

ТЕХНОЛОГИЯ ФОРМИРОВАНИЯ ДВОИЧНОЙ МАСКИ УКАЗАТЕЛЕЙ ПУТЕМ ВЫДЕЛЕНИЯ ДИНАМИЧЕСКОЙ И СТАЦИОНАРНОЙ СОСТАВЛЯЮЩИХ В ДИФФЕРЕНЦИАЛЬНОМ КАДРЕ

Баранник В.В., Хаханова А.В., Кулица О.С., Кобцева В.М.

Введение

Технология кодирования двоичных масок стационарной составляющей дифференциально-представленных кадров должна базироваться на следующих позициях:

1. Обеспечение возможности адаптироваться к структурным особенностям двоичных масок кадров.
2. Формирование подхода для организации процесса кодирования, основанного на принципе гарантированного сжатия.
3. Наличие свойств относительно дополнительного сокращения времени обработки, в том числе и на основе распараллеливания процесса кодирования.
4. Дополнительное повышение эффективности обработки массивов двоичной маски на основе блочного кодирования путем адаптации к особенностям их структурного содержания. Здесь под блочным кодом понимается код мощности алфавита для длины двоичной серии, т.е. блоком является двоичная серия.

Разработка компактного представления массива двоичной маски стационарной составляющей дифференциально-представленного кадра

Для выявления закономерностей в последовательностях длин серий необходимо обосновать информативный признак, обладающий следующими свойствами:

- 1) являться информативным для длин двоичных серий с учетом адаптации к особенностям формирования массивов двоичной маски дифференциального кадра. Здесь требуется обеспечить потенциальную возможность для сокращения избыточности для произвольного содержания битовой плоскости;
- 2) не требовать значительных вычислительных затрат для оценки и выявления закономерностей, не превышающих порядка $O(n)$;
- 3) обеспечить учет наличия резких структурных отличий для двоичных указателей стационарной и динамической составляющих дифференциального представленного кадра.

Проведем разработку подхода для сокращения избыточности в массивах двоичной маски дифференциального кадра на основе выдвинутых требований.

Для учета предложенных требований предлагается использовать подход для кодового представления последовательности длин двоичных серий маски на основе выявления закономерностей в мощности алфавита Ω . Под алфавитом источника данных понимается набор значений, которые могут принимать элементы сообщений. Тогда мощностью $|\Omega|$ алфавита источника

сообщений является количество различных элементов в алфавите. Одним из простых и в тоже время эффективных кодов, обеспечивающих учет ограничений на мощность алфавита, являются коды Бодо. Код Бодо соответствует первым двум требованиям.

Простой поэлементный код Бодо предусматривает наличие сведений о:

- размере ячейки машинной памяти;

- максимальном значении r_{\max} длины серии в массивах двоичной маски

дифференциального кадра.

Если задана последовательность длин двоичных серий, т.е. $\Theta = \{r_1, \dots, r_\Phi\}$, то простой код Бодо формируется из трех этапов:

Этап 1. Отыскивается максимальное значение длины двоичной серии, для чего используется формула:

$$r_{\max} = \max_{1 \leq i \leq \Phi} (r_i). \quad (1)$$

Этап 2. Определяется количество $L(r)$ бит, которое требуется для представления максимального значения длины двоичной серии r_{\max} , что задается соотношением:

$$L(r) = \lceil \log_2 r_{\max} \rceil + 1. \quad (2)$$

Этап 3. Величина $L(r)$ записывается в начале кодового представления и является служебной информацией, указывающей на границы кодового описания соседних элементов изображения. После чего под каждую длину двоичной серии отводится на кодовое представление ровно $L(r)$ бит. Суммарное количество $L(r)_\Sigma$ бит, которое требуется для представления всех длин двоичной серии, задается выражением:

$$L(r)_\Sigma = \Phi \cdot L(r). \quad (3)$$

Простой блочный код Бодо заключается в представлении в каждом кодовом слове нескольких элементов исходного фрагмента изображения. Например, такая ситуация возникает в случае представления нескольких элементов кодируемой последовательности в одно машинное слово (один регистр внешней памяти).

Отсюда следует, что метод Бодо является одноалфавитным. В этом случае все элементы обрабатываемой последовательности принадлежат одному алфавиту. Такие последовательности называют одноалфавитными.

Однако код Бодо не соответствует третьему требованию. Это обусловлено тем, что двоичная маска дифференциально-представленного кадра в условиях съема стационарной камерой имеет значительную неоднородность структурного содержания. Под неоднородностью структурного содержания понимается то, что стационарная составляющая может занимать значительное пространство, разрезаемое незначительными по размеру элементами динамической составляющей. В этом случае длины $r(0)$, сформированные для нулевых последовательностей, будут преобладать по значению относительно длин $r(1)$, сформированных для последовательностей единичных элементов. Для такой ситуации использование кода мощности по

одному алфавиту приведет к образованию кодовой избыточности. Действительно, в соответствии с кодом мощности одного алфавита для всех длин серий, в независимости от их происхождения, формируются кодовые последовательности одинаковой длины $L(r)$. В этом случае суммарное количество разрядов $L(r)_{\Sigma}$ на представление всей последовательности длин двоичных серий будет равно:

$$L(r)_{\Sigma} = \sum_{i=1}^{\Phi} L(r)_i, \quad (4)$$

где $L(r)_i$ - количество разрядов на представление i -го элемента последовательности длин двоичных серий маски; Φ - количество длин двоичных серий сформированных для массива двоичной маски дифференциального кадра.

В тоже время, в силу неоднородности структурного содержания, реальное количество двоичных разрядов $L(r)'_{\Sigma}$ необходимое для представления всей последовательности длин двоичных серий будет значительно меньше величины $L(r)_{\Sigma}$, т.е.:

$$L(r)'_{\Sigma} \lll L(r)_{\Sigma}. \quad (5)$$

Это приводит к наличию кодовой избыточности R , а именно (рис. 1):

$$R = L(r)_{\Sigma} - L(r)'_{\Sigma}.$$

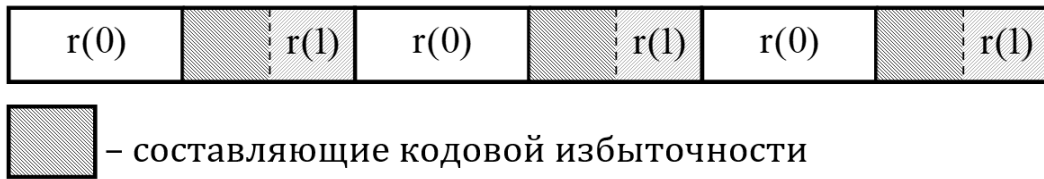


Рис. 1. Схема наличия кодовой избыточности в последовательности длин двоичных серий

Такая ситуация обусловлена тем, что для кодового представления длин серий единиц требуется значительно меньшее количество разрядов по сравнению с кодовым представлением длин серий нулей, т.е.:

$$L(r(1)) \lll L(r(0)), \quad (6)$$

где $L(r(1))$ - количество разрядов для кодового представления длин серий нулей; $L(r(0))$ - количество разрядов для кодового представления длин серий единиц.

В связи с чем, для устранения кодовой избыточности, предлагается использовать для последовательности Θ длин двоичных серий два алфавита. Первый алфавит Ω_0 определяется для длин серий нулей, соответственно второй алфавит Ω_1 определяется для длин серий единиц.

Такой подход позволяет учесть наличие резкой неоднородности в структурном содержании массива двоичной маски.

В соответствии с чем, формирование кода мощности для таких последовательностей будет реализовываться по двухалфавитной схеме. Суть схемы заключается в том, что:

1. Формируются длины серий нулей и единиц на основе массива двоичной маски.

2. Вся последовательность длин двоичных серий разбивается на две подпоследовательности.

Первая подпоследовательность образовывается на основе длин серий нулей:

$$\Theta^{(0)} = \{r(0)_1, \dots, r(0)_{\Phi_0}\}.$$

Вторая подпоследовательность образовывается на основе длин серий единиц:

$$\Theta^{(1)} = \{r(1)_1, \dots, r(1)_{\Phi_1}\}.$$

Тогда суммарное количество разрядов на представление подпоследовательностей длин серий нулей будет равно:

$$L(r(0))_{\Sigma} = \Phi_0 \log_2 r(0)_{\max}, \quad (7)$$

а суммарное количество разрядов на представление подпоследовательностей длин серий единиц будет равно:

$$L(r(1))_{\Sigma} = \Phi_1 \log_2 r(1)_{\max}. \quad (8)$$

3. Для каждой подпоследовательности формируется свой алфавит, соответственно Ω_1 и Ω_0 .

4. Код мощности строится в соответствии с построенными алфавитами.

Определение 1. Код мощности, построенный по рассмотренной выше схеме, называется двухалфавитным кодом.

Другими словами, двухалфавитным кодом мощности называется код по мощности, формируемый для двухалфавитных последовательностей.

В соответствии со свойствами двухалфавитного кода мощности его место в системе классификации методов битового кодирования приведено на рис. 2.

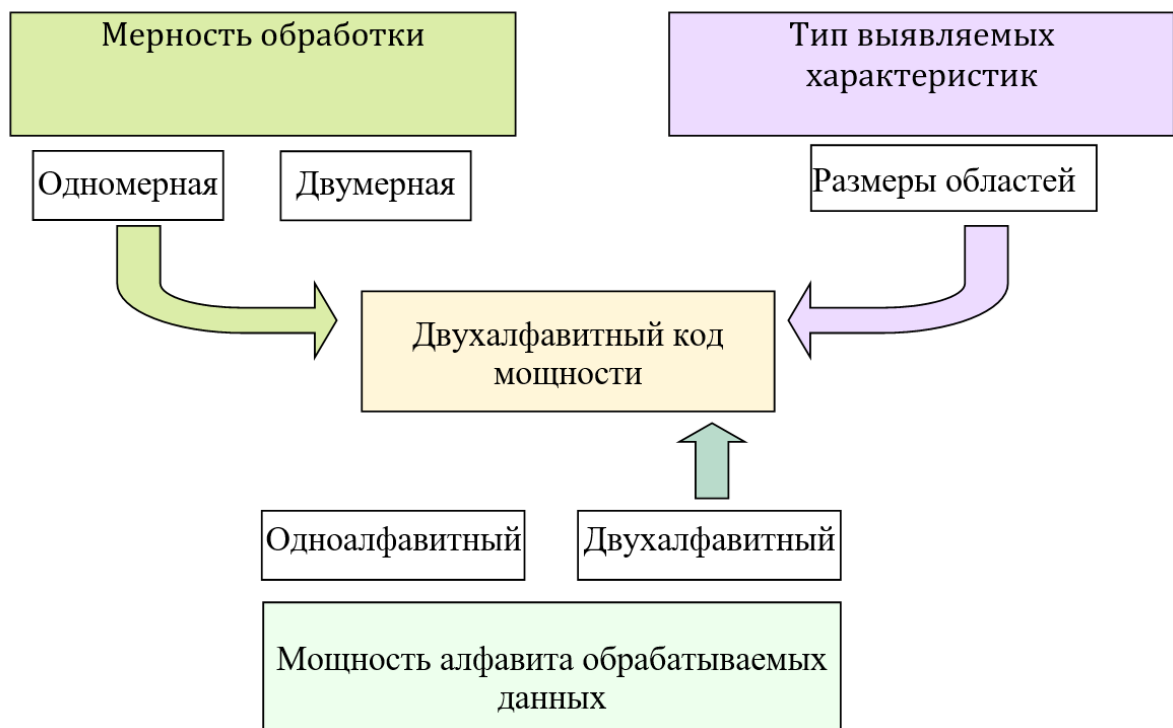


Рис. 2. Место двухалфавитного кода мощности в системе классификации методов битовой обработки

Здесь размеры двоичных областей учитываются в результате выявления длин двоичных серий.

Покажем, что для двухалфавитного кода мощности относительно длин двоичных серий двоичной маски дифференциального кадра выполняется условие, т.е. обеспечивается степень сжатия:

$$\eta_M = \frac{m_M n_M}{\Phi_0 \log_2 r(0)_{\max} + \Phi_1 \log_2 r(1)_{\max}} = \frac{\sum_{i=1}^{\Phi} r_i}{\log_2 (r(0)_{\max}^{\Phi_0} \times r(1)_{\max}^{\Phi_1})},$$

где Φ_0 - количество длин серий нулей для двоичной маски дифференциально-представленного кадра; Φ_1 - количество длин серий единиц для двоичной маски дифференциально-представленного кадра.

С учетом выявленных особенностей структурная схема метода двухалфавитного кодирования матрицы двоичной маски дифференциально-представленного кадра приведена на рис. 3.

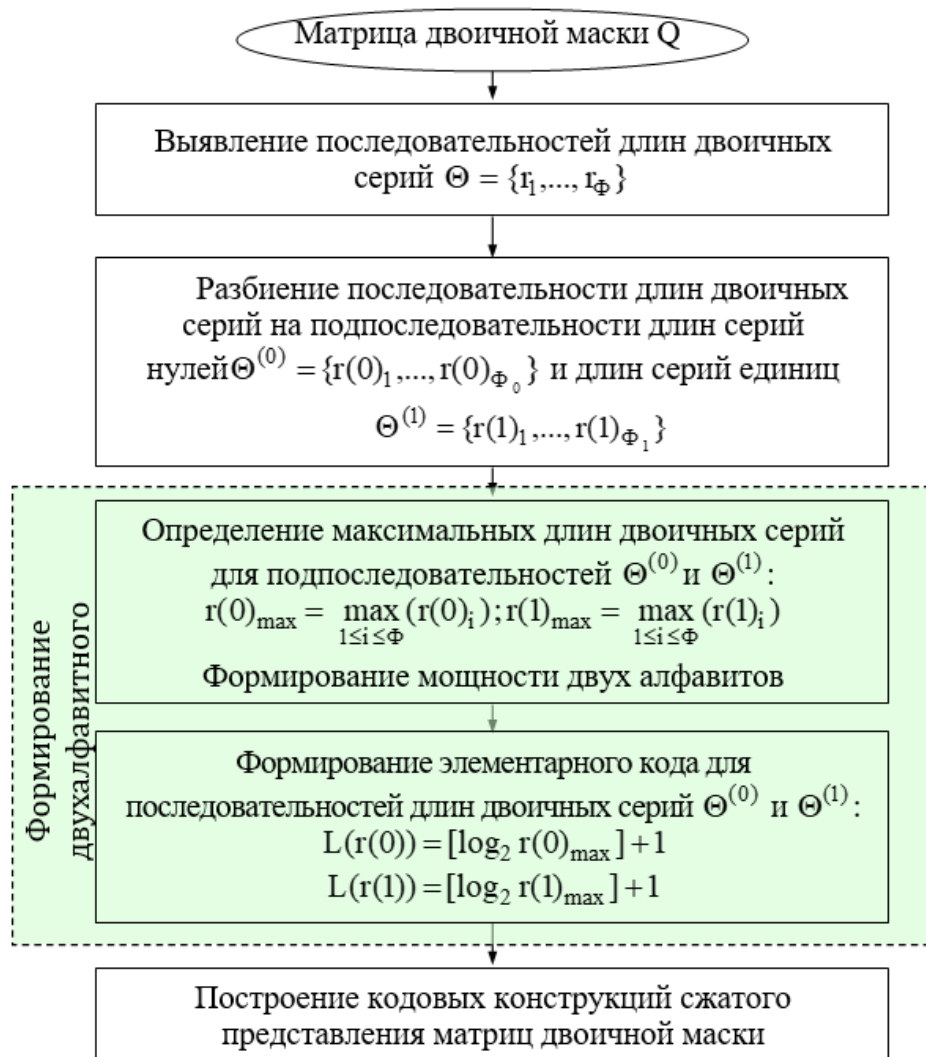


Рис. 3. Структурная схема метода двухалфавитного кодирования двоичной маски дифференциально-представленного кадра

Пример 1. Вычислим количество разрядов $L(r)_\Sigma$ на представление всей последовательности длин серий для двоичной маски дифференциально-представленного кадра Q за счет одноалфавитного кода мощности.

Двоичная маска дифференциально-представленного кадра Q

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
1	1	1	1	1	0
1	1	1	0	0	0

Вначале определим максимальное значение длины двоичной серии r_{\max} в последовательности длин двоичных серий $\Theta = \{r_1 = 19; r_2 = 1; r_3 = 4; r_4 = 5; r_5 = 1; r_6 = 3; r_7 = 3\}$ на основании выражения (1):

$$r_1 = 19; L(r)_1 = 5 \text{ бит}; r_2 = 1; L(r)_2 = 1 \text{ бит};$$

$$r_3 = 4; L(r)_3 = 2 \text{ бит}; r_4 = 5; L(r)_4 = 3 \text{ бит};$$

$$r_5 = 1; L(r)_5 = 1 \text{ бит}; r_6 = 3; L(r)_6 = 2 \text{ бит};$$

$$r_7 = 3; L(r)_7 = 2 \text{ бит}.$$

Максимальная длина серии двоичной маски дифференциально-представленного кадра $r_{\max} = 19$. Тогда на основании выражения (2) количество разрядов, затрачиваемое на представление максимальной длины серии двоичной маски равно $L(r) = 5$ битам.

Количество длин двоичных серий, сформированных для массива двоичной маски дифференциального кадра, $\Phi = 7$, то на основании выражения (3) суммарное количество разрядов на представление последовательности длин двоичных серий будет равно $L(r)_\Sigma = 5 \cdot 7 = 35$ бит.

При этом на кодовое представление исходного фрагмента изображения (фрагмент изображения классифицируется как сильнонасыщенный деталями, имеющих различные динамические составляющие) понадобится 36 разрядов. Следовательно, за счет применения одноалфавитного кода мощности для всех последовательностей длин серий, объем двоичной маски дифференциально-представленного кадра снизится на 3%.

Пример 2. Вычислим суммарное количество разрядов $L(r)_\Sigma$ на представление всей последовательности длин серий для двоичной маски дифференциально-представленного кадра Q за счет двухалфавитного кода мощности.

Двоичная маска дифференциально-представленного кадра Q

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
1	1	1	1	1	0
1	1	1	0	0	0

Вначале определим максимальное значение $r(0)_{\max}$ в подпоследовательностях длин серий нулей

$\Theta^{(0)} = \{r(0)_1=19; r(0)_2=4; r(0)_3=1; r(0)_4=3\}$ на основании выражения (1):

$$r(0)_1 = 19; L(r(0))_1 = 5 \text{ бит};$$

$$r(0)_2 = 4; L(r(0))_2 = 2 \text{ бит};$$

$$r(0)_3 = 1; L(r(0))_3 = 1 \text{ бит};$$

$$r(0)_4 = 3; L(r(0))_4 = 2 \text{ бит}.$$

Максимальная длина серии нулей двоичной маски дифференциально-представленного кадра $r(0)_{\max} = 19$. Тогда количество разрядов, затрачиваемое на представление максимальной длины серии нулей двоичной маски равно $L(r(0)) = 5$ битам.

Количество длин серий нулей, сформированных для массива двоичной маски дифференциального кадра, равно $\Phi = 4$, то на основании выражения (4) суммарное количество разрядов на представление подпоследовательности длин серий нулей будет равно $L(r(0))_{\Sigma} = 5 \cdot 4 = 20$ бит.

Затем, таким же образом, определим максимальное значение $r(1)_{\max}$ в подпоследовательностях длин серий единиц $\Theta^{(1)} = \{r(1)_1 = 1; r(1)_2 = 5; r(1)_3 = 3\}$ на основании выражения (2):

$$r(1)_1 = 1; L(r(1))_1 = 2 \text{ бит};$$

$$r(1)_2 = 5; L(r(1))_2 = 3 \text{ бит};$$

$$r(1)_3 = 3; L(r(1))_3 = 2 \text{ бит}.$$

Максимальная длина серии единиц двоичной маски дифференциально-представленного кадра $r(1)_{\max} = 5$. Тогда количество разрядов на представление максимальной длины серии единиц двоичной маски равно $L(r(1)) = 3$ битам.

Количество длин двоичных серий единиц, сформированных для массива двоичной маски дифференциального кадра, $\Phi = 3$, то на основании выражения (6) суммарное количество разрядов на представление последовательности длин серий единиц будет равно $L(r(1))_{\Sigma} = 3 \cdot 3 = 9$ бит.

Тогда суммарное количество разрядов на представление серий нулей и единиц будет равно $L(r)_{\Sigma} = L(r(0))_{\Sigma} + L(r(1))_{\Sigma} = 20 + 9 = 29$ бит.

При этом на кодовое представление исходного фрагмента изображения (фрагмент изображения классифицируется как сильнонасыщенный деталями, имеющих различные динамические составляющие) понадобится 36 разрядов. Следовательно, за счет применения двухалфавитного кода мощности для подпоследовательностей длин серий нулей и единиц, объем двоичной маски дифференциально-представленного кадра снизится на 20%. Также за счет двухалфавитного кода мощности объем двоичной маски дифференциально-представленного кадра относительно одноалфавитного кода снизится на 17%.

Разработка метода сжатия динамической составляющей дифференциально-представленного кадра

Позиционному кодированию с адаптивным выбором основания подвергаются не отдельные элементы, а их совокупности (строки). Поэтому перед началом позиционного кодирования с адаптивным выбором основания требуется из отдельных элементов сформировать массивы.

Динамическая составляющая $I_{\text{дин}}$ разбивается на массивы, как показано на рис. 4.

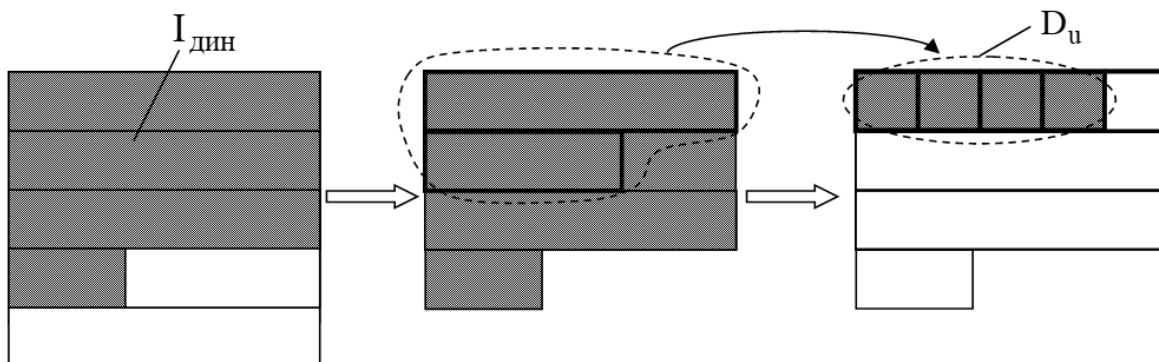
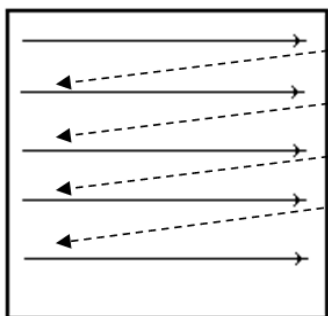


Рис. 4. Сегментация динамической составляющей на массивы

На время формирования массивов D влияет порядок обхода динамической составляющей. С технической точки зрения наименьшее время отображения информации на экране достигается при построчном порядке обхода динамической составляющей. Это объясняется тем, что в процессе отображения изображения оно выводится на экран по строкам (строчная развертка). Тогда при восстановлении изображения оно будет выводиться на экран по мере восстановления элементов, принадлежащих динамической составляющей. В этом случае динамическая составляющая разбивается на блоки равных размеров (рис. 5).



—————→ - направление обхода динамической составляющей

Рис. 5. Фрагмент разбиения динамической составляющей

Поэтому порядок обхода динамической составляющей предлагается организовывать по строкам в направлении слева - направо. Для каждой последующей динамической составляющей $I_{дин}$ процесс разбиения ее на массивы начинается сначала.

Количество элементов в динамической составляющей $v_{дин}$ нам известно, и оно определяется как количество единичных элементов в стационарной составляющей $I_{ст}$. Поэтому предлагается разбиение динамической составляющей на массивы осуществлять с учетом наличия информации о количестве элементов динамической составляющей. Отсюда размер массива предлагается выбирать с учетом:

1. Максимального заполнения массивов элементами динамической составляющей т.е., в последнем массиве сформированном для динамической составляющей должно быть как можно меньше незаполненных строк.

2. С учетом того, что позиционное число и код формируются для строки массива. В этом случае необходимо обеспечить отсутствие ситуаций, приводящих к переполнению максимально допустимой длины кодового слова.

3. С учетом того, что количество потенциальной сокращаемой избыточности увеличится с ростом размера позиционного числа. Это обусловлено тем, что для большей длины блока можно выявить больше закономерностей, используемых в процессе сокращения избыточности.

В связи с чем, предлагается следующая процедура определения размеров массивов, на которые разбивается динамическая составляющая:

1) если длина допустимого кодового слова ограничена 64 битами, то в соответствии со свойствами позиционного кодирования длина строки массива динамической составляющей не должна превышать 8. С другой стороны, в соответствии с третьим требованием относительно выбора размера массива динамической составляющей, длина строки не должна быть меньше 4. Поэтому на первом этапе предлагается выбирать длину строки массивов динамической составляющей равную 6, т.е.

$$(n_{\text{дин}} = 6), \quad \text{если } n_{\text{дин}} \leq v_{\text{дин}},$$

где $v_{\text{дин}}$ - количество элементов в динамической составляющей.

Тогда суммарное количество строк динамической составляющей $m(\Sigma)_{\text{дин}}$ при учете знания величины $v_{\text{дин}}$ определяется по формуле:

$$m(\Sigma)_{\text{дин}} = \begin{cases} \left[\frac{v_{\text{дин}}}{n_{\text{дин}}} \right] + 1, & \text{если } v_{\text{дин}} - \left[\frac{v_{\text{дин}}}{n_{\text{дин}}} \right] m_{\text{дин}} \neq 0; \\ \frac{v_{\text{дин}}}{n_{\text{дин}}}, & \text{если } v_{\text{дин}} - \left[\frac{v_{\text{дин}}}{n_{\text{дин}}} \right] m_{\text{дин}} = 0, \end{cases}$$

где $\frac{v_{\text{дин}}}{n_{\text{дин}}}$ – количество строк динамической составляющей, на основе которых

формируются полные массивы; $v_{\text{дин}} - \left[\frac{v_{\text{дин}}}{n_{\text{дин}}} \right] m_{\text{дин}}$ – остаточное количество

строк динамической составляющей, на основе которого нельзя сформировать полный массив;

2) суммарное количество строк разбивается на отдельные массивы. Здесь необходимо учитывать, что количество строк в массивах должно быть одинаковым. С другой стороны, на выбор количества строк в массиве влияет условие сохранения однородности свойств в локальной области по динамическим диапазонам. Это позволит выбирать адаптивное основание позиционного числа, которое наиболее адекватно описывает свойство локальной области. В связи с чем, предлагается количество строк в массиве

выбирать равным $n_{\text{дин}} = 6$. Тогда количество $v_M^{(\text{дин})}$ массивов, на которые разбивается вся динамическая составляющая, определяется по формуле:

$$v_M^{(\text{дин})} = \left[\frac{m(\Sigma)_{\text{дин}}}{m_{\text{дин}}} \right].$$

При этом, чтобы сократить количество оснований позиционных чисел, выбираемых для каждого массива в отдельности, предлагается следующая процедура:

$$m(\Sigma)_{\text{дин}} - \left[\frac{m(\Sigma)_{\text{дин}}}{m_{\text{дин}}} \right] m_{\text{дин}} < 3,$$

то последний полный массив заполняется оставшимися строками из неполного массива.

В противном случае, когда

$$m(\Sigma)_{\text{дин}} - \left[\frac{m(\Sigma)_{\text{дин}}}{m_{\text{дин}}} \right] m_{\text{дин}} \geq 4,$$

формируется неполный массив.

В общем виде схема позиционного кодирования массивов динамической составляющей с адаптивным выбором основания, приведена на рис. 6.

Таблица Ω составляется из всех возможных массивов данных (в общем виде), распределенных в зависимости от значения основания, где $U+1$ – количество различных массивов динамической составляющей заданного размера, а u – индекс подмножества массивов динамической составляющей, удовлетворяющих значению основания λ_u . Массивы динамической составляющей образуются из исходного изображения на основе функционалов $\varphi_{\text{дин}}$. Затем для полученных массивов находится основание λ_u . Найденное основание служит входным параметром для считывания из таблицы Ω множеств Ω_u соответственно возможных массивов динамической составляющей.

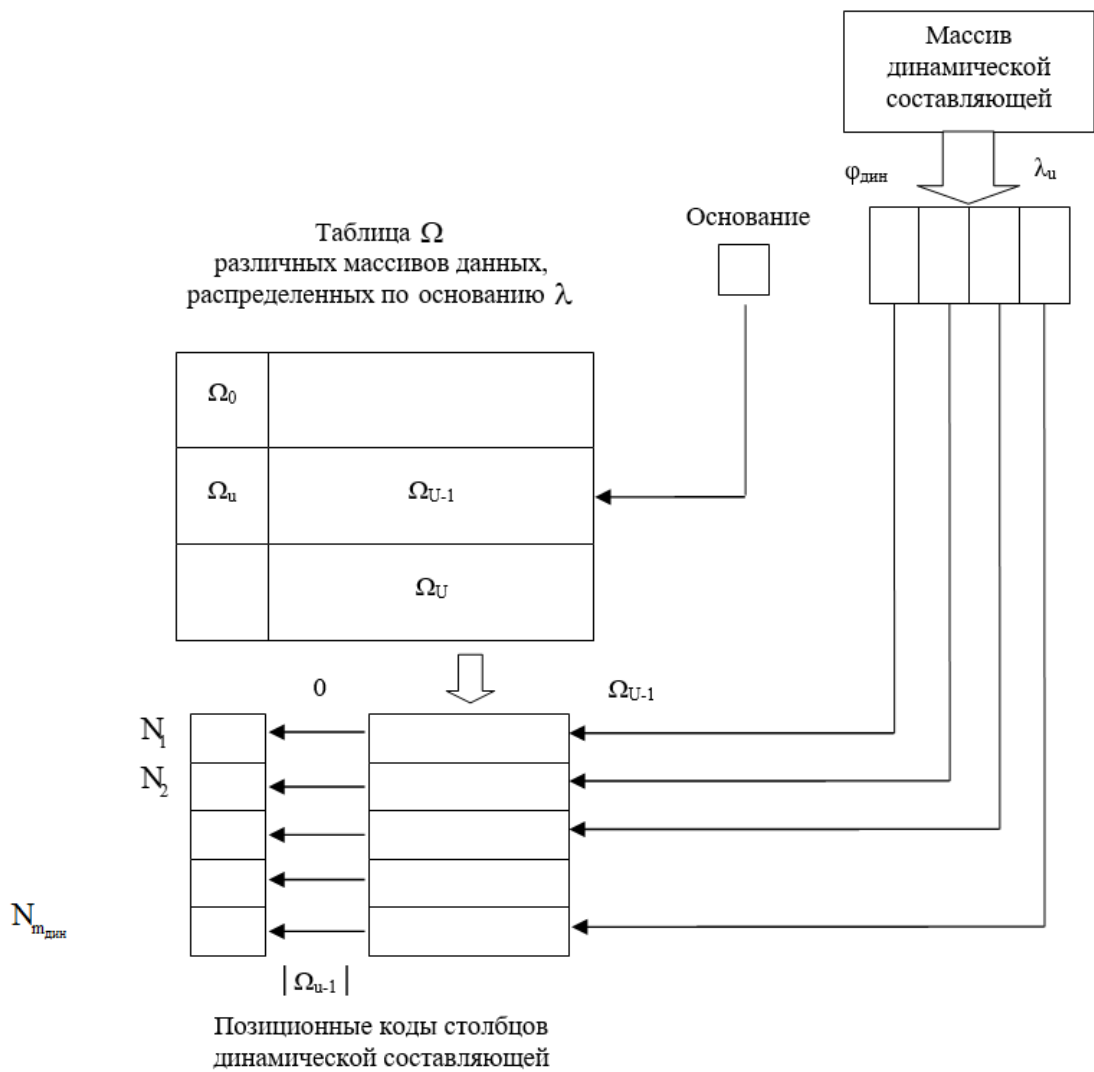


Рис. 6. Схема позиционного кодирования массива динамической составляющей с адаптивным выбором основания

В пределах выбранных множеств Ω_u столбцы массивов динамической составляющей нумеруются по порядку соответственно от 0 до $|\Omega_u|$. Поэтому за счет сопоставления конкретных массивов динамической составляющей столбцам множеств Ω_u определяются соответствующие коды для позиционных чисел $N(u)_i$ (рис. 6).

В общем случае полный массив D_u динамической составляющей имеет вид:

$$D_u = \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,j} & \dots & d_{1,n_{\text{дин}}} \\ & & \dots & & & \\ d_{i,1} & d_{i,2} & \dots & d_{i,j} & \dots & d_{i,n_{\text{дин}}} \\ & & \dots & & & \\ d_{m_{\text{дин}},1} & d_{m_{\text{дин}},2} & \dots & d_{m_{\text{дин}},j} & \dots & d_{m_{\text{дин}},n_{\text{дин}}} \end{pmatrix},$$

где $d_{i,j}$ – $(i; j)$ -й элемент массива динамической составляющей; $m_{\text{дин}}, n_{\text{дин}}$ – соответственно количество строк и столбцов в массивах D_u .

Требуется заметить, что массив D_u может быть заполненный не полностью.

Формирование строки массива динамической составляющей проводится с учетом сдвига на количество элементов, равное длине строки массива $n_{\text{дин}}$.

На процесс разбиения динамической составляющей на массивы D_u влияет максимальное значение d_{max} элемента $I_{\text{дин}}$ и количество $n_{\text{дин}}$ столбцов. Это обусловлено тем, что код формируется не для отдельного элемента динамической составляющей, а для строки массива динамической составляющей. Поэтому на значения кода позиционного числа $N(u)_i$ и на длину его кодограммы L_i оказывает влияние адаптивное основание, которое выбирается в соответствии с величиной d_{max} и количества элементов в позиционном числе, что определяется величиной $n_{\text{дин}}$. При этом мы можем управлять только величиной $n_{\text{дин}}$. Величина d_{max} определяется отдельно для каждой динамической составляющей. В связи с этим, перед выбором величины $n_{\text{дин}}$ предлагается рассмотреть особенности позиционного кодирования с адаптивным выбором основания.

Первая особенность заключается в ограничении, накладываемом на длину кодограммы L_j , которое отводится под позиционное число $N(u)_j$. Выполнение этого ограничения необходимо для исключения потерь информации из-за нехватки разрядов для представления значения кода $N(u)_j$. Для этого величина максимально возможного числа, которое может храниться в кодограмме L_j разрядов, не должна превышать величины динамического диапазона $D_{(N)}$ значений кодов позиционных чисел, т.е.:

$$2^{L_j} \geq D_{(N)},$$

где 2^{L_j} – максимально возможное число, которое может храниться в кодограмме длиной L_j бит.

Динамический диапазон $D_{(N)}$ определяется как максимально возможное значение кода позиционных чисел $N(u)_j$ для заданного значения основания λ_u .

Вторая особенность вызвана тем, что значения кодов $N(u)_i$ формируются для позиционных чисел равномерной длины $n_{\text{дин}}$. Это обусловлено тем, что позиционное число формируется на базе элементов отдельных строк массивов динамической составляющей. Поэтому значения кодов $N(u)_j$ будут принимать различные значения для разных строк. Отсюда

для сокращения кодовой избыточности предлагается длину кодограммы L_j выбирать неравномерной, т.е. $L_j = \text{VAR}$.

Для предложенной организации массивов D_u схема вычисления позиционных чисел для строк $D_{u,j} = \{d_{1,j}; d_{2,j}; \dots, d_{i,j}, \dots, d_{j,m_{\text{дин}}}\}$ этих массивов с учетом адаптивного выбора основания λ_u задается выражениями:

$$N(u)_j = \sum_{j=1}^{m_{\text{дин}}} d_{i,j} P_j; \quad (9)$$

$$P_j = \lambda_u^{(m_{\text{дин}} - j)}; \quad (10)$$

$$\lambda_u = \max \{d; j\} + 1, \quad (11)$$

$$\begin{matrix} 1 \leq j \leq n_{\text{дин}} \\ 1 \leq i \leq m_{\text{дин}} \end{matrix}$$

где $N(u)_j$ – значение кода для j -го позиционного числа j -го столбца u -го массива динамической составляющей; P_j – весовой коэффициент j -го элемента позиционного числа; λ_u – основание u -го массива динамической составляющей.

Рассмотрим пример вычисления кодов для позиционных чисел строк массива динамической составляющей.

Пример 3. Вычислим $N(u)_j$ для кодов позиционных чисел массивов D_u :

$$D_u = \begin{vmatrix} 10 & 23 & 30 \\ 20 & 18 & 9 \\ 4 & 10 & 15 \end{vmatrix}$$

Вначале определим позиционные числа для строк массива $D_{u,i}$ динамической составляющей. Для этого найдем основание λ_u и вычислим по формуле (10) их накопленные произведения P_j . Величина основания $\lambda_u = 31$. Тогда весовой коэффициент $P_1 = 31^2 = 961$; $P_2 = 31^1 = 31$; $P_3 = 31^0 = 1$. На основе этих значений по формуле (9) вычислим значения $N(u)_j$ для кодов позиционных чисел $D_{u,j}$:

$$N(u)_1 = 10 \times 961 + 20 \times 31 + 4 \times 1 = 10234; L_1 = 14 \text{ бит};$$

$$N(u)_2 = 23 \times 961 + 18 \times 31 + 10 \times 1 = 22671; L_2 = 15 \text{ бит};$$

$$N(u)_3 = 30 \times 961 + 9 \times 31 + 15 \times 1 = 29124; L_3 = 15 \text{ бит}.$$

Максимальное количество разрядов, затрачиваемое на представление позиционных чисел динамической составляющей равно 15 разрядам. При этом на кодовое представление исходного фрагмента изображения (фрагмент классифицируется как сильнонасыщенный динамическими составляющими) понадобится 24 разряда. Следовательно, за счет позиционного кодирования объем динамической составляющей снизится на 38%.

Позиционные коды позволят компактно представить массивы динамической составляющей, если соответствующие максимальные значения λ_u имеют ограниченные значения, т.е.: $\lambda_u < d_{\max}$, $u = \overline{1, v_{\text{дин}}}$. Здесь d_{\max} - максимальное числовое значение динамической составляющей

Таким образом, в результате предложенного кодирования массивы динамической составляющей заменяются последовательностью кодовых значений позиционных чисел с адаптивным основанием $\{N(u)_1, \dots, N(u)_j, \dots, N(u)_{m_{\text{дин}}}\}$.

Рассмотрим интерпретацию предложенного кодирования. Каждой строке массива динамической составляющей, рассматриваемой как позиционное число, ставится в соответствие один код $N(u)_j$ неравномерной длины. Тогда можно сказать, что за счет позиционного кодирования с адаптивным основанием равномерные части динамической составляющей заменяются кодограммами неравномерной длины. Поэтому всю цепь последовательных преобразований можно рассматривать как равномерное векторное квантование с последующим неравномерным распределением количества двоичных разрядов (рис. 7).

Обозначим данное преобразование следующим образом:

$$N(u)_j = f_{\text{рвк}} \{D_{u,j}; \lambda_u\};$$

$$L_j = f_{\text{нрр}} \{N(u)_j; \delta_u\} = f_{\text{нрр}} \{f_{\text{рвк}} \{D_{u,j}; \lambda_u\}; \delta_u\},$$

где $f_{\text{рвк}}$ - функционал, задающий равное векторное квантование динамической составляющей; $f_{\text{нрр}}$ - функционал, описывающий процесс распределения количества двоичных разрядов под каждый код позиционного числа.

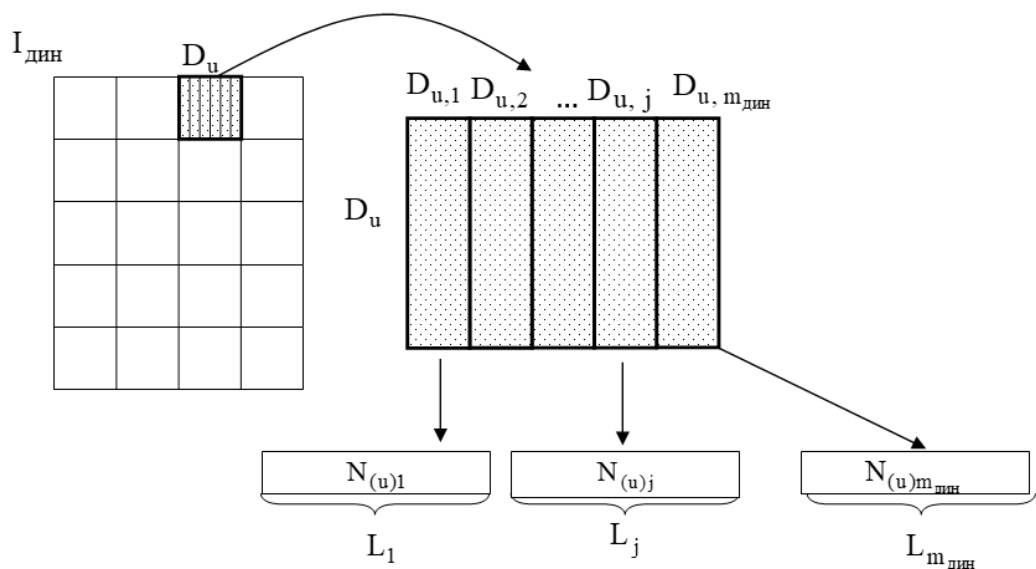


Рис. 7. Схема получения кодов – номеров для частей фрагмента изображения

Поскольку с одной стороны, количество элементов в позиционном числе является равномерным, а с другой стороны значения их элементов имеет

неравномерное распределение на натуральной оси, то значение кода $N(u)_j$ будет неравномерным, т.е.:

$$N(u)_j = \text{VAR} .$$

В этом случае, если под каждый код формировать кодограмму заранее заданной равномерной длины L_c , будет образовываться кодовая избыточность R_c . Это обусловлено появлением незначимых старших разрядов кодограммы, как показано на рис. 8:

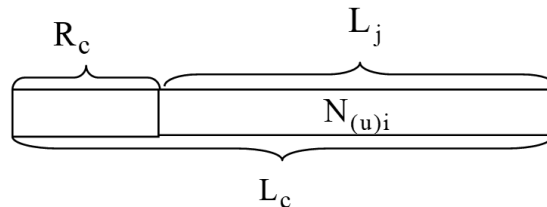


Рис. 8. Схема формирования кодовой избыточности

Кодовая избыточность R_c определяется следующей формулой:

$$R_c = L_c - L_j,$$

где L_j - количество разрядов на представление значения кода позиционного числа, формируемое адаптивно для каждого столбца массива динамической составляющей D_u ; L_c - заранее выбранная равномерная длина кодограммы, не учитывающая структурные особенности массивов динамической составляющей D_u .

В связи с чем, для сокращения кодовой избыточности предлагается формировать длину кодограммы, учитывая структурные особенности каждого столбца $D_{u,j}$ массива динамической составляющей. При этом необходимо учитывать, что длина кодограммы будет неравномерной, т.е., для каждой строки массива динамической составляющей в общем случае может выполняться неравенство:

$$L_1 \neq \dots L_j \neq \dots L_{m_{\text{дин}}} .$$

Для таких условий существуют две базовые стратегии позиционирования кодограмм сжатого представления.

Первая стратегия – заключается в использовании маркерных разделителей между кодограммами. Маркерные разделители представляют собой кодовую посылку, содержание которой не может повторяться ни в одной кодограмме. С одной стороны, это позволяет обеспечить разделение кодограмм на приемной стороне, но с другой стороны, это приводит к увеличению объема сжатого представления.

Для устранения данного недостатка предлагается использовать вторую стратегию. Суть второй стратегии заключается в том, что количество разрядов под кодограмму выбирается с учетом особенностей динамических диапазонов обрабатываемых строк динамической составляющей. При этом для сокращения служебных данных предлагается использовать только ту

служебную информацию, которая использовалась в процессе формирования кода $N(u)_j$.

В случае формирования кода для позиционного числа предлагается учитывать следующие их свойства.

Максимальное значение кода $N(u)_{\max}$ позиционного числа для заданного основания λ_u определяется накопленным произведением оснований его элементов, т.е.:

$$N(u)_{\max} = \lambda^{m_{\text{дин}}} - 1.$$

Откуда значение кода $N(u)_j$ для позиционных чисел с адаптивным основанием λ_u будет ограничено следующей величиной:

$$N(u)_j \leq \lambda^{m_{\text{дин}}} - 1.$$

Отсюда количество двоичных разрядов L_j на представление кода $N(u)_j$ по адаптивному основанию λ_u будет иметь следующую верхнюю границу:

$$L_j = [\log_2 N(u)_j] + 1 \leq [\log_2 \lambda^{m_{\text{дин}}} - 1] + 1. \quad (9)$$

Как следует из выражения (9) правая часть неравенства зависит от величины адаптивного основания λ_u и длины столбца $m_{\text{дин}}$ массива динамической составляющей. Величина λ_u используется в процессе кодирования, т.е. не является дополнительной служебной информацией, поэтому для позиционирования кодограмм позиционных чисел предполагается осуществлять распределение количества разрядов под каждую кодограмму в соответствии с неравенством (9). В этом случае длина кодограммы L_j будет равна:

$$L_j = [\log_2 \lambda^{m_{\text{дин}}} - 1] + 1,$$

тогда $\delta = \lambda_u$.

В соответствии с предложенной стратегией количество разрядов под каждую кодограмму в пределах одного массива динамической составляющей будет постоянным. Но для разных массивов динамической составляющей затраты количества разрядов под каждую кодограмму будут неравномерным. Это объясняется тем, что будут различия значения оснований λ_u . Поэтому предлагается принцип распределения количества разрядов под кодограмму называть локально-равномерным.

Выводы

1. С увеличением коэффициента корреляции между смежными кадрами увеличивается степень сжатия двоичной маски дифференциально-представленного кадра.

2. Степень сжатия двоичной маски дифференциально-представленного кадра изменяется от 3 до 21 в зависимости от коэффициента корреляции между смежными кадрами.

3. Оценка информативности битового представления двоичной маски дифференциально-представленного кадра на основе учета неравновесности оснований длин двоичных серий не требует увеличения сложности программно-аппаратной реализации.

4. За счет двухалфавитного кода мощности объем двоичной маски дифференциально-представленного кадра относительно одноалфавитного кода снизится на 17%.

5. Предложены требования относительно разбивки динамической составляющей на массивы, учитывающие с одной стороны необходимость дополнительного сокращения избыточности, с другой стороны исключение случаев переполнения кодового слова.

6. Разработан метод сжатия массивов динамической составляющей на основе одномерного позиционного кодирования с адаптивным выбором основания при этом, код позиционного числа формируется для отдельных строк массива динамической составляющей, а длина кодового слова выбирается неравномерной длины.

7. Длины кодовых слов позиционируются друг относительно друга в соответствии с известным значением длины позиционного числа $m_{\text{дин}}$ и адаптивным основанием $\lambda_{\text{ц}}$.

Литература

1. Barannik, V., Podlesny, S., Krasnorutskiy, A., Musienko, A. and Himenko, V. (2016), The ensuring the integrity of information streams under the cyberattacks action, 2016 IEEE East-West Design & Test Symposium (EWDTs), Yerevan, pp. 1-5, doi: 10.1109/EWDTs.2016.7807752.

2. Barannik, V., Ryabukha, Yu. and Podlesnyi, S. (2017) Structural slotting with uniform redistribution for enhancing trustworthiness of information streams, Telecommunications and Radio Engineering (English translation of *Elektrosvyaz and Radiotekhnika*, No 76(7) , pp. 607, doi: /10.1615/TelecomRadEng.v76.i7.40

3. Barannik, V., Podlesny S., Yalivets K. and Bekirov A. The analysis of the use of technologies of error resilient coding at influence of an error in the codeword, (2016) 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, 2016, pp. 52-54, doi: 10.1109/TCSET.2016.7451965.

4. Barannik, V., Krasnorutskiy, A., Ryabukha, Y. and Okladnoy, D. (2016) Model intelligent processing of aerial photographs with a dedicated key features interpretation, Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, pp. 736, doi: 10.1109/TCSET.2016.7452167.

5. Баранник Д.В., Баранник В.В., Красноруцкий А.А., Хаханова А.В. Метод снижения информационной интенсивности достаточно информативных сегментов аэрофотоснимка //Радиоэлектроника и информатика. 2018. № 1(75).С. 71-75.

6. Barannik D.V., Barannik V.V., Hahanova A.V., Medvedev D.O., Strelbtskiy M. Method of Ciphergrams Coding for Increasing the Effectiveness of Technologies of Selective Cyber-Protection // Radioelectronics and informatics. 2016. N. 4. P. 37-44.