

УДК 519.6

WTT-РАЗЛОЖЕНИЕ ДЛЯ СЕМЕЙСТВ МАССИВОВ И ЕГО ПРИМЕНЕНИЕ ДЛЯ СЖАТИЯ ИЗОБРАЖЕНИЙ

П. В. Харюк¹, И. В. Оселедец²

Рассмотрено применение WTT-разложения (Wavelet Tensor Train) к семействам массивов на примере набора монохромных изображений. WTT-разложение использует алгебраический подход к построению адаптивного вейвлет-преобразования. Главный недостаток разложения заключается в необходимости хранения фильтров разложения для каждого отдельного массива. Предложен новый подход к применению этого разложения, основанный на построении единого фильтра для всего набора массивов.

Ключевые слова: вычислительные тензорные методы, вейвлет-преобразование, WTT-разложение, TT-разложение, сжатие данных.

1. Введение. Классические вейвлеты играют важную роль в анализе сигналов различной природы. Вейвлет-преобразования облегчают анализ сигналов и используются в различных прикладных вычислительных задачах, в том числе в задаче сжатия данных. Например, рассмотрим одномерный случай. Для заданного сигнала $f(t)$ непрерывное вейвлет-преобразование имеет вид

$$W(\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t)\psi^*\left(\frac{t-\tau}{s}\right) dt,$$

где ψ — материнский вейвлет, ψ^* — комплексное сопряжение к ψ , τ — параллельный перенос и s — масштаб. Заметим, что $\int_{-\infty}^{+\infty} \psi(t) dt = 0$.

В классической книге по вейвлет-преобразованию [1] в качестве материнского вейвлета $\psi(t)$ рассматривается представитель семейства функций вида

$$\psi^{a,b}(t) = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right),$$

где $b \in \mathbb{R}$, $a \in \mathbb{R}_+$ и $\psi(t)$ удовлетворяет условию допустимости $\int_0^{+\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$, $\Psi(\omega)$ — преобразование Фурье от $\psi(t)$.

Дискретное вейвлет-преобразование обычно определяется как последовательное применение линейных преобразований (называемых фильтрами) к частям исходного вектора. Например, это преобразование можно определить в виде следующего многошагового алгоритма.

На первом шаге сигнал параллельно пропускается через фильтр высоких частот и фильтр нижних частот с последующим прореживанием в два раза:

$$\hat{y}^{\text{low}}[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[n]g[n-k], \quad y^{\text{low}} = \hat{y}^{\text{low}} \downarrow 2,$$

$$\hat{y}^{\text{high}}[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[n]h[n-k], \quad y^{\text{high}} = \hat{y}^{\text{high}} \downarrow 2.$$

Последующие вычисления заключаются в последовательном применении первого шага к y^{low} , полученного на предыдущем уровне преобразования. Самым простым вейвлетом является вейвлет Хаара [2].

¹ Московский государственный университет им. М.В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, д. 1, стр. 52, 119992, Москва; студент, e-mail: hariyuki.pavel@gmail.com

² Институт вычислительной математики РАН, ул. Губкина, д. 8, 119333, Москва; ст. науч. сотр.; e-mail: ivan.oseledets@gmail.com

Таким образом, информация не теряется, однако коэффициенты разложения по “плохим” функциям оказываются малыми и получается псевдоразреженный вектор (т.е. большинство элементов преобразованного вектора малы). Одна из проблем дискретных вейвлет-преобразований состоит в том, что для каждого конкретного преобразования фильтры и последовательность их применения фиксированы и не адаптируются к конкретному сигналу. Другим классическим подходом к построению адаптивных вейвлет-преобразований является *лифтинговая схема* [3]

Дискретное вейвлет-преобразование используется в самых разнообразных прикладных вычислительных задачах, среди которых обработка изображений [4, 5], обработка данных в задачах нейрофизиологии [6–8] и др.

В работах [9, 10] предложено семейство преобразований, сочетающее в себе достоинства вейвлетов и свойство адаптивности. Такое преобразование было названо WTT-разложением (Wavelet Tensor Train). WTT-преобразование основано на TT-разложении (Tensor Train) [11]. Вейвлет-преобразование Хаара принадлежит классу WTT-преобразований при правильной перестановке элементов результирующего вектора. Для построения WTT-разложения нужно знать только дискретизированный сигнал: этого достаточно для его представления в псевдоразреженном виде в базисе некоторого набора фильтров. К очевидному недостатку WTT-преобразования относится необходимость дополнительного хранения фильтров разложения: мы выигрываем по разреженности полученного вектора, но требуется дополнительная память для хранения фильтров разложения. В этой связи возникает необходимость в разработке “гибридных” методов.

В настоящей статье предложен метод сжатия данных, основанный на WTT-разложении семейств массивов. Идея метода состоит в следующем: вместо вычисления и хранения фильтров для каждого отдельного массива производится вычисление некоторого общего фильтра для всего семейства. При этом подход к построению фильтров может быть разным — можно представить набор данных как один большой многомерный массив и на его основе вычислить общие фильтры, а можно выбрать часть данных из этого набора и поступить аналогично или ограничиться отдельным представителем как источником фильтров. В рамках нашей работы проведено исследование подходов к построению фильтров разложения на примере массива изображений.

2. WTT-разложение: основные сведения.

2.1. Квантизация. Основной идеей WTT-разложения является использование “виртуальных” индексов, которые появляются при работе с исходным массивом как многомерным. Например, рассмотрим $f(x) = \sin x$ на равномерной сетке из $n = n_1 \cdot n_2 \cdot \dots \cdot n_k$ элементов, введенной на интервале $[0, \pi]$. Вектор исходных данных имеет вид $(f(x_1), \dots, f(x_n)) = (a_1, \dots, a_n)$; представим его в виде многомерного массива размерности k с размерами $n_1 \times n_2 \times \dots \times n_k$. К полученному тензору можно применить тензорные разложения, в частности TT-разложение [11]. В общем случае исходный массив уже может быть тензором размерности l с размерами $n_1 \times n_2 \times \dots \times n_l$, и тогда “виртуальные” индексы можно ввести следующим образом. Пусть для $i = \overline{1, l}$ справедливо $n_i = \prod_{j=1}^{k_i} n_i^j$, $k_i \geq 1$; следовательно, исходный тензор можно представить в виде тензора большей размерности k размера $(\times_{j=1}^{k_1} n_1^j) \times \dots \times (\times_{j=1}^{k_l} n_l^j)$. Таким образом, мы провели процедуру “квантизации” исходного массива.

Определение. *Квантизация* — это представление исходного тензора $V(i) = V(i_1, \dots, i_d)$ размерности d с размерами $n_1 \times \dots \times n_d$, $i_p = \overline{1, n_p}$, $p = \overline{1, d}$, в виде тензора

$$\hat{V}(\hat{i}) = \hat{V}(\times_{j=1}^{l_1} k_1^j, \dots, \times_{j=1}^{l_d} k_d^j) = \hat{V}(k_1^1, \dots, k_1^{l_1}, \dots, k_d^1, \dots, k_d^{l_d}),$$

где $k_p^{l_p} = \overline{1, n_p^{l_p}}$, $\prod_{j=1}^{l_p} n_p^{l_p} = n_p$, $p = \overline{1, d}$.

Кроме того, индексы получившегося массива можно *переставлять*. Смысл данной операции заключается в том, чтобы соседние элементы исходного тензора оставались соседними в его квантизации. Это особенно актуально при построении QTT-представлений (Quantized Tensor Train) двумерных и трехмерных массивов.

Предположим, что исходный массив — двумерный размера $n^d \times n^d$. Проведем его квантизацию и рассмотрим развертку квантизации по k -му индексу:

$$\begin{aligned} A(\hat{i}, \hat{j}) &\rightarrow A(i, j) = A(i_1, \dots, i_d, j_1, \dots, j_d) = A(l_1, \dots, l_d), \\ A_k &= (l_1 l_2 \dots l_{k-1}, l_k l_{k+1} \dots l_d) = [A^{(1)}, A^{(2)}, \dots, A^{(p)}]. \end{aligned}$$

Для того чтобы блоки $A^{(t)}$ в развертке A_k были блоками исходного массива, произведем в квантизованном тензоре перестановку индексов по следующему правилу:

$$A(i, j) = A(i_1, \dots, i_d, j_1, \dots, j_d) \rightarrow B(i_1, j_1, i_2, j_2, \dots, i_d, j_d),$$

т.е. индексы (i_k, j_k) оказываются рядом.

2.2. QTT-разложение. Введение дополнительных размерностей позволяет применять методы аппроксимации тензоров. Будем использовать TT-разложение. *QTT-разложением* [12, 13] называется применение TT-разложения к квантизованному тензору. TT-разложение имеет вид [11]

$$A(i_1, \dots, i_d) = G_1(i_1)G_2(i_2) \dots G_d(i_d),$$

где $G_k(i_k)$ — матрица размерами $r_{k-1} \times r_k$, $r_0 = r_d = 1$; r_k — ранг матрицы развертки по k -му индексу: $r_k = \text{rank } A_k$.

WTT-разложение возникло как модификация QTT-разложения для случая, когда массив не представляется в QTT-формате с небольшими рангами. Принципиальное отличие WTT-разложения в том, что в этом разложении используется свойство псевдоразреженности результата работы с каждой локальной матрицей развертки. В качестве примера, иллюстрирующего вышесказанное, в работе [9] рассмотрен пример массива, представляющего собой монохромное изображение диска на сером фоне. При вычислении QTT-разложения такого массива TT-ранги r_k будут величиной порядка $O(n)$, где $n = 2^d$ — число элементов, а для хранения фильтров U_1, \dots, U_d потребуется память порядка $O(n^2)$. Более подробно с QTT-разложением можно ознакомиться в работах [13, 14].

2.3. WTT-разложение. В рамках настоящей статьи ограничимся формальным описанием процедуры построения разложения, подробности см. в [9, 10]. Описание приведено в алгоритмах 1 и 2 (рекурсивные).

Алгоритм 1. Построение фильтров разложения $(U_1, \dots, U_d) = F(a_k; n_k, \dots, n_d; r_{k-1}, \dots, r_{d-1})$

Исходные параметры: $k = \overline{1, d}$, размеры мод n_k, \dots, n_d , ранги r_{k-1}, \dots, r_{d-1} , вектор a_k (размером $r_{k-1} \cdot n_k \cdot \dots \cdot n_d$)

Результат. фильтры WTT-разложения U_1, \dots, U_d

$$A_k = \text{reshape}(a_k, r_{k-1} \cdot n_k \times n_{k+1} \cdot \dots \cdot n_d)$$

$$A_k = U_k S_k V_k^T (SVD), U_k = U_k^{r_{k-1} n_k \times r_{k-1} n_k}$$

if $k < d$ **then**

$$\hat{A}_k = U_k^T A_k = S_k V_k^T$$

$$A_{k+1} = \text{cut}(\hat{A}_k[1 : r_k, :])$$

$$a_{k+1} = \text{reshape}(A_{k+1}, r_k \cdot n_{k+1} \cdot \dots \cdot n_d \times 1)$$

$$(U_{k+1}, \dots, U_d) = F_{k+1}(a_{k+1}; n_{k+1}, \dots, n_d; r_k, \dots, r_{d-1})$$

endif

Алгоритм 2. WTT-преобразование $y_k = W_k(x_k; U_k, \dots, U_d; n_k, \dots, n_d; r_{k-1}, \dots, r_{d-1})$

Исходные параметры: уровень рекурсии k , $k = \overline{1, d}$, размеры мод n_k, \dots, n_d , ранги r_{k-1}, \dots, r_{d-1} , фильтры U_k, \dots, U_d , вектор x_k (размером $r_{k-1} \cdot n_k \cdot \dots \cdot n_d$)

Результат. преобразованный вектор y_k

if $k == d$ **then**

$$y_d = W_d(x_d; U_d; n_d; r_{d-1}) = U_d^T x_d$$

else

$$X_k = \text{reshape}(x_k, r_{k-1} \cdot n_k \times n_{k+1} \cdot \dots \cdot n_d)$$

$$\hat{X}_k = U_k^T X_k$$

$$X_{k+1} = \text{cut}(\hat{X}_k[1 : r_k, :]), Z_{k+1} = \text{cut}(\hat{X}_k[r_k + 1 : r_{k-1} n_k, :])$$

$$x_{k+1} = \text{reshape}(X_{k+1}, r_k \cdot n_{k+1} \cdot \dots \cdot n_d \times 1)$$

$$y_{k+1} = W_{k+1}(x_{k+1}; U_{k+1}, \dots, U_d; n_{k+1}, \dots, n_d; r_k \cdot \dots \cdot r_{d-1})$$

$$\hat{Y}_{k+1} = \text{reshape}(y_{k+1}, r_k \times n_{k+1} \cdot \dots \cdot n_d)$$

$$Y_k = \begin{pmatrix} \hat{Y}_{k+1} \\ Z_{k+1} \end{pmatrix}$$

$$y_k = \text{reshape}(Y_k, r_{k-1} \cdot n_k \cdot \dots \cdot n_d)$$

endif

Для построенного WTT-преобразования справедливы следующие утверждения.

Утверждение 1. WTT-преобразование является линейным и ортогональным.

Доказательство. WTT-преобразование является линейным в силу того, что все операции над исходным вектором сводятся к умножению частей вектора на матрицы (фильтры). Для доказательства ортогональности достаточно показать, что норма исходного вектора не изменяется для преобразованного

вектора. Матрицы (фильтры) U_k являются ортогональными, так как получены из сингулярного разложения. На каждом шаге алгоритма части вектора умножаются на указанные фильтры; следовательно, WTT-преобразование ортогонально.

Утверждение 2. Сложность WTT-преобразования имеет порядок $O(dnr^2)$, где $r = \max r_k$, n — общее число элементов вектора и d — размерность составленного из элементов вектора тензора.

Доказательство. На k -м шаге алгоритма требуется перемножение двух матриц размерами $r_{k-1}n_k \times r_{k-1}n_k$ и $r_{k-1} \cdot n_k \times n_{k+1} \dots n_d$, что потребует $r_{k-1}^2 \prod_{l=k}^d n_l \leq \max_{i=1, \dots, d} r_i \cdot \prod_{k=1}^d n_k = n \cdot r^2$ операций. Всего шагов d , откуда следует доказываемое утверждение.

На рис. 1 иллюстрируется изменение квантизированного изображения после первого и второго шагов алгоритма построения WTT-разложения. Изображение квантизировалось в блоки по 2×2 пиксела.



Рис. 1: Исходное изображение и его изменение на первых двух шагах WTT-преобразования

3. Численные эксперименты.

3.1. Алгоритм построения WTT-разложения для набора однородных массивов. WTT-преобразование определяется фильтрами разложения. Допустим, мы имеем некоторое множество однородных массивов. Под однородностью будем понимать соответствие следующим требованиям для каждой пары $A(i) = A(i_1, \dots, i_{d_A})$ и $B(j) = B(j_1, \dots, j_{d_B})$ из множества:

1. $d_A = d_B = d$ (эквивалентность размерностей);
2. $i_p, j_p = \overline{1, n_p}$ и $p = \overline{1, d}$ (эквивалентность размеров).

Наша цель состоит в том, чтобы для каждого представителя указанного множества применялись единые фильтры для построения разложения. Первый и самый простой подход заключается в выборе единичного источника фильтров разложения. Наиболее естественным будет выбор некоторого характерного представителя множества, имеющего повторяющиеся из массива в массив области.

В нашей работе предложен отличный от указанного выше подход, основанный на вычислении “эффективного фильтра” для однородного набора данных. Идея метода (и это основной результат настоящей статьи!) заключается в том, что исходное множество интерпретируется как тензор большей размерности, для которого строится свое WTT-разложение. Алгоритм 3 описывает процедуру построения таких фильтров.

Алгоритм 3. Построение общих фильтров разложения для семейства однородных массивов $(U_1, \dots, U_d) = \hat{F}(a_1(i), \dots, a_p(i); n_1, \dots, n_d; r_0, \dots, r_{d-1})$

Исходные параметры: размеры мод n_1, \dots, n_d ; ранги r_0, \dots, r_{d-1} ; тензоры $a_1(i), \dots, a_p(i)$ (размерами $n_1 \times \dots \times n_d$ каждый)

Результат. фильтры WTT-разложения U_1, \dots, U_d

$$\{a_1(i), \dots, a_p(i)\} \Rightarrow A(l, i), l = \overline{1, p}$$

$$A(l, i) \Rightarrow A(i, l)$$

$$A = A.reshape(1, A.size)$$

$$\{U_1, \dots, U_d, U_{d+1}\} = F(A, n_1, \dots, n_d; r_0, \dots, r_{d-1}) \text{ [см. алгоритм 1]}$$

$$\{U_1, \dots, U_d, U_{d+1}\} \Rightarrow \{U_1, \dots, U_d\}$$

Выбор подхода к построению фильтров WTT-разложения следует осуществлять на основании свойств исходной задачи. Назовем набор фильтров разложения, полученных на основании единственного представителя, частным фильтром, а на основании всех представителей — общим.

3.2. Тестовые данные и критерии эффективности работы алгоритма. Из предложенных подходов к построению фильтров разложения изучим вопрос эффективности первого и второго вариантов: соответственно, нашей задачей является изучение эффективности WTT-преобразования в зависимости от того, используется ли полный общий фильтр или частный. Для этого определим набор сравнительных характеристик и проведем исследование эффективности преобразования в зависимости от источника фильтров.

В качестве исследуемых характеристик возьмем стандартные: Compress Ratio (CR) и Peak Signal-to-Noise Ratio (PSNR). Напомним, что WTT-разложение представляется в виде

$$w = Wv,$$

где $W = W(U_1, \dots, U_d)$ — WTT-преобразование на основе U_1, \dots, U_d , v — исходный массив данных и w — псевдоразреженный массив.

Характеристика CR определяется как отношение числа ненулевых элементов N_w^{nz} массива w к общему числу элементов N_w массива w :

$$CR = \frac{N_w^{nz}}{N_w}.$$

Характеристика PSNR определяется следующим образом:

$$PSNR = 20 \lg \frac{MAX_I}{\sqrt{MSE}}, \quad MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |I(i, j) - K(i, j)|^2,$$

где I, K — изображения в виде массива размера $m \times n$, MSE — среднеквадратичная ошибка (Mean Square Error), MAX_I — максимальное значение, принимаемое пикселем изображения (если пиксели имеют разрядность d , то $MAX_I = 2^d - 1$).

Для проведения исследования было выбрано 10 тестовых изображений, удовлетворяющих следующим требованиям:

- 1) размерность изображения равна 2 (черно-белые изображения);
- 2) изображения имеют эквивалентные размеры (512×512);
- 3) выбран tif-формат изображений.

Ограничения 1 и 3 не являются обязательными и выбраны только для упрощения задачи программирования. На рис. 2 приведены отобранные материалы для исследования.

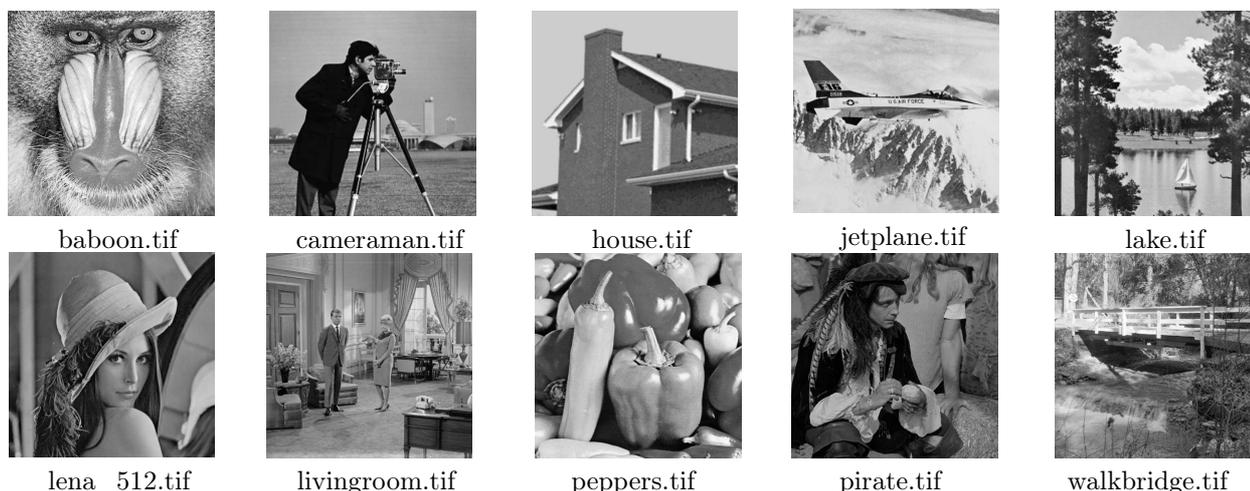


Рис. 2: Набор монохромных изображений, использованных в исследовании

Алгоритм WTT-преобразования реализован на интерпретируемом языке программирования Python с использованием библиотек Numpy и Scipy. Исходный код доступен по ссылке <http://bitbucket.org/hariyuki/wtt-transform-open>.

Проводились эксперименты нескольких видов. В экспериментах первого типа производился сбор данных о зависимости характеристик CR и PSNR от значений ранга WTT-преобразования и от параметра асс, имеющего смысл величины минимального ненулевого значения элементов массива w (параметр сжатия).

В первой серии рассматриваемых экспериментов фиксировались ранг и изображение-источник частного фильтра, строились общий и частный фильтры; далее построенные фильтры применялись ко всем тестовым изображениям. Регулируя параметр сжатия (фактически, превращая псевдоразреженный массив в разреженный), получали различные значения величины CR и затем вычисляли соответствующее значение PSNR для восстановленного из уже разреженного массива изображения.

Вторая серия экспериментов проводилась аналогичным образом с тем лишь различием, что фиксировалось изображение, к которому применяют разложение, и строились все частные фильтры.

Эксперименты второго типа проводились при фиксированном значении PSNR = 30 дБ и при фиксированном тестовом изображении и изображении-источнике фильтров. Выбор параметра сжатия осуществлялся таким образом, чтобы удовлетворить указанное требование о значении PSNR.

3.3. Результаты экспериментов. Результаты экспериментов первого типа оформлены в виде графиков в осях CR и PSNR.

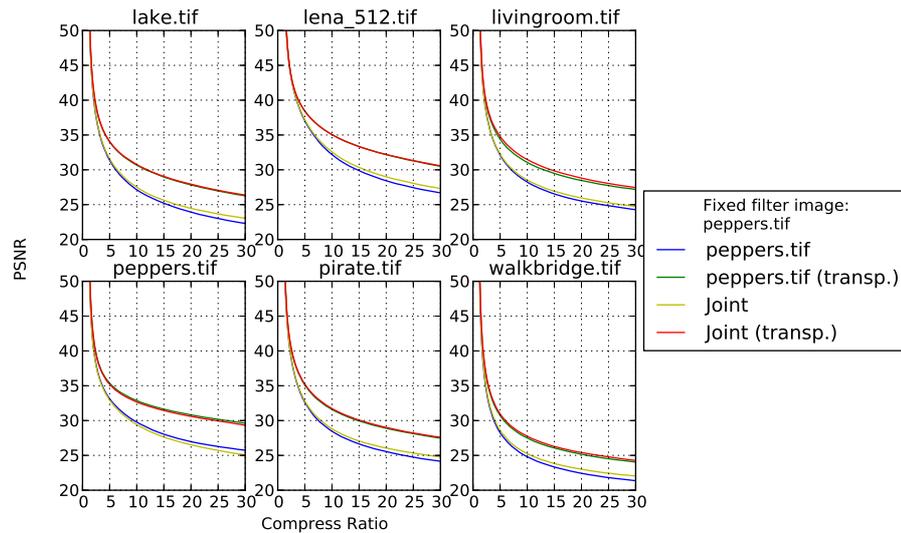


Рис. 3: Применение общего и частного на основе *peppers.tif* фильтров к различным изображениям при ранге разложения 12

На рис. 3 приведены сравнительные графики результатов работы общего фильтра и фильтра частного, построенного на основе *peppers.tif*, в применении к разным данным из начального набора изображений при ранге 12 с использованием квантизации и без нее. Синим цветом обозначен частный фильтр, желтым — общий, зеленым — частный с использованием перестановки, красным — общий с использованием перестановки. Видно, что фильтр по *peppers.tif* в сравнении с общим показывает лучшие результаты на “своем” изображении, в остальных же случаях общий выигрывает по качеству сжатия. Кроме того, хорошо заметно повышение качества сжатия при использовании перестановки индексов.

На рис. 4 представлены графики результатов работы общего фильтра в сравнении с частными в применении к тестовому изображению *lena_512.tif*. Обозначения и оси графика аналогичны рис. 3. Хорошо видно, что лучшие результаты получены для собственного фильтра. Заметим, что при этом для данного изображения общий фильтр дает довольно близкое качество сжатия. Аналогично, использование индексной перестановки позволяет добиться более эффективной работы алгоритма.

Для экспериментов второго типа приводим графики результатов на примере *baboon.tif* (собственный фильтр, фильтр по *lena_512.tif* в сравнении с общим, рис. 5) и *cameraman.tif* (собственный фильтр, фильтр по *peppers.tif* в сравнении с общим, рис. 6 и 7), с использованием квантизации. Заметим, что при росте рангов растёт разрыв между эффективностью общего и частного фильтров. Кроме того, в результате исследований получено, что общий фильтр почти всегда лучше частного несобственного.

Кроме того, на рис. 8 приведены изображения *baboon.tif*, слева направо: исходная картинка, картинка после собственного фильтра, после фильтра по *lena_512.tif*, после общего фильтра и после общего без перестановки. Во всех случаях используется значение ранга 10, PSNR = 30 дБ.

4. Выводы. Согласно результатам численных экспериментов, общий фильтр в большинстве случаев дает более хорошие результаты, чем фильтр, построенный по одному представителю. Исключение составляет свой “собственный” фильтр, который, как и ожидалось, сохраняет более полную информацию о “своем” изображении. Кроме того, есть несколько изображений, для которых фильтр на основе другого

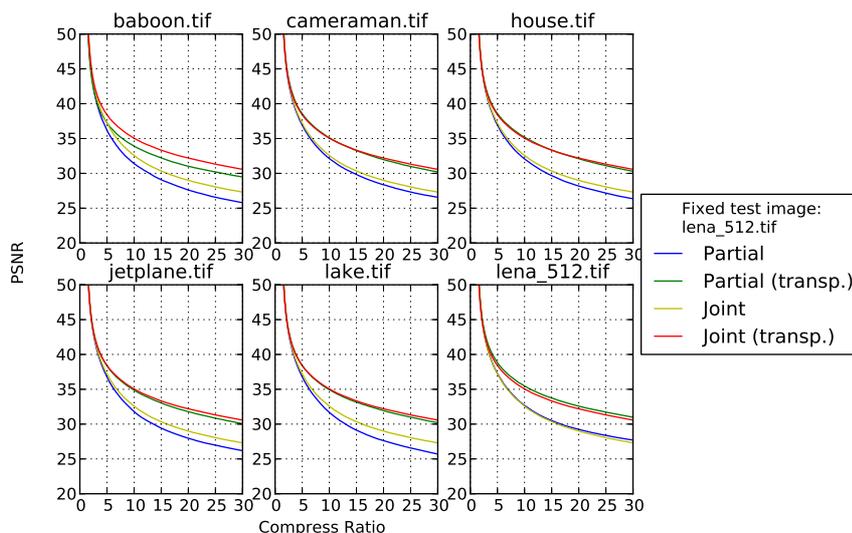


Рис. 4: Применение общего и частного на основе различных изображений фильтров к lena_512.tif при ранге разложения 12

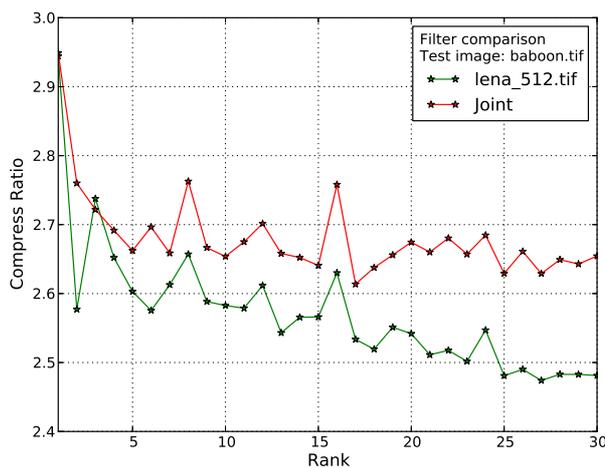


Рис. 5: Исследование эффективности разложения в зависимости от величины ранга при фиксированном PSNR = 30 дБ. Тестовое изображение baboon.tif, частный фильтр на основе lena_512.tif, с использованием перестановки индексов

изображения сохраняет информацию лучше, чем общий. Однако отличия в этом случае незначительны.

Следует также отметить, что лучшая степень сжатия для общего фильтра достигается на сравнительно небольших рангах. Кроме того, при росте ранга разложения растет разрыв между эффективностью общего и частного фильтров, что вполне ожидаемо, так как при этом увеличивается размер фильтров и сохраняется большее количество информации.

В силу сказанного выше, можно выделить несколько стратегий применения WTT-разложения. Первая заключается в использовании общего фильтра большого ранга с последующим его применением и может быть использована в том случае, когда имеется достаточно большой набор данных, скорость не важна, но важно максимальное сжатие. Дополнительно возникает вопрос правильного выбора ранга, поскольку именно он определяет размеры (а значит, и затраты на хранение) фильтров.

Вторая стратегия использует не очень большой ранг и общий фильтр и может применяться в тех

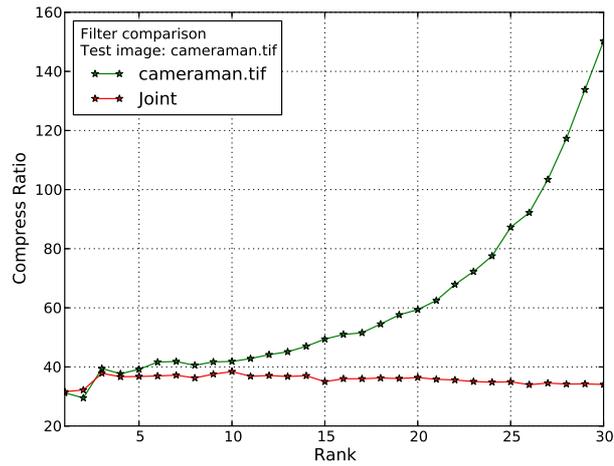


Рис. 6: Исследование эффективности разложения в зависимости от величины ранга при фиксированном PSNR = 30 дБ. Тестовое изображение cameraman.tif, собственный частный фильтр, с использованием перестановки индексов

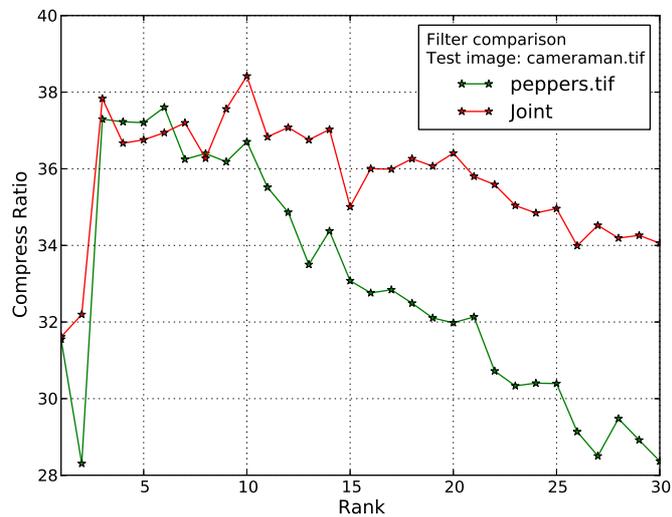


Рис. 7: Исследование эффективности разложения в зависимости от величины ранга при фиксированном PSNR = 30 дБ. Тестовое изображение cameraman.tif, частный фильтр на основе peppers.tif, с использованием перестановки индексов

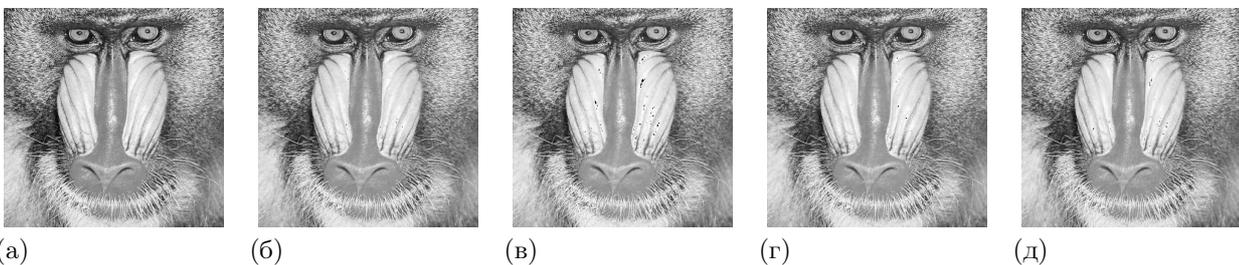


Рис. 8: Восстановленные варианты baboon.tif в сравнении с исходным изображением: (а) — исходное изображение; (б) — восстановленное, собственный фильтр; (в) — восстановленное, фильтр на основе lena_512.tif; (г) — восстановленное, общий фильтр; (д) — восстановленное, общий фильтр, с перестановкой

задачах, где важна скорость сжатия/восстановления.

Таким образом, в зависимости от выбора ранга и источника для построения фильтров можно полу-

чить метод сжатия/восстановления данных, применимый в задачах сжатия различного типа.

Дальнейшее направление исследований состоит в изучении эффективности WTT-преобразования к задаче сжатия многомерных массивов различной природы, например видеозаписей. Кроме того, представляет интерес изучение третьего подхода к построению фильтров, когда из множества однородных данных выбирается некое подмножество, на основании которого и строится набор фильтров разложения.

СПИСОК ЛИТЕРАТУРЫ

1. *Daubechies I.* Ten lectures on wavelets. Philadelphia: SIAM, 1992.
2. *Stanković R.S., Falkowski B.J.* The Haar wavelet transform: its status and achievements // *Comput. Electr. Eng.* 2003. **29**, N 1. 25–44.
3. *Sweldens W.* The lifting scheme: a custom-design construction of biorthogonal wavelets // *Appl. Comput. Harmon. Anal.* 1996. **3**, N 2. 186–200.
4. *Demirel H., Anbarjafari G.* Image resolution enhancement by using discrete and stationary wavelet decomposition // *IEEE Trans. Image Process.* 2011. **20**, N 5. 1458–1460.
5. *Sulochana S., Vidhya R.* Satellite image contrast enhancement using multiwavelets and singular value decomposition (SVD) // *Int. J. Comput. Appl.* 2011. **35**, N 7. 1–5.
6. *Canales-Rodríguez E.J., Radua J., Pomarol-Clotet E., et al.* Statistical analysis of brain tissue images in the wavelet domain: wavelet-based morphometry // *NeuroImage.* 2013. **72**. 214–226.
7. *Patil M.M., Yardi A.R.* Classification of 3D magnetic resonance images of brain using discrete wavelet transform // *Int. J. Comput. Appl.* 2011. **31**, N 7. 23–27.
8. *Omerhodžić I., Avdaković S., Nuhanović A., Dizdarević K.* Energy distribution of EEG signals: EEG signal wavelet-neural network classifier // *World Acad. Sci. Eng. Technol.* 2010. **61**. 1190–1195.
9. *Oseledets I.V., Tyrtyshnikov E.E.* Algebraic wavelet transform via quantics tensor train decomposition // *SIAM J. Sci. Comput.* 2011. **33**, N 3. 1315–1328.
10. *Kazeev V.A., Oseledets I.V.* The tensor structure of a class of adaptive algebraic wavelet transforms. Preprint N 2013-28. Zürich: Swiss Federal Institute of Technology, 2013.
11. *Oseledets I.V.* Tensor-train decomposition // *SIAM J. Sci. Comput.* 2011. **33**, N 5. 2295–2317.
12. *Oseledets I.V.* Approximation of $2^d \times 2^d$ matrices using tensor decomposition // *SIAM J. Matrix Anal. Appl.* 2010. **31**, N 4. 2130–2145.
13. *Khoromskij B.N.* $O(d \log N)$ -quantics approximation of $N - d$ tensors in high-dimensional numerical modeling // *Constr. Appr.* 2011. **34**, N 2. 257–280.
14. *Оседедец И.В.* О приближении матриц логарифмическим числом параметров // *Докл. АН.* 2009. **428**, № 1. 23–24.

Поступила в редакцию
28.02.2014

WTT Decomposition for the Compression of Array’s Families and Its Application to Image Processing

P. V. Kharyuk¹ and I. V. Oseledets²

¹ *Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics; Leninskie Gory, Moscow, 119992, Russia; Student, e-mail: hariyuki.pavel@gmail.com*

² *Institute of Numerical Mathematics, Russian Academy of Sciences; ulitsa Gubkina 8, Moscow, 119333, Russia; Ph.D., Senior Scientist, e-mail: ivan.oseledets@gmail.com*

Received February 28, 2014

Abstract: The application of Wavelet Tensor Train decomposition to the compression of array’s families to image processing is considered. The WTT decomposition is an algebraic technique for the construction of adaptive wavelet transforms. Its main disadvantage is that it requires to store filters for each image. A new approach is proposed on the basis of the construction of a single filter for a sequence of images.

Keywords: numerical tensor methods, wavelet transform, Wavelet Tensor Train decomposition, Tensor Train decomposition, data compression.

References

1. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992).
2. R. S. Stanković and B. J. Falkowski, “The Haar Wavelet Transform: Its Status and Achievements,” *Comput. Electr. Eng.* **29** (1), 25–44 (2003).
3. W. Sweldens, “The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets,” *Appl. Comput. Harmon. Anal.* **3** (2), 186–200 (1996).
4. H. Demirel and G. Anbarjafari, “Image Resolution Enhancement by Using Discrete and Stationary Wavelet Decomposition,” *IEEE Trans. Image Process.* **20** (5), 1458–1460 (2011).
5. S. Sulochana and R. Vidhya, “Satellite Image Contrast Enhancement Using Multiwavelets and Singular Value Decomposition (SVD),” *Int. J. Comput. Appl.* **35** (7), 1–5 (2011).
6. E. J. Canales-Rodríguez, J. Radua, E. Pomarol-Clotet, et al., “Statistical Analysis of Brain Tissue Images in the Wavelet Domain: Wavelet-Based Morphometry,” *NeuroImage* **72**, 214–226 (2013).
7. M. M. Patil and A. R. Yardi, “Classification of 3D Magnetic Resonance Images of Brain Using Discrete Wavelet Transform,” *Int. J. Comput. Appl.* **31** (7), 23–27 (2011).
8. I. Omerhodzic, S. Avdakovic, A. Nuhanovic, and K. Dizdarevic, “Energy Distribution of EEG Signals: EEG Signal Wavelet-Neural Network Classifier,” *World Acad. Sci. Eng. Technol.* **61**, 1190–1195 (2010).
9. I. V. Oseledets and E. E. Tyrtyshnikov, “Algebraic Wavelet Transform Via Quantics Tensor Train Decomposition,” *SIAM J. Sci. Comput.* **33** (3), 1315–1328 (2011).
10. V. A. Kazeev and I. V. Oseledets, *The Tensor Structure of a Class of Adaptive Algebraic Wavelet Transforms*, Preprint No. 2013-28 (Swiss Federal Inst. of Technology, Zürich, 2013).
11. I. V. Oseledets, “Tensor-Train Decomposition,” *SIAM J. Sci. Comput.* **33** (5), 2295–2317 (2011).
12. I. V. Oseledets, “Approximation of $2^d \times 2^d$ Matrices Using Tensor Decomposition,” *SIAM J. Matrix Anal. Appl.* **31** (4), 2130–2145 (2010).
13. B. N. Khoromskij, “ $O(d \log N)$ -Quantics Approximation of $N-d$ Tensors in High-Dimensional Numerical Modeling,” *Constr. Appr.* **34** (2), 257–280 (2011).
14. I. V. Oseledets, “Approximation of Matrices with Logarithmic Number of Parameters,” *Dokl. Akad. Nauk* **428** (1), 23–24 (2009) [*Dokl. Math.* **80** (2), 653–654 (2009)].