

## ЦЕЛОЧИСЛЕННОЕ КОСИНУСНОЕ ПРЕОБРАЗОВАНИЕ ВЫСОКОГО ПОРЯДКА: МЕТОД ПОСТРОЕНИЯ И БЫСТРЫЕ АЛГОРИТМЫ ДЛЯ КОДИРОВАНИЯ ИЗОБРАЖЕНИЙ И ВИДЕО С ВЫСОКИМ РАЗРЕШЕНИЕМ

**Аннотация.** Предложен матричный метод построения однонормового целочисленного косинусного преобразования высокого порядка. Построено однонормовое целочисленное преобразование порядка 32 и разработаны его быстрые алгоритмы низкой вычислительной сложности (меньшей в 3,24 раза, чем в известных алгоритмах, и в 15,6 раз — чем в стандарте H.265), и они требуют только целочисленных операций. Это преобразование не только наиболее близкое к дискретному косинусному преобразованию, но и имеет превосходящие характеристики кодирования.

**Ключевые слова:** дискретное косинусное преобразование, целочисленное косинусное преобразование, масштабированное преобразование, факторизация, быстрое преобразование, эффективность кодирования, вычислительная сложность, коэффициент сжатия, видеокodирование, H.264, H.265, AVS.

### ВВЕДЕНИЕ

Эффективными инструментами для устранения избыточности в изображениях и видеосигналах являются методы сжатия, основанные на модификации и сжатии результатов преобразования, или так называемые методы трансформационного кодирования [1]. Системы кодирования с преобразованием, основанные на различных дискретных преобразованиях, достаточно хорошо исследованы и изучены. Выбор наилучшего преобразования для конкретного приложения зависит от величины допустимой ошибки восстановления и от имеющихся вычислительных ресурсов. Сжатие же возникает не в процессе преобразования, а на этапе квантования полученных коэффициентов.

Стандарты сжатия видеоданных расширяют трансформационные методы сжатия неподвижных изображений путем сокращения временной или межкадровой избыточности. Хотя в настоящее время имеются различные стандарты кодирования, тем не менее большинство из них основано на использовании подобных методов сжатия видеоданных. Цель этапа преобразования при обработке видеокodeком — отображение цифровых данных полных или остаточных кадров в другую координатную область (область преобразования). Выбор такого преобразования определяется по многим критериям [2].

1. Цифровые данные, отображаемые в область преобразования, должны быть декоррелированными и компактными (т.е. большая часть энергии преобразованных данных должна быть сконцентрирована в малом числе коэффициентов преобразования).

2. Преобразование должно быть обратимым.

3. Преобразование должно быть эффективным относительно его вычислимости (малый объем требуемой памяти, возможность реализации в арифметике с ограниченной точностью вычислений, малое число арифметических операций в алгоритме и т.д.)

Для сжатия изображений и видео существует много преобразований, но все они разделяются на две категории: преобразования на основе блоков и преобразования на основе изображений.

Примерами преобразований на основе блоков являются преобразования Карунена–Лоэва (KLT), разложение по сингулярным числам (SVD) и наиболее известное дискретное косинусное преобразование (ДКП). В каждом таком преоб-

разовании используются блоки  $N \times N$  сэмплов исходного или остаточного изображения. Блочные преобразования имеют низкие требования к объему памяти и являются эффективными при сжатии остаточных изображений, полученных на основе блоковой компенсации движения. Однако они уязвимы для артефактов, возникающих на стыке блоков, что приводит к появлению блочности.

Преобразования на основе изображений работают с целым кадром (или большей частью изображения). Из такого типа преобразований наиболее известным является дискретное вейвлет-преобразование (ДВП). Такие преобразования изображений превосходят блочные преобразования при сжатии неподвижных изображений, но при этом требуется большой объем памяти (поскольку необходимо хранить целиком весь кадр или большую его часть), и они плохо сочетаются с компенсацией движения на основе блоков. Преобразования ДКП и ДВП реализованы в стандарте MPEG-4 Visual [2]. В зависимости от предназначения стандарты подразделяются на две большие группы: стандарты для видеоконференций и мультимедийные стандарты.

Современные видеостандарты, такие, как H.264/AVC [2, 3], VC-1 [4], AVS [5], H.265/HEVC [6, 7], используют целочисленные косинусные преобразования (ЦКП) размеров в пределах от 4 до 32. Среди известных преобразований дискретное косинусное преобразование обеспечивает упаковку наибольшего количества информации в наименьшее число коэффициентов (для большинства реальных изображений), а также минимизирует эффект появления блочной структуры (искусственных искажений), называемой блоковыми искажениями (артефактами), когда на изображении проявляются границы между соседними блоками. Кроме этого, ДКП имеет быстрые алгоритмы вычисления. Современные стандарты видеокодирования применяют целочисленную реализацию ДКП. Это связано с тем, что вычисление ЦКП выполняется без плавающей запятой и его ядро содержит только целые и упаковывает энергию подобно ДКП. Однако преобразования, используемые в стандарте H.265, не имеют быстрых алгоритмов, что приводит к значительным вычислительным затратам при больших размерах 2D преобразования.

В работах [8–10] рассмотрены преобразования, основанные на целочисленной аппроксимации ДКП с быстрыми алгоритмами меньшей вычислительной сложности, чем преобразования в стандарте H.265. В [11] предложен метод построения ЦКП с быстрыми алгоритмами, а в [12] предложено ЦКП порядка 32 с быстрыми алгоритмами низкой вычислительной сложности, которая меньше в 1,5 и 2,7 раза, чем соответственно в известных алгоритмах из [9] и из [10], и в 13 раз меньше, чем в стандарте H.265 [7].

В [13, 14] предложен метод построения ЦКП порядков 8 и 16 с быстрыми алгоритмами, которые в отличие от известных алгоритмов [9, 10] имеют в 3–5 раз меньшую вычислительную сложность и в 10 раз меньшую, чем преобразования в стандарте H.265. В [15] было предложено несколько эффективных методов уменьшения сложности кодера, таких, как улучшенное кодирование intra-picture и упрощенные VLC коэффициенты.

В настоящей статье предложен матричный метод построения однонормового целочисленного косинусного преобразования высокого порядка и на его основе построено однонормовое ЦКП порядка 32 с алгоритмами быстрых целочисленных преобразований, которые приводят к существенному сокращению вычислений, т.е. имеют низкую вычислительную сложность. Экспериментальные результаты показали, что это преобразование имеет такие же превосходные характеристики кодирования, как и преобразование в стандарте H.265, т.е. более близкое к ДКП.

#### **МЕТОД ПОСТРОЕНИЯ ОДНОНОРМОВОГО ЦЕЛОЧИСЛЕННОГО КОСИНУСНОГО ПРЕОБРАЗОВАНИЯ ПОРЯДКА 32**

Рассмотрим метод построения однонормового ЦКП порядка 32, который представлен в работе [11].

Пусть матрица  $ICT_{32}^*$  размера  $32 \times 32$  ЦКП с переставленными строками получена из матрицы  $ICT_{32}$  перестановкой строк сначала на основе двоично-ин-

версных перестановок (ДИП), а затем обратных (инверсных) перестановок (ОП) и перестановок по коду Грея (ПКГ) [16]:

$$ICT_{32}^* = P_2 P_1 P_{32} ICT_{32}, \quad (1)$$

где  $P_{32}$  — матрица  $32 \times 32$  ДИП;  $P_1$  — блочно-диагональная матрица  $32 \times 32$  с единичными матрицами  $I_{12}$  размера  $12 \times 12$ ,  $I_8$  размера  $8 \times 8$ , матрицами обратных перестановок  $\bar{I}_4$  размера  $4 \times 4$  и  $\bar{I}_8$  размера  $8 \times 8$ ;  $P_1 = \text{diag} [I_{12}, \bar{I}_4, I_8, \bar{I}_8]$ ,  $\bar{I}_k = \text{antidiag} [I_k]$ ,  $k = 4, 8$ ,  $\bar{I}_4, \bar{I}_8$  — антидиагональные единичные матрицы размера  $4 \times 4$  и  $8 \times 8$ ;  $P_2$  — блочно-диагональная матрица, содержащая матрицы  $G_4$  размера  $4 \times 4$  ПКГ и  $P_4$  ДИП,  $G_4 = \text{diag} [I_2, \bar{I}_2]$ ,  $P_4 = \text{diag} [1, \bar{I}_2, 1]$ ,  $I_2$  — матрица размера  $2 \times 2$  ОП,  $P_2 = \text{diag} [G_8', I_2 \otimes G_4, I_2 \otimes G_8']$ ,  $G_8' = \text{diag} [G_4, P_4]$ ,  $\bar{I}_2 = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}$ . Здесь символ  $\otimes$  означает операцию кронекеровского произведения матриц,  $I_2$  — единичная матрица  $2 \times 2$ ,  $G_8'$  — блочно-диагональная матрица  $8 \times 8$ , содержащая матрицы  $G_4$  и  $P_4$ .

Матрица  $ICT_{32}^*$  размера  $32 \times 32$  ЦКП с переставленными строками может быть изображена матрицей ядра ЦКП:

$$ICT_{32}^* = B_{32} C_{32}^*, \quad (2)$$

где  $C_{32}^*$  — матрица  $32 \times 32$  ядра ЦКП с переставленными строками,  $B_{32}$  — диагональная матрица  $32 \times 32$  коэффициентов нормирования. Матрица  $C_{32}^*$  может быть представлена рекуррентным методом [11]:

$$C_{32}^* = \text{diag} [C_{16}^*, Q_{16}] H_{32}^*, \quad (3)$$

где  $C_{16}^*$  — матрица  $16 \times 16$  ядра ЦКП с переставленными строками на основе ДКП, ПКГ и ОП;  $H_{32}^*$  — фактор-матрица  $32 \times 32$  с ненулевыми элементами  $\pm 1$ ;  $Q_{16}$  — матрица  $16 \times 16$ , которая может быть представлена матрицей ЦКП типа IV (ЦКП-IV):

$$Q_{16} = C_{16}^{IV*} \bar{I}_{16}, \quad \bar{I}_{16} = \text{antidiag} [I_{16}], \quad (4)$$

где  $C_{16}^{IV*}$  — матрица  $16 \times 16$  ядра ЦКП-IV с переставленными строками,  $\bar{I}_{16}$  — антидиагональная единичная матрица  $16 \times 16$ .

Матрица  $C_{16}^{IV*}$  может быть представлена через матрицу ЦКП типа II (ЦКП-II) [12]:

$$C_{16}^{IV*} = H_{16}^0 \text{diag} [T_8^*, T_8^*] R_{16}, \quad T_8^* = G_8' P_8 T_8, \quad (5)$$

где  $T_8^*$  — матрица  $8 \times 8$  ядра ЦКП-II (или ЦКП) с переставленными строками на основе ДИП и ПКГ,  $P_8$  — матрица  $8 \times 8$  ДИП,  $H_{16}^0$  — фактор-матрица  $16 \times 16$  с ненулевыми элементами  $\pm 1$ , которая на основной диагонали имеет единичные матрицы  $I_8$  размера  $8 \times 8$  и  $\pm I_4$  размера  $4 \times 4$ , а на другой диагонали — матрицы  $G_8^0$  и  $\tilde{G}_8^0$  размера  $8 \times 8$ ,  $G_8^0 = \text{diag} [G_4^0, \bar{I}_4]$ ,  $\tilde{G}_8^0 = \text{diag} [-G_4^0, \bar{I}_4]$ ,  $\tilde{G}_8^0, G_8^0$  — блочно-диагональные матрицы  $8 \times 8$  с матрицами  $\pm G_4^0$  размера  $4 \times 4$  и  $\bar{I}_4$ ,  $G_4^0 = \text{diag} [0, 1, \bar{I}_2]$ . Здесь  $R_{16}$  — матрица  $16 \times 16$  растягивания (для ДКП-IV представляет матрицу вращения Гивенса), которая на основной диагонали содержит целые элементы  $\pm r_i$ , а на другой диагонали — целые элементы  $\pm s_i$ , причем  $r_1 = r_2$ ,  $r_i > r_{i+1}$ ,  $i = 2, 7$ ;  $s_i < s_{i+1}$ ,  $i = 1, 7$ .

Отметим, что

$$H_{32}^* = \begin{bmatrix} I_{16} & \bar{I}_{16} \\ \bar{I}_{16} & -I_{16} \end{bmatrix}, H_{16}^0 = \left[ \begin{array}{c|c} I_8 & -G_4^0 \\ \hline G_4^0 & I_4 \\ \hline \bar{I}_4 & -I_4 \end{array} \right], \quad (6)$$

$$G_4^0 = \left[ \begin{array}{c|c} 0 & \\ \hline 1 & \\ \hline & 1 \end{array} \right], \bar{I}_4 = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{bmatrix},$$

$$R_{16} = \begin{bmatrix} r_1 & & & & & & & & & & & & & & & s_1 \\ & r_2 & & & & & & & & & & & & & & s_2 \\ & & r_3 & & & & & & & & & & & & & s_3 \\ & & & r_4 & & & & & & & & & & & & s_4 \\ & & & & r_5 & & & & & & & & & & & s_5 \\ & & & & & r_6 & & & & & & & & & & s_6 \\ & & & & & & r_7 & & & & & & & & & s_7 \\ & & & & & & & r_8 & s_8 & & & & & & & \\ & & & & & & & -s_8 & r_8 & & & & & & & \\ & & & & & & & & & -r_7 & & & & & & \\ & & & & & & & & & & r_6 & & & & & \\ & & & & & & & & & & & -r_5 & & & & \\ & & & & & & & & & & & & r_4 & & & \\ & & & & & & & & & & & & & -r_3 & & \\ & & & & & & & & & & & & & & r_2 & \\ & & & & & & & & & & & & & & & -r_1 \\ s_1 & & & & & & & & & & & & & & & \end{bmatrix}.$$

Матрица  $Q_{16}$  на основании (4) и (5) может быть представлена как произведение четырех матриц:

$$Q_{16} = H_{16}^0 \text{diag} [T_8^*, T_8^*] R_{16} \bar{I}_{16} \quad (7)$$

или с учетом произведения двух матриц  $R_{16} \bar{I}_{16} = \bar{R}_{16}$  как произведение трех матриц:

$$Q_{16} = H_{16}^0 \text{diag} [T_8^*, T_8^*] \bar{R}_{16}, \quad (8)$$

где

$$\bar{R}_{16} = \begin{bmatrix} s_1 & & & & & & & & & & & & & & & r_1 \\ & s_2 & & & & & & & & & & & & & & r_2 \\ & & s_3 & & & & & & & & & & & & & r_3 \\ & & & s_4 & & & & & & & & & & & & r_4 \\ & & & & s_5 & & & & & & & & & & & r_5 \\ & & & & & s_6 & & & & & & & & & & r_6 \\ & & & & & & s_7 & & & & & & & & & r_7 \\ & & & & & & & s_8 & r_8 & & & & & & & \\ & & & & & & & r_8 & -s_8 & & & & & & & \\ & & & & & & & & & -r_7 & & & & & & \\ & & & & & & & & & & s_7 & & & & & \\ & & & & & & & & & & & -s_6 & & & & \\ & & & & & & & & & & & & r_6 & & & \\ & & & & & & & & & & & & & -r_5 & & \\ & & & & & & & & & & & & & & s_5 & \\ & & & & & & & & & & & & & & & -s_4 \\ & & & & & & & & & & & & & & & r_4 \\ & & & & & & & & & & & & & & & -r_3 \\ & & & & & & & & & & & & & & & s_3 \\ & & & & & & & & & & & & & & & -s_2 \\ -r_1 & & & & & & & & & & & & & & & s_1 \end{bmatrix}, \quad (9)$$

$$T_8^* = \begin{bmatrix} g' & g' & g' & g' & g' & g' & g' & g' & g' \\ g' & -g' & -g' & g' & g' & -g' & -g' & g' & g' \\ f' & -e' & e' & -f' & -f' & e' & -e' & f' & f' \\ e' & f' & -f' & -e' & -e' & -f' & f' & e' & e' \\ a' & b' & c' & d' & -d' & -c' & -b' & -a' & -a' \\ b' & -d' & -a' & -c' & c' & a' & d' & -b' & -b' \\ c' & -a' & d' & b' & -b' & -d' & a' & -c' & -c' \\ d' & -c' & b' & -a' & a' & -b' & c' & -d' & -d' \end{bmatrix},$$

здесь  $a' > b' > c' > d'$ ,  $e' > f'$ ,  $g' = 2^n$ .

Матрица  $C_{16}^{IV*}$  ядра ЦКП-IV порядка 16 содержит целые элементы  $\pm c_{2i-1}$ ,  $i = \overline{1, 16}$ ,  $\pm a_i$ ,  $\pm b_i$ ,  $\pm d_i$ ,  $\pm e_i$ ,  $\pm f_i$ ,  $\pm g_i$ ,  $\pm h_i$  и имеет вид:

$$C_{16}^{IV*} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 & c_9 & c_{11} & c_{13} & c_{15} & c_{17} & c_{19} & c_{21} & c_{23} & c_{25} & c_{27} & c_{29} & c_{31} \\ h_{16} & -h_{15} & -h_{14} & h_{13} & h_{12} & -h_{11} & -h_{10} & h_9 & -h_8 & -h_7 & h_6 & h_5 & -h_4 & -h_3 & h_2 & h_1 \\ d_{16} & -d_{15} & d_{14} & -d_{13} & d_{12} & d_{11} & -d_{10} & d_9 & -d_8 & -d_7 & d_6 & -d_5 & d_4 & -d_3 & -d_2 & d_1 \\ e_1 & e_2 & -e_3 & -e_4 & -e_5 & e_6 & e_7 & e_8 & e_9 & -e_{10} & -e_{11} & -e_{12} & e_{13} & e_{14} & e_{15} & e_{16} \\ b_1 & b_2 & b_3 & -b_4 & -b_5 & -b_6 & -b_7 & -b_8 & -b_9 & -b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ g_1 & -g_2 & -g_3 & -g_4 & g_5 & g_6 & -g_7 & -g_8 & -g_9 & g_{10} & g_{11} & -g_{12} & -g_{13} & -g_{14} & g_{15} & g_{16} \\ f_{16} & -f_{15} & f_{14} & f_{13} & -f_{12} & f_{11} & f_{10} & -f_9 & f_8 & f_7 & -f_6 & f_5 & f_4 & -f_3 & f_2 & f_1 \\ a_{16} & -a_{15} & a_{14} & -a_{13} & a_{12} & -a_{11} & a_{10} & -a_9 & a_8 & -a_7 & a_6 & a_5 & -a_4 & a_3 & -a_2 & a_1 \\ c_{31} & -c_{29} & c_{27} & -c_{25} & c_{23} & -c_{21} & c_{19} & -c_{17} & c_{15} & -c_{13} & c_{11} & -c_9 & c_7 & -c_5 & c_3 & -c_1 \\ h_1 & -h_2 & -h_3 & h_4 & h_5 & -h_6 & -h_7 & h_8 & h_9 & h_{10} & -h_{11} & -h_{12} & h_{13} & h_{14} & -h_{15} & -h_{16} \\ d_1 & d_2 & -d_3 & -d_4 & -d_5 & -d_6 & -d_7 & d_8 & d_9 & d_{10} & d_{11} & -d_{12} & -d_{13} & -d_{14} & -d_{15} & -d_{16} \\ e_{16} & -e_{15} & e_{14} & -e_{13} & -e_{12} & e_{11} & -e_{10} & -e_9 & e_8 & -e_7 & e_6 & e_5 & -e_4 & e_3 & e_2 & -e_1 \\ b_{16} & -b_{15} & b_{14} & -b_{13} & b_{12} & -b_{11} & -b_{10} & b_9 & -b_8 & b_7 & -b_6 & b_5 & -b_4 & -b_3 & b_2 & -b_1 \\ g_{16} & -g_{15} & -g_{14} & g_{13} & -g_{12} & -g_{11} & g_{10} & g_9 & -g_8 & g_7 & g_6 & -g_5 & -g_4 & g_3 & -g_2 & -g_1 \\ f_1 & -f_2 & -f_3 & -f_4 & f_5 & f_6 & f_7 & -f_8 & -f_9 & -f_{10} & f_{11} & f_{12} & f_{13} & -f_{14} & -f_{15} & -f_{16} \\ a_1 & a_2 & a_3 & a_4 & a_5 & -a_6 & -a_7 & -a_8 & -a_9 & -a_{10} & -a_{11} & -a_{12} & -a_{13} & -a_{14} & -a_{15} & -a_{16} \end{bmatrix}, \quad (10)$$

где  $c_1 = c_3$ ,  $c_{2i-1} > c_{2i+1}$ ,  $i = \overline{2, 15}$ .

Для однонормового преобразования матрица  $H_{16}^0$  согласно (6) содержит ненулевые элементы 1 и  $\pm P_i / 2^{m_i}$ ,  $i = \overline{1, 3}$ ,  $P_i$  — целые.

#### ЦЕЛОЧИСЛЕННОЕ КОСИНУСНОЕ ПРЕОБРАЗОВАНИЕ ПОРЯДКА 16

Представим матрицу  $C_{16}$  ядра ЦКП порядка 16 [13]:

$$C_{16} = \begin{bmatrix} k & k & k & k & k & k & k & k & k & k & \dots & k \\ A & B & C & D & E & F & G & H & -H & \dots & -A \\ a & b & c & d & -d & -c & -b & -a & -a & \dots & a \\ A_1 & A_2 & A_3 & -A_4 & -A_5 & -A_6 & -A_7 & -A_8 & -A_8 & \dots & -A_1 \\ i & j & -j & -i & -i & -j & j & i & i & \dots & i \\ B_1 & B_2 & -B_3 & -B_4 & -B_5 & B_6 & B_7 & B_8 & -B_8 & \dots & -B_1 \\ e & -f & -g & -h & h & g & f & -e & -e & \dots & e \\ C_1 & -C_2 & -C_3 & C_4 & C_5 & C_6 & -C_7 & -C_8 & C_8 & \dots & -C_1 \\ k & -k & -k & k & k & -k & -k & k & k & \dots & k \\ C_8 & -C_7 & -C_6 & C_5 & -C_4 & -C_3 & C_2 & C_1 & -C_1 & \dots & -C_8 \\ h & -g & f & e & -e & -f & g & -h & -h & \dots & h \\ B_8 & -B_7 & B_6 & B_5 & -B_4 & B_3 & B_2 & -B_1 & B_1 & \dots & -B_8 \\ j & -i & i & -j & -j & i & -i & j & j & \dots & j \\ A_8 & -A_7 & A_6 & -A_5 & A_4 & A_3 & -A_2 & A_1 & -A_1 & \dots & -A_8 \\ d & -c & b & -a & a & -b & c & -d & -d & \dots & d \\ H & -G & F & -E & D & -C & B & -A & A & \dots & -H \end{bmatrix}. \quad (11)$$

Таким образом, матрица  $C_{32}$  ядра ЦКП на основании (3), (4) и с учетом (10), (11) может быть представлена как

$$C_{32} =$$

<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>
<i>c</i> <sub>1</sub>	<i>c</i> <sub>3</sub>	<i>c</i> <sub>5</sub>	<i>c</i> <sub>7</sub>	<i>c</i> <sub>9</sub>	<i>c</i> <sub>11</sub>	<i>c</i> <sub>13</sub>	<i>c</i> <sub>15</sub>	<i>c</i> <sub>17</sub>	<i>c</i> <sub>19</sub>	<i>c</i> <sub>21</sub>	<i>c</i> <sub>23</sub>	<i>c</i> <sub>25</sub>	<i>c</i> <sub>27</sub>	<i>c</i> <sub>29</sub>	<i>c</i> <sub>31</sub>	<i>c</i> <sub>31</sub>	<i>c</i> <sub>31</sub>	<i>c</i> <sub>31</sub>	<i>c</i> <sub>31</sub>
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>-H</i>	<i>-G</i>	<i>-F</i>	<i>-E</i>	<i>-D</i>	<i>-C</i>	<i>-B</i>	<i>-A</i>	<i>-A</i>	<i>-A</i>	<i>-A</i>	<i>-A</i>
<i>a</i> <sub>1</sub>	<i>a</i> <sub>2</sub>	<i>a</i> <sub>3</sub>	<i>a</i> <sub>4</sub>	<i>a</i> <sub>5</sub>	<i>-a</i> <sub>6</sub>	<i>-a</i> <sub>7</sub>	<i>-a</i> <sub>8</sub>	<i>-a</i> <sub>9</sub>	<i>-a</i> <sub>10</sub>	<i>-a</i> <sub>11</sub>	<i>-a</i> <sub>12</sub>	<i>-a</i> <sub>13</sub>	<i>-a</i> <sub>14</sub>	<i>-a</i> <sub>15</sub>	<i>-a</i> <sub>16</sub>	<i>a</i> <sub>16</sub>	<i>a</i> <sub>16</sub>	<i>a</i> <sub>16</sub>	<i>a</i> <sub>16</sub>
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>-d</i>	<i>-c</i>	<i>-b</i>	<i>-a</i>	<i>-a</i>	<i>-b</i>	<i>-c</i>	<i>-d</i>	<i>d</i>	<i>c</i>	<i>b</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>
<i>b</i> <sub>1</sub>	<i>b</i> <sub>2</sub>	<i>b</i> <sub>3</sub>	<i>-b</i> <sub>4</sub>	<i>-b</i> <sub>5</sub>	<i>-b</i> <sub>6</sub>	<i>-b</i> <sub>7</sub>	<i>-b</i> <sub>8</sub>	<i>-b</i> <sub>9</sub>	<i>-b</i> <sub>10</sub>	<i>b</i> <sub>11</sub>	<i>b</i> <sub>12</sub>	<i>b</i> <sub>13</sub>	<i>b</i> <sub>14</sub>	<i>b</i> <sub>15</sub>	<i>b</i> <sub>16</sub>	<i>-b</i> <sub>16</sub>	<i>-b</i> <sub>16</sub>	<i>-b</i> <sub>16</sub>	<i>-b</i> <sub>16</sub>
<i>A</i> <sub>1</sub>	<i>A</i> <sub>2</sub>	<i>A</i> <sub>3</sub>	<i>-A</i> <sub>4</sub>	<i>-A</i> <sub>5</sub>	<i>-A</i> <sub>6</sub>	<i>-A</i> <sub>7</sub>	<i>-A</i> <sub>8</sub>	<i>A</i> <sub>8</sub>	<i>A</i> <sub>7</sub>	<i>A</i> <sub>6</sub>	<i>A</i> <sub>5</sub>	<i>A</i> <sub>4</sub>	<i>-A</i> <sub>3</sub>	<i>-A</i> <sub>2</sub>	<i>-A</i> <sub>1</sub>	<i>-A</i> <sub>1</sub>	<i>-A</i> <sub>1</sub>	<i>-A</i> <sub>1</sub>	<i>-A</i> <sub>1</sub>
<i>d</i> <sub>1</sub>	<i>d</i> <sub>2</sub>	<i>-d</i> <sub>3</sub>	<i>-d</i> <sub>4</sub>	<i>-d</i> <sub>5</sub>	<i>-d</i> <sub>6</sub>	<i>-d</i> <sub>7</sub>	<i>d</i> <sub>8</sub>	<i>d</i> <sub>9</sub>	<i>d</i> <sub>10</sub>	<i>d</i> <sub>11</sub>	<i>-d</i> <sub>12</sub>	<i>-d</i> <sub>13</sub>	<i>-d</i> <sub>14</sub>	<i>-d</i> <sub>15</sub>	<i>-d</i> <sub>16</sub>	<i>d</i> <sub>16</sub>	<i>d</i> <sub>16</sub>	<i>d</i> <sub>16</sub>	<i>d</i> <sub>16</sub>
<i>i</i>	<i>j</i>	<i>-j</i>	<i>-i</i>	<i>-i</i>	<i>-j</i>	<i>j</i>	<i>i</i>	<i>i</i>	<i>j</i>	<i>-j</i>	<i>-i</i>	<i>-i</i>	<i>-j</i>	<i>j</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>
<i>e</i> <sub>1</sub>	<i>e</i> <sub>2</sub>	<i>-e</i> <sub>3</sub>	<i>-e</i> <sub>4</sub>	<i>-e</i> <sub>5</sub>	<i>e</i> <sub>6</sub>	<i>e</i> <sub>7</sub>	<i>e</i> <sub>8</sub>	<i>e</i> <sub>9</sub>	<i>-e</i> <sub>10</sub>	<i>-e</i> <sub>11</sub>	<i>-e</i> <sub>12</sub>	<i>e</i> <sub>13</sub>	<i>e</i> <sub>14</sub>	<i>e</i> <sub>15</sub>	<i>e</i> <sub>16</sub>	<i>-e</i> <sub>16</sub>	<i>-e</i> <sub>16</sub>	<i>-e</i> <sub>16</sub>	<i>-e</i> <sub>16</sub>
<i>B</i> <sub>1</sub>	<i>B</i> <sub>2</sub>	<i>-B</i> <sub>3</sub>	<i>-B</i> <sub>4</sub>	<i>-B</i> <sub>5</sub>	<i>B</i> <sub>6</sub>	<i>B</i> <sub>7</sub>	<i>B</i> <sub>8</sub>	<i>-B</i> <sub>8</sub>	<i>-B</i> <sub>7</sub>	<i>-B</i> <sub>6</sub>	<i>B</i> <sub>5</sub>	<i>B</i> <sub>4</sub>	<i>B</i> <sub>3</sub>	<i>-B</i> <sub>2</sub>	<i>-B</i> <sub>1</sub>	<i>-B</i> <sub>1</sub>	<i>-B</i> <sub>1</sub>	<i>-B</i> <sub>1</sub>	<i>-B</i> <sub>1</sub>
<i>f</i> <sub>1</sub>	<i>-f</i> <sub>2</sub>	<i>-f</i> <sub>3</sub>	<i>-f</i> <sub>4</sub>	<i>f</i> <sub>5</sub>	<i>f</i> <sub>6</sub>	<i>f</i> <sub>7</sub>	<i>-f</i> <sub>8</sub>	<i>-f</i> <sub>9</sub>	<i>-f</i> <sub>10</sub>	<i>f</i> <sub>11</sub>	<i>f</i> <sub>12</sub>	<i>f</i> <sub>13</sub>	<i>-f</i> <sub>14</sub>	<i>-f</i> <sub>15</sub>	<i>-f</i> <sub>16</sub>	<i>f</i> <sub>16</sub>	<i>f</i> <sub>16</sub>	<i>f</i> <sub>16</sub>	<i>f</i> <sub>16</sub>
<i>e</i>	<i>-f</i>	<i>-g</i>	<i>-h</i>	<i>h</i>	<i>g</i>	<i>f</i>	<i>-e</i>	<i>-e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>-h</i>	<i>-g</i>	<i>-f</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>	<i>e</i>
<i>g</i> <sub>1</sub>	<i>-g</i> <sub>2</sub>	<i>-g</i> <sub>3</sub>	<i>-g</i> <sub>4</sub>	<i>g</i> <sub>5</sub>	<i>g</i> <sub>6</sub>	<i>-g</i> <sub>7</sub>	<i>-g</i> <sub>8</sub>	<i>-g</i> <sub>9</sub>	<i>g</i> <sub>10</sub>	<i>g</i> <sub>11</sub>	<i>-g</i> <sub>12</sub>	<i>-g</i> <sub>13</sub>	<i>-g</i> <sub>14</sub>	<i>g</i> <sub>15</sub>	<i>g</i> <sub>16</sub>	<i>-g</i> <sub>16</sub>	<i>-g</i> <sub>16</sub>	<i>-g</i> <sub>16</sub>	<i>-g</i> <sub>16</sub>
<i>C</i> <sub>1</sub>	<i>-C</i> <sub>2</sub>	<i>-C</i> <sub>3</sub>	<i>C</i> <sub>4</sub>	<i>C</i> <sub>5</sub>	<i>C</i> <sub>6</sub>	<i>-C</i> <sub>7</sub>	<i>-C</i> <sub>8</sub>	<i>C</i> <sub>8</sub>	<i>C</i> <sub>7</sub>	<i>-C</i> <sub>6</sub>	<i>-C</i> <sub>5</sub>	<i>-C</i> <sub>4</sub>	<i>C</i> <sub>3</sub>	<i>C</i> <sub>2</sub>	<i>-C</i> <sub>1</sub>	<i>-C</i> <sub>1</sub>	<i>-C</i> <sub>1</sub>	<i>-C</i> <sub>1</sub>	<i>-C</i> <sub>1</sub>
<i>h</i> <sub>1</sub>	<i>-h</i> <sub>2</sub>	<i>-h</i> <sub>3</sub>	<i>h</i> <sub>4</sub>	<i>h</i> <sub>5</sub>	<i>-h</i> <sub>5</sub>	<i>-h</i> <sub>7</sub>	<i>h</i> <sub>8</sub>	<i>h</i> <sub>9</sub>	<i>h</i> <sub>10</sub>	<i>-h</i> <sub>11</sub>	<i>-h</i> <sub>12</sub>	<i>h</i> <sub>13</sub>	<i>h</i> <sub>14</sub>	<i>-h</i> <sub>15</sub>	<i>-h</i> <sub>16</sub>	<i>h</i> <sub>16</sub>	<i>h</i> <sub>16</sub>	<i>h</i> <sub>16</sub>	<i>h</i> <sub>16</sub>
<i>k</i>	<i>-k</i>	<i>-k</i>	<i>k</i>	<i>k</i>	<i>-k</i>	<i>-k</i>	<i>k</i>	<i>k</i>	<i>-k</i>	<i>-k</i>	<i>k</i>	<i>k</i>	<i>-k</i>	<i>-k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>	<i>k</i>
<i>h</i> <sub>16</sub>	<i>-h</i> <sub>15</sub>	<i>-h</i> <sub>14</sub>	<i>h</i> <sub>13</sub>	<i>h</i> <sub>12</sub>	<i>-h</i> <sub>11</sub>	<i>-h</i> <sub>10</sub>	<i>h</i> <sub>9</sub>	<i>-h</i> <sub>8</sub>	<i>-h</i> <sub>7</sub>	<i>h</i> <sub>6</sub>	<i>h</i> <sub>5</sub>	<i>-h</i> <sub>4</sub>	<i>-h</i> <sub>3</sub>	<i>h</i> <sub>2</sub>	<i>h</i> <sub>1</sub>	<i>-h</i> <sub>1</sub>	<i>-h</i> <sub>1</sub>	<i>-h</i> <sub>1</sub>	<i>-h</i> <sub>1</sub>
<i>C</i> <sub>8</sub>	<i>-C</i> <sub>7</sub>	<i>-C</i> <sub>6</sub>	<i>C</i> <sub>5</sub>	<i>-C</i> <sub>4</sub>	<i>-C</i> <sub>3</sub>	<i>C</i> <sub>2</sub>	<i>C</i> <sub>1</sub>	<i>-C</i> <sub>1</sub>	<i>-C</i> <sub>2</sub>	<i>C</i> <sub>3</sub>	<i>C</i> <sub>4</sub>	<i>-C</i> <sub>5</sub>	<i>C</i> <sub>6</sub>	<i>C</i> <sub>7</sub>	<i>-C</i> <sub>8</sub>	<i>-C</i> <sub>8</sub>	<i>-C</i> <sub>8</sub>	<i>-C</i> <sub>8</sub>	<i>-C</i> <sub>8</sub>
<i>g</i> <sub>16</sub>	<i>-g</i> <sub>15</sub>	<i>-g</i> <sub>14</sub>	<i>g</i> <sub>13</sub>	<i>-g</i> <sub>12</sub>	<i>-g</i> <sub>11</sub>	<i>g</i> <sub>10</sub>	<i>g</i> <sub>9</sub>	<i>-g</i> <sub>8</sub>	<i>g</i> <sub>7</sub>	<i>g</i> <sub>6</sub>	<i>-g</i> <sub>5</sub>	<i>-g</i> <sub>4</sub>	<i>g</i> <sub>3</sub>	<i>-g</i> <sub>2</sub>	<i>-g</i> <sub>1</sub>	<i>g</i> <sub>1</sub>	<i>g</i> <sub>1</sub>	<i>g</i> <sub>1</sub>	<i>g</i> <sub>1</sub>
<i>h</i>	<i>-g</i>	<i>f</i>	<i>e</i>	<i>-e</i>	<i>-f</i>	<i>g</i>	<i>-h</i>	<i>-h</i>	<i>g</i>	<i>-f</i>	<i>-e</i>	<i>e</i>	<i>f</i>	<i>-g</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>
<i>f</i> <sub>16</sub>	<i>-f</i> <sub>15</sub>	<i>f</i> <sub>14</sub>	<i>f</i> <sub>13</sub>	<i>-f</i> <sub>12</sub>	<i>f</i> <sub>11</sub>	<i>f</i> <sub>10</sub>	<i>-f</i> <sub>9</sub>	<i>f</i> <sub>8</sub>	<i>f</i> <sub>7</sub>	<i>-f</i> <sub>6</sub>	<i>f</i> <sub>5</sub>	<i>f</i> <sub>4</sub>	<i>-f</i> <sub>3</sub>	<i>f</i> <sub>2</sub>	<i>f</i> <sub>1</sub>	<i>-f</i> <sub>1</sub>	<i>-f</i> <sub>1</sub>	<i>-f</i> <sub>1</sub>	<i>-f</i> <sub>1</sub>
<i>B</i> <sub>8</sub>	<i>-B</i> <sub>7</sub>	<i>B</i> <sub>6</sub>	<i>B</i> <sub>5</sub>	<i>-B</i> <sub>4</sub>	<i>B</i> <sub>3</sub>	<i>B</i> <sub>2</sub>	<i>-B</i> <sub>1</sub>	<i>-B</i> <sub>2</sub>	<i>-B</i> <sub>3</sub>	<i>B</i> <sub>4</sub>	<i>-B</i> <sub>5</sub>	<i>-B</i> <sub>6</sub>	<i>B</i> <sub>7</sub>	<i>-B</i> <sub>8</sub>	<i>-B</i> <sub>8</sub>	<i>-B</i> <sub>8</sub>	<i>-B</i> <sub>8</sub>	<i>-B</i> <sub>8</sub>	<i>-B</i> <sub>8</sub>
<i>e</i> <sub>16</sub>	<i>-e</i> <sub>15</sub>	<i>e</i> <sub>14</sub>	<i>-e</i> <sub>13</sub>	<i>-e</i> <sub>12</sub>	<i>e</i> <sub>11</sub>	<i>-e</i> <sub>10</sub>	<i>-e</i> <sub>9</sub>	<i>e</i> <sub>8</sub>	<i>-e</i> <sub>7</sub>	<i>e</i> <sub>6</sub>	<i>e</i> <sub>5</sub>	<i>-e</i> <sub>4</sub>	<i>e</i> <sub>3</sub>	<i>e</i> <sub>2</sub>	<i>-e</i> <sub>1</sub>	<i>e</i> <sub>1</sub>	<i>e</i> <sub>1</sub>	<i>e</i> <sub>1</sub>	<i>e</i> <sub>1</sub>
<i>j</i>	<i>-i</i>	<i>i</i>	<i>-j</i>	<i>-j</i>	<i>i</i>	<i>-i</i>	<i>j</i>	<i>j</i>	<i>-i</i>	<i>i</i>	<i>-j</i>	<i>-j</i>	<i>i</i>	<i>-i</i>	<i>j</i>	<i>j</i>	<i>j</i>	<i>j</i>	<i>j</i>
<i>d</i> <sub>16</sub>	<i>-d</i> <sub>15</sub>	<i>d</i> <sub>14</sub>	<i>-d</i> <sub>13</sub>	<i>d</i> <sub>12</sub>	<i>d</i> <sub>11</sub>	<i>-d</i> <sub>10</sub>	<i>d</i> <sub>9</sub>	<i>-d</i> <sub>8</sub>	<i>-d</i> <sub>7</sub>	<i>d</i> <sub>6</sub>	<i>-d</i> <sub>5</sub>	<i>d</i> <sub>4</sub>	<i>-d</i> <sub>3</sub>	<i>-d</i> <sub>2</sub>	<i>d</i> <sub>1</sub>	<i>-d</i> <sub>1</sub>	<i>-d</i> <sub>1</sub>	<i>-d</i> <sub>1</sub>	<i>-d</i> <sub>1</sub>
<i>A</i> <sub>8</sub>	<i>-A</i> <sub>7</sub>	<i>A</i> <sub>6</sub>	<i>-A</i> <sub>5</sub>	<i>A</i> <sub>4</sub>	<i>A</i> <sub>3</sub>	<i>-A</i> <sub>2</sub>	<i>A</i> <sub>1</sub>	<i>-A</i> <sub>1</sub>	<i>A</i> <sub>2</sub>	<i>-A</i> <sub>3</sub>	<i>-A</i> <sub>4</sub>	<i>A</i> <sub>5</sub>	<i>-A</i> <sub>6</sub>	<i>A</i> <sub>7</sub>	<i>-A</i> <sub>8</sub>	<i>-A</i> <sub>8</sub>	<i>-A</i> <sub>8</sub>	<i>-A</i> <sub>8</sub>	<i>-A</i> <sub>8</sub>
<i>b</i> <sub>16</sub>	<i>-b</i> <sub>15</sub>	<i>b</i> <sub>14</sub>	<i>-b</i> <sub>13</sub>	<i>b</i> <sub>12</sub>	<i>-b</i> <sub>11</sub>	<i>-b</i> <sub>10</sub>	<i>b</i> <sub>9</sub>	<i>-b</i> <sub>8</sub>	<i>b</i> <sub>7</sub>	<i>-b</i> <sub>6</sub>	<i>b</i> <sub>5</sub>	<i>-b</i> <sub>4</sub>	<i>-b</i> <sub>3</sub>	<i>b</i> <sub>2</sub>	<i>-b</i> <sub>1</sub>	<i>b</i> <sub>1</sub>	<i>b</i> <sub>1</sub>	<i>b</i> <sub>1</sub>	<i>b</i> <sub>1</sub>
<i>d</i>	<i>-c</i>	<i>b</i>	<i>-a</i>	<i>-b</i>	<i>c</i>	<i>-d</i>	<i>-d</i>	<i>c</i>	<i>-b</i>	<i>a</i>	<i>-a</i>	<i>b</i>	<i>-c</i>	<i>d</i>	<i>-c</i>	<i>d</i>	<i>-c</i>	<i>-c</i>	<i>-c</i>
<i>a</i> <sub>16</sub>	<i>-a</i> <sub>15</sub>	<i>a</i> <sub>14</sub>	<i>-a</i> <sub>13</sub>	<i>a</i> <sub>12</sub>	<i>-a</i> <sub>11</sub>	<i>a</i> <sub>10</sub>	<i>-a</i> <sub>9</sub>	<i>a</i> <sub>8</sub>	<i>-a</i> <sub>7</sub>	<i>a</i> <sub>6</sub>	<i>a</i> <sub>5</sub>	<i>-a</i> <sub>4</sub>	<i>a</i> <sub>3</sub>	<i>-a</i> <sub>2</sub>	<i>a</i> <sub>1</sub>	<i>-a</i> <sub>1</sub>	<i>-a</i> <sub>1</sub>	<i>-a</i> <sub>1</sub>	<i>-a</i> <sub>1</sub>
<i>H</i>	<i>-G</i>	<i>F</i>	<i>-E</i>	<i>D</i>	<i>-C</i>	<i>B</i>	<i>-A</i>	<i>A</i>	<i>-B</i>	<i>C</i>	<i>-D</i>	<i>E</i>	<i>-F</i>	<i>G</i>	<i>-H</i>	<i>-H</i>	<i>-H</i>	<i>-H</i>	<i>-H</i>
<i>c</i> <sub>31</sub>	<i>-c</i> <sub>29</sub>	<i>c</i> <sub>27</sub>	<i>-c</i> <sub>25</sub>	<i>c</i> <sub>23</sub>	<i>-c</i> <sub>21</sub>	<i>c</i> <sub>19</sub>	<i>-c</i> <sub>17</sub>	<i>c</i> <sub>15</sub>	<i>-c</i> <sub>13</sub>	<i>c</i> <sub>11</sub>	<i>-c</i> <sub>9</sub>	<i>c</i> <sub>7</sub>	<i>-c</i> <sub>5</sub>	<i>c</i> <sub>3</sub>	<i>-c</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>c</i> <sub>1</sub>	<i>c</i> <sub>1</sub>

(12)

где  $c_1 = c_3$ ,  $c_{2i-1} > c_{2i+1}$ ,  $i = \overline{2, 15}$ ,  $A > B > C > D > E > F > G > H$ ,  $a > b > c > d$ ,  $g > e > h > f$ ,  $i > j$ .

Элементы матрицы  $C_{32}$  предложенного однонормового масштабированного ЦКП  $c_{2i-1}$ ,  $a_i$ ,  $b_i$ ,  $d_i$ ,  $e_i$ ,  $f_i$ ,  $g_i$  и  $h_i$  представлены в табл. 1, а элементы матрицы  $C_{16}$  однонормового масштабированного ЦКП порядка 16 из работы [13], которые являются элементами матрицы  $C_{32}$ , принимают следующие значения:  $A = 180$ ,  $B = 172$ ,  $C = 160$ ,  $D = 140$ ,  $E = 116$ ,  $F = 84$ ,  $G = 56$ ,  $H = 20$ ,  $a = 180$ ,  $b = 152$ ,  $c = 96$ ,  $d = 36$ ,  $e = 152$ ,  $f = 36$ ,  $g = 176$ ,  $h = 100$ ,  $i = 172$ ,  $j = 56$ ,  $k = 128$ . Значения элементов  $A_i$ ,  $B_i$ ,  $C_i$  ( $i = \overline{1, 8}$ ) представлены в табл. 2.

**Таблица 1**

Элементы матрицы $C_{32}$	Значения элементов матрицы $C_{32}$ при $i$															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$c_{2i-1}$	180	180	176	168	164	156	144	136	120	108	92	80	64	44	24	8
$a_i$	206	133	110	111	53	8	49	127	167	139	148	204	176	100	65	31
$b_i$	202	111	50	34	128	133	147	182	105	8	64	168	205	140	98	50
$d_i$	170	89	24	132	182	143	47	82	162	174	111	13	122	180	158	59
$e_i$	164	44	106	180	123	27	154	171	61	94	179	135	2	147	176	74
$f_i$	126	13	190	120	20	198	155	38	143	139	69	148	85	90	210	76
$g_i$	119	68	210	32	129	156	90	147	20	188	140	75	143	10	199	87
$h_i$	133	110	156	73	173	45	179	11	182	25	176	59	164	93	145	122

Таблица 2

Элементы матрицы $C_{16}$	Значения элементов матрицы $C_{16}$ при $i$							
	1	2	3	4	5	6	7	8
$A_i$	174	116	18	87	159	180	139	48
$B_i$	160	11	138	173	55	117	181	85
$C_i$	141	82	172	16	181	53	161	113

Отметим, что элементы матрицы  $R_{16}$  согласно (6) принимают следующие значения:  $r_1 = r_2 = 45$ ,  $r_3 = 44$ ,  $r_4 = 42$ ,  $r_5 = 41$ ,  $r_6 = 39$ ,  $r_7 = 36$ ,  $r_8 = 34$ ,  $s_1 = 2$ ,  $s_2 = 6$ ,  $s_3 = 11$ ,  $s_4 = 16$ ,  $s_5 = 20$ ,  $s_6 = 23$ ,  $s_7 = 27$ ,  $s_8 = 30$ , а элементы матрицы  $T_8^*$  согласно (9) принимают значения:  $a' = 5$ ,  $b' = 3$ ,  $c' = 2$ ,  $d' = 1$ ,  $e' = 5$ ,  $f' = 2$ ,  $g' = 4$ . Следует отметить, что квадраты норм базисных векторов матрицы  $C_{32}$  однонормового масштабированного ЦКП приближаются к числу степени два:  $q_i = \|c_i\|^2 = 524288 \pm \Delta_i$  ( $\Delta_i$  (%) — отклонение параметра  $q_i$ ),  $\Delta_i = 0,01-0,43$ ,  $i = 1, 3, 5, \dots, 15$ ;  $\Delta_{2i} = 0,11-0,28$ ,  $i = 1, 3, 5, 7$ ;  $\Delta_{4i} = 0,15-0,7$ .

Таким образом, предложенное целочисленное преобразование имеет такие же свойства, как и преобразование, принятое в стандарте H.265: коэффициенты преобразования представляют 8 бит; ширина накопительного сумматора для матричного умножения не превышает 32 бит; симметричность/антисимметричность соответствует ДКП; коэффициенты масштабированного преобразования достаточно близки к ДКП; базисные векторы почти ортогональны; одинаковая схема квантования и деквантования для преобразований всех размеров; множители зависят от параметра квантования QR [2] и сдвиги зависят только от  $\log_2 N$ , где  $N$  — размерность преобразования; коэффициенты квантования могут представлять 16 бит.

**АЛГОРИТМ БЫСТРОГО ПРЯМОГО ЦКП ПОРЯДКА 32**

Матрица  $T_8^*$  согласно (9) может быть факторизовано представлена как произведение трех матриц:

$$T_8^* = T_{8,3} T_{8,2} T_{8,1}, \tag{13}$$

где  $T_{8,i}$  —  $i$ -е,  $i = \overline{1, 3}$ , фактор-матрицы  $8 \times 8$  алгоритма быстрого прямого ЦКП из работы [17]:

$$T_{8,1} = H_8^*, T_{8,2} = \text{diag} [H_4^*, T_4], T_{8,3} = [T_2, Q_2, \tilde{R}_4]. \tag{14}$$

При этом

$$H_{2k}^* = \begin{bmatrix} I_k & \bar{I}_k \\ \bar{I}_k & -I_k \end{bmatrix}, k = 2, 3, T_4 = \begin{bmatrix} 1 & 0 & q & 1 \\ 0 & -1 & -1 & q \\ q & 1 & -1 & 0 \\ -1 & q & 0 & 1 \end{bmatrix}, \tilde{R}_4 = \begin{bmatrix} r & & s \\ & r & -s \\ & s & r \\ -s & & r \end{bmatrix},$$

$$T_2 = \begin{bmatrix} g' & g' \\ g' & -g' \end{bmatrix}, Q_2 = \begin{bmatrix} -e' & f' \\ f' & e' \end{bmatrix}, \text{ где } q \geq 1, r > s.$$

Матрица  $Q_{16}$  на основании (8) и с учетом (13), (14) может быть факторизовано представлена как произведение пяти матриц:

$$Q_{16} = T_{16,5} T_{16,4} T_{16,3} T_{16,2} T_{16,1}, \tag{15}$$

где  $T_{16,i}$  —  $i$ -е,  $i = \overline{1, 5}$ , фактор-матрицы  $16 \times 16$  ядра ЦКП-IV:

$$T_{16,1} = \bar{R}_{16}; T_{16,k+1} = I_2 \otimes T_{8,k}, k = \overline{1, 3}; T_{16,5} = H_{16}^0. \tag{16}$$

Матрица  $C_{16}^*$  согласно (9) может быть факторизовано представлена как произведение пяти матриц:

$$C_{16}^* = C_{16,5}C_{16,4}C_{16,3}C_{16,2}C_{16,1}, \quad (17)$$

где  $C_{16,i}$  —  $i$ -е,  $i=\overline{1,5}$ , фактор-матрицы  $16 \times 16$  алгоритма быстрого прямого ЦКП из работы [13].

Таким образом, матрица  $C_{32}^*$  на основании (3), (15), (16) и с учетом алгоритма быстрого прямого ЦКП порядка 16 согласно (17) может быть факторизовано представлена как произведение шести матриц:

$$C_{32}^* = C_6 C_5 C_4 C_3 C_2 C_1, \quad (18)$$

где  $C_i$  —  $i$ -е,  $i=\overline{1,6}$ , фактор-матрицы  $32 \times 32$  алгоритма быстрого прямого ЦКП, предложенного в [11, 12]:

$$C_1 = H_{32}^*; C_2 = \text{diag} [C_{16,1}, \overline{R}_{16}]; \\ C_{2+k} = \text{diag} [C_{16,k+1}, I_2 \otimes T_{8,k}], k = \overline{1,3}; C_6 = \text{diag} [C_{16,5}, H_{16}^0]. \quad (19)$$

#### АЛГОРИТМ БЫСТРОГО ОБРАТНОГО ЦКП ПОРЯДКА 32

Матрицу  $C_{16i}$  обратного ЦКП порядка 16 можно получить транспонированием:

$$C_{32i} = C_{32}^{*T} / k. \quad (20)$$

Матрица  $C_{32i}$  на основании (20), (18), (19) и с учетом симметричности фактор-матрицы ( $H_{32}^{*T} = H_{32}^*$ ) может быть факторизовано представлена произведением шести матриц:

$$C_{32i} = C_{1i} C_{2i}^T C_{3i}^T C_{4i}^T C_{5i}^T C_{6i}^T, \quad (21)$$

где  $C_{ki}^T$  —  $k$ -е,  $k=\overline{1,6}$ , транспонированные фактор-матрицы  $32 \times 32$  алгоритма быстрого обратного ЦКП:

$$C_{1i} = H_{32}^*, C_{2i}^T = \text{diag} [C_{16i,1}^T, \overline{R}_{16i}^T]; \\ C_{(2+k)i}^T = \text{diag} [C_{16i,k+1}^T, I_2 \otimes T_{8i,k}^T], k = \overline{1,3}; C_{6i}^T = \text{diag} [C_{16i,5}^T, H_{16i}^{0T}]; \\ \overline{R}_{16i}^T = \overline{R}_{16}^T / k', k' = k / g'; \tilde{R}_{4i}^T = \tilde{R}_4^T / g', T_{2i}^T = T_{2i} / g' = H_2, \\ Q_{2i}^T = Q_{2i} = Q_2 / g', T_{4i}^T = T_4^T.$$

При этом

$$\overline{R}_{16}^T = \begin{bmatrix} s_1 & -r_1 \\ & s_2 & & & & & & & & & & & & & & & & & & & r_2 \\ & & s_3 & & & & & & & & & & & & & & & & & & -r_3 \\ & & & s_4 & & & & & & & & & & & & & & & & & r_4 \\ & & & & s_5 & & & & & & & & & & & & & & & & -r_5 \\ & & & & & s_6 & & & & & & & & & & & & & & & r_6 \\ & & & & & & s_7 & & & & & & & & & & & & & & -r_7 \\ & & & & & & & s_8 & r_8 & & & & & & & & & & & & \\ & & & & & & & r_8 & -s_8 & & & & & & & & & & & & \\ & & & & & & & & r_7 & s_7 & & & & & & & & & & & \\ & & & & & & & & & r_6 & -s_6 & & & & & & & & & & \\ & & & & & & & & & & r_5 & s_5 & & & & & & & & & \\ & & & & & & & & & & & r_4 & -s_4 & & & & & & & & \\ & & & & & & & & & & & & r_3 & s_3 & & & & & & & \\ & & & & & & & & & & & & & r_2 & -s_2 & & & & & & \\ r_1 & s_1 \end{bmatrix}, \quad (23)$$



$$H_{16}^{0T} = \left[ \begin{array}{c|c} I_8 & G_4^0 \\ \hline -G_4^0 & \bar{I}_4 \end{array} \right], \tilde{R}_4^T = \begin{bmatrix} r & -s \\ r & s \\ -s & r \\ s & r \end{bmatrix}, T_4^T = \begin{bmatrix} 1 & 0 & q & -1 \\ 0 & -1 & 1 & q \\ q & -1 & -1 & 0 \\ 1 & q & 0 & 1 \end{bmatrix}.$$

Для однонормового обратного преобразования матрица  $H_{16}^{0T}$  содержит ненулевые элементы 1 и  $\pm P_{ki} / 2^{m_k}$ ,  $k=1, 3$ ,  $P_{ki}$  — целые.

#### РЕАЛИЗАЦИЯ БЫСТРОГО ЦЕЛОЧИСЛЕННОГО ПРЕОБРАЗОВАНИЯ БЕЗ МНОЖИТЕЛЕЙ

При реализации быстрых целочисленных преобразований используются операции типа «бабочка» («butterfly»), где выполняются парные умножения, которые могут быть реализованы с выполнением операций только сдвига и сложения, а в некоторых случаях (с целью уменьшения вычислительной сложности) и операций умножения.

Для реализации предложенного 1D 32-точечного целочисленного косинусного преобразования используются в основном операции сдвига, сложения и некоторые операции умножения.

В табл. 3 представлена схема выполнения специальных умножений, используемых в операциях «бабочка», для реализации предложенного 1D 32-точечного быстрого обратного целочисленного преобразования.

Таблица 3

Значения множителей		Алгоритм выполнения операций $y = s_i * x$ ; $z = r_i * x$	Вычислительная сложность операций			Число используемых операций
			Сложение	Сдвиг	Умножение	
$s_1 = 2 / 32$	$r_1 = 45 / 32$	$y = x \gg 4$ ; $x_1 = x + (x \gg 1)$ ; $z = x_1 - (x_1 \gg 4)$	2	3	—	2
$s_2 = 6 / 32$	$r_2 = 45 / 32$	$x_1 = x + (x \gg 1)$ ; $y = x_1 \gg 3$ ; $z = x_1 - (y \gg 1)$	2	3	—	2
$s_3 = 11 / 32$	$r_3 = 44 / 32$	$x_1 = x - (x \gg 2)$ ; $z = x + (x_1 \gg 1)$ ; $y = z \gg 2$	2	3	—	2
$s_4 = 16 / 32$	$r_4 = 42 / 32$	$x_1 = x + (x \gg 2)$ ; $z = x + (x_1 \gg 2)$ ; $y = x \gg 1$	2	3	—	2
$s_5 = 20 / 32$	$r_5 = 41 / 32$	$x_1 = x + (x \gg 2)$ ; $y = x_1 \gg 1$ ; $z = x_1 + (x \gg 5)$	2	3	—	2
$s_6 = 23 / 32$	$r_6 = 39 / 32$	$x_1 = x - (x \gg 3)$ ; $z = x + (x_1 \gg 2)$ ; $y = z - (x \gg 1)$	3	3	—	2
$s_7 = 27 / 32$	$r_7 = 36 / 32$	$z = x + (x \gg 3)$ ; $y = z - (z \gg 2)$	2	2	—	2
$s_8 = 30 / 32$	$r_8 = 34 / 32$	$x_1 = x \gg 4$ ; $y = x - x_1$ ; $z = x + x_1$	2	1	—	2
$p_{1i} = p_1 / 2^{m_1}$	0	$x_1 = p_1 * x$ ; $y = x_1 \gg m_1$ ; $z = 0$	—	1	1	4
$p_{2i} = p_2 / 2^{m_2}$	0	$x_1 = p_2 * x$ ; $y = x_1 \gg m_2$ ; $z = 0$	—	1	1	4
$p_{3i} = p_3 / 2^{m_3}$	0	$x_1 = p_3 * x$ ; $y = x_1 \gg m_3$ ; $z = 0$	—	1	1	4
Всего		—	34	56	14	—

Таблица 4

Операции и характеристики	Оценка сложности 2D обратных преобразований 32 × 32					
	Предложенного	Известных			Сравнительный анализ характеристик преобразований	
		Н.265 из [6, 7]	из [9]	из [10]	относительно [9] (и [10])	относительно Н.265
Умножение	1408	21888	5888	5568	В 4,18 (и в 3,95) раза меньше	В 15,55 раз меньше
Сложение	17280	23808	11904	14144	—	—
Сложение + сдвиг	25088	25856	11904	14656	На 107,5 % (и на 71,2 %) больше сложенных	На 3,1 % меньше сложенных
Число итераций для 1D	6	5	7	8	На 1 (и на 2) итерацию меньше	На 1 итерацию больше
Память для квантования	—	—	+	—	—	—
Число битов элемента матрицы	9	8	11	14	На 2 (и на 5) бита меньше	На 1 бит больше
Общее уменьшение	—				В 3,11 (и 3,24) раза меньше	В 15,6 раза меньше

Для реализации 1D 16-точечного быстрого обратного целочисленного преобразования, предложенного в работе [13], требуется выполнить 8 операций умножения, 106 операций сложения и 50 операций сдвига. Для реализации 1D 8-точечного быстрого обратного целочисленного преобразования  $T_{8i}^*$  согласно (13), (14) требуется выполнить 34 операции сложения и 8 операций сдвига [17]. Всего для реализации предложенного 1D 32-точечного быстрого обратного целочисленного преобразования требуется выполнить 22 операции умножения, 270 операций сложения и 122 операции сдвига.

Вычислительная сложность предложенного, известных [9, 10] и принятого в стандарте Н.265 [6, 7] 2D 32-точечных целочисленных обратных преобразований приведена в табл. 4.

#### ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ И ИХ АНАЛИЗ

На рис. 1 представлены исходные изображения классов А и В для тестирования. В табл. 5 даны экспериментальные результаты эффективности кодирования по характеристике стандартной количественной оценки искажений PSNR (дБ) для сжатых тестовых изображений класса А (рис. 1, а, б) с разрешающей способностью 2560×1600 пикселей и класса В (рис. 1, в, г) — 1920×1056 пикселей при нормальном (22–37) диапазоне QP и значениях (37, 42) высокого диапазона QP для предложенного 2D преобразования с блоками 32×32. Эти результаты представляют разность на основе предложенного преобразования и на основе преобразования Н.265. Диапазоны параметра QP низкий (1–13), нормальный (22–37) и высокий (36–51) определены в [18].

Результаты эффективности кодирования по критерию (в уровнях яркости) оценки среднеквадратического отклонения (СКО) разности входного и восстановленного изображений для предложенного 2D преобразования с блоками

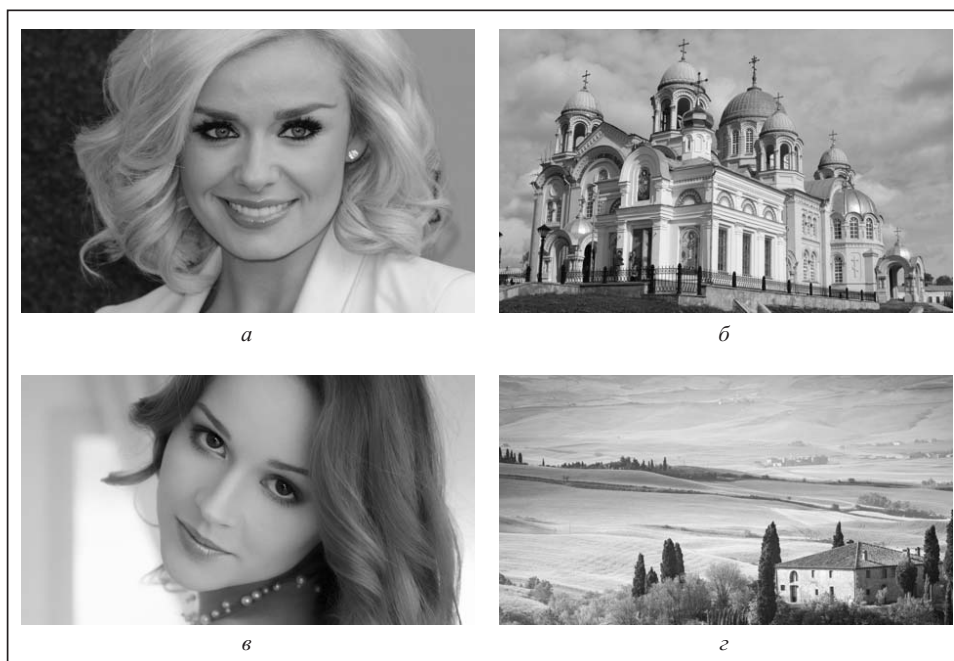


Рис. 1. Исходные изображения класса А: фото 1, размер  $2560 \times 1600$  (а); храм, размер  $2560 \times 1600$  (б); и класса В: фото 2, размер  $1920 \times 1056$  (в); пейзаж, размер  $1920 \times 1056$  (г)

$32 \times 32$ , представляющие разность на основе преобразования H.265 и на основе предложенного преобразования приведены в табл. 6.

Экспериментальные результаты кодирования по характеристике коэффициента сжатия  $K:1$  для предложенного 2D преобразования с блоками  $32 \times 32$  приведены в табл. 7. Результаты эффективности кодирования по характеристике коэффициента сжатия  $K$  (в %), представляющие разницу на основе предложенного преобразования и на основе преобразования H.265, приведены в табл. 8.

В таблицах даны средние значения экспериментальных результатов эффективности кодирования по характеристикам PSNR, СКО и коэффициенту сжатия  $K$  по четырем тестовым изображениям классов А и В.

Предложенное целочисленное косинусное преобразование порядка 32 по сравнению с преобразованием в стандарте H.265 по характеристике искажения PSNR для четырех тестовых изображений классов А и В понижает среднее значение на  $0,085\text{--}0,128$  дБ, а по характеристике СКО повышает среднее значение на  $0,094\text{--}0,99$  уровней яркости. При этом среднее значение коэффициента сжатия  $K$  уменьшается на  $0,018 \div 0,44$  %. Согласно принятого Комитетом MPEG используемого субъективного порога  $\text{PSNR} = 0,5$  дБ при принятии кодовой оптимизации считается, что увеличение или уменьшение на эту величину будет заметно визуально [19], а при  $\text{PSNR} < 0,5$  дБ — зрительно не ощущается. Таким образом, понижение наибольшего среднего значения PSNR на  $0,128$  дБ будет зрительно незаметно, т.е. сохраняется визуальное качество изображения.

Следует заметить, что высококоррелированное (ВК) изображение класса А (см. рис.1, а) имеет в  $2,77\text{--}5,47$  раза больший коэффициент сжатия для заданного параметра квантования QP, чем низкокоррелированное (НК) изображение того же класса (рис. 1, б; табл. 7). Изображение ВК класса В (рис. 1, в) имеет в  $2,38\text{--}4,12$  раза больший коэффициент сжатия для заданного QP, чем НК изображение того же класса (рис. 1, г). При этом среднее значение коэффициента сжатия ВК изображений классов А и В составляет в  $2,57\text{--}4,79$  раза больше, чем для случая НК изображений этих классов.

Таблица 5

Класс	Изображение с блоками 32 × 32	Результаты эффективности кодирования по характеристике PSNR (дБ) для QP				
		22	27	32	37	42
А 2560 × 1600	Фото 1	-0,14	-0,15	-0,15	-0,11	-0,06
	Храм	-0,22	-0,18	-0,16	-0,16	-0,14
В 1920 × 1056	Фото 2	-0,07	-0,06	-0,08	-0,08	-0,05
	Пейзаж	-0,08	-0,05	-0,07	-0,09	-0,09
Среднее значение		-0,128	-0,110	-0,115	-0,110	-0,085

Таблица 6

Класс	Изображение с блоками 32 × 32	Результаты эффективности кодирования по характеристике СКО (в уровнях яркости) для QP				
		22	27	32	37	42
А 2560 × 1600	Фото 1	-0,065	-0,075	-0,196	-0,234	-0,221
	Храм	-0,164	-0,281	-0,543	-1,120	-2,057
В 1920 × 1056	Фото 2	-0,052	-0,096	-0,182	-0,260	-0,249
	Пейзаж	-0,094	-0,145	-0,344	-0,819	-1,433
Среднее значение		-0,094	-0,149	-0,316	-0,608	-0,990

Таблица 7

Класс	Изображение с блоками 32 × 32	Результаты кодирования по коэффициенту сжатия K : 1 для QP				
		22	27	32	37	42
А 2560 × 1600	Фото 1	11,88	23,76	45,37	91,16	176,99
	Храм	4,28	6,31	9,69	17,01	32,36
В 1920 × 1056	Фото 2	6,16	14,82	35,59	81,57	176,99
	Пейзаж	2,59	4,73	9,11	19,35	43,23
Среднее значение		6,228	12,405	24,940	52,273	107,393

Таблица 8

Класс	Изображение с блоками 32 × 32	Результаты эффективности кодирования по характеристике коэффициента сжатия K (%) для QP				
		22	27	32	37	42
А 2560 × 1600	Фото 1	-0,328	-0,152	-0,041	-0,001	-0,003
	Храм	-0,739	-0,554	-0,338	-0,159	-0,050
В 1920 × 1056	Фото 2	-0,257	-0,147	-0,046	0,0	-0,001
	Пейзаж	-0,436	-0,363	-0,245	-0,100	-0,025
Среднее значение		-0,440	-0,304	-0,168	-0,065	-0,018

#### ЗАКЛЮЧЕНИЕ

Предложен метод построения однонормового целочисленного косинусного преобразования высокого порядка. Построено однонормовое целочисленное преобразование порядка 32 и разработаны его быстрые алгоритмы низкой вычислительной сложности, в 3,24 раза меньшей, чем в известных алгоритмах, и в 15,6 раза меньшей, чем в стандарте H.265. Это преобразование не только

наиболее близкое к ДКП, но также имеет превосходные характеристики кодирования: среднее значение понижения по характеристике искажений PSNR составляет 0,085–0,128 дБ, а среднее значение повышения по характеристике СКО составляет 0,094–0,99 уровней яркости. При этом среднее значение коэффициента сжатия уменьшается на 0,018–0,44%. Данное преобразование обеспечивает такое же визуальное качество, а также и качество по характеристикам PSNR, СКО и коэффициенту сжатия, как в стандарте H.265.

В настоящей статье предложен подход разделения изображений по характеристике коэффициента сжатия на ВК и НК изображения. Поэтому при тестировании классы А и В изображений представляют ВК и НК изображения. Анализ результатов (см. табл. 7) показал, что ВК изображения классов А и В (см. рис. 1, а, в) имеют в среднем в 2,57–4,79 раза больший коэффициент сжатия для заданного параметра квантования QP, чем НК изображения (см. рис. 1, б, г). Таким образом, разработанное целочисленное преобразование порядка 32 может быть использовано для улучшения стандарта H.265 в целях увеличения быстродействия и уменьшения вычислительных и энергетических затрат.

#### СПИСОК ЛИТЕРАТУРЫ

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Москва: Техносфера, 2005. 1072 с.
2. Ричардсон Я. Видеокодирование. H.264 и MPEG-4 — стандарты нового поколения. Москва: Техносфера, 2005. 368 с.
3. ITU-T Rec. H.264 | ISO/IEC 14496-10: 2009. Information technology — Coding of audio-visual objects. Part 10: Advanced Video Coding, 2009.
4. SMPTE standard 421 M-2006: VC-1 compressed video bitstream format and decoding process, 2006.
5. PRC National Standard (AVS Working Group) GB/T 20090.2-2006. Information technology — Advanced coding of audio and video. Part 2: Video, Chinese AVS standard, 2006.
6. ITU-T Rec. H.265 | ISO/IEC 23008-2: 2013. Information technology — High efficiency coding and media delivery in heterogeneous environments. Part 2: High efficiency video coding, 2013.
7. Fuldseth A., Bjøntegaard G., Budagavi M., Sze V. CE10: Core transform design for HEVC. Doc. JCTVC-G495. Geneva. CH, Nov. 2011.
8. Joshi R., Reznik Y., Karczewicz M. Efficient large size transforms for high performance video coding. *Proc. SPIE Appl. of Digital Image Process. XXXIII*. 2010. Vol. 7798. P. 779831–1–7.
9. Joshi R., Reznik Y., Sole J., Karczewicz M. Efficient 16 and 32-point transforms. Doc. JCTVC- D256. Daegu. KR, Jan. 2011.
10. Alshina E., Alshin A., Kim I.-K., Topiwala P. CE10: Full-factorized core transform proposal by Samsung/FastVDO. Doc. JCTVC-F251. Torino. Italy, July 2011.
11. Гнатів Л.О. Метод побудови швидких цілочисельних косинусних перетворень великої розмірності для високоефективного кодування зображень і відео. *Пр. Міжнар. конф. «Питання оптимізації обчислень (ПОО-2013)»*. Україна, Крим. смт. Кацивелі, 30 вересня 2013. С. 66–67.
12. Гнатів Л.О., Луц В.К. Цілочисельне косинусне перетворення порядку 32 низької складності для високоефективного кодування зображень і відео. *Пр. Міжнар. конф. «Питання оптимізації обчислень (ПОО-2013)»*. Україна, Крим. смт. Кацивелі, 30 вересня 2013. С. 68–69.
13. Гнатів Л.А. Целочисленные косинусные преобразования для высокоэффективного кодирования изображений и видео. *Кибернетика и системный анализ*. 2016. Т. 52, № 5. С. 161–176.
14. Гнатів Л.А. Метод построения простого целочисленного косинусного преобразования большой размерности для кодирования изображений и видео с высоким разрешением. *Кибернетика и системный анализ*. 2016. Т. 52, № 6. С. 143–155.

15. Ugur K., Andersson K., Fuldseth A. at el. High performance low complexity video coding and the emerging HEVC standard. *IEEE Trans. Circuits Syst. Video Technol.* 2012. Vol. 20, N 12. P. 1688–1697.
16. Шевчук Б.М., Задірака В.К., Гнатів Л.О., Фраєр С.В. Технологія багатофункціональної обробки і передачі інформації в моніторингових мережах. Київ: Наук. думка. 2010. 378 с.
17. Гнатів Л.А., Целочисленные косинусные преобразования: методы построения новых быстрых преобразований порядка 8,16 и их применение. *Кибернетика и системный анализ.* 2014. Т. 50, № 6. С. 104–121.
18. Fuldseth A., Bjøntegaard G., Sadafale M., Budagavi M. Transform design for HEVC with 16 bit intermediate data representation. Doc. JCTVC-E243. Geneva. CH, Mar. 2011.
19. Сэломон Д. Сжатие данных, изображений и звука. Москва: Техносфера, 2004. 368 с.

Надійшла до редакції 08.12.2016

### Л.О. Гнатів

#### ЦІЛОЧИСЕЛЬНЕ КОСИНУСНЕ ПЕРЕТВОРЕННЯ ВИСОКОГО ПОРЯДКУ: МЕТОД ПОБУДОВИ І ШВИДКІ АЛГОРИТМИ ДЛЯ КОДУВАННЯ ЗОБРАЖЕННЯ І ВІДЕО ВИСОКОЇ РОЗДІЛЬНОЇ ЗДАТНОСТІ

**Анотація.** Запропоновано матричний метод побудови однонормового цілочисельного косинусного перетворення високого порядку. Побудовано однонормове цілочисельне перетворення порядку 32 і розроблено його швидкі алгоритми низької обчислювальної складності (яка менша в 3,24 рази, ніж у відомих алгоритмах, та в 15,6 разів, ніж у стандарті H.265), і вони потребують тільки цілочисельних операцій. Це перетворення не тільки найближче до дискретного косинусного перетворення, але також має відмінні характеристики кодування.

**Ключові слова:** дискретне косинусне перетворення, цілочисельне косинусне перетворення, масштабоване перетворення, факторизація, швидке перетворення, ефективність кодування, обчислювальна складність, коефіцієнт стиснення, відеокодування, H.264, H.265, AVS.

### L.O. Hnativ

#### HIGH-ORDER INTEGER COSINE TRANSFORM: CONSTRUCTION METHOD AND FAST ALGORITHMS FOR HIGH-RESOLUTION IMAGE AND VIDEO CODING

**Abstract.** Matrix method of constructing one-norm high integer cosine transform is proposed. The one-norm order-32 integer transform is constructed and its fast algorithms of low computational complexity are developed, whose computational complexity is 3.24 times less than that in the well-known algorithms and is 15.6 times less than one in standard H.265 and requires integer operations only. This transform is not only very close to the discrete cosine transform, but also has excellent coding performance.

**Keywords:** discrete cosine transform, integer cosine transform, scaled transform, factorization, fast transform, coding gain, computational complexity, compression ratio, video coding, H.264, H.265, AVS.

### Гнатів Лев Алексеевич,

кандидат техн. наук, старший научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: levhnativ@gmail.com.