РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
АВТОМЕТРИЯ
№ 1 1994

УДК 621.397.2 : 519.685

И. М. Бокштейн
(Москва)

МЕТОД СЖАТИЯ ОПИСАНИЙ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРИ ИНФОРМАЦИИ

Предлагается метод сжатия описаний изображений, обеспечивающий их точное восстановление при 2-, 3-кратном сжатии. Его основой является преобразование компонент с интерполяцией по отсчетам. Рассматриваются конкретные проблемы, связанные с практической реализацией этого метода. Описываются эффективные способы хранения дополнительной информации, необходимой для декодирования. Эксперименты показывают, что программы, реализующие описанный метод, являются гораздо более эффективными, чем существующие средства архивации данных.

Введение. Значение изображений как источника информации невозможно переоценить. В настоящее время изображения широко применяются в медицине, экологических исследованиях, дистанционном зондировании поверхности Земли и многих других областях. Большое количество и большие размеры изображений делают необходимым создание средств их компактного хранения; применение таких средств позволяет значительно снизить требования к внешним запоминающим устройствам. Существующие программы архивации, обеспечивающие компактное хранение данных без потери информации, ориентированы в основном на работу с текстами и поэтому крайне неэффективны для изображений.

Зная статистические особенности изображений, можно создать методы и средства их компактного описания. Работы в этой области ведутся уже многие годы (см., например, [1, 2]), однако при этом обычно не ставится задача *точного* восстановления яркости всех закодированных отсчетов. При создании систем цифровой обработки изображений и баз видеоданных необходимость точного восстановления выходит на первый план.

Многие из существующих методов эффективного кодирования изображений (например, подавляющая часть трансформационных методов [1]) в принципе не обеспечивают точного восстановления, так как процедура преобразования существенно увеличивает объем описания. Другие методы, например методы кодирования с полосовой фильтрацией (subband coding) [3], могут обеспечить точное восстановление, но сложны в реализации.

Наиболее целесообразным для сжатия описаний изображений без потери информации представляется применение методов кодирования с предсказанием [2] и интерполяционных методов. В [4, 5] нами был предложен и теоретически проанализирован интерполяционный метод кодирования изображений, реализующий так называемое преобразование компонент с интерполяцией по отсчетам. Он оказался весьма эффективным средством сжатия описаний высококачественных изображений реального мира при небольших погрешностях восстановления. Настоящая работа посвящена анализу возможностей применения несколько иной версии этого метода для сокращенного описания таких изображений с последующим *точным* восстановлением.

1. **Кодирование и восстановление изображений.** *Алгоритм кодирования.* Основной частью предлагаемого алгоритма кодирования изображений, блок-

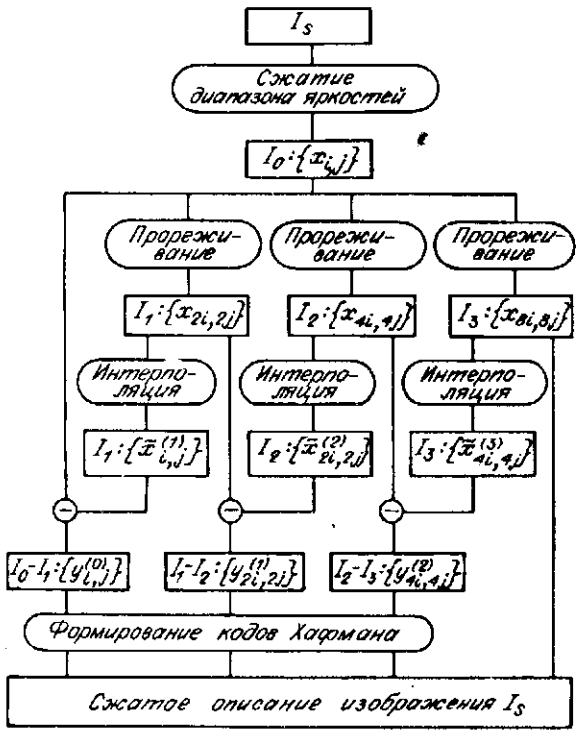

Рис. 1

схема которого показана на рис. 1, является описанное в [4] преобразование компонент с интерполяцией по отсчетам, сводящееся к следующему:

1. Компандированное изображение I_0, полученное после сжатия диапазона яркостей кодируемого изображения I_s (подробнее об этом см. в разд. 2) и заданное своими отсчетами $x_{i,j}$, $i, j = 0, 1, 2, \ldots$, разбивается на компоненты I_k, $k = 1, 2, 3$, содержащие его «прореженные» отсчеты $x_{2^k i, 2^k j}$, $i, j = 0, 1, 2, \ldots$ (эти отсчеты отмечены кружками на рис. 2). Причины выбора $k_{\max} = 3$ приведены в [4].

2. Путем билинейной интерполяции формируются промежуточные отсчеты $\tilde{x}^{(k)}_{2^{k-1} i, 2^{k-1} j}$, $i, j = 0, 1, 2, \ldots$ каждой из компонент I_k (эти отсчеты показаны на рис. 2 в виде квадратов). Билинейная интерполяция при этом фактически сводится к вычислению полусуммы двух «прореженных» отсчетов, если промежуточный отсчет (светлый квадрат на рис. 2) лежит между ними, или четверти суммы четырех отсчетов, когда промежуточный отсчет (заштрихованный квадрат на рис. 2) находится в центре образованного «прореженными» отсчетами квадрата. Важно отметить, что при таком способе построения промежуточных отсчетов $\tilde{x}^{(k)}_{2^k i, 2^k j} \equiv x_{2^k i, 2^k j}$.

3. Составляются разностные компоненты $I_0 - I_1$, $I_1 - I_2$ и $I_2 - I_3$; компонента $I_{k-1} - I_k$ при этом задается отсчетами $y^{(k-1)}_{2^{k-1} i, 2^{k-1} j} = x_{2^{k-1} i, 2^{k-1} j} - \tilde{x}^{(k)}_{2^{k-1} i, 2^{k-1} j}$, $i, j = 0, 1, 2, \ldots$. Четверть отсчетов каждой из разностных компонент, а именно $y^{(k-1)}_{2^k i, 2^k j}$, всегда имеет тождественно нулевые значения, и их нет необходимости хранить.

Рис. 2

Описанное преобразование приводит к разложению изображения I_0 на компоненты $I_0 - I_1$, $I_1 - I_2$, $I_2 - I_3$ и I_3. Так же как и обычное

пирамидальное представление [6], покомпонентное представление можно грубо считать разбиением по различным диапазонам пространственных частот; так, компонента I_3 описывает низкочастотные свойства изображения, компонента $I_2 - I_3$ — полосу несколько более высоких частот и т. д. Однако предлагаемое представление выгодно отличается от пирамидального, поскольку *имеет тот же объем*, что и кодируемое изображение* (пирамидальное представление, как известно, *расширяет* объем описания примерно на треть). В силу известных свойств изображений (наличия значительной корреляции между близкими отсчетами, принадлежащими к одной и той же области) описанная процедура формирования разностных компонент (декоррелирующая их отсчеты по сравнению с отсчетами изображений) приводит к тому, что эти компоненты всегда имеют значительно более низкую энтропию, чем кодируемое изображение. Поэтому сжатие покомпонентного представления может быть достигнуто за счет применения обычных средств статистического кодирования (например, кода Хафмана [7]). Поскольку целью является получение максимального сжатия, этот код следует строить, исходя из статистических свойств конкретной реализации каждой из разностных компонент кодируемого изображения. Отсчеты опорной компоненты I_3 составляют всего 1/64 общего объема описания, а их энтропия сравнительно высока; почти без ущерба для результата эту компоненту можно хранить непосредственно, не подвергая статистическому кодированию.

Алгоритм декодирования. Блок-схема алгоритма восстановления изображений, закодированных описанным выше способом, приведена на рис. 3. Первым этапом декодирования является расшифровка кодов Хафмана для компонент I_3, $I_2 - I_3$, $I_1 - I_2$ и $I_0 - I_1$ (в случае отсутствия потерь при хранении данных эти компоненты восстанавливаются *точно*). Дальнейшее декодирование сводится к следующему:

1) путем билинейной интерполяции (см. п. 2 рассмотренного выше алгоритма кодирования) формируются промежуточные отсчеты $\tilde{x}^{(3)}_{4i,\,4j}$ компоненты I_3 (напомним, что $\tilde{x}^{(3)}_{8i,\,8j} \equiv x_{8i,\,8j}$);

2) к полученным значениям добавляются значения отсчетов разностной компоненты $I_2 - I_3$, $y^{(2)}_{4i,\,4j}$; в результате *точно* восстанавливаются отсчеты $x_{4i,\,4j} = y^{(2)}_{4i,\,4j} + \tilde{x}^{(3)}_{4i,\,4j}$, $i, j = 0, 1, 2, \ldots$, «прореженной» компоненты I_2;

3—4) указанные в пп. 1 и 2 действия повторяются для отсчетов компонент I_2 и $I_1 - I_2$; при этом восстанавливается компонента I_1;

5—6) третье повторение интерполяции и суммирования, на сей раз для компонент I_1 и $I_0 - I_1$, позволяет *точно* восстановить компандированное изображение I_0;

7) для декодирования исходного изображения I_s значения отсчетов изображения I_0 подвергаются табличному преобразованию, восстанавливающему исходный диапазон яркостей (см. разд. 2).

В отличие от *параллельной* процедуры кодирования алгоритм восстановления является принципиально *последовательным*: компоненты I_2, I_1 и изображение I_0 приходится восстанавливать поочередно.

2. Особенности реализации алгоритмов кодирования и восстановления. *Разрядность представления.* Будем считать, что отсчеты изображений I_s и I_0 представлены, как это обычно принято, 8-разрядным двоичным кодом. Зададимся вопросом: какова в этом случае разрядность представления отсчетов компонент?

Как «прореженные» компоненты, так и их промежуточные отсчеты имеют ту же разрядность, что и исходное изображение. Разностные компоненты,

---

* В самом деле, пусть исходное изображение содержит $M \times N$ отсчетов. Пренебрегая краевыми эффектами, можно считать, что «прореженная» компонента I_k состоит из $M/2^k \times N/2^k$ отсчетов. Разностная компонента $I_{k-1} - I_k$ описывается $M/2^{k-1} \times N/2^{k-1}$ отсчетами, однако 1/4 этих отсчетов — тождественно нулевые. Поэтому число *содержательных* отсчетов этой компоненты равно $3/4(M/2^{k-1} \times N/2^{k-1}) = 3MN/4^k$, а общее число отсчетов всех четырех компонент составляет $3MN(1/4 + 1/16 + 1/64) + MN/64 = MN$.

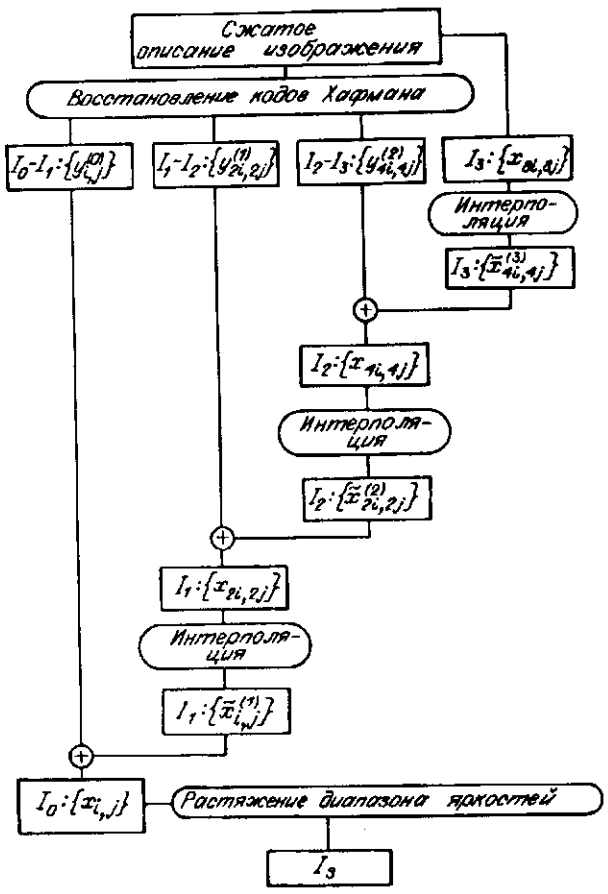

Рис. 3

однако, состоят из 9-разрядных отсчетов, ибо разность двух 8-разрядных чисел может находиться в диапазоне (±255). Это означает, вопреки изложенному ранее, *увеличение* объема покомпонентного представления на 1/8 по сравнению с исходным изображением. Устранить этот неприятный эффект можно, если в п. 3 алгоритма кодирования заменить обычное вычитание вычитанием по модулю 256 и определять значения $y^{(k-1)}$ по формуле

$$y_2^{k-1}{}_{i,2^{k-1}j} = (x_2{}^{k-1}{}_{i,2^{k-1}j} - \tilde{x}_2^{(k)}{}^{k-1}{}_{i,2^{k-1}j} + 128)_{\mathrm{mod}256},$$

а в пп. 2, 4 и 6 алгоритма декодирования перейти от сложения к вычислению суммы по модулю 256 в соответствии с формулой

$$x_2{}^{k-1}{}_{i,2^{k-1}j} = (y_2^{(k-1)}{}^{k-1}{}_{i,2^{k-1}j} + \tilde{x}_2^{(k)}{}^{k-1}{}_{i,2^{k-1}j} - 128)_{\mathrm{mod}256}.$$

Нетрудно проверить, что получаемые *8-разрядные* двоичные коды всегда обеспечивают правильное восстановление «прореженных» компонент, при этом объем представления действительно становится *равным* объему изображения. Кроме того, переход к вычислениям по модулю 256 сужает распределения и уменьшает (правда, всего примерно на 0,1 бита) энтропию разностных компонент. Этот эффект, а также сам принцип подобных вычислений проанализированы в [8].

Компандирование диапазона яркостей. Попытка прямой реализации предложенных алгоритмов наталкивается на весьма неприятный эффект, свя-

занный с билинейной интерполяцией. Если исходное изображение содержит *не все* возможные значения яркости (что бывает достаточно часто), то интерполяция приводит к появлению большого числа *новых* значений. Результатом этого является увеличение, иногда значительное, энтропии разностных компонент и резкое снижение эффективности кодирования (объем закодированного изображения может даже оказаться больше его исходного объема).

Способ борьбы с указанным эффектом достаточно очевиден. Если изображение содержит лишь отсчеты с яркостями $p_0 < p_1 < ... < p_K$, $K < 255$, то перед началом формирования «прореженных» компонент необходимо сжать диапазон его яркостей таким образом, чтобы в нем не оставалось промежутков, т. е. выполнить табличное преобразование $p'_k = C(p_k) = k$, $k = 0, ..., K$, переводящее редко расположенные по диапазону (0,255) яркости в *целиком* заполненный диапазон $(0, K)$. После окончания декодирования изображения I_0 следует выполнить табличное преобразование, обратное описанному. Оно всегда существует, ибо прямое преобразование строго монотонно; поэтому исходное изображение I_0 всегда можно восстановить *без потерь*.

3. Служебная информация и ее представление. При оценке эффективности алгоритма кодирования следует иметь в виду необходимость хранения служебной информации. Прежде всего, для работы с кодами Хафмана нужно в ходе декодирования иметь кодовые книги, построенные при кодировании. Даже если использовать алгоритм, позволяющий строить эти книги по спискам длин кодовых слов, то общий объем затрат на их хранение будет составлять 480 байтов уже при предельной длине кодового слова 32 разряда.

Поскольку вид кодовой книги однозначно задается статистикой отсчетов, для которых эта книга строится, можно перейти от хранения таких книг к хранению гистограмм распределения значений отсчетов. Для непосредственного хранения гистограммы необходимо 1024 байта, однако в реальных гистограммах разностных компонент малые значения встречаются часто, а большие — гораздо реже. Это дает возможность сильно уменьшить затраты памяти, используя для хранения значений гистограммы $f(y_i)$ приведенный в табл. 1 неравномерный код, близкий по виду к коду Хафмана. Эксперименты показали, что у большинства подвергавшихся кодированию изображений суммарные затраты на хранение трех гистограмм указанным способом не превышали 100 байтов (хотя иногда они доходили до 800 байтов).

Кроме сведений о гистограммах, для декодирования необходимо хранить таблицу, по которой выполнялось нелинейное преобразование $C(p_k)$. Эта таблица содержит до 256 байтов. Однако она обычно слабо заполнена, и для ее восстановления необходимо знать лишь номера $K + 1$ содержательного значения аргумента p_k. Поскольку расстояние $\Delta p_k = p_k - p_{k-1} > 0$ между со-

Таблица 1

Код, используемый для хранения гистограмм

Диапазон значений гистограммы $f(y)$	Кодовые слова	Длина слова, бит
0...7 (000...111)	0000...1110	4
8...63 (001000...111111)	00110000...11111110	8
64...511 (001000000...111111111)	001100010000... 111111111110	12
512...4095 (001000000000... 1111111111111)	0011000100010000... 1111111111111110	16
.....	.....	....

седними содержательными значениями в среднем обратно пропорционально K, целесообразно хранить значения не p_k, а Δp_k, $k = 0, 1, ..., K$, и использовать для этого неравномерный код с длиной слов, пропорциональной Δp_k. Объем описания таблицы в этом случае почти не будет зависеть от вида преобразования. Нами применялся неравномерный код, приведенный в табл. 2. Нетрудно убедиться, что при работе с таким кодом объем описания таблицы преобразования никогда не превышает 60 байтов; в большинстве практических случаев он составляет 33 байта.

Таблица 2

Код, используемый для описания таблицы преобразования

Δp_k	Кодовое слово	Длина слова, бит
1	0	1
2	10	2
3	110	3
4	1110	4
5	11110	5
6	111110	6
7	1111110	7
8...256	11111111 [$\Delta p_k - 8$]	16

4. Экспериментальные результаты. *Методика проведения экспериментов.* Нами созданы две программы (PRESS и UNPRESS) для IBM-совместимых персональных ЭВМ, реализующие описанные выше алгоритмы кодирования и декодирования изображений. Эти программы написаны на языке Си; в некоторых критичных местах встречаются также команды языка Ассемблер.

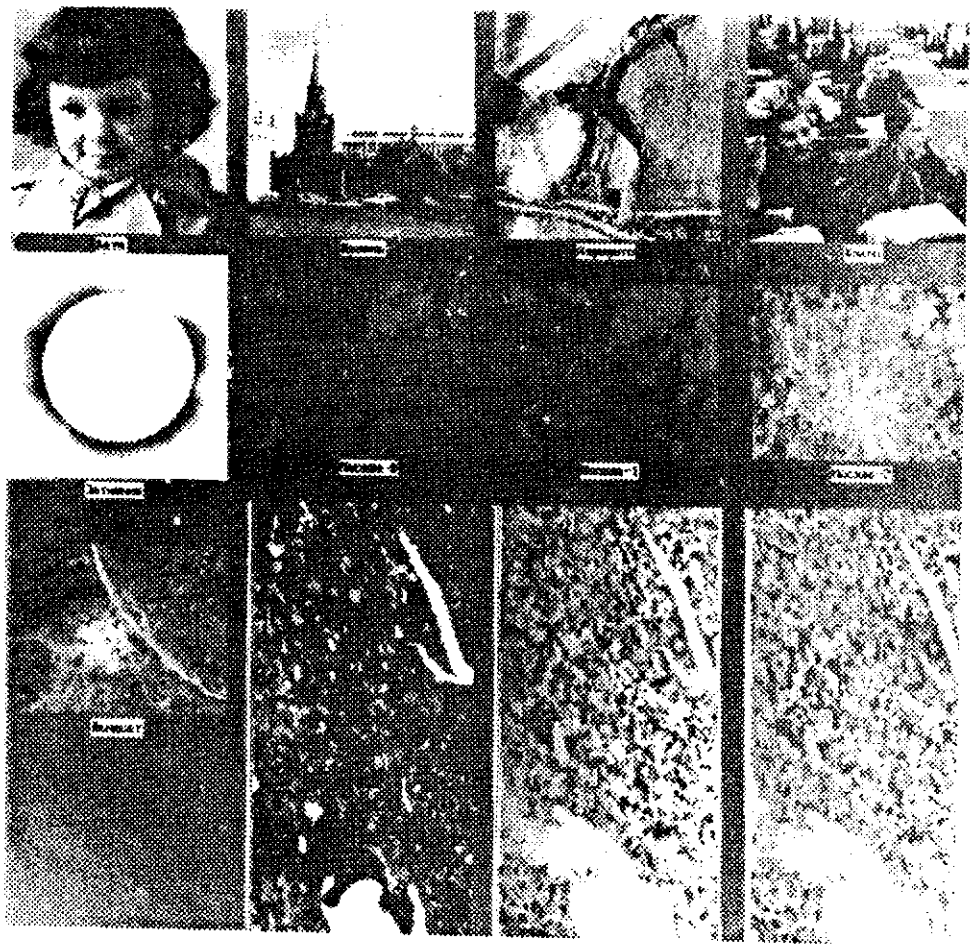

Рис. 4

Для оценки работоспособности и эффективности предложенного способа кодирования использовались двенадцать тестовых изображений, показанных на рис. 4. Энтропия четырех верхних изображений («Катя», «Кремль», «Аэрофото» и «Класс») — специально отобранных изображений высокого качества — составляла 7,2—7,6 бит (табл. 3). Восемь нижних изображений (снимок солнечного затмения «Затмение» и спектрозональные космические снимки «Ландсат», «Москва-0, -1, -2» и «Одесса-0, -1, -2») были получены в ходе реальных прикладных исследований и имели меньшую энтропию (4,0—6,1 бит).

Эксперименты проводились на IBM-совместимой персональной ЭВМ «Шнайдер» класса PC/AT с тактовой частотой 12 МГц. Изображения хранились на магнитном диске среднего быстродействия в виде стандартных файлов, содержащих прямоугольные байтовые массивы различных размеров. Результаты кодирования также записывались на этот диск. В ходе экспериментов измерялся коэффициент сжатия S — отношение длины файла, полученного при кодировании, к длине исходного файла V (при таком определении степень сжатия описания тем выше, чем *меньше* S). Определялось также время работы программ кодирования и восстановления. Для сравнения тестовые изображения кодировались не только программой PRESS, но и тремя наиболее распространенными архивирующими программами — PKZIP, LHA (ICE) и ARJ.

Обсуждение результатов. Результаты измерений коэффициента сжатия приведены в табл. 3. Из нее видно, что все «серийные» архивирующие программы обеспечивают практически одинаковую степень сжатия. Высокоэнтропийные изображения сжимаются ими не более чем до 75 % исходного объема. Сжатие низкоэнтропийных изображений оказывается более существенным: коэффициент их сжатия доходит до 43 %. Программа PRESS, реализующая кодирование с преобразованием компонент, ведет себя гораздо лучше, сжимая высокоэнтропийные изображения до 55—65 %, а низкоэнтропийные — до 34—45 % исходного объема. Выигрыш, связанный с применением этой программы вместо наиболее эффективной (и наиболее медленной) «серийной» архивирующей программы ARJ (см. табл. 3), составляет для

Таблица 3

Результаты измерений коэффициентов сжатия S

Изображение	Исходный объем V, байт	Размеры, $M \times N$	Энтропия	Коэффициент сжатия S, %				$S_{arj} - S_{press}$
				PKZIP	LHA	ARJ	PRESS	
«Катя»	262144	512 × 512	7,6	83	85	82	55	27
«Кремль»	262144	512 × 512	7,5	76	75	75	66	9
«Аэрофото»	262144	512 × 512	7,2	85	86	85	65	30
«Класс»	262144	512 × 512	7,6	85	87	85	63	22
«Затмение»	331776	576 × 576	5,3	61	62	61	45	16
«Ландсат»	759003	769 × 987	6,1	74	74	74	61	13
«Москва-0»	808960	790 × 1024	4,0	43	44	43	34	9
«Москва-1»	808960	790 × 1024	4,3	48	49	49	38	11
«Москва-2»	808960	790 × 1024	5,5	59	60	59	44	15
«Одесса-0»	2046976	1999 × 1024	6,1	60	61	60	44	16
«Одесса-1»	2046976	1999 × 1024	5,3	54	55	54	40	14
«Одесса-2»	2046976	1999 × 1024	4,2	44	45	44	35	9

первой группы изображений в среднем 22 %, для второй — в среднем 13 % исходного объема.

Времена кодирования и декодирования при фиксированном S у всех программ практически линейно возрастают с увеличением V. Программа PRESS является самой быстрой из всех кодирующих программ; ее быстродействие составляет примерно 0,8 Мбайт/мин. Программа UNPRESS, однако, работает в 1,5—2 раза медленнее, чем другие декодирующие программы; скорость ее работы — 1 Мбайт/мин.

Заключение. Проведенная оценка свойств рассмотренного метода кодирования изображений показывает, что он является гораздо более эффективным, чем доступные в настоящее время «серийные» средства их архивации. Программы PRESS и UNPRESS, реализующие этот метод, являются законченными программными модулями и могут с успехом использоваться при практической работе с изображениями любых типов.

Можно наметить следующие перспективы дальнейшего развития исследований, связанных с методом преобразования компонент:

— распространение алгоритмов кодирования и декодирования на случай сжатия описаний цветных и спектрозональных изображений; наличие сильной корреляции между их цветовыми или спектральными составляющими должно привести к повышению эффективности кодирования по сравнению с раздельным сжатием каждой из составляющих;

— применение адаптивных методов статистического кодирования с динамическим формированием кодовой книги вместо метода Хафмана, при этом степень сжатия описаний должна сильно возрасти, но, к сожалению, может увеличиться вычислительная сложность (а значит, и время реализации) процедур кодирования и восстановления.

СПИСОК ЛИТЕРАТУРЫ

1. **Джайн А. К.** Сжатие видеоинформации: Обзор // ТИИЭР.—1981.—69, вып. 3.
2. **Netravali A. N., Haskell B. G.** Digital pictures, representation and compression.—N. Y.: Plenum Press, 1988.
3. **Woods J. W., O'Neil S. D.** Subband coding of images // IEEE Trans. on Acoustics, Speech and Signal Proces.—1986.—ASSP-34, N 5.
4. **Бокштейн И. М.** Метод преобразования компонент и его предельные возможности // Кодирование и обработка изображений.—М.: Наука, 1988.
5. **Бокштейн И. М.** Адаптивные варианты метода преобразования компонент с интерполяцией по отсчетам // Автометрия.—1990.—№ 3.
6. **Burt P. J., Adelson E. H.** The laplacian pyramid as a compact image code // IEEE Trans. Communs.—1983.—COM-31, N 4.
7. **Huffman D. A.** A method for the construction of minimum redundancy codes // Proc. IRE.—1952.—40, N 9.
8. **Einarsson G.** An improved implementation of predictive coding compression // IEEE Trans. Communs.—1991.—COM-39, N 2.

Поступила в редакцию 15 февраля 1993 г.