узлов (рис. 1) и обозначаем их как L_k . Для n -шагового преобразования в исходном дереве T имеем $n+1$ уровней $\{L_0, \dots, L_n\}$, при этом нулевой уровень состоит из единственного узла, $L_0 = \{0\}$, а слой с номером n состоит из всех узлов-листьев $L_n = \{2^{2(n-1)}, \dots, 2^{2n}-1\}$.

Общая схема разработки алгоритма вейвлет-компрессии, основанного на подрезании ветвей, предусматривает решение следующих вопросов:

- Каким образом кодировать топологию дерева S (фактически, как сократить битовые затраты на кодирование нулевых ветвей)?
- Каким образом кодировать сохраненные коэффициенты из дерева S ?

Как и в работе [3], мы ограничим рассмотрение случаем, когда единственным параметром кодирования коэффициентов из дерева S является шаг q равномерного скалярного квантования вейвлет-коэффициентов. Тогда задача оптимизации кодирования может быть сформулирована в следующих хорошо известных терминах RD-теории: для заданного значения $\lambda \geq 0$ найти среди всех возможных вариантов топологии $\{S | S \subseteq T\}$ и всех возможных уровней скалярного квантования $q > 0$ такую пару (S^*, q^*) , чтобы функция Лагранжа $J = D + \lambda R$ принимала минимальное значение, т.е.

$$J(S^*, q^*) = \min_{S \subseteq T, q > 0} [D(S, q) + \lambda R(S, q)], \quad (1)$$

где D – квадрат ошибки, внесенной в спектр в результате скалярного квантования и подрезания ветвей, а R – длина кода (в битах) на выходе кодера. Параметр $\lambda \geq 0$ устанавливает баланс между битовыми затратами R и ошибкой представления данных D : увеличивая λ , получаем снижение битовых затрат, сопровождающееся повышением величины ошибки (потерь данных).

Так же, как и в работе [3], минимизацию (1) будем проводить последовательно, выполняя перебор (возможно, направленный) значений q , и для каждого q проводя поиск оптимальной топологии:

$$J(S^*, q^*) = \min_{q > 0} \min_{S \subseteq T} [D(S, q) + \lambda R(S, q)]. \quad (2)$$

Основная задача, таким образом, состоит в выполнении внутренней минимизации (поиска оптимальной топологии), при фиксированных значениях λ, q :

$$J(S^*) = \min_{S \subseteq T} [D(S) + \lambda R(S)], \quad (3)$$

$$q = \text{const}, \quad \lambda = \text{const}.$$

3. ОТЛИЧИТЕЛЬНЫЕ ЧЕРТЫ ПРЕДЛАГАЕМОГО АЛГОРИТМА

Перечислим те особенности, которые отличают предлагаемый способ решения задачи (3) от алгоритма, описанного в работе [3].

3.1. Арифметическое кодирование данных

Недостатком алгоритма [3] является использование для статистического кодирования одного единственного потока данных, формируемого из всех проквантованных коэффициентов, которые остались после подрезания ветвей дерева. При этом используемый вариант арифметического кодера [4] настраивается на некоторую усредненную статистику, между тем саббэнды низкого уровня разрешения (низкочастотные) имеют большую дисперсию значений вейвлет-коэффициентов, а саббэнды высокого уровня разрешения (высокочастотные) имеют меньшую дисперсию – данный факт хорошо известен [5]. Ясно, что смешивание разных статистик в единый поток данных снижает эффективность статистического кодирования, но насколько именно, сказать трудно. Поэтому, прежде всего, откажемся от реализации алгоритма арифметического кодирования [4] и воспользуемся созданной М.Е.Кочетковым

и Ф.В.Стрелковым программной библиотекой, допускающей одновременную обработку нескольких входных потоков данных (каждому потоку соответствует своя адаптивная статистическая модель) с выводом в единый сжатый выходной поток данных [6].

Для оценки длины кода, необходимого при кодировании очередного символа в любом из входных потоков, в указанной библиотеке имеются соответствующие функции, которые используют те же самые внутренние статистические модели, что и арифметический кодер. Отметим также, что оценка длины кода требует пренебрежимо малых вычислительных затрат, так как использует табличные значения функции $\log_2(x)$ (особенности арифметических кодеров [4,6] позволяют это сделать). Оценка длины кода кодируемого символа является важной функцией, необходимой при построении различных процедур RD-оптимизации. Программная библиотека позволяет точно учесть влияние внутреннего статистического моделирования кодера и, таким образом, представляет собой новый инструмент для создания различных алгоритмов и программ компрессии данных.

3.2. Кодирование топологии дерева

При кодировании топологии S в [3] для всех узлов дерева, кроме листьев, предложено использовать бинарную карту $\{n_i\}$: если $n_i=0$, то дерево в данном узле подрезается, а если $n_i=1$, то по крайней мере непосредственные потомки – узлы из множества C_i – сохраняются. Сказанное иллюстрируется рис. 3а. Статистическое кодирование признаков $\{n_i\}$ в алгоритме [3] не используется.

Вместе с тем, признаки $\{n_i\}$ соседних (по положению внутри саббэнда) узлов являются коррелированными величинами. Чтобы учесть эту корреляцию, признаки для соседних узлов будем группировать. Поскольку попасть или не попасть в нулевое дерево четверка непосредственных потомков C_j некоторого узла j может лишь целиком, то четыре соответствующих признака $\{n_i\}_{i \in C_j}$ логично объединить в единый элемент данных. Карту подрезания ветвей (топологию дерева) тогда будем описывать новым алфавитом данных с символами $N_i=(n_{i1}, n_{i2}, n_{i3}, n_{i4})$, которые будем кодировать статистически. В виде числа новый признак можно выразить как $N_i=n_{i1}+2n_{i2}+4n_{i3}+8n_{i4}$. В новом алфавите, очевидно, 16 символов, что не так много и позволяет сохранить способность к быстрой настройке и адаптации соответствующих статистических моделей арифметического кодера. Таким образом, новый расширенный признак N_j оказывается связанным уже с узлом j более высокого уровня. Это вносит существенное изменение (по сравнению с [3]) и в алгоритм RD-оптимизации: просматривая узлы $\{i\}$ дерева T снизу вверх, мы анализируем теперь такую возможность подрезания дерева, когда нулевые ветви могут начинаться от узлов $m \in C_i$, но сами непосредственные потомки C_i в нулевые ветви не включаются. Сказанное иллюстрируется рис. 3б.

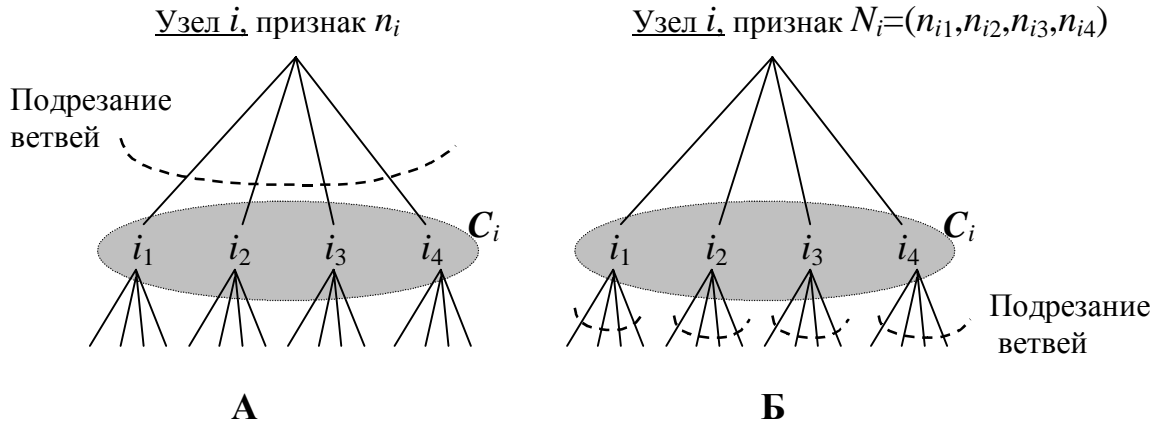


Рисунок 3. Отличие RD-анализа подрезания ветвей при послойном просмотре узлов: А – алгоритм [3], Б – модифицированное кодирование топологии. $C_i=\{i_1, i_2, i_3, i_4\}$

3.3. Учет статистических зависимостей между родителями и потомками

Идея, которая восходит к работе [2] и в том же виде используется в [3], заключается в следующем: чем больше абсолютная величина вейвлет-коэффициента $|w_i|$ (или энергия, w_i^2) узла-родителя i , тем менее вероятно появление у данного узла нулевой (т.е. подрезанной) ветви. Более точное предсказание появления нулевой ветви можно составить, если использовать в качестве прогнозной величины P_i сумму, включающую в себя, помимо w_i^2 , также квадраты значений вейвлет-коэффициентов, соседних в саббэнде к узлу i [2,3]. Для того, чтобы и кодер, и декодер оперировали при построении прогноза одними и теми же данными, вместо оригинальных значений w_i необходимо использовать $\tilde{w}_i = \text{Round}(w_i / q)$ или $\hat{w}_i = q\tilde{w}_i$. В работе [2] учитывались четверки узлов, имеющих общего родителя, в работе [3] – множество из девяти узлов саббэнда, накрываемое окном 3×3 отсчета с центром в узле i . Назначение величины P_i в работах [2,3] состоит в сокращении битовых затрат на кодирование карты признаков: если P_i меньше определенного порога t_{low} , то ветвь обязательно подрезается, а если P_i больше некоторого порога t_{high} , то ветвь обязательно сохраняется. Битовые карты $\{n_i\}$ передаются без статистического кодирования, только для тех узлов, для которых $t_{low} \leq P_i \leq t_{high}$.

Мы будем использовать выражение для прогнозной величины узла i , которое представляет собой следующую взвешенную сумму абсолютных величин вейвлет-коэффициентов:

$$P_i = \frac{1}{16} \left(4|\tilde{w}_i| + 2 \sum_{m \in I_1(i)} |\tilde{w}_m| + \sum_{m \in I_2(i)} |\tilde{w}_m| \right), \quad (4)$$

где множества для индексов суммирования определяются среди соседей узла i в саббэнде по рисунку 4.

Весовые коэффициенты, используемые в формуле (4), были получены в результате статистической обработки ряда тестовых изображений с целью

выявления максимума выборочного значения для коэффициента корреляции между прогнозной величиной P_i и энергией проквантованных вейвлет-коэффициентов непосредственных потомков C_i :

$$\rho \left(P_i, \sum_{j \in C_i} \tilde{w}_j^2 \right) \rightarrow \max.$$



Рисунок 4. Весовые коэффициенты для составления прогноза (4)

4. АЛГОРИТМ КОДИРОВАНИЯ И ОПТИМИЗАЦИИ

В предлагаемом алгоритме прогнозная величина вида (4) используется следующим образом.

- Для кодирования расширенных признаков карты топологии $N_i=(n_{i1},n_{i2},n_{i3},n_{i4})$ будем использовать несколько моделей арифметического кодера, выбор между которыми осуществляется с использованием прогноза вида (4) при помощи функции, которую обозначим $\text{IndMap}(\bullet)$.
- Значение прогноза вида (4) будем учитывать также и при выборе моделей кодирования для прямых потомков $j \in C_i$ узла i . Соответствующее правило (функцию) выбора модели обозначим $\text{IndSpec}(\bullet)$.

Таким образом, предлагаемый алгоритм вейвлет-компрессии использует несколько статистических моделей для кодирования топологии и несколько статистических моделей для кодирования не попавших в нулевые ветви спектральных коэффициентов. Каждая модель соответствует отдельному потоку входных данных арифметического кодера. Функцию арифметического кодера, которая вычисляет по внутренним статистическим моделям количество бит, необходимое для кодирования символа c в k -ом потоке, обозначим $H(k,c)$. Данная функция представляет собой уточненное (учитывающее специфику примененного варианта арифметического кодирования) значение для оценки энтропии символа, которую можно получить по частоте $\nu_{k,c}$ появления символа c в k -ом потоке, $-\log_2(\nu_{k,c})$. Обозначение H_{spec} относится к потокам, в которых кодируются проквантованные вейвлет-коэффициенты, H_{map} – к потокам, в которых кодируются признаки начала нулевого дерева. Деревья спектра обрабатываются последовательно; после оптимизации топологии очередного дерева его необходимо закодировать, и тем самым произвести подстройку внутренних статистических моделей арифметического кодера.

4.1. Описание алгоритма

Функция Лагранжа $J^* = D^* + \lambda R^*$, которая вычисляется в результате работы алгоритма, характеризует ошибку и битовые затраты от кодирования топологии и всех спектральных коэффициентов, не включая корневого. Предлагаемый алгоритм поиска оптимальной топологии (3) выглядит следующим образом.

ШАГ 0. /* инициализация */

$S \leftarrow T$.

$l \leftarrow n-2$. /* l – просматриваемый уровень дерева */

ШАГ 1. /* подготовительный */

$\forall i \in L_{n-1}$: /* просматриваются все узлы предпоследнего уровня */

ШАГ 1.1. /* коррективировка проквантованных коэффициентов */

$\forall j \in C_i$:

$$\tilde{w}_j \leftarrow \arg_k \left(J(k) = \min_{k \in \{\tilde{w}_j, \tilde{w}_j \pm 1, 0\}} \left[(kq - w_j)^2 + \lambda H_{spec}(\text{IndSpec}(j), k) \right] \right).$$

ШАГ 1.2. /* расчет RD-функций Лагранжа для вариантов сохранения и подрезания листьев */

$$J_{U_i}^0 \leftarrow \sum_{j \in U_i} w_j^2.$$

$$J_{U_i}^1 \leftarrow \sum_{j \in U_i} \left((\hat{w}_j - w_j)^2 + \lambda H_{spec}(\text{IndSpec}(j), \tilde{w}_j) \right)$$

ШАГ 2. /* просмотр текущего уровня с попыткой подрезания ветвей */

$\forall i \in L_l$:

Если $i \neq 0$ **то** /* не достигли начала дерева */

ШАГ 2.1. /* определение оптимальной топологии ветвей */

$$J_i^* \leftarrow J_{\sum_{j \in C_i} U_j}^* = \min_{n_j \in \{0,1\}} \left(\sum_{j \in C_i} J_{U_j}^{n_j} + \lambda H_{map}(\text{IndMap}(\bar{P}_i), N_i) \right),$$

$$\text{где } \bar{P}_i = \frac{1}{4} \sum_{j \in C_i} P_j \text{ (см.(4)), } N_i = 8n_{i_4} + 4n_{i_3} + 2n_{i_2} + n_{i_1},$$

$$\text{здесь: } i_4 > i_3 > i_2 > i_1, \quad \{i_1, i_2, i_3, i_4\} = C_i.$$

ШАГ 2.2. /* изменение топологии S */

$\forall j \in C_i$: **Если** $n_j = 0$, **то** $U_j \leftarrow \emptyset$. /* подрезать ветвь */

ШАГ 2.3. /* коррективировка проквантованных коэффициентов */

$\forall j \in C_i$:

$$\tilde{w}_j \leftarrow \arg_k \left(J(k) = \min_{k \in \{\tilde{w}_j, \tilde{w}_j \pm 1, 0\}} \left[(kq - w_j)^2 + \lambda H_{spec}(\text{IndSpec}(j), k) \right] \right).$$

ШАГ 2.4. /* подготовка для просмотра следующего уровня */

$$J_{U_i}^0 \leftarrow \sum_{j \in C_i} (w_j^2 + J_{U_j}^0)$$

$$J_{U_i}^1 \leftarrow J_i^* + \sum_{j \in C_i} ((\hat{w}_j - w_j)^2 + \lambda H_{spec}(\text{IndSpec}(j), \tilde{w}_j))$$

ШАГ 2.5. $l \leftarrow l-1$. Перейти на шаг 2.

иначе /* $i=0$, достигли начала дерева */

ШАГ 2.6. /* определение оптимальной топологии дерева */

$$J_0^* \leftarrow J_{\sum_{j \in C_0} U_j}^* = \min_{\substack{n_j \in \{0,1\} \\ j \in C_0}} \left(\sum_{j \in C_0} J_{U_j}^{n_j} + \lambda H_{map}(0, 4n_3 + 2n_2 + n_1) \right),$$

где $C_0 = \{1, 2, 3\}$.

ШАГ 2.7. /* изменение топологии S */

$\forall j \in C_0$: **Если** $n_j=0$, **то** $U_j \leftarrow \emptyset$. /* подрезать ветвь */
Перейти на шаг 3.

ШАГ 3. /* вычисление RD-функции Лагранжа для всего дерева */

$$J^* \leftarrow J_0^* + \sum_{j \in C_0} ((\hat{w}_j - w_j)^2 + \lambda H_{spec}(\text{IndSpec}(j), \tilde{w}_j))$$

$$S^* \leftarrow S. \quad /* \text{оптимальная топология} */$$

КОНЕЦ.

4.2. Комментарии к алгоритму

Просмотр узлов дерева проводится от листьев к корню. На первом (подготовительном) шаге просматриваются те узлы i , которые имеют только непосредственных потомков ($C_i = U_i$), формируются массивы из значений RD-функций, соответствующих вариантам подрезания ($J_{U_i}^0$) и сохранения ($J_{U_i}^1$) листьев. Предварительно, на шаге 1.1, для каждого узла-листа анализируется возможность дополнительной минимизации RD-функции Лагранжа, характеризующей скалярное квантование вейвлет-коэффициентов. Данная процедура основана на известных свойствах распределения вероятностей вейвлет-коэффициентов, в соответствии с которыми битовые затраты R на кодирование коэффициента априори тем меньше, чем меньше его абсолютное значение (по этой причине для случая $\tilde{w}_j = 0$ процедура дополнительной минимизации не применяется). Вместе с тем, значительное изменение величины проквантованного коэффициента влечет и существенное увеличение составляющей ошибки в RD-функции Лагранжа, потому проверка большого числа вариантов изменения значений \tilde{w}_j лишена смысла - эксперименты это подтверждают. Тем не менее, возможность обнуления коэффициента все-таки проверяется, поскольку для высокочастотных составляющих вейвлет-спектра

нулевое значение часто имеет очень малые битовые затраты на кодирование (доли бита).

На втором шаге, который выполняется для всех узлов i следующих уровней, $i \in L_l$ ($l=n-2, \dots, 0$), производится выбор RD-оптимального способа подрезания ветвей (шаги 2.1-2.2), начинающихся из узлов $j \in C_i$. Сами узлы $j \in C_i$ могут попасть в нулевые ветви лишь на последующих повторениях шага 2, см. также рис. 3б. Шаги 2.3 и 2.4 несут тот же смысл, что и шаги 1.1, 1.2. Отметим, что введение дополнительной оптимизации скалярного квантования (шаги 1.1 и 2.3) позволяет при том же уровне сжатия дополнительно понизить среднеквадратичную ошибку обработки изображений на 0,02-0,03 дБ.

Если на шаге 2 производится обработка последнего корневого узла, то в действиях, проводимых на шагах 2.3-2.5, нет необходимости, а вместо шагов 2.1, 2.2 выполняются 2.6, 2.7. Отличия шага 2.6 от 2.1 обусловлено наличием только трех прямых потомков корневого узла $i=0$. Для всех узлов, за исключением корневого, выбор модели для кодирования признаков N_i (на шаге 2.1) производится с использованием правила, определяемого функцией $\text{IndMap}(i)$. Для кодирования признака N_0 , связанного с корнем дерева, используется отдельный поток данных, которому условно присвоен номер 0 (см. шаг 2.6). Алфавит нулевого потока признаков содержит не 16, а 8 символов, поскольку у корневого узла только три непосредственных потомка.

4.3. Правила выбора моделей (потоков) для кодирования данных

Как следует из приведенного описания алгоритма, важнейшую роль в его работе играют функции $\text{IndMap}(\bar{P}_i)$, и $\text{IndSpec}(i)$. Первая функция производит выбор модели кодирования признака $N_i=(n_{i1}, n_{i2}, n_{i3}, n_{i4})$ по среднему значению прогнозных величин $P_i, i=i_1, i_2, i_3, i_4$, и имеет следующий вид.

$$\text{Indmap}(P) = \begin{cases} 1, & \text{если } P < 0.3 \\ 2, & \text{если } 0.3 \leq P < 1.1 \\ 3, & \text{если } 1.1 \leq P < 4 \\ 4, & \text{если } 4 \leq P \end{cases} \quad (5)$$

Таким образом, общее число моделей для кодирования признаков равно пяти, т.к. отдельная модель используется для кодирования значений признаков $N_0=(n_1, n_2, n_3)$, см. шаг 2.6. Пороги $(t_1, t_2, t_3)=(0.3; 1.1; 4.0)$ для выбора моделей находились итерационно, в результате минимизации длины выходного битового кода $R(t_1, t_2, t_3)$ при обработке известных тестовых изображений *Lena*, *Barbara*, *Goldhill*. Эксперименты показали также, что введение большего числа моделей для признаков лишено смысла.

Вторая ключевая функция алгоритма вейвлет-сжатия - $\text{IndSpec}(i)$, которая представляет собой правило выбора модели для кодирования не попавших в нулевые ветви вейвлет-коэффициентов. Для кодирования спектральных коэффициентов более эффективным оказывается использование прогнозной величины, полученной не только по родительскому узлу, но и по вейвлет-коэффициентам узлов, которые находятся в том же саббэнде, рядом с обрабатываемым. Для того чтобы декодер мог использовать ту же информацию, что и кодер, прогнозную величину необходимо составлять только

по уже обработанным узлам. В рассматриваемом алгоритме последовательность кодирования и декодирования узлов вейвлет-спектра определяется схемой рисунка 5, а прогнозная величина для текущего узла (помечен черным) составляется по трем уже обработанным узлам-соседям (помечены серым) и родительскому прогнозу (4).

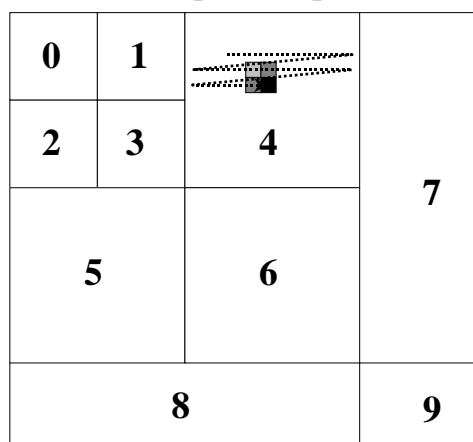


Рисунок 5. Порядок обработки коэффициентов вейвлет-спектра; цифрами обозначен порядок просмотра саббэндов

Для использования в функции $\text{IndSpec}(j)$ в результате экспериментов по обработке тестовых изображений был установлен следующий вид прогнозной величины узла j :

$$s_j = 0.36P_i + 1.06(|\tilde{w}_{j_y}| + |\tilde{w}_{j_x}| + 0.4|\tilde{w}_{j_d}|), \quad (6)$$

где j_y – узел-сосед по вертикали, j_x – узел-сосед по горизонтали, j_d – узел-сосед по диагонали (см. рис. 5), P_i определяется по (4) для узла-родителя i ($j \in C_i$). При этом для самой функции, определяющей номер модели кодирования вейвлет-коэффициентов, получен следующий вид:

$$\text{IndSpec}(j) = \begin{cases} 5, & \text{если } j > 3 \text{ и } s_j < 1.72 \\ 4, & \text{если } j > 3 \text{ и } 1.72 \leq s_j < 4.10 \\ 3, & \text{если } j > 3 \text{ и } 4.10 \leq s_j < 9.80 \\ 2, & \text{если } j > 3 \text{ и } 9.80 \leq s_j < 26 \\ 1, & \text{если } 0 < j \leq 3 \text{ или } 26 \leq s_j \\ 0, & \text{если } j = 0 \end{cases} \quad (7)$$

В результате для кодирования коэффициентов вейвлет-спектра получаем 6 моделей. Нулевая модель относится к кодированию спектральных коэффициентов при масштабирующих функциях и вновь обособлена. Первая модель включает в себя самые низкочастотные вейвлет-коэффициенты (уровня L_1), а также коэффициенты, для которых прогноз (6) – наибольший. Пороги (1.72;4.10;9.80;26) для прогноза (6), которые и определяют выбор модели для кодирования вейвлет-коэффициентов, также уточнялись итерационно, по результатам обработки тестовых изображений.

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для оценки характеристик описанного алгоритма компрессии использованы общепринятые параметры: ошибка кодирования измерялась как отношение пикового значения сигнала к шуму (peak signal to noise ratio, PSNR) [3], а битовые затраты выражались в количестве бит, приходящемся на

кодирование одного пикселя изображения (bits per pixel, BPP). Характеристики, которые дает алгоритм вейвлет-компрессии для стандартных тестовых изображений, приведены в таблицах 1 и 2. В экспериментах применялись пятишаговые вейвлет-преобразования: результаты таблицы 1 получены при использовании наиболее распространенного набора вейвлет-фильтров из работы [5], а данные таблицы 2 получены при обработке изображений с использованием другого набора фильтров, предложенного Wei и др. [7]. Большая часть работы по созданию программного кода, реализующего описанный алгоритм, была проделана студентом МИЭТ А.А.Марковым.

Как показывает сравнение достигнутых характеристик сжатия с доступными результатами использования других алгоритмов (см., например, http://www.icsl.ucla.edu/~ipl/psnr_results.html), предложенный алгоритм имеет весьма высокие показатели при обработке изображения *Lena* и *Goldhill*, прочно занимая место в группе лидеров. Скорее всего выглядят результаты обработки изображения *Barbara*, для которого алгоритм [3] превзойти не удалось. Объяснение подобного отрицательного эффекта кроется в специфике изображения *Barbara*, в котором чередуются фрагменты изображения, имеющие существенно различающиеся свойства: достаточно монотонные, нерезкие переходы сменяются мелкодетальными текстурами одежды, в связи с чем статистическое моделирование арифметического кодера при обработке деревьев сбивается, а примененное прогнозирование не дает желаемого высокого эффекта.

Таблица 1. Величина PSNR (в дБ), полученная при сжатии тестовых изображений с использованием фильтров [5]

| BPP | <i>Lena</i> | <i>Barbara</i> | <i>Goldhill</i> |
|--------|-------------|----------------|-----------------|
| 0.2500 | 34.43 | 28.04 | 30.77 |
| 0.500 | 37.49 | 31.87 | 33.43 |
| 1.000 | 40.72 | 36.86 | 36.93 |

Таблица 2. Величина PSNR (в дБ), полученная при сжатии тестовых изображений с использованием фильтров [7]

| BPP | <i>Lena</i> | <i>Barbara</i> | <i>Goldhill</i> |
|--------|-------------|----------------|-----------------|
| 0.2500 | 34.59 | 28.37 | 30.81 |
| 0.500 | 37.66 | 32.36 | 33.49 |
| 1.000 | 40.81 | 37.36 | 36.99 |

6. ЗАМЕЧАНИЯ О РЕАЛИЗАЦИИ И ПРИМЕНЕНИИ АЛГОРИТМА

При рассмотрении алгоритма RD-оптимизации топологии мы предполагали, что такие два важных параметра, как λ и q , заданы внешне, см. соотношение (3). Вообще же на практике может быть задан только параметр λ , определяющий баланс между качеством и сжатием, и для фиксированного значения λ необходимо проводить минимизацию функции $J(S, q) = D(S, q) + \lambda R(S, q)$ по всем способам формирования топологии S и всем

возможным значениям q , см. формулы (1), (2). При минимизации функции Лагранжа (2) по q , как показывают эксперименты, допустимо использование направленного поиска, возможные локальные минимумы встречаются вблизи минимума глобального и дают пренебрежимо малые отличия в значении целевой функции $J(S, q)$. Начальный интервал $[q_{\min}, q_{\max}]$ для направленного поиска q^* можно выбрать так: $q_{\min} = 2\sqrt{\lambda}$, $q_{\max} = 4.5\sqrt{\lambda}$, поскольку, как было эмпирически установлено, в результате минимизации (2) прослеживается следующая зависимость: $q^* \approx 3.1\sqrt{\lambda}$. Затем на интервале $[q_{\min}, q_{\max}]$ производится направленная минимизация $J(S, q)$ по q , например, методом золотого сечения. Для каждого q производится поиск минимума (3) в соответствии алгоритмом, описанным в данной работе.

Если же задача сжатия поставлена как минимизация ошибки кодирования при фиксированных битовых затратах, т.е. $D(S, q) \rightarrow \min, R(S, q) = R_b$, то процедуру (2) следует проводить итерационно, с целью нахождения такого значения λ , что

$$R(S^*(\lambda), q^*(\lambda)) = R_{\text{opt}}(\lambda) = R_b.$$

Предположив монотонность функции $q^* = q^*(\lambda)$, можем записать обратную функцию $\lambda = \lambda(q^*)$, и минимизацию ошибки при фиксированных битовых затратах искать как решение уравнения

$$R(S^*(q^*), q^*) = R_{\text{opt}}(q^*) = R_b, \quad (8)$$

где

$$q^* = \arg_q \left(J(q) = \min_{\lambda \geq 0} \min_{S \subseteq T} [D(S, q) + \lambda R(S, q)] \right). \quad (9)$$

Внутренняя минимизация здесь совершенно аналогична (3), а внешняя заменена с минимизации по q на минимизацию по λ . Проистекающее из этого преимущество состоит в том, что ошибки квантования вейвлет-коэффициентов $(\hat{w}_j - w_j)^2$, вычисление которых составляет значительную часть арифметических операций алгоритма RD-оптимизации, не меняются при итерационной минимизации (9) по λ и потому могут быть вычислены лишь один раз для каждого значения q , возникающего при последовательном приближении к решению уравнения (8). Поиск корня уравнения (8) может быть осуществлен направленно на некотором отрезке, заведомо содержащем решение: $q^* \in [q_{\min}, q_{\max}]$. Например, во всех экспериментах, проведенных автором с пятишаговыми вейвлет-преобразованиями, значения q не выходили из диапазона [4;64]. При проведении внешней минимизации в (9) по λ вновь имеет смысл использовать метод золотого сечения, уже для отрезка $[\lambda_{\min}, \lambda_{\max}]$, где границы определяются по текущему значению q : $\lambda_{\min} \approx 0.05q^2$, $\lambda_{\max} \approx 0.2q^2$, исходя из существования упомянутой зависимости $q^* \approx 3.1\sqrt{\lambda}$.

7. ЗАКЛЮЧЕНИЕ

Описанный алгоритм вейвлет-компрессии показал высокие результаты, заметно превосходя многие из широко известных алгоритмов по характеристикам сжатия изображений. Отличительной особенностью

предложенного алгоритма является его универсальность и связанная с этим высокая вычислительная эффективность: одна итерация оптимизации топологии (3) для изображения размером 512×512 пикселей на персональном компьютере с процессором уровня Pentium-II 500 MHz выполняется около секунды.

Если усложнение вычислений допустимо, то возможно и дальнейшее развитие алгоритма. В первую очередь, для повышения эффективности сжатия следует уточнить прогноз вида (6), который может строиться по большому количеству обработанных узлов - соседей кодируемого. Особенно заметный эффект от этого следует ожидать как раз на обработке изображений, характеристики которых подобны тестовому изображению *Barbara*, для которого именно контекст кодирования узлов-соседей является весьма информативным. Требуют исследований и проверки относительно свежие идеи, связанные с возможностями учета и прогнозирования знаков вейвлет-коэффициентов [8]. Более принципиальные и сложные изменения, которые видятся естественными в качестве следующего шага, состоят во введении в предложенный алгоритм возможности выбора базиса вейвлет-пакетов, аналогично тому, как был доработан алгоритм [3] его авторами в следующей работе [9].

ЛИТЕРАТУРА

1. *Петухов А.П.* Введение в теорию базисов всплесков. СПб.: Изд-во СПбГТУ, 1999. 132 с.
2. *Lewis A.S., Knowles G.* Image Compression Using the 2-D Wavelet Transform // IEEE Trans. Image Proc. – Vol. 1. - №2, 1992. – P.244-250.
3. *Z.Xiong, K.Ramchandran and M.T.Orchard.* Space-frequency Quantization for Wavelet Image Coding // IEEE Trans. Image Proc. - Vol.6, May 1997, P. 677-693.
4. *Witten I., Neal R.M., Cleary J.G.* Arithmetic coding for data compression // Comm. ACM. - 1987. - Vol.30. - №6. – P. 520-540.
5. *M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies,* Image coding using wavelet transform// IEEE Trans. on Image Proc. - Vol. 1. - №2, 1992. –P. 205-220.
6. *Кочетков М.Е., Умняшкин С.В.* Многопоточная реализация алгоритма арифметического кодирования / М.: МГИЭТ (ТУ), 1998. - 21 с. Депонировано в ВИНТИ 25.12.98 № 3884-B98.
7. *D. Wei, H.-T. Pai, and A. C. Bovik.* Antisymmetric Biorthogonal Coiflets for Image Coding // Proceedings of IEEE International Conference on Image Processing, Chicago, IL, Oct. 1998, vol. 2, pp. 282-286.
8. *A. Deever and S. Hemami.* What's your sign?: Efficient sign coding for embedded wavelet image coding // Proc. of Data Compression Conference, 2000. - P. 273-282.
9. *Z. Xiong, K. Ramchandran, and M. Orchard.* Wavelet packets image coding using space-frequency quantization // IEEE Trans. Image Proc. - Vol. 7, June 1998. - P. 892-898.