УДК 621.391.1

# ОЦЕНКА ВЕРОЯТНОСТИ ПОЯВЛЕНИЯ СИМВОЛА ПРИ АДАПТИВНОМ ДВОИЧНОМ АРИФМЕТИЧЕСКОМ КОДИРОВАНИИ В ЗАДАЧАХ СЖАТИЯ ВИДЕОИНФОРМАЦИИ

***Беляев Е.А., Тюрликов А.М.***

*Представлено описание ряда известных алгоритмов адаптивной оценки вероятности появления символа на выходе двоичного источника. Предложен алгоритм «виртуального скользящего окна». Приведены результаты практического использования алгоритма для стандарта сжатия видеоинформации H.264/AVC.*

**Введение**

В современных алгоритмах сжатия информации широко применяется адаптивное арифметическое кодирование данных с заранее неизвестной статистикой. Предполагается, что эти данные можно разбить на последовательность интервалов так, что статистические характеристики данных внутри каждого интервала постоянны, но могут существенно отличаться для разных интервалов. Степень сжатия таких данных во многом зависит как от точности оцениваемой внутри каждого интервала статистики, так и от скорости адаптации к изменяющейся между интервалами статистике.

В настоящей работе рассматриваются алгоритмы адаптивной оценки вероятности появления символа на выходе источника с двоичным алфавитом, сочетающие в себе как точность оценки, так и скорость адаптации.

Ниже показано, что описанные в работах [1,3-6] алгоритмы адаптивной оценки требуют высоких затрат как памяти так и вычислительных ресурсов кодера и декодера. Один из подходов, позволяющих понизить эти затраты, заключается в использовании конечного автомата при оценке вероятности [9-15]. Такой подход применяется, например, в стандарте сжатия видеоинформации H.264/AVC [16]. Конечный автомат хранится в памяти кодера и декодера и используется при оценке вероятности для всех контекстных моделей, имеющихся в стандарте. Подобная реализация не требует операций умножения и деления при вычислении оценок вероятности. Недостаток такого подхода состоит в том, что при использовании одного и того же конечного автомата не учитываются отличия статистических свойств двоичных источников, соответствующих различным контекстным моделям. Один из способов, позволяющих учесть статистические отличия источников, заключается в использовании нескольких конечных автоматов [18], что приводит к дополнительному увеличению затрат памяти.

Второй способ, позволяющий учесть различия статистических свойств двоичных источников может быть реализован путем применения алгоритма оценки вероятности с периодическим масштабированием счетчиков [2]. Такой способ оценки позволяет повысить степень сжатия, но при этом существенно увеличивает требования к вычислительным ресурсам кодера и декодера, так как в нем используются операции умножения и деления.

В данной работе предложен целочисленный алгоритм «виртуального скользящего окна», который, с одной стороны, не использует операции умножения и деления при вычислении оценки вероятности и является наиболее предпочтительным с точки зрения степени сжатия, с другой стороны. Схожая целочисленная реализация адаптивной оценки вероятности описана в работе [19]. Однако она является более сложной как с вычислительной точки зрения (используется операция умножения), так и с точки зрения количества параметров алгоритма, которые необходимо назначить перед началом кодирования.

**Алгоритм адаптивной оценки с периодическим масштабированием счетчиков.** Пусть на вход кодера подается последовательность двоичных символов $x_1, x_2, ..., x_N$. Для оценки вероятности появления символа $x_{t+1} \in \{0,1\}, 1 \le t \le N,$ на выходе двоичного источника может быть использована оценка Кричевского-Трофимова [1]. Согласно этой оценке, вероятность того, что символ $x_{t+1}$ будет равен единице

$$\hat{p}_{t+1} = \frac{n_t^1 + 1/2}{n_t^0 + n_t^1 + 1} \qquad (1)$$

где $n_t^1$ и $n_t^0$ - число единиц и нулей в последовательности $x_1, x_2, ..., x_t$.

Оценка (1) имеет следующие недостатки. Во-первых, данная оценка неэффективна при сжатии данных с изменяющейся статистикой. Во-вторых, при реализации на практике существует вероятность переполнения счетчиков, содержащих число единиц и нулей в последовательности поступивших символов. Описанные недостатки (см., например, работу [2]) могут быть устранены путем периодического масштабирования счетчиков. Если выполняется условие $\min\{n_t^0, n_t^1\} > N_{\min}$ или $\max\{n_t^0, n_t^1\} > N_{\max}$, то значения счетчиков масштабируются следующим образом:

$$\begin{cases} n_0(t) = \beta n_0(t) \\ n_1(t) = \beta n_1(t) \end{cases},$$

где $\beta \in (0...1]$ - коэффициент масштабирования, а $N_{\min}$ и $N_{\max}$ - пороговые значения счетчиков.

**«Скользящее окно» и его аппроксимации.** Другой способ устранения описанных выше недостатков заключается в использовании конструкции так называемого «скользящего окна» [3], в котором вероятность появления очередного символа $x_{t+1}$ источника определяется за счет анализа содержимого окна, то есть последова-

тельности символов $x_{t-W+1}x_{t-W+2}...x_t$, где $W \geq 1$ - длина окна. После кодирования очередного символа содержимое окна сдвигается на одну позицию, новый символ $x_{t+1}$ заносится в освободившуюся ячейку, а последний символ $x_{t-W+1}$ удаляется. Оценка вероятности появления единицы на выходе двоичного источника при использовании «скользящего окна» определяется по формуле (1) с учетом того, что $n_t^0$ и $n_t^1$ - число нулей и единиц в окне соответственно и $n_t^0 + n_t^1 = W$. Основным недостатком данного подхода является необходимость хранения последних $W$ закодированных символов в памяти кодера и декодера.

При аппроксимации «скользящего окна» в памяти хранится только число символов в окне, которое вычисляется по некоторому правилу. Одно из таких правил, названное «мнимым скользящим окном», предложено в [4,5] для двоичного источника и в [3] для недвоичного источника. Согласно этому правилу, после кодирования символа $x_t$, из окна удаляется не последний, а случайный символ $y_t \in \{0,1\}$, и число единиц в окне
$n_{t+1}^1 = n_t^1 - y_t + x_t$,
где $y_t$ - случайна величина, которая генерируется со следующими вероятностями:

$$\begin{cases} \Pr\{y_t = 1\} = \dfrac{n_t^1}{W}, \\ \Pr\{y_t = 0\} = 1 - \dfrac{n_t^1}{W}. \end{cases}$$

В алгоритме «мнимого скользящего окна» требуется генерировать случайную величину $y_t$, значение которой должно поступать на вход кодера и декодера. Поэтому в работе [3] предлагается использовать специальным образом формируемую псевдослучайную последовательность, что затрудняет реализацию алгоритма.

В [6-8] показано, как можно отказаться от генерирования случайной величины. Для этого случайная величина $y_t$ заменяется ее математическим ожиданием. При этом правило пересчета числа единиц в окне изменяется следующим образом:

$$n_{t+1}^1 = n_t^1 - \frac{n_t^1}{W} + x_t = \left(1 - \frac{1}{W}\right) n_t^1 + x_t. \quad (3)$$

В соответствии с (3), вероятность появления единицы

$$\hat{p}_{t+1} = \left(1 - \frac{1}{W}\right)\hat{p}_t + \frac{1}{W} x_t. \quad (4)$$

Выражение (4) можно получить и исходя из других соображений. В [9] предлагается использовать геометрически убывающую оценку вероятности:

$$\hat{p}_{t+1} = \gamma\left(x_t + (1-\gamma)x_{t-1} + (1-\gamma)^2 x_{t-2}...\right), \quad (5)$$

где $\gamma$ - действительное число, $\gamma \in [0...1]$.

После записи в рекуррентной форме выражение (5) примет следующий вид:

$$\hat{p}_{t+1} = (1-\gamma)\hat{p}_t + \gamma x_t \quad (6)$$

Аналогичное правило вычисления приведено в [6,10]. При значении $\gamma = 1/W$ выражения (4) и (6) совпадают. Это означает, что проведение процедуры «дерандомизации» «мнимого скользящего окна» приводит к частному случаю геометрически убывающей оценки вероятности (подразумевается, что $W$ принимает целые значения).

**Реализация алгоритмов оценки вероятности при помощи конечного автомата**. Во многих случаях при реализации ранее рассмотренных алгоритмов оценки вероятности целесообразно воспользоваться конечным автоматом, заданным в виде таблиц состояний и переходов. Каждое состояние автомата соответствует некоторой оценке вероятности. В зависимости от значения входного символа $x_t$ происходит переход из одного состояния автомата в другое. Подобная реализация не требует операций умножения и деления при вычислении оценок вероятности. Кроме того, фиксированный набор оценок вероятностей позволяет отказаться от операции умножения при выполнении собственно арифметического кодирования. Этот способ оценки вероятности используется в таких реализациях арифметического кодера как Q-coder [11], его модификациях QM-coder [12] и MQ-coder [13], а также в Quasi-arithmetic coder [10], Z-coder [14], ELS-coder [9]. Табличная реализация алгоритма оценки с периодическим масштабированием счетчиков рассмотрена в работе [15].

Покажем, каким образом происходит переход от алгоритма оценки вероятности к конечному автомату на примере стандарта H.264/AVC [16]. Для кодирования недвоичных данных в этом стандарте может использоваться контекстный адаптивный двоичный арифметический кодер (Context-Based Adaptive Binary Arithmetic Coder). Перед кодированием очередной недвоичный символ при помощи процедуры бинаризации отображается в некоторую двоичную последовательность. Для каждого двоичного символа этой последовательности из заранее определенного фиксированного множества по некоторому правилу выбирается контекстная модель. Каждая контекстная модель содержит описание двоичного источника (фактически текущую оценку вероятности). Двоичный символ кодируется с использованием оценок вероятностей, задаваемых выбранной контекстной моделью. По завершении кодирования производится модификация оценок вероятностей соответствующей контекстной модели. Для кодирования двоичной последовательности используется M-coder, рассмотренный в работе [17]. Алгоритм оценки вероятности в M-coder основан на выражении (6), реализованном путем использования конечного автомата (см. рис.1), состоящего из 64-х состояний. Каждое состояние этого конечного автомата определяет фиксированную оценку вероятности входного символа. При этом входные символы делятся на наиболее вероятные символы (Most Probable Symbol – MPS) и наименее вероятные символы (Least Probable Symbol – LPS). Множество значений оценок вероятности $\left\{\hat{p}_0, \hat{p}_1, ..., \hat{p}_{63}\right\}$ задается следующим образом:
$\hat{p}_i = (1-\gamma)\hat{p}_{i-1}$, где $i = 1,...,63$, $\hat{p}_0 = 0.5$, $\gamma = 1 - \left(\dfrac{\hat{p}_{\min}}{0.5}\right)^{1/63}$,
$\hat{p}_{\min} = 0.01875$.

При этом оценка вероятности

$$\hat{p}_{t+1} = \begin{cases} (1-\gamma)\hat{p}_t + \gamma, \text{если } x_t \text{- наименее вероятный символ}, \\ \max\left[(1-\gamma)\hat{p}_t, \hat{p}_{62}\right], \text{в противном случае}. \end{cases}$$

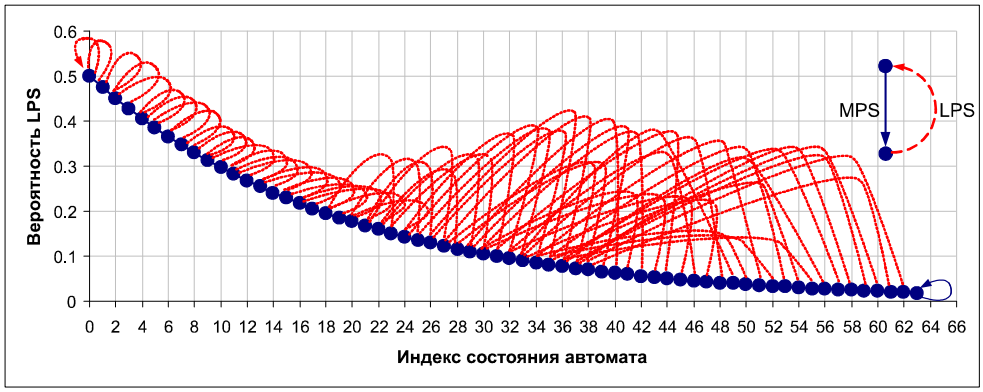


*Рис.1 Оценка вероятности в стандарте H.264/AVC*

В схеме кодирования, предложенной в [17], параметр адаптации $\gamma$ одинаков для источников, соответствующих различным контекстным моделям. Однако можно достичь лучшего сжатия, если каждому источнику ставить в соответствие собственное значение $\gamma$. Один из способов, позволяющих реализовать этот подход, заключается в использовании нескольких значений $\gamma$, каждому из которых соответствует конечный автомат, заданный в виде таблиц состояний и переходов (подобный метод описан в работе [18]).

**«Виртуальное скользящее окно».** В настоящей работе предлагается целочисленная реализация (6) без использования таблиц состояний и переходов. Можно дать следующую интерпретацию работы алгоритма «мнимого скользящего окна». Пусть имеется «скользящее окно» из $W$ ячеек. При поступлении очередного символа случайным образом выбирается одна ячейка. Символ, находящийся в этой ячейке заменятся на поступивший символ. Тогда правило пересчета (6) при целочисленной реализации допускает следующую интерпретацию. Имеется «скользящее окно» из $cW$ ячеек, где $c$ - параметр алгоритма. Значение поступившего символа заносится в $c$ ячеек, выбранных случайным образом. При этом из окна удаляется среднее число единиц в выбранных $c$ ячейках. Тогда число единиц $s_{t+1}$ в окне из $cW$ ячеек после кодирования очередного символа $x_t$ можно пересчитывать по следующему правилу:

$$s_{t+1} = \begin{cases} s_t + \left\lfloor \dfrac{cW - s_t + W/2}{W} \right\rfloor, \text{ при } x_t = 1, \\ s_t - \left\lfloor \dfrac{s_t + W/2}{W} \right\rfloor, \text{ при } x_t = 0. \end{cases} \quad (7)$$

Далее в тексте правило пересчета (7) называется «виртуальным скользящим окном». Оценка вероятности появления единицы для алгоритма «виртуального скользящего окна»

$$\hat{p}_t = \frac{s_t}{cW}. \quad (8)$$

При аппроксимации алгоритма «скользящего окна» алгоритмом «виртуального скользящего окна», параметр $c$ необходимо выбрать так, чтобы минимальные и максимальные оценки вероятностей для этих алгоритмов совпадали. Рассмотрим сначала, каким образом будет обеспечено совпадение минимальных оценок вероятностей. Для этого укажем следующее множество наборов входных данных, для которых оценка вероятности принимает минимальное значение. Сначала подается произвольная конечная последовательность длины $k$. Затем подается последовательность из $t$ нулей. Далее эти последовательности будем называть начальной и заключительной последовательностями соответственно. В этом случае, при любой начальной последовательности должно выполняться равенство следующих пределов:

$$\lim_{t\to\infty} \frac{n^1_{k+t} + 1/2}{W+1} = \lim_{t\to\infty} \frac{s_{k+t}}{cW}, \text{ при } \forall k < \infty. \quad (9)$$

Левая и правая части (9) соответствуют оценке вероятности при использовании «скользящего окна» и «виртуального скользящего окна» для случая, когда заключительная последовательность имеет неограниченную длину.

Рассмотрим сначала левую часть равенства (9). При поступлении на вход кодера, использующего «скользящее окно», заключительной последовательности длины $W$ и более количество единиц в окне станет равным нулю, поэтому левая часть (9) не зависит от начальной последовательности и равна:

$$\lim_{t\to\infty} \frac{n^1_{k+t} + 1/2}{W+1} = \frac{1}{2(W+1)}, \text{ при } \forall k < \infty. \quad (10)$$

Теперь рассмотрим правую часть равенства (9). Если перед началом кодирования $s_1 \in \{0,1,...,W/2 - 2\}$, то всегда найдутся две такие начальные последовательности, что значения правой части (9) будут различны при заключительной последовательности любой длины. Например, если начальная последовательность состоит из нулей, то значение $s_{t+k} = s_1$ и не меняется в процессе работы кодера, так как значение

$$\left\lfloor \frac{s_t + W/2}{W} \right\rfloor = 0, \text{ при } s_t < W/2.$$

Если начальная последовательность содержит хотя бы одну единицу, то $s_{t+k} > s_1$. Поэтому необходимым условием выполнения равенства (9) является выбор начального значения $s_1 \geq W/2 - 1$. Очевидно, что при таком выборе для любой начальной последовательности правая часть (9) равна:

$$\lim_{t\to\infty}\frac{s_{k+t}}{cW}=\frac{W/2-1}{cW},\ \text{при}\ \forall k<\infty. \qquad (11)$$

Приравнивая (10) и (11), получим:

$\frac{1}{2(W+1)}=\frac{W/2-1}{cW}$, тогда $c=W-1-\frac{2}{W}$.

Аналогичным образом можно показать, что при выборе начального значения $s_1\le cW-W/2+1$, и параметре $c=W-1-2/W$, будут совпадать и максимальные оценки вероятностей. При $W>>1$, можно считать, что $c=W-1-\frac{2}{W}\approx W$.

Таким образом, для совпадения соответствующих оценок вероятности алгоритмов «скользящего окна» и «виртуального скользящего окна» необходимо и достаточно, чтобы перед началом кодирования значение $s_1\in\{W/2-1,...,cW-W/2+1\}$ и параметр алгоритма $c=W$.

Применение предложенного алгоритма «виртуального скользящего окна» позволяет отказаться от использования таблиц состояний и переходов. При выборе $W=2^i$, где $i$- положительное целое число, вместо операции деления можно использовать операцию побитового сдвига. Таким образом, при вычислении оценки вероятности используются только операции сложения и сдвига. Характеристики алгоритма легко изменяются путем выбора различных значений длины окна $W$. Поэтому каждому источнику можно назначить собственное значение параметра $\gamma=1/W$ без использования таблиц.

**Практическое сравнение алгоритмов.** Сравнение практической эффективности алгоритмов оценки вероятности проводилось на примере стандарта сжатия видеоинформации H.264/AVC. Алгоритмы сжатия видеоинформации сравниваются по двум основным параметрам: визуальное качество декодированной видеопоследовательности и битовая скорость (число бит на выходе кодера в единицу времени). Для получения практических результатов алгоритм оценки вероятности, используемый в кодере и декодере H.264/AVC, был заменен сначала алгоритмом с периодическим масштабированием счетчиков, а затем алгоритмом «виртуального скользящего окна». Для этого использовался открытый JVT кодек, версии JM. 10.2 (FREext), поддерживающий данный стандарт. Результаты были получены для тридцати первых кадров известных тестовых HDTV видеопоследовательностей (“riverbed”, “rush hour”, “station”, “sunflower”, “tractor”) разрешением 1920x1080. Алгоритм с периодическим масштабированием счетчиков был взят из работы [2], в которой $N_{\max}=\infty$,и коэффициент масштабирования

$\beta=\frac{N_{\min}+\Delta-1}{N_{\min}+\Delta}$, где $\Delta=0.4$.

При реализации алгоритма «виртуального скользящего окна» каждой контекстной модели с номером $i$ необходимо назначить параметр кодирования $W_{opt}(i)$. Для этого в процессе кодирования тестовой видеопоследовательности (в данном случае использовалась последовательность “tractor”) для каждого двоичного символа с номером $t$ источника, соответствующего контекстной модели с номером $i$, и для каждой длины окна $W_l=2^l, l=3,4,...,10$, вычислялась оценка вероятности появления единицы

$$\hat{p}_t(W_l,i)=\frac{s_t(i)}{W_l^2},$$

где $s_t(i)$- число единиц в «виртуальном скользящем окне» для контекстной модели с номером $i$ перед кодированием символа $x_t(i)$.

По завершении кодирования вычислялись оценки битовых затрат при адаптивном арифметическом кодировании

$\hat{R}(W_l,i)=\sum_t \hat{r}_t(W_l,i)$, где

$$\hat{r}_t(W_l,i)=\begin{cases}-\log_2\hat{p}_t(W_l,i), & \text{если } x_t(i)=1,\\ -\log_2\left(1-\hat{p}_t(W_l,i)\right), & \text{если } x_t(i)=0.\end{cases}$$

Для контекстной модели с номером $i$ параметр кодирования $W_{opt}(i)$ устанавливался равным

$$W_{opt}(i)=\arg\min_{W_l}\hat{R}(W_l,i).$$

Начальное значение $s_1(i)$ определялось следующим образом:

$$s_1(i)=\min\left\{W_{opt}^2(i)-W_{opt}(i)/2+1,\max\left\{W_{opt}(i)/2-1,\left\lfloor W_{opt}^2(i)\cdot\hat{p}(i)\right\rfloor\right\}\right\},$$

где $\hat{p}(i)$ - определенная в стандарте H.264/AVC начальная оценка вероятности единицы для контекстной модели с номером $i$.

В представленных ниже таблицах приведено уменьшение битовой скорости (в процентах) относительно оригинальной версии кодека, при фиксированном качестве, в зависимости от значения номера шага квантования (QP). Таблица №1 относится к случаю использования алгоритма с периодическим масштабированием счетчиков. Таблица №2 относится к случаю использования алгоритма «виртуального скользящего окна».

Таблица №1.

Уменьшение битовой скорости при использовании алгоритма с периодическим масштабированием счетчиков

| QP | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| riverbed | 0.72 | 0.67 | 0.40 | 0.23 | 0.65 |
| rush hour | 0.58 | 0.47 | 0.57 | 0.17 | 1.04 |
| station | 0.77 | 0.57 | -0.15 | 0.35 | 1.43 |
| sunflower | 0.25 | 0.19 | 0.05 | 0.22 | 1.66 |
| tractor | 0.53 | 0.61 | 0.79 | 0.89 | 1.45 |

Таблица №2.

Уменьшение битовой скорости при использовании алгоритма «виртуального скользящего окна»

| QP | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| riverbed | 0.98 | 0.93 | 0.93 | 0.90 | 0.90 |
| rush hour | 0.86 | 0.73 | 0.92 | 0.50 | 1.23 |
| station | 1.08 | 0.98 | 0.27 | 0.63 | 1.43 |
| sunflower | 0.68 | 0.50 | 0.45 | 0.58 | 1.58 |
| tractor | 0.84 | 0.90 | 1.10 | 1.10 | 1.62 |

Как видно из таблиц 1-2, назначение каждому двоичному источнику индивидуального параметра кодирования приводит к уменьшению битовой скорости (для таблицы №1 близкие результаты получены в [20]) при фиксированном качестве. В случае практической реализации такого подхода с использованием оценки с периодическим масштабированием счетчиков необходимо либо использовать операции умножения и деления при вычислении оценки вероятности либо вводить большое количество заранее вычисленных таблиц. Алгоритм «виртуального скользящего окна» может быть реализован как без использования таблиц, так и без операций умножения и деления. При этом он является наиболее предпочтительным с точки зрения эффективности кодирования.

**Литература**


1. R. E. Krichevski and V. E. Trofimov, “The performance of universal encoding” IEEE Trans. Inform. Theory, vol. IT-27, pp. 199–207, Mar. 1981.
2. D.L. Duttweiler and C. Chamzas, “Probability estimation in arithmetic and adaptive-Huffman entropy coders” IEEE Transactions on Image Processing Vol. 4, pp. 237-246 Mar. 1995
3. Рябко Б.Я., Фионов А.Н., Эффективный метод арифметического кодирования для источников с большими алфавитами // Проблемы передачи информации Т.35, №4. С.95-108, 1999.
4. T. Leighton and R. L. Rivest, “Estimating a probability using finite memory” IEEE Trans. Inform. Theory, vol. IT-32, pp. 733–742, Nov. 1986.
5. A. Zandi, G.G. Langdon, "Adaptation for non-stationary binary sources for data compression", p.224, 29th Asilomar Conference on Signals, Systems and Computers (2-Volume Set), 1996.
6. E. Meron and M. Feder, “Finite-Memory Universal Prediction of Individual Sequences” IEEE Trans. Inform. Theory, vol. 50-7, pp. 1506–1523, July 2004.
7. E. Belyaev, M. Gilmutdinov and A. Turlikov, “Binary Arithmetic Coding System with Adaptive Probability Estimation by “Virtual Sliding Window” // Proc. of the 10th IEEE International Symposium on Consumer Electronics – ISCE’06, St.-Petersburg, Russia, pp. 194–198, 2006.
8. Беляев Е.А. Использование «виртуального скользящего окна» при адаптивном арифметическом кодировании// Научная сессия ГУАП, посвященная всемирному Дню авиации и космонавтики и 65-летию ГУАП: Сб. докл./ ГУАП. СПб., 2006.
9. D. Withers, “The ELS-coder: a rapid entropy coder”, Data Compression Conference, p. 475,1997.
10. P. G. Howard and J. S. Vitter, “Pracical implementations of arithmetic coding”, in Image and Text Compression, Storer, Ed., pp. 85–112, Kluwer Academic,1992.
11. Pennebaker W. B., Mitchell J. L., Langdon G. G., Arps R. B. “An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder” // IBM J. Research and Development. Vol. 32, N.6. P. 717-726, 1988.
12. ITU-T and ISO/IEC JTC 1, ITU-T Recommendation T.81 and ISO/IEC 10918-1 Digital Compression and Coding of Continuous-Tone Still Images, ITU-T Recommendation T.81 and ISO/IEC 10918-1 (JPEG), Sep. 1992.
13. ITU-T and ISO/IEC JTC 1, JPEG2000 Image Coding System: Core Coding System, ITU-T Recommendation T.800 and ISO/IEC 15444-1 (JPEG2000 Part 1), 2000.
14. L. Bottou, P. G. Howard, and Y. Bengio, “The Z-coder adaptive binary coder”, Proc. IEEE Data Compression Conference, Snowbird (USA), pp. 13–22, 1998.
15. Detlev Marpe, Heiko Schwarz, Gabi Blättermann, Guido Heising, and Thomas Wiegand, **“**Context-Based Adaptive Binary Arithmetic Coding In JVT/H.26L”, Image Processing Department, Heinrich-Hertz-Institute, Berlin, Germany.
16. ITU-T and ISO/IEC JTC 1, Advanced video coding for generic audiovisual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (AVC), May 2003.
17. Detlev Marpe, Heiko Schwarz, and Thomas Wiegand, “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 620–636, July 2003.
18. Eeckhaut, H.; Schrauwen, B.; Christiaens, M.; Van Campenhout, J. “Tuning the M-coder to improve Dirac`s Entropy Coding”. WSEAS transactions on Information Science and Applications. WSEAS. Vol. 2 (10). pp. 1563-1571, 2005.
19. Семенюк В.В., Метод адаптивного кодирования двоичной информационной выборки. Известия вузов. Приборостроение.-Т.47, Вып. 5.-С.36-41, 2004.
20. Detlev Marpe and Thomas Wiegand, “A Highly Efficient Multiplication-Free Binary Arithmetic Coder and its Application in Video Coding”, Proceedings IEEE International Conference on Image Processing (ICIP ’03), vol. II, pp. 263–266, Oct. 2003.