

---

УДК 004.627:004.272.26

**С. Ю. Скрупский**, аспирант  
Запорожский национальный технический университет  
(Украина, 69063, Запорожье, ул. Жуковского, 64,  
тел. (061) 7698249, E-mail: 88sts88@mail.ru)

## **Повышение эффективности сжатия видеоинформации в распределенных компьютерных системах**

Проанализирован метод разделения на равные части видеоинформации в распределенных системах и определены его основные недостатки. Предложен метод, основанный на корреляционном анализе видеопоследовательности с динамической чувствительностью анализатора, не имеющий таких недостатков, что позволяет уменьшить битрейт и регулировать нагрузку на вычислительные узлы распределенной системы.

Проаналізовано метод розділення на рівні частини відеоінформації в розподілених системах та визначено його основні недоліки. Запропоновано метод, базований на кореляційному аналізі відеопослідовності з динамічною чутливістю аналізатора, який не має таких недоліків, що дозволяє знизити бітрейт та регулювати навантаження на обчислювальні вузли розподіленої системи.

*Ключевые слова:* видеоинформация, корреляция, компрессия, распределенная система, битрейт.

В современной системе телевидения передача видеосигнала на дальние расстояния (от вещательной станции до приемника конечного пользователя) осуществляется в сжатом цифровом виде. Стандарт телевидения высокой четкости HDTV (High-Definition Television) [1] обеспечивает более высокое качество (HD) представления видеоинформации по сравнению с существующими стандартами. Основной недостаток этого стандарта состоит в том, что несжатая видеопоследовательность занимает значительный объем памяти, а требования к производительности оборудования, осуществляющего кодирование (сжатие) и декодирование видеоинформации, высоки.

Для сжатия видеоинформации применяют современные методы компрессии, основанные на устранении интракадровой и интеркадровой избыточности видеоинформации. Одной из основных характеристик кодера является битрейт — число бит, формируемых в единицу времени, для кодирования видеопоследовательности. Чем больше битрейт закодиро-

ванной видеопоследовательности, тем меньше ее искажения относительно исходной, но тем больший объем памяти она занимает. Битрейт косвенно выражает степень сжатия видеопоследовательности при фиксированном уровне ее искажения.

Высокая вычислительная сложность современных методов компрессии видеоинформации не позволяет сжимать видеоинформацию стандарта HDTV в реальном времени на однопроцессорных компьютерах (даже с четырьмя логическими ядрами). Это обуславливает актуальность разработки новых методов и средств компрессии видеоинформации, в частности, с использованием параллельных и распределенных компьютерных систем.

**Постановка задачи.** Существующие способы компрессии видеоинформации в распределенных компьютерных системах основаны на предварительном разделении видеопоследовательностей на равные части с последующим сжатием их на вычислительных узлах [2 — 5]. Метод деления видеопоследовательностей на равные части имеет следующие недостатки: возрастание битрейта сжатых видеопоследовательностей и отсутствие возможности регулирования нагрузки на вычислительные узлы без увеличения битрейта сжатых видеопоследовательностей. Проанализируем указанные недостатки и рассмотрим метод деления на части видеоинформации в распределенных системах, основанный на корреляционном анализе видеопоследовательности с динамической чувствительностью анализатора, который позволяет снизить битрейт сжатых видеопоследовательностей и регулировать нагрузку на вычислительные узлы без увеличения битрейта.

**Цветовые модели представления видеоинформации.** Видеоинформация представляет собой массив байт на носителе, описывающий цветные компоненты видеопоследовательности. Представим видеопоследовательность  $VS$ , состоящую из  $N$  кадров с разрешением  $WH$ , в следующем виде:  $VS = \{F^1, F^2, \dots, F^N\}$ , где  $F^i$  — кадр видеопоследовательности;  $i$  — номер кадра,  $i = 1, 2, \dots, N$ ;  $N$  — число кадров в видеопоследовательности.

Каждый кадр видеопоследовательности представлен в виде массива чисел, описывающих либо цветные компоненты, либо яркостную и цветоразностные компоненты пикселей. В телевидении для представления видеопоследовательностей в цифровом виде наиболее часто применяются три цветовые модели:

1.  $RGB$  — число байт на красную  $R$ , зеленую  $G$  и синюю  $B$  компоненты цвета одинаково. В этом случае объем памяти, занимаемый видеоинформацией, расходуется нерационально. Видеоинформация высокой четкости в такой цветовой модели занимает излишние объемы памяти и обязательно должна быть сжата.

2.  $YUV 4:2:2$  — на каждые два байта  $Y$ -компоненты приходится один байт  $U$ -компоненты и один байт  $V$ -компоненты, т.е.  $U$ - и  $V$ -компоненты

содержат в два раза меньше информации по горизонтали. Такая цветовая модель основана на том факте, что человеческая зрительная система более чувствительна к яркости, чем к цветности.

3.  $YUV\ 4:2:0$  — на каждые четыре байта  $Y$ -компоненты приходится один байт  $U$ -компоненты и один байт  $V$ -компоненты, т.е.  $U$ - и  $V$ -компоненты содержат в два раза меньше информации по горизонтали и в два раза меньше информации по вертикали — четырехкратная экономия битрейта по  $U$ - и  $V$ -компонентам.

Рассмотрим метод разделения видеоинформации в распределенной системе с использованием этих цветовых моделей представления видеоинформации.

**Анализ метода разделения видеоинформации на равные части.**

Метод разделения видеоинформации на равные части можно представить в следующем виде:

$$VS = \{VS_1, VS_2, \dots, VS_j, \dots, VS_K\},$$

$$VS_j = \{F^{i+1}, F^{i+2}, \dots, F^{i+\frac{N}{K}}\}, \quad (1)$$

где  $VS$  — исходная видеопоследовательность;  $j$  — номер ее части;  $K$  — число частей, на которые разделяется видеопоследовательность (обычно  $K$  выбирают пропорциональным числом вычислительных узлов);  $N$  — число кадров в исходной видеопоследовательности;  $i$  — номер группы кадров,  $i=0, N/K, \dots, N-N/K$ .

Недостатки этого метода следующие.

1. При разделении видеопоследовательности на равные части не учитывается ее корреляционная структура, что приводит к нерациональной расстановке опорных кадров и, как следствие, к росту битрейта. Поясним это на примере.

Пусть в исходной видеопоследовательности  $VS = \{F^1, F^2, \dots, F^{100}\}$ , состоящей из 100 кадров, существует значительная корреляционная зависимость между кадрами  $F^1 \dots F^{40}$  и между кадрами  $F^{41} \dots F^{100}$ , т.е. в видеопоследовательности сцены меняются редко. При разделении  $VS$  по формуле (1) на 10 частей по 10 кадров каждая какой-либо вычислительный узел получает видеопоследовательность  $VS_1 = \{F^1, F^2, \dots, F^{10}\}$ . Первому кадру группы —  $F^1$  присваивается  $I$ -тип, который сжимается интракадровым методом. Остальные девять кадров сжимаются с учетом оценки движения (ОД). Другой вычислительный узел получает видеопоследовательность  $VS_2 = \{F^{11}, F^{12}, \dots, F^{20}\}$  и  $F^{11}$  (как первый кадр группы) сжимается интракадровым методом, остальные кадры в  $VS_2$  сжимаются интеркадровым методом.

Таким образом, в видеопоследовательности, объединяющей  $VS_1$  и  $VS_2$ , находится лишний  $I$ -кадр. Поскольку кадр такого типа сжимается только интракадровым методом, это увеличивает битрейт. По такому принципу во всей сжатой видеопоследовательности  $VS$  определяются восемь лишних  $I$ -кадров (кроме  $F^1$  и  $F^{41}$ ). Следовательно, такой метод разделения приводит к значительному увеличению битрейта.

2. Разделение видеопоследовательности на длинные равные части обуславливает потребность в большом количестве времени для их пересылки на вычислительные узлы, поскольку пропускная способность сети ограничена применяемым стандартом передачи данных в компьютерной сети. Чем больше вычислительных узлов, тем дольше они простаивают в ожидании входных данных для сжатия.

Например, в распределенной системе для сжатия видеоинформации размером в 1 Гб выделено четыре узла, объединенных средой передачи с пропускной способностью 1 Гбит/с. Разделив видеоинформацию на четыре равные части по 256 МБ, отправим их на вычислительные узлы. Первая часть доставляется первому узлу за 2 с (остальные узлы в это время простаивают), вторая часть доставляется второму узлу через 4 с, третья — через 6 с, а четвертая — через 8 с.

Таким образом, время простоя вычислительных узлов в ожидании доставки видеопоследовательности для сжатия пропорционально размеру частей, полученных после разделения видеопоследовательности, числу вычислительных узлов и обратно пропорционально пропускной способности сети. Вместе с тем, разделение видеопоследовательности на короткие равные части приводит к возрастанию битрейта по причине вынужденной расстановки лишних  $I$ -кадров на вычислительных узлах распределенной системы. Следовательно, регулировать нагрузку на вычислительные узлы распределенной системы, используя метод разделения на равные части, без увеличения битрейта невозможно.

**Корреляционный анализ видеопоследовательности.** Проведем аналогию между видеопоследовательностью и временным рядом. Временной ряд — это последовательность элементов, в которой каждый элемент является реализацией некоторого протекающего во времени процесса. Элементы временного ряда располагаются последовательно, представляя последовательные реализации процесса, обычно через равные промежутки времени. Видеопоследовательность — это последовательность кадров, в которой каждый кадр представляет собой реализацию отображаемого в ней процесса. Кадры видеопоследовательности также располагаются последовательно, представляя последовательные реализации процесса.

Таким образом, видеопоследовательность является частным случаем временного ряда. Поскольку пиксели кадров видеопоследовательности —

случайные величины (в любой момент времени невозможно достоверно предсказать их яркость и цвет), к видеопоследовательности целесообразно применить методы корреляционного анализа для выявления статистической зависимости между пикселями соседних кадров и межкадровой разницы.

Образуем из видеопоследовательности  $VS$  пары смежных кадров:  $(F^1, F^2), (F^2, F^3), \dots, (F^k, F^{k+1}), \dots, (F^{N-1}, F^N)$ . Первый элемент каждой пары можно рассматривать как реализацию случайной величины, а второй — как реализацию случайной величины с лагом, равным единице, относительно первого элемента, т.е. второй элемент каждой пары — это реализация случайной величины, отстающая на один кадр в видеопоследовательности относительно первого элемента.

Воспользуемся выборочной автокорреляционной функцией [6] временного ряда  $x_1, x_2, \dots, x_N$  для оценки коэффициента корреляции его смежных значений:

$$r_1 = \frac{\sum_{t=1}^{N-1} (x_t - \bar{x})(x_{t+1} - \bar{x})}{\sqrt{\sum_{t=1}^{N-1} (x_t - \bar{x})^2 (x_{t+1} - \bar{x})^2}}, \quad (2)$$

где  $r_1$  — коэффициент корреляции для смежных значений временного ряда;  $x_t, x_{t+1}$  — смежные значения временного ряда;  $\bar{x}$  — среднее значение  $x_t$ ,

$$\bar{x} = \frac{\sum_{t=1}^{N-1} x_t}{N-1}.$$

Рассмотрим цветовую модель  $YUV$  4:2:2 и найдем коэффициенты корреляции для всех пар смежных кадров видеопоследовательности, учитывая яркостную и цветоразностные компоненты пикселей. Применительно к яркостной компоненте  $Y$  для видеопоследовательности  $VS$  формула (2) примет вид

$$r^{Y,Y+1} = \frac{\sum_{i=1}^W \sum_{j=1}^H (f_{ij}^Y - \overline{f^Y})(f_{ij}^{Y+1} - \overline{f^{Y+1}})}{\sqrt{\sum_{i=1}^W \sum_{j=1}^H (f_{ij}^Y - \overline{f^Y})^2 (f_{ij}^{Y+1} - \overline{f^{Y+1}})^2}},$$

где  $r^{Y,Y+1}$  — коэффициент корреляции по яркости для смежных кадров видеопоследовательности;  $W$  и  $H$  — ширина и высота кадра;  $i, j$  — индексы

сы пиксела в кадре;  $f_{ij}^Y, f_{ij}^{Y+1}$  — яркости пикселей смежных кадров в соответствующих позициях;  $\overline{f^Y}, \overline{f^{Y+1}}$  — средние значения яркости пикселей по смежным кадрам,

$$\overline{f^Y} = \frac{\sum_{i=1}^W \sum_{j=1}^H f_{ij}^Y}{WH}, \quad \overline{f^{Y+1}} = \frac{\sum_{i=1}^W \sum_{j=1}^H f_{ij}^{Y+1}}{WH}.$$

Коэффициент корреляции по  $U$ -компоненте для пикселей смежных кадров видеопоследовательности найдем по формуле (2) с учетом двукратного уменьшения количества информации по горизонтали (в соответствии с цветовой моделью  $YUV$  4:2:2):

$$r^{U,U+1} = \frac{\sum_{i=1}^{W/2} \sum_{j=1}^H (f_{ij}^U - \overline{f^U})(f_{ij}^{U+1} - \overline{f^{U+1}})}{\sqrt{\sum_{i=1}^{W/2} \sum_{j=1}^H (f_{ij}^U - \overline{f^U})^2 (f_{ij}^{U+1} - \overline{f^{U+1}})^2}},$$

где  $r^{U,U+1}$  — коэффициент корреляции по  $U$ -компоненте пикселей для смежных кадров видеопоследовательности;  $f_{ij}^U, f_{ij}^{U+1}$  — значения  $U$ -компоненты пикселей смежных кадров в соответствующих позициях;  $\overline{f^U}, \overline{f^{U+1}}$  — средние значения  $U$ -компоненты по смежным кадрам,

$$\overline{f^U} = \frac{\sum_{i=1}^{W/2} \sum_{j=1}^H f_{ij}^U}{W/2 \cdot H}, \quad \overline{f^{U+1}} = \frac{\sum_{i=1}^{W/2} \sum_{j=1}^H f_{ij}^{U+1}}{W/2 \cdot H}.$$

Аналогично определим коэффициент корреляции по  $V$ -компоненте пикселей для смежных кадров видеопоследовательности. Для цветовой модели  $RGB$  коэффициенты корреляции  $r^{R,R+1}, r^{G,G+1}, r^{B,B+1}$  можно найти по формуле (2) с учетом того, что матрицы, описывающие компоненты цвета, одинакового размера.

Результирующий коэффициент корреляции для пары смежных кадров видеопоследовательности можно определить как среднее значение полученных коэффициентов корреляции для всех компонент цветовой модели с учетом их весовых коэффициентов. Так, для цветовой модели  $YUV$  4:2:2 результирующий коэффициент корреляции имеет вид

$$r_{YUV}^{k,k+1} = \frac{2r^{Y,Y+1} + r^{U,U+1} + r^{V,V+1}}{4}.$$

Для цветовой модели  $RGB$  результирующий коэффициент корреляции определим в виде

$$r_{RGB}^{k,k+1} = \frac{r^{R,R+1} + r^{G,G+1} + r^{B,B+1}}{3}.$$

Найдем коэффициенты корреляции для всех пар смежных кадров видеопоследовательности (получим  $N-1$  значений)  $\{r_{YUV}^{1,2}, r_{YUV}^{2,3}, \dots, r_{YUV}^{k,k+1}, \dots, r_{YUV}^{N-1,N}\}$ , по которым построим коррелограмму. Аналогичные действия можно выполнить для цветовой модели  $RGB$ .

**Функция чувствительности к смене сцен анализатора видеопоследовательности.** Для анализа полученной коррелограммы введем функцию чувствительности анализатора видеопоследовательности к смене сцен  $\varphi(k)$ . В местах, где корреляция между соседними кадрами видеопоследовательности определена как слабая в соответствии с чувствительностью анализатора, т.е.  $|r_{YUV}^{k,k+1}| \leq \varphi(k)$ , разделим видеопоследовательность на части: первая часть — видеопоследовательность, содержащая кадры исходной видеопоследовательности до места слабой межкадровой корреляции, а вторая — видеопоследовательность с кадрами после места слабой межкадровой корреляции. После каждого разделения видеопоследовательности функция  $\varphi(k)$  принимает первоначальное значение,  $\varphi(1)$  и анализ оставшейся части видеопоследовательности продолжается с ее «нового» первого кадра.

Функция  $\varphi(k)$  должна монотонно возрастать с увеличением номера текущего кадра ( $k=1,2,\dots,\infty$ ), т.е. чувствительность анализатора видеопоследовательности должна повышаться. Это позволит избежать ситуации, при которой корреляция между кадрами сохраняется на стабильно высоком уровне слишком долго. Следовательно, места разделения видеопоследовательности будут находиться на очень большом расстоянии одно от другого, что приведет к созданию больших частей видеопоследовательности и к неравномерной нагрузке на вычислительные узлы. Поскольку  $0 \leq |r_{YUV}^{k,k+1}| \leq 1$ , для анализа коррелограммы необходима функция с областью значений  $\varphi(k) \in (0; 1)$ .

Исходя из перечисленных требований, введем следующую функцию чувствительности:

$$\varphi(k) = 1 - \frac{1}{\alpha k + 1}, \quad (3)$$

где  $k$  — номер текущего кадра;  $\alpha$  — параметр, с помощью которого можно управлять скоростью возрастания функции чувствительности. Например,



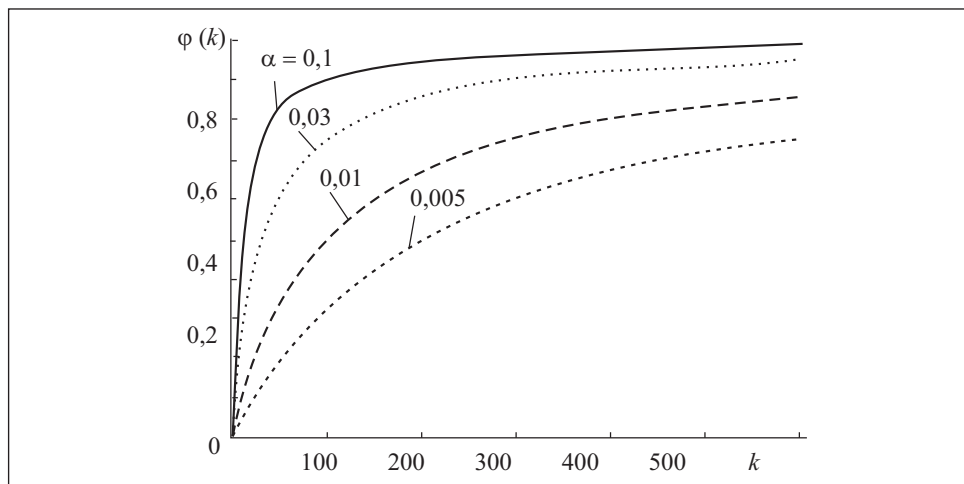


Рис. 1

при  $\alpha = 0,005$  чувствительность анализатора низкая, при  $\alpha = 0,01$  — средняя, при  $\alpha = 0,03$  — высокая, а при  $\alpha = 0,1$  — очень высокая. Такие выводы делаем, исходя из номера кадра, при достижении которого значение функции чувствительности станет более 0,8, что свидетельствует о значительной корреляционной зависимости между элементами временного ряда. Семейство кривых  $\varphi(k)$  в зависимости от параметра  $\alpha$  показано на рис. 1.

Предложенная функция чувствительности  $\varphi(k)$  анализатора видеопоследовательности к смене сцен имеет область определения  $k = 1, 2, \dots, \infty$  (исходя из физического смысла функции), область значений  $\varphi(k) \in (0; 1)$  и является монотонно возрастающей на всей области определения.

Таким образом, в местах, где анализатор, в соответствии с его текущей чувствительностью, определил корреляцию между соседними кадрами как слабую, будем разделять видеопоследовательность на части для отправки на вычислительные узлы. При этом могут возникнуть три случая.

1. В видеопоследовательности произошла смена сцены, и первый кадр новой сцены закодирован как  $I$ -кадр. В этом случае разделение видеопоследовательности на сцены вообще не приведет к возрастанию битрейта по сравнению с последовательным сжатием.

2. В видеопоследовательности длительное время, на протяжении которого чувствительность анализатора повышалась, существенных изменений не происходило. Тогда предложенный метод не обеспечит выигрыша в битрейте по сравнению с методом разделения видеoinформации на равные части, уступив при этом по вычислительной сложности.



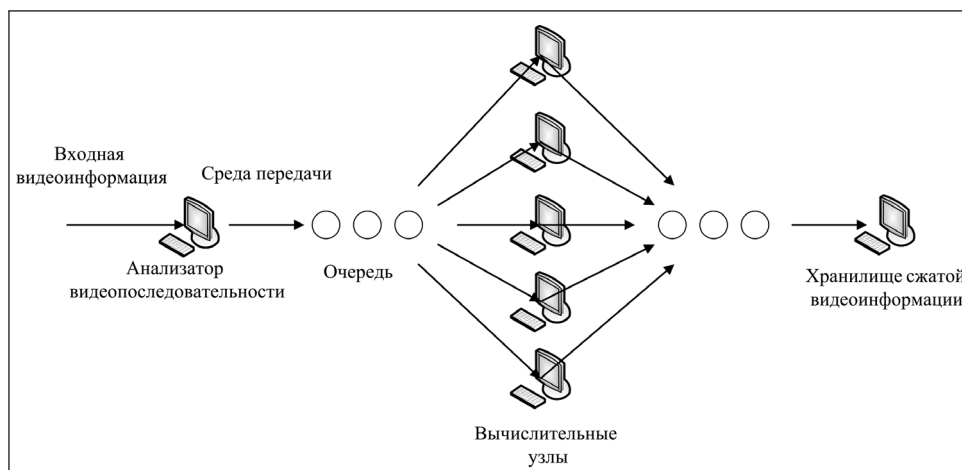


Рис. 2

3. Анализатор определил корреляцию между двумя соседними кадрами в видеопоследовательности как слабую, но смены сцены не произошло. Слабая корреляция свидетельствует о значительной межкадровой разнице, сжатие которой интеркадровым методом не обеспечит большого выигрыша в битрейте по сравнению с интракадровым. Тогда разделение видеопоследовательности в этом месте не приведет к значительному увеличению битрейта по сравнению с последовательным сжатием.

**Экспериментальная проверка метода разделения видеоинформации в распределенных системах.** В распределенную компьютерную систему поступала видеоинформация для сжатия (рис. 2). Анализатор принимал ее и выполнял разделение видеопоследовательности на части описанным методом. Получаемые в результате разделения части видеопоследовательности по мере появления становились в очередь на сжатие вычислительными узлами. Если какой-либо вычислительный узел был свободен, он изымал видеопоследовательность из очереди и сжимал ее. Сжатая часть видеоинформации поступала в хранилище. Если все ресурсы были заняты, то видеопоследовательность ожидала в очереди, пока не освободится хотя бы один процессор. Такая организация вычислительного процесса позволила избежать накладных расходов на синхронизацию процессоров и сжимать части видеопоследовательностей по мере их появления.

Для проведения эксперимента использован пакет MATLAB и свободный для распространения консольный видеокомпрессор с открытым кодом на базе программы FFmpeg [7], который поддерживает стандарты MPEG-2 и MPEG-4/H.264. Для тестов использованы видеопоследовательности, полученные конкатенацией файлов, содержащих общепринятые

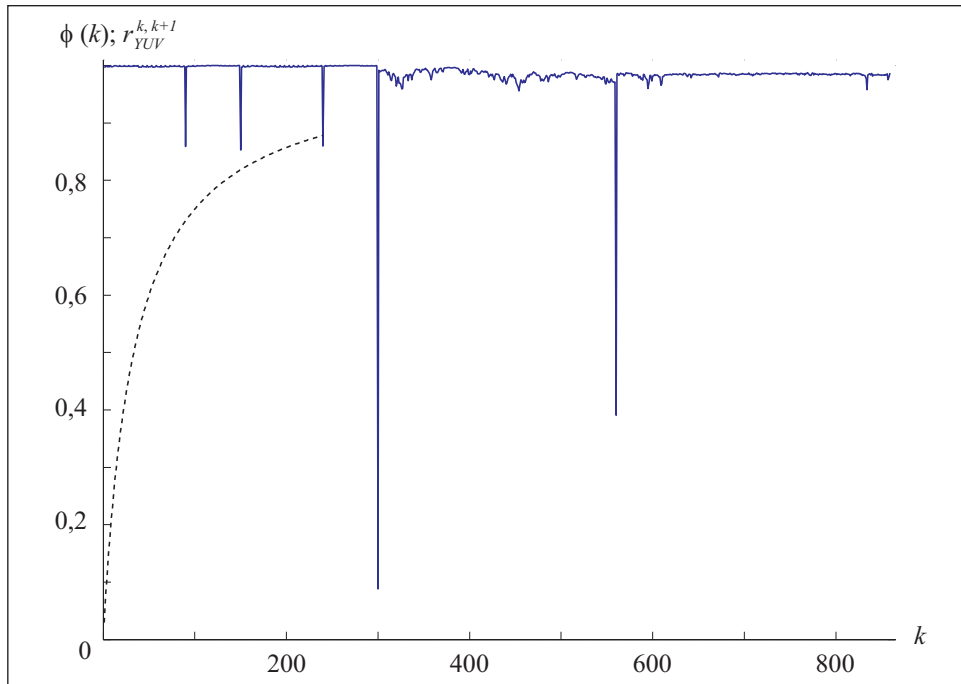


Рис. 3

для тестирования методов сжатия видеопоследовательности, охватывающие все доступные типы цифровой видеоинформации [8, 9]. Конкатенация обеспечивала дополнительные смены сцен, так как тестовые видеопоследовательности записаны в одной сцене и не позволяют в полной мере протестировать предложенный метод разделения.

Программа, реализованная в MATLAB и установленная на узле-анализаторе, считывала в память тестовую видеопоследовательность, вычисляла коэффициенты корреляции для смежных кадров видеопоследовательности, анализировала видеопоследовательность в соответствии с функцией чувствительности ( $\alpha = 0,03$ ) и разделяла видеопоследовательность на части (рис. 3).

Для сравнения предложенного метода с методом разделения видеопоследовательности на равные части, программа на MATLAB разделяла тестовые видеопоследовательности на равные части, число которых, совпадало с числом частей, полученных после применения предложенного метода. Так были обеспечены равные условия для испытания двух методов.

Части видеопоследовательностей, получаемые после применения методов разделения, поступали в видео кодер FFmpeg, который выполнял их

сжатие по стандартам MPEG-2 и MPEG-4/H.264 на вычислительных узлах распределенной системы. Для MPEG-2 настройки кодера следующие:

число *B*-кадров в группе: 3;

число восстановленных кадров для ОД: 2;

целопиксельная точность для ОД полным перебором с блоками размером 16×16 пикселей и областью поиска 32 пикселя;

битрейт переменный с сохранением исходного качества.

Для MPEG-4/H.264 настройки кодера следующие:

число *B*-кадров в группе: 3;

число восстановленных кадров для ОД: 5;

четвертьпиксельная точность для ОД полным перебором с блоками переменного размера и областью поиска 128 пикселей;

битрейт переменный с сохранением исходного качества.

После сжатия соответствующие части видеопоследовательностей объединялись конкатенацией файлов в хранилище (без дополнительных расчетов). Для сравнения битрейта сжатых видеопоследовательностей при последовательной реализации компрессии и реализации в распределенной системе тестовые видеопоследовательности сжимались на одном процессоре без предварительного разделения на части.

В результате эксперимента получены такие видеопоследовательности:

сжатая по обоим стандартам после разделения предложенным методом;

сжатая по обоим стандартам после применения метода разделения на равные части;

сжатая по обоим стандартам без разделения на части.

Затем сравнивались битрейты и уровни искажения между соответствующими видеопоследовательностями. Результаты эксперимента представлены в табл. 1. Величина  $\Delta$  bitrate 1 характеризует уменьшение битрейта видеопоследовательности, разделенной предложенным методом и сжатой в распределенной системе по стандартам MPEG-2 и MPEG-4/H.264, по сравнению с битрейтом видеопоследовательности, полученной в результате применения метода разделения на равные части и сжатой в распределенной системе по соответствующему стандарту, при одинаковом уровне искажения этих видеопоследовательностей, а величина  $\Delta$  bitrate 2 — уменьшение битрейта видеопоследовательности, сжатой на одном процессоре (без разделения на части) по стандартам MPEG-2 и MPEG-4/H.264, по сравнению с битрейтом видеопоследовательности, полученной в результате применения метода разделения на равные части и сжатой в распределенной системе по соответствующему стандарту при одинаковом уровне искажения этих видеопоследовательностей.

Для каждой сжатой видеопоследовательности вычислена величина  $AFS$ , равная среднему числу кадров в сценах видеопоследовательности:  $AFS = N / S$ , где  $N$  — число кадров в видеопоследовательности;  $S$  — число сцен, определенных анализатором в соответствии с функцией чувствительности. С учетом того, что частота смены кадров в видеопоследовательности фиксирована, величину  $AFS$  можно использовать для определения частоты смены сцен в видеопоследовательности.

Результаты экспериментальной проверки предложенного метода свидетельствуют об уменьшении битрейта сжатых видеопоследовательностей от 0,42 до 2,45 % для стандарта MPEG-2 и от 1,18 до 6,4 % для стандарта MPEG-4/H.264 по сравнению с методом разделения видеопоследовательности на равные части. Таким образом, предложенный метод позволяет уменьшить битрейт видеопоследовательностей, сжимаемых в распределенной системе. Основными характеристиками, влияющими на уменьшение битрейта, являются величина корреляционной зависимости между

Таблица 1

Тестовая видеопоследовательность	Разрешение кадра	Число кадров	$AFS$	MPEG-2		MPEG-4	
				$\Delta$ bitrate 1, %	$\Delta$ bitrate 2, %	$\Delta$ bitrate 1, %	$\Delta$ bitrate 2, %
Akiyo Carphone	176×144	1176	392	0,42	0,42	4,16	4,16
Claire							
Foreman	352×288	300	150	1,15	1,18	1,18	1,23
Coastguard	352×288	860	286,7	1,75	1,75	1,87	1,87
Container							
Waterfall	352×288	750	250	1,06	1,06	2	2
Silent Mother	352×288	1000	333,3	0,94	0,94	1,9	1,9
and Daughter							
Bus	352×288	860	215	2,45	2,47	6,4	6,44
Mobile	1920×1080	510	255	2,1	2,1	3,2	3,2
Bridge Paris							
News Tempete	1920×1080	520	260	2,08	2,08	3,08	3,08
Hall monitor	1920×1080	480	240	2,11	2,11	3,31	3,31
Blue sky	1920×1080	480	240	2,11	2,11	3,31	3,31
Pedestrian area							
Riverbed	1920×1080	520	260	2,08	2,08	3,08	3,08
Station	1920×1080	480	240	2,11	2,11	3,31	3,31
Rush hour	1920×1080	480	240	2,11	2,11	3,31	3,31
Tractor							

кадрами видеопоследовательности, среднее число кадров в сцене, разрешение кадра и применяемый стандарт сжатия. На видеопоследовательностях «Foreman» и «News Tempete Hall monitor» анализатор определил корреляцию между соседними кадрами как слабую внутри сцены (случай 3, указанный выше), вследствие чего однопроцессорная реализация компрессии видеопоследовательности оказалась эффективнее по битрейту, чем реализация в распределенной системе.

Для проверки эффективности предложенного метода на видеопоследовательностях с частой сменой сцен, которая характерна для рекламных роликов, спортивных передач и блоков новостей, было сокращено среднее число кадров в сценах посредством удаления равного числа последних кадров тестовых видеопоследовательностей. Результаты повторного эксперимента приведены в табл. 2.

Как видно из табл. 2, уменьшение битрейта при использовании предложенного метода для сжатых видеопоследовательностей составило от 0,81

Таблица 2

Тестовая видеопоследовательность	Разрешение кадра	Число кадров	AFS	MPEG-2		MPEG-4	
				Δ bitrate 1, %	Δ bitrate 2, %	Δ bitrate 1, %	Δ bitrate 2, %
Akiyo Carphone Claire	176×144	588	196	0,81	0,81	8,5	8,5
Foreman	352×288	200	100	1,51	1,57	1,9	2
Coastguard Container Waterfall	352×288	165	55	3,57	3,57	6,57	6,57
Silent Mother and Daughter Bus	352×288	155	51,7	3,14	3,14	6,19	6,19
Mobile Bridge Paris	352×288	175	58,3	3,2	3,2	6,22	6,22
News Tempete Hall monitor	352×288	220	55	5,28	5,32	8,58	8,67
Blue sky Pedestrian area	1920×1080	110	55	6,9	6,9	8,9	8,9
Riverbed Station	1920×1080	110	55	6,82	6,82	8,82	8,82
Rush hour Tractor	1920×1080	110	55	7	7	9	9

Таблица 3

Разрешение кадра тестовых видеопоследовательностей	Доля накладных затрат в вычислительном процессе (%) при сжатии по стандарту	
	MPEG-2	MPEG-4/H.264
176 × 144	67	8,33
352 × 288	40	9,52
1920 × 1080	33,96	7,83

до 7 % для стандарта MPEG-2 и от 1,9 до 9 % для стандарта MPEG-4/H.264. Лучшие результаты по сравнению с предыдущим экспериментом обусловлены сокращением числа кадров в сценах, так как уменьшение битрейта при использовании предложенного метода происходит в местах слабой межкадровой корреляции, в том числе в местах смены сцен, которые в данном эксперименте встречаются чаще.

Известно, что стандарт MPEG-4/H.264 значительно лучше сжимает видеоинформацию интеркадровым методом, чем стандарт MPEG-2 [10]. Поэтому сокращение числа лишних *I*-кадров (сжимаемых только интракадровым методом) с помощью предложенного метода для видеопоследовательностей, сжатых по стандарту MPEG-4/H.264, уменьшает битрейт значительно, чем для видеопоследовательностей, сжатых по стандарту MPEG-2.

Поскольку число свободных процессоров распределенной системы, которые можно задействовать для сжатия видеоинформации, в каждый момент времени меняется, невозможно изначально подобрать оптимальный размер блоков данных для деления исходной видеоинформации методом деления на равные части. Намеренное уменьшение размера частей видеопоследовательности приведет к вынужденному увеличению битрейта при сжатии (расстановка лишних *I*-кадров на вычислительных узлах). Предложенный метод позволяет уменьшить размер блоков данных без увеличения битрейта.

**Оценка накладных затрат при использовании предложенного метода.** Описанные эксперименты были проведены 10 раз в распределенной системе. Полученные при этом количественные оценки накладных затрат приведены в табл. 3.

Таким образом, применение предложенного метода в распределенной системе оправдано в случае сжатия видеопоследовательностей по стандарту MPEG-4/H.264. При сжатии видеопоследовательностей по стандарту MPEG-2 доля накладных затрат достаточно велика и преимущества предложенного метода в этом случае теряют практическую ценность.

## Выводы

Предложенный метод разделения видеoinформации в распределенной системе, основанный на корреляционном анализе видеопоследовательности с динамической чувствительностью анализатора, позволяет сократить битрейт и регулировать нагрузку на вычислительные узлы без увеличения битрейта.

При разделении учитывается корреляционная структура, что позволяет рационально выполнять расстановку опорных кадров на узлах распределенной системы, осуществляющих компрессию видеопоследовательности, в следствие чего битрейт сохраняется на уровне, близком к уровню, соответствующему последовательной компрессии.

Качество сжатой видеопоследовательности (по метрике PSNR) после обработки на вычислительных узлах соответствует качеству последовательной компрессии, т.е. не деградирует вследствие разделения.

Поскольку чувствительность анализатора к смене сцен видеопоследовательности увеличивается с возрастанием номера текущего кадра и управляется параметром  $\alpha$  из (3), получаемые части имеют ограниченные размеры, что позволяет регулировать нагрузку на вычислительные узлы без увеличения битрейта.

Эффективность предложенного метода зависит от следующих факторов:

- корреляционной зависимости между кадрами: чем она слабее, тем эффективнее предложенный метод;

- среднего числа кадров в сценах (чем оно меньше, тем значительнее уменьшение битрейта);

- размера кадра видеопоследовательности: чем он больше, тем значительнее уменьшение битрейта при сохранении качества. Например, уменьшение битрейта приводит к экономии объема памяти, занимаемого сжатой видеопоследовательностью, от нескольких килобайт для видеопоследовательностей с разрешением кадра  $176 \times 144$  до нескольких мегабайт для видеопоследовательностей с разрешением кадра  $1920 \times 1080$  пикселей.

При анализе видеопоследовательностей с очень редкой сменой сцен предложенный метод не обеспечивает выигрыша в битрейте, уступая по вычислительной сложности методу разделения на равные части.

Таким образом, основной сферой применения предложенного метода является киноиндустрия и телевидение в формате высокой четкости, где для сжатия видеопоследовательностей с большим разрешением кадра по стандарту MPEG-4/H.264 применяются распределенные компьютерные системы для достижения максимального качества и степени сжатия за относительно приемлемое время.



The paper deals with the split methods of video information in distributed systems. The method of video sequence split into equal parts has been analyzed. Its main disadvantages are considered. The new method of a video sequence split based on the correlation analysis with the dynamic analyzer sensitivity has been proposed. The proposed method has no such disadvantages and allows to decrease bitrate and to regulate computational nodes load.

1. *Parameter values for the HDTV standards for production and international programme exchange* : ITU-R Recommendation BT.709, 2008.
2. *Kent S., Broadbent P., Warren N., Gulliver S. On-Demand HD Video Using Jini Based Grid.* // ICME.— 2008. — P. 1045—1048.
3. *Distributed Video-on-Demand — A grid based VoD solution* / Infosys. — \www/ URL: <http://www.infosys.com/offers/industries/studios-networks/white-papers/Documents/video-on-demand.pdf> — 1.10.2010 г.
4. *Farin D., With P. Generic Framework for Parallel and Distributed Processing of Video-Data* // 4th Intern. Symp. on Parallel and Distributed Processing and Applications (ISPA). — Sorrento, Italy. — 2006. — vol. LNCS 4331. — P. 823—832.
5. *Fernández J. C., Malumbres M. P. A Parallel implementation of H.26L video encoder* // Proc. of EuroPar 2002 conf. (LNCS 2400).— Paderborn, 2002. — P. 830—833.
6. *Тюрин Ю. Н., Макаров А. А. Анализ данных на компьютере* / Под ред. В.Э. Фигурнова. 3-е изд. — М. : ИНФРА—М, 2003. — 544 с.
7. *FFmpeg.* — \www/ URL: <http://www.ffmpeg.org/>— 1.10.2010 г.
8. *YUV Video Sequences* / Arizona State University. — \www/ URL: <http://trace.eas.asu.edu/yuv/> — 1.11.2010 г.
9. *Xiph.org Test Media.* — \www/ URL: <http://media.xiph.org/video/derf/> — 1.11.2010 г. —
10. *Advanced video coding for generic audiovisual services* : ITU-T Recommendation H.264 and ISO/IEC 14496-10 (AVC), 2003.

Поступила 18.03.11;  
после доработки 27.05.11

*СКРУПСКИЙ Степан Юрьевич, аспирант кафедры компьютерных систем и сетей Запорожского национального технического университета, который окончил в 2010 г. Область научных исследований — методы компрессии видеoinформации в распределенных компьютерных системах.*