

Предварительная обработка изображений для улучшения качества работы алгоритмов сжатия с потерями, основанных на всплесках

Д.А. Ямковой
dmitriyamkovoii@bk.ru

ИММ УрО РАН (Екатеринбург)

Аннотация

В статье разработан метод экстраполяции функций $2k$ -гладких в некоторой окрестности границы прямоугольника Ω на $\mathbb{R}^2 \setminus \Omega$ с сохранением гладкости порядка k и со свойством компактности носителя. Также, в данной работе с помощью численных экспериментов показано, что применение алгоритмов сжатия с потерями, основанных на всплесках, для таким способом экстраполированных изображений дает улучшение основных показателей сжатия по сравнению с существующими методами.

1 Введение

Рассмотрим задачу аппроксимации функций, заданных на отрезке (в данном разделе будем в основном говорить только об одномерном случае, так как в двумерном случае возникают те же проблемы) с помощью аппарата теории всплесков. Такая задача может возникнуть, например, при сжатии и дальнейшем восстановлении некоторых достаточно гладких данных. При решении поставленной задачи, однако, возникают некоторые трудности – аппарат классических базисных масштабирующих функций пространств кратномасштабного анализа (КМА) (см., например, [1], гл. 5), используемый для аппроксимации, плохо приспособлен для приближения функций, определенных на отрезке. Существуют различные способы, описанные в [2], адаптации базиса $L^2(\mathbb{R})$ к случаю $L^2([a, b])$.

Один из способов решения данной проблемы заключается в том, чтобы продолжить исходную функцию нулем на всю вещественную ось и использовать стандартный аппарат КМА. Также, можно сначала продолжить исходную функцию на более широкий промежуток симметрично, периодически или непрерывно, а затем уже положить функцию нулем вне новой области определения.

Довольно часто применяют периодизированные всплески (см., например, [1], гл. 9, §9.3). Суть данного метода заключается в том, чтобы периодизировать каждую функцию из набора всплеск функций $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ и набора масштабирующих функций $\{\varphi_{j,k}\}_{j,k \in \mathbb{Z}}$, получив таким образом ортонормированные базисы подпространств $V_j^{\text{пер}}$ пространства $\overline{\bigcup_{j \in \mathbb{Z}} V_j^{\text{пер}}} = L^2([a, b])$.

Еще один способ решить возникающую проблему – метод так называемых «folding» всплесков, определенных в [3]. Основная идея состоит в том, чтобы исходную функцию f симметрично «согнуть» относи-

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A.A. Makhnev, S.F. Pravdin (eds.): Proceedings of the International Youth School-conference «SoProMat-2017», Yekaterinburg, Russia, 06-Feb-2017, published at <http://ceur-ws.org>

тельно правого и левого края промежутка $[a, b]$:

$$f(x) = \begin{cases} f(-x + 2b), & x \in [b, 2b - a], \\ f(-x + 2a), & x \in [2a - b, a], \end{cases}$$

а затем уже для новой функции, определённой на $[2a - b, 2b - a]$, проделать тоже самое еще раз и так далее. Также, техника «сгибания» применяется для набора всплеск функций $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ и набора масштабирующих функций $\{\varphi_{j,k}\}_{j,k \in \mathbb{Z}}$. Однако, может оказаться, что данная процедура не приведет нас к ортонормированным базисам всплесков, поэтому «сгибания» используют в основном для биортогональных всплесков.

Также существуют другие способы, такие как использование «left edge», «right edge», «interior» всплесков (предложены Мейером в [4]) или схожего с методом Мейера способа, предложенного Коэном, Добеши и Виалом в [2] («interior» и «edge» всплески). Оба варианта отличаются громоздкостью требуемых построений.

У всех вышеперечисленных методов имеются недостатки, например, при продолжении исходной функции одним из способов, описанных во втором абзаце данного раздела, серьезным недостатком является потеря точности аппроксимации из-за в общем случае нарушения гладкости исходной функции (см. результаты численного эксперимента в Таб. 1. Целью данной работы является устранение недостатков приведенных выше методов за счет экстраполяции исходной функции $2k$ -гладкой в некоторой окрестности границы прямоугольника Ω на $\mathbb{R}^2 \setminus \Omega$ с сохранением гладкости порядка k и со свойством компактности носителя. Также, в данной работе проведено сравнение результатов применения основанных на всплесках алгоритмов сжатия изображений с потерями к экстраполированным изображениям с результатами применения тех же алгоритмов к исходным изображениям.

2 Экстраполяция

Пусть $f : \Omega \rightarrow \mathbb{R}$ — $2k$ -гладкая функция в некоторой окрестности границы Ω (то есть существуют и непрерывны все частные производные $\frac{\partial^k f}{\partial x^{k_1} \partial y^{k_2}}$ функции $f = f(x, y)$, где $k \in \{0\} \cup \mathbb{N}$, $k_1, k_2 \in [0, k] \cap \mathbb{Z}$, $k_1 + k_2 = k$), $\Omega = [a_1, a_2] \times [b_1, b_2]$. Пусть $A_1 < a_1 < a_2 < A_2$, $B_1 < b_1 < b_2 < B_2$. Требуется построить функцию

$$F(x, y) := \begin{cases} f(x, y), & (x, y) \in \Omega \\ g(x, y), & (x, y) \in ([A_1, A_2] \times [B_1, B_2]) \setminus \Omega \\ 0, & (x, y) \notin [A_1, A_2] \times [B_1, B_2] \end{cases} \quad (1)$$

такую, чтобы выполнялись условия

$$\begin{cases} g \in C^k(([A_1, A_2] \times [B_1, B_2]) \setminus \Omega), \\ g_x^{(i)}(a_j, y) = f_x^{(i)}(a_j, y), \quad g_x^{(i)}(A_j, y) = 0, \quad y \in [b_1, b_2], \quad i = 0, \dots, k, \quad j = 1, 2, \\ g_y^{(i)}(x, b_j) = f_y^{(i)}(x, b_j), \quad g_y^{(i)}(x, B_j) = 0, \quad x \in [a_1, a_2], \quad i = 0, \dots, k, \quad j = 1, 2. \end{cases} \quad (2)$$

2.1 Одномерный случай

Покажем сначала, как произвольную функцию $f = f(x)$, $x \in [c, d]$, обладающую односторонними производными вплоть до k -го порядка в точке d , продолжить на $(d, e]$ гладко до порядка k и с условием $f(e) = 0$ с помощью многочлена $H_n(x)$ подходящей степени, то есть так, чтобы выполнялись следующие условия:

$$\begin{cases} H_n(x) \in C^k((d, e)), \\ H_n^{(i)}(d) = f^{(i)}(d), \quad H_n^{(i)}(e) = 0, \quad i = 0, 1, \dots, k. \end{cases} \quad (3)$$

Лемма 1 *Явный вид многочлена $H_n(x)$, удовлетворяющего условиям (3), можно получить с помощью следующей формулы:*

$$H_{2k+1}(x) = (x - e)^{k+1} \left[\sum_{i=0}^k f^{(i)}(d) \frac{(x-d)^i}{i!} \left[\sum_{j=0}^{k-i} (-1)^j \binom{k+j}{k} (d-e)^{-(j+k+1)} (x-d)^j \right] \right]. \quad (4)$$

Доказательство Известно (как частный случай эрмитовой интерполяции), что задача нахождения многочлена $H_n(x)$ из условий (3) разрешима и, при этом имеет единственное решение (см., например [5], гл. 2, §2.7), если $n = 2k + 1$. $H_n(x)$ – многочлен степени $n = 2k + 1$, удовлетворяющий условиям (3) и называемый интерполяционным многочленом Эрмита, имеет вид:

$$H_{2k+1}(x) = \frac{\omega_{2k+2}(x)}{(x-d)^{k+1}} \left[\sum_{i=0}^k f^{(i)}(d) \frac{(x-d)^i}{i!} \left\{ \frac{(x-d)^{k+1}}{\omega_{2k+2}(x)} \right\}_{[d]}^{[k-i]} \right], \quad (5)$$

где $\omega_{2k+2}(x) = (x-d)^{k+1}(x-e)^{k+1}$, а выражение $\left\{ \frac{(x-d)^{k+1}}{\omega_{2k+2}(x)} \right\}_{[d]}^{[k-i]}$ – есть совокупность первых членов разложения в ряд Тейлора по степеням $(x-d)$ функции $\frac{(x-d)^{k+1}}{\omega_{2k+2}(x)}$ в точке d до $(k-i)$ -го члена включительно. Путем несложных выкладок (5) преобразуется к виду (4). ■

Таким образом, с помощью леммы 1 мы можем построить многочлен $H_n = H_n(x)$, $x \in (d, e]$, удовлетворяющий всем условиям из (3). В качестве численного эксперимента экстраполируем с сохранением гладкости порядка $k = 3$ функцию $\alpha(x) = 3x^2 \cdot \cos x + 2x \cdot \sin x + 5$ с отрезка $[0, 4]$ на всю ось в виде функции с носителем $[-2, 6]$ и проведем сначала всплеск-разложение так экстраполированной функции до уровня 5 с помощью всплесков семейства Добеши с 15-ю нулевыми моментами $\int_{-\infty}^{\infty} x^n \psi(x) dx = 0$, $n = 0, 1, \dots, 14$ (это соответствует длине носителей масштабирующей функции φ и всплеска ψ равной 29). Далее, применим разложение до того же уровня по тем же всплескам к функции α , продолженной описанными во втором абзаце введения способами. Применим также к исходной функции описанную процедуру, но основанную на периодизированных и «interior», «edge» всплесках, упомянутых в третьем и пятом абзацах введения. Будем сравнивать время выполнения (в секундах) всплеск-разложения до выбранного уровня и погрешности приближения на отрезке $[0, 4]$ в нормах C и L^2 при аппроксимации с помощью всплесков Добеши с использованием указанных выше способов продолжения исходной функции, а также периодизированных и «interior», «edge» всплесков.

В качестве среды для программной реализации экстраполяции исходной функции и выполнения всплеск-разложения был выбран пакет прикладных программ *Matlab* версии 8.4.0.150421 (R2014b) с расширениями *Wavelet Toolbox* версии 4.14 и *WaveLab* версии 0.850. Всплеск-разложение исходной функции f и её последующая аппроксимация всплесками Добеши реализованы с помощью соответственно функций *wavedec* и *wrccoef* из библиотеки *Wavelet Toolbox*. Разложение с использованием периодизированных и «interior», «edge» всплесков исходной функции f и её последующая аппроксимация реализованы с помощью соответственно функций *MakeONFilter*, *FWT_PO*, *MakeOBFfilter*, *FWT_CDJV* и *UpDyadLo*, *CDJVDyadUp* из библиотеки *WaveLab*.

Как видно из данных численного эксперимента, представленных в Таб. 1, с точки зрения погрешности приближения при аппроксимации с помощью всплесков метод предварительной гладкой экстраполяции исходной функции с помощью многочленов Эрмита дает результаты лучше, чем другие сравниваемые методы. Эффект потери точности аппроксимации из-за нарушения гладкости продолжения исходной функции можно наблюдать при других уровнях разложения и для других семейств всплесков.

Таблица 1: Результаты описанного численного эксперимента в одномерном случае

Продолжение		Числовые характеристики		
		Время(с)	Погрешность в норме C	Погрешность в норме L^2
1.	Нулем	0.0144	$0.1680 \cdot 10^2$	0.4645
2.	Периодическое	0.0128	$0.1780 \cdot 10^2$	0.4671
3.	Симметричное	0.0148	$0.1613 \cdot 10^{-1}$	$0.3119 \cdot 10^{-3}$
4.	Непрерывное	0.0142	$0.1595 \cdot 10^{-1}$	$0.2864 \cdot 10^{-3}$
5.	Эрмитово ($k = 3$)	0.0150	$0.1558 \cdot 10^{-4}$	$0.4121 \cdot 10^{-6}$

Всплески		Числовые характеристики		
		Время(с)	Погрешность в норме C	Погрешность в норме L^2
1.	Периодизированные	0.0048	$0.1899 \cdot 10^2$	0.7407
2.	«interior», «edge»	0.0825	$0.2378 \cdot 10^{-2}$	$0.2421 \cdot 10^{-3}$

2.2 Двумерный случай

Покажем, как построить $g = g(x, y)$ из (1) на $(a_2, A_2] \times [b_1, b_2]$ и $[a_1, a_2] \times (b_2, B_2]$ так, чтобы выполнялись условия (2). На $[A_1, a_1] \times [b_1, b_2]$ и $[a_1, a_2] \times [B_1, b_1]$ продолжение будет строиться аналогично. Используем формулу (4) из леммы 1 и построим g на $(a_2, A_2] \times [b_1, b_2]$ с помощью функции $H_{a_2}(x, y)$:

$$H_{a_2}(x, y) = (x - A_2)^{k+1} \left[\sum_{i=0}^k f_x^{(i)}(a_2, y) \frac{(x - a_2)^i}{i!} \left[\sum_{j=0}^{k-i} (-1)^j \binom{k+j}{k} (a_2 - A_2)^{-(j+k+1)} (x - a_2)^j \right] \right]. \quad (6)$$

Используя аналог формулы (4) из леммы 1 и построим g на $[a_1, a_2] \times (b_2, B_2]$ с помощью функции $H_{b_2}(x, y)$:

$$H_{b_2}(x, y) = (y - B_2)^{k+1} \left[\sum_{i=0}^k f_y^{(i)}(x, b_2) \frac{(y - b_2)^i}{i!} \left[\sum_{j=0}^{k-i} (-1)^j \binom{k+j}{k} (b_2 - B_2)^{-(j+k+1)} (y - b_2)^j \right] \right]. \quad (7)$$

Очевидно, что так построенная функция удовлетворяет условиям (2).

Покажем теперь, как построить g на $(a_2, A_2] \times (b_2, B_2]$ в виде многочлена

$$P(x, y) = \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} c_{i,j} (x - A_2)^i (y - B_2)^j, \quad (x, y) \in [a_2, A_2] \times [b_2, B_2],$$

подчиненного условиям:

$$\begin{cases} P_y^{(i)}(x, B_2) = 0, & P_y^{(i)}(x, b_2) = (H_{a_2})_y^{(i)}(x, b_2), & x \in (a_2, A_2], & i = 0, \dots, k, \\ P_x^{(i)}(A_2, y) = 0, & P_x^{(i)}(a_2, y) = (H_{b_2})_x^{(i)}(a_2, y), & y \in (b_2, B_2], & \end{cases} \quad (8)$$

где $c_{i,j}$ – неизвестные коэффициенты. На $[A_1, a_1] \times [B_1, b_1]$ и $[A_1, a_1] \times (b_2, B_2]$, $(a_2, A_2] \times [B_1, b_1]$ продолжение строится аналогично. Покажем, как найти все $c_{i,j}$. Обозначим

$$\begin{aligned} \tilde{c}_{a_2}^{0,i} &= \sum_{j=1}^{2k+1} c_{i,j} (b_2 - B_2)^j, & \tilde{c}_{a_2}^{l,i} &= \sum_{j=l}^{2k+1} \frac{j!}{(j-l)!} c_{i,j} (b_2 - B_2)^{j-l}, & i = 1, \dots, 2k+1, & l = 1, \dots, k, \\ \tilde{c}_{b_2}^{0,j} &= \sum_{i=1}^{2k+1} c_{i,j} (a_2 - A_2)^i, & \tilde{c}_{b_2}^{l,j} &= \sum_{i=l}^{2k+1} \frac{i!}{(i-l)!} c_{i,j} (a_2 - A_2)^{i-l}, & j = 1, \dots, 2k+1, & \end{aligned} \quad (9)$$

тогда

$$\begin{aligned} P_y^{(l)}(x, b_2) &= \sum_{i=1}^{2k+1} \tilde{c}_{a_2}^{l,i} (x - A_2)^i, \\ P_x^{(l)}(a_2, y) &= \sum_{j=1}^{2k+1} \tilde{c}_{b_2}^{l,j} (y - B_2)^j, \end{aligned} \quad l = 0, \dots, k. \quad (10)$$

Из (6)–(10) можно видеть, что

$$\begin{aligned} \tilde{c}_{a_2}^{l,i} &= 0, & i = 1, \dots, k, & l = 0, \dots, k, \\ \tilde{c}_{b_2}^{l,j} &= 0, & j = 1, \dots, k, & \end{aligned} \quad (11)$$

Перепишем соответствующие условия из (8) с учетом (11)

$$\begin{aligned} \sum_{i=0}^k \tilde{c}_{a_2}^{l,i+k+1} (x - A_2)^i &= \sum_{i=0}^k \frac{(f_x^{(i)})_y^{(l)}(a_2, b_2)}{i!} \left[\sum_{j=0}^{k-i} (-1)^j \binom{k+j}{k} (a_2 - A_2)^{-(j+k+1)} (x - a_2)^{i+j} \right], \\ \sum_{i=0}^k \tilde{c}_{b_2}^{l,i+k+1} (y - B_2)^i &= \sum_{i=0}^k \frac{(f_y^{(i)})_x^{(l)}(a_2, b_2)}{i!} \left[\sum_{j=0}^{k-i} (-1)^j \binom{k+j}{k} (b_2 - B_2)^{-(j+k+1)} (y - b_2)^{i+j} \right], \\ & l = 0, \dots, k. \end{aligned}$$

В правых частях двух последних уравнений сделаем замену $j = p - i$, поменяем порядок суммирования и получим

$$\begin{aligned} \sum_{i=0}^k \tilde{c}_{a_2}^{l,i+k+1} (x - A_2)^i &= \sum_{p=0}^k \left[\sum_{i=0}^k \frac{(f_x^{(i)})_y^{(l)}(a_2, b_2)}{i!} (-1)^{p-i} \binom{k+p-i}{k} (a_2 - A_2)^{-(p-i+k+1)} (x - a_2)^p \right], \\ \sum_{i=0}^k \tilde{c}_{b_2}^{l,i+k+1} (y - B_2)^i &= \sum_{p=0}^k \left[\sum_{i=0}^k \frac{(f_y^{(i)})_x^{(l)}(a_2, b_2)}{i!} (-1)^{p-i} \binom{k+p-i}{k} (b_2 - B_2)^{-(p-i+k+1)} (y - b_2)^p \right], \\ & l = 0, \dots, k. \end{aligned} \quad (12)$$

Обозначим

$$\begin{aligned} S_{a_2}^{l,p} &= \sum_{i=0}^k \frac{(f_x^{(i)})_y^{(l)}(a_2, b_2)}{i!} (-1)^{p-i} \binom{k+p-i}{k} (a_2 - A_2)^{-(p-i+k+1)}, \\ S_{b_2}^{l,p} &= \sum_{i=0}^k \frac{(f_y^{(i)})_x^{(l)}(a_2, b_2)}{i!} (-1)^{p-i} \binom{k+p-i}{k} (b_2 - B_2)^{-(p-i+k+1)}, \\ & p = 0, \dots, k, \quad l = 0, \dots, k. \end{aligned}$$

Тогда дифференцируя соответствующие выражения из (12) в точках A_2 и B_2 соответственно получим

$$\begin{aligned} \tilde{c}_{a_2}^{l,i+k+1} &= \sum_{p=i}^k S_{a_2}^{l,p} (A_2 - a_2)^{p-i} \binom{p}{i}, \\ \tilde{c}_{b_2}^{l,i+k+1} &= \sum_{p=i}^k S_{b_2}^{l,p} (B_2 - b_2)^{p-i} \binom{p}{i}, \\ & i = 0, \dots, k, \quad l = 0, \dots, k. \end{aligned} \quad (13)$$

Обозначим $\alpha = a_2 - A_2$, $\beta = b_2 - B_2$. Пусть $\tilde{c}_{a_2}^i, \tilde{c}_{b_2}^i$, $i = 0, \dots, k$ – вектор-столбцы, координаты которых вычислены по формулам (11) и (13), тогда мы можем переписать (9) в виде системы линейных уравнений

$$M_k \cdot c = \tilde{c}_k \quad (14)$$

относительно столбца неизвестных c , где

$$M_k = \begin{pmatrix} A(\alpha) & A^2(\alpha) & A^3(\alpha) & \dots & A^{2k+1}(\alpha) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{(k)}(\alpha) & (A^2(\alpha))^{(k)} & (A^3(\alpha))^{(k)} & \dots & (A^{2k+1}(\alpha))^{(k)} \\ B_1(\beta) & B_2(\beta) & B_3(\beta) & \dots & B_{2k+1}(\beta) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_1^{(k)}(\beta) & B_2^{(k)}(\beta) & B_3^{(k)}(\beta) & \dots & B_{2k+1}^{(k)}(\beta) \end{pmatrix}$$

– матрица размерности $2(k+1)(2k+1) \times (2k+1)^2$; $A(\alpha) = \text{diag}\{\alpha, \alpha, \dots, \alpha\}$ – диагональная матрица размерности $(2k+1) \times (2k+1)$; $B_i(\beta)$ – матрица размерности $(2k+1) \times (2k+1)$:

$$(B_i(\beta))(j, \cdot) = \begin{cases} (\beta \ \beta^2 \ \dots \ \beta^{2k+1}), & j = i, \\ (0 \ 0 \ \dots \ 0), & j \neq i, \end{cases}$$

$c = (c_{1,1}, \dots, c_{1,2k+1}, \dots, c_{2k+1,1}, \dots, c_{2k+1,2k+1})^T$, $\tilde{c}_k = ((\tilde{c}_{b_2}^0)^T, \dots, (\tilde{c}_{b_2}^k)^T, (\tilde{c}_{a_2}^0)^T, \dots, (\tilde{c}_{a_2}^k)^T)^T$. Решив систему (14), мы определим все неизвестные коэффициенты $c_{i,j}$ многочлена $P(x, y)$.

Лемма 2 *Справедливы следующие утверждения.*

1. Матрицу $m \times k$

$$\begin{pmatrix} 1 & 2 & 3 & \dots & j & \dots & k \\ \alpha & \alpha^2 & \alpha^3 & \dots & \alpha^j & \dots & \alpha^k \\ 1 & 2\alpha & 3\alpha^2 & \dots & j\alpha^{j-1} & \dots & k\alpha^{k-1} \\ 0 & 2 \cdot 1 & 3 \cdot 2\alpha & \dots & j(j-1)\alpha^{j-2} & \dots & k(k-1)\alpha^{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & j \cdot \dots \cdot (j-m+2)\alpha^{j-m+1} & \dots & k \cdot \dots \cdot (k-m+2)\alpha^{k-m+1} \end{pmatrix}$$

эквивалентными преобразованиями можно привести к виду

$$\begin{pmatrix} 1 & 2 & 3 & \dots & j & \dots & k \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{j-1} & \dots & \alpha^{k-1} \\ 0 & 1 & 2\alpha & \dots & \frac{j-1}{1} \alpha^{j-2} & \dots & \frac{k-1}{1} \alpha^{k-2} \\ 0 & 0 & 1 & \dots & \frac{(j-1)(j-2)}{1 \cdot 2} \alpha^{j-3} & \dots & \frac{(k-1)(k-2)}{1 \cdot 2} \alpha^{k-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{(j-1) \cdot \dots \cdot (j-m+1)}{1 \cdot \dots \cdot (m-1)} \alpha^{j-m} & \dots & \frac{(k-1) \cdot \dots \cdot (k-m+1)}{1 \cdot \dots \cdot (m-1)} \alpha^{k-m} \end{pmatrix}$$

2. Вектор-столбец $(c_1, \dots, c_i, \dots, c_m)^T$ преобразованиями, осуществляющими пункт 1, приводится к виду

$$\left(\frac{c_1}{\alpha}, \dots, \sum_{j=0}^{i-1} (-1)^j \frac{1}{(i-1-j)!} \alpha^{-j-1} c_{i-j}, \dots, \sum_{j=0}^{m-1} (-1)^j \frac{1}{(m-1-j)!} \alpha^{-j-1} c_{m-j} \right)^T.$$

Доказательство Докажем утверждение 1 с помощью индукции по n – количество первых строк исходной матрицы ($n < m$).

База индукции при $n = 1$ очевидна. Предположим, что исходную матрицу можно эквивалентными преобразованиями привести к искомому виду

$$\begin{matrix} & 1 & \cdots & n & & n+1 & \cdots & & k \\ n & \left(\begin{array}{ccccccc} \vdots & \ddots & \vdots & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 1 & & \frac{n \cdots 2}{1 \cdots (n-1)} \alpha & \cdots & \frac{(k-1) \cdots (k-n+1)}{1 \cdots (n-1)} \alpha^{k-n} \\ 0 & \cdots & n \cdots 1 & & (n+1) \cdots 2 \alpha & \cdots & k \cdots (k-n+1) \alpha^{k-n} \\ \vdots & \ddots & \vdots & & \vdots & & \ddots & & \vdots \end{array} \right)
 \end{matrix}$$

В полученной матрице вычтем из $(n+1)$ -й строки n -ю строку, умноженную на $n!$,

$$\begin{matrix} & 1 & \cdots & n & & n+1 & \cdots & & k \\ n & \left(\begin{array}{ccccccc} \vdots & \ddots & \vdots & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 1 & & \frac{n \cdots 2}{1 \cdots (n-1)} \alpha & \cdots & \frac{(k-1) \cdots (k-n+1)}{1 \cdots (n-1)} \alpha^{k-n} \\ 0 & \cdots & 0 & & n \cdots 1 \alpha & \cdots & (k-1) \cdots (k-n) \alpha^{k-n} \\ \vdots & \ddots & \vdots & & \vdots & & \ddots & & \vdots \end{array} \right) \sim \\ n & \left(\begin{array}{ccccccc} \vdots & \ddots & \vdots & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 1 & & \frac{n \cdots 2}{1 \cdots (n-1)} \alpha & \cdots & \frac{(k-1) \cdots (k-n+1)}{1 \cdots (n-1)} \alpha^{k-n} \\ 0 & \cdots & 0 & & 1 & \cdots & \frac{(k-1) \cdots (k-n)}{1 \cdots n} \alpha^{k-n-1} \\ \vdots & \ddots & \vdots & & \vdots & & \ddots & & \vdots \end{array} \right)
 \end{matrix}$$

Доказательство утверждения 2 аналогично доказательству утверждения 1. \blacksquare

Теорема Система (14) совместна и ее решение имеет вид:

при $k = 0$: $c_{1,1} = \frac{\tilde{c}_{b_2}^{0,1}}{\alpha}$;

при $k = 1$:

$$\left\{ \begin{array}{l} c_{3,2} = \frac{\tilde{c}_{a_2}^{1,3}}{1 \cdot \beta} - \frac{\tilde{c}_{a_2}^{0,3}}{\beta^2} - 2\beta c_{3,3}, \quad c_{3,1} = \frac{2\tilde{c}_{a_2}^{0,3}}{\beta} - \tilde{c}_{a_2}^{1,3} + \beta^2 c_{3,3}, \\ c_{2,1} = -\frac{4\alpha \tilde{c}_{a_2}^{0,3}}{\beta} + 2\alpha \tilde{c}_{a_2}^{1,3} - 2\alpha \beta^2 c_{3,3}, \quad c_{2,2} = \frac{\tilde{c}_{b_2}^{1,2}}{1 \cdot \alpha} - \frac{\tilde{c}_{b_2}^{0,2}}{\alpha^2} - \frac{2\alpha \tilde{c}_{a_2}^{1,3}}{1 \cdot \beta} + \frac{2\alpha \tilde{c}_{a_2}^{0,3}}{\beta^2} + 4\alpha \beta c_{3,3}, \quad c_{2,3} = \frac{\tilde{c}_{b_2}^{1,3}}{1 \cdot \alpha} - \frac{\tilde{c}_{b_2}^{0,3}}{\alpha^2} - 2\alpha c_{3,3}, \\ c_{1,1} = \frac{2\alpha^2 \tilde{c}_{a_2}^{0,3}}{\beta} - \alpha^2 \tilde{c}_{a_2}^{1,3} + \alpha^2 \beta^2 c_{3,3}, \quad c_{1,2} = \frac{2\tilde{c}_{b_2}^{0,2}}{\alpha} - \tilde{c}_{b_2}^{1,2} + \frac{\alpha^2 \tilde{c}_{a_2}^{1,3}}{1 \cdot \beta} - \frac{\alpha^2 \tilde{c}_{a_2}^{0,3}}{\beta^2} - 2\alpha^2 \beta c_{3,3}, \quad c_{1,3} = \frac{2\tilde{c}_{b_2}^{0,3}}{\alpha} - \tilde{c}_{b_2}^{1,3} + \alpha^2 c_{3,3}, \end{array} \right. \quad (15)$$

где $c_{3,3}$ – свободная переменная;

при $k = 2$:

$$\left\{ \begin{array}{l} c_{5,3} = \frac{\tilde{c}_{a_2}^{2,5}}{2 \cdot 1 \cdot \beta} - \frac{\tilde{c}_{a_2}^{1,5}}{1 \cdot \beta^2} + \frac{\tilde{c}_{a_2}^{0,5}}{\beta^3} - \frac{3 \cdot 2}{2 \cdot 1} c_{5,4} - \frac{4 \cdot 3}{2 \cdot 1} c_{5,5}, \\ c_{5,2} = \frac{\tilde{c}_{a_2}^{1,5}}{1 \cdot \beta} - \frac{\tilde{c}_{a_2}^{0,5}}{\beta^2} - 2\beta c_{5,3} - 3\beta^2 c_{5,4} - 4\beta^3 c_{5,5}, \\ c_{5,1} = \frac{\tilde{c}_{a_2}^{0,5}}{\beta} - \beta c_{5,2} - \beta^2 c_{5,3} - \beta^3 c_{5,4} - \beta^4 c_{5,5}, \\ c_{4,3} = \frac{\tilde{c}_{a_2}^{2,4}}{2 \cdot 1 \cdot \beta} - \frac{\tilde{c}_{a_2}^{1,4}}{1 \cdot \beta^2} + \frac{\tilde{c}_{a_2}^{0,4}}{\beta^3} - \frac{3 \cdot 2}{2 \cdot 1} c_{4,4} - \frac{4 \cdot 3}{2 \cdot 1} c_{4,5}, \\ c_{4,2} = \frac{\tilde{c}_{a_2}^{1,4}}{1 \cdot \beta} - \frac{\tilde{c}_{a_2}^{0,4}}{\beta^2} - 2\beta c_{4,3} - 3\beta^2 c_{4,4} - 4\beta^3 c_{4,5}, \\ c_{4,1} = \frac{\tilde{c}_{a_2}^{0,4}}{\beta} - \beta c_{4,2} - \beta^2 c_{4,3} - \beta^3 c_{4,4} - \beta^4 c_{4,5}, \\ c_{3,i} = -\frac{3 \cdot 2}{2 \cdot 1} \alpha c_{4,i} - \frac{4 \cdot 3}{2 \cdot 1} \alpha^2 c_{5,i} + \frac{\tilde{c}_{b_2}^{2,i}}{2 \cdot 1 \cdot \alpha} - \frac{\tilde{c}_{b_2}^{1,i}}{1 \cdot \alpha^2} + \frac{\tilde{c}_{b_2}^{0,i}}{\alpha^3}, \quad i = 1, \dots, 5, \\ c_{2,i} = -2\alpha c_{3,i} - 3\alpha^2 c_{4,i} - 4\alpha^3 c_{5,i} + \frac{\tilde{c}_{b_2}^{1,i}}{1 \cdot \alpha} - \frac{\tilde{c}_{b_2}^{0,i}}{\alpha^2}, \quad i = 1, \dots, 5, \\ c_{1,i} = -\alpha c_{2,i} - \alpha^2 c_{3,i} - \alpha^3 c_{4,i} - \alpha^4 c_{5,i} + \frac{\tilde{c}_{b_2}^{0,i}}{\alpha}, \quad i = 1, \dots, 5, \end{array} \right. \quad (16)$$

где $c_{4,4}, c_{4,5}, c_{5,4}, c_{5,5}$ – свободные переменные.

Используя (13) получаем

$$\begin{aligned} \tilde{c}_{a_2}^{0,k+1} \alpha^{k+1} + \dots + \tilde{c}_{a_2}^{0,2k+1} \alpha^{2k+1} &= \alpha^{k+1} \sum_{p=0}^k S_{a_2}^{0,p} (A_2 - a_2)^{p-0} \binom{p}{0} + \dots + \alpha^{2k+1} \sum_{p=k}^k S_{a_2}^{0,p} (A_2 - a_2)^{p-k} \binom{p}{k} = \\ S_{a_2}^{0,0} \alpha^{k+1} \frac{0!}{(0-0)!0!} + S_{a_2}^{0,1} [(A_2 - a_2) \alpha^{k+1} \frac{1!}{(1-0)!0!} + \alpha^{k+2} \frac{1!}{(1-1)!1!}] + \dots + S_{a_2}^{0,k} [(A_2 - a_2)^k \alpha^{k+1} \frac{k!}{(k-0)!0!} + \\ \dots + (A_2 - a_2)^0 \alpha^{2k+1} \frac{k!}{(k-k)!k!}] &= S_{a_2}^{0,0} \alpha^{k+1} + S_{a_2}^{0,1} \alpha^{k+2} (-1+1)^1 + \dots + S_{a_2}^{0,k} \alpha^{2k+1} (-1+1)^k = S_{a_2}^{0,0} \alpha^{k+1}. \end{aligned}$$

Аналогично доказывается, что $\tilde{c}_{b_2}^{0,k+1} \beta^{k+1} + \dots + \tilde{c}_{b_2}^{0,2k+1} \beta^{2k+1} = S_{b_2}^{0,0} \beta^{k+1}$. Таким образом, (17) верно тогда и только тогда, когда $S_{a_2}^{0,0} \alpha^{k+1} = S_{b_2}^{0,0} \beta^{k+1}$, но

$$\begin{aligned} \sum_{i=0}^k \frac{f_x^{(i)}(a_2, b_2)}{i!} (-1)^{-i} \binom{k-i}{k} (a_2 - A_2)^{-(-i+k+1)} \alpha^{k+1} &= \sum_{i=0}^k \frac{f_y^{(i)}(a_2, b_2)}{i!} (-1)^{-i} \binom{k-i}{k} (b_2 - B_2)^{-(-i+k+1)} \beta^{k+1} \Leftrightarrow \\ f(a_2, b_2) (a_2 - A_2)^{-(k+1)} \alpha^{k+1} &= f(a_2, b_2) (b_2 - B_2)^{-(k+1)} \beta^{k+1}. \end{aligned}$$

Аналогично доказывается, что

$$\sum_{i=k+1}^{2k+1} \tilde{c}_{a_2}^{i,i} \left[\frac{(i-1) \cdot \dots \cdot (i-m+1)}{1 \cdot \dots \cdot (m-1)} \alpha^{i-m} \right] - \sum_{i=0}^{m-1} (-1)^i \frac{1}{(m-1-i)!} \alpha^{-i-1} \left[\sum_{j=k+1}^{2k+1} \tilde{c}_{b_2}^{m-1-i,j} \frac{j!}{(j-l)!} \beta^{j-l} \right] = 0$$

для любого $m = 1, \dots, k+1$ и любого $l = 0, \dots, k$. Таким образом, $rank(M_k) = rank(M_k, \tilde{c}_k)$.

При $k = 0$ сразу же получаем $c_{1,1} = \frac{\tilde{c}_{b_2}^{0,1}}{\alpha}$. Выпишем теперь решение (14) при $k = 1$ в явном виде. Можно видеть, что $rank(M_1)$ равен 8; в качестве свободной переменной можем взять $c_{3,3}$. Выразим оставшиеся переменные через свободную и получим (15). Также, выпишем решение (14) при $k = 2$ в явном виде. Можно видеть, что $rank(M_2)$ равен 21; в качестве свободных переменных можем взять $c_{4,4}, c_{4,5}, c_{5,4}, c_{5,5}$. Выразим оставшиеся переменные через свободные и получим (16). ■

Таким образом, с помощью формул (6), (7) и теоремы 1 мы можем построить требуемую функцию g из (1), удовлетворяющую всем условиям из (2). Пример экстраполяции с $[0, 1] \times [0, 1]$ на $[-1, 2] \times [-1, 2]$ (границы экстраполяции выбраны произвольным образом) для функции $f(x, y) = 2 \cdot \exp(x^4) \cdot \cos y + \exp(y^3) \cdot \sin x + 13$ при $k = 3$ изображен на Рис. 1.

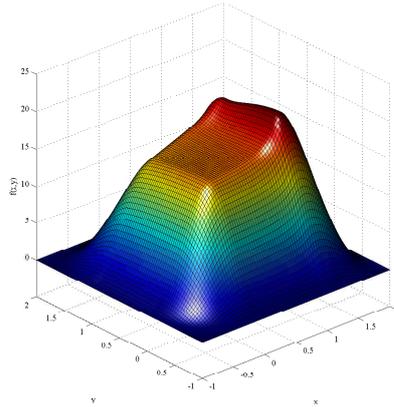


Рис. 1: Пример экстраполяции функции $f(x, y) = 2 \cdot \exp(x^4) \cdot \cos y + \exp(y^3) \cdot \sin x + 13$, $(x, y) \in [0, 1] \times [0, 1]$

3 Применение экстраполяции для сжатия изображений с потерями

В качестве среды для программной реализации экстраполяции и применения алгоритмов сжатия изображений на основе аппарата теории всплесков был выбран пакет прикладных программ Matlab с расширением Wavelet Toolbox указанных во втором разделе версий. Пусть у нас есть цифровое растровое изображение, которое можно рассматривать как функцию двух переменных $X = X(i, j), (i, j) \in \mathbb{Z}^2$, определенную в точках некоторого прямоугольника. Мы будем рассматривать только монохромные (с глубиной цвета 8 бит на пиксель) изображения размера $2^l \times 2^l$ ($l \in \mathbb{N}$) пикселей, определенные в целых точках $[0, 2^l - 1] \times [0, 2^l - 1]$. Экстраполируем исходное изображение X размера $2^l \times 2^l$ пикселей, определенное на $[0, 2^l - 1] \times [0, 2^l - 1]$, до изображения размера $2^{l+1} \times 2^{l+1}$ пикселей, определенное на $[-2^{l-1}, 2^l + 2^{l-1} - 1] \times [-2^{l-1}, 2^l + 2^{l-1} - 1]$, с требуемой гладкостью k с помощью формул (6), (7) и теоремы 1 и получим новое изображение $X_E = X_E(i, j), (i, j) \in \mathbb{Z}^2$. В терминах второго раздела данной статьи $a_1 = b_1 = 0, a_2 = b_2 = 2^l - 1, A_1 = B_1 = -2^{l-1}, A_2 = B_2 = 2^l + 2^{l-1} - 1$. На Рис. 2 показан пример экстраполяции при $k = 1$, где из цифрового растрового монохромного изображения X размера 512×512 пикселей (область, ограниченная красной рамкой) получено новое изображение X_E размера 1024×1024 пикселей.

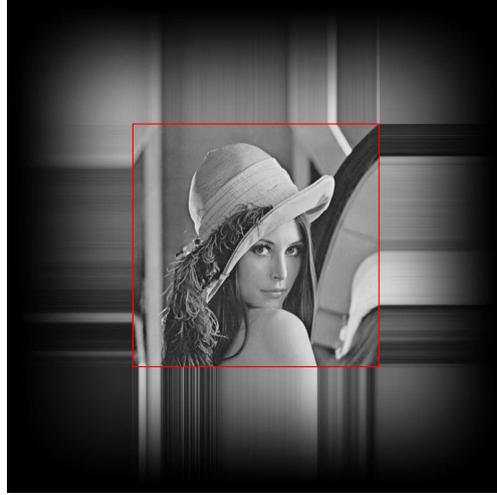


Рис. 2: Пример экстраполяции для изображения Lena.png

После предварительной обработки исходного изображения будем применять к X и X_E четыре основных на всплесках алгоритма сжатия:

- *EZW* (*Embedded Zerotree Wavelet*, построил Шапиро в работе [6]);
- *SPIHT* (*Set Partitioning in Hierarchical Trees*, построили Саид и Перлман [7]);
- *STW* (*Spatial-orientation Tree Wavelet*, построили Саид и Перлман [8]);
- *WDR* (*Wavelet Difference Reduction*, построили Тянь и Уэллс [9], [10], [11]).

Отметим, что алгоритмы *SPIHT* и *STW* являются измененными версиями алгоритма *EZW*. Как правило, в качестве характеристик, по которым сравниваются алгоритмы сжатия, выбирают показатели *CR* и *PSNR*:

- *CR* (*Compression Ratio*) = $\frac{S_c}{S_0}$ (измеряется в процентах), где S_0 – объём исходных данных, а S_c – объём сжатых данных;
- *PSNR* (*Peak Signal-to-Noise Ratio*) = $10 \cdot \log_{10} \frac{255^2}{MSE}$ (измеряется в децибелах), где *MSE* (*Mean Squared Error*) = $\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |X(i, j) - X^{comp}(i, j)|^2$, $X = X(i, j)$ – исходное изображение, $X^{comp} = X^{comp}(i, j)$ – восстановленное после сжатия изображение, $m \times n$ – размер изображений X и X^{comp} .

Однако, так как само X_E изначально нигде не хранится и содержит незначимые пиксели, нужные лишь в процессе сжатия и отбрасываемые после восстановления изображения, то показатель CR не подходит для сравнения алгоритмов сжатия изображений с предварительной обработкой и без неё. Поэтому вместо CR мы будем использовать показатель FS (*File Size*) – размер файла сжатого изображения с предварительной обработкой (FS_E) и без предварительной обработки (FS). MSE для алгоритмов с предварительной обработкой изображений будем вычислять следующим образом: $MSE_E := \frac{1}{2^{l+1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} |X(i, j) - X_E^{comp}(i, j)|^2$, где X_E^{comp} – изображение, полученное восстановлением после сжатия изображения X_E и взятия только той его части, которая задана на $[0, 2^l - 1] \times [0, 2^l - 1]$ (то есть изображения $2^l \times 2^l$ пикселей). Для алгоритмов без предварительной обработки $MSE := \frac{1}{2^{l+1}} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} |X(i, j) - X^{comp}(i, j)|^2$. Тогда $PSNR_E = 10 \cdot \log_{10} \frac{255^2}{MSE_E}$, $PSNR = 10 \cdot \log_{10} \frac{255^2}{MSE}$.

Для того, чтобы сравнить перечисленные алгоритмы сжатия в двух вариациях: без предварительной обработки и с предварительной обработкой, будем подбирать такие значения параметров в самих алгоритмах, чтобы при $FS_E \leq FS$ вариант с предварительной обработкой давал большее PSNR (то есть лучшее качество изображения). Все четыре алгоритма сжатия реализованы с помощью функции *wcompress* из библиотеки Wavelet Toolbox. Основным параметром функции *wcompress* является максимальное число итераций соответствующего алгоритма сжатия – *maxloop*. Его увеличение ведет к улучшению качества изображения (к большему PSNR), но к увеличению размера файла изображения (к большему FS). Еще один важный параметр – *level*, который определяет уровень, до которого происходит всплеск-разложение.

Все численные эксперименты проведены для монохромного изображения *Lena.png* размера 512×512 пикселей с размером файла 167034 байт, а также для его экстраполированной ($k = 1$) версии размера 1024×1024 . В приведенных алгоритмах сжатия на этапе всплеск-разложения были использованы все базы всплесков доступные в расширении Wavelet Toolbox: дискретные всплески Мейера (*dmey*), всплески Добеши (*db1 – db45*), койфлеты (*coif1 – coif5*), симмлеты (*sym1 – sym45*), биортогональные всплески (*bior1.1, bior1.3, bior1.5, bior2.2, bior2.4, bior2.6, bior2.8, bior3.1, bior3.3, bior3.5, bior3.7, bior3.9, bior4.4, bior5.5, bior6.8*), обратные биортогональные всплески (*rbio1.1, rbio1.3, rbio1.5, rbio2.2, rbio2.4, rbio2.6, rbio2.8, rbio3.1, rbio3.3, rbio3.5, rbio3.7, rbio3.9, rbio4.4, rbio5.5, rbio6.8*). Далее будут приведены результаты только для тех базисов всплесков, которые дали улучшение в выбранных показателях сжатия в варианте с предварительной обработкой изображения.

В Таб. 2 приведены результаты численных экспериментов для алгоритма сжатия EZW (левая часть таблицы – вариант без предварительной обработки, правая часть – с предварительной обработкой). Для дискретных всплесков Мейера (*dmey*) всплеск-разложение проводилось до уровня $level = 5$, число итераций алгоритма сжатия без предварительной обработки $maxloop = 12$, с предварительной обработкой $maxloop_E = 11$. Для остальных базисов всплесков (*db3, db4, db10, db12, coif5, sym3, sym4, sym8, sym10, sym12, sym13, sym23, sym25, sym27, bior4.4, bior5.5, rbio4.4*) $level = 4$, $maxloop = 11$, $maxloop_E = 10$.

В Таб. 3 приведены результаты численных экспериментов для алгоритма сжатия SPIHT (левая часть таблицы – вариант без предварительной обработки, правая часть – с предварительной обработкой). Для дискретных всплесков Мейера (*dmey*) $level = 5$, $maxloop = 13$, $maxloop_E = 12$. Для остальных базисов всплесков (*db3, db4, db10, db12, coif5, sym3, sym4, sym8, sym10, sym12, sym13, sym23, sym25, bior4.4, bior5.5, rbio4.4*) $level = 4$, $maxloop = 12$, $maxloop_E = 11$.

В Таб. 4 приведены результаты численных экспериментов для алгоритма сжатия WDR (левая часть таблицы – вариант без предварительной обработки, правая часть – с предварительной обработкой). Для дискретных всплесков Мейера (*dmey*) $level = 5$, $maxloop = 11$, $maxloop_E = 10$. Для остальных базисов всплесков (*coif5, bior4.4, bior5.5*) $level = 4$, $maxloop = 10$, $maxloop_E = 9$.

Для алгоритма сжатия STW ни один из базисов всплесков при всевозможных параметрах не дал положительных результатов в варианте сжатия с предварительной обработкой изображения. Из Таб. 2, 3, 4 видно, что наибольшее значение $PSNR_E - PSNR$ для алгоритмов EZW, SPIHT, WDR получено с использованием биортогональных всплесков *bior5.5*, а наибольшее значение $FS_E - FS$ для тех же алгоритмов получено с использованием дискретных всплесков Мейера *dmey*.

Таблица 2: Результаты численных экспериментов для алгоритма сжатия EZW

Всплески	Показатели сжатия		Всплески	Показатели сжатия	
	FS (байт)	$PSNR$ (дБ)		FS_E (байт)	$PSNR_E$ (дБ)
db3	157041	40.8522	db3	121002	42.7873
db4	156024	36.8092	db4	120931	37.4881
db10	156293	40.6419	db10	122333	42.5396
db12	156942	40.7429	db12	122830	42.7604
coif5	153755	38.3765	coif5	118125	42.6963
sym3	157043	40.8522	sym3	121004	42.7873
sym4	156004	40.7542	sym4	120142	42.6876
sym8	154422	40.6459	sym8	118491	42.5209
sym10	154312	40.7670	sym10	118207	41.4837
sym12	154322	40.7153	sym12	118269	42.5855
sym13	154213	40.7887	sym13	118600	42.6128
sym23	154423	39.9929	sym23	119873	42.5573
sym25	154899	39.4138	sym25	119071	41.4616
sym27	155281	40.8059	sym27	120551	42.6504
dmey	154371	35.6957	dmey	115462	42.4252
bior4.4	151223	37.9625	bior4.4	114871	42.5363
bior5.5	154840	33.8310	bior5.5	118983	42.1763
rbio4.4	164147	40.7066	rbio4.4	129227	42.3120

Таблица 3: Результаты численных экспериментов для алгоритма сжатия SPIHT

Всплески	Показатели сжатия		Всплески	Показатели сжатия	
	FS (байт)	$PSNR$ (дБ)		FS_E (байт)	$PSNR_E$ (дБ)
db3	105561	40.6507	db3	91923	41.9744
db4	104675	36.7293	db4	91884	37.2345
db10	104408	40.4485	db10	92269	41.7706
db12	104771	40.5452	db12	92673	41.9442
coif5	103330	38.2626	coif5	90068	41.9338
sym3	105563	40.6507	sym3	91924	41.9744
sym4	104680	40.5580	sym4	91076	41.9092
sym8	103448	40.4553	sym8	89888	41.7851
sym10	103348	40.5700	sym10	89747	40.8933
sym12	103394	40.5214	sym12	89760	41.8377
sym13	103260	40.5918	sym13	89646	41.8677
sym23	103474	39.8281	sym23	90492	41.8076
sym25	103468	39.2689	sym25	89887	40.8666
dmey	101152	35.6334	dmey	81015	41.6875
bior4.4	102195	37.8512	bior4.4	88231	41.7458
bior5.5	105452	33.7903	bior5.5	92536	41.3213
rbio4.4	108951	40.5387	rbio4.4	96576	41.6475

Таблица 4: Результаты численных экспериментов для алгоритма сжатия WDR

Всплески	Показатели сжатия		Всплески	Показатели сжатия	
	FS (байт)	$PSNR$ (дБ)		FS_E (байт)	$PSNR_E$ (дБ)
coif5	101088	37.5026	coif5	63700	38.8236
dmey	102063	35.2019	dmey	61629	38.6860
bior4.4	98239	37.1329	bior4.4	61027	38.6562
bior5.5	101805	33.4843	bior5.5	60774	38.2070

Благодарности

Работа выполнена за счет гранта Российского научного фонда (проект 14-11-00702).

Список литературы

- [1] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
- [2] A. Cohen, I. Daubechies, P. Vial. Wavelets on the interval and fast wavelet transforms. *Appl. Comp. Harm. Anal.*, 1(1):54–81, 1992.
- [3] A. Cohen, I. Daubechies, J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992.
- [4] Y. Meyer. Wavelets on the interval. *Revista Matemática Iberoamericana*, 7(2):115–133, 1991.
- [5] A.A. Privalov. *Theory of approximation of functions*. Izd-vo Saratovskogo gos-go un-ta, Saratov, 1990 (in Russian). = A.A. Привалов. *Теория приближения функций*. Изд-во Саратовского гос-го ун-та, Саратов, 1990.
- [6] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Processing*, 41:3445–3462, 1993.
- [7] A. Said, W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology*, 6(3):243–250, 1996.
- [8] A. Said, W.A. Pearlman. Image compression using the spatial-orientation tree. *IEEE Int. Symp. on Circuits and Systems, Chicago, IL*, 279–282, 1993.
- [9] J. Tian, R.O. Wells, Jr. A lossy image codec based on index coding. *IEEE Data Compression Conference, DCC'96*, 456, 1996.
- [10] J. Tian, R.O. Wells, Jr. Embedded image coding using wavelet-difference reduction. *Wavelet Image and Video Compression, Topiwala, P., Ed., Kluwer Academic, Norwell, MA*, 289–301, 1998.
- [11] J. Tian, R.O. Wells, Jr. Image data processing in the compressed wavelet domain. *3rd International Conference on Signal Processing Proc., Yuan, B. and Tang, X., Eds., Beijing, China*, 978–981, 1996.

Image preprocessing for improving performance of lossy compression algorithms based on wavelets

Dmitriy A. Yamkovoy

Krasovskii Institute of Mathematics and Mechanics (Yekaterinburg, Russia)

Keywords: smooth extrapolation, image compression, lossy compression.

In this paper we developed extrapolation method of $2k$ -smooth functions in a neighborhood of the rectangle Ω to the $\mathbb{R}^2 \setminus \Omega$ with k -smooth-preserving and finite support properties. Also, advantages of new construction applying to lossy image compression using wavelet theory in comparison with other methods are discussed according to basic characteristics of image compression.