Соответственно $p_2^3 = 3/4$, $p_2^4 = 1/4$.

7 По аналогии с Z_1 код Хаффмена для $Z_2 = 3$ - **1**.

8 Опять возвращаемся к пункту 3. Координата Z_3 может равняться 4 или 5 (№11 - №13).

Соответственно $p_3^4 = 2/3$, $p_3^5 = 1/3$.

9 Код Хаффмена для $Z_3 = 4$ - **1**.

10 Возможные значения четвертой и последней координаты: 5 или 6 - $p_4^5 = 1/2$, $p_4^6 = 1/2$.

11 Код Хаффмена для $Z_4 = 6$- **0**.

Так как число закодированных координат равно k, то на этом процесс сжатия заканчивается.

В результате сжатия получилась последовательность **01.1.1.0**, которая содержит информацию об исходной комбинации. Очевидно, что произведено сжатие информации в данном случае 1,2 раза.

SUMMARY

This article is aimed the new method of compression information, called the method of local shifts. It allows to compress a data without losing information. This method doesn't need a difficult calculations and has enough high parameters.

СПИСОК ЛИТЕРАТУРЫ

1. Амелькин В. А. Методы нумерационного кодирования. - Новосибирск: Наука, 1986.
2. Кулик И.А. Оценка степени сжатия адресно-векторного кодирования // Вісник СумДУ, 1996.- №2.- С. 84 - 87.
3. Борисенко А.А., Губарев С.И. Кодирование изображений с точечной структурой. - Радиотехника: Респ. межвед. научн.- техн. сб.-Вып.69. - Харьков: Выща школа, ХГУ, 1981. -С.52-57.
4. Борисенко А.А. О разложении бернуллиевских источников информации// Вісник СумДУ, 1995.- №3.- С.57 - 60.
5. Хаффмен Д. Метод построения кодов с минимальной избыточностью: Кибернетический сб. - М.: ИЛ, 1961.-№3.- С.37-54.

Поступила в редколлегию 3 февраля 2000 г.

УДК 681.32+621.397+621.391.1

ОБРАБОТКА ЦИФРОВЫХ ТЕЛЕВИЗИОННЫХ ИЗОБРАЖЕНИЙ МЕТОДОМ ЛОКАЛЬНЫХ СРЕЗОВ

А.А.Борисенко, проф.; Т.А. Протасова, ассист.

Цифровые изображения широко используются в различных областях науки и техники для решения задач автоматизации. Однако эти изображения несут избыточную информацию, которая является во многих случаях бесполезной для принятия управляющих решений и создает дополнительные трудности для анализа изображений.

Поэтому в процессе предварительной обработки изображения частично фильтруются, а частично сжимаются. Фильтрация сопровождается безвозвратной потерей информации, а при сжатии информация может быть в случае необходимости восстановлена.

Существуют различные методы фильтрации и сжатия изображений, в частности, сглаживание путем усреднения [1], методы статистического и нумерационного кодирования [2] и др.

В данной работе предлагается особый метод фильтрации и затем сжатия телевизионных изображений, в основу которого положено разложение изображения по уровням яркости - метод локальных срезов. В результате разложения изображение представляется в виде множества более простых изображений, отличающихся друг от друга уровнями яркости [3].

Рассмотрим более подробно работу указанного метода при обработке одной телевизионной строки.

Телевизионную строку l можно представить в виде последовательности знаков $A = \{l_1, l_2, ..., l_n\}$, порождаемых алфавитом $l_i \in A$. Предполагается, что заранее можно установить, сколько в строке знаков вида $a_1, a_2, \dots, a_к$ соответственно, определить по какому-либо алгоритму количество этих букв: $r_1(a_1), r_2(a_2), \dots, r_к(a_к)$.

Очевидно, что

$$\sum_{i=1}^{k} r_i(a_i) = n,$$

где n - число букв в строке.

Например, допустим, что строка l состоит из $n = 9$ букв ($a\ b\ a\ a\ d\ c\ d\ d\ a$) из алфавита $A = \{ a, b, c, d \}$, для которых $к = 4$. Очевидно, $r(a) = 4$, $r(b) = 2$, $r(c) = 1$, $r(d) = 2$.

Сущность метода локальных срезов для строки l состоит в том, что эта строка представляется в виде матрицы (таблица 1), состоящей из единиц и нулей, строки в которой представляют собой двоичные последовательности, соответствующие буквам алфавита в порядке a, b, c, d, а столбцы - номерам по порядку расположения многозначных букв алфавита А в исходной строке $l = (\overset{1}{a}\overset{2}{b}\overset{3}{a}\overset{4}{a}\overset{5}{b}\overset{6}{c}\overset{7}{d}\overset{8}{d}\overset{9}{a})$ или в общем случае:

$$l = \overset{1}{l_1}\ \overset{2}{l_2}\ \overset{3}{l_3}\ \overset{4}{l_4}\ \overset{5}{l_5}\ \overset{6}{l_6}\ \overset{7}{l_7}\ \overset{8}{l_8}\ \overset{9}{l_9}.$$

Матрица, соответствующая строке l, имеет вид, представленный в табл.1.

Таблица 1 – Матрица локальных срезов для телевизионной строки l

	1	2	3	4	5	6	7	8	9
a	1		1	1					1
b		1			1				
c						1			
d							1	1	

В каждой строке матрицы проставляются единицы под теми номерами, где в строке l находится многозначная буква, соответствующая данной строке. Эти строки характеризуют определенный уровень яркости – ***локальный срез***. В нашем примере таких срезов будет четыре. Из многозначной строки l с очевидностью следует, что если в каком-либо локальном срезе находится единица, то ни в каком другом локальном срезе на позиции с тем же номером этой единицы быть не может. Это значит, что в одном и том же столбце может быть одна единица. Число единиц в строке, в принципе, не ограничивается, но существует связь между количеством единиц в различных строках.

Теорема 1 Число возможных расположений единиц в локальном срезе j:

$$N = C^{k_j}_{n-\sum_{i=1}^{j-1} k_i},$$

где k_j - число единиц в j- м срезе; $j = 1, 2, \ldots, n$;

k_i – число единиц в строке, соответствующей срезу l_i, $i = 1, 2, \ldots, n$, $j \neq i$,

$$\sum_{i=1}^{0} k_i = 0.$$

Доказательство теоремы следует из ограничений. По условию, если в каком-либо локальном срезе присутствует единица, то ни в каком другом срезе на позиции с этим номером единица присутствовать не может, т.е. появление единицы в последующих строках на месте присутствия единицы в предыдущих запрещено. Следовательно, количество единиц, присутствующих в предыдущих срезах данной многозначной строки при анализе среза необходимо вычитать. Очевидно, что для первого среза количество единиц в предыдущем, т.е. нулевом, срезе равно нулю и, следовательно,

$$\sum_{i=1}^{0} k_i = 0.$$

Теорема 2 Произведение возможных расположений единиц в локальных срезах равно $\prod_{j=1}^{k} N_j = P = \frac{n!}{k_1!k_2!\ldots k_k!}$.

Доказательство. Из теоремы 1 следует, что количество вариантов расположения единиц в первом локальном срезе равно $C_n^{k_1}$, во втором - $C^{k_2}_{n-k_1}$, в j- м - $C^{k_j}_{k_j} = 1$. Соответственно произведение

$$\prod_{j=1}^{k} N_j = \prod_{j=1}^{k} C^{k_j}_{n-\sum_{i=1}^{j-1} k_i} = C_n^{k_1} C^{k_2}_{n-k_1} \ldots C^{k_j}_{k_j} =$$

$$= \frac{n!}{k_1!(n-k_1)!} \frac{(n-k_1)!}{k_2!(n-k_1-k_2)!} \ldots \frac{(n-k_1-k_2-\ldots-k_{i-1})!}{k_j!(n-k_1-k_2-\ldots-k_k)!} =$$

$$= \frac{n!}{k_1!k_2!\ldots k_k!}.$$

Следствие. Минимальная длина кодовой комбинации, кодирующая матрицу срезов, равна

$$\log_2 n! - \sum_{i=1}^{k} \log_2 k_i!.$$

Это соотношение следует из

$$\log_2 P = \log_2 \left(\frac{n!}{k_1!k_2!\ldots k_k!}\right) =$$

$$= \log_2 n! - (\log_2 k_1! + \log_2 k_2! + \ldots + \log_2 k_k!) =$$

$$= \log_2 n! - \sum_{i=1}^{k} \log_2 k_i ! \cdot$$

Пример 1 Для матрицы, представленной в таблице 1, подсчитаем варианты возможных расположений единиц в полученных локальных срезах. Так как алфавит исходной многозначной строки состоит из четырех символов, то *соответствующая матрица локальных срезов имеет* четыре строки. Проанализируем возможные варианты расположения единиц для каждого из полученных срезов.

В срезе, соответствующему символу *a*, расположено четыре единицы. Длина среза равна девяти. Так как рассматриваемый срез по счету является первым, то для него выполняется соотношение $\sum_{i=1}^{0} k_i = 0$, и, следовательно, формула для первого среза имеет вид

$$N_1 = C^{k_j}_{n-\sum_{i=1}^{j-1} k_i} = C^4_{9-k_0} = C^4_{9-0} = C^4_9 = \frac{9!}{4!5!} = 126 \cdot$$

Таким образом, для первого среза, длиной девять, существует 126 возможных вариантов расположения четырех единиц.

В срезе, соответствующему символу *в*, две единицы и предыдущий срез содержат четыре единицы $\sum_{i=1}^{1} k_i = 4$, которые надо учесть, вычитая это значение из общего количества единиц. Формула будет иметь следующий вид

$$N_2 = C^{k_j}_{n-\sum_{i=1}^{j-1} k_i} = C^2_{9-\sum_{i=1}^{1} k_i} = C^2_{9-k_1} = C^2_{9-4} = C^2_5 = \frac{5!}{2!3!} = 10 \cdot$$

Получили десять возможных вариантов расположения двух единиц в оставшихся пяти ячейках.

Произведем аналогичный подсчет для следующего среза. При этом необходимо учесть единицы двух предыдущих срезов. Формула будет иметь вид

$$N_3 = C^{k_j}_{n-\sum_{i=1}^{j-1} k_i} = C^1_{9-\sum_{i=1}^{2} k_i} = C^1_{9-k_1-k_2} = C^1_{9-4-2} = C^1_3 = \frac{3!}{2!1!} = 3.$$

Это значит, что одна единица в трех оставшихся ячейках может быть расположена тремя различными способами.

Решим данную задачу для последнего, четвертого среза, соответствующего символу *d*. В последнем анализируемом срезе осталось две единицы и две свободные ячейки (клетки) матрицы. Очевидно, что две единицы в двух оставшихся ячейках матрицы могут расположиться единственным образом. Убедимся в этом. Формула для последнего среза будет иметь вид

$$N_4 = C^{k_j}_{n-\sum_{i=1}^{j-1} k_i} = C^2_{9-\sum_{i=1}^{3} k_i} = C^2_{9-k_1-k_2-k_3} = C^2_{9-4-2-1} = C^2_2 = \frac{2!}{2!0!} = 1 \cdot$$

Таким образом две единицы могут быть записаны в двух ячейках одним единственным способом.

Пример 2 Для матрицы, представленной в таблице 1, подсчитаем произведение количества возможных вариантов расположения единиц в локальных срезах. Для этого необходимо перемножить значения N_j, полученные в предыдущем примере. Формула имеет вид

$$\prod_{j=1}^{k} N_j = \prod_{j=1}^{4} N_j = N_1 N_2 N_3 N_4 = 126 \cdot 10 \cdot 3 \cdot 1 = 3780.$$

Проверим правильность полученного результата, применив формулу перестановок:

$$P = \frac{n!}{k_1! k_2! \ldots k_k!}$$

для исходной многозначной строки, имеющей длину девять символов, каждый из которых представлен одной из четырех букв первичного алфавита $A = \{a, b, c, d\}$. Таким образом, n-длина анализируемой многозначной строки, в нашем случае n соответственно равно девяти, k равно четырем, так как первичный алфавит содержит четыре буквы, причем буква a встречается в многозначной строке четыре раза, и поэтому $k_a = k_1 = 4$, буква b - два раза, c - один раз и d - два раза, и соответственно $k_b = k_2 = 2$, $k_c = k_3 = 1$, $k_d = k_4 = 2$. Подставим данные значения в формулу. Для анализируемой многозначной строки формула приобретает вид

$$P = \frac{n!}{k_1! k_2! k_3! k_4!} = \frac{9!}{4!\,2!\,1!\,2!} = 3780\,.$$

Полученное число совпадает с предыдущим результатом. Следовательно, он верен.

Пример 3 Подсчитаем минимальную требуемую емкость памяти. Прологарифмируем по основанию 2 формулу произведения сочетаний:

$$\log_2 (C_n^{k_1} C_{n-k_1}^{k_2} \ldots C_{k_j}^{k_j}) =$$

$$= \log_2 C_n^{k_1} + \log_2 C_{n-k_1}^{k_2} + \ldots + \log_2 C_{k_j}^{k_j} =$$

$$= \log_2 C_9^4 + \log_2 C_{9-4}^2 + \log_2 C_{9-4-2}^1 + \log_2 C_{9-4-2-1}^2 =$$

$$= \log_2 C_9^4 + \log_2 C_5^2 + \log_2 C_3^1 + \log_2 C_2^2 =$$

$$= \log_2 126 + \log_2 10 + \log_2 3 + \log_2 1 =$$

$$= 6.978 + 3.322 + 1.585 = 11.885\,.$$

Проверим полученное значение, оценив минимально возможную длину кодовой комбинации для многозначной строки, воспользовавшись формулой $\log_2 n! - \sum_{i=1}^{k} \log_2 k_i!$. Для нашего примера $n=9$, $k_1=4$, $k_2=2$, $k_3=1$, $k_4=2$, поэтому формула преобразуется следующим образом:

$$\log_2 9! - \sum_{i=1}^{4} \log_2 k_i! = \log_2 9! - (\log_2 4! + \log_2 2! + \log_2 1! + \log_2 2!) =$$

$$= \log_2 362880 - (\log_2 24 + \log_2 2 + \log_2 1 + \log_2 2) =$$

$$= 18.46917 - (4.585 + 1 + 0 + 1) = 18.46917 - 6.585 = 11.88417 \text{ бит.}$$

Определим значение $\log_2 P = \log_2 3780 = 11.88417$.

Для кодирования каждой многозначной буквы исходной строки необходимо два бита, так как алфавит, используемый для ее записи, содержит четыре буквы. Так как многозначная строка состоит из девяти символов, то для ее кодирования потребуется $n\log_2 k$ двоичных символов, т.е.

$$9\cdot\log_2 4=9\cdot 2=18 \text{ бит.}$$

Эффект очевиден, произошло сжатие на 18-11.88417=6.11583 бит. Если к матрице применить фильтрацию, т.е. исключить те срезы, которые содержат избыточную или несуществующую информацию, то эффект сжатия увеличивается, т.к. в формуле, представляющей собой сумму логарифмов сочетаний, некоторые слагаемые исчезают.

Разложение многозначной строки на локальные срезы позволяет применить различные алгоритмы сжатия. Например, так как количество возможных расположений единиц в каждом локальном срезе определяется числом $C_n^{k_i}$, которое является сочетанием числа единиц из значения длины локального среза, то появляется возможность сжимать срезы на основе биномиальных кодов [4].

Таким образом, применение метода локальных срезов позволяет путем фильтрации срезов и дальнейшего их сжатия получить предельный коэффициент сжатия телевизионных изображений.

SUMMARY

The original method of TV-image filtration and compression is proposed in the paper. The base of the method is a separation of a TV-image into several levels of brightness by analogy with the method of local cuts. The basic theorems about a quantity of different positions of units into the local cuts and multiplication these quantities are proved out. The corollaries of the theorems give a possibility to estimate a minimum length of code sequences for coding a matrix of the cuts and, therefore, a maximum degree of compression by the method of cuts.

СПИСОК ЛИТЕРАТУРЫ

1. Словарь по кибернетике: Св.2000 ст./ Под ред. В.С.Михалевича . - 2-е изд. - К., 1989.- 751 с.
2. Амелькин В.А. Методы нумерационного кодирования.- Новосибирск: Наука, 1986. - 158 с.
3. Борисенко А.А., Протасова Т.А. Фильтрация телевизионных изображений в физическом эксперименте // Вісник СумДУ, 1997. - № 1(7). - С.174-176.
4. Протасова Т.А. Сжатие телевизионных изображений методом слоев // Вісник СумДУ, 1996. - № 2(6). - С.76-80.

Поступила в редколлегию 22 февраля 2000 г.

УДК 681. 335.001.53

КОНЦЕПЦИЯ ПРОГРАММНО-ИНФОРМАЦИОННОГО КОНВЕЙЕРА В МАТРИЧНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ УПРАВЛЕНИЯ СЛОЖНЫМИ НАРОДНОХОЗЯЙСТВЕННЫМИ ПРОЕКТАМИ

Ю.Н.Тесля, доц.
(Черкасский инженерно-технологический институт)

Необходимость создания значительного экономического потенциала у отечественных предприятий требует поиска новых форм и методов реализации крупных государственных программ, методов, которые могут применятся в сложившихся на Украине условиях. Одним из эффективных