

ИНФОРМАТИКА

УДК 004.627

**АЛГОРИТМ СЖАТИЯ ИЗОБРАЖЕНИЙ
НА БАЗЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ**

Р.Х. САДЫХОВ, А.Н. КОЗЛОВСКИЙ

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь**Поступила в редакцию 12 сентября 2006*

Разработан новый алгоритм сжатия изображений на базе вейвлет-преобразования, основанный на оригинальном способе хранения информации о размещении вейвлет-коэффициентов. Тестирование разработанного алгоритма вейвлет-сжатия изображений подтвердило его эффективность.

Ключевые слова: алгоритм сжатия изображений, вейвлет-преобразование, разреженные матрицы, информация о размещении вейвлет-коэффициентов.

Введение

Проблема сжатия изображений привлекает в последние годы внимание специалистов для решения широкого спектра прикладных задач, связанных с теорией принятия решений. Развитие Интернета, использование цифровых камер, сканеров, принтеров привело к широкому использованию цифровых изображений и, как следствие, обусловило постоянный интерес к улучшению алгоритмов сжатия изображений.

На сегодняшний день известно много алгоритмов, основанных на вейвлет-преобразовании. Приведем наиболее популярные: рекурсивный (волновой) алгоритм, фрактальный метод сжатия изображений, алгоритм JPEG-2000, простой метод вейвлет-сжатия изображений.

Рассмотрим простой метод вейвлет-сжатия изображений более детально. После вычисления вейвлет-преобразования запоминаются только x наибольших коэффициентов, а оставшиеся $(100-x)$ коэффициентов полагаются равными нулю, причем значение x выбирается пользователем. Декодирование производится путем применения обратного вейвлет-преобразования к прореженному массиву коэффициентов.

На первый взгляд может показаться, что вейвлет-сжатие изображений с использованием 10% вейвлет-коэффициентов обеспечивает степень сжатия 10:1. Однако все не так просто. Проблема в том, что вдобавок к значениям коэффициентов мы должны знать, какие именно 10% коэффициентов должны быть использованы для представления сжатого изображения. Если мы хотим сохранить строки и столбцы информации изображения размерностью 256×256 , нам понадобятся дополнительные 2 байта на каждый коэффициент. Кроме того, коэффициенты 8-битового изображения потребуют полные 2 байта для их представления, так как имеют диапазон от -32768 до 32767 . Значит, необходимо примерно 4 байта на хранение каждого коэффициента (если мы не применяем какие-либо специальные методы кодирования). 10%-ное вейвлет-сжатие изображения 256×256 будет иметь 6554 коэффициентов, на каждый из которых требуется 4 байта, и при этом мы получим степень сжатия приблизительно 2,5:1, а не 10:1. 5%-ное

вейвлет-сжатие изображения обеспечивает степень сжатия 5:1. Использование более 25% вейвлет-коэффициентов не дает вообще никакого сжатия [1].

Следует отметить, что простому вейвлет-сжатию изображений присущ следующий недостаток:

зависимость размера блока памяти, выделяемого под хранение информации о размещении вейвлет-коэффициентов, от размера сжимаемого изображения.

Для устранения вышеуказанного недостатка и был разработан описанный ниже алгоритм вейвлет-сжатия изображений.

Алгоритм сжатия изображений на базе вейвлет-преобразования

В основе простого метода вейвлет-сжатия изображений лежит способность дискретного вейвлет-преобразования к простому и полезному усечению, что приводит нас к работе с разреженным пространством. Рассмотрим случай усечения. После применения дискретного вейвлет-преобразования к некоторому вектору X конечной размерности усечение будет не что иное, как просто обнуление вейвлет-коэффициентов, лежащих ниже некоторого порога, при этом размерность вектора остается неизменной. Следует отметить, что усечение вектора в вейвлет-пространстве может происходить только по амплитуде, а местоположение коэффициентов роли не играет [2].

Пусть нам дано некоторое изображение A в градациях серого размерностью $N \times M$, где N — ширина, а M — высота изображения. Применяя вейвлет-преобразование к изображению A и произведя усечение пространства вейвлет-коэффициентов, мы получим в терминах линейной алгебры — разреженную матрицу $\bar{A} = WA$, где W — матрица вейвлет-преобразования. Теперь мы можем вести речь об использовании компактных форматов хранения разреженных матриц.

Воспользуемся для хранения разреженной матрицы $\bar{A} = WA$ методом строковой нумерации хранения разреженных матриц. Основное преимущество данного метода заключается в том, что для хранения матрицы \bar{A} необходим объем памяти, в два раза превышающий объем памяти, необходимый для хранения ненулевых элементов матрицы. В свою очередь, другие методы требуют объем памяти втрое, а то и более раз превосходящий объем памяти, необходимый для хранения ненулевых элементов матрицы, что делает их использование неэффективным в системах сжатия изображений.

Для представления разреженной матрицы \bar{A} методом строковой нумерации хранения разреженных матриц необходимо заполнить два одномерных массива, назовем их, к примеру, pze и isj . Первый массив хранит ненулевые элементы матрицы, второй информацию о размещении ненулевых элементов матрицы, а заполнение массивов осуществляется по определенным правилам.

Как уже отмечалось выше, одним из недостатков простого вейвлет-сжатия изображений является зависимость размера блока памяти, выделяемого под хранение информации о размещении вейвлет-коэффициентов, от размера сжимаемого изображения. Так для изображений размерностью $N \times M$, причем $(N \times M)^2 < 65535$, необходимо 2 байта памяти для хранения информации о размещении вейвлет-коэффициентов, а для изображений, у которых $(N \times M)^2 > 65535$, необходимо соответственно 3 и более байт памяти для хранения информации о размещении вейвлет-коэффициентов, что делает неэффективным применение простого метода вейвлет-сжатия изображений к изображениям большой размерности. Для устранения этого недостатка воспользуемся методом строковой нумерации, а для достижения максимального эффекта сжатия применим принцип блочного кодирования — разбиение входного потока на блоки и кодирование каждого блока отдельно. Будем применять метод строковой нумерации к блокам вейвлет-коэффициентов по 225 элементов в каждом. Так как блок вейвлет-коэффициентов из 225 элементов соответствует квадратной матрице порядка 15, это позволяет выделять 1 байт памяти под информацию о размещении ненулевых вейвлет-коэффициентов (массив isj). Из поблочного рассмотрения вейвлет-коэффициентов следует, что необходимо учитывать случай, когда все элементы блока равны нулю. Для этого следует усовершенствовать правила, по которым осуществляется формирование массивов pze и isj . Описание усовершенствованных правил представлено ниже:

первые N элементов массива nze хранят элементы главной диагонали матрицы A ;
 первые N элементов массива isj , хранят индексы массива nze , которые указывают на первый элемент, не принадлежащий главной диагонали соответствующей строки матрицы A ;
 первый элемент isj всегда равен $N+2$ или нулю, если все элементы матрицы равны нулю, в данном случае массив nze опускается;

элемент $N+1$ массива isj на единицу больше индекса последнего элемента, не принадлежащего главной диагонали и находящегося в последней строке матрицы A из массива nze . Элемент $N+1$ массива nze не используется;

элементы массива $nze \geq N+2$ содержат элементы, не принадлежащие главной диагонали матрицы A , упорядоченные по строкам, а в строке упорядочивание осуществляется по столбцам;

элементы массива $isj \geq N+2$ содержат номер столбца соответствующего элемента из массива nze .

Схема предлагаемого алгоритма вейвлет-сжатия изображения на основе усечения пространства вейвлет-коэффициентов и модифицированного метода строковой нумерации хранения разреженных матриц представлена на рис. 1.

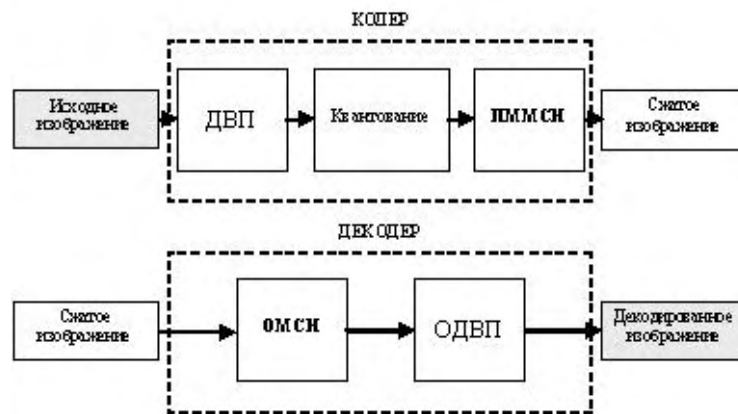


Рис. 1. Схема кодирования и декодирования изображений посредством предлагаемого алгоритма сжатия изображений на базе вейвлет-преобразования

ПММСН — это прямой модифицированный метод строковой нумерации хранения разреженных матриц, т.е. из входного блока вейвлет-коэффициентов получаем соответствующие массивы nze и isj . ОМСН — это обратный метод строковой нумерации хранения разреженных матриц, т.е. из входных массивов nze и isj получаем соответствующий блок вейвлет-коэффициентов.

Ниже представлены процедуры `pmmsn` и `omsn`, которые реализуют ПММСН и ОМСН соответственно.

```
void pmmsn ( short *a, int n, int nmax, short *nze, unsigned char *isj )
// a — это указатель на входной массив;
// n — количество элементов строки входного массива;
// nmax — максимальное число элементов входного массива;
// nze — массив содержащий ненулевые элементы;
// isj — информация о размещении ненулевых элементов;
{
  int i = 0, j = 0;
  unsigned char k;
  bool zero1 = false;
  for ( j = 0; j < n; j++ ) {
    nze[j] = a[j*n + j];
    if ( ( a[j*n + j] != 0 ) && !zero1 )
      zero1 = true;
  }
  isj[0] = (unsigned char) n+1;
}
```

```

k = n+1;
for ( i = 0; i < n; i++ ) { // цикл по строкам
  for ( j = 0; j < n; j++ ) { // цикл по столбцам
    if ( i != j ) {
      if ( (int) k > nmax ) {
        fprintf(stderr,"%s\n", " pmmsn: nmax too small");
        exit(1);
      }
      nze[(int)k] = a[i*n+j];
      isj[(int)k] = j;
      k++;
    }
  }
  isj[i+1] = k;
}
if ( !zero1 && ( k == n+1 ) ) // Установить признак нулевого блока
  isj[0] = (unsigned char) 0;
}

void omsn ( short *a, short *nze, unsigned char *isj )
// a – это указатель на восстановленный массив;
// nze – массив содержащий ненулевые элементы;
// isj – информацию о размещении ненулевых элементов;
{
  int i = 0, j = 0, n = 0;
  unsigned char k;
  n = (int) isj[0] - 1;
  for ( j = 0; j < n; j++ )
    a[j*n + j] = nze[j];
  for ( i = 0; i < n; i++ )
    for ( k = isj[i]; k <= isj[i+1] - 1; k++ )
      a[i*n + isj[k]] = nze[k];
}

```

Следует также отметить, что предложенный алгоритм вейвлет-сжатия изображений можно улучшить путем применения вторичного сжатия массива вейвлет-коэффициентов, представленного в компактной форме, посредством модифицированного метода строковой нумерации хранения разреженных матриц.

Сравнение качества восстановленных изображений

В данной статье был представлен алгоритм сжатия с потерей качества. Для оценки результатов алгоритма не только по эффективности сжатия, но и по степени достоверности восстановленного изображения используется критерий отношения пикового значения сигнала к шуму (PSNR — peak signal-to-noise ratio):

$$PSNR = -10 \log_{10} \left(\frac{MSE}{S^2} \right), \quad (1)$$

где среднеквадратическая ошибка (MSE — mean squared error) равна сумме квадратов разностей между оригиналом и восстановленным изображением, а S — это максимальное значение пикселя, причем

$$MSE = \frac{1}{MN} \sum_{m=0}^M \sum_{n=0}^N |x(m, n) - \hat{x}(m, n)|^2. \quad (2)$$

Для оценки эффективности сжатия используется понятие коэффициента сжатия. Коэффициент сжатия определяется как отношение размера входного потока к размеру выходного потока. Значения больше единицы обозначают сжатие, а значения меньше единицы — расширение. Исходное изображение "goldhill.pgm", представлено ниже:



Рис. 2. Исходное изображение "goldhill.pgm" размерностью 512×512, длина кода изображения 262 180 байт

Результаты работы простого метода вейвлет-сжатия изображений с использованием различного количества вейвлет-коэффициентов сведены в табл. 1.

Таблица 1. Данные экспериментов

Тип сжатия	Длина кода	Коэффициент сжатия	PSNR
Простое вейвлет-сжатие с использованием 5% вейвлет-коэффициентов	65551	3,99963	24,73599
Простое вейвлет-сжатие с использованием 15% вейвлет-коэффициентов	196626	1,33339	31,47907
Простое вейвлет-сжатие с использованием 25% вейвлет-коэффициентов	327696	0,80007	34,64534

Результаты работы разработанного алгоритма вейвлет-сжатия изображений с использованием различного количества вейвлет-коэффициентов сведены в табл. 2, а также представлены на рис. 3.

Таблица 2. Данные экспериментов

Тип сжатия	Длина кода	Коэффициент сжатия	PSNR
Разработанный алгоритм вейвлет-сжатия с использованием 5% вейвлет-коэффициентов	52980	4,94866	24,73181
Разработанный алгоритм вейвлет-сжатия с использованием 15% вейвлет-коэффициентов	153724	1,70552	31,47828
Разработанный алгоритм вейвлет-сжатия с использованием 25% вейвлет-коэффициентов	234324	1,11888	34,64219

Полученные результаты подтверждают, что с увеличением размерности сжимаемого изображения использование простого метода вейвлет-сжатия теряет свою эффективность, в то время как применение разработанного алгоритма остается эффективным.



Рис. 3. Исходное изображение "goldhill.pgm", к которому был применен разработанный алгоритм кодирования-декодирования изображений на базе вейвлет-преобразования, с использованием 5% (а) и 15% (б) наибольших вейвлет-коэффициентов

Следует отметить, что представленные выше результаты работы простого метода вейвлет-сжатия изображений, а также разработанного алгоритма вейвлет-сжатия изображений были получены с использованием материнского вейвлета — Добеши [3].

Заключение

В данной статье представлен новый алгоритм сжатия изображений, в основе которого лежит вейвлет-преобразование, модифицированный метод строковой нумерации хранения разреженных матриц и принцип блочного кодирования. Сжатие достигается за счет разбиения массива вейвлет-коэффициентов на блоки и кодирование каждого блока отдельно посредством модифицированного метода строковой нумерации хранения разреженных матриц. Эффективность данного алгоритма подтверждается результатами тестирования, представленными в статье.

WAVELET-BASED ALGORITHM OF IMAGE COMPRESSION

R.K. SADYKHOV, A.N. KOZLOVSKY

Abstract

We have worked out the new wavelet-based algorithm of image compression based on the original method of storage of information about placement of wavelet coefficients. The analysis of the results of the algorithm has confirmed its efficiency.

Литература

1. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии. М., 2003. С. 182–186.
2. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical recipes in C. The Art of Scientific Computing. Cambridge University Press, 1992. P. 71–90, 591–606.
3. Добеши И. Десять лекций по вейвлетам. М., Ижевск, 2004. 463 с.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М., 2005. 1072 с.