РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ

АВТОМЕТРИЯ

2006, том 42, № 5

УДК 004.932

МЕТОД ИЕРАРХИЧЕСКОЙ КОМПРЕССИИ КАРТОГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ *

А. Ю. Баврина [1], Н. И. Глумов [2], В. В. Сергеев [2], Е. И. Тимбай [1]

[1] *Самарский государственный аэрокосмический университет, Самара*
E-mail: alina@smr.ru
[2] *Институт систем обработки изображений РАН, Самара*
E-mail: vserg@smr.ru

Описывается метод безошибочной компрессии цифровых палитровых изображений, который основан на иерархическом представлении изображения в виде набора матриц или иерархических уровней (ИУ) уменьшенного размера и запоминании только информации, необходимой для восстановления очередного ИУ по результатам расчета предыдущих ИУ. Рассмотрены различные варианты реализации метода, часть из которых активно используют особенности картографических изображений. Экспериментальные исследования показали преимущества предлагаемого метода перед широко используемыми стандартами безошибочной компрессии.

Введение. В предлагаемой работе рассматривается метод безошибочной компрессии палитровых (индексных) изображений искусственного происхождения. Важным примером таких изображений являются растровые картографические изображения (рис. 1), изображения технических чертежей, диаграмм и т. д. Такие изображения создаются, хранятся и используются с помощью специализированных программных систем в основном в векторной форме. Однако в некоторых случаях (например, для передачи по сети Интернет) эти изображения преобразуются в растровую форму, что делает актуальной задачу их компрессии.

Как правило, изображения рассматриваемого класса содержат однородные области, очерченные контуром и залитые одним цветом (либо некоторой регулярной текстурой с малым количеством цветов), отдельные знаки, линии, текст. Количество цветов на таких изображениях обычно ограничено несколькими десятками. На них практически не встречаются плавные

* Работа выполнена при поддержке Министерства образования РФ, Администрации Самарской области и Американского фонда гражданских исследований и развития (CRDF, Project SA-014-02) в рамках российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE), а также при поддержке Российского фонда фундаментальных исследований (проект № 04-01-96507).

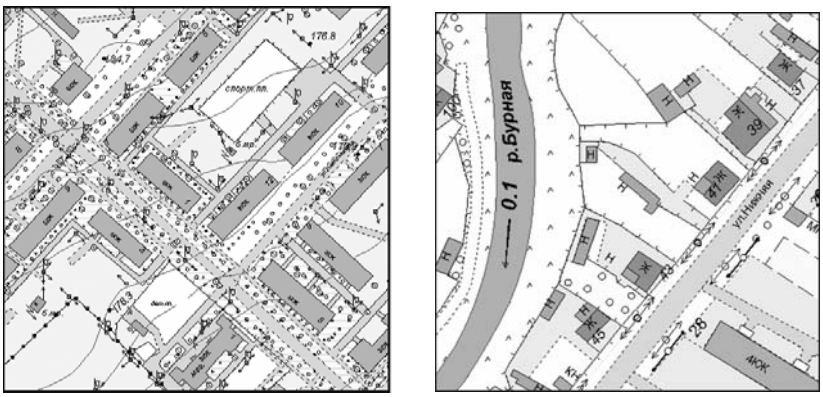

Рис. 1. Примеры картографических изображений

изменения яркости, характерные для полутоновых и цветных (RGB) изображений.

Наиболее распространенные методы компрессии изображений не являются безошибочными и основаны на том, что небольшое изменение значения пикселя приводит к несущественному визуальному изменению изображения. Для палитровых изображений необходимо применять методы безошибочной компрессии, поскольку в таких изображениях вместо значений яркости используются индексы (как правило, в формате байт на пиксель изображения), определяющие цвет в RGB-координатах с помощью отдельно сохраняемой таблицы (палитры).

В настоящее время наиболее часто используются следующие стандарты безошибочной компрессии изображений: GIF[1], Lossless JPEG [2], JBIG [3]. Однако большинство из них не учитывает специфики изображений рассматриваемого класса и, следовательно, не исчерпывает всех ресурсов повышения степени компрессии.

1. Краткое описание метода и базового алгоритма. Суть предлагаемого метода заключается в иерархическом представлении изображения в виде набора матриц или иерархических уровней (ИУ) уменьшенного размера и кодировании только информации, необходимой для восстановления очередного ИУ по восстановленным значениям предыдущих ИУ.

Пусть исходное изображение является матрицей $x(n_1, n_2)$ размером $N_1 \times N_2$, а некоторый рассматриваемый l-й уровень представляет собой матрицу $x^{(l)}(n_1, n_2)$ размером $(N_1 / M_1^l) \times (N_2 / M_2^l)$, где M_1, M_2 – коэффициенты масштабирования при переходе к очередному ИУ. На текущем ИУ оценивается распределение вероятностей (гистограмма) блоков размером $M_1 \times M_2$, покрывающих площадь изображения, осуществляется нумерация (составляется список) блоков в порядке убывания их вероятностей, после чего в очередной $(l+1)$-й уровень вместо каждого блока записывается соответствующий индекс (номер в списке) блока. Очевидно, что для восстановления всех ИУ (и, следовательно, исходного изображения) достаточно иметь списки, сформированные на всех уровнях, и матрицу старшего уровня размером $(N_1 / M_1^{L-1}) \times (N_2 / M_2^{L-1})$, где L – количество ИУ.

Компрессия изображения обеспечивается благодаря наличию на изображениях рассматриваемого класса большого количества одинаковых блоков, информация о которых в формируемых списках записывается однократно. Кроме того, для получения большей степени сжатия списки подвергаются статистическому кодированию.

Для описания алгоритма компрессии на основе предлагаемого метода примем следующие допущения. Во-первых, ограничимся рассмотрением блоков размером 2×2. Обоснование выбора таких размеров будет приведено в разд. 4. Во-вторых, компрессируемое изображение (особенно больших размеров, что характерно для изображений рассматриваемого класса) разобьем на непересекающиеся фрагменты, каждый из которых обрабатывается независимо от других. Далее в качестве изображения рассматривается один независимо обрабатываемый фрагмент, размеры которого являются степенью двойки и определяются с учетом ограничений конкретного алгоритма обработки.

Алгоритм компрессии заключается в последовательном формировании и сжатии списков четверок индексов (элементов блоков размером 2×2) для всех ИУ, начиная с нулевого уровня $x^{(0)}(n_1, n_2) = x(n_1, n_2)$. Пусть на уровне l формируется (и упорядочивается по частоте встречаемости в порядке убывания) список четверок индексов

$$C^{(l)} = \{C_k^{(l)}(0), C_k^{(l)}(1), C_k^{(l)}(2), C_k^{(l)}(3)\}_{k=0}^{K^{(l)}-1}.$$

Здесь $K^{(l)}$ – длина списка четверок. Тогда значения матрицы следующего ИУ $l+1$ формируются по правилу

$$x^{(l+1)}(n_1, n_2) = k, \quad k: \begin{cases} x^{(l)}(2n_1, 2n_2) = C_k^{(l)}(0); \\ x^{(l)}(2n_1, 2n_2 + 1) = C_k^{(l)}(1); \\ x^{(l)}(2n_1 + 1, 2n_2) = C_k^{(l)}(2); \\ x^{(l)}(2n_1 + 1, 2n_2 + 1) = C_k^{(l)}(3). \end{cases} \tag{1}$$

Матрица старшего уровня $x^{(L-1)}(n_1, n_2)$, как и списки $C^{(l)}$, $0 \le l < L-1$, подвергается статистическому кодированию и сохраняется в массиве сжатой информации.

Параметрами данного алгоритма являются размеры фрагмента и количество ИУ. Количество ИУ определяется из ограничения

$$L \le \min\{[\log_2 N_1]; [\log_2 N_2]\},$$

где $[x]$ – целая часть числа x. Размеры фрагмента выбираются таким образом, чтобы все данные (матрицы ИУ и списки) были сохранены в требуемом формате (количество бит на значение). При использовании байтового формата для хранения всех данных (что мы и будем предполагать в дальнейшем) необходимо выполнение условия

$$K^{(l)} \le 256. \tag{2}$$

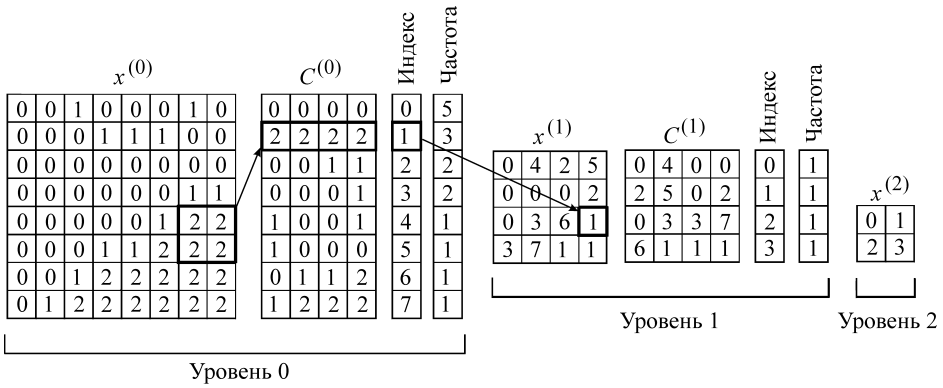

Рис. 2. Пример кодирования изображения базовым алгоритмом

На рис. 2 показан пример работы данного алгоритма для изображения размером 8×8. Матрица $x^{(0)}$ представляет собой исходное изображение; матрицы $x^{(1)}$, $x^{(2)}$ получены с помощью правила (1).

При декомпрессии матрица старшего уровня и списки ИУ декодируются, после чего последовательно восстанавливаются матрицы всех ИУ:

$$\begin{cases} x^{(l)}(2n_1, 2n_2) = C_k^{(l)}(0), \\ x^{(l)}(2n_1, 2n_2 + 1) = C_k^{(l)}(1), \\ x^{(l)}(2n_1 + 1, 2n_2) = C_k^{(l)}(2), \\ x^{(l)}(2n_1 + 1, 2n_2 + 1) = C_k^{(l)}(3), \end{cases} \quad \text{если } x^{(l+1)}(n_1, n_2) = k.$$

Однако непосредственное использование описанного алгоритма неэффективно вследствие огромного количества комбинаций значений матрицы ИУ в блоках, большая часть которых может встретиться на матрице ИУ лишь однократно. Далее предлагаются пути сокращения объемов данных в списках ИУ, что приводит к существенному повышению эффективности компрессии.

2. Повышение эффективности базового алгоритма. 2.1. *Использование общего индекса для кодирования однократно встречающихся блоков.* Пусть $T^{(l)}$ – число четверок в списке $C^{(l)}$ с частотой встречаемости (количеством) более единицы, т. е. частоты всех элементов $C_k^{(l)}$, $k \geq T^{(l)}$, равны единице.

Следующий уровень будем формировать по правилу

$$x^{(l+1)}(n_1, n_2) = \begin{cases} k, & k < T^{(l)}, \\ T, & k \geq T^{(l)}, \end{cases} \qquad k: \begin{cases} x^{(l)}(2n_1, 2n_2) = C_k^{(l)}(0); \\ x^{(l)}(2n_1, 2n_2 + 1) = C_k^{(l)}(1); \\ x^{(l)}(2n_1 + 1, 2n_2) = C_k^{(l)}(2); \\ x^{(l)}(2n_1 + 1, 2n_2 + 1) = C_k^{(l)}(3). \end{cases} \tag{3}$$

Здесь $x^{(l+1)}(n_1,\ n_2)$ – матрица уровня $(l+1)$, $n_1 = \overline{0,\ (N_1/2^{l+1})-1}$, $n_2 = \overline{0,\ (N_2/2^{l+1})-1}$, $l = \overline{0, L-2}$.

Использование этого правила приведет к сокращению диапазона значений матрицы $x^{(l+1)}(n_1, n_2)$ и уменьшению длины списка на следующих ИУ.

Введение порогового значения $T^{(l)}$ позволяет смягчить ограничение (2) на длину списка, в данном случае требуется выполнение условия

$$T^{(l)} \le 255. \tag{4}$$

На рис. 3 показан пример работы данного алгоритма для изображения размером 8×8. Матрицы $x^{(1)}$, $x^{(2)}$ получены с помощью правила (3). При сравнении рис. 3 и рис. 2 видно, что диапазон значений матрицы $x^{(1)}$ ИУ сократился с [0; 7] до [0; 4].

При декомпрессии блок уровня $x^{(l)}$ соответствует элементу списка $C_k^{(l)}$:

$$\begin{cases} x^{(l)}(2n_1, 2n_2) = C_k^{(l)}(0), \\ x^{(l)}(2n_1, 2n_2+1) = C_k^{(l)}(1), \\ x^{(l)}(2n_1+1, 2n_2) = C_k^{(l)}(2), \\ x^{(l)}(2n_1+1, 2n_2+1) = C_k^{(l)}(3), \end{cases} \qquad k = \begin{cases} x^{(l+1)}(n_1, n_2), x^{(l+1)}(n_1, n_2) < T^{(l)}; \\ x^{(l+1)}(n_1, n_2) + P,\ x^{(l+1)}(n_1, n_2) = T^{(l)}, \end{cases}$$

P – количество обработанных $x^{(l+1)}(n_1, n_2) = T^{(l)}$, $P = \overline{0, (K^{(l)} - T^{(l)} - 1)}$.

Декодирование однократно встречающихся блоков обеспечивается тем, что при кодировании они заносятся в список в некотором наперед заданном порядке обхода элементов матрицы.

2.2. *Приведение списка ИУ к случаю* $T^{(l)} = 255$. Для изображений с большим количеством деталей использование одного индекса для кодирования однократно встречающихся блоков не гарантирует выполнения неравенства (4), т. е. индекс первой четверки списка C^l с частотой, равной единице, мо-

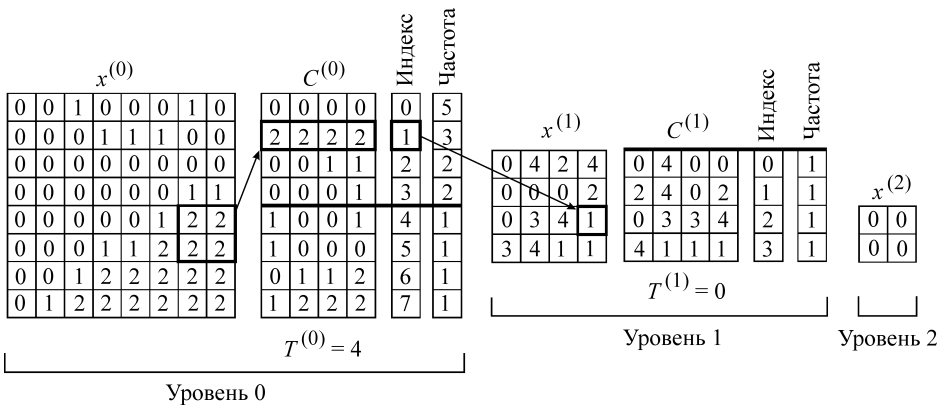

Рис. 3. Пример кодирования изображения по алгоритму с порогом

жет оказаться больше 255. В этом случае предлагается искусственно устанавливать значение порога $T^{(l)} = 255$. Таким образом, приходим к следующему способу преобразования списка иерархического уровня.

Если индекс первой четверки списка $C^{(l)}$ с частотой, равной единице, меньше 255, т. е. выполняется (4), то список не меняется. В противном случае применяется «размножение» четверок, индекс которых больше или равен 255, и приведение списка четверок к случаю $T^{(l)} = 255$. Формирование матрицы $x^{(l+1)}$ осуществляется по алгоритму, описанному в разд. 2.1.

«Размножение» четверок заключается в следующем. После построения списка четверок и его сортировки формируется новый список, первые 254 элемента которого копируются из исходного списка. Остальные элементы помещаются в список в порядке обхода элементов матрицы независимо от того, встречались ли они в списке ранее (в предыдущих реализациях метода, если четверка уже присутствовала в списке, происходило увеличение ее частоты, в текущей реализации четверка снова заносится в список). Элементы нового списка, индексы которых превышают 254, кодируются в матрице следующего уровня одним значением.

2.3. *Адаптивный выбор порога.* При уменьшении величины порога $T^{(l)}$ увеличивается объем списка $C^{(l)}$, но при этом матрица следующего уровня становится более однородной, что улучшает компрессию. Следовательно, рациональный выбор порога может обеспечить повышение эффективности компрессии.

Для оценки объема компрессированных данных текущего и следующего уровней будем использовать оценку энтропии, вычисленную для данных списка текущего уровня и значений матрицы следующего уровня. При этом выбор значения порога будем осуществлять по критерию

$$H(C^{(l)})V(C^{(l)}) + aH(x^{(l+1)})V(x^{(l+1)}) \xrightarrow[T^{(l)}]{} \min, \qquad (5)$$

где H – энтропия; $T^{(l)}$ – значение порога, начиная с которого элементы списка кодируются одним индексом; $V(C^{(l)})$ – объем данных списка текущего уровня (в битах); $V(x^{(l+1)})$ – объем данных матрицы следующего уровня (в битах); a – параметр, значение которого подбирается заранее (экспериментально установлено, что для широкого класса изображений можно принять $a = 1$).

Пусть $T_{\text{opt}}^{(l)}$ – оптимальное значение параметра $T^{(l)}$ критерия (5). После определения $T_{\text{opt}}^{(l)}$ строится список, в котором элементы с индексами, большими или равными $T_{\text{opt}}^{(l)}$, кодируются одним числом и заносятся в список в порядке обхода (при необходимости применяется «размножение» четверок, описанное в разд. 2.2).

В процессе оптимизации значения $T^{(l)} > 255$ не рассматриваются. Таким образом, алгоритм также не накладывает ограничений на длину списка (и, следовательно, размер компрессируемого фрагмента).

Предлагаемый алгоритм имеет малую трудоемкость, и кодирование с его помощью дает размер архива, близкий к размеру, полученному с использова-

нием оптимальных значений порогов на ИУ, которые могут быть найдены только путем полного перебора всех возможных значений порога (от 0 до $K^{(l)} - 1$).

3. Применение метода к цветовым плоскостям. Как упоминалось выше, небольшое количество цветов является особенностью картографических изображений. Это позволяет применить предлагаемый метод к так называемым цветовым плоскостям.

Цветовая плоскость – это бинарная матрица, содержащая единицы в точках, соответствующих точкам определенного цвета на исходном изображении, и нули во всех остальных точках.

Объем архива компрессированных данных можно сократить за счет того, что данные одной из цветовых плоскостей помещать в архив не нужно, так как точки этого цвета будут располагаться в местах, не заполненных точками остальных цветов. В качестве такой плоскости целесообразно выбирать плоскость, данные которой занимали бы в архиве наибольший объем. Дополнительное сокращение размера архива возможно за счет использования взаимосвязей между цветовыми плоскостями.

3.1. *Использование взаимосвязей между цветовыми плоскостями.* Описание предлагаемого подхода удобнее начать с этапа декомпрессии. На этом этапе изображение вначале полностью заполняется цветом первой цветовой плоскости, которая не была помещена в архив. Затем выполняется последовательное восстановление и использование цветовых плоскостей (бинарных матриц) для формирования изображения. Для каждой бинарной матрицы изображение меняется (точки окрашиваются в цвет, соответствующий плоскости) только там, где эта матрица содержит единицы. Таким образом, происходит последовательное «наслаивание» цветовых плоскостей на первую плоскость.

Такая схема декомпрессии дает возможность на этапе компрессии на каждой рассматриваемой цветовой плоскости заменять нули единицами при условии, что эти нули «покроются» единицами следующих цветовых плоскостей. Замену нулей единицами следует производить так, чтобы цветовая плоскость была наиболее «удобной» для компрессии.

Как правило, лучше сжимаются «однородные» плоскости, содержащие меньшее число деталей, поэтому можно предложить следующий принцип (алгоритм) их формирования. По рассматриваемой цветовой плоскости формируется список различных блоков и вектор чисел встречаемости этих блоков. Затем в некотором заданном порядке обхода рассматриваются блоки цветовой плоскости и выполняется замена нулей единицами, если эти нули «покрываются» единицами следующих цветовых плоскостей и такая замена уменьшает номер блока в списке. В результате будут устранены небольшие разрывы однородных областей на плоскостях и, как следствие, повышена эффективность алгоритма компрессии в целом.

3.2. *Порядок рассмотрения цветовых плоскостей.* При обработке цветовых плоскостей алгоритмом, повышающим их однородность, важным является порядок рассмотрения этих плоскостей.

Объекты на картографических изображениях можно условно подразделить на площадные и контурные (линейные) [4]. На одной цветовой плоскости могут присутствовать объекты обоих типов. Будем говорить о преимущественно площадных или преимущественно контурных цветовых плоскостях в зависимости от того, объекты какого типа занимают на рассматриваемой цветовой плоскости большую площадь.

Таблица 1

Размер блока	Средний относительный объем архива, %
1 × 2	5,73
1 × 3	5,91
1 × 4	6,17
2 × 2	5,69
2 × 3	5,94
2 × 4	6,41
3 × 3	6,50

В описываемом методе компрессии цветовая плоскость относится к преимущественно площадной или контурной в зависимости от того, каких четверок при разбиении исходного изображения на блоки размером 2×2 больше на рассматриваемой цветовой плоскости: полностью состоящих из единиц или содержащих хотя бы один нуль. Такая селекция плоскостей реализуется за один проход по изображению и не требует больших вычислительных затрат.

Эксперименты показали эффективность следующего порядка обработки: сначала обрабатываются площадные плоскости в порядке убывания количества точек соответствующего им цвета, а затем – контурные плоскости в порядке возрастания количества точек цвета. Не включается в архив площадная плоскость с наибольшим количеством единиц.

3.3. *Кодирование контурных цветовых плоскостей.* Цветовые плоскости, отнесенные к преимущественно контурным, как правило, содержат линии, небольшие знаки, текст. Обработка этих плоскостей предлагаемым методом иногда не дает высокой степени компрессии. Поэтому для описания цветовых плоскостей, содержащих длинные тонкие линии, используется цепной код.

Цепной код применяется не ко всем связным цепочкам, а лишь к тем, чья длина больше некоторого порогового значения, остальная информация на цветовой плоскости кодируется предлагаемым методом.

4. Экспериментальные исследования. Для оценки эффективности метода компрессии были проведены экспериментальные исследования на тестовых палитровых изображениях. Этот метод применялся к 14 изображениям различной сложности и с разным количеством цветов – фрагментам цифровых карт размерами до 1024×1024. В качестве алгоритма статистического кодирования было выбрано арифметическое кодирование [5].

Первая серия экспериментов была направлена на выбор размеров блоков, на которые разбиваются матрицы иерархических уровней (см. разд. 1). В табл. 1 приведены средние объемы файлов компрессированных данных относительно исходных изображений для различных размеров блоков при использовании алгоритма кодирования, описанного в разд. 2.2. Видно, что наибольшая степень компрессии достигается при размерах блоков 2×2, таким образом, допущение о размерах блоков, сделанное для упрощения описания метода, является обоснованным.

Вторая часть экспериментальных исследований была направлена на сравнение эффективности различных реализаций предлагаемого метода между собой и с некоторыми известными методами безошибочной компрессии изображений. В табл. 2 приводятся результаты сравнения предложенного метода (в шести вариантах реализации) с методами GIF, JPEG-LS [6], HGI (метод компрессии на основе иерархической сеточной интерполяции [7]) и архиватором WinZIP (версия 8.0, режим наилучшего сжатия).

Таблица 2

Методы компрессии	Средний относительный объем архива, %
WinZIP	7,11
GIF	8,03
JPEG-LS	7,93
HGI	7,19
Базовый алгоритм (разд. 1)	11,25
Алгоритм с порогом (разд. 2.1)	6,82
Алгоритм с порогом, равным 255 (разд. 2.2)	5,84
Алгоритм с адаптивным порогом (разд. 2.3)	5,81
Алгоритм с разделением на цветовые плоскости	4,78
Алгоритм с добавлением цепного кода	4,44

При реализации алгоритмов с разделением на цветовые плоскости к каждой цветовой плоскости применялся алгоритм кодирования с адаптивным выбором порогового значения.

Как и ожидалось, непосредственная реализация базового алгоритма не обеспечивает приемлемой эффективности компрессии. Однако модификация алгоритма с фиксированием одного индекса для однократно встречающихся блоков сразу уменьшает объем сжатых данных в 1,65 раза.

Наибольшую степень компрессии из рассмотренных вариантов реализации предлагаемого метода дает алгоритм с разделением на цветовые плоскости и применением цепного кода. Он обеспечивает лучшую компрессию по сравнению как с известными стандартами GIF (в среднем в 1,81 раза) и JPEG-LS (в среднем в 1,79 раза), так и с архиватором WinZIP (в среднем в 1,6 раза).

Заключение. В работе рассмотрен метод безошибочной компрессии палитровых изображений. Использование особенностей палитровых изображений в целом и картографических изображений в частности обеспечивает преимущество предлагаемого метода в степени компрессии перед известными стандартами безошибочной компрессии.

Предлагаемый метод не является симметричным (по отношению ко времени компрессии/декомпрессии) – декомпрессия происходит быстрее, в особенности для варианта с адаптивным выбором порога, что делает метод привлекательным для приложений, связанных с формированием баз данных растровых цифровых карт и быстрым извлечением из них палитровых изображений, например для передачи через Интернет.

СПИСОК ЛИТЕРАТУРЫ

1. **Мюррей Д., Ван Райпер У.** Энциклопедия форматов графических файлов: Пер. с англ. Киев: Издательская группа BHV, 1997.

2. **Weinberger M., Seroussi G., Sapiro G., Marcellin M. W.** The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS // Hewlett-Packard Computer Systems Laboratory, HPL-98-193. 1998.

3. **JBIG,** Progressive Bi-level Image Compression // Intern. Standard ISO/IEC 11544, ITU-T Recommendation T. 82. 1993.

4. **Основы** геоинформатики: Учебное пособие для студ. вузов /Под ред. В. С. Тикунова. М.: Издательский центр «Академия», 2004.

5. **Langdon G. G.** An introduction to arithmetic coding // IBM Journ. Research and Development. 1984. **28**, N 2. P. 135.

6. **HP Labs** LOCO-I/JPEG-LS Home Page // http://www.hpl.hp.com/loco

7. **Gashnikov M. V., Glumov N. I., Sergeyev V. V.** Compression method for real-time systems of remote sensing // Proc. of 15th Intern. conf. on Pattern Recognition. Barcelona, 2000. Vol. 3. P. 235.

Поступила в редакцию 21 марта 2006 г.