

## АДАПТИВНЫЙ МЕТОД СЖАТИЯ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРИ ИНФОРМАЦИИ

Предлагается новый метод кодирования цветных изображений, который обеспечивает их точное восстановление после 2-3-кратного сжатия. Основой этого метода является т.н. преобразование компонент с интерполяцией по отсчетам (ПКИО). Для обеспечения дополнительного сжатия находятся различные комбинации результатов этого преобразования. Описываются две возможности поиска оптимальных комбинаций. Наиболее удачным способом адаптивного формирования комбинаций представляется способ, связанный с вычислением попарных разностей разностных компонент ПКИО. Эксперименты подтверждают высокую эффективность программ, реализующих предложенный метод сжатия.

### 1. ВВЕДЕНИЕ

Изображения - это важнейшие источники информации о различных явлениях реального мира. Наряду с черно-белыми изображениями, в настоящее время во многих областях (например, в медицине, при создании баз данных о картинных галереях и т.п.) начинают широко использоваться многоградационные цветные изображения. Поскольку размеры описаний этих изображений их цветовыми составляющими  $R$ ,  $G$  и  $B$  очень велики, возникает необходимость создать средства сжатия таких описаний; применение этих средств позволит снизить требования к объему внешних запоминающих устройств. При создании центров цифровой обработки изображений и баз данных, содержащих цветные изображения, особое значение приобретают методы сжатия изображений *без потери информации*. Существующие программы архивации данных, которые обеспечивают сжатие без потери информации, ориентированы в первую очередь на компактное представление текстовых файлов и весьма неэффективны даже в случае работы с многоградационными черно-белыми изображениями [1],[2]. Единственная возможность использовать их для сжатия цветных изображений - это возможность сжимать каждую цветовую компоненту в отдельности.

Нами были разработаны методы и средства сокращенного описания многоградационных черно-белых изображений [1],[2]. Эти методы реализуют так называемое преобразование компонент с интерполяцией по отсчетам (ПКИО) - иерархическую процедуру интерполяционного кодирования. Поскольку указанная процедура учитывает известные статистические свойства реальных изображений [3], она оказывается весьма эффективным средством сжатия этих изображений без потери информации. Поэтому было решено создать новые варианты методов ПКИО, позволяющие сжимать цветные изображения реального мира и точно восстанавливать их. В данной работе описываются основные результаты разработки и реализации этих методов.

### 2. СЖАТИЕ И ВОССТАНОВЛЕНИЕ МНОГОГРАДАЦИОННЫХ ЧЕРНО-БЕЛЫХ ИЗОБРАЖЕНИЙ

Прежде всего опишем вкратце, как выглядит метод ПКИО при работе с многоградационными черно-белыми изображениями (более подробное описание этого метода можно найти в [1],[2]). Блок-схема алгоритма сжатия черно-белого изображения методом ПКИО приведена на рис. 1. Алгоритм сжатия выглядит следующим образом:

1) компандированное изображение  $I_0$ , полученное путем сжатия динамического диапазона исходного черно-белого изображения  $I_s$  и заданное отсчетами яркости

---

Работа частично финансировалась в рамках гранта "Разработка оптико-цифровых, параллельных, рекурсивных способов реализации методов и алгоритмов адаптивной обработки двумерных сигналов и изображений" Российского фонда фундаментальных исследований

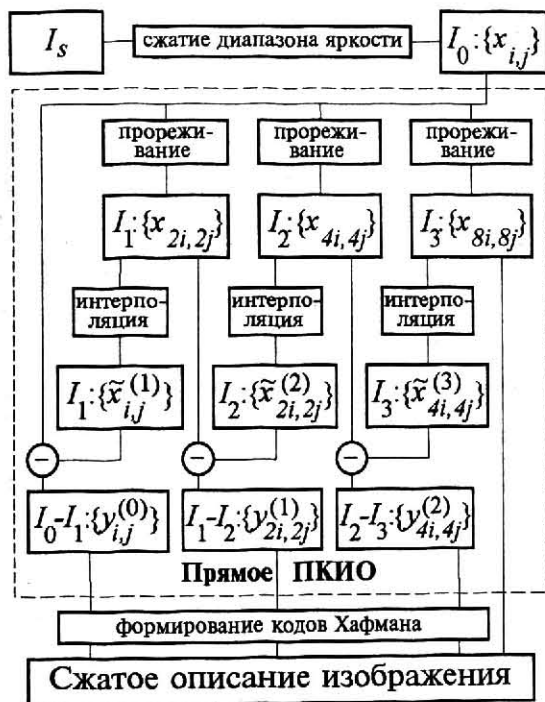


Рис. 1. Алгоритм кодирования, использующий ПКЦИО

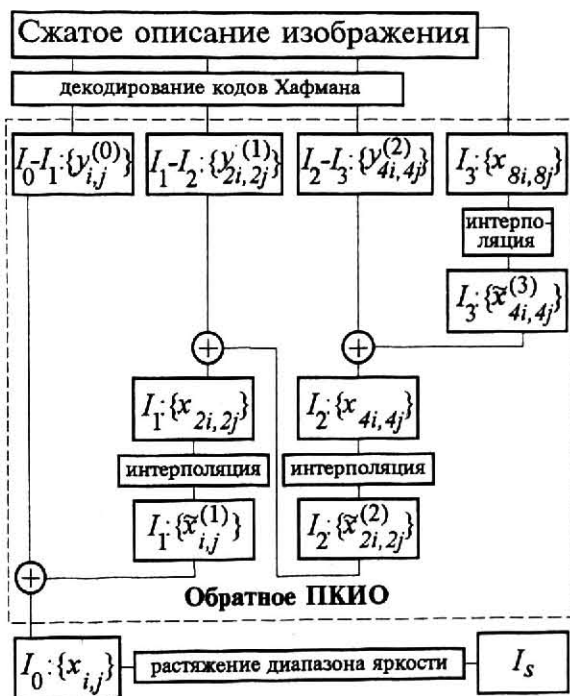


Рис. 2. Алгоритм декодирования, использующий ПКЦИО

$x_{ij}$ ,  $ij=0,1,2,\dots$ , разбивается на компоненты  $I_k$ ,  $k=1,2,3$ , содержащие его *прореженные* отсчеты  $x_{2^k i, 2^k j}$ ,  $ij=0,1,2,\dots$  (например, четные отсчеты четных строк, отсчеты 0,4,8,... строк 0,4,8,... и т.д.);

2) с помощью билинейной интерполяции формируются промежуточные отсчеты  $\tilde{x}^{(k)}_{2^{k-1}i, 2^{k-1}j}$ ,  $ij=0,1,2,\dots$  каждой из компонент  $I_k$  (т.е. вычисляются полусуммы и четверти сумм их прореженных отсчетов). Важно отметить, что  $\tilde{x}^{(k)}_{2^{k-1}i, 2^{k-1}j} = x_{2^k i, 2^k j}$ ;

3) составляются *разностные компоненты*  $I_0-I_1$ ,  $I_1-I_2$  и  $I_2-I_3$ ; компонента  $I_{k-1}-I_k$  задается отсчетами  $y^{(k-1)}_{2^{k-1}i, 2^{k-1}j} = x_{2^k i, 2^k j} - \tilde{x}^{(k)}_{2^{k-1}i, 2^{k-1}j}$ ,  $ij=0,1,2,\dots$ . Как показано в [1],[2], при определении  $y^{(k-1)}$  целесообразно использовать вместо обычного вычитания вычитание по модулю 256, то есть пользоваться формулой  $y^{(k-1)}_{2^{k-1}i, 2^{k-1}j} = (x_{2^k i, 2^k j} - \tilde{x}^{(k)}_{2^{k-1}i, 2^{k-1}j} + 128) \bmod 256$ . Четверть отсчетов каждой из дифференциальных компонент  $(y^{(k-1)}_{2^{k-1}i, 2^{k-1}j})$  всегда имеет значение 0 (или 128), и эти отсчеты нет необходимости хранить.

Описанная процедура приводит к разложению изображения  $I_0$  на четыре компоненты:  $I_0-I_1$ ,  $I_1-I_2$ ,  $I_2-I_3$  и  $I_3$ . Как и обычное пирамидальное представление [4], представление изображения этими компонентами можно в первом приближении считать разбиением на полосы различных пространственных частот. Так, компонента  $I_3$  описывает низкочастотные свойства изображения, компонента  $I_2-I_3$  - полосу несколько более высоких пространственных частот и т.д. Представление изображения компонентами ПКЦИО, однако, выгодно отличается от его пирамидального представления тем, что компоненты ПКЦИО имеют *тот же объем*, что и кодируемое изображение (размеры пирамидального представления, как известно, на 1/3 превышают объем изображения).

Благодаря хорошо известным свойствам изображений реального мира (в первую очередь - высокой корреляции отсчетов в пределах сравнительно протяженных гладких областей) процедура формирования разностных компонент ПКЦИО (приводящая к декорреляции этих отсчетов) порождает компоненты, которые имеют гораздо меньшую суммарную энтропию, чем исходное изображение. Поэтому результаты ПКЦИО легко сжать, применяя обычные средства статистического кодирования (например, код Хаффмана [5]). Отсчеты базовой компоненты  $I_3$  занимают всего 1/64 общего объема описания, а их энтропия велика. Поэтому указанные отсчеты целесообразно хранить непосредственно, не используя статистического кодирования.

Блок-схема алгоритма декодирования изображения, сжатого с помощью ПКИО, показана на рис.2. Первый этап декодирования сводится к расшифровке кодовых слов кода Хаффмана и точному восстановлению компонент  $I_3$ ,  $I_2-I_3$ ,  $I_1-I_2$  и  $I_0-I_1$ . Дальнейшее декодирование выполняется следующим образом:

1) посредством билинейной интерполяции (см. шаг 2 описанного выше алгоритма кодирования) восстанавливаются промежуточные отсчеты  $\tilde{x}^{(3)}$  компоненты  $I_3$  (напомним, что  $\tilde{x}^{(3)}_{8i,8j} = x_{8i,8j}$ );

2) к значениям отсчетов, найденным по окончании шага 1, добавляются значения отсчетов  $y^{(2)}_{4i,4j}$  разностной компоненты  $I_2-I_3$ . В результате точно восстанавливаются отсчеты  $x_{4i,4j} = (y^{(2)}_{4i,4j} + \tilde{x}^{(3)}_{4i,4j} - 128) \bmod 256$ ,  $i, j=0,1,2,\dots$  компоненты  $I_2$ ;

3) - 4) шаги 1 и 2 повторяются для отсчетов компонент  $I_2$  и  $I_1-I_2$ . При этом восстанавливается компонента  $I_1$ ;

5) - 6) еще одно повторение интерполяции и суммирования, на сей раз для компонент  $I_1$  и  $I_0-I_1$ , приводит к точному восстановлению компандированного изображения  $I_0$ ;

7) для восстановления исходного изображения  $I_s$  отсчеты изображения  $I_0$  подвергаются табличному преобразованию, восстанавливающему исходный диапазон яркости.

В отличие от параллельной процедуры кодирования, алгоритм декодирования является принципиально последовательным, поскольку компоненту  $I_2$ , компоненту  $I_1$  и изображение  $I_0$  приходится восстанавливать поочередно.

### 3. СЖАТИЕ И ВОССТАНОВЛЕНИЕ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

Успешное применение ПКИО для сжатия черно-белых изображений [1],[2] вдохновило нас на использование аналогичной процедуры для сокращенного описания цветных изображений реального мира без потери информации. Проще всего независимо сжимать методом ПКИО каждую из цветовых составляющих  $R$ ,  $G$  и  $B$  цветного изображения (рис. 3). Составляющие большинства реальных цветных изображений, как правило, сильно коррелированы; желательно использовать это обстоятельство для увеличения степени их сжатия. Способы декорреляции цветовых составляющих в случае кодирования с потерями информации хорошо известны: необходимо вычислять взвешенные суммы этих составляющих  $X = \alpha_1 R + \beta_1 G + \gamma_1 B$ ,  $Y = \alpha_2 R + \beta_2 G + \gamma_2 B$  и  $Z = \alpha_3 R + \beta_3 G + \gamma_3 B$ . Нетрудно найти оптимальный набор коэффициентов  $\{\alpha_k, \beta_k, \gamma_k\}$ , применение которого обеспечивает минимальную среднеквадратичную погрешность восстановления цветовых компонент (см., например, [3]). К сожалению, при необходимости восстанавливать изображения без потери информации использовать дробные коэффициенты почти невозможно, поскольку повторное квантование взвешенных сумм ведет к потере информации. Попытки работы с целыми коэффициентами вместо дробных посредством увеличения длины кодовых слов приводят к значительному расширению описания и к увеличению энтропии данных; в результате

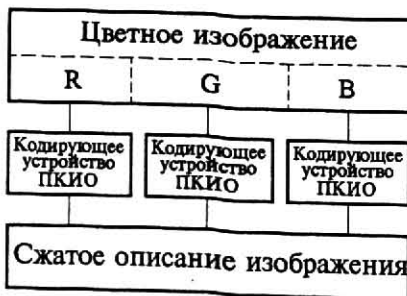


Рис. 3. Простейшая возможность кодирования цветных изображений

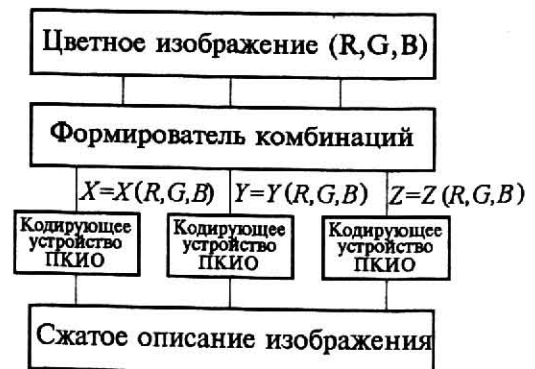


Рис. 4. Улучшенный метод кодирования цветных изображений

эффективность кодирования резко падает. Единственной реальной возможностью является применение усеченных взвешенных сумм (УВС) пар цветовых составляющих (можно, например, использовать УВС  $R+\beta_1G$  и  $B+\beta_2G$ ). Один из весовых коэффициентов в каждой сумме должен равняться единице. Если в дополнение к УВС располагать информацией о составляющей с дробным весом (в приведенном примере -  $G$ ), то можно точно восстановить составляющие с единичными весами ( $R$  и  $B$ ). Полный перечень возможных УВС будет приведен ниже. Если вычислять УВС по модулю 256 (теоретические предпосылки целесообразности этого можно найти в [6]), то объем описания цветного изображения с помощью цветовых составляющих и их УВС ( $X$ ,  $Y$  и  $Z$ ) окажется равным суммарному объему цветовых составляющих  $R$ ,  $G$  и  $B$ . Каждую из составляющих  $X$ ,  $Y$  и  $Z$  можно сжать с применением какого-либо метода кодирования, например, описанной выше процедуры ПКЦИО (рис. 4).

Если использовать в качестве средства сжатия метод ПКЦИО, то можно предложить и другой вариант кодирования цветных изображений. В самом деле, можно находить взвешенные суммы и после реализации ПКЦИО, вычисляя их для триад разностных компонент, полученных на каждом из трех иерархических уровней этого преобразования. Будем называть этот вариант "методом ПКЦИО с комбинированием компонент". Он имеет ряд преимуществ по сравнению с непосредственным комбинированием цветовых составляющих изображений. Во-первых, результаты моделирования указывают на его более высокую эффективность; повышение эффективности объясняется лучшей адаптацией комбинаций компонент ПКЦИО различных уровней к свойствам составляющих кодируемого цветного изображения в различных диапазонах пространственных частот. Кроме того, при применении метода ПКЦИО с комбинированием компонент отпадает необходимость согласовывать динамические диапазоны этих компонент, поскольку функции распределения разностных компонент ПКЦИО всегда имеют сходные быстро спадающие распределения с максимумом в нуле (или в точке 128 при вычислениях по модулю 256). В связи с наличием указанных преимуществ именно метод ПКЦИО с комбинированием компонент будет являться основным объектом дальнейших исследований.

### 3.1. Метод ПКЦИО с комбинированием компонент

Блок-схема алгоритма, реализующего метод ПКЦИО с комбинированием компонент, показана на рис. 5. Этот алгоритм имеет следующий вид:

1) сжимается динамический диапазон цветовых составляющих  $R$ ,  $G$  и  $B$  кодируемого цветного изображения. Такая предварительная процедура необходима для предотвращения возможного увеличения числа значений отсчетов разностных компонент после ПКЦИО (как показано в [1],[2], оно может привести к резкому снижению эффективности кодирования в случае, если число значений отсчетов реальных цветовых составляющих значительно меньше максимально возможного числа этих значений (256);

2) каждая из составляющих, полученных после выполнения первого шага ( $R_0$ ,  $G_0$  и  $B_0$ ) подвергается ПКЦИО. В результате кодируемое изображение представляется тремя базовыми компонентами -  $R_3$ ,  $G_3$  и  $B_3$  и тремя триадами разностных компонент -  $\{\Delta R_{23}=R_2-R_3, \Delta G_{23}=G_2-G_3, \Delta B_{23}=B_2-B_3\}$ ,  $\{\Delta R_{12}=R_1-R_2, \Delta G_{12}=G_1-G_2, \Delta B_{12}=B_1-B_2\}$  и  $\{\Delta R_{01}=R_0-R_1, \Delta G_{01}=G_0-G_1, \Delta B_{01}=B_0-B_1\}$ , которые отвечают различным диапазонам пространственных частот;

3) триады разностных компонент комбинируются с помощью трех формирователей комбинаций. Для этого вычисляются все возможные УВС этих компонент и для каждого из трех иерархических уровней находится наилучшая комбинация компонент и УВС (эта процедура будет подобно описана ниже). В результате формируются три триады новых компонент -  $\{X_{23}(\Delta R_{23}, \Delta G_{23}, \Delta B_{23}), Y_{23}(\Delta R_{23}, \Delta G_{23}, \Delta B_{23}), Z_{23}(\Delta R_{23}, \Delta G_{23}, \Delta B_{23})\}$ ,  $\{X_{12}(\Delta R_{12}, \Delta G_{12}, \Delta B_{12}), Y_{12}(\Delta R_{12}, \Delta G_{12}, \Delta B_{12}), Z_{12}(\Delta R_{12}, \Delta G_{12}, \Delta B_{12})\}$  и  $\{X_{01}(\Delta R_{01}, \Delta G_{01}, \Delta B_{01}), Y_{01}(\Delta R_{01}, \Delta G_{01}, \Delta B_{01}), Z_{01}(\Delta R_{01}, \Delta G_{01}, \Delta B_{01})\}$ ;

4) для компактного представления компонент  $X_i$ ,  $Y_i$  и  $Z_i$ , как и при работе с черно-белыми изображениями, используется код Хаффмана. Максимальное сжатие при этом достигается в случае, когда для кодирования каждой из девяти компонент используется своя кодовая книга. Поскольку объем базовых компонент  $R_3$ ,  $G_3$  и  $B_3$  очень мал

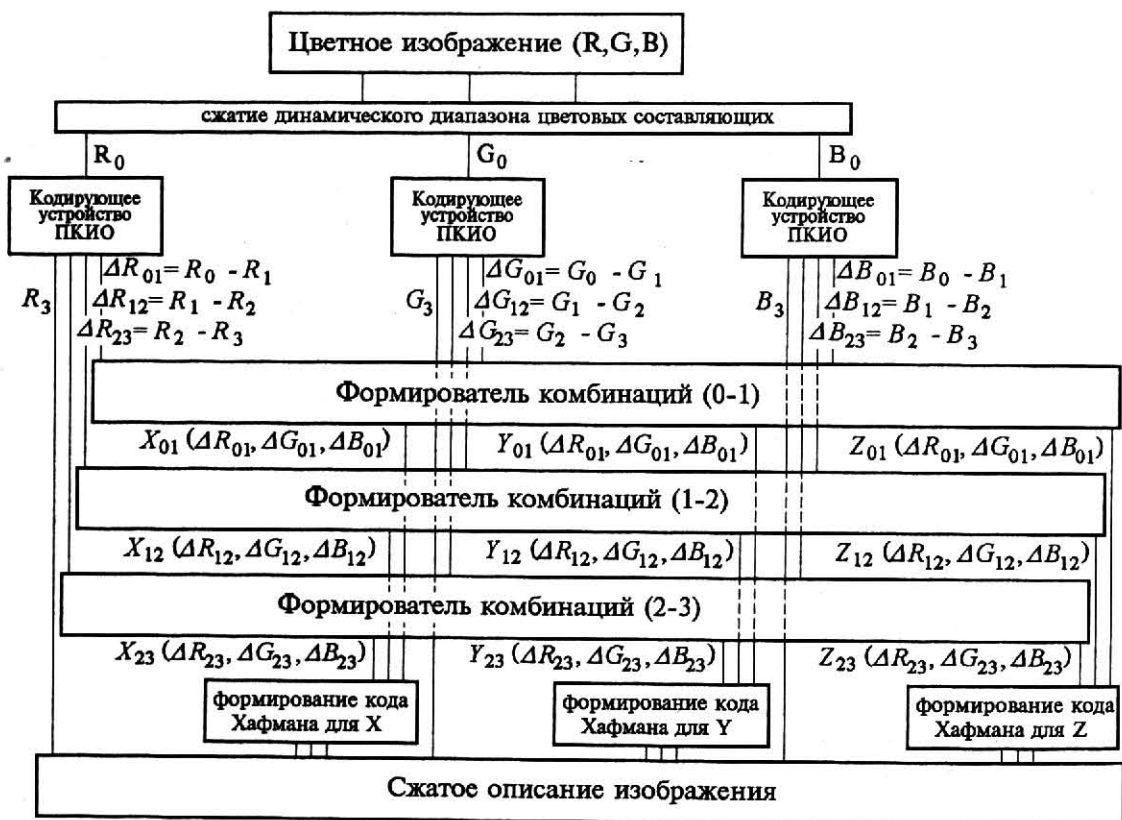


Рис. 5. Алгоритм кодирования, реализующий метод ПКМО с комбинированием компонент



Рис. 6. Алгоритм декодирования изображения, сжатого методом ПКМО с комбинированием компонент

(1/64 общего объема описания), а их энтропии обычно велики, лучше всего непосредственно хранить отсчеты этих компонент, не применяя для них статистического кодирования.

Процедура восстановления цветного изображения, сжатого методом ПККО с комбинированием компонент (рис. 6) фактически является инверсией процедуры сжатия, показанной на рис. 5. Ее первым шагом является декодирование кодовых слов Хаффмана и восстановление компонент  $X_{23}, X_{12}, X_{01}, Y_{23}, Y_{12}, Y_{01}, Z_{23}, Z_{12}$  и  $Z_{01}$ . Затем в трех блоках восстановления находятся разностные компоненты  $\Delta R_{23}, \Delta R_{12}, \Delta R_{01}, \Delta G_{23}, \Delta G_{12}, \Delta G_{01}, \Delta B_{23}, \Delta B_{12}$  и  $\Delta B_{01}$  для всех цветовых составляющих. Применение обратного ПККО (см. рис. 2) к каждой из трех триад разностных компонент (и соответствующим базовым компонентам -  $R_3, G_3$  или  $B_3$ ) приводит к точному восстановлению комбинированных цветовых составляющих  $R_0, G_0$  и  $B_0$ . После этого достаточно подвергнуть их трем табличным преобразованиям, восстанавливающим динамические диапазоны, чтобы восстановить исходные цветовые составляющие  $R, G$  и  $B$ .

### 3.2. Формирователи комбинаций и блоки восстановления компонент

Совершенно очевидно, что эффективность показанного на рис. 5 алгоритма сжатия всецело зависит от способа комбинирования компонент в формирователях комбинаций. Как уже отмечалось выше, единственной доступной возможностью является вычисление для этих компонент УВС вида  $U+cV$ ; для того, чтобы определить каждую такую УВС, необходимо найти один коэффициент.

Список возможных комбинаций  $X(\Delta R, \Delta G, \Delta B), Y(\Delta R, \Delta G, \Delta B)$  и  $Z(\Delta R, \Delta G, \Delta B)$  трех компонент  $\Delta R, \Delta G$  и  $\Delta B$  и их УВС приведен в табл. 1 (напомним, что УВС естественно вычислять по модулю 256). Легко видеть, что приведенные в таблице шестнадцать комбинаций исчерпывают их перечень (с точностью до перестановок); комбинации, не включенные в таблицу (например,  $\{X=\Delta R+\beta\Delta G, Y=\Delta G+\gamma\Delta B, Z=\Delta B+\alpha\Delta R\}$ ) порождают системы зависимых уравнений, из которых нельзя найти  $\Delta R, \Delta G$  и  $\Delta B$ . Для обеспечения максимальной эффективности нашего алгоритма сжатия, в котором используется статистическое кодирование, необходимо выбрать на каждом из трех иерархических уровней  $i$  (01, 12 или 23) такую комбинацию  $n_i$  и такие коэффициенты  $\alpha_i, \beta_i$  или  $\gamma_i$ , чтобы получить минимально возможную суммарную энтропию  $H_{n_i} = H_{X_{n_i}} + H_{Y_{n_i}} + H_{Z_{n_i}}$ . К сожалению, нет возможности найти энтропию взвешенной суммы, исходя из статистики слагаемых. Поэтому невозможно и найти оптимальные коэффициенты. В нашем распоряжении имеется лишь субоптимальная процедура, гарантирующая получение УВС с минимальной дисперсией (что совсем не всегда приводит к наилучшему статистическому сжатию этих УВС). Субоптимальное значение весового коэффициента  $c$  в УВС  $W=U+cV$  определяется формулой

$$c = -\chi_{UV} / D_V, \quad (1)$$

где  $\chi_{UV} = E(UV) - E(U)E(V)$  - ковариация компонент  $U$  и  $V$ ,  $D_V = E(V^2) - (E(V))^2$  - дисперсия компоненты  $V$ ,  $E(*)$  - математическое ожидание случайной величины  $*$ . При субоптимальном выборе  $c$  достигается минимальное значение дисперсии  $D_W$  УВС  $W$ :

$$D_W = D_U(1 - \kappa^2), \quad (2)$$

где  $D_U = E(U^2) - (E(U))^2$  - дисперсия компоненты  $U$ ,  $\kappa = \chi_{UV} / \sqrt{D_U D_V}$  - коэффициент корреляции компонент  $U$  и  $V$ .

Применение описанной субоптимальной процедуры приводит к следующему алгоритму адаптации формирователей комбинаций каждого из трех иерархических уровней  $i$  (01, 12 и 23) к статистическим свойствам разностных компонент кодируемого цветного изображения (рис. 7а):

1) анализируется статистика разностных компонент  $\Delta R_i, \Delta G_i$  и  $\Delta B_i$  и находятся их дисперсии  $D_{\Delta R_i}, D_{\Delta G_i}$  и  $D_{\Delta B_i}$ , а также их попарные ковариации  $\chi_{\Delta R_i \Delta G_i}, \chi_{\Delta R_i \Delta B_i}$  и  $\chi_{\Delta G_i \Delta B_i}$ ;

2) из выражений (1) и (2) для каждой из перечисленных в табл. 1 возможных комбинаций  $n_i$  находятся субоптимальные значения коэффициентов и отвечающее им значение суммарной дисперсии  $D_{n_i} = D_{X_{n_i}} + D_{Y_{n_i}} + D_{Z_{n_i}}$ ;

Таблица 1. Возможные комбинации компонент и их УВС

$n$	$X(\Delta R, \Delta G, \Delta B)$	$Y(\Delta R, \Delta G, \Delta B)$	$Z(\Delta R, \Delta G, \Delta B)$
1	$\Delta R$	$\Delta G$	$\Delta B$
2	$\Delta R + \beta \Delta G$	$\Delta G$	$\Delta B$
3	$\Delta R + \gamma \Delta B$	$\Delta G$	$\Delta B$
4	$\Delta R$	$\Delta G + \alpha \Delta R$	$\Delta B$
5	$\Delta R$	$\Delta G + \gamma \Delta B$	$\Delta B$
6	$\Delta R$	$\Delta G$	$\Delta B + \alpha \Delta R$
7	$\Delta R$	$\Delta G$	$\Delta B + \beta \Delta G$
8	$\Delta R$	$\Delta G + \alpha' \Delta R$	$\Delta B + \alpha'' \Delta R$
9	$\Delta R$	$\Delta G + \alpha \Delta R$	$\Delta B + \beta \Delta G$
10	$\Delta R$	$\Delta G + \gamma \Delta B$	$\Delta B + \alpha \Delta R$
11	$\Delta R + \beta' \Delta G$	$\Delta G$	$\Delta B + \beta'' \Delta G$
12	$\Delta R + \beta \Delta G$	$\Delta G$	$\Delta B + \alpha \Delta R$
13	$\Delta R + \gamma \Delta B$	$\Delta G$	$\Delta B + \beta \Delta G$
14	$\Delta R + \gamma' \Delta B$	$\Delta G + \gamma'' \Delta B$	$\Delta B$
15	$\Delta R + \gamma \Delta B$	$\Delta G + \alpha \Delta R$	$\Delta B$
16	$\Delta R + \beta \Delta G$	$\Delta G + \gamma \Delta B$	$\Delta B$

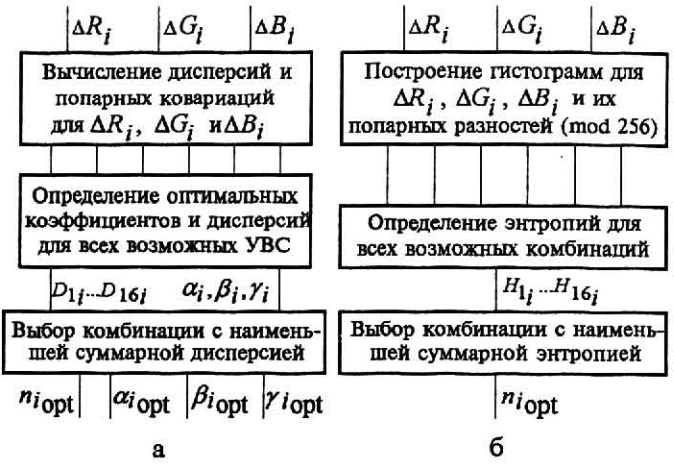


Рис. 7. Адаптация формирователей комбинаций к содержимому цветного изображения: а - при применении УВС с оптимальными весами; б - при использовании попарных разностей

3) выбирается комбинация  $n_{i,opt}$ , которой отвечает наименьшее значение  $D_{n_i}$ .

В результате адаптации для каждого иерархического уровня определяется номер оптимальной комбинации и величины одного или двух отвечающих этой комбинации коэффициентов, минимизирующих суммарную дисперсию.

Поскольку реализация описанного метода адаптации требует большого объема вычислений, а получаемые оценки статистических характеристик невозможно использовать при последующем кодировании компонент, интересно рассмотреть и более простую процедуру адаптации. Она предполагает все коэффициенты в УВС априори равными -1, то есть в ее рамках объектами статистического кодирования являются комбинации разностных компонент и их попарных разностей по модулю 256 (см. табл. 2). При этом алгоритм адаптации для иерархического уровня  $i$  выглядит следующим образом (рис. 7б):

1) строятся шесть гистограмм распределений для разностных компонент  $\Delta R_i, \Delta G_i$  и  $\Delta B_i$ , а также для их попарных разностей (по модулю 256):  $f(\Delta R_i), f(\Delta G_i), f(\Delta B_i), f(\Delta R_i - \Delta G_i), f(\Delta G_i - \Delta B_i)$  и  $f(\Delta R_i - \Delta B_i)$ ;

2) по этим гистограммам находятся 6 значений энтропии  $H = -\sum_k f(k) \cdot \log_2 f(k)$ ;

3) для каждой из 16 перечисленных в левой части табл. 2 комбинаций  $n_i$  определяется суммарная энтропия  $H_{n_i} = H_{X_{n_i}} + H_{Y_{n_i}} + H_{Z_{n_i}}$ ;

4) выбирается комбинация  $n_{i,opt}$ , которой отвечает наименьшее значение  $H_{n_i}$ . Описанный способ адаптации минимизирует суммарную длину кода Хаффмана для выбранной триады компонент  $X_{n_i}, Y_{n_i}$  и  $Z_{n_i}$ ; это означает, что он действительно является оптимальным. Гистограммы, построенные на этапе адаптации, могут быть использованы при построении кодовых книг кода Хаффмана.

Проведенное моделирование показало, что работа с попарными разностями компонент при сжатии цветных изображений реального мира методом ПКЮ ухудшает степень сжатия менее, чем на 1% по сравнению с применением УВС. Более того, оптимальные значения коэффициентов УВС в большинстве практических случаев оказываются близкими к -1. При этом алгоритм адаптации рис. 7б, работающий с попарными разностями, иногда оказывается даже более эффективным, чем алгоритм рис. 7а, использующий УВС, поскольку процедура поиска коэффициентов является лишь субоптимальной. Таким образом, на практике целесообразно вычислять в формирователях комбинаций попарные разности компонент и использовать оптимальную процедуру адаптации рис. 7б.

Структура блоков, обеспечивающих восстановление компонент  $\Delta R_i, \Delta G_i$  и  $\Delta B_i$  в декодирующем устройстве по номеру выбранной комбинации  $n_{i,opt}$  и компонентам  $X_{n_i}, Y_{n_i}$

Таблица 2. Возможные комбинации компонент и их попарных разностей по модулю 256 (левая часть); аналитические решения, позволяющие восстановить  $\Delta R$ ,  $\Delta G$  и  $\Delta B$  (правая часть)

$n$	$X(\Delta R, \Delta G, \Delta B)$	$Y(\Delta R, \Delta G, \Delta B)$	$Z(\Delta R, \Delta G, \Delta B)$	$\Delta R(X, Y, Z)$	$\Delta G(X, Y, Z)$	$\Delta B(X, Y, Z)$
1	$\Delta R$	$\Delta G$	$\Delta B$	$X$	$Y$	$Z$
2	$\Delta R - \Delta G$	$\Delta G$	$\Delta B$	$X + Y$	$Y$	$Z$
3	$\Delta R - \Delta B$	$\Delta G$	$\Delta B$	$X + Z$	$Y$	$Z$
4	$\Delta R$	$\Delta G - \Delta R$	$\Delta B$	$X$	$Y + X$	$Z$
5	$\Delta R$	$\Delta G - \Delta B$	$\Delta B$	$X$	$Y + Z$	$Z$
6	$\Delta R$	$\Delta G$	$\Delta B - \Delta R$	$X$	$Y$	$Z + X$
7	$\Delta R$	$\Delta G$	$\Delta B - \Delta G$	$X$	$Y$	$Z + Y$
8	$\Delta R$	$\Delta G - \Delta R$	$\Delta B - \Delta R$	$X$	$Y + X$	$Z + X$
9	$\Delta R$	$\Delta G - \Delta R$	$\Delta B - \Delta G$	$X$	$Y + X$	$Z + Y + X$
10	$\Delta R$	$\Delta G - \Delta B$	$\Delta B - \Delta R$	$X$	$Y + Z + X$	$Z + X$
11	$\Delta R - \Delta G$	$\Delta G$	$\Delta B - \Delta G$	$X + Y$	$Y$	$Z + Y$
12	$\Delta R - \Delta G$	$\Delta G$	$\Delta B - \Delta R$	$X + Y$	$Y$	$Z + X + Y$
13	$\Delta R - \Delta B$	$\Delta G$	$\Delta B - \Delta G$	$X + Z + Y$	$Y$	$Z + Y$
14	$\Delta R - \Delta B$	$\Delta G - \Delta B$	$\Delta B$	$X + Z$	$Y + Z$	$Z$
15	$\Delta R - \Delta B$	$\Delta G - \Delta R$	$\Delta B$	$X + Z$	$Y + X + Z$	$Z$
16	$\Delta R - \Delta G$	$\Delta G - \Delta B$	$\Delta B$	$X + Y + Z$	$Y + Z$	$Z$

и  $Z_{n_p}$  совершенно очевидна. В этих блоках должно определяться решение системы уравнений, отвечающей  $n_{i_{opt}}$ . Нетрудно найти такие решения аналитически для всех возможных комбинаций. При работе с попарными разностями процедура восстановления сводится к суммированию (по модулю 256) компонент  $X_{n_p}$ ,  $Y_{n_i}$  и  $Z_{n_i}$  (см. правую часть табл. 2).

### 3.3. Дополнительная информация

При оценке реальной эффективности предложенного метода сжатия цветных изображений не следует забывать о необходимости хранить некоторое количество дополнительной информации. Во-первых, компоненты  $X$ ,  $Y$  и  $Z$  на каждом иерархическом уровне имеют различные статистические свойства. Поэтому степень сжатия оказывается максимальной в случае, когда для кодирования компонент  $X_{01}$ ,  $X_{12}$ ,  $X_{23}$ ,  $Y_{01}$ ,  $Y_{12}$ ,  $Y_{23}$ ,  $Z_{01}$ ,  $Z_{12}$  и  $Z_{23}$  используется 9 различных кодовых книг Хафмана. Для декодирования кодов Хафмана необходимо иметь информацию об этих кодовых книгах. Каждая из них содержит 256 кодовых слов, длина которых переменна и может составлять до 4 байтов. При непосредственном хранении объем книги доходит до 1184 байтов [1],[2]. Поскольку вид кодовой книги Хафмана неоднозначно определяется статистикой кодируемых данных, можно вместо нее хранить гистограмму их распределения. В нашем случае эти гистограммы имеют выраженный максимум, и их можно компактно описать с помощью простого кода, похожего на код Хафмана [1],[2]. При этом для хранения всех девяти гистограмм требуется лишь около 900 байтов дополнительной информации.

Помимо информации о гистограммах, для правильного декодирования необходимо иметь информацию о трех таблицах, использованных для сжатия динамических диапазонов цветовых составляющих на первом этапе процедуры кодирования. Метод компактного описания этих таблиц описан в [1],[2]; при его применении объем их описания составляет всего 100-180 байтов.

Для работы блоков восстановления компонент необходимо знать номера наилучших комбинаций на каждом из иерархических уровней. Эти номера можно непосредственно запомнить, используя 1,5 байта памяти.

Таким образом, общий объем дополнительной информации составляет около 1000 байтов.

## 4. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Описанный выше метод сжатия цветных изображений был реализован в виде двух завершающих программ (CPRESS и CUNPRESS) для IBM-совместимых персональных ЭВМ. С помощью этих программ можно сжимать (и восстанавливать без потери



информации) цветные изображения любых размеров, заданные в виде трех файлов с цветовыми составляющими  $R$ ,  $G$  и  $B$  в байтовом формате. Большая часть текста этих программ написана на языке "Си"; для повышения их быстродействия в некоторых местах использовался также язык "Ассемблер".

Для оценки возможностей предложенного метода использовались высококачественные цветные изображения (изображения картин с высокой насыщенностью и детальностью, качественные цветные фотографии и спектрзональное спутниковое изображение). Их энтропия была достаточно высокой (до 21,9 бит). В ходе экспериментов определялся коэффициент сжатия  $S$  - отношение длины файла, получаемого в результате кодирования, к суммарной длине трех файлов, изначально задающих цветное изображение (чем меньше  $S$ , тем больше степень сжатия). Для сравнения мы сжимали каждую из трех составляющих  $R$ ,  $G$  и  $B$  цветного изображения с помощью программы PRESS (которая реализует кодирование черно-белых изображений методом ПКИО). Это отвечало простейшей схеме кодирования, приведенной на рис. 3. Возможности программы PRESS были изучены в [1],[2]; ее применение обеспечивало сжатие черно-белых изображений в 2-3 раза и являлось на 10-30% более эффективным, чем использование обычных средств архивации (программ PKZIP, LHA и ARJ).

Таблица 3. Экспериментальные результаты

Изображение	Объем исходного описания, байт	Размеры изображения	Энтропия, бит	Коэффициент сжатия $S$ , %		$S_{3 \times \text{PRESS}} - S_{\text{CPRESS}}$ , %
				3xPRESS	CPRESS	
"Фрукты"	786432	512 x 512	21,9	61	54	7
"Барабанщики"	691200	512 x 450	21,3	59	57	2
"Христос"	2674305	1005 x 887	20,6	60	57	3
"Святой"	1874763	653 x 957	21,2	73	68	5
"Одесса"	6140928	1999 x 1024	15,6	40	39	1

Результаты измерений коэффициентов сжатия приведены в табл. 3. Они указывают на то, что применение программы CPRESS для сжатия цветных изображений до 7% более эффективно, чем трехкратное применение программы PRESS для сжатия их цветовых составляющих. Получаемый выигрыш более высок для изображений с сильно коррелированными составляющими (например, для изображения "Фрукты"). Корреляция составляющих в спектрзональном изображении "Одесса" значительно ниже; поэтому и выигрыш для него оказывается незначительным. Абсолютные значения коэффициентов сжатия при применении программы CPRESS оказываются достаточно малыми; объем сжатого изображения может достигать до 1/3 объема его исходного описания.

Высокая эффективность программ CPRESS и CUNPRESS означает наличие широких возможностей их практического использования в системах обработки изображений и в базах видеоданных различных типов. Эти программы могут быть предоставлены в распоряжение всех потенциальных потребителей.

## ЛИТЕРАТУРА

1. Бокштейн И.М. Метод сжатия изображений без потери информации. Автометрия, 1994, № 1.
2. Bockstein I.M. A method of lossless image compression. Pattern recognition and image analysis, 1993, Vol. 3, No. 2, pp. 92-98.
3. Netravali A.N. and Haskell B.G. Digital Pictures: Representation and Compression. N.Y.: Plenum, 1988.
4. Burt P.J. and Adelson E.H. The laplacian pyramid as a compact image code. IEEE Transactions on Communications, 1983, Vol. COM-31, No. 4.
5. Huffman D.A. A method for the construction of minimum redundancy codes. Proceedings of the IRE, 1952, Vol.40, No. 9.
6. Einarsson G. An improved implementation of predictive coding compression. IEEE Transactions on Communications, 1991, Vol. COM-39, No. 2.