

УДК 621.391

АППАРАТНО-ОРИЕНТИРОВАННЫЙ АЛГОРИТМ СЖАТИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АДАПТИВНОЙ ИЕРАРХИЧЕСКОЙ КЛАСТЕРИЗАЦИИ ВЕЙВЛЕТ-КОЭФФИЦИЕНТОВ

В.В. НОВИЦКИЙ, В.Ю. ЦВЕТКОВ, К.М.О. АЛЬ-ГЕРТАНИ, А.А.Р. АЛЬ-АББУДИ

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь*

Поступила в редакцию 10 мая 2016

Дано строгое математическое описание нового алгоритма вейвлет-сжатия изображений, обладающего низкой вычислительной сложностью и высокой степенью параллелизма. Он предназначен для аппаратной реализации в ПЛИС, которая будет сжимать поток аэрокосмических изображений на борту спутника дистанционного зондирования Земли. В качестве прототипов были выбраны алгоритмы SPECK и МЕСТ. Численное моделирование на тестовых аэрокосмических изображениях показало хорошие результаты.

Ключевые слова: вейвлет-сжатие, спутник дистанционного зондирования Земли, ПЛИС.

Введение

Актуальность проблемы разработки новых алгоритмов сжатия обусловлена быстрым ростом количества информации, которое необходимо передавать по различным сетям и относительно медленным ростом пропускной способности каналов, на которых строятся данные сети. Частным случаем сетей с существенно ограниченной пропускной способностью каналов является система дистанционного зондирования земли (ДЗЗ), где самым медленным является канал между спутником ДЗЗ и пунктом приема целевой информации (ЦИ). Отсюда вытекает задача разработки алгоритма вейвлет-сжатия с высокой степенью параллелизма и низкой вычислительной сложностью, который хорошо подходит для реализации в кристалле ПЛИС и в то же время сопоставим по качеству восстановленных изображений с существующими алгоритмами.

Математическое описание нового алгоритма вейвлет-сжатия

Формирование масштабируемых вейвлет-структур. Наиболее эффективные алгоритмы сжатия изображений [1–5] работают в области дискретного вейвлет-преобразования. С его помощью осуществляется формирование матрицы вейвлет-коэффициентов $C = c(i, j)_{(i=0, \sqrt{N}-1, j=0, \sqrt{N}-1)}$ исходного полутонового изображения I . Двумерное дискретное вейвлет-преобразование, формирующее пространственно-частотное представление исходного изображения $I = I(i, j)_{(i=0, \sqrt{N}-1, j=0, \sqrt{N}-1)}$ можно представить в виде

$$W(I, J) = C, \quad (1)$$

где J – заданная глубина вейвлет-разложения изображения I ; $0 < J \leq J_{\max}$; $J_{\max} = \log_2(\sqrt{N})$ – максимально возможное количество уровней разложения; N – число вейвлет-коэффициентов исходного изображения I .

Математическое описание алгоритма НВСТ (Hardware Block Cluster Tree). Для формирования прогрессивного выходного потока (масштабируемого кода) необходимо кодировать матрицу вейвлет-коэффициентов C в определенном порядке (начиная со старшей битовой плоскости и пространственно-частотного поддиапазона L). В связи с этим предлагается алгоритм быстрого кодирования вейвлет-коэффициентов исходного полутонового изображения, допускающий множественное распараллеливание и предполагающий аппаратную реализацию с низким количеством вычислительных ресурсов. Данный алгоритм основан на иерархической кластеризации сегмента битовой плоскости фиксированного размера и построении оптимального кода на основе оценки компактности кластеризованного сегмента. В качестве прототипов были выбраны алгоритмы SPECK [4] и МЕСТ [6]. Алгоритм НВСТ состоит из следующих шагов.

Шаг 1. Сканирование матрицы вейвлет-коэффициентов $C = c(i, j)_{(i=0, \overline{\sqrt{N}-1}, j=0, \overline{\sqrt{N}-1})}$ размерностью $\sqrt{N} \times \sqrt{N}$ в одномерный массив $C_Z = c_Z(i)_{(i=0, \overline{N-1})}$ по Z-развертке (рис. 1):

1.1 Формирование двумерного массива координат Z-развертки $Z = z(i, j)_{(i=0, \overline{1}, j=0, \overline{N-1})}$ (пример на рис. 2);

1.2 Инициализация массива C_Z элементами матрицы C по массиву координат Z :

$$\begin{cases} C_Z = c_Z(i)_{(i=0, \overline{N-1})}, \\ c_Z(i) = c(z(0, i), z(1, i)). \end{cases} \quad (2)$$

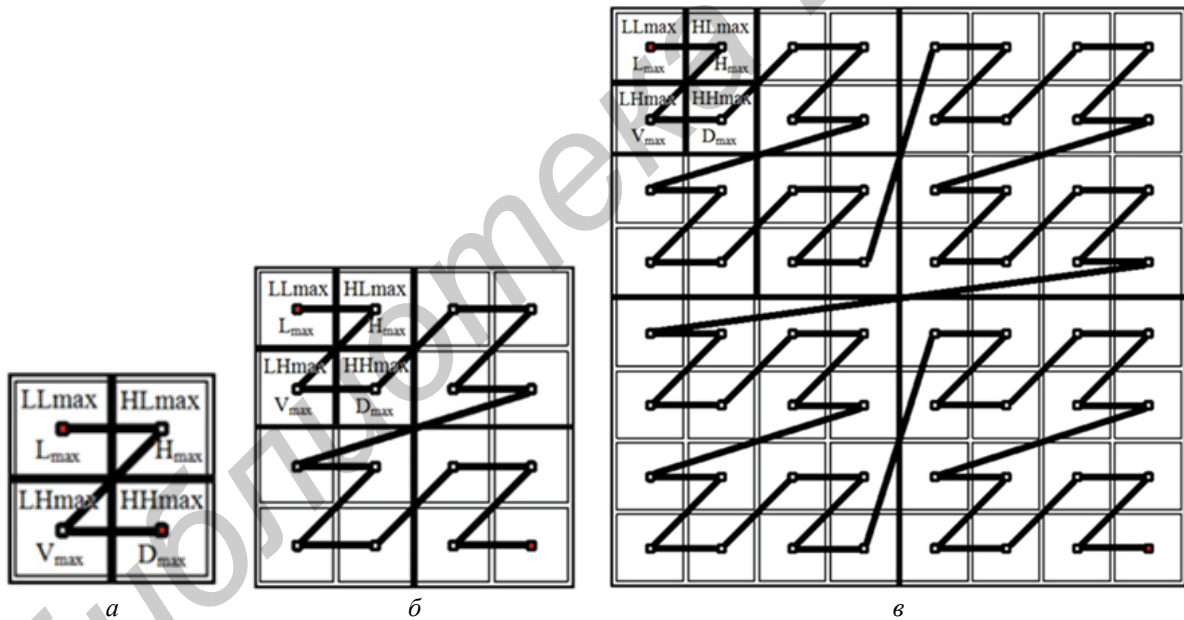


Рис. 1. Три масштаба Z-развертки: $a - (m = 1)$; $b - (m = 2)$; $v - (m = 3)$

Сканирование в таком порядке позволяет сохранить пространственное масштабирование выходного потока и существенно упрощает дальнейшее кодирование матрицы вейвлет-коэффициентов.

Шаг 2. Выбор длины кодового блока N_{CB} (сегмента битовой плоскости массива C_Z), причем длина массива C_Z должна быть кратна N_{CB} и $\log_2 N_{CB}$ должен быть целым числом не меньше трех. Оптимальным считается $N_{CB} = 1024$. Исходя из выбранного размера кодового блока, рассчитывается максимальное число уровней кластеризации

$$L_M = \log_4 N_{CB}. \quad (3)$$

В процессе кодирования используется два значения уровней кластеризации $\{1, L_M\}$. Таким образом выбор N_{CB} влияет на быстродействие, степень сжатия и вычислительную сложность алгоритма НВСТ. Как правило, чем больше N_{CB} , тем выше степень сжатия, но тем больше аппаратных ресурсов требует алгоритм для своей реализации.

	Текущий индекс	(Y,X)-координаты массива
LL2	0	(0,0)
HL2	1	(0,1)
LH2	2	(1,0)
HH2	3	(1,1)
	4	(0,2)
HL1	5	(0,3)
	6	(1,2)
	7	(1,3)
	8	(2,0)
LH1	9	(3,0)
	10	(2,1)
	11	(3,1)
	12	(2,2)
HH1	13	(2,3)
	14	(3,2)
	15	(3,3)

Рис. 2. Пример массива формирования массива координат Z-развертки для $m = 2$

Шаг 3. Задание максимально допустимого количества бит B_{MAX} для выходного прогрессивного потока. При сжатии без потерь и с потерями B_{MAX} определяется с помощью соотношения

$$B_{MAX} = \left(\frac{N_{CB} \times (N_{CB} + 2)}{N} \times B_{PP} + 8 \right) / C_R^*, \quad (4)$$

где B_{PP} – число бит на пиксель исходного изображения; C_R^* – заданный коэффициент сжатия. В случае сжатия без потерь $C_R^* = 1$.

Шаг 4. Инициализация массива M_{SP} значимых вейвлет-коэффициентов (маски)

$$\begin{cases} M_{SP} = m_{SP}(i)_{(i=0, \overline{N-1})}, \\ m_{SP}(i) = 0, \end{cases} \quad (5)$$

где $m_{SP}(i) \in \{0,1\}$; i – номер вейвлет-коэффициента $c_Z(i)$.

Шаг 5. Определение начального значения порога T значимости вейвлет-коэффициентов

$$T = 2^n, \quad (6)$$

где $n = \overline{n_{MAX}}$, 0 – номер битовой плоскости массива вейвлет-коэффициентов $C_Z = c_Z(i)_{(i=0, \overline{N-1})}$,

$c_Z(i) = \text{sign}(c_Z(i)) \times \left(\sum_{n=0}^{n_{MAX}} b_n 2^n \right)$ – значение i -го вейвлет-коэффициента; $b_n \in \{0,1\}$ – n -й бит $c_Z(i)$.

Шаг 6. Инициализация массива M_{PS} (в который будет записываться выходной прогрессивный поток алгоритма НВСТ) и текущего числа бит B_{PS} :

$$\begin{cases} M_{PS} = m_{PS}(b)_{(b=0, B_{MAX}-1)}, \\ m_{PS}(i) = \begin{cases} n_{MAX_i} & \text{при } 0 \leq i < 8, \\ 0 & \text{при } 8 \leq i < B_{MAX}, \end{cases} \end{cases} \quad (7)$$

$$B_{PS} = 8, \quad (8)$$

где $m_{PS}(b) \in \{0,1\}$; b – номер элемента (бита) массива M_{PS} ; n_{MAX_i} – i -й бит номера старшей битовой плоскости, который записывается в начало потока.

Шаг 7. Инициализация указателя P_{CB} на текущий сегмент кодируемой битовой плоскости и номера текущей битовой плоскости

$$P_{CB} = 0, \quad (9)$$

$$n = n_{MAX}. \quad (10)$$

Кодирование начинается со старшей битовой плоскости и нулевого сегмента. Количество сегментов (кодowych блоков) битовой плоскости определяется частным N / N_{CB} .

Шаг 8. Начало цикла кодирования по алгоритму НВСТ. Инициализация массива кодового блока M_{CB} битами вейвлет-коэффициентов в текущем сегменте n -й битовой плоскости

$$\begin{cases} M_{CB} = m_{CB}(i)_{(i=0, N_{CB}-1)}, \\ m_{CB}(i) = \begin{cases} 1 & \text{при } n \leq \log_2 |c_Z(P_{CB} + i)| \leq n + 1, \\ 0 & \text{иначе,} \end{cases} \end{cases} \quad (11)$$

где $m_{CB}(i) \in \{0,1\}$.

Шаг 9. Инициализация L_M -уровневого массива M_{CL} кластеризации сегмента n -й битовой плоскости

$$\begin{cases} M_{CL} = m_{CL}(i)_{(i=0, S_{LM}-1)}, \\ m_{CL}(i) = \begin{cases} m_{CB}(i) & \text{при } 0 \leq i < N_{CB}, \\ 0 & \text{при } N_{CB} \leq i < S_{LM}, \end{cases} \end{cases} \quad (12)$$

где $m_{CL}(i) \in \{0,1\}$; $S_{LM} = (4N_{CB} - N_{LM}) / 3$ – общее число исходных и кластеризованных элементов массива M_{CL} на уровнях $0, L_M$; N_{LM} – число кластерных деревьев на L_M -м уровне кластеризации

$$N_{LM} = N_{CB} / N_C^{L_M}, \quad (13)$$

где N_C – количество элементов в кластере (для алгоритма НВСТ равно 4).

Шаг 10. Инициализация коэффициентов оценки компактности для уровней кластеризации $L=1$ и $L=L_M$:

$$\{C_{L_1}, C_{L_M}\} = 0. \quad (14)$$

Шаг 11. L_M -уровневая иерархическая кластеризация M_{CL} . Значения элементов массива M_{CL} на L уровне определяются с помощью соотношения

$$m_{CL}(S(L) + i) = \text{Bin} \left(\sum_{j=0}^3 m_{CL} \left(S(L) - \frac{N_{CB}}{4^{L-1}} + 4i + j \right) \right), \quad (15)$$

где $L = \overline{1, L_M}$; $0 \leq i < \frac{N_{CB}}{4^L}$; $S(L) + i$ – абсолютный номер элемента массива M_{CL} на L -м уровне кластеризации; i – номер кластера на $(L-1)$ -м уровне или относительный номер элемента на текущем L -м уровне кластеризации; $S(L) = \sum_{l=0}^{L-1} \left(\frac{N_{CB}}{4^l} \right)$ – общее количество элементов M_{CL} , принадлежащих уровням кластеризации $[0, L-1]$; j – номер элемента в пределах кластера, $\text{Vin}(x) = \begin{cases} 1 & \text{при } |x| > 0, \\ 0 & \text{при } x = 0, \end{cases}$ – пороговая функция. Конечным результатом шага является массив

M_{CL} вплоть до L_M -го уровня, элементы которого определяются соотношением

$$m_{CL}(i) = \begin{cases} m_{CB}(i) & \text{при } 0 \leq i < N_{CB}, \\ m_{CL}(S(L) + i) & \text{при } N_{CB} \leq i < S_{L_M}. \end{cases} \quad (16)$$

Шаг 12. Оценка компактности кодирования для уровней кластеризации $L=1$ и $L=L_M$:

$$C_{L_1} = \left(\sum_{j=0}^{\frac{N_{CB}-1}{4}} m_{CL}(S(1) + j) \right) \times 4 + \frac{N_{CB}}{4}, \quad (17)$$

$$C_{L_M} = \sum_{l=1}^{L_M} \left(\left(\sum_{j=0}^{\frac{N_{CB}-1}{4^l}} m_{CL}(S(l) + j) \right) \times 4 \right), \quad (18)$$

$$C_{\min} = \min(C_L, C_{L_M}, N_{CB}), \quad (19)$$

где C_{\min} – минимально возможное число бит для кодирования заданного сегмента M_{CB} алгоритмом НВСТ.

Шаг 13. Проверка выполнения условия $C_{\min} = C_{L_M}$. В случае истины оптимальным является код, получаемый в результате кластеризации с числом уровней $L=L_M$. Перед началом кодирования в выходной поток необходимо поместить флаг, который будет указывать декодеру количество уровней кластеризации:

$$\begin{cases} m_{PS}(B_{PS}) = 0, \\ m_{PS}(B_{PS} + 1) = \begin{cases} 0 & \text{при } C_{L_M} = 0, \\ 1 & \text{иначе,} \end{cases} \\ B_{PS} = B_{PS} + 2. \end{cases} \quad (20)$$

Флаг «00» сигнализирует о том, что в текущем сегменте нет значимых вейвлет-коэффициентов (все биты сегмента равны нулю). После записи флага идет формирование и запись в выходной поток кода, компактно описывающего текущий сегмент n -й битовой плоскости:

$$m_{PS}(B_{PS}) = \begin{cases} m_{CL}(S(L-1) + j) & \text{при } m_{CL}\left(S(L) + \frac{j}{4}\right) > 0, \\ B_{PS} = B_{PS} + 1 & \text{при } m_{CL}\left(S(L) + \frac{j}{4}\right) > 0, \end{cases} \quad (21)$$

где $L = \overline{1, L_M}$; $0 \leq j < \frac{N_{CB}}{4^{L-1}}$; $S(L) + \frac{j}{4}$ – абсолютный номер вершины текущего кластерного дерева.

Шаг 14. Проверка выполнения условия $C_{\min} = C_{L_1}$. В случае истины оптимальным является код, получаемый в результате кластеризации с числом уровней $L = L_1$. Перед началом кодирования в выходной поток необходимо поместить флаг, который будет указывать декодеру количество уровней кластеризации

$$\begin{cases} m_{PS}(B_{PS}) = 1, \\ m_{PS}(B_{PS} + 1) = 0, \\ B_{PS} = B_{PS} + 2. \end{cases} \quad (22)$$

После записи флага идет формирование и запись в выходной поток кода, компактно описывающего текущий сегмент n -й битовой плоскости:

$$m_{PS}(B_{PS}) = \begin{cases} m_{CL}(S(1) + j) \text{ при } 0 \leq j < \frac{N_{CB}}{4}, \\ B_{PS} = B_{PS} + 1, \end{cases} \quad (23)$$

$$m_{PS}(B_{PS}) = \begin{cases} m_{CL}(S(0) + j) \text{ при } m_{CL}\left(S(1) + \frac{j}{4}\right) > 0, \\ B_{PS} = B_{PS} + 1 \text{ при } m_{CL}\left(S(1) + \frac{j}{4}\right) > 0, \end{cases} \quad (24)$$

где $0 \leq j < N_{CB}$; $S(1) + \frac{j}{4}$ – абсолютный номер вершины текущего кластерного дерева.

Шаг 15. Проверка выполнения условия $C_{\min} = N_{CB}$. В случае истины текущий сегмент n -й битовой плоскости является шумом, и оптимальным вариантом будет его непосредственная передача в выходном потоке после записи соответствующего флага:

$$\begin{cases} m_{PS}(B_{PS}) = 1, \\ m_{PS}(B_{PS} + 1) = 1, \\ B_{PS} = B_{PS} + 2. \end{cases} \quad (25)$$

$$m_{PS}(B_{PS}) = \begin{cases} m_{CL}(S(0) + j) \text{ при } 0 \leq j < N_{CB}, \\ B_{PS} = B_{PS} + 1. \end{cases} \quad (26)$$

Шаг 16. Формирование информации о знаках значимых в текущем сегменте n -й битовой плоскости вейвлет-коэффициентов и запись их в выходной поток:

$$\begin{cases} m_{PS}(B_{PS}) = \begin{cases} 1 \text{ при } m_{CB}(i) > 0 \text{ и } m_{SP}(P_{CB} + i) = 0 \text{ и } c_z(P_{CB} + i) \geq 0, \\ 0 \text{ при } m_{CB}(i) > 0 \text{ и } m_{SP}(P_{CB} + i) = 0 \text{ и } c_z(P_{CB} + i) < 0, \end{cases} \\ B_{PS} = B_{PS} + 1, \end{cases} \quad (27)$$

где $0 \leq i < N_{CB}$.

Шаг 17. Запись информации о значимости вейвлет-коэффициентов текущего сегмента n -й битовой плоскости в массив M_{SP} . В данном массиве также хранится информация о значимости вейвлет-коэффициентов всех предыдущих битовых плоскостей:

$$m_{SP}(i) = m_{CB}(P_{CB} + i) \text{ при } m_{CB}(P_{CB} + i) > 0, \quad (28)$$

где $0 \leq i < N_{CB}$.

Шаг 18. Окончание цикла кодирования вейвлет-коэффициентов и завершающий шаг алгоритма. Выполнение инкремента указателя на текущий сегмент n -й битовой плоскости:

$$P_{CB} = P_{CB} + N_{CB} \cdot \quad (29)$$

Проверка на конец битовой плоскости $P_{CB} = N_{CB}$. В случае истины декремент номера текущей битовой плоскости

$$n = n - 1, P_{CB} = 0, \quad (30)$$

иначе – переход к шагу 8. Проверка n на равенство нулю и завершение алгоритма НВСТ, иначе – переход к шагу 8.

Результаты и их обсуждение

В таблице представлены численные характеристики алгоритмов JPEG2000 [5], МЕСТ [6] и НВСТ для известного тестового изображения «Lena».

Численные результаты алгоритмов для изображения «Lena»

Алгоритм	Режим сжатия	Без потерь		Потери, 2 bpp		Потери, 1 bpp		Потери, 0.5 bpp	
JPEG2000	Семейство вейвлетов	CDF 5/3		CDF 9/7		CDF 9/7		CDF 9/7	
	Схема, уровни ДВП	Пирамида, 5		Пирамида, 5		Пирамида, 5		Пирамида, 5	
	CR	1,861		4,032		8,033		16,072	
	PSNR, дБ	∞		43,364		39,290		36,289	
МЕСТ	Семейство вейвлетов	CDF 5/3		CDF 9/7		CDF 9/7		CDF 9/7	
	Схема, уровни ДВП	Пирамида, 7		Пирамида, 7		Пирамида, 7		Пирамида, 7	
	CR	1,757		3,998		7,995		15,981	
	PSNR, дБ	∞		42,856		38,602		35,242	
НВСТ	Семейство вейвлетов	CDF 5/3		CDF 9/7		CDF 9/7		CDF 9/7	
	Схема, уровни ДВП	Пир. 4	Клас. 4	Пир. 4	Клас. 4	Пир. 4	Клас. 4	Пир. 4	Клас. 4
	CR	1,791	1,750	3,998	3,999	7,999	7,993	15,972	15,994
	PSNR, дБ	∞	∞	43,565	43,111	38,945	38,665	35,655	35,404
	Семейство вейвлетов	CDF 5/3		CDF 5/3	Haar	CDF 5/3	Haar	CDF 5/3	Haar
	Схема, уровни ДВП	Пирамида, 3		Пир. 3	Пир. 4	Пир. 3	Пир. 4	Пир. 3	Пир. 4
	CR	1,780		3,999	3,999	7,993	7,998	15,999	15,988
	PSNR, дБ	∞		41,875	41,128	37,539	36,107	33,431	32,360

При вычислении численных результатов в алгоритме НВСТ использовались различные вариации вейвлет-декомпозиции исходного изображения, такие как: количество уровней декомпозиции, схема декомпозиции и семейство вейвлетов. Это делалось с учетом того факта, что аппаратная реализация, к примеру, декомпозиции по классической схеме требует меньше ресурсов и времени, чем пирамидальная схема. В итоге можно сделать вывод, что алгоритм НВСТ выигрывает 0,6 дБ у алгоритма МЕСТ и проигрывает 1 дБ алгоритму JPEG2000 (кодэк Jasper) независимо от типа изображения.

Заключение

Разработан простой и эффективный метод вейвлет-сжатия НВСТ, который обладает низкой вычислительной сложностью и высокой степенью параллелизма, что делает его хорошо пригодным для реализации в кристалле ПЛИС. Численное моделирование для тестовых изображений показало, что НВСТ выигрывает у прототипа МЕСТ порядка 0,6 дБ по метрике PSNR при сжатии с потерями и проигрывает 1 дБ JPEG2000, который считается лучшим алгоритмом вейвлет-сжатия изображений с потерями, но обладает высокой вычислительной сложностью.

THE HARDWARE-ORIENTED IMAGE COMPRESSION ALGORITHM BASED ON ADAPTIVE HIERARCHICAL CLUSTERING OF THE WAVELET COEFFICIENTS

V.U. NAVITSKI, V.Yu. TSVIATKOU, Q.M.O. AL-GERTANI, A.A.R. AL-ABBOODI

Abstract

The new algorithm of image wavelet-compression has been mathematically described, it has low-complexity and high degree of parallelism. It is destined to FPGA hardware implementation, which will compress a stream of aerospace on the board of ERS satellite. There are two algorithms, such as SPECK and MECT, were chosen as prototypes. Numerical modeling with test aerospace images has shown good results.

Keywords: wavelet compression, Earth remote sensing satellite, FPGA.

Список литературы

1. *Shapiro J.M.* // IEEE Trans. Signal Processing. 1993. №41. P. 3445-3462.
2. *Said A.A., Pearlman W.A.* // IEEE Trans. On Circuits and Systems for Video Technology. 1996. Vol. 6. P. 243-250.
3. *Taubman D.S.* // IEEE Transactions on Image Processing. 2000. Vol. 9. №7. P. 1158-1170.
4. *Islam A., Pearlman W.A.* // ISO/IEC/JTC1/SC29. 1998. №873. P. 312-326.
5. *Taubman D.S., Marcellin M.W.* // JPEG2000: Image Compression, Fundamentals, Standards, and Practice. Berlin, 2002.
6. *Борискевич А.А., Цветков В.Ю.* // Докл. НАН Беларуси. 2009. Т. 53. №3. С. 38-48.