

В.Ч. НГУЕН, А.А. ТРОПЧЕНКО
**ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ МЕТОДА
ЭНТРОПИЙНОГО КОДИРОВАНИЯ В СОВРЕМЕННЫХ
СТАНДАРТАХ ВИДЕОСЖАТИЯ**

Нгуен В.Ч., Тропченко А.А. **Повышение эффективности метода энтропийного кодирования в современных стандартах видеосжатия.**

Аннотация. Современные стандарты кодирования видеоданных имеют высокую эффективность кодирования, но скорость кодирования может быть улучшена для удовлетворения растущих потребностей мультимедийных приложений. В статье рассматриваются методы и алгоритмы энтропийного кодирования в стандартах кодирования видеоданных H.264/AVC и H.265/HEVC. Контекстно-зависимое адаптивное кодирование с переменной длиной кодового слова CAVLC (Context-based Adaptive Variable Length) для стандарта H.264/AVC изначально предназначалось для кодирования с потерями и как таковое не давало адекватной производительности для кодирования без потерь. Контекстно-зависимое адаптивное бинарное арифметическое кодирование CABAC (Context-Adaptive Binary Arithmetic Coding) — это метод энтропийного кодирования, впервые введенный в H.264/AVC и используемый в стандарте H.265/HEVC. Хотя он обеспечивает высокую эффективность кодирования, зависимости данных в H.264/AVC CABAC затрудняют распараллеливание и, таким образом, ограничивают его пропускную способность. Соответственно, во время стандартизации энтропийного кодирования для HEVC были рассмотрены как эффективность кодирования, так и пропускная способность. На основе анализа их достоинств и недостатков предложен метод энтропийного кодирования с последующим применением нумерационного иерархического кодирования. Он состоит из алгоритма CABAC и алгоритма нумерационного кодирования с применением иерархического подхода. Предложенный метод протестирован в среде разработки Visual C++ на различных тестовых видеопоследовательностях. Результаты экспериментов показали большую эффективность кодирования мультимедийных данных (уменьшает в среднем до 15% объема памяти хранения по сравнению с традиционным методом CABAC), но при этом метод требует большее время кодирования (примерно в два раза). Предложенный метод можно рекомендовать для применения в телекоммуникационных системах для решения задач хранения, передачи и обработки мультимедийных данных, где в первую очередь требуется большая степень сжатия.

Ключевые слова: энтропийное кодирование, метод CABAC, метод CAVLC, алгоритм нумерационного кодирования, H.265/HEVC, H.264/AVC.

1. Введение. Энтропийный кодер преобразовывает последовательности символов, представляющих элементы видеопоследовательности, в сжатый поток битов, который можно сохранять в файле или передавать по сетям связи [1, 2]. Этот кодер играет важную роль в целом процессе кодирования. Основными методами энтропийного кодирования служат коды Хаффмана [3] и арифметическое кодирование [3]. На их основе были разработаны современные методы и алгоритмы для кодирования бинарных нестационарных последовательностей такие. как методы CAVLC [5, 6] и CABAC [7, 8], которые широко

применяются в стандартах H.264 [9] и H.265 [10, 11]. Все эти алгоритмы в первую очередь ориентированы на снижение объема выходного файла. Именно эту величину и будем использовать в дальнейшем как показатель эффективности того или иного алгоритма. В работе [12] предлагается адаптивный двоичный арифметический кодер, но снижение объема оказывается небольшим (до 2.3% по стандарту H.264 и до 3.6% по H.265). В [13] предложен новый аппаратно-эффективный адаптивный кодер двоичного диапазона и его архитектура с очень крупномасштабной интеграцией. Экспериментальные результаты предложенного кодера показывают, что он значительно выигрывает у существующих кодеров (до 8% у MQ-кодера, и до 24.2% – M-кодера). Однако результаты получены для вейвлет-видеокодека с энтропийным кодером, подобным JPEG2000. А в [14] предполагается контекстно-адаптивное двоичное арифметическое кодирование с использованием кодовых слов с фиксированной длиной, которое обеспечивает упрощенную вычислительную сложность по сравнению с MQ-кодером JPEG2000 и M-кодером HEVC (в порядке примерно 2%). После анализа преимуществ и недостатков существующих методов был предложен новый.

2. Метод CAVLC [4, 5]. Метод CAVLC является одним из методов энтропийного кодирования, который используется в стандарте H.264/AVC. По своей сути, метод представляет собой технологию сжатия без потерь, как и большинство методов энтропийного кодера. CAVLC кодирование блока преобразования коэффициентов происходит следующим образом:

Стадия 1. Кодировать число ненулевых коэффициентов и замыкающих единиц $T1(\text{coeff_token})$.

coeff_token кодируется как общее количество ненулевых коэффициентов N и количество замыкающих $+/-1(T1)$. N может принимать значения от 0 (нет ненулевых коэффициентов в блоке 4x4) до 16 (16 ненулевых коэффициентов), а $T1$ от 0 до 3. В случае, если есть более 3-х замыкающих $+/-1$, только последние 3 рассматриваются как «особый случай», а другие кодируются как обычные коэффициенты.

Существуют 4 варианта выбора справочной таблицы (таблица 1) для кодирования coeff_token , описанные как $Num-VLC0$, $Num-VLC1$, $Num-VLC2$ и $Num-FLC$ (3 кодовые таблицы переменной длины и 1 кодовая таблица фиксированной длины).

Таблица 1. Справочная таблица для кодирования $coeff_token$

N	Вариант для кодирования $coeff_token$
0,1	$Num - VLC0$
2,3	$Num - VLC1$
4,5,6,7	$Num - VLC2$
8 и больше	$Num - FLC$

Выбор таблиц зависит от числа ненулевых коэффициентов в верхнем и левом закодированных блоках $N_{верх}$ и $N_{лев}$. Параметр N вычисляется следующим образом.. Если верхний и левый блоки доступны (то есть в том же закодированном слайсе), $N = (N_{верх} + N_{лев}) / 2$; если только верхний блок доступен, $N = N_{верх}$; если только левый блок доступен, $N = N_{лев}$; если ни один не доступен, $N = 0$.

Стадия 2. Кодировать знак каждого значения $T1$.

Каждое значение $T1(\pm 1)$ описывается с помощью $coeff_token$ ($0 = +, 1 = -$). Они кодируются, начиная со значения $T1$ с наивысшей частотой.

Стадия 3. Кодировать уровни остальных ненулевых коэффициентов.

Уровень (знак и величина) каждого оставшегося ненулевого коэффициента в блоке кодируется, начиная с самой высокой частоты. Выбор VLC таблицы для кодирования каждого уровня адаптируется в зависимости от величины каждого успешно закодированного уровня. Есть 7 таблиц для выбора: от $Level_VLC0$ до $Level_VLC6$ как показано в таблице 2. $Level_VLC0$ смещается в сторону более низких величин; $Level_VLC1$ в сторону несколько более высоких величин и так далее.

Таблица 2: Пороговое значение

Текущая VLC таблица	Пороговое значение
$VLC0$	0
$VLC1$	3
$VLC2$	6
$VLC3$	12
$VLC4$	24
$VLC5$	48
$VLC6$	N/A (наивысшая таблица)

Выбор таблицы осуществляется следующим образом:

- инициализировать таблицу $Level_VLC0$ (если не более 10 ненулевых коэффициентов и менее 3-х завершающих единиц, в этом случае начинаются с $Level_VLC1$);
- закодировать высокочастотные ненулевые коэффициенты;
- если величина этого коэффициента больше, чем предопределенное пороговое значение, перейти к следующей VLC таблице.

Стадия 4. Кодировать число нулей перед последним ненулевым коэффициентом.

$TotalZeros$ является суммой всех нулей, предшествующих наивысшему ненулевому коэффициенту в переупорядоченном массиве. Это число кодируется вместе с VLC .

Стадия 5. Кодировать все длины серий нулей.

Длины серий нулей, предшествующих каждому ненулевому коэффициенту (run_before), кодируется в обратном порядке. Параметр run_before кодируется для каждого ненулевого коэффициента, начиная с имеющего самую высокую частоту.

VLC для каждого пробегания нулей выбирается в зависимости от числа нулей, которые еще не закодированы ($ZerosLeft$) и run_before . Например, если имеется только 2 нуля, run_before может принимать только 3 значения (0, 1 или 2), и поэтому VLC не должен превышать 2 бита длины; если существует 6 нулей, следовательно, run_before может принимать 7 значений, и таблица VLC станет, соответственно, больше.

Недостаток этого метода заключается в том, что правильность предсказания таблицы VLC не высока. Правильность предсказания VLC таблицы для тестовых последовательностей составляет около 55% в среднем [15]. В работе [16] было проведено сравнение полученных скоростей передачи битов между H.264 CAVLC и идеальным алгоритмом CAVLC. Идеальный CAVLC имеет 100% правильность предсказания таблицы VLC . В результате, если мы используем правильную таблицу VLC для текущего блока 4x4, мы можем уменьшить скорость передачи данных примерно на 3%, по сравнению с CAVLC в H.264. Следовательно, если мы можем увеличить правильность прогнозирования таблицы VLC для каждой тестовой последовательности, мы можем повысить эффективность кодирования.

3. Метод САВАС [7, 8]. Метод САВАС основан на арифметическом кодировании с несколькими изменениями, чтобы адаптировать его

к потребностям стандартов видео кодирования [17] и имеет следующие особенности:

- кодирует двоичные символы, что снижает сложность метода и позволяет использование вероятностных моделей для наиболее часто используемых последовательностей битов;
- вероятностные модели выбираются адаптивно на основе локального контекста, так как режимы кодирования локально хорошо коррелированы;
- использует безраздельное деление диапазона с использованием квантованных вероятностных диапазонов и вероятностных состояний.

На рисунке 1 показана общая блок-схема кодирования одного элемента синтаксиса в САВАС. Процесс кодирования состоит из трех элементарных этапов:

Этап 1: бинаризация;

Этап 2: моделирование контекста;

Этап 3: бинарное арифметическое кодирование.

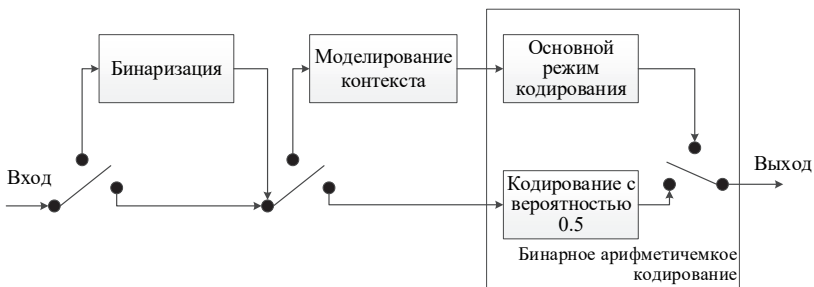


Рис. 1. Общая блок-схема кодирования методом САВАС

На первом этапе данный недвоичный синтаксический элемент однозначно отображается в двоичную последовательность. Когда дается двоичный элемент, этот шаг пропускается. В зависимости от режима кодирования для каждого элемента могут следовать один или два последовательных шага.

В так называемом основном режиме кодирования двоичной элемент входит в стадию моделирования контекста, где вероятностная модель выбирается таким образом, чтобы соответствующий выбор зависел от заранее закодированных элементов. Затем, после присвоения модели, двоичной элемент со своей моделью переходит к основному режиму кодирования, где и происходит конечная стадия арифметического кодирования вместе с последующей модификацией модели (например, если значение было «1», счетчик единиц увеличивается).

В качестве альтернативы для выбранных двоичных элементов выбирается кодирование с вероятностью 0.5 с помощью упрощенного механизма кодирования (рисунок 1).

В H.264/AVC арифметический кодер представляет собой совокупность процедур низкой сложности, в которых отсутствуют операции умножения. Процедуры включают в себя сдвиги и обращения к таблицам. Использование CABAC позволяет снизить битрейт в среднем на 10-15%. Наибольший выигрыш получается обычно при обработке чересстрочных сигналов ТВ.

В целом CABAC обеспечивает повышенную эффективность кодирования по сравнению с кодированием CAVLC. Чтобы доказать этот вывод проводим эксперимент для сравнения эффективности методов CAVLC и CABAC с помощью Эталонного программного обеспечения JM 19.0 основного профиля (H.264/1449610 AVC Reference Software). Сравнение осуществляется по качеству кодирования и требуемое время выполнения. Эксперимент проводится на платформе Windows 10 с процессором Intel (R) Core i5-4210U CPU @ 1,70 ГГц 2,40 ГГц и 6 ГБ оперативной памяти.

Чтобы сравнить эффективность метода CABAC над CAVLC в качестве тестовых используются 3 основных типа последовательностей: с малым количеством движущихся объектов в кадре, с их большим количеством и D1-последовательности полного разрешения (таблица 3).

Эффективность кодирования оценивается путем сравнения объема выходного потока, представленного в таблице 4. Результаты показали, что алгоритм CABAC уменьшает объем выходного файла в среднем до 10 % по сравнению с алгоритмом CAVLC.

Таблица 3. Тестовые видеопоследовательности

Тип	Видеопоследовательность	Расширение	Частота кадров
с малым количеством движущихся объектов в кадре	Bus_QCIF	176x144	15
	Football_QCIF	176x144	15
	Foreman_QCIF	176x144	15
	Ice_QCIF	176x144	15
	Mobile_QCIF	176x144	15
	City_CIF	352x288	30
с их большим количеством	Bus_CIF	352x288	15
	Football_CIF	352x288	30
	Foreman_CIF	352x288	30
	Ice_CIF	352x288	30
	Soccer_CIF	352x288	30
полного разрешения	Crew_D1	704x576	60
	Ice_D1	704x576	60

Таблица 4. Объем выходного потока разных типов видеопоследовательностей

Тип	Видеопоследовательность	Процент уменьшения объема метода CABAC над CAVLC, %			
		QP = 22	QP = 28	QP = 32	QP = 36
с малым количеством движущихся объектов в кадре	Bus_QCIF	2.04	2.22	3.28	6.17
	Football_QCIF	2.79	2.20	2.95	4.13
	Foreman_QCIF	6.27	6.86	7.92	9.98
	Ice_QCIF	3.09	3.63	5.23	7.87
	Mobile_QCIF	3.34	3.44	3.11	4.26
	City_CIF	4.00	4.57	6.66	9.94
с их большим количеством	Bus_CIF	3.53	3.83	5.01	7.23
	Football_CIF	3.76	4.31	4.82	6.69
	Foreman_CIF	7.63	8.84	10.67	13.16
	Ice_CIF	3.35	5.16	7.37	10.75
	Soccer_CIF	5.52	5.97	7.30	10.13
полного разрешения	Crew_D1	13.59	10.13	13.88	18.27
	Ice_D1	12.04	7.11	10.68	15.25
<i>В среднем</i>		<i>5.46</i>	<i>5.25</i>	<i>6.84</i>	<i>9.53</i>

Вторым по важности работы алгоритма видеокodeка показателем является время кодирования. Оба метода оцениваются путем сравнения среднего необходимого времени для кодирования каждой последовательности, показанной в таблице 5. Видно, что алгоритм CABAC требует больше времени кодирования, в среднем до 16% по сравнению с алгоритмом CAVLC.

Таблица 5. Среднее время для кодирования разных типов видеопоследовательностей

Тип	Видеопоследовательность	Степень увеличения среднего времени метода CABAC над CAVLC, %			
		QP = 22	QP = 28	QP = 32	QP = 36
с малым количеством движущихся объектов в кадре	Bus_QCIF	10.71	10.48	14.24	9.26
	Football_QCIF	15.89	14.53	19.03	13.07
	Foreman_QCIF	12.58	10.51	14.17	10.16
	Ice_QCIF	16.12	11.71	14.04	10.62
	Mobile_QCIF	16.91	13.94	42.38	12.13
	City_CIF	18.03	13.21	18.63	7.39
с их большим количеством	Bus_CIF	10.90	10.68	6.58	9.69
	Football_CIF	8.44	13.12	14.07	7.92
	Foreman_CIF	13.63	13.18	10.91	13.69
	Ice_CIF	13.75	11.56	6.97	13.349
	Soccer_CIF	12.58	14.22	11.87	19.68
полного разрешения	Crew_D1	16.87	17.03	14.38	9.408
	Ice_D1	17.62	12.46	15.32	19.37
<i>В среднем</i>		<i>14.16</i>	<i>12.82</i>	<i>15.58</i>	<i>11.98</i>

4. Модифицированный метод энтропийного кодирования.

Предложенный энтропийный кодер состоит из алгоритма САВАС и алгоритма нумерационного кодирования с применением иерархического подхода (рисунок 2).



Рис. 2. Общая схема кодирования предложенным методом

Входные данные кодера представляют собой параметры временной модели (векторы движения) и пространственной модели (коэффициенты преобразования), а также маркеры (коды, обозначающие точки синхронизации в видеопоследовательности) и заголовки (заголовки макроблоков, изображений, последовательностей и других объектов). После выполнения алгоритма САВАС, входные данные превращаются в бинарные последовательности, являющиеся входными для алгоритма нумерационного кодирования с применением иерархического подхода. После него получаются конечные выходные данные энтропийного кодера или, иными словами, сжатые данные видеокodeка.

Алгоритм нумерационного кодирования с применением иерархий [18, 19] включает в себя метод Линча — Дэвиссона [20, 21] и метод нумерационного кодирования ограниченных целых чисел.

Пусть $\{0,1\}^n$ — вектор n двоичных чисел и $x = (x_1, x_2, \dots, x_n)$ — элементы этого вектора, S — множество векторов n двоичных чисел. Определим n_S — число элементов в S и обозначим $n_S(x_1, x_2, \dots, x_k)$ — число элементов в S , для которого были определены первые k элементов.

Предложим, что вектор имеет w элементов 1 или $\sum_{j=1}^n x_j = w$.

Кодирование методом нумерационного кодирования с применением иерархий включает следующие стадии:

1. Построить дерево сумм (рисунок 3).
2. Кодировать $w_{p_{\max,1}}$ с использованием $\lceil \log_2(N+1) \rceil$ бит.
3. Выполнить цикл для p от $p_{\max} - 1$ до 1 с единичным шагом:

$$- \text{выполнить цикл для } i \text{ от } 1 \text{ до } \left\lceil \frac{N}{\prod_{j=1}^p n_j} \right\rceil;$$

– кодировать вектор $\{w_{p,(i-1)n_p+1}, \dots, w_{p,in_p}\}$ методом нумерационного кодирования ограниченных целых чисел.

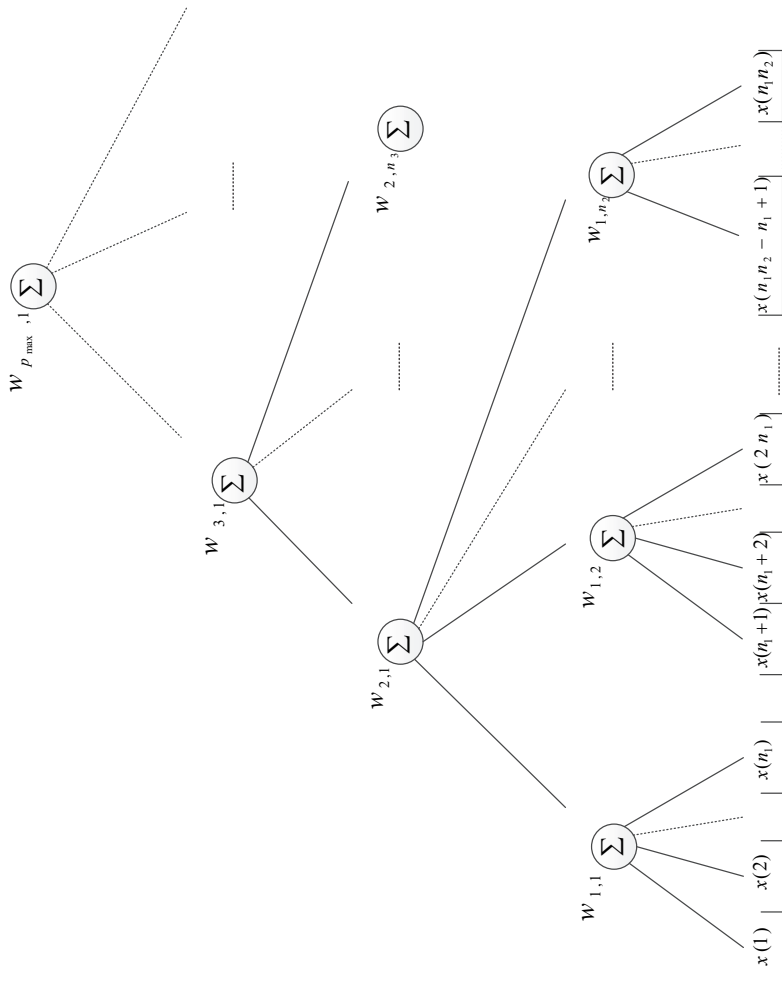


Рис. 3. Дерево сумм

4. Выполнить цикл для i от 1 до $\left\lceil \frac{N}{n_1} \right\rceil$:

– кодировать $\{w_{p,(i-1)n_p+1}, \dots, w_{p,in_p}\}$ методом Линча — Дэвиссона, чтобы получить выходную последовательность.

Процесс декодирования методом нумерационного кодирования с применением иерархий производится в таком же порядке, как и процесс кодирования.

4.1. Метод Линча — Дэвиссона. Процесс кодирования заключается в вычислении лексикографического индекса [18] (обычное словарное упорядочивание при интерпретации $0 < 1$) вектора $x \in S \subseteq \{0, 1, 2, \dots, M\}^n$ определяется по следующей формуле:

$$i_S(x) = \sum_{j=1}^n \sum_{m=0}^{x_j-1} x_j n_S(x_1, x_2, \dots, x_{j-1}, m). \quad (1)$$

Для построения выходного кода требуется $\left\lceil \log_2 C_w^n \right\rceil$ битов.

Процесс декодирования осуществляется по следующему алгоритму: если $i_S(x) > n_S(x_1, x_2, \dots, x_{k-1}, 0)$, $k = 1, 2, \dots, n$ то $x_k = 1$; в противном случае — $x_k = 0$. Выполнить до конца последовательности.

4.2. Метод нумерационного кодирования ограниченных целых чисел. Вектор x с размером n , каждый элемент которого x_i удовлетворяет условию: $0 \leq x_i \leq M$, где M — постоянное положительное целое число. По [19] имеем:

$$i_S(x) = \sum_{j=1}^{n-1} \sum_{m=0}^{x_j-1} f_M \left(\left(w - m - \sum_{i=1}^k x_i \right), n - j \right), \quad (2)$$

$$\text{где: } f_M(p, q) = \sum_{i=p-M}^p f_M(i, q-1), \quad f_M(p, 1) = \begin{cases} 1, & \text{если } : 0 \leq p \leq M \\ 0, & \text{иначе} \end{cases}.$$

Для построения выходного кода требуется $\left\lceil \log_2 f_M(w, n) \right\rceil$ битов.

При $M = 1$ этот процесс превращается в кодирование по методу Линча — Дэвиссона.

Процесс декодирования осуществляется по алгоритму, представленному в работах [18, 19].

Рассмотрим пример кодирования и декодирования двоичной последовательности из 24-и элементов '000010000011000000001001' методом нумерационного кодирования с применением иерархического подхода.

Кодирование последовательности. Имеем $p_{\max} = 2$, $n_1 = 8$, $n_2 = 3$. Дерево сумм описано на рисунке 4.

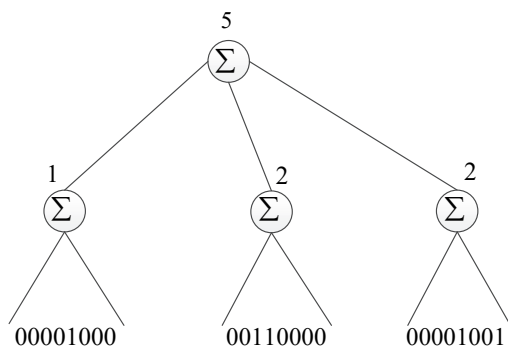


Рис. 4. Дерево сумм для примера

Процесс кодирования продолжается следующим образом:

- кодирует $w_{p_{\max,1}} = 5$ с использованием код фиксированной длины $\lceil \log_2(24+1) \rceil = 5$ бит;

- кодирует $\{w_{1,1}, w_{1,2}, w_{1,3}\}$ методом нумерационного кодирования ограниченных целых чисел. Итак, $x = \{1, 2, 2\}$ имеет $n = 3$, $w = 5$, $M = 2$. По (2): $i_S(x) = 0$ и код фиксированной длины $\lceil \log_2 f_2(5, 3) \rceil = \lceil \log_2 3 \rceil = 2$ бит;

- кодирует 00001000, 00110000, 00001001 методом Линча — Дэвиссона. По (1) для 00001000: $i_S(x) = 3$ и код фиксированной длины $\lceil \log_2 C_1^8 \rceil = 3$ бит; для 00110000: $i_S(x) = 14$ и код фиксированной длины $\lceil \log_2 C_2^8 \rceil = 5$ бит; для 00001001: $i_S(x) = 3$ и код фиксированной длины $\lceil \log_2 C_2^8 \rceil = 5$ бит.

Таким образом, общее количество бит, требуемое для хранения закодированной заданной последовательности, равно $5+2+3+5+5=20$ бит. Для хранения конечного результата требуется на 4 бита меньше.

Процесс декодирования проводится в таком же порядке. Сначала получают $w_{p_{\max,1}} = 5$; затем $\{w_{1,1}, w_{1,2}, w_{1,3}\} = \{1, 2, 2\}$ и, наконец, составляющие последовательности 00001000, 00110000, 00001001.

5. Экспериментальные результаты. Предложенный выше подход был реализован в среде разработки C++ и протестирован с видеопоследовательностями разного информационного характера и различных размерностей.

В проведенном эксперименте будем сравнивать предложенный метод со стандартным алгоритмом САВАС, используемыми в видеокодеке H.264/AVC основного профиля (с помощью Эталонного программного обеспечения JM 19.0 (H.264/1449610 AVC Reference Software)) по следующим критериям: качество кодирования (по объему выходного потока) и время кодирования. Эксперимент проводится на платформе Windows 10 с процессором Intel (R) Core i5-4210U CPU @ 1,70 ГГц 2,40 ГГц и 6 ГБ оперативной памяти.

Эффективность кодирования также оценивается путем сравнения объема выходного потока, представленного в таблице 6 и на рисунке 5. Экспериментальные результаты показали, что предложенный алгоритм уменьшает объем выходного файла в среднем на 12.42 – 14.76% по сравнению с методом САВАС.

Таблица 6. Объем выходного потока для разных типов видеопоследовательностей

Тип	Видеопоследовательность	Процент уменьшения объема предложенного метода над САВАС, %			
		QP = 24	QP = 28	QP = 32	QP = 36
с малым количеством движущихся объектов в кадре	Bus_QCIF	5.46	13.21	22.75	5.55
	Football_QCIF	15.26	13.34	16.85	16.68
	Foreman_QCIF	13.52	13.75	17.06	23.15
	Ice_QCIF	15.43	5.53	16.83	22.95
	Mobile_QCIF	5.343	15.36	22.52	22.43
	City_CIF	5.43	13.22	5.49	5.58
с их большим количеством	Bus_CIF	15.25	13.23	5.32	9.26
	Football_CIF	9.86	15.25	5.58	13.30
	Foreman_CIF	16.47	15.23	22.55	15.35
	Ice_CIF	37.88	15.36	5.39	15.56
	Soccer_CIF	9.84	13.37	5.56	15.37
полного разрешения	Crew_D1	5.27	15.35	5.57	13.31
	Ice_D1	10.04	6.20	9.96	13.41
<i>В среднем</i>		<i>12.70</i>	<i>12.95</i>	<i>12.42</i>	<i>14.76</i>

Второй показатель — время кодирования оценивается путем сравнения среднего необходимого времени для кодирования каждого

видеопотока, показанного в таблице 7. Видно, что предложенный алгоритм повышает время кодирования в среднем на 50% по сравнению с методом САВАС.

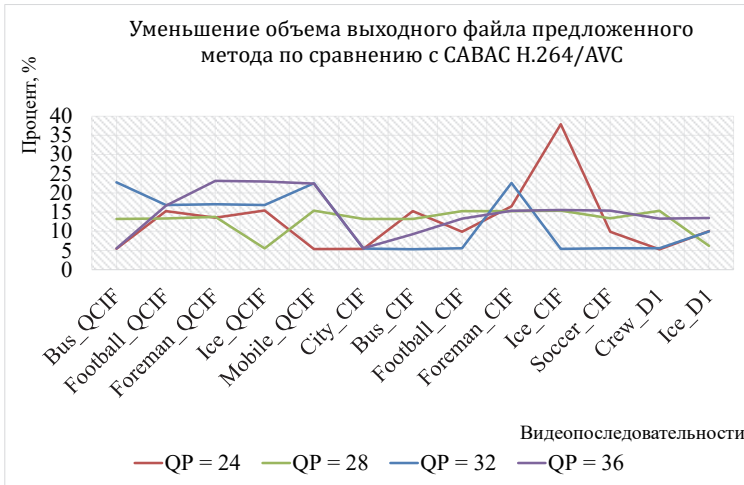


Рис. 5. Уменьшение объема выходного файла предложенного метода по сравнению с САВАС

Таблица 7. Среднее время для кодирования разных типов видеопоследовательностей

Тип	Видеопоследовательность	Степень увеличения среднего времени предложенного метода над САВАС, %			
		QP = 24	QP = 28	QP = 32	QP = 36
с малым количеством движущихся объектов в кадре	Bus_QCIF	88.88	67.98	53.06	74.53
	Football_QCIF	72.18	68.02	62.44	35.98
	Foreman_QCIF	43.72	35.42	47.53	40.73
	Ice_QCIF	60.12	84.23	52.16	51.53
	Mobile_QCIF	91.95	76.34	64.05	52.48
	City_CIF	84.38	64.02	74.07	63.53
с их большим количеством	Bus_CIF	63.36	64.18	75.67	39.77
	Football_CIF	83.57	69.55	76.68	36.62
	Foreman_CIF	33.20	28.72	31.82	43.61
	Ice_CIF	42.70	30.99	63.72	42.18
	Soccer_CIF	77.64	59.95	68.88	52.09
полного разрешения	Crew_D1	78.68	49.64	46.16	22.73
	Ice_D1	47.41	55.09	37.44	28.10
<i>В среднем</i>		<i>66.75</i>	<i>58.01</i>	<i>57.98</i>	<i>44.91</i>

Таким образом, можно утверждать, что предложенный метод обеспечивает больший коэффициент сжатия (в среднем до 15%), но при этом требуется большее время на выполнение по сравнению с другими алгоритмами (примерно на 50 %).

6. Заключение. В данной работе предложен модифицированный подход к построению энтропийного кодека с параллельным выполнением операций для повышения эффективности сжатия мультимедийных данных. Экспериментальные результаты показали что предложенный метод обеспечивает увеличение от 15 до 30% степени сжатия по сравнению с традиционным методом, но при этом увеличивается время выполнения всей процедуры кодирования. Таким образом, предлагаемый подход может быть рекомендован прежде всего для приложений, критичных к объему выходного файла и мало критичным к скорости обработки видеопотока, например, если не требуется обеспечить реальный масштаб времени.

Литература

1. *Ричардсон Я.* Видеокодирование. H.264 и MPEG-4 – стандарты нового поколения // М.: Техносфера. 2005. 368 с.
2. *Zhang H. et al.* Evaluation of beyond-HEVC entropy coding methods for DCT transform coefficients // 2016 Visual Communications and Image Processing (VCIP). 2016. pp. 1–4.
3. *Huffman D.A.* A method for the construction of minimum-redundancy codes // Proc. IRE. 1952. vol. 40. no. 9. pp. 1098–1101.
4. *Rissanen J., Langdon G.G.* Arithmetic Coding // IBM Journal of Research and Development. 1979. vol. 23. no. 2. pp. 149–162.
5. *Acharyya A., Kothari S., Reeve J.* A new CAVLC algorithm for higher bit compression by introducing the concept of Position Coding of the coefficients in H.264/AVC // 2012 IEEE International Conference on Industrial Technology. 2012. pp. 123–128.
6. *Nargundmath S., Nandibewor A.* Entropy coding of H.264/AVC using Exp-Golomb coding and CAVLC coding // International Conference on Advanced Nanomaterials & Emerging Engineering Technologies. 2013. pp. 607–612.
7. *Rapaka K., Yang En-Hui.* A High Throughput Multi Symbol CABAC Framework for Hybrid Video Codescs // 2013 Data Compression Conference. 2013. pp. 515–515.
8. *Marpe D., Schwarz H., Wiegand T.* Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard // IEEE Trans. Circuits Syst. Video Technol. 2003. vol. 13. no. 7. pp. 620–636.
9. *Wiegand T.* Overview of the H. 264/AVC video coding standard // IEEE Transactions on Circuits and Systems for Video Technology. 2003. vol. 13. no 7. pp. 560–576.
10. *Sullivan G.J. et al.* Overview of the high efficiency video coding (HEVC) standard // IEEE Transactions on Circuits and Systems for Video Technology. 2012. vol. 22. no. 12. pp. 1649–1668.
11. ITU-T. ITU-T Recommendation H.265 High Efficiency Video Coding // ITU Telecommunication Standardization Sector (ITU-T). 2015. 609 p.
12. *Belyaev E., Turlikov A., Egiazarian K.* An Efficient Adaptive Binary Arithmetic Coder With Low Memory Requirement // IEEE Journal of Selected Topics in Signal Processing. 2013. vol. 7. no. 6. pp. 1053–1061.
13. *Belyaev E. et al.* An Efficient Adaptive Binary Range Coder and Its VLSI Architecture

14. // IEEE Trans. Circuits Syst. Video Technol. 2015. vol. 25. no. 8. pp. 1435–1446.
14. *Auli-Llinàs F.* Context-Adaptive Binary Arithmetic Coding With Fixed-Length Codewords // IEEE Trans. Multimed. 2015. vol. 17. no. 8. pp. 1385–1390.
15. *Heo J., Ho Y.-S.* VLC table prediction for CAVLC in H.264/AVC using correlation, statistics, and structural characteristics of mode information // Telecommun. Syst. 2013. vol. 52, no 3. pp. 1633–1641.
16. *Heo J., Oh K.-J., Ho Y.-S.* An Efficient Table Prediction Scheme for CAVLC in H.264. URL:<https://pdfs.semanticscholar.org/a383/2912d3fbb463b4dde1d286c7f138226171ed.pdf> (accessed: 15.03.2017).
17. *Neji N. et al.* FPGA implementation of improved binarizer design for context-based adaptive binary arithmetic coder // 2016 International Image Processing, Applications and Systems (IPAS). 2016. pp. 1–4.
18. *Нгуен В.Ч., Тропченко А.А.* Повышение эффективности сжатия данных с помощью иерархического перечислительного кодирования // Известия высших учебных заведений. Приборостроение. 2016. Т. 59. № 12. С. 991–996.
19. *Нгуен В.Ч.* Анализ и разработка методов и алгоритмов энтропийного кодирования. Deutschland: LAP LAMBERT Academic Publishing. 2017. 52 с.
20. *Lynch T.J.* Sequence time coding for data compression // Proc. IEEE. 1966. vol. 54. no. 10. pp. 1490–1491.
21. *Davisson L.D.* Comments on “Sequence Time Coding for Data Compression” // Proceedings of the IEEE. 1966. vol. 54. no. 12. pp. 2010–2011.

Нгуен Ван Чыонг — аспирант кафедры вычислительной техники факультета программной инженерии и компьютерной техники, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО). Область научных интересов: алгоритмы обработки видео и мультимедийных данных, сжатие мультимедийных данных. Число научных публикаций — 15. vcnguen@corp.ifmo.ru; Кронверкский пр., 49, 197101, Санкт-Петербург; р.т.: +79632492202.

Тропченко Андрей Александрович — к-т техн. наук, доцент кафедры вычислительной техники факультета программной инженерии и компьютерной техники, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО). Область научных интересов: алгоритмы обработки аудио, видео и мультимедийных данных, биометрические системы идентификации личности человека, сжатие мультимедийных данных. Число научных публикаций — 41. aatropchenko@corp.ifmo.ru, http://cse.ifmo.ru/ru/viewperson/111848/tropchenko_andrey_aleksandrovich.htm; Кронверкский пр., 49, 197101, Санкт-Петербург; р.т.: +7(812)232-5278.

Поддержка исследований. Работа выполнена при государственной поддержке ведущих университетов Российской Федерации (субсидия 08-08)

V.T. NGUYEN, A.A. TROPCHENKO
**IMPROVING EFFICIENCY OF ENTROPY CODING METHOD IN
MODERN VIDEO COMPRESSION STANDARDS**

Nguyen V.T., Tropchenko A.A. Improving Efficiency of Entropy Coding Method in Modern Video Compression Standards.

Abstract. Modern video coding standards have high coding efficiency, but the encoding performance has to be improved to meet the growing multimedia applications. The paper deals with the entropy encoding methods and algorithms in video coding standard H.264/AVC and H.265/HEVC. Context-based Adaptive Variable Length Coding (CAVLC) for the H.264/AVC standard was originally designed for lossy video coding, and as such does not yield adequate performance for lossless video coding. Context-Adaptive Binary Arithmetic Coding (CABAC) is a method of entropy coding first introduced in H.264/AVC and now used in the standard H.265/HEVC. While it provides high coding efficiency, the data dependencies in H.264/AVC CABAC make it challenging to parallelize and thus, limit its throughput. Accordingly, during the standardization of entropy coding for HEVC, both coding efficiency and throughput were considered. Based on an analysis of their advantages and disadvantages, a method called the entropy coding algorithm using the enumerative coding of the hierarchical approach is proposed. The proposed algorithm consists of the Context-Adaptive Binary Arithmetic Coding algorithm and the enumerative coding algorithm with a hierarchical approach. The proposed algorithm is tested in the Visual C++ development environment on various test video sequences. The results of the experiments showed a greater efficiency of coding of multimedia data (the proposed one reduces on average up to 15% of the storage volume compared to the traditional CABAC method), while the method requires a longer coding time (approximately twice). The proposed method can be recommended for use in telecommunication systems for storage, transmission and processing of multimedia data, where a high degree of compression is required first.

Keywords: entropy coding, method CABAC, method CAVLC, enumerative coding algorithm, H.265/HEVC, H.264/AVC.

Nguyen Van Truong — Ph.D. student of computation technologies department of software engineering and computer systems faculty, ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics). Research interests: algorithms for video and multimedia data processing, multimedia data compression. The number of publications — 15. vcnguen@corp.ifmo.ru; 49, Kronverksky pr., St.-Petersburg, 197101, Russia; office phone: +79632492202.

Tropchenko Andrey Alexandrovitch — Ph.D., associate professor of computation technologies department of software engineering and computer systems faculty, ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics). Research interests: algorithms for audio, video and multimedia data processing, Biometric systems of human personality identification, multimedia data compression. The number of publications — 41. aatropchenko@corp.ifmo.ru, http://cse.ifmo.ru/ru/viewperson/111848/tropchenko_andrey_aleksandrovich.htm; 49, Kronverksky pr., St.-Petersburg, 197101, Russia; office phone: +7(812)232-5278.

Acknowledgements. This work is supported by Government of the Russian Federation (Grant 08-08).

References

1. Richardson I.E. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons. 2004. 306 p. (Russ. ed.: Richardson Ja. *Videokodirovanie. H.264 i MPEG-4 – standarty novogo pokolenija*. M.: Tehnosfera. 2005. 368 p.).

2. Zhang H. et al. Evaluation of beyond-HEVC entropy coding methods for DCT transform coefficients. 2016 Visual Communications and Image Processing (VCIP). 2016. pp. 1–4.
3. Huffman D.A. A method for the construction of minimum-redundancy codes. Proc. IRE. 1952. vol. 40. no. 9. pp. 1098–1101.
4. Rissanen J., Langdon G.G. Arithmetic Coding. *IBM Journal of Research and Development*. 1979. vol. 23. no. 2. pp. 149–162.
5. Acharyya A., Kothari S., Reeve J. A new CAVLC algorithm for higher bit compression by introducing the concept of Position Coding of the coefficients in H.264/AVC. 2012 IEEE International Conference on Industrial Technology. 2012. pp. 123–128.
6. Nargundmath S., Nandibewoor A. Entropy coding of H.264/AVC using Exp-Golomb coding and CAVLC coding. International Conference on Advanced Nanomaterials & Emerging Engineering Technologies. 2013. pp. 607–612.
7. Rapaka K., Yang En-Hui. A High Throughput Multi Symbol CABAC Framework for Hybrid Video Codecs. 2013 Data Compression Conference. 2013. pp. 515–515.
8. Marpe D., Schwarz H., Wiegand T. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Trans. Circuits Syst. Video Technol.* 2003. vol. 13. no. 7. pp. 620–636.
9. Wiegand T. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2003. vol. 13. no. 7. pp. 560–576.
10. Sullivan G.J. et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2012. vol. 22. no. 12. pp. 1649–1668.
11. ITU-T. ITU-T Recommendation H.265 High Efficiency Video Coding. ITU Telecommunication Standardization Sector (ITU-T). 2015. 609 p.
12. Belyaev E., Turlikov A., Egiazarian K. An Efficient Adaptive Binary Arithmetic Coder With Low Memory Requirement. *IEEE Journal of Selected Topics in Signal Processing*. 2013. vol. 7. no. 6. pp. 1053–1061.
13. Belyaev E. et al. An Efficient Adaptive Binary Range Coder and Its VLSI Architecture. *IEEE Trans. Circuits Syst. Video Technol.* 2015. vol. 25. no. 8. pp. 1435–1446.
14. Aulí-Llinàs F. Context-Adaptive Binary Arithmetic Coding With Fixed-Length Codewords. *IEEE Trans. Multimed.* 2015. vol. 17. no. 8. pp. 1385–1390.
15. Heo J., Ho Y.-S. VLC table prediction for CAVLC in H.264/AVC using correlation, statistics, and structural characteristics of mode information. *Telecommun. Syst.* 2013. vol. 52. no. 3. pp. 1633–1641.
16. Heo J., Oh K.-J., Ho Y.-S. An Efficient Table Prediction Scheme for CAVLC in H.264. Available at : <https://pdfs.semanticscholar.org/a383/2912d3fbb463b4dde1d286c7f138226171ed.pdf> (accessed: 15.03.2017).
17. Neji N. et al. FPGA implementation of improved binarizer design for context-based adaptive binary arithmetic coder. 2016 International Image Processing, Applications and Systems (IPAS). 2016. pp. 1–4.
18. Nguyen V. T., [Improving the efficiency of data compression using a hierarchically enumerative coding]. *Izv. vuzov. Priborostroenie – Journal of Instrument Engineering*. 2016. vol. 59. no. 12. pp. 991–996. (In Russ.).
19. Nguyen V.T. *Analiz i razrabotka metodov i algoritmov entropinogo kodirovaniya* [Analysis and development of entropy coding methods and algorithms. Deutschland: LAP LAMBERT Academic Publishing. 2017. 52 p. (In Russ.).
20. Lynch T.J. Sequence time coding for data compression. Proc. IEEE. 1966. vol. 54. no. 10. pp. 1490–1491.
21. Davisson L.D. Comments on “Sequence Time Coding for Data Compression”. Proceedings of the IEEE. 1966. vol. 54. no. 12. pp. 2010–2011.