КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

УДК 004.932

В.Н. ЛАВРЕНТЬЕВ

АЛГОРИТМ СЖАТИЯ ИЗОБРАЖЕНИЙ

Вступление. Проблема сжатия изображений является постоянно актуальной, несмотря на растущее число предложений, поскольку динамично развивающиеся области, использующие представление изображений в электронном виде предъявляют все новые и новые требования. С одной стороны от алгоритма сжатия требуют высокого качества восстановленного изображения, а с другой – высокой скорости и степени сжатия.

С одной стороны изображения становятся все более качественными и занимают все большие объемы памяти, с другой необходимо иметь возможность быстрого просмотра множества изображений для выбора нужного, а значит иметь уменьшенную копию с низким разрешением, например, в базах данных изображений или сетевых приложениях. Есть области, имеющие специфические требования, например, в геоинформационных системах есть необходимость выборки лишь части изображения по требованию или масштабирование изображения и т. д., что достаточно подробно описано в [1].

В работе рассматривается алгоритм сжатия-восстановления для полноцветных и черно-белых статических изображений большого объема, в котором сделана попытка удовлетворить вышеуказанные иногда противоречивые требования.

Алгоритм предназначен для обработки растровых изображений, в которых каждая точка поля изображения (пиксел) имеет значение цвета, выраженное, например, в системе цветопредставления RGB амплитудами красного, зеленого и синего цветов, в системе YUV яркостной и цветоразностными ам-

V. Lavrentyev

THE COMPRESSION ALGORITHM FOR IMAGES

The compression algorithm for raster and fair-sized image files is consided. The basis of the algorithm is the color matrix decomposition to several layers and then layers compression separately.
Key words: compression algorithm, images compression.

Пропонується алгоритм стискання файлів растрових зображень великого об'єму. Основою алгоритму є розкладення матриці кольору на декілька шарів та стискання кожного шару окремо.
Ключові слова: алгоритм стискання, стискання зображень.

Предлагается алгоритм сжатия файлов растровых изображений большого объема. В основу алгоритма положено представление матрицы цвета несколькими слоями и сжатие каждого слоя в отдельности.
Ключевые слова: алгоритм сжатия, сжатие изображений.

плитудами и т. д. [2].

Алгоритм относится к типу алгоритмов сжатия с потерями. Однако, для того, чтобы потери были не заметными человеческому глазу, при создании алгоритма использованы как особенности восприятия изображения человеком, так и особенности структуры самого изображения. В первом случае учитывается то, что человеческое зрение при анализе изображения оперирует контурами, общим переходом цветов, сравнительно нечувствительно к малым цветовым изменениям и чувствительно к изменениям в яркости изображения. Во втором случае используется то, что изображение обладает избыточностью, так как соседние точки, как правило, близки по цвету и, кроме этого, цвета могут значительно коррелировать друг с другом.

Алгоритм сжатия-восстановления рассматривается на примере системы цветопредставления RGB, но возможно преобразование, например, в компоненты YCrCb форматаYUV:

$$X = 0{,}299R + 0{,}587G + 0{,}114B,$$
$$Y = 0{,}5R + 0{,}4187G + 0{,}0813B + 128,$$
$$Z = 0{,}1687R + 0{,}3313G + 0{,}5B + 128$$

или в нестандартные компоненты, например:

$$X = 1/3(R+G+B),\ Y = R - X,\ Z = G - X.$$

Исходной информацией для алгоритма являются три матрицы X, Y и Z каждая размером $I \times J$ пикселей, а значение интенсивности цвета в каждом пикселе может иметь любую разрядность. Процесс обработки и сжатия изображения осуществляется в несколько этапов.

Структура алгоритма сжатия. Определение. Слой матрицы $M \equiv [m_{i,j}]$, размерностью $I \times J$ есть матрица $N \equiv [n_{i,j}]$, той же размерности и такая, что

$$\sum_{i,j} \left| m_{i,j} - n_{i,j} \right| < \sum_{i,j} n_{i,j}.$$

Очевидно, что в соответствии с этим определением, любая матрица $M \equiv [m_{i,j}]$ может быть разложена на h слоев $M1 \equiv [m1_{i,j}]$, $M2 \equiv [m2_{i,j}]$,…, $Mh \equiv [mh_{i,j}]$, причем таких, что удовлетворяют следующим условиям.

Во-первых, матрица $M1$ является слоем матрицы M, причем таким, что $\sum_{i,j} \left| m_{i,j} - m1_{i,j} \right| < \sum_{i,j} m1_{i,j}$, матрица $M2$ является слоем разностной матрицы $R1 \equiv [r1_{i,j}]$ причем таким, что $\sum_{i,j} \left| r1_{i,j} - m2_{i,j} \right| < \sum_{i,j} m2_{i,j}$, где $r1_{i,j} = m_{i,j} - m1_{i,j}$, матрица $M3$ является слоем разностной матрицы $R2 \equiv [r2_{i,j}]$, причем таким, что

$\sum_{i,j}\left|r2_{i,j} - m3_{i,j}\right| < \sum_{i,j} m3_{i,j}$, где $r2_{i,j} = r1_{i,j} - m2_{i,j}$ и т. д. до матрицы Mh, при этом, совершенно очевидно, что любая матрица, которая является слоем разностной матрицы, является также слоем исходной матрицы.

Во-вторых, $m_{i,j} = m1_{i,j} + m2_{i,j} + \cdots + mh_{i,j}$.

Если на величину каждого элемента $rk_{i,j}$ какой-либо k-й разностной матрицы наложить ограничение $rk_{i,j} \le \delta_k$, то сумму слоев с 1-го по $(k-1)$-й можно считать приближением исходной матрицы M с погрешностью δ_k. Таким образом, задав величину погрешности δ_k, можно найти последовательность слоев исходной матрицы, дающих ее приближение с этой погрешностью.

Ясно, что суммарный объем данных в слоях $M1$ … Mk в общем случае будет в k раз больше, чем в исходной матрице M. Однако, если потребовать, чтобы каждый слой имел некоторую упорядоченную структуру, такую, что его элементы будут находиться в функциональной зависимости друг от друга, то объем данных, необходимых и достаточных для вычисления всех элементов слоя может быть значительно меньшим. Это собственно и составляет суть алгоритма сжатия. Для решения этой задачи необходимо во-первых, минимизировать количество слоев исходной матрицы, при этом позволяющих с заданной точностью восстановить все ее элементы, и во-вторых, для каждого слоя минимизировать количество данных, необходимых для вычисления всех его элементов.

Таким образом, на первом этапе для каждой исходной матрицы X, Y и Z формируется первый полный слой – три матрицы $X1$, $Y1$ и $Z1$, которые являются приближением соответственно матриц X, Y и Z. Затем вычисляются три разностные матрицы $Rx1$, $Ry1$ и $Rz1$.

На втором этапе формируется второй полный слой – матрицы $X2$, $Y2$ и $Z2$, которые являются приближением соответственно матриц $Rx1$, $Ry1$ и $Rz1$. Затем, как и на первом этапе вычисляются три разностные матрицы $Rx2$, $Ry2$ и $Rz2$.

Аналогичным образом на последующих этапах формируются слои до тех пор, пока не выполнится условие останова по заданной погрешности или по заданному числу слоев.

Так как структура алгоритма содержит практически одинаковые этапы, рассмотрим подробно один из этапов.

Формирование первого слоя. Из каждой исходной матрицы X, Y и Z выделяются строки, отстоящие друг от друга на расстоянии $d1$, начиная с нулевой, а также последняя строка I-1. Далее, из каждой исходной матрицы X, Y и Z выделяются столбцы, отстоящие друг от друга на таком же расстоянии, начиная с нулевого, а также последний столбец J-1. Таким образом, из каждой исходной матрицы X, Y и Z получены по две матрицы: из выделенных строк –

X^S, Y^S и Z^S и из выделенных столбцов – X^C, Y^C и Z^C, причем размер матриц, полученных из строк равен $D1^S \times J$, где $D1^S = \lceil I/d1 \rceil + 1$ – число выделенных строк, а размер матриц, полученных из столбцов равен $I \times D1^C$, где $D1^C = \lceil J/d1 \rceil + 1$ – число выделенных столбцов.

Анализ строк и столбцов. Рассмотрим анализ матриц X^S и X^C. Так как анализ каждой строки матрицы X^S осуществляется независимо от других строк, рассмотрим этот процесс на примере анализа строки $[X_p^S] \equiv \left(x_{p,0}^S, x_{p,1}^S, \ldots, x_{p,j}^S, \ldots, x_{p,J-1}^S\right)$, который заключается в поиске особых точек.

Если рассматривать последовательность значений элементов строки $[X_p^S]$ как точки некоторой произвольной кривой, то особые точки, как всякие точки, не являющиеся регулярными, разбивают эту последовательность на $[U_p^S]$ интервалов, каждый из которых u_p^S содержит только регулярные точки и ограничен правой $j_u^+ \in \{0, J-1\}$ и левой $j_u^- \in \{0, J-1\}$ особыми точками.

Особые точки должны удовлетворять следующим условиям. С одной стороны, точек должно быть столько, чтобы внутри любого интервала u_p^S можно было вычислить все значения элементов строки $[X_p^S]$ с заданной погрешностью δ_1, с помощью одной из функций $\varphi(*)$ заданного множества функций $\Phi(*)$:

$$\max_{j \in u_p^S} \left| x_{p,j}^S - x1_{p,j}^S \right| \le \delta_1, \qquad (1)$$

где $x1_{p,j}^S = \varphi\left(x_{p,j_u^+}^S, x_{p,j_u^-}^S, j_u^+, j_u^+ \right)$.

С другой стороны, количество особых точек, а следовательно и интервалов, должно быть минимально.

Местоположение особых точек в строке отмечается с помощью бинарного вектора $[\alpha X1_p^S] \equiv \left(\alpha x1_{p,0}^S, \alpha x1_{p,1}^S, \ldots, \alpha x1_{p,j}^S, \ldots, \alpha x1_{p,J-1}^S\right)$, где $\alpha x1_{p,j}^S = \{0, 1\}$ и в котором особые точки отмечены, например, «1» как показано на рис. 1.

Аналогичным образом анализируется каждая из $D1^S$ выделенных строк матрицы X^S, в результате чего получаем бинарную матрицу строк $\alpha X1^S$, указывающую местоположение особых точек. Значения элементов $x1_{i,j}^S$ первого слоя в особых точках или вычисляются с помощью функций $\varphi(*)$ из соответствующих элементов $x_{i,j}^S$ исходной строки или приравниваются им $x1_{i,j}^S = x_{i,j}^S$.

Аналогичным образом анализируется каждая из $D1^r$ выделенных строк матрицы X^r, в результате чего получаем бинарную матрицу строк $\alpha X1^r$, указывающую местоположение особых точек. Значения $x1_{i,j}$ элементов первого слоя $X1$ в особых точках получают из $x_{i,j}$ элементов исходной матрицы X с помощью тех же функций $x1_{i,j} = \varphi(*)$ или каким-либо другим способом, например, просто $x1_{i,j} = x_{i,j}$.

j	0	1	2	3	4	5	6	7	8	9	10	…
$[X^s_p]$	$x^s_{p,0}$	$x^s_{p,1}$	$x^s_{p,2}$	$x^s_{p,3}$	$x^s_{p,4}$	$x^s_{p,5}$	$x^s_{p,6}$	$x^s_{p,7}$	$x^s_{p,8}$	$x^s_{p,9}$	$x^s_{p,10}$	…
$[\alpha X1^s_p]$	1	0	1	0	0	0	0	1	1	0	1	…
$x1^s_{p,j}$	$x1^s_{p,0}$	-	$x1^s_{p,2}$	-	-	-	-	$x1^s_{p,7}$	$x1^s_{p,8}$	-	$x1^s_{p,10}$	…

РИС. 1. Строка $[X^s_p]$, бинарный вектор $[\alpha X1^s_p]$ и значения $x1^s_{p,j}$ в особых точках строки p

Таким образом, имея бинарную матрицу строк $\alpha X1^r$, значения $x1_{i,j}$ в особых точках, а также информацию о функциях $\varphi(*)$ (например, номер) на всех интервалах, можно получить все значения $x1_{i,j}$ в соответствующих строках первого слоя $X1$, как схематически показано на рис. 2.

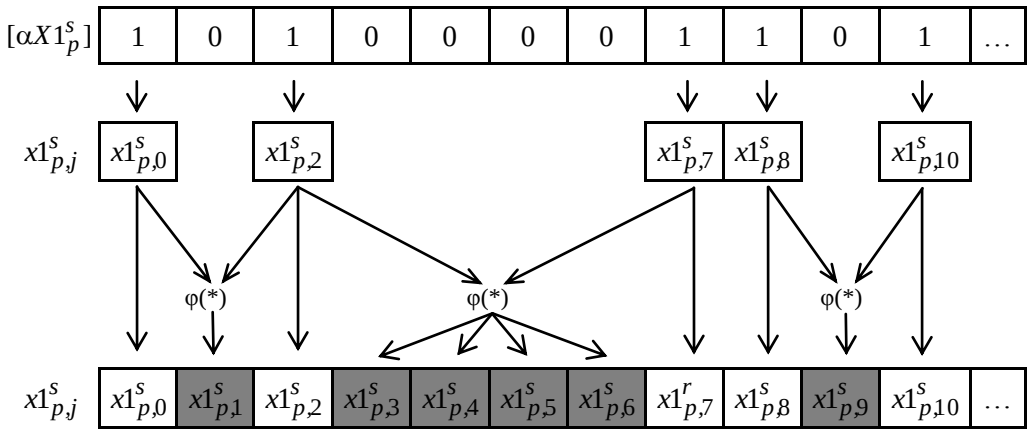

РИС. 2. Схема получения значений $x1^s_{p,j}$ выделенной строки p первого слоя $X1$

В результате анализа всех $D1^C$ выделенных столбцов матрицы X^C получаем бинарную матрицу столбцов $\alpha X1^C$, а также значения $x1^C_{p,j}$ элементов первого слоя $X1$ в особых точках столбцов. Добавив к этому информацию о функциях $\varphi(*)$ на всех интервалах, можно получить все значения в соответствующих столбцах первого слоя $X1$, аналогично тому, как показано на рис. 2 для строк. Что касается функций аппроксимации $\varphi(*)$ то, как оказалось, четырех функций (прямая линия и три кривые) вполне достаточно для решения задачи.

Здесь необходимо сделать некоторые замечания к анализу строк и столбцов.

Погрешность δ_1 не обязательно должна быть величиной постоянной, а может зависеть от характера изменения кривой $\left(x^s_{p,0}, x^s_{p,1}, ..., x^s_{p,j}, ..., x^s_{p,J\text{-}1}\right)$ на конкретном интервале и величины значений элементов кривой. Объясняется это тем, что ошибки аппроксимации на линейных «горизонтальных» участках даже в 1–2 единицы очень заметны, если значения точек кривой близки к максимальным и совсем неразличимы ошибки в несколько единиц, если значения точек кривой близки к минимальным. Незаметными могут быть и ошибки аппроксимации в несколько единиц на нелинейных участках кривой при большой разнице в значениях на краях интервалов.

Количество единичных значений в бинарных матрицах зависит как от структуры изображения, так и от заданной погрешности аппроксимации δ_1, поэтому даже незначительное изменение величины погрешности может привести к существенному сокращению числа особых точек.

В процессе анализа некоторые особые точки могут быть аннулированы. Например, если кривая содержит шумовую составляющую, что зачастую имеет место в изображениях, полученных с фотокамер, то в результате получим множество особых точек, при этом разность значений в соседних точках менее погрешности δ_1. Для этого использовался несколько модифицированный метод Гильберта – Хуанга [3], который позволяет отделить шумовую составляющую, уменьшить количество особых точек и упростить процесс аппроксимации.

Не могут быть аннулированы особые точки, которые ограничивают интервал, внутри которого значения элементов равны между собой.

В точках пересечения выделенных строк и столбцов значения элементов $x1^s_{i,j}$ и $x1^c_{i,j}$ должны быть равны. Поэтому, точки пересечения строк и столбцов заранее отмечают как особые точки.

Совмещенный анализ строк и столбцов. Суть совмещенного анализа состоит в следующем. Вначале устанавливается приоритетность исходных матриц, указывающая порядок их обработки. Положим, установлена следующая приоритетность: X, Y, Z. Это означает, что первой анализируется строка $[X^s_p]$ и определяется бинарный вектор $[\alpha X1^s_p]$. Затем анализируется строка $[Y^s_p]$, но при

этом особые точки в строке устанавливаются в соответствии с бинарным вектором $[\alpha X1_p^S]$. Каждый полученный интервал uY_p^S проверяется в соответствии с условием $\max\limits_{j \in uY_p} \left| y_{p,j}^S - y_{p,j}^S \right| \le \delta_1$ и если оно не выполняется, то в этот интервал добавляются особые точки. В результате получаем бинарный вектор $[\alpha Y1_p^S]$, в котором отмечены только добавленные точки. Затем анализируется строка $[Z_p^S]$, в которой особые точки устанавливаются в соответствии с бинарными векторами $[\alpha X1_p^S]$ и $[\alpha Y1_p^S]$. Аналогичным образом проверяются интервалы uZ_p^r, в которые добавляются или нет особые точки. В результате получаем бинарный вектор, $[\alpha Z1_p^S]$, в котором также, как и в векторе $[\alpha Y1_p^S]$, отмечены только добавленные точки.

Возможен также другой вариант совмещенного анализа, когда все три строки $[X_p^S]$, $[Y_p^S]$ и $[Z_p^S]$ анализируются одновременно. В результате получаем только один бинарный вектор $[\alpha XYZ1_p^S]$, в котором отмечены особые точки всех трех строк.

Выбор того или иного варианта анализа производится в зависимости от взаимной корреляции выделенных строк (столбцов) матриц.

Анализ исходных матриц Y и Z может быть проведен независимо друг от друга, Но, как показывает практика, значения исходных матриц X, Y и Z могут довольно сильно коррелировать между собой, особенно в цветовом пространстве YUV (формат YCrCb). Поэтому более эффективным с точки зрения степени сжатия оказался совмещенный анализ исходных матриц.

Чтобы завершить анализ всех матриц, необходимо получить значения элементов $x1_{i,j}$, $y1_{i,j}$ и $z1_{i,j}$ во всех точках первого слоя, не принадлежащих выделенным строкам и столбцам.

Значения элементов $x1_{i,j}$ матрицы $X1$, не принадлежащих выделенным строкам и столбцам вычисляются как элементы $x1_{i',j'}$, находящиеся внутри квадрата, ограниченного соседними выделенными строками a и $a+1$ и столбцами b и $b+1$ с помощью функций ψ:

$$x1_{i',j'} = \psi 1 + \psi 2, \tag{2}$$

где

$$\psi 1 = \psi\left(x1_{i',0}, x1_{i',d1}, l_a, l_{a+1}, l_b, l_{b+1}, i'\right),$$
$$\psi 2 = \psi\left(x1_{0,j'}, x1_{d1,j'}, l_a, l_{a+1}, l_b, l_{b+1}, j'\right),$$
$$i' = \mathrm{mod}(i, d1), \quad j' = \mathrm{mod}(j, d1).$$

Присутствующие в функциях $\psi 1$ и $\psi 2$ аргументы $l_a, l_{a+d1}, l_b, l_{b+d1}$, обозначают кривые аппроксимации в строках a и $a+1$ и столбцах b и $b+1$ соответственно.

Схема вычисления показана на рис. 3, где значение каждого элемента $x1_{i',j'}$ вычисляется как элемент некоторой поверхности, которая ограничена кривыми аппроксимации соседних выделенных строк и столбцов.

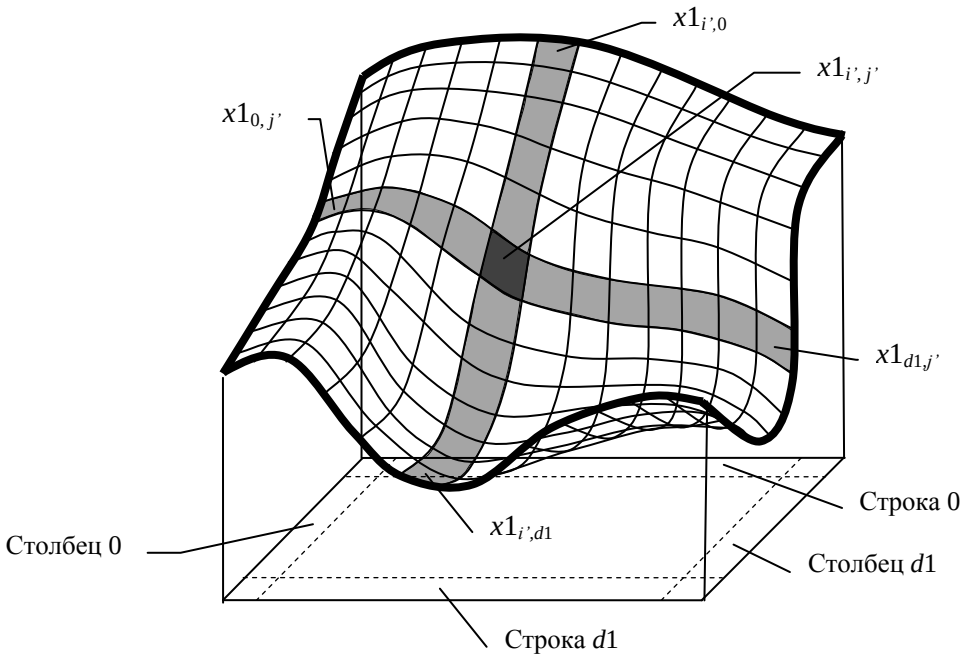

РИС. 3. Вычисление значений элементов $x1_{i',j'}$

После вычисления аналогичным образом элементов матриц $Y1$ и $Z1$ получаем первый полный слой – три матрицы $X1$, $Y1$ и $Z1$, которые являются приближением соответственно матриц X, Y и Z.

Сжатие данных слоя. Вся информация о полном первом слое содержится в бинарных матрицах строк и столбцов, множестве значений в особых точках выделенных строк и столбцов, а также множестве номеров аппроксимирующих функций на всех интервалах выделенных строк и столбцов.

Бинарные матрицы строк и столбцов представляют собой достаточно разреженные матрицы, причем в зависимости от выбранного алгоритма обработки строк, это может быть всего лишь одна матрица. Кроме этого, отмечать единичными значениями особые точки пересечения строк и столбцов нет необходимо-

сти, так как они заранее известны. Применив к этим данным один из известных алгоритмов архивации можно значительно сократить их объем.

Объем данных множества значений в особых точках и номеров функций аппроксимации в интервалах также может быть уменьшен с помощью алгоритмов архивации. Практика показала, что объем данных полного первого слоя сокращается в сотни раз по отношению к объему в исходных матрицах.

Формирование последующих слоев. Путем вычитания из каждой исходной матрицы X, Y и Z матриц первого слоя $X1$, $Y1$ и $Z1$ вычисляются три разностные матрицы $Rx1$, $Ry1$ и $Rz1$.

В результате вычитания может оказаться, что некоторые значения элементов разностной матрицы отрицательны, поэтому, чтобы избежать введения дополнительного разряда для знака, все значения такой матрицы смещаются на одну и ту же величину.

Аналогично вышеописанному алгоритму, в соответствии с которым были обработаны исходные матрицы X, Y и Z, на следующем этапе обрабатываются разностные матрицы $Rx1$, $Ry1$ и $Rz1$. Отличие состоит лишь в том, что устанавливается новое расстояние между выделяемыми строками и столбцами $d2 < d1$, а также может быть изменена величина погрешности $\delta 2 < \delta 1$ в выражении, аналогичном (1), для аппроксимации строк и столбцов.

В результате обработки разностных матриц $Rx1$, $Ry1$ и $Rz1$ будет получена информация о полном втором слое, а именно – в бинарных матрицах строк и столбцов, множестве значений в особых точках выделенных строк и столбцов, а также множестве номеров аппроксимирующих функций на всех интервалах выделенных строк и столбцов. Далее полученные данные сжимаются с помощью алгоритмов архивации. Аналогичным образом формируются компоненты по всем слоям.

Структура сжатого файла. В общем случае сжатый файл может иметь простую структуру, когда последовательность обработки изображения определяет последовательность данных в файле. Но более оптимально формировать сжатый файл исходя из его дальнейшего использования, т. е. исходя из алгоритма восстановления изображения.

Алгоритм восстановления. Так как алгоритм восстановления содержит многие операции, использованные при сжатии, поэтому не требует подробного описания.

После чтения из сжатого файла данных первого полного слоя и декомпрессии данных будет получена информация о полном первом слое – бинарные матрицы строк и столбцов, множество значений в особых точках выделенных строк и столбцов, а также множество номеров аппроксимирующих функций на всех интервалах выделенных строк и столбцов.

На основании этих данных восстанавливаются выделенные строки и столбцы матриц первого слоя $X1$, $Y1$ и $Z1$. Затем с помощью функций $\psi(*)$ по формуле (2) вычисляются значения всех элементов находящихся внутри квадратов,

ограниченных соседними выделенными строками и столбцами, в результате чего полностью восстанавливается первый слой.

На этом алгоритм восстановления может быть остановлен, если требуется быстрый просмотр сжатых изображений. Хотя изображение, полученное из матриц $X1$, $Y1$, $Z1$ и является грубым приближением оригинала, но вполне достаточным для распознавания.

Если необходимо полностью восстановить изображение, то последовательно считываются и восстанавливаются все имеющиеся слои.

Заключение. Программа сжатия изображений, реализующая описанный алгоритм показала хорошие результаты. Так при сжатии изображений объемом более 200 Мб, исходные BMP-файлы в среднем были сжаты более чем в 100 раз, причем, по мере увеличении объема исходных файлов изображений, средний коэффициент сжатия увеличивался.

Кроме того, алгоритм имеет такие достоинства:

- позволяет выделить участки изображения, которые необходимо сжать с высокой точностью или даже без искажений;

- обеспечивает возможность быстрого просмотра всего изображения с низким разрешением, используя только начало файла, т. е. только первый слой, что актуально для различного рода сетевых приложений;

- обеспечивает возможность распараллеливания процесса обработки как при сжатии, так и при восстановлении изображения.

1. http://www.compression.ru.
2. *Keith Jack.* Video demystified. A handbook for the digital endineer. Fifth eddition. Elsevier. Amsterdam. – 2007. – 920 p.
3. *Huang N.E., Shen Z., Long S.R., et al.* The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc. R. Soc. Lond. A. – 1998. – **454**. – С. 903 – 995.

Получено 17.10.2011