

# ТЕТРАКОДЫ В КОДИРОВАНИИ И РАСПОЗНАВАНИИ ОБРАЗОВ

Аноприенко А. Я., Кривошеев С. В., Приходько Т. А.

Кафедра ЭВМ ДонГТУ  
anoprien@dstu.donetsk.ua

## **Abstract**

*Anoprienko A., Kryvoshejev S., Prihod'ko T. Tetracodes in Image Coding and Pattern Recognition. Variant of expansion of classical binary codes is presented, which allows to essentially increase their information capacity and efficiency for in image coding and pattern recognition. As an example the results of experimental research of efficiency of tetracode for binary image encoding are presented. Various algorithms of binary image encoding on the basis of tetracode were developed and investigated. The main results are as follows: the compression of information is in the majority of cases more efficient, than by use of such a well known format as gif; the codes obtained contain an important generalised information on some essential attributes of the images (degree of regularity, orientation of basic elements, distribution of elements by sizes, etc.); the codes obtained can be effectively used for further analysis and processing of images.*

## **1. Введение**

Тетракоды, которые могут рассматриваться как расширение и развитие традиционных бинарных кодов [1], были разработаны как эффективное средство кодирования упорядоченной особым образом информации, в первую очередь ритмично организованной или представленной множествами циклически повторяющихся значений. Однако представляется целесообразным и оправданным использование тетракодов для кодирования также любой другой произвольно организованной информации. Для подтверждения данной гипотезы, которая по сути есть утверждение о универсальности тетракодов, было проведено исследование их эффективности в случае кодирования произвольных двумерных образов. В качестве основного критерия при этом рассматривалась получаемая степень сжатия информации.

Сжатие данных в настоящее время является не менее актуальной областью исследований, чем в те времена, когда ресурсы ЭВМ были скудны и дорогостоящи. Особую актуальность в связи с интенсивным развитием мультимедийных средств хранения, обработки и передачи информации в последнее время приобретает вопрос сжатия графических изображений. Кроме этого, двумерные изображения могут рассматриваться как один из наиболее наглядных и репрезентативных классов произвольных информационных структур. Другими словами, кодирование двумерных изображений может рассматриваться как модельная система, результаты исследования которой могут быть затем обобщены на широкий класс других структур и процессов. При этом, по крайней мере на первом этапе, достаточно ограничиться черно-белыми изображениями.

## **2. Обзор и анализ существующих методов кодирования**

Одним из наиболее ранних и хорошо известных методов сжатия является алгоритм Хаффмана (1952 г. [2]). Его суть заключается в сопоставлении входным символам, представленным цепочками бит одинаковой длины, кодов переменной длины, в

которых количество бит для кодирования символов обратно пропорционально частоте их появления во входном сообщении. В конце семидесятых годов были реализованы еще две важных идеи. Суть первой заключалась в использовании метода арифметического кодирования [3,4], имеющего схожую с кодированием Хаффмана функцию, но обладающего несколькими существенными свойствами, которые дают возможность достичь значительного превосходства в сжатии. Одно из свойств заключается в том, что исходный поток символов рассматривается как запись двоичной дроби из интервала  $[0,1)$ , где каждый входной символ является “цифрой” с весом, пропорциональным вероятности его появления. Другим нововведением был метод Зива-Лемпела [5,6], дающий эффективное сжатие и применяющий подход, совершенно отличный от хаффмановского и арифметического. Обе эти идеи со времени своей первой публикации были значительно усовершенствованы, и легли в основу эффективных алгоритмов сжатого кодирования информации.

В настоящее время используются три основных способа проведения сжатия: статистический, словарный и длин серий. Лучшие статистические методы применяют арифметическое кодирование, лучшие словарные - метод Зива-Лемпела (LZ). В статистическом сжатии каждому сообщению присваивается код, основанный на вероятности его появления в потоке. Высоко-вероятные сообщения получают короткие коды, и наоборот, сообщение с самой низкой вероятностью получает самый длинный код. В словарном методе группы последовательных символов или “фраз” заменяются кодом. Замененная фраза может быть найдена в некотором “словаре”. При кодировании длин серий, цепочка повторяющихся сообщений заменяется на это сообщение и счетчик повторения.

При проведении сравнительного анализа были рассмотрены широко распространенные растровые форматы хранения графической информации, реализующие (кроме BMP) в том или ином виде перечисленные выше способы сжатия: BMP (Windows Bitmap - двоичная карта), PCX (PC Paintbrush), GIF (CompuServe Graphics Interchange Format - формат графического обмена), TIFF (Tagged Image File Format - тегированный формат файлов изображений), JPG (стандарт сжатия JPEG - Joint Photographic Experts Group - Объединенная группа экспертов по фотографии).

Формат *Windows Bitmap* - самый простой формат файла для хранения графических изображений на IBM-совместимых компьютерах, разработанный фирмой Microsoft. В этом формате запоминается двоичное представление изображения. В формате *PC Paintbrush* для сжатия изображения используется метод группового кодирования (RLE), в котором группа повторяющихся байтов заменяется двумя байтами: байтом-повторителем и повторяющимся байтом. Формат *CompuServe Graphics Interchange Format*, при достаточно простой структуре файла и наличии небольшого числа атрибутов изображения, использует эффективный алгоритм LZ-сжатия - LZW (Lempel, Ziv, Welch). В формате *Tagged Image File Format* для сжатия можно использовать один из следующих алгоритмов сжатия: алгоритмы CCITT Group 3 и CCITT Group 4, предложенные Международным комитетом по телеграфии и телефонии, в основе которых лежат модифицированный алгоритм Хаффмена; алгоритм сжатия данных упаковкой бит; алгоритм кодирования длин серий и алгоритм LZW.

Тестовые изображения были получены на основе методики, предложенной в [7], но при этом диапазон исследуемых образов был существенно расширен и включал в себя структуры следующих видов:

- текстуры с равномерным распределением закрашенных точек (рис.1);
- решетки, образуемые горизонтальными и вертикальными прямыми (рис.2);
- векторы произвольного направления с равномерным распределением их длины;

- векторы с экспоненциальным распределением их длины (рис.3);
- прямоугольники с равномерным распределением размеров сторон и равномерным размещением в пределах растра (рис.4);
- произвольные четырехугольники с равномерным распределением размеров и равномерным размещением в пределах растра;
- произвольные четырехугольники с экспоненциальным распределением размеров сторон и экспоненциальным размещением в пределах растра (рис.5);
- произвольные четырехугольники с нормальным распределением размеров сторон и равномерным размещением в пределах растра;
- произвольные четырехугольники с нормальным распределением размеров сторон и нормальным размещением по оси Y в пределах растра (рис.6, многоугольники показаны не закрашенными);
- произвольные четырехугольники с нормальным распределением размеров сторон и нормальным размещением по осям X и Y в пределах растра;

В качестве тестовых образов генерировались черно-белые изображения с разрешением 256 точек на 256 точек, занимающие 8192 байт при хранении их в виде битовых карт (формат BMP). В качестве основной количественной характеристики кодируемых изображений использовался коэффициент заполнения, под которым в данном исследовании понималось отношение числа закрашенных точек к общему количеству точек, составляющих изображение (65536 точек).

Основные результаты исследования традиционных форматов приведены на рисунках 7-12 (по оси X - коэффициент заполнения, по оси Y - размер файла в байтах), при этом дополнительно к перечисленным выше использовались следующие сокращения:

- TIF1 - формат *Tagged Image File Format* (упаковка бит);
- TIF2 - формат *Tagged Image File Format* (CCITT Group 3);
- TIF3 - формат *Tagged Image File Format* (CCITT Group 4);
- TIF4 - формат *Tagged Image File Format* (кодирование длин серий).

### **3. Кодирование на базе тетракодов**

На базе тетракодов могут быть достаточно просто реализованы два следующих подхода к кодированию графической информации:

Первый подход аналогичен процедуре минимизации по карте Карно в бинарной логике. Но поглощению булевой переменной при минимизации логического выражения в бинарной логике в тетралогике соответствует присвоение текущему разряду кода значения множественности "11", что означает объединение двух повторяющихся подструктур изображения в единый код.

Второй подход основан на особенности тетракода задавать положение совокупности значений относительно определенной точки или оси, разбивающей на каждом шаге кодирования определенную часть пространства изображения пополам. Если ось вертикальная, то значение тетракода "10" обозначает расположение кодируемых подструктур слева от оси, "01" - справа от оси, "11" - по обе стороны от оси. При наличии горизонтальной оси значение "10" обозначает расположение кодируемых подструктур выше оси, "01" - ниже оси, "11" - расположение значений по обе стороны от оси.

На основе данных подходов возможно построение нескольких различных алгоритмов для кодирования информации. Так как алгоритмы предполагается использовать для сжатия графических изображений, то будем использовать блочное кодирование типа

BV (блоки кодируются словами переменной длины) [8], причем предполагается, что блок имеет размер  $N \times M$  (в частном случае  $N = M$ ).

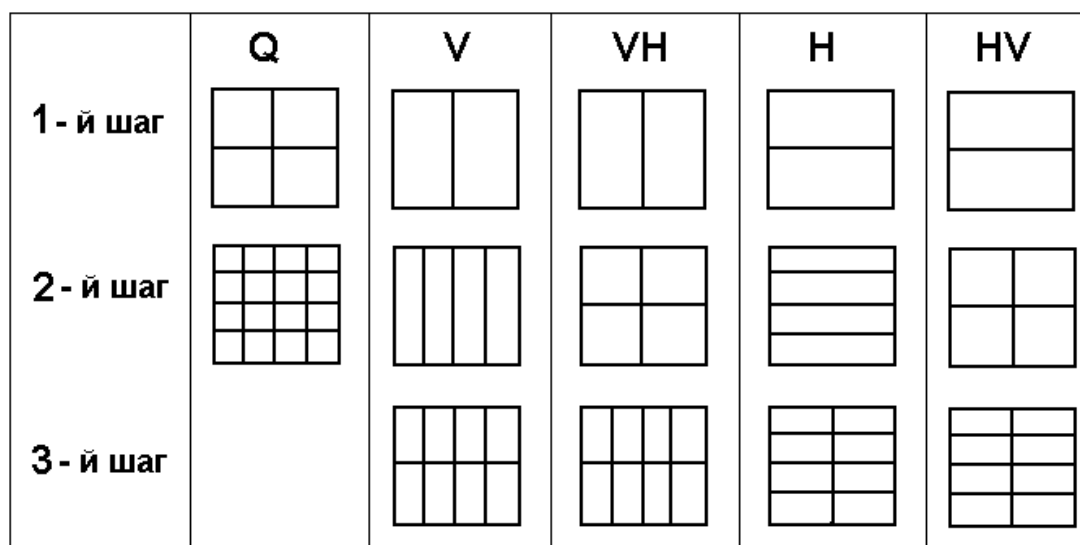


Рис. 13. Последовательность разбиения пространства изображения (на примере образа  $4 \times 4$  элемента) при реализации различных алгоритмов: Q - “квадратичного”, V - “вертикального”, VH - “вертикально-горизонтального”, H - “горизонтального”, HV - “горизонтально-вертикального”.

На базе второго подхода построен также т.н. “двоичный” алгоритм. Этот алгоритм аналогичен построению двоичного дерева. Разработано несколько вариантов этого алгоритма, которые различаются способом деления блока на 2 подблока:

**Горизонтальный (H).** Блок  $N \times M$  разбивается на 2 подблока размером  $N \times K$  ( $K = M/2$ ). Полученные блоки разбиваются на 2 подблока размером  $N \times K$  ( $K = K/2$ ) и т. д., пока полученный блок не будет иметь размер  $N \times 1$ . После этого блок  $N \times 1$  разбивается на 2 подблока размером  $K \times 1$  ( $K = N/2$ ). Разбиение происходит до тех пор пока блок не будет иметь размер  $2 \times 1$ . Из полученных блоков строится бинарное дерево, но в узлы заносятся только те блоки, который содержат хотя бы один активный бит.

**Вертикальный (V).** Блок  $N \times M$  разбивается на 2 подблока размером  $K \times M$  ( $K = N/2$ ). Полученные блоки разбиваются на 2 подблока размером  $K \times M$  ( $K = K/2$ ) и т. д., пока полученный блок не будет иметь размер  $1 \times M$ . После этого блок  $1 \times M$  разбивается на 2 подблока размером  $1 \times K$  ( $K = M/2$ ). Разбиение происходит до тех пор пока блок не будет иметь размер  $1 \times 2$ . Из полученных блоков строится бинарное дерево, но в узлы заносятся только те блоки, который содержат хотя бы один активный бит.

**Горизонтально-вертикальный (HV).** Блок  $N \times M$  разбивается на 2 подблока размером  $N \times K$  ( $K = M/2$ ). Полученные блоки разбиваются на 2 подблока размером  $L \times K$  ( $L = N/2$ ), затем на подблоки размером  $L \times K$  ( $K = K/2$ ), после этого - на подблоки размером  $L \times K$  ( $L = L/2$ ). Разбиение происходит до тех пор пока блок не будет иметь размер  $2 \times 1$ . Из полученных блоков строится бинарное дерево, но в узлы заносятся только те блоки, который содержат хотя бы один активный бит.

**Вертикально-горизонтальный (VH).** Блок  $N \times M$  разбивается на 2 подблока размером  $L \times M$  ( $L = N/2$ ). Полученные блоки разбиваются на 2 подблока размером  $L \times K$  ( $K = M/2$ ), затем на подблоки

размером  $L \times K$  ( $L=L/2$ ), после этого - на подблоки размером  $L \times K$  ( $K = K/2$ ). Разбиение происходит до тех пор пока блок не будет иметь размер  $1 \times 2$ . Из полученных блоков строится бинарное дерево, но в узлы заносятся только те блоки, который содержат хотя бы один активный бит.

Для всех приведенных выше алгоритмов, кроме “точечного” существует также их разновидность, которая называется “позиционной” и обозначается добавлением к аббревиатуре обозначения алгоритма символа P. Суть этой разновидности алгоритмов заключается в том, что блок размером в 4 бита кодируется обычным позиционным двоичным кодом и дальнейшее разбиение не происходит.

Для оценки эффективности алгоритмы кодирования на базе тетракодов были программно реализованы на языке C. В качестве тестовых были использованы изображения описанные выше. Результаты экспериментального исследования разработанных алгоритмов кодирования приведены на рис.14-19, где MIN\_ST - минимальный размер файла изображения, обеспечиваемый лучшим из известных форматов (см. рис. 7-12).

При моделировании точечный алгоритм не рассматривался из-за его низкого быстродействия при программной реализации. При исследовании других алгоритмов было установлено, что их быстродействие примерно одного порядка.

По эффективности сжатия “двоичные” алгоритмы дают код, который по размеру превосходит или соизмерим с минимальным размером кода, полученным при сжатии известными методами, которые рассмотрены в п.2. В большинстве экспериментов коды, полученные с помощью двоичных позиционных алгоритмов имеют меньший размер по сравнению с двоичными алгоритмами.

На основании результатов моделирования можно сделать вывод, что кодирование на базе тетракода является наиболее эффективным при сжатии графической информации в случае, когда изображение содержит достаточно большие блоки с однородной закраской (рис. 17-19). При этом для ряда алгоритмов получаемое сжатие существенно (в среднем в 2 раза!) превосходит результаты, получаемые при использовании метода LZW. В противном случае эффективность тетракода следует признать явно недостаточной (рис. 14-16). Однако чрезвычайно важным является следующее замечание. При кодировании произвольных решетчатых и точечных образов разработанные алгоритмы дают недостаточное сжатие лишь при работе с **конкретными** изображениями стохастического происхождения. Если же речь идет о кодировании некоторого класса подобных изображений с заданными статистическими свойствами, то некоторые варианты тетракодов, не включенных в данное исследование, позволяют кодировать такие структуры буквально несколькими кодовыми комбинациями.

#### **4. Распознавание образов**

Эффект сжатия информации при использовании тетракодов достигается в основном за счет выявления в кодируемых образах однородных и повторяющихся структур и представления их единым кодом. Причем, чем более однородным и упорядоченным является образ, тем более сжатым является его представление в виде тетракода. Например, одной-двумя кодовыми комбинациями могут быть представлены такие структуры, как горизонтальная или вертикальная линия, различные правильные (с одинаковыми интервалами между элементами) решетки и точечные сетки. Но, чем большим разнообразием по форме и/или размещению и ориентации обладают кодируемые образы, тем более сложным оказывается их представление в виде

тетракодов. Другими словами, тетракоды могут служить мерой упорядоченности или структурной сложности кодируемых образов. Но главное даже не это, а тот факт, что сжатый тетракодами образ остается достаточно информативным для его дальнейшего алгоритмического анализа. Например, сравнительно простой статистический анализ тетракодов позволяет выявлять в кодируемых структурах различные элементы упорядоченности и классифицировать их, что в совокупности с некоторыми интегральными оценками [9] позволяет эффективно организовать процесс распознавания образов.

## **Заключение**

Таким образом, экспериментальные исследования подтвердили эффективность использования тетракодов для компактного представления широкого класса образов, что позволяет с достаточной степенью уверенности утверждать об их универсальности.

Дальнейшее повышение результативности алгоритмов кодирования на базе тетракодов и расширение области их эффективного применения может быть достигнуто за счет разработки тетракодов, ориентированных на различные виды симметричных структур, а также - специфических вариантов кодирования, ориентированных на одномерные и трехмерные структуры. К последним, в частности, могут быть отнесены многоуровневые графические изображения. Представляют также интерес различные варианты ускорения кодирования вплоть до создания специальных аппаратных средств, позволяющих использовать разработанные методы кодирования в системах реального времени.

Одной из наиболее перспективных областей применения тетракодов представляются различные высокопроизводительные средства моделирования сложных систем.

## **Литература**

1. Аноприенко А.Я. Тетралогия и тетракоды. / В кн. "Сборник трудов факультета вычислительной техники и информатики". Вып.1. Донецк, ДонГТУ, 1996, с.32-43.
2. Huffman D.A. A method for the construction of minimum-redundancy codes.- Proc. Inst. Electr. Radio Eng. - 40, No. 9 (Sept. 1952), p. 1098-1101.
3. Witten I.H., Neal R.M. Arithmetic coding for data compression.- Commun. ACM.-30, No.6 (June 1987), p. 520-540.
4. Rissanen J.J. Arithmetic codings as number representations.- Acta Polytech. Scand. Math.- 31 (Dec.1979), p. 44-51.
5. Ziv J., Lempel A. An universal algorithms for sequential data compression.- IEEE Trans. Inf. Theory IT-23.-No.3 (May 1977), p. 337-343.
6. Ziv J.,Lempel A. Compression of individual sequences via variable-rate coding.- IEEE Trans.Inf.Theory IT-24.-No.5 (Sept.1978), p. 530-536.
7. Аноприенко А. Я. "Повышение производительности систем генерации изображений: структуры и алгоритмы на уровне регенерационной памяти". Автореферат диссертации на соискание учёной степени кандидата технических наук. — Киев, 1987, 20 с.
8. Кричевский Р.Е. Сжатие и поиск информации. - М.: Радио и связь,1989.
9. Аноприенко А.Я. О некоторых приложениях стохастической геометрии к анализу и синтезу вычислительных систем и алгоритмов / В кн. "Сборник трудов факультета вычислительной техники и информатики". Вып.1. Донецк, ДонГТУ, 1996, с.129-137.

**Как правильно ссылаться на эту статью:**

Аноприенко А.Я., Кривошеев С.В., Приходько Т.А. Тетракоды в кодировании и распознавании образов // Сборник научных трудов ДонГТУ. Серия «Информатика, кибернетика и вычислительная техника». Выпуск 1 (ИКВТ-97). – Донецк: ДонГТУ. – 1997. С. 99-104.