

§ 5 КОМПЬЮТЕРНАЯ ГРАФИКА, ОБРАБОТКА ИЗОБРАЖЕНИЙ И РАСПОЗНАВАНИЕ ОБРАЗОВ

Арзуманян Р.В., Сухинов А.И.

ФАКТОРИЗАЦИЯ МАТРИЦЫ ОБРАТНОГО ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ 8x8 СТАНДАРТА КОДИРОВАНИЯ ВИДЕО HEVC И БЫСТРЫЙ АЛГОРИТМ НА ЕЁ ОСНОВЕ

Аннотация: Предметом данной работы является разработка алгоритма быстрого обратного дискретного преобразования размера 8x8 стандарта кодирования видео ITU-T H265 (HEVC). Рассматриваются отличия в структуре матрицы обратного преобразования от матрицы обратного дискретного косинусного преобразования, а также подходы, которые могут быть применены для факторизации упомянутой матрицы. Дается оценка числа операций, необходимых для выполнения преобразования и проводится численный эксперимент для проверки эффективности разработанного алгоритма с точки зрения скорости его исполнения на центральном процессоре (CPU). Метод проведения работы – теоретическое исследование с последующим проведением численного эксперимента со сбором интересующей информации и анализом результатов. Для проведения численного эксперимента была написана программа на языке Си, реализующая стандартный алгоритм обратного преобразования (непосредственное перемножение матрицы преобразования и вектора коэффициентов) и предлагаемый быстрый алгоритм обратного преобразования, разработанный в теоретической части данной работы. Проведено сравнение результатов производительности. Новизна работы заключается в том, что в ней предложен ранее неизвестный алгоритм быстрого обратного преобразования 8x8 стандарта HEVC и схема факторизации матрицы преобразования. По сравнению с предшествующими работами, предложенный алгоритм требует меньшего числа арифметических операций, а значит, может быть исполнен за меньшее время. В данной статье были сделаны выводы относительно возможности реализации быстрого обратного преобразования стандарта HEVC, предложена схема факторизации матрицы указанного преобразования размера 8x8 и разработан быстрый алгоритм обратного преобразования на основе найденной схемы факторизации.

Ключевые слова: HEVC, быстрый алгоритм, факторизация матрицы, обратное дискретное преобразование, сжатие видео, кодек, сжатие с потерями, анализ алгоритма, программный кодек, аппаратный кодек

Abstract: The subject of the present research is the development of the algorithm for the fast inverse discrete transform of the 8x8 ITU-T H265 (HEVC) video coding standard. The authors of the article analyze differences between the structures of the inverse transform matrix and inverse discrete cosine transform matrix as well as approaches that may be applied to factorization of the aforesaid matrix. They also provide an evaluation of a number of operations necessary to perform transformation. The authors conduct a numerical experiment to prove the efficiency of the developed algorithm from the point of view of the speed of performance on the central processing unit (CPU). The research method used by the authors is the theoretical analysis and numerical experiment including collection of relevant information and analysis of results. To conduct the numerical experiment the authors have written a C-language program that executes a standard algorithm of inverse transformation (direct multiplication of the transformation matrix and vector coefficients) and the described algorithm for the fast inverse discrete transform presented in the theoretical part of this work. The authors give a comparison of performance results. The novelty of the work lies in the fact that authors offered a previously unknown algorithm for the fast inverse discrete transform of the 8x8 ITU-T H265 (HEVC) video coding standard along with a scheme of factorization of the transformation matrix. Compared with previous researches, the proposed algorithm requires less number of arithmetic operations, and thus it requires less execution time. In this article, the conclusions were made on how to implement the fast inverse discrete transform of the HEVC standard. The authors the presented a scheme of factorization of the 8x8 matrix of this transformation and developed a fast algorithm based on inverse transform factorization scheme.

Keywords: lossy compression, codec, video compression, inverse discrete transform, matrix factorization, fast algorithm, HEVC, algorithm analysis, software codec, hardware codec

Введение

Дискретное преобразование является одним из ключевых компонентов в компрессии видео на протяжении последних 25 лет совместно с компенсацией движения. Созданный в 2013г. организацией ITU-T стандарт кодирования видео H265 (HEVC) так же задействует алгоритм обратного дискретного преобразования для сжатия видеосигнала. Основные подходы, применённые в процессе разработки алгоритмов данного преобразования освещены в работах [1]-[5]. Обратное преобразование HEVC не обладает свойствами симметрии, присущими преобразованию Фурье, так как является целочисленным. Более того, некоторые коэффициенты матрицы преобразования были подобраны вручную [2]. Так, авторы стандарта HEVC подчёркивают, что дискретное преобразование не обладает тригонометрическими свойствами, которые могли бы уменьшить число операций, необходимое для проведения преобразования [2, с. 1031]. Обратное преобразование для стандарта HEVC создавалось в расчёте на баланс между вычислительной сложностью и степенью компрессии [2, с. 1030]. Для простоты реализации кодеков, преобразование

обладает следующими свойствами:

1. Коэффициенты матрицы являются 8-ми битными числами.
2. Для промежуточных вычислений достаточно 16-ти битных регистров.
3. Регистры-аккумуляторы могут иметь длину менее 32 бит.
4. Возможно частичное рекурсивное разложение подматрицы, состоящей из чётных строк матрицы преобразования.

Однако, алгоритмы полной факторизации (разложения) матрицы преобразования в стандарте не предусмотрены. Это обусловлено тем, что подматрица, состоящая из нечётных строк матрицы преобразования, обладает свойством антисимметрии [2, с. 1036-1038].

Актуальность работы

Несмотря на то, что алгоритм преобразования разработан таким образом, чтобы его реализации могли иметь частично вложенную структуру, в случае высокого визуального качества сжатого видеосигнала, кадр с адаптивным разбиением может содержать большое количество блоков обратного преобразования малого размера, его декодирование представляет сложную вычислительную задачу. В этой связи, актуальной является задача разработки быстрых алгоритмов обратного преобразования для стандарта HEVC. Данная работа осложнена тем, что преобразование не обладает всеми свойствами дискретного косинусного преобразования, и классические алгоритмы быстрых преобразований Фурье [6], алгоритм Чена [7] и алгоритмы на основе метода Кули-Тьюки[8] в данном случае неприменимы. Фактически, разработка такого алгоритма сводится к нахождению схемы факторизации подматрицы коэффициентов преобразования, состоящей из нечётных строк матрицы. На данный момент в работе [9] предложен один из вариантов осуществления факторизации, однако он разработан для аппаратных кодеков, и не оптимален с точки зрения количества выполняемых арифметических действий для программных реализаций. Авторами работы [9] в КНР поданы 2 патентные заявки на способ осуществления быстрого обратного дискретного преобразования для стандарта ITU-T H265 (HEVC) размера 8x8.

Цель работы

Целью данной работы является поиск схемы факторизации подматрицы матрицы обратного дискретного преобразования HEVC, состоящей из нечётных строк, для разработки быстрого алгоритма обратного преобразования.

Постановка задачи

Обратное одномерное дискретное преобразование HEVC можно записать в виде:

$$X_k = \sum_{n=0}^{N-1} x_n A_{kn}$$

Коэффициенты x_k хранятся в битовом потоке, а коэффициенты матрицы A заданы не аналитически, а таблично, и определены стандартом HEVC. Стандарт определяет преобразования размером 4×4 , 8×8 , 16×16 и 32×32 . В данной работе рассматривается преобразование 8×8 , поэтому приведём матрицу коэффициентов для него:

$$A = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix}$$

Как можно заметить, коэффициенты матрицы A обладают свойством зеркальной симметрии. Коэффициенты правой части матрицы симметричны коэффициентам левой части с точностью до знака. Эта особенность заложена в стандарт HEVC для облегчения разработки быстрых реализаций. Такой подход носит название «Partial butterfly». Рассмотрим его более подробно. Так как правая часть матрицы зеркально симметрична левой с точностью до знака, выделим из матрицы A размером 8×8 матрицу A' размером 8×4 :

$$A' = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 \\ 83 & 36 & -36 & -83 \\ 75 & -18 & -89 & -50 \\ 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 \\ 36 & -83 & 83 & -36 \\ 18 & -50 & 75 & -89 \end{bmatrix}$$

Далее, разделим матрицу A' на две матрицы размером 4×4 :

$$A_4 = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 18 & -50 & 75 & -89 \end{bmatrix} R_4 = \begin{bmatrix} 89 & 75 & 50 & 18 \\ 75 & -18 & -89 & -50 \\ 50 & -89 & 18 & 75 \\ 18 & -50 & 75 & -89 \end{bmatrix}$$

Легко заметить, что матрица A_4 получена из чётных строк матрицы A' , а матрица R_4 - из нечётных строк. Матрица A_4 обладает теми же свойствами, что и исходная матрица A - её можно рекурсивно раскладывать на матрицы меньшего размера для упрощения преобразования [10]. Матрица R_4 такими свойствами не обладает.

Разработка алгоритма

Предложенный Ченом алгоритм факторизации матрицы R_4 дискретного косинусного преобразования основан на том, что плотную матрицу размера $N \times N$ можно представить в виде произведения трёх разреженных матриц размера $N \times N$. За счёт регулярной и разреженной структуры матриц, составляющих факторизацию, операция прямого перемножения матриц заменяется на последовательность более простых операций с частичной заменой умножений на сложения. Благодаря этому, достигается прирост производительности. Предполагаемая форма записи матрицы R_4 :

$$R_4 = M_1 \cdot M_2 \cdot M_3 \quad (1)$$

$$M_1 = \begin{bmatrix} \sin\left(\frac{\pi}{16}\right) & 0 & 0 & \cos\left(\frac{\pi}{16}\right) \\ 0 & \sin\left(\frac{5\pi}{16}\right) & \cos\left(\frac{5\pi}{16}\right) & 0 \\ 0 & -\sin\left(\frac{3\pi}{16}\right) & \cos\left(\frac{3\pi}{16}\right) & 0 \\ -\sin\left(\frac{7\pi}{16}\right) & 0 & 0 & \cos\left(\frac{7\pi}{16}\right) \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & 0 \\ 0 & \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Данный метод факторизации необходимо модифицировать для того, чтобы он стал применим к целочисленному преобразованию. Матрицу M_1 составим в соответствие с методом Чена — по главной и побочной диагоналям расположим элементы из первого и четвертого столбцов:

$$M_1 = \begin{bmatrix} 18 & 0 & 0 & 89 \\ 0 & 75 & 50 & 0 \\ 0 & -50 & 75 & 0 \\ -89 & 0 & 0 & 18 \end{bmatrix}$$

Матрица M_2 в оригинальном алгоритме состоит из целых чисел, поэтому оставим её

без изменений. Матрицу M_3 запишем в виде:

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a & c & 0 \\ 0 & b & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

В таком случае, произведение трёх матриц будет равно:

$$M_1 \cdot M_2 \cdot M_3 = \begin{bmatrix} 18 & 19a + 89c & 18b + 89d & 89 \\ 75 & -75a + 50c & -75b + 50d & 50 \\ -50 & 50a - 75c & 50b - 75d & 75 \\ -89 & -89a + 18c & 89b + 18d & 18 \end{bmatrix}$$

Не существует целочисленных значений a, b, c, d , при которых матрицу R_4 можно было бы представить в виде (1).

Второй возможный метод декомпозиции состоит в том, чтобы в матрице M_3 ненулевые элементы округлить до ближайшего целого, а M_2 записать в виде:

$$M_2 = \begin{bmatrix} a & c & 0 & 0 \\ b & d & 0 & 0 \\ 0 & 0 & d & b \\ 0 & 0 & c & a \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Найдём произведение матриц M_1, M_2 , составив СЛАУ из матриц R_4, M_3 :

$$X = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} Y = \begin{bmatrix} 50 \\ 75 \\ 18 \\ -89 \\ -89 \\ -18 \\ 75 \\ -50 \end{bmatrix}$$

$$M_1 \cdot M_2 = \begin{bmatrix} 18 & 25/2 & 125/2 & 89 \\ 75 & 107/2 & -71/2 & 50 \\ -50 & 71/2 & -107/2 & 75 \\ -89 & -125/2 & 25/2 & 18 \end{bmatrix}$$

Если записать произведение M_1, M_2 в символьном виде, получим:

$$M_1 \cdot M_2 = \begin{bmatrix} 18a & 18c & 89c & 89a \\ 75b & 75d & 50d & 50b \\ -50b & -50d & 75d & 75b \\ -89a & -89c & 18c & 18a \end{bmatrix}$$

В этом случае так же не существует целочисленных значений a, b, c, d , при которых матрицу R_4 можно было бы записать в виде (1). Однако, если заменить коэффициент 89 на 90, то целочисленные значения могут быть найдены:

$$M_1 = \begin{bmatrix} 18 & 0 & 0 & 90 \\ 0 & 75 & 50 & 0 \\ 0 & -50 & 75 & 0 \\ -90 & 0 & 0 & 18 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 25/36 & 0 & 0 \\ 1 & -18/25 & 0 & 0 \\ 0 & 0 & -18/25 & 1 \\ 0 & 0 & 25/36 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_c = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Однако, в таком случае матрицу R_4 можно представить только в виде

$$R_4 = M_1 \cdot M_2 \cdot M_3 + M_c$$

Далее, необходимо избавиться от дробных коэффициентов в матрице M_2 , поскольку преобразование является целочисленным, и деление может привести к округлению с потерей точности. Матрицу M_1 можно представить в виде произведения двух матриц:

$$M_1 = K \cdot L$$

$$K = \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 3 & 2 & 0 \\ 0 & -2 & 3 & 0 \\ -5 & 0 & 0 & 1 \end{bmatrix} L = \begin{bmatrix} 18 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 18 \end{bmatrix}$$

Умножим матрицу L на матрицу M_2 и проведём череду простых преобразований:

$$M_1 \cdot M_2 = K \cdot (L \cdot M_2)$$

$$M_1 \cdot M_2 = \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 3 & 2 & 0 \\ 0 & -2 & 3 & 0 \\ -5 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 18 & 25/2 & 0 & 0 \\ 25 & -18 & 0 & 0 \\ 0 & 0 & -18 & 25 \\ 0 & 0 & 25/2 & 18 \end{bmatrix}$$

$$M_1 \cdot M_2 = \left(\begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 3 & 2 & 0 \\ 0 & -2 & 3 & 0 \\ -5 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 36 & 25 & 0 & 0 \\ 50 & -36 & 0 & 0 \\ 0 & 0 & -36 & 50 \\ 0 & 0 & 25 & 36 \end{bmatrix} \right) / 2$$

$$R_4 = \left(\begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 3 & 2 & 0 \\ 0 & -2 & 3 & 0 \\ -5 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 36 & 25 & 0 & 0 \\ 50 & -36 & 0 & 0 \\ 0 & 0 & -36 & 50 \\ 0 & 0 & 25 & 36 \end{bmatrix} \right) / 2 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Операцию деления на 2 можно реализовать простым битовым сдвигом. В данном случае потери значимых бит не произойдёт. Проиллюстрируем полученное разложение на рис. 1.

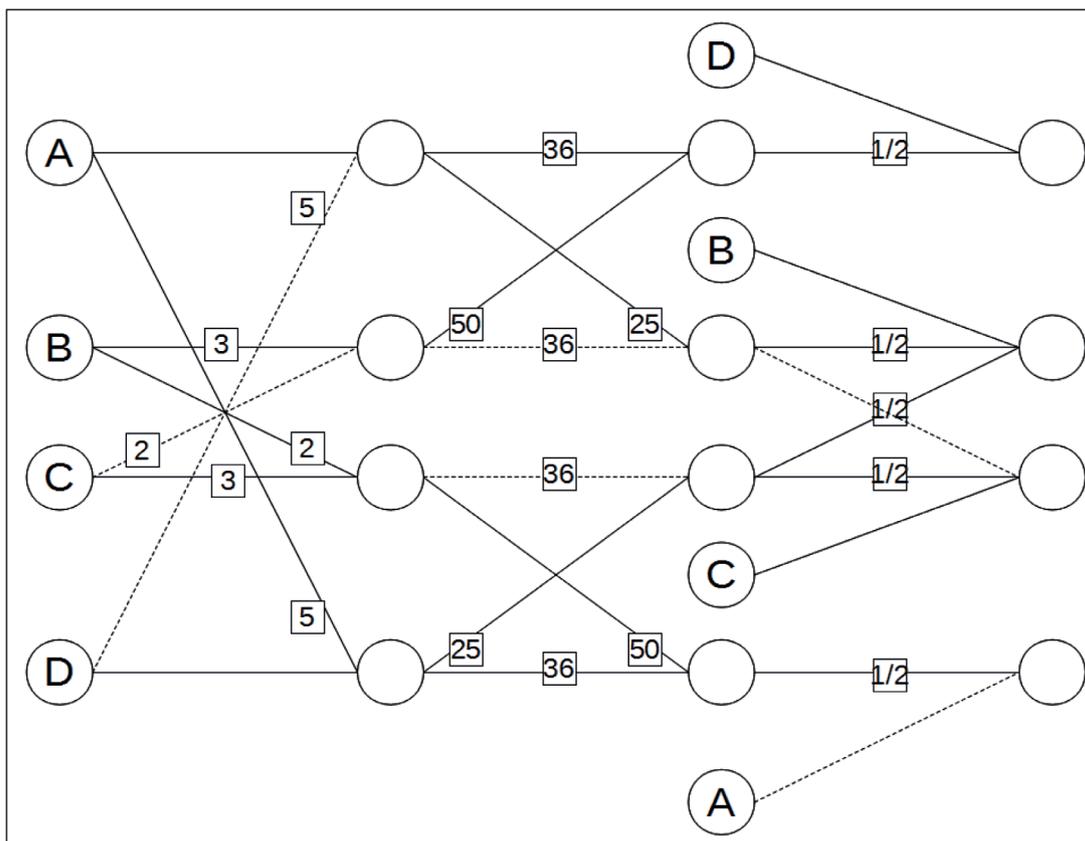


Рисунок 1: граф потока управления для разложенной матрицы R_4

Оценка результатов

Для оценки полученных в ходе работы результатов, подсчитаем количество и состав арифметических операций, необходимых для умножения подматрицы R_4 на входной вектор-столбец коэффициентов, полученных из битового потока:

Прямое перемножение	Предлагаемый алгоритм	Алгоритм [9]
12 сложений 16 умножений	4 битовых сдвига 14 сложений 12 умножений	4 битовых сдвига 14 сложений 14 умножений

Таблица 1: Сравнение числа и типа арифметических операций

Как видно из таблицы 1, предложенный алгоритм предполагает большее количество арифметических операций, однако сами операции проще, поскольку выполняется меньше умножений, которые заменены на более быстрые операции сложения и побитового сдвига. Для экспериментальной проверки полученного алгоритма, была написана программа на языке программирования Си, которая осуществляла обратное дискретное преобразование HEVC. В качестве тестовой системы использовался компьютер на базе процессора ARM. Его характеристики перечислены в таблице 2:

CPU	4 Cortex A-15 2Hz 4 Cortex A-7 1.3 Hz
RAM	2 Gb DDR3 800Mhz
OS	Ubuntu linux

Таблица 2: характеристики тестового компьютера

В качестве входных данных подавался массив из 16588800 (15.82 Мбайт) целых 2-х байтовых чисел. Были получены следующие результаты:

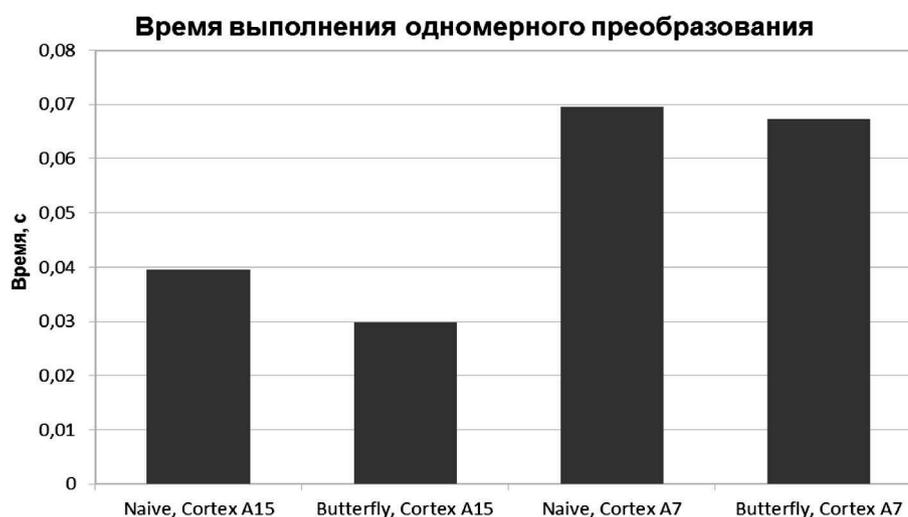


Рисунок 3: время выполнения одномерного преобразования.

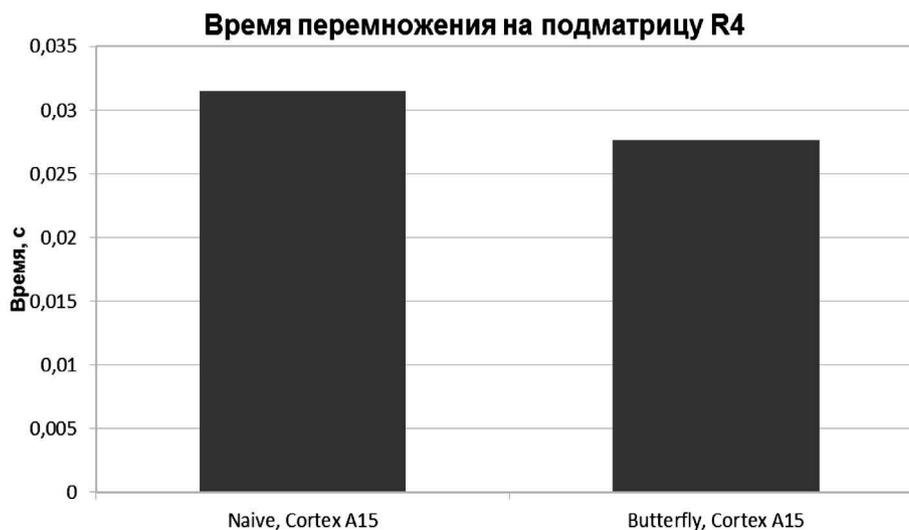


Рисунок 4: время перемножения на подматрицу R_4 .

На рисунках 3, 4 представлены результаты замеров времени осуществления обратного дискретного преобразования в целом, и результаты времени перемножения матрицы R_4 и столбца коэффициентов. Название столбца “Naive” соответствует прямому перемножению матрицы и вектора, название “Butterfly” – предложенному алгоритму. Как видно из полученных результатов, предложенный алгоритм факторизации подматрицы обеспечивает 24% прирост производительности при перемножении матрицы R_4 и столбца коэффициентов, и на 12% большую производительность при выполнении одномерного преобразования в целом.

Выводы

В данной работе был найден алгоритм факторизации подматрицы R_4 матрицы обратного дискретного преобразования стандарта кодирования ITU-T H265 размером 8x8. Предложенный алгоритм содержит меньшее количество умножений по сравнению с прямым перемножением матрицы на вектор за счёт использования большего числа сложений и битовых сдвигов. Реализация предложенного алгоритма на языке Си для процессора ARM обеспечивает на 24% большую производительность перемножения матрицы на вектор и 12% большую производительность одномерного преобразования в целом по сравнению с алгоритмом, предложенным в стандарте при сохранении побитовой идентичности результата. По сравнению с предыдущими работами, на 2 операции уменьшено число умножений.

Библиография :

1. Belghith F., Loukil H., Masmoudi N. Efficient Hardware Architecture of a Modified 2 - D Transform for the HEVC Standard // International Journal of Computer Science and Application. 2013. № 4. С. 59-69.
2. Budagavi M., Fuldseth A., Bjøntegaard G., Sze V., Sadafale M. Core Transform Design in the High Efficiency Video Coding (HEVC) Standard // IEEE Journal of Selected Topics in Signal Processing. 2013. № 6. С. 1029-1041.
3. Hung C.-Y., Landman P. Compact inverse discrete cosine transform circuit for MPEG video decoding // В кн.: IEEE SIPS. 1997. С. 364–373.
4. Park J.-S., Nam W.-J., Han S.-M., Lee S. 2-D Large Inverse Transform (16x16, 32x32) for HEVC (High Efficiency Video Coding) // Journal of semiconductor technology and science. 2012. № 12. С. 204-208.
5. Winken M., Helle P., Marpe D., Schwarz H., Wiegand T. Transform coding in the HEVC test model // В кн.: IEEE International Conference on Image Processing. 2011. С. 3693–3696.
6. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989. 448 с.
7. Chen W.-H., Harrison-Smith C., Fralick S. C. A Fast Computational Algorithm for the Discrete Cosine Transform // IEEE transactions on communications. 1977. № 9. С. 1004-1009.
8. Cooley J.W., Tukey J.W. An Algorithm for the Machine Calculation of Complex Fourier Series // Mathematics of Computation. 1965. № 90. С. 297-301.
9. Ma T., Liu C., Yibo F., Zeng X. A fast 8x8 IDCT algorithm for HEVC // В кн.: IEEE ASIC (ASICON). 2013. С. 203-208.
10. Budagavi M., Sze V. Unified forward+inverse transform architecture for HEVC // В кн.: IEEE International Conference on Image Processing. 2012. С. 209-212.

References:

1. Belghith F., Loukil H., Masmoudi N. Efficient Hardware Architecture of a Modified 2 - D Transform for the HEVC Standard // International Journal of Computer Science and Application. 2013. № 4. S. 59-69.
2. Budagavi M., Fuldseth A., Bjøntegaard G., Sze V., Sadafale M. Core Transform Design in the High Efficiency Video Coding (HEVC) Standard // IEEE Journal of Selected Topics in Signal Processing. 2013. № 6. S. 1029-1041.
3. Hung C.-Y., Landman P. Compact inverse discrete cosine transform circuit for MPEG video decoding // В кн.: IEEE SIPS. 1997. S. 364–373.
4. Park J.-S., Nam W.-J., Han S.-M., Lee S. 2-D Large Inverse Transform (16x16, 32x32) for HEVC (High Efficiency Video Coding) // Journal of semiconductor technology and science. 2012. № 12. S. 204-208.
5. Winken M., Helle P., Marpe D., Schwarz H., Wiegand T. Transform coding in the HEVC test model // В кн.: IEEE International Conference on Image Processing. 2011. S. 3693–3696.
6. Bleikhut R. Bystrye algoritmy tsifrovoi obrabotki signalov. M.: Mir, 1989. 448 s.
7. Chen W.-H., Harrison-Smith C., Fralick S. C. A Fast Computational Algorithm for the Discrete Cosine Transform // IEEE transactions on communications. 1977. № 9. S. 1004-1009.

8. Cooley J.W., Tukey J.W. An Algorithm for the Machine Calculation of Complex Fourier Series // Mathematics of Computation. 1965. № 90. S. 297-301.
9. Ma T., Liu C., Yibo F., Zeng X. A fast 8x8 IDCT algorithm for HEVC // V kn.: IEEE ASIC (ASICON). 2013. S. 203-208.
10. Budagavi M., Sze V. Unified forward+inverse transform architecture for HEVC // V kn.: IEEE International Conference on Image Processing. 2012. S. 209-212.