

УДК 004.627, 004.921, 621.396.946

ЭФФЕКТИВНОЕ СЖАТИЕ ИЗОБРАЖЕНИЙ НА БАЗЕ ДИФФЕРЕНЦИАЛЬНОГО АНАЛИЗА

А. Ю. Гришенцев

**Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики
сайт автора: <http://www.moveinfo.ru/>**

Получена 1 октября 2012 г.

Аннотация. В данной работе предложен метод сжатия изображений на основе анализа дифференциальной структуры. Изначально была поставлена задача разработки метода сжатия специфических высококонтрастных космических снимков с целью сокращения дорогостоящего трафика связи с космическими аппаратами. В результате разработки выполненной на базе СПб НИУ ИТМО был получен новый метод сжатия изображений на основе анализа дифференциальной структуры. Полученный метод показал возможность применения к широкому классу графических изображений в частности, и полевым структурам, в общем. В работе рассмотрены концепция метода, варианты программной реализации, дополнительные приёмы, позволяющие повысить эффективность метода, а так же примеры использования и некоторые численные оценки. Данный метод сжатия может быть применён для статических и динамических (видео) изображений. Заложенные в основу концепции сжатия принципы позволяют создавать многопоточные вычислительные решения как на аппаратно-программном, так и только на аппаратном уровне. Вычислительную сложность ядра алгоритма сжатия (в самой простой реализации) можно оценить как произведение линейных размеров статического изображения. Степень сжатия изображений на базе анализа дифференциальной структуры сопоставима со степенью сжатия JPEG, при этом качество полученных изображений по численным критериям (MSE , $PSNR$, SNR) выше, что может быть существенным для машинной

обработки изображений. Для сжатия требуется относительно малое число операций, что может положительно сказаться на разгрузке вычислительных мощностей космических аппаратов, осуществляющих передачу изображений.

Ключевые слова: сжатие, компрессия, сжатие данных, сжатие изображений, алгоритм сжатия, анализ дифференциальной структуры изображения, дифференциальное сжатие.

Abstract: In this paper we propose a method for image compression based on the analysis of the differential structure. Was initially given the task of developing a method of compression specific high contrast satellite images in order to reduce costly traffic due to the spacecraft. As a result of the development made on the basis of SPb ITMO was obtained new image compression method based on the analysis of the differential structure. The resulting method has shown the possibility of applying to a wide range of graphic images in particular, and field in general. The paper discusses the concept of method, options software implementation, advanced techniques that improve the efficiency of the method, as well as examples and numerical estimates. This method of compression can be used for static and dynamic (video) images. Laid the basis for the conception principles allow you to create multi-threaded computing solutions both hardware and software, and only hardware. Computational complexity of the core compression algorithm (in its simplest implementation) can be estimated as the product of the linear dimensions of static images. Image compression based on the analysis of the differential structure is comparable to the degree of compression JPEG, the quality of the images obtained by the numerical criterion (MSE , $PSNR$, SNR) above, that may be important for computer image processing. To compress a relatively small number of operations that can have a positive impact on the loading of computing power spacecraft carrying the transfer of images.

Key words: compression, compression, data compression, image compression, the compression algorithm, the analysis of the differential image structure, differential compression.

1. Введение

В современном мире происходит генерация контента в значительных объёмах, наибольшую часть занимают различные медиа-данные. При непрерывном росте объёма данных требующих передачи, полоса пропускания стандартных радиоканалов связи остаётся неизменной, что в свою очередь порождает существенные проблемы. Одним из путей решения является компрессия данных и передача при помощи соответствующих протоколов связи. Компрессия данных так же позволяет снизить используемые для хранения ресурсы памяти вычислительных машин.

Наибольший интерес для разработки способов компактного хранения представляют сигналы, относящиеся к графическим, т.е. передающие различные виды изображений, например: фотографии, видеофильмы, сигналы цифрового телевиденья и прочие многомерные сигналы, сопряжённые с передачей графической информации. В большинстве случаев передача и хранение графических данных потребляет существенные ресурсы при значительном количестве избыточной информации. Применение только статистических способов сжатия как алгоритм Хаффмана, арифметическое кодирование и подобных, в большинстве случаев не даёт существенного эффекта. Поэтому для эффективного сжатия к цифровому сигналу применяют несколько последовательных преобразований. Часть преобразований основана на рассмотрении сигнала как поля, к которому применимы: спектральный анализ, анализ геометрической структуры, корреляционный анализ, дифференциальный анализ. Таким образом, изображения рассматриваются как полевые структуры, а цифровые изображения – дискретные полевые структуры, соответственно.

Различная природа цифровых сигналов определяет их особенности. Сигналы принято разделять на группы по преимущественному сосредоточению информации в частотной, либо в пространственной области. При таком

разделении для исключения потери значимой информации необходимо учитывать специфику конкретного сигнала или некоторой группы сигналов.

Частотные характеристики сигнала могут быть получены с помощью способов частотной декомпозиции (Фурье, вейвлет и др.). Пространственные характеристики сигнала могут быть получены с помощью геометрической интерпретации сигнала, определения его производных, вейвлет анализа, оконного преобразования Фурье. Вейвлет анализ, как и оконное преобразование Фурье, интересно тем, что позволяют получить не только частотную характеристику сигнала, но пространственную локализацию спектра.

Используя особенности концентрации информации, можно исключать некоторую «избыточную» часть из частотного или пространственного описания сигнала, приводящую к допустимой потере информации, т.е. допустимым искажениям.

Методы получения частотных и пространственных характеристик сигнала позволяют синтезировать компактную форму информационной составляющей сигнала. Множество различных способов перевода информации в компактную форму не имеют универсальности для различных видов цифровых сигналов. Таким образом, конкретный метод получения компактной формы, хорошо применимый к одному сигналу или группе сигналов, может быть малоэффективен для другого сигнала или группы сигналов. Не случайно в современном мире цифровой обработки сигналов присутствует такое разнообразие форматов данных.

В данной работе будет рассмотрен новый метод, позволяющий осуществить сжатие изображений (с потерями или без потерь) на базе их дифференциального анализа как полевых структур.

2. Предпосылки к формированию компактной формы на базе анализа дифференциальной структуры

Рассмотрение цифровых сигналов как полевых структур позволяет использовать численные методы расчёта полей к анализу и преобразованию

цифровых сигналов. Наиболее интересны методы частотного и дифференциального анализа. Основы методов частотного анализа цифровых сигналов, в том числе с целью последующего сжатия, хорошо разобраны в ставшей уже классической литературе по цифровой обработке сигналов [1-5], поэтому сконцентрируем внимание на рассмотрении пространственного анализа сигналов. Существует ряд способов формирования компактного вида цифровых сигналов путём анализа пространственной структуры. В частности для получения компактного вида изображений применяют векторизацию, т.е. в ходе морфологического анализа изображения выявляют графические примитивы (прямые, окружности, прямоугольники и т.д.), совокупность описания которых позволяет в некоторых случаях получить компактную форму. Данный способ не является универсальным, т.е. не все изображения можно, с целью сжатия, эффективно разделить на графические примитивы. Разделением на графические примитивы наиболее эффективно можно сжимать техническую графику (чертежи, эскизы и др.) и некоторые виды изображений, объекты которых имеют чёткие границы и относительно простые формы. Морфологический анализ изображений в большинстве случаев является достаточно сложной вычислительной задачей.

Следующим способом является фрактальное сжатие, основанное на том, что изображение можно представить более компактно, с помощью коэффициентов системы итерируемых функций (фракталов) [6]. С точки зрения вычислений, фрактальное сжатие является очень затратным способом, но на некоторых образцах изображений показывает очень хорошие результаты. Известен способ, в котором из изображения, применяя метод «brush fire» (лесной пожар), формируют скелет многоугольника, толщиной 1-2 пикселя, являющийся диаграммой Вороного [7]. Данный способ так же достаточно затратен с точки зрения вычислений и не получил широкого распространения. Существуют и другие способы пространственного анализа цифровых сигналов с целью получения компактной формы.

Способ дельта-кодирования. В самом простом случае дельта кодирование строится следующим образом: для элементов цифрового сигнала $f[x]$, $x \in X$, вычисляется ряд дельта-кода $d[x] = f[x+1] - f[x]$, $x \in (X-1)$, в силу того, что соседние элементы исходного сигнала $f[x]$ обычно отличаются друг от друга значительно меньше диапазона допустимых значений $f[x]$, полученный ряд дельта-кода $d[x]$ может быть эффективно сжат [8]. Фактически значения ряда дельта-кода $d[x]$ являются производной $f[x]$. Для восстановления исходного сигнала, кроме самого ряда дельта-кода, необходимо знать начальное условие. Рассмотрим пример получения и некоторые особенности дельта-кода: диаграмма (рис. 1) и ряды численных значений (таблица 1), соответственно. Очевидно, что значения дельта-кода имеют меньшую амплитуду и для их кодирования потребуется меньшее число разрядов. Значения дельта-кода будут чаще повторяться и, следовательно, к ним может быть эффективно применено статистическое кодирование.

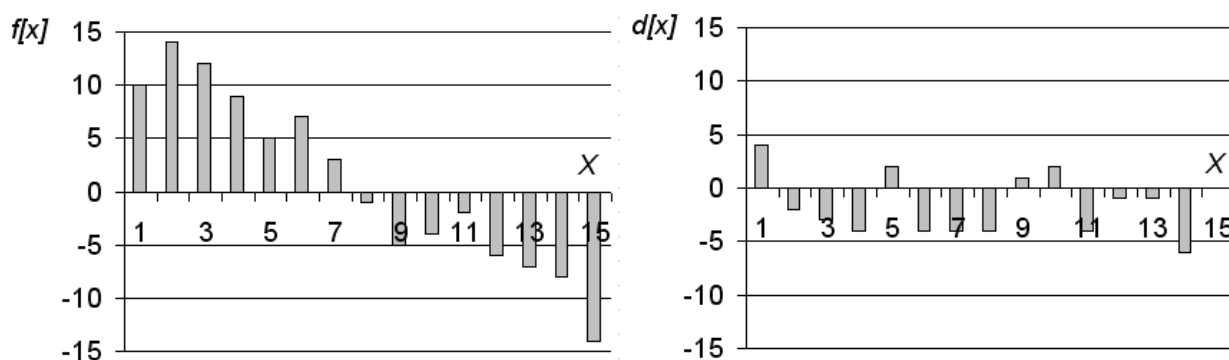


Рис. 1. Исходный сигнал $f[x]$ и дельта-код $d[x]$.

Таблица 1. Значения функции $f[x]$ и её дельта-кода $d[x]$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f[x]$	10	14	12	9	5	7	3	-1	-5	-4	-2	-6	-7	-8	-14
$d[x] = f[x+1] - f[x]$	4	-2	-3	-4	2	-4	-4	-4	1	2	-4	-1	-1	-6	14

Развивая идею дельта-кодирования, можно заметить, что большинство сигналов, полученных в результате регистрации реальных физических

процессов, например, фотографирование объектов реального мира, могут быть описаны с той или иной степенью точности уравнениями. Дифференциальные уравнения – наиболее распространённый тип уравнений, описывающий физические процессы окружающего нас мира. Следовательно, дифференциальные уравнения могут быть применены для описания цифровых сигналов, являющихся образами реальных физических явлений.

Рассмотрим основные типы уравнений математической физики с частными производными второго порядка для двух- и трёх мерных пространств (x, y – координаты пространства, t – координата времени) [9].

Волновые уравнения:

$$\frac{\partial^2 f(x, t)}{\partial t^2} = a^2 \frac{\partial^2 f(x, t)}{\partial x^2}, \quad (1)$$

$$\frac{\partial^2 f(x, y, t)}{\partial t^2} = a^2 \left(\frac{\partial^2 f(x, y, t)}{\partial x^2} + \frac{\partial^2 f(x, y, t)}{\partial y^2} \right). \quad (2)$$

К уравнениям данного типа приводит рассмотрение процессов поперечных колебаний струны, продольных колебаний стержней, электрических колебаний в проводах, крутильных колебаний валов, колебаний газа и т.д. Отметим, что не все процессы, квалифицируемые современной физикой как волновые, описываются приведёнными волновыми уравнениями, примером может быть распространение волн в нелинейной среде, солитоны.

Уравнения теплопроводности, или уравнения Фурье:

$$\frac{\partial f(x, t)}{\partial t} = a^2 \frac{\partial^2 f(x, t)}{\partial x^2}, \quad (3)$$

$$\frac{\partial f(x, y, t)}{\partial t} = a^2 \left(\frac{\partial^2 f(x, y, t)}{\partial x^2} + \frac{\partial^2 f(x, y, t)}{\partial y^2} \right). \quad (4)$$

К этим уравнениям приводит рассмотрение процессов теплопроводности, фильтрации жидкости и газов в пористой среде, некоторые задачи теории вероятностей и т.д.

Уравнения Лапласа и Пуассона:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} = \varphi(x, y), \quad (5)$$

$$\frac{\partial^2 f(x, y, z)}{\partial x^2} + \frac{\partial^2 f(x, y, z)}{\partial y^2} + \frac{\partial^2 f(x, y, z)}{\partial z^2} = \varphi(x, y, z). \quad (6)$$

Уравнения этого типа позволяет производить моделирование электромагнитных полей, стационарных тепловых состояний, задач гидродинамики, диффузии и т.д. Уравнения Пуассона (5) и (6) при равенстве правой части нулю ($\varphi(x, y) = 0$ и $\varphi(x, y, z) = 0$ для всех значений x, y, z) обращаются в уравнения Лапласа.

Таким образом, большинство процессов (стационарных и динамических), которые являются первичными источниками сигналов, имеют аналитическое описание в виде дифференциальных уравнений. Важно заметить, что дифференциал и производная функции связаны с рядом Фурье, в соответствии с теоремой о почленном дифференцировании тригонометрического ряда Фурье [10].

Можно записать выражение, связывающее производную порядка p (где p любое из чисел $p = 1, 2, \dots, m$) и члены ряда Фурье функции $f(x)$:

$$f^{(p)}(x) = \sum_{k=1}^{\infty} k^p \left[a_k \cos\left(kx - \frac{\pi p}{2}\right) + b_k \sin\left(kx - \frac{\pi p}{2}\right) \right], \quad (7)$$

где: a_n, b_n – коэффициенты ряда Фурье функции $f(x)$. Из уравнения (7) следует, что производные более высокого порядка более зависимы от высокочастотных составляющих ряда Фурье.

Основываясь на приведённых фактах, перейдём к рассмотрению способа сжатия, в основу которого положен анализ дифференциальной структуры цифрового сигнала. Под дифференциальной структурой подразумевается производная, обычно до третьего порядка, данного цифрового сигнала, а для многомерных сигналов частные производные, соответственно.

3. Анализ дифференциальной структуры и формирование паттерна краевых условий цифрового сигнала

Определим некоторые понятия используемые далее. Совокупность условий, описывающих состояние системы в пространстве на границах и некоторых внутренних областях, называют граничными условиями. Совокупность, описывающая состояние системы в некоторые моменты времени (обычно начальные $t = 0$), называют начальными условиями. Совокупность граничных и начальных условий называется краевыми условиями.

Пусть в области W задан сигнал $f(R^n)$, предположим, что в некоторой части заданной области $U \subset W$ сигнал удовлетворяет дифференциальному уравнению вида:

$$Af(R^n) = u(R^n), \quad (8)$$

где: A – некоторый дифференциальный оператор, $u(R^n)$ – правая часть. Краевые условия $p(R^n)$, заданные в области $V \subset W$, таким образом, что: $U \cup V = W$. Функцию $p(R^n)$, заданную в области V , будем называть паттерном исходного сигнала $f(R^n)$. Зная уравнение (8) и паттерн сигнала $p(R^n)$, можно восстановить исходный сигнал $f(R^n)$ в области U . Таким образом, суть анализа дифференциальной структуры сигнала $f(R^n)$, состоит в выборе анализирующего уравнения (или уравнений) (8) и отыскании паттерна $p(R^n)$.

Определение краевых условий, путём анализа дифференциальной структуры заданного сигнала, является обратной задачей вычислительной математики.

Рассмотрим формирование паттерна на примере одномерного сигнала $f[x]$. Положим, что $f(x)$ есть непрерывный прообраз дискретного образа некоторого сигнала $f[x]$, при этом $f(x)$ описывается во всех областях U анализирующим уравнением вида:

$$\frac{d^2 f(x)}{dx^2} = 0. \quad (9)$$

Переходя к дискретному образу $f[x_i]$ и полагая, что он задан на сетке $x_0, x_1, \dots, x_{n-1} \in \mathbf{W}$, с равномерным шагом $h = x_{i+1} - x_i$, где $i \in 0 \dots n-1$, для всех $x_i \in \mathbf{U}$, можно записать уравнение (9) в виде:

$$\frac{d^2 f(x)}{dx^2} \approx \frac{\frac{f[x_{i+1}] - f[x_i]}{h} - \frac{f[x_i] - f[x_{i-1}]}{h}}{h} = \frac{f[x_{i+1}] - 2f[x_i] + f[x_{i-1}]}{h^2} = 0. \quad (10)$$

Выражая из (10) $f[x_i]$, получим:

$$f[x_i] = \frac{f[x_{i+1}] + f[x_{i-1}]}{2}. \quad (11)$$

Вычислять паттерн $p[x_i]$ будем по следующей схеме.

1. В начале положим, что во всей области $\mathbf{W}$ все значения паттерна равны значениям сигнала т.е.: $p[x_i] = f[x_i]$.

2. Далее выберем некоторое числовое значение Δ -критерия, по которому будем производить исключение элементов из паттерна.

3. Следующим шагом будем вычислять для всех значений $f[x_i]$ в области $\mathbf{W}$ значения в соответствии с выражением, полученным с учётом (11):

$$s = \left| f[x_i] - \frac{f[x_{i+1}] + f[x_{i-1}]}{2} \right|, \quad (12)$$

если условие $s < \Delta$ истинно, то значение $p[x_i]$ (для соответствующего x_i) исключается из паттерна, в противном случае, элемент остаётся в паттерне.

В результате выполнения указанной последовательности действий, мы получим паттерн сигнала $f[x_i]$.

На изображении (рис. 2) показан пример исходного сигнала (data) и паттерна (patt).

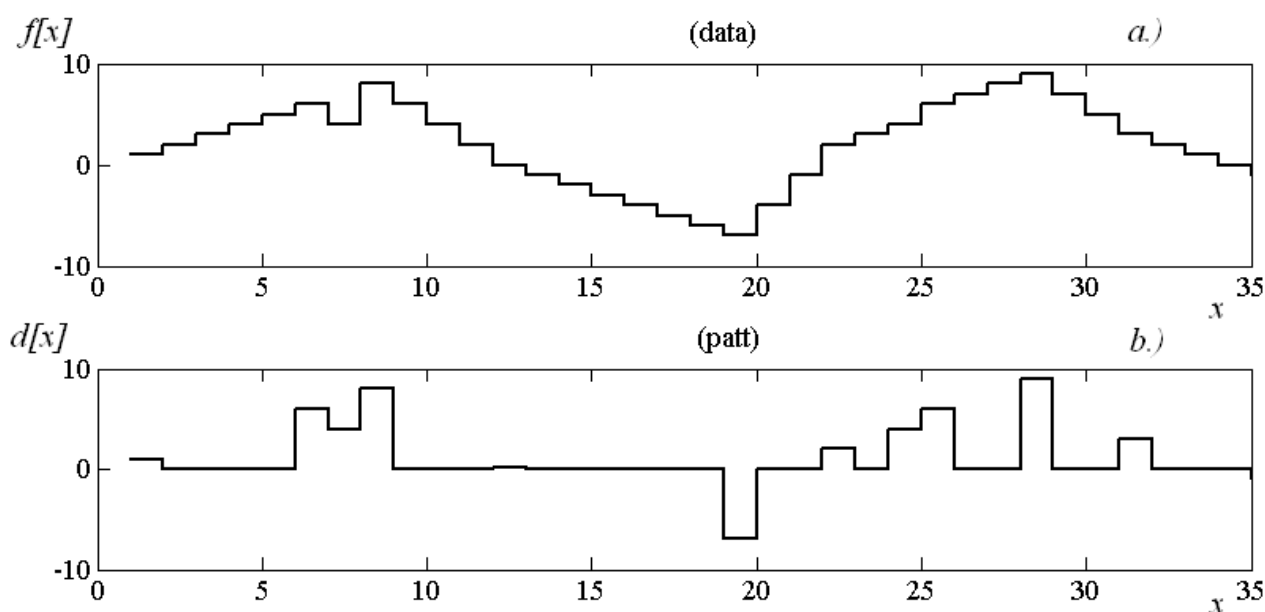


Рис. 2. Исходный сигнал (data) и паттерн (patt).

Очевидно, что паттерн (рис. 2) содержит значительно большее число равнозначных элементов, чем исходный сигнал, что позволяет его эффективнее сжать. В качестве признака исключения элемента из паттерна соответствующее значение возможно, например, приравнивать нулю. Такой принцип может вызвать неоднозначные («спорные») ситуации, например, если в исходном сигнале присутствуют нулевые значения, т.к. для восстановления сигнала необходимо точно знать принадлежит конкретный элемент паттерна краевым условиям или нет. Избежать такой ситуации поможет изменение «спорных» значений в исходном сигнале на некоторую определённую величину в зависимости от формата и допустимого диапазона цифрового сигнала. В качестве признака того, что элемент не является частью краевых условий, можно использовать величины выходящие за диапазон значений сигнала, но допустимые форматом данных.

Изменяя значение величины Δ -критерия, можно управлять числом элементов вошедших в паттерн, от этого будет зависеть количество исключаемой информации и степень сжатия, соответственно. Для точного восстановления сигнала потребуется принять значение Δ -критерия меньше шага квантования сигнала и в некоторых случаях сохранять, например, в виде

битовой маски, данные позволяющие восстановить нулевые (или другие «спорные») значения в сигнале. На изображении (рис. 3) показан исходный сигнал (рис. 3, а.) и его поле паттернов (рис. 3, б.). Поле паттернов получено в результате объединения серии паттернов (для различных значений Δ -критерия) в один массив. Чёрный цвет соответствует нулевым элементам паттерна, не являющимися краевыми условиями, прочие (оттенки серого) являются краевыми условиями. При увеличении значения Δ -критерия происходит уменьшение числа элементов паттерна, принадлежащих множеству краевых условий.

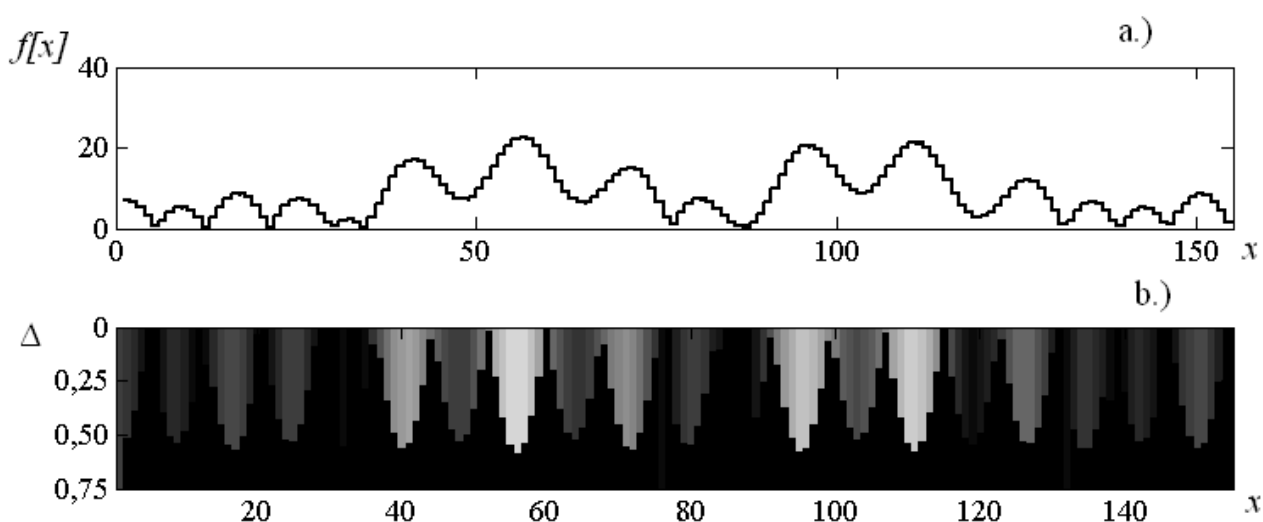


Рис. 3. Сигнал (а.) и его поле паттернов (б.) в зависимости от Δ -критерия.

Паттерн, содержащий относительно небольшое число элементов, являющихся краевыми условиями, позволяющими достаточно точно восстановить исходный сигнал, может быть получен только в случае хорошо подобранного уравнения для формирования паттерна, что в большинстве случаев возможно лишь для гладких функций. Следовательно, применение данного способа для сжатия шумоподобных сигналов (с высокой энтропией) неэффективно.

Изменяя порядок производных дифференциального оператора A (8) и выбирая некоторые функции правой части $u(R^n)$, можно влиять на качество восстановления сигнала и число исключённых элементов из паттерна.

Восстановление исходного сигнала осуществляется с помощью решения выбранного ранее в качестве анализирующего дифференциального уравнения и краевых условий, содержащихся в паттерне. Численное решение можно осуществить, например, с помощью метода конечных разностей (МКР).

4. Сжатие многомерных цифровых сигналов с помощью анализа их дифференциальной структуры

Наиболее востребовано сжатие для данных называемых медиа-контент, к ним относятся статические и динамические (видео) изображения, аудио данные и др. В последнее время активно развиваются различные форматы видео данных [5,11], на базе которых синтезируются объёмные видео-сцены изображения высокой четкости и др. Цифровые изображения это один из видов сигналов, к которым можно эффективно применять сжатие за счёт анализа дифференциальной структуры. Многие изображения содержат достаточно значительные поля градиентных переходов (цвета, яркости); контрастные границы объектов в пространстве для статических и во времени и пространстве для динамических изображений. Границы объектов, содержащихся в изображениях, можно рассматривать как краевые условия, а поля достаточно плавных переходов, как области, удовлетворяющие выбранному анализирующему дифференциальному уравнению.

В качестве примера рассмотрим последовательно все этапы сжатия и восстановления статического изображения при помощи анализа дифференциальной структуры. Далее приведено краткое описание последовательности действий для сжатия и восстановления изображения.

Сжатие изображения:

- выбор дифференциального анализирующего уравнения (или системы) и цветового пространства для формирования паттерна;
- если необходимо преобразование цветового пространства;
- формирование паттерна (граничных условий);

– компрессия паттерна и дополнительной информации для восстановления сигнала.

Восстановление изображения:

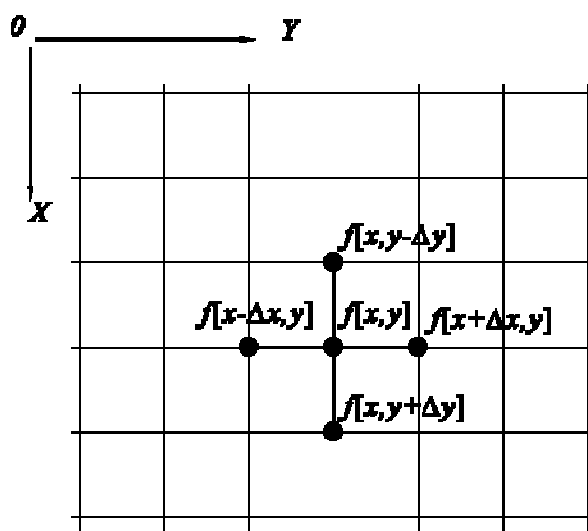
- декомпрессия паттерна и дополнительной информации;
- восстановление изображения при помощи решения дифференциального уравнения (или системы) с учётом граничных условий;
- если необходимо, преобразование цветового пространства.

Подробно рассмотрим каждый из перечисленных выше пунктов. Первая задача это выбор анализирующего уравнения (или системы) для формирования паттерна и последующего восстановления двумерного сигнала изображения $f[x, y]$, заданного на плоскости XOY . Остановимся на уравнении Лапласа, т.к. оно хорошо описывает плавные переходы (градиент цвета и яркости) в областях между граничными условиями:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} = 0. \quad (13)$$

Выбрав шаблон дифференцирования для функции (рис. 4), переходим к дискретной форме $f[x, y]$. Считая что шаг сетки, с учётом индексации пикселей изображения, $\Delta x = \Delta y = 1$, уравнение (13) можно записать в виде:

$$\begin{aligned} \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} &\approx \\ &\approx \frac{f[x + \Delta x, y] - 2f[x, y] + f[x - \Delta x, y]}{\Delta x^2} + \frac{f[x, y + \Delta y] - 2f[x, y] + f[x, y - \Delta y]}{\Delta y^2} =. (14) \\ &= f[x + \Delta x, y] + f[x - \Delta x, y] + f[x, y + \Delta y] + f[x, y - \Delta y] - 4f[x, y] = 0 \end{aligned}$$

Рис. 4. Шаблон дифференцирования функции $f[x, y]$.

Выбор шаблона может оказать значительное влияние на конечный результат, в данном случае (рис. 4) выбран шаблон, показавший достаточно высокую эффективность в рамках рассматриваемого метода.

Выражая из (14) $f[x, y]$, получим:

$$f[x, y] = \frac{1}{4} (f[x + \Delta x, y] + f[x - \Delta x, y] + f[x, y + \Delta y] + f[x, y - \Delta y]). \quad (15)$$

Следующая задача – выбор цветового пространства. Практические исследования показали, что наиболее компактный паттерн, следовательно, наиболее высокое сжатие, можно получить в цветовом пространстве RGB и YCbCr (вариация $Y_{709}CbCr$), в зависимости от специфики изображения. Цветовое пространство RGB ассоциировано с особенностями восприятия зрительной системы человека и в нём хорошо передаётся специфика цветового градиента, сохраняются цвета исходного изображения. Пространство YCbCr может более эффективно обеспечить сохранение границ объектов изображения при некоторой потере цветовой информации.

Формирование паттерна является ключевым этапом сжатия изображения. От качества формирования паттерна будет зависеть возможное качество восстановленного сигнала и эффективность сжатия. Отметим некоторые особенности, замеченные при экспериментах с различными подходами к

формированию паттерна. Наиболее высокое качество восстановленного сигнала удаётся получить при генерации паттерна, в котором каждому значащему элементу ставится в соответствие все компоненты цветового пространства. При этом конфигурация паттернов, с точки зрения положения элементов граничных условий, всех цветовых слоёв одинакова, а цвет элементов граничных условий остаётся в полном соответствии с исходным изображением. В результате того, что паттерны всех цветовых компонентов (слоёв) совпадают, возможно, сжимать последовательность серий, не значащих элементов для одного слоя и использовать его как шаблон для восстановления других, что позволяет существенно повысить фактор сжатия.

В паттерн обязательно включаются граничные пиксели изображения (два столбца – справа и слева, и две строки – сверху и снизу). Для исключения элементов из паттерна применяют Δ -критерий на базе сравнения абсолютной величины разности значения вычисленного по (15) и фактического в точке (x, y) , с некоторым заданным коэффициентом. Причём для каждого компонента цветового пространства, возможно своё значение Δ -критерия и, следовательно, различное количество исключённой информации о дифференциальной структуре сигнала.

В программе (листинг 1) приведён пример кода формирования внутренних элементов паттерна, наружные элементы паттерна формируются с помощью копирования. Двумерный массив `data[][]` содержит исходное изображение, каждый элемент массива имеет тип `Image`, паттерн формируется в массиве `pattern[][]` каждый элемент массива имеет тип `ColorSpace`. Изначально в массив `pattern[][]` полностью копируется изображение `data[][]`, далее в циклах (листинг 1) часть элементов исключается (исключением считается приравнивание к нулю). Переменные (`delLo`, `delAv`, `delHi`) содержат значения Δ -критериев сравнения, по каждому цветовому каналу. Маскирование (корректировка) изначально нулевых значений, содержащихся в изображении и скопированных в паттерн, необходима для их отличия от исключённых из

паттерна и исключения «спорных» ситуаций. Отметим, что расчёт производится по несколько видоизменённому выражению (15), к виду:

$$(f[x + \Delta x, y] + f[x - \Delta x, y] + f[x, y + \Delta y] + f[x, y - \Delta y]) - 4 \cdot f[x, y], \quad (16)$$

причём, умножение на четыре в программе производится с помощью поразрядного сдвига влево (<<2), такой подход позволяет выполнять все вычисления в целочисленной арифметике с сохранением точности расчётов. Выполнение целочисленных операций процессор производит существенно быстрее, чем операции над числами с плавающей точкой. От величины Δ -критериев зависит число исключённых из паттерна элементов, а в конечном счёте, и степень сжатия изображения. Чем больше величина Δ -критерия для цветового канала, тем больше значений будет исключено.

Листинг 1. Фрагмент программы формирования внутренних элементов паттерна полноцветного изображения (C++).

```
//=====
...
// цветовая палитра
// покомпонентное представление, в скобках указаны компоненты
// палитр (RGB) и (YCbCr) соответственно
struct ColorSpace {
    uchar byteLo; // компонента младший байт (Lower) (R) (Y)
    uchar byteAv; // компонента средний байт (Average) (G) (Cb)
    uchar byteHi; // компонента старший байт (Hight) (B) (Cr)
};
// структура для сохранения цветов в виде int
struct IImage {
    int iLo; // компонента младший "байт" (Lower) (R) (Y)
    int iAv; // компонента средний "байт" (Average) (G) (Cb)
    int iHi; // компонента старший "байт" (Hight) (B) (Cr)
};
...
for (iy=1; iy<Ysize; iy++) {
    for (ix=1; ix<Xsize; ix++) {
        ...
        // расчёт разностных элементов для сравнения с критерием
        int sLo=abs(data[iy-1][ix].iLo+data[iy+1][ix].iLo+
            data[iy][ix-1].iLo+data[iy][ix+1].iLo-(data[iy][ix].iLo<<2));
        int sAv=abs(data[iy-1][ix].iAv+data[iy+1][ix].iAv+
            data[iy][ix-1].iAv+data[iy][ix+1].iAv-(data[iy][ix].iAv<<2));
        int sHi=abs(data[iy-1][ix].iHi+data[iy+1][ix].iHi+
            data[iy][ix-1].iHi+data[iy][ix+1].iHi-(data[iy][ix].iHi<<2));

        // сравнение разностных элементов с критериями
        // исключение из паттерна
        if (sLo<=delLo && sAv<=delAv && sHi<=delHi) {
            // установка признака исключения элемента из паттерна
            // (не являются краевыми условиями)
            pattern[iy][ix].byteLo=0;
        }
    }
}
```

```

    pattern[iy][ix].byteAv=0;
    pattern[iy][ix].byteHi=0;
} else {
    // корректировка нулевых элементов паттерна
    if (!pattern[iy][ix].byteLo) {
        pattern[iy][ix].byteLo=1;
    }
    if (!pattern[iy][ix].byteAv) {
        pattern[iy][ix].byteAv=1;
    }
    if (!pattern[iy][ix].byteHi) {
        pattern[iy][ix].byteHi=1;
    }
}
}
}
...
//=====

```

В том случае, если полученный в результате рассмотренного выше преобразования паттерн содержит значительное число повторяющихся элементов и граничных условий достаточно для восстановления исходного сигнала с заданной точностью, можно говорить о эффективном (с целью сжатия) исключении избыточной информации.

Для сжатия паттерна предлагаются следующие преобразования. Первое преобразование ориентированно на сжатие значительного числа повторяющихся элементов (кодирование длин серий), в первую очередь элементов исключённых из паттерна и отмеченных в данной реализации нулевым значением. Как уже было сказано, наилучшее качество восстановления изображения получается при одинаковой структуре (по положению исключённых элементов) для всех цветовых компонент, а число последовательных серий, исключённых элементов кратно трём (по числу элементов цветового пространства), что можно учесть при сжатии серий. Полученные на выходе, после кодирования длин серий, данные могут быть сжаты статистическим методом на основе преобразования Хаффмана или арифметического кодирования.

Восстановление изображения производится в обратной последовательности, начиная с декомпрессии паттерна. После получения массива паттерна его используют в качестве граничных условий для решения

МКР анализирующего дифференциального уравнения (13) (или другого выбранного при сжатии), в соответствии с разностной схемой (15) и шаблоном (рис. 4). Итерационную схему решения можно выразить уравнением:

$$f^t[x, y] = \frac{1}{4} \left(f^{t-1}[x + \Delta x, y] + f^{t-1}[x - \Delta x, y] + f^{t-1}[x, y + \Delta y] + f^{t-1}[x, y - \Delta y] \right), \quad (17)$$

где t – шаг итерации. Если сопоставлять начало итерации с началом циклов обхода массива восстанавливаемого изображения, а завершение итерации с завершением циклов обхода, то выбрав в циклах положительное приращение значений счётчиков, и используя для последующих расчётов ранее вычисленные значения ($f[x - \Delta x, y]$, $f[x, y - \Delta y]$), уравнение (17) можно записать в виде:

$$f^t[x, y] = \frac{1}{4} \left(f^{t-1}[x + \Delta x, y] + f^t[x - \Delta x, y] + f^{t-1}[x, y + \Delta y] + f^t[x, y - \Delta y] \right). \quad (18)$$

При распараллеливании вычислительного процесса МКР по элементам сетки будет работать итерационная схема в соответствии с выражением (17), при реализации каждой итерации циклическими переборами (по элементам сетки) будет работать итерационная схема (18).

Завершение итерационного процесса можно производить по критерию невязки, вычисляемой как сумма разностей всех значений элементов (без граничных условий) для соседних итераций t и $t-1$:

$$E = \sum_{x=1}^{Xsize-2} \sum_{y=1}^{Ysize-2} (f^t[x, y] - f^{t-1}[x, y]), \quad (19)$$

где: $Xsize = 0 \dots (Xsize - 1)$ и $Ysize = 0 \dots (Ysize - 1)$ – соответственно размеры изображения.

5. Повышение эффективности сжатия с учётом специфики цифрового сигнала

Как уже было отмечено, многообразие цифровых сигналов очень велико, как велико и разнообразие методов сжатия. При сжатии с потерями различны требования, предъявляемые к допустимой потере информации. Значительным

фактором успешного сжатия цифровых сигналов является выбор наиболее подходящего способа. Сжатие на базе анализа дифференциальной структуры может быть наиболее эффективно применено к сигналам (в частности к изображениям), содержащим значительные поля градиентных переходов амплитуды и достаточно контрастные (с точки зрения изменения амплитуды) границы таких полей. К сигналам подобного вида можно отнести изображения с чёткими границами объектов и значительными областями градиентного или однотонного наполнения. Примером изображений подобного вида могут быть:

- высококонтрастные фотографии технических или иных объектов, содержащие контрастные контуры и однотонные или градиентные поля;
- некоторые виды высококонтрастных космических фотографий, в том числе фотографий поверхности планет;
- картографические изображения;
- рисованные изображения с контрастными границами объектов;
- текстурные изображения;
- некоторые виды изображений, синтезируемые с помощью машинной графики и др.

Исследования сжатия изображений с помощью анализа дифференциальной структуры позволили определить дополнительные преобразования, применение которых способствует эффективности сжатия.

При выборе в качестве анализирующего уравнения – уравнение Пуассона (5), (6), которое может способствовать формированию более компактного паттерна по сравнению с уравнением Лапласа, необходимо определить его правую часть. В качестве правой части можно использовать матрицу определённых значений, сформированную по исходному изображению, разделённому на области, например, 8×8 пикселей. Значения матрицы могут быть получены выбором минимальной амплитуды из поля (8×8), такой подход в большинстве случаев, позволит корректировать плотность распределения

значений амплитуд в паттерне таким образом, что к нему можно более эффективно применять статистическое сжатие.

В случае сжатия фотоизображений, возможно, применять специальную обработку для удаления из исходного изображения шума возникающего, например, в результате дефектов ПЗС-матрицы (сокращение от прибор с зарядовой связью).

Формирование копии изображения в результате операции свертка с функцией рассеяния точки (ФРТ) и последующее использование этой копии для анализа дифференциальной структуры, при этом в паттерн сохраняются в качестве граничных условий элементы исходного изображения. Использование свёртки при сжатии в некоторых случаях позволяет повысить визуальное качество восстановленного сигнала. Низкочастотная фильтрация изображения позволяет исключить из паттерна случайный шум, например, свойственный цифровым фотографиям. Отрицательной стороной применения низкочастотных фильтров, является возможная потеря полезной информации из области высоких частот, например, мелких деталей изображения. В некоторых случаях может быть полезным применение высокочастотных фильтров подчёркивающих границы объектов на изображении. Практические исследования показали, что для реальных целей повышения эффективности сжатия с помощью свёртки необходимо использовать функции рассеяния точки размером $[3 \times 3]$ или $[5 \times 5]$.

Использование, для анализа дифференциальной структуры, производных высших порядков и различных шаблонов дифференцирования так же позволяет в некоторых случаях повысить эффективность сжатия. Выбор вида дифференциального уравнения в значительной мере зависит от особенностей изображения. Эксперименты показали, что изображения, полученные в результате фотографирования, отображения реальных и многих синтезированных объектов хорошо анализируются с помощью производных второго порядка, для них хорошо работают уравнения Лапласа и Пуассона.

На изображении (рис. 5) приведён пример потери полезной информации о форме сигнала в области, где мало значение второй производной. Наиболее уязвимы с точки зрения потери информации области, где происходит плавное изменение функции сигнала, особенно критичным это может быть для областей, где меняется знак первой производной. Противодействовать потере такой информации может включение в паттерн локальных минимумов и максимумов сигнала.

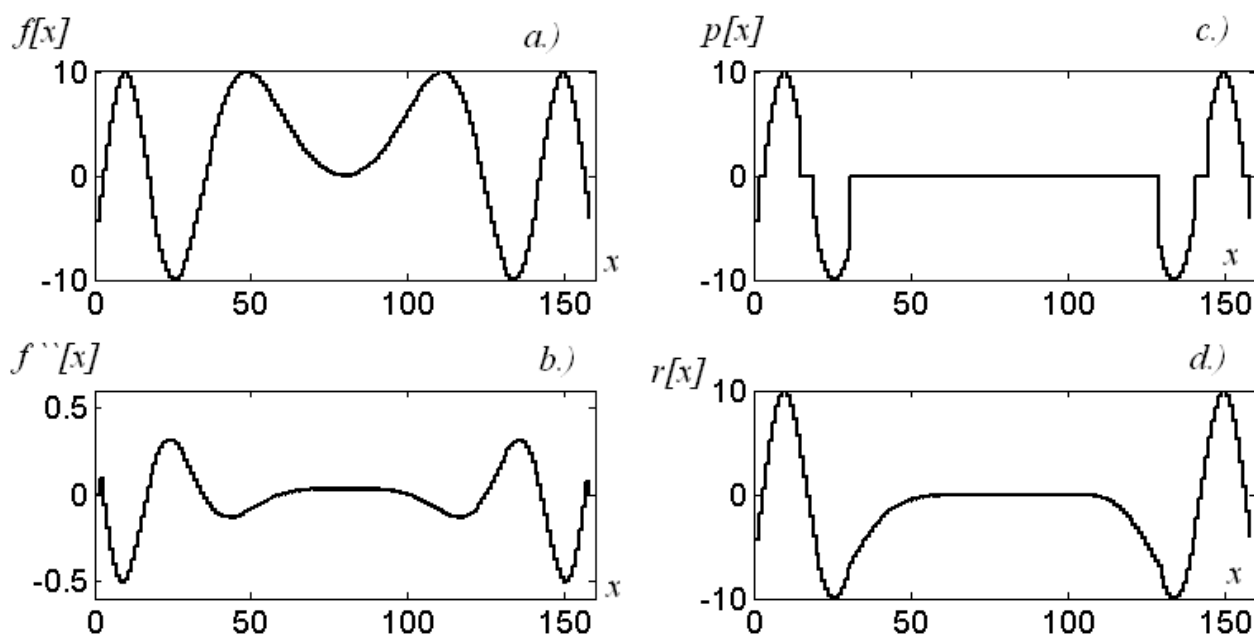


Рис. 5. Потеря информации о сигнале при малых значениях второй производной. На изображении обозначены: а.) – исходный сигнал $f[x]$; б.) – вторая производная $f''[x]$; в.) – паттерн $p[x]$; г.) – восстановленный сигнал $r[x]$.

Ниже приведен код фрагмента программы для формирования паттерна (листинг 2). По сравнению с программой (листинг 1) в код (листинг 2) добавлена свёртка, в качестве примера задана ФРТ называемая «скользящего среднего» подавляющая высокие частоты сигнала, в том числе случайные шумы и усиливающая сигнал до 9 раз. Размер квадратной матрицы ФРТ в данном примере хранится в константе `psfxu`. При практической реализации для большей функциональности ФРТ с размером `psfxu` можно динамически изменять (например, передавать в качестве параметра), что позволит задавать

для различных типов сигналов различные ФРТ и формировать наилучший паттерн с точки зрения минимизации размера сжатого паттерна при максимуме качества восстановленного сигнала. Для цифровых фотографий содержащих контрастные объекты и однотонные поля, с присутствием случайных шумов, достаточно хорошие результаты формирования паттерна и последующего сжатия показала приведённая в программе (листинг 2) ФРТ «скользящего среднего». Обратим внимание, что в паттерн копируются не результаты свёртки изображения, а элементы исходного изображения, что позволяет оставить больше исходной информации и получить более качественный результат при восстановлении. Свёртка с заданной ФРТ производит размытие полей и границ, что в свою очередь позволяет учесть в паттерне не только элементы самих границ, но и более удалённые от границ элементы. На цифровых фотографиях и подобных им цифровых сигналах, значения элементов изображения, образующие границы объектов, могут иметь значительные отличия от значений элементов внутри или снаружи границы, поэтому в качестве граничных элементов необходимо выбирать элементы вблизи границ, близкие по значениям к элементам ограниченной области. В примере (листинг 2) в результате свёртки значения элементов результата свёртки могут существенно превышать (до девяти раз) значения элементов исходного сигнала. Нормирование динамического диапазона или использование не целочисленных коэффициентов ФРТ будет увеличивать время вычисления, более эффективным, с точки зрения скорости вычислений, является изменение значений Δ -критериев (Δ_{Lo} , Δ_{Av} , Δ_{Hi}) до соответствующей величины с учётом коэффициентов матрицы ФРТ.

Листинг 2. Фрагмент программы формирования паттерна с предварительной свёрткой полноцветного изображения (C++).

```
//=====
...
// цветовая палитра
// покомпонентное представление, в скобках указаны компоненты
// палитр (RGB) и (YCbCr) соответственно
struct ColorSpace {
    uchar byteLo; // компонента младший байт (Lower) (R) (Y)
```

```

    uchar byteAv; // компонента средний байт (Average) (G) (Cb)
    uchar byteHi; // компонента старший байт (Hight) (B) (Cr)
};
// структура для сохранения цветов в виде int
struct IImage {
    int iLo; // компонента младший "байт" (Lower) (R) (Y)
    int iAv; // компонента средний "байт" (Average) (G) (Cb)
    int iHi; // компонента старший "байт" (Hight) (B) (Cr)
};
...
uint jmax=Xsize-1; // размер изображения по X минус единица
uint imax=Ysize-1; // размер изображения по Y минус единица
uint ix, iy, tix, tiy, nzero; // счётчики
const int psfxy=3; // размеры функции рассеяния точки ФРТ

// матрица ФРТ (PSF - point spread function)
int pulse[psfxy][psfxy]={1, 1, 1},
                        {1, 1, 1},
                        {1, 1, 1}};

// выделяем память для образца копии изображения
IImage** itemp2d=new IImage*[imax+psfxy];
for (unsigned int i=0; i<imax+psfxy; i++){
    itemp2d[i]=new IImage[jmax+psfxy];
    memset(itemp2d[i], 0, sizeof(IImage)* (jmax+psfxy));
}

// производим 2D быструю свёртку в целочисленной арифметике
for (int iyd=0; iyd<(int)Ysize; iyd++){
    for (int ixd=0; ixd<(int)Xsize; ixd++){
        // проход по импульсу
        for (int iyp=0; iyp<psfxy; iyp++){
            iy=iyd+iyp;
            for (int ixp=0; ixp<psfxy; ixp++){
                ix=ixd+ixp;
                itemp2d[iy][ix].iLo+=pulse[iyp][ixp]*(int)data[iyd][ixd].byteLo;
                itemp2d[iy][ix].iAv+=pulse[iyp][ixp]*(int)data[iyd][ixd].byteAv;
                itemp2d[iy][ix].iHi+=pulse[iyp][ixp]*(int)data[iyd][ixd].byteHi;
            }
        }
    }
}

// формируем паттерн
// копируем угловые точки с заменой нулей на единицы
swap(pattern[0][0], data[0][0]);
swap(pattern[0][jmax], data[0][jmax]);
swap(pattern[imax][0], data[imax][0]);
swap(pattern[imax][jmax], data[imax][jmax]);

for (iy=1; iy<imax; iy++){
    // координата по y для массива результат свёртки itemp2d
    tiy=iy+(psfxy>>1);
    // копируем крайние столбцы
    // с заменой нулей на единицы
    swap(pattern[iy][0], data[iy][0]);
    swap(pattern[iy][jmax], data[iy][jmax]);

    for (ix=1; ix<jmax; ix++){
        // координата по x для массива результат свёртки itemp2d
        tix=ix+(psfxy>>1);
        if (iy==1){

```



```

// на первом проходе копируем крайние строки
// с заменой нулей на единицы
swap(pattern[0][ix], data[0][ix]);
swap(pattern[imax][ix], data[imax][ix]);
}

int tLo=abs(itemp2d[tiy-1][tix].iLo+itemp2d[tiy+1][tix].iLo+
            itemp2d[tiy][tix-1].iLo+itemp2d[tiy][tix+1].iLo-
            (itemp2d[tiy][tix].iLo<<2));
int tAv=abs(itemp2d[tiy-1][tix].iAv+itemp2d[tiy+1][tix].iAv+
            itemp2d[tiy][tix-1].iAv+itemp2d[tiy][tix+1].iAv-
            (itemp2d[tiy][tix].iAv<<2));
int tHi=abs(itemp2d[tiy-1][tix].iHi+itemp2d[tiy+1][tix].iHi+
            itemp2d[tiy][tix-1].iHi+itemp2d[tiy][tix+1].iHi-
            (itemp2d[tiy][tix].iHi<<2));

// обработка внутренних элементов
// сравнение с  $\Delta$ -критериями по каждому цветовому каналу
if ((tLo<=delLo && tAv<=delAv && tHi<=delHi)){
    // установка признака исключения элемента из паттерна
    // (не являются краевыми условиями)
    pattern[iy][ix].byteLo=0;
    pattern[iy][ix].byteAv=0;
    pattern[iy][ix].byteHi=0;
    nzero+=3; // счётчик исключённых из паттерна элементов
} else {
    // элементы паттерна с нулевыми значениями, корректировка
    if (!pattern[iy][ix].byteLo){
        pattern[iy][ix].byteLo=1;
    }
    if (!pattern[iy][ix].byteAv){
        pattern[iy][ix].byteAv=1;
    }
    if (!pattern[iy][ix].byteHi){
        pattern[iy][ix].byteHi=1;
    }
}
}
}

// чистим память
if (itemp2d && sizeY()){
    for (ix=0; ix<imax+psfxy; ix++){
        delete[] itemp2d[ix];
        delete[] itemp2d;
        itemp2d=0;
    }
    ...
}
// Функция swap
// копирование точки (в трёхцветной палитре с заменой нулей на единицы)
inline void swap(ColorSpace& receiver, ColorSpace& source){
    // делаем проверку и при необходимости заменяем нули единицей
    receiver=source; // копируем
    if (!receiver.byteLo){
        receiver.byteLo=1;
    }
    if (!receiver.byteAv){
        receiver.byteAv=1;
    }
    if (!receiver.byteHi){
        receiver.byteHi=1;
    }
}

```

}

...

//=====

6. Вычислительная оптимизация МКР на базе отыскания промежуточного решения

При восстановлении сжатых данных немаловажным фактором является скорость. МКР достаточно медленный итерационный способ решения дифференциальных уравнений, следовательно, значительное время может быть затрачено для решения многомерных задач. Поэтому существенным вопросом является оптимизация вычислений.

Известен способ сокращения времени вычислений МКР для мягких дифференциальных уравнений [12] в некоторой области U с краевыми (граничными для физических областей и начальными для времени) условиями в областях V , сущность которого состоит в укрупнении шага h сетки U в области $U \cup V = W$. При этом происходит потеря точности решения вплоть до того, что оно становится непригодным. Например, в основной задаче электростатики могут быть рассмотрены электрические заряды, описанные единственной точкой. При укрупнении сетки такие граничные условия могут быть потеряны, что принципиально меняет сущность промежуточного решения. При формировании паттерна могут возникать краевые условия, например, в виде уединённой точки. Таким образом, для наших задач потеря точности в результате укрупнения сетки так же актуальна.

По этой причине обычным является решение, содержащее несколько последовательно применяемых сеток. Первоначально применяется самая крупная сетка U_1 , позволяющая получить приближение решения. Затем решение уточняют, последовательно применяя сетки с меньшим шагом $U_k, k > 1$. Общее число различных сеток обычно составляет 2–3. Шаг сетки h может быть различным в разных направлениях и областях сетки U . Таким образом, ускорение процедуры сходимости к решению задачи с заданной

точностью p происходит за счет более быстрого формирования некоторого промежуточного решения в области U , что является прямым следствием теоремы о сходимости разностного решения [13]. При этом каждое уточнение решения является итерационным, имеющим вычислительную сложность

$$O\left(k \prod_{i=1}^n N_i\right), \quad (20)$$

где: n – размерность рассматриваемой задачи; N_i – число узлов сетки в каждом направлении; k – число итераций, обеспечивающее заданную точность на данном этапе. В многомерных задачах величина (20) может быть очень велика даже при разреженной сетке.

Указанную задачу нахождения промежуточного (приближённого) решения в области U можно решить путем аппроксимации значений $f(R^n)$ уравнения (8). Аппроксимацию $f(R^n)$ можно производить различными функциями, в зависимости от физической сущности задачи, например, многочленом вида:

$$a_0 + a_1 z + a_2 z^2 + \dots + a_p z^p, \quad (21)$$

где $a_0, a_1, a_2, \dots, a_p$ – коэффициенты многочлена; z – переменная, вдоль которой происходит аппроксимация.

В общем случае, если линии сетки не параллельны координатным осям, образующим пространство W , то z может не совпадать с набором переменных исходной задачи. В этом случае необходимо учитывать поворот системы координат, в которых рассмотрен аргумент z , относительно исходной системы координат. В случае n -мерной ($n > 1$) задачи, аппроксимирование необходимо производить последовательно для всех линий, образующих сетку в каждом направлении, с последующей оценкой средних значений для каждого узла сетки. Среднее значение узла сетки вычисляется как среднее значение аппроксимации в точках линий сетки, принадлежащих данному узлу. В определенном смысле можно говорить, что для нахождения некоторого

промежуточного решения задача МКР разбивается на множество одномерных задач МКЭ, число которых равно числу линий сетки МКР.

Возможно совместное применение способа укрупнения сетки и предлагаемого способа, ведь даже для укрупненной сетки процесс ускорения сходимости остается актуальным. В общем случае применение предложенного способа ускорения сходимости должно производиться с учетом физической сущности процессов и условий каждой конкретной задачи. При укрупнении сетки используются не все граничные условия, а только их часть, поэтому можно также использовать некоторые усредненные граничные условия. В этом случае, фактически для получения новых граничных условий, будет применен предложенный способ, но в локальной области.

Промежуточное решение, полученное предлагаемым способом [14,15], в некоторых случаях (в зависимости от особенностей граничных условий), может быть «не очень» гладким. На практике возможно применение некоторых дополнительных действий, повышающих гладкость. Например, при аппроксимации возможен учет (с некоторыми весовыми коэффициентами) значений в соседних узлах, не вошедших в качестве значения или аргумента в функцию аппроксимации, применяемую к данной линии сетки.

Предлагаемый способ нахождения предварительного решения базируется не на последовательных итерационных приближениях, а на аппроксимации промежуточного решения в области U по имеющимся данным о граничных условиях B , с учетом правой части (8). При этом вычислительная сложность может быть оценена как:

$$O\left(n \prod_{i=1}^n N_i\right), \quad (22)$$

где, как правило, $n < k$, причем n и k имеют тот же смысл, что и в (20), что позволяет говорить о снижении вычислительной сложности предлагаемого способа по отношению к последовательному измельчению сетки.

В общем случае указанный подход повышения скорости МКР применим не только для восстановления исходного сигнала по паттерну, но и для других задач, решение которых возможно с помощью МКР. Дополнительно способ тестировался на различных задачах электротехники, описываемых уравнением Лапласа и Пуассона, в пространствах с числом измерений пространства $n = 2$ и $n = 3$.

Во всех случаях полученное промежуточное решение позволяло существенно сократить число итераций МКР для получения решения с заданной точностью. Особый выигрыш в сокращении числа итераций наблюдается при гладких краевых условиях, вплоть до того, что при некоторой достаточной гладкости может быть получено решение только с применением указанного способа без итерационного процесса МКР. Такие краевые условия вполне актуальны для некоторых типов изображений.

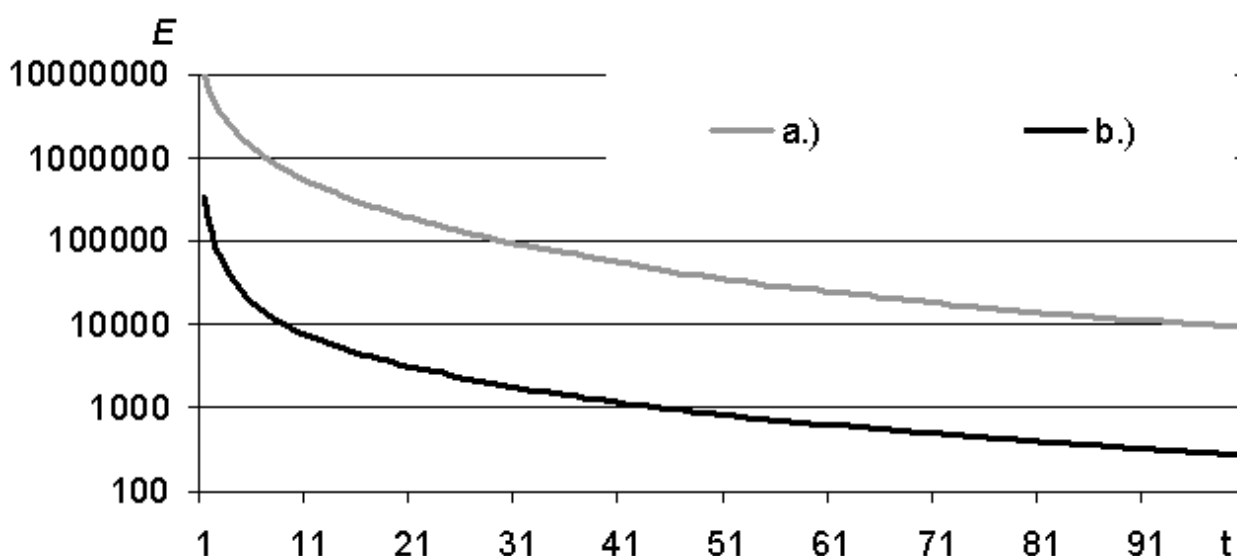


Рис. 6. Значения суммарных невязок E в логарифмическом масштабе, по всем узлам сетки в зависимости от номера итерации t в двумерном пространстве XOY . а.) – без поиска предварительного решения; б.) – с поиском предварительного решения.

Существенное ускорение сходимости, в результате оптимизации рассмотренным способом, на примере невязок в решении двумерной (плоско-параллельной) задачи электростатики, приведено на изображении (рис. 6). Для разбиения области W выбиралась прямоугольная сетка, образованная линиями,

параллельными координатным осям, с равным во всех направлениях шагом h . В качестве многочлена (21) было использовано уравнение прямой $a_0 + a_1 z$, коэффициенты которой (a_0, a_1) рассчитывались по двум элементам краевых условий из **В**.

Отметим, что применение рассмотренного способа в некоторых двумерных задачах, для достижения решения с заданной по невязке точностью, позволило уменьшить число итераций в 10–100 раз, что говорит о существенном повышении эффективности МКР.

Ещё одним дополнительным способом повышения скорости вычислений МКР является распараллеливание итерационного процесса, например, по элементам сетки. Достаточно просто такое решение реализуется на базе современных графических процессоров, например, при помощи технологии CUDA, что позволяет многократно (в сотни раз) повысить скорость вычислений [16].

7. Сравнение и некоторые результаты сжатия цифровых сигналов с помощью анализа дифференциальной структуры

Рассмотрим некоторые результаты практического применения изложенного выше принципа сжатия на примере компрессии изображений. В качестве анализирующего уравнения было выбрано уравнение Лапласа (13). Изображения для тестирования выбирались из базы данных стандартных тестовых изображений [17], с учётом фактора, что сжатие на основании анализа дифференциальной структуры, наиболее эффективно может быть применено к графическим объектам, содержащим контрастные границы и значительные поля градиента или одного тона.

Одно из выбранных для тестирования изображений (рис. 7) обладает следующими свойствами: является цифровым сигналом, полученным в результате фотографирования; содержит контрастные границы объектов;

содержит значительные поля одного тона. Рассмотрим зависимости от значения Δ -критерия следующих величин:

- число исключённых элементов из паттерна;
- коэффициент сжатия:

$$k = \frac{D_{out}}{D_{in}}, \quad (23)$$

где: D_{out} – объём сжатых (выходных) данных (бит); D_{in} – объём исходных (входных) данных (бит);

- среднеквадратичную ошибку (MSE – mean square error):

$$MSE = \frac{1}{n} \sum_{i=1}^n (f[i] - r[i])^2, \quad (24)$$

где: $f[i]$ – значения исходного и $r[i]$ – восстановленного сигналов;

- пиковое отношение сигнал/шум (PSNR – peak signal to noise ratio):

$$PSNR = 20 \log_{10} \frac{\max_i (|f[i]|)}{\sqrt{MSE}}; \quad (25)$$

- отношение сигнал/шум (SNR – signal to noise ratio):

$$SNR = 20 \log_{10} \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n f[i]^2}}{RMSE}, \quad (26)$$

где: RMSE – корень среднеквадратической ошибки т.е.: $RMSE = \sqrt{MSE}$.

Как отмечалось выше, от величины Δ -критерия зависит число исключённых из паттерна элементов, в конечном результате, сжатие. При этом, можно ожидать следующую закономерность: чем больше величина Δ -критерия, тем больше исключено элементов из паттерна и тем менее точно можно восстановить исходный сигнал. Когда величина Δ -критерия будет иметь значение больше или равное некоторому пороговому, паттерн, будет содержать только элементы принадлежащие границе изображения. Следовательно при пороговой величине Δ -критерия можно получить максимальное сжатие. При этом вопрос о качестве восстановленного изображения, полученного из

паттерна сформированного по Δ -критерию большего или равного пороговому, остаётся открытым.



Рис. 7. Стандартное тестовое изображение «4.1.05.bmp». Размер исходного файла изображения 196662 байт, размеры 256×256 пикселей, глубина цвета 24 бита, размер цифрового сигнала изображения $256 \times 256 \times 3 = 196608$ байт.

В таблице (таблица 2) приведены численные результаты экспериментов. Для всех цветовых каналов величина Δ -критерия была выбрана одинаковая, формирование паттерна производилось в цветовом пространстве RGB. После формирования паттерна производилось сжатие длин серий, а затем кодирование Хаффманна. Для результирующих данных был разработан специальный формат позволяющий сохранить в файл не только сами данные но и некоторую служебную информацию: дату, размеры изображения, используемое цветовое пространство, некоторые другие данные записанные пользователем. Служебная информация несколько увеличивает размер готового файла по сравнению размером сжатого сигнала. Длина служебных данных в файлах, создаваемых сжатием с помощью анализа дифференциальной структуры, составила 52 байта, а в исходном файле «4.1.05.bmp» 54 байта. При общем размере исходного файла «4.1.05.bmp» 196662 байт.

Для восстановления изображения применялся МКР (1000 итераций) с предварительным отысканием промежуточного решения. Время формирования паттерна в одном вычислительном потоке (ОС Windows, C++) на процессоре

Core2Duo E8500 (с эффективной тактовой частотой процессора 3,16 ГГц, частота процессорной шины 1.333 ГГц, кэш-память второго уровня размером 6МБ), составило 31250 мкс, время восстановления 1343750 мкс. Расспаралеливание вычислительного потока не применялось. Расчёт коэффициента сжатия и фактора сжатия производился путём соотношения размеров конечных файлов исходного изображения «4.1.05.bmp» (196662) и файла полученного в результате сжатия с помощью анализа дифференциальной структуры.

Таблица 2. Некоторые оценки сжатия изображения «4.1.05.bmp».

Δ- критерий	Число исключённых элементов		Коэффициент сжатия k	Фактор сжатия c	MSE	$PSNR$	SNR
	байт	%					
0	60	0,030518	0,889	1,125	0,001302	76,98	72,37041
5	23535	11,97052	0,840	1,191	0,121638	57,28	52,66611
10	57828	29,41284	0,699	1,431	0,800832	49,10	44,48139
15	84129	42,79022	0,582	1,719	2,204839	44,70	40,08303
20	107364	54,60815	0,481	2,080	5,200922	40,97	36,356
25	127779	64,99176	0,388	2,578	11,78631	37,42	32,80302
30	144492	73,49243	0,304	3,290	22,84199	34,54	29,92946
35	157290	80,00183	0,234	4,275	36,16757	32,55	27,93361
40	166086	84,47571	0,184	5,436	56,36549	30,62	26,00667
45	171954	87,46033	0,148	6,759	85,77358	28,80	24,18327
50	176058	89,54773	0,123	8,132	127,981	27,06	22,44535
55	179193	91,14227	0,104	9,618	192,4735	25,29	20,67309
60	181608	92,37061	0,089	11,239	268,6562	23,84	19,22483
65	183483	93,32428	0,077	12,991	384,8277	22,28	17,66414
70	184998	94,09485	0,068	14,710	503,2751	21,11	16,49875

На (рис. 8) приведены некоторые изображения, позволяющие получить визуальное представление о качестве восстановленного изображения и соответствующем паттерне.

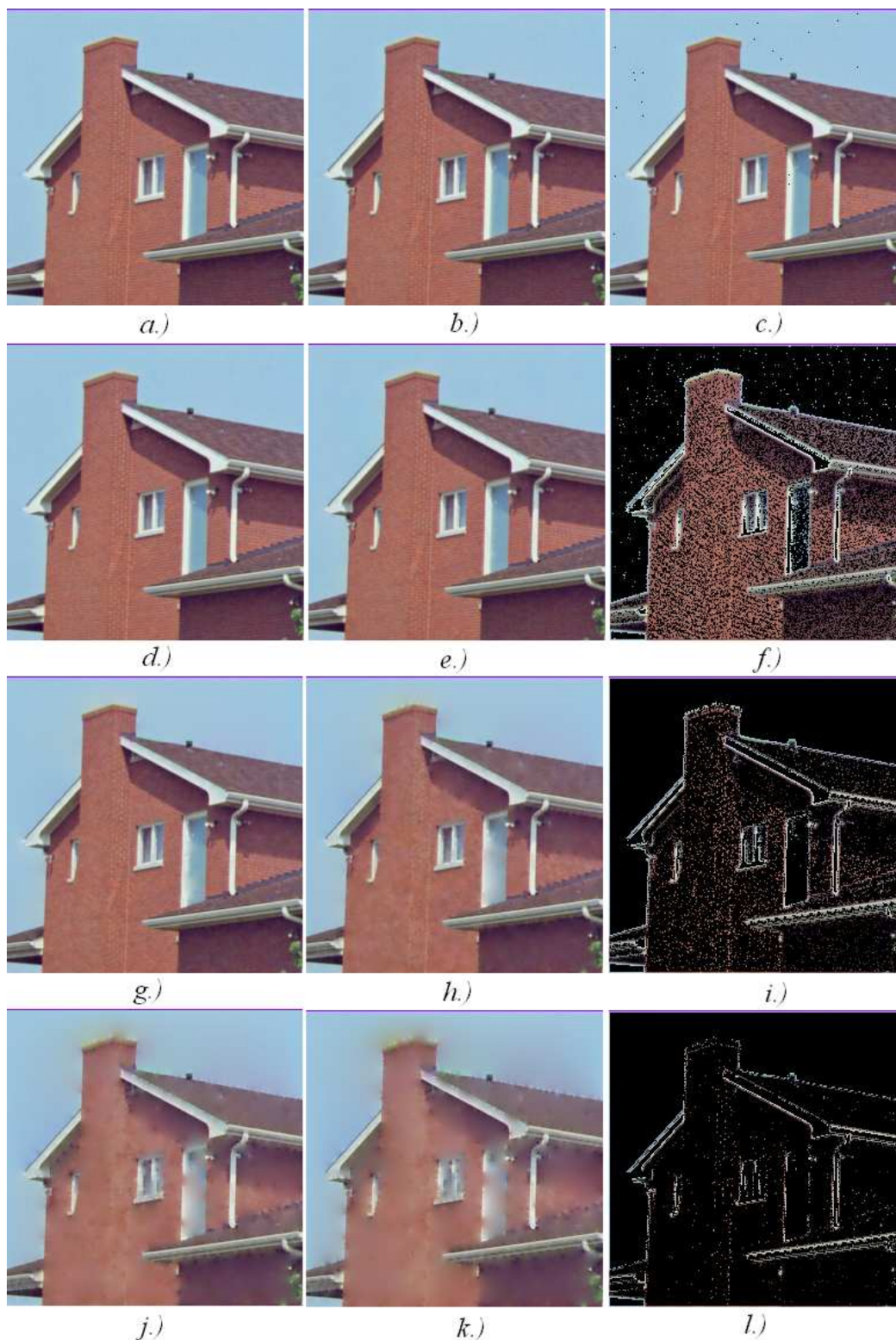


Рис. 8. Исходное изображение (a.), далее восстановленные (b., d., e., g., h., j., k.) 1000 итераций МКР с использованием промежуточного решения. Колонка справа (c., f., i., l.) соответствует паттернам. Значения Δ -критерия следующие: b., c. – 0; d. – 10; e., f. – 20; g. – 30; h., i. – 40; j. – 50; k., l. – 60.

Обычное значение коэффициента $PSNR$ для изображений приемлемого качества составляет величину порядка 20...40 единиц [8]. Сопоставляя

результаты (таблица 2) и (рис. 8) можно заметить, что в области значений Δ -критерия порядка 40...50 наблюдается существенное, для визуального восприятия, ухудшение качества изображения, возникают области размытия вблизи границ объектов, при этом значения $PSNR = 30,62$ (для $\Delta = 40$) и $PSNR = 27,06$ (для $\Delta = 50$), что соответствует ожидаемой области ограничения приемлемого качества. Коэффициент сжатия $k = 0,184$ (для $\Delta = 40$) и $k = 0,123$ (для $\Delta = 50$). При дальнейшем росте величины Δ -критерия наблюдается дальнейшее ухудшение качества, особенно проявляемое на границах графических объектов, изучение структуры паттерна показывает, что снижается и качество отображения текстуро-подобной однотонной области кирпичных стен, при сравнении паттерна (рис. 8, *h.*) с (рис. 8, *j.*) заметно, что пропадают граничные элементы, разделяющие отдельные малоконтрастные элементы изображения. При дальнейшем увеличении величины Δ -критерия до 70...80 единиц содержимое изображения, несмотря на значительные искажения, оставалось узнаваемым. Далее рассмотрим сопоставление полученных результатов с некоторыми другими, наиболее распространёнными графическими форматами (таблицы 3, 4, 5).

Таблица 3. Некоторые оценки JPEG-сжатия изображения «4.1.05.bmp».

Качество	Коэффициент сжатия k	Фактор сжатия c	MSE	$PSNR$	SNR
12	0,494	2,023	1,56	46,17	41,56
11	0,359	2,778	5,63	40,62	36,01
10	0,282	3,538	9,35	38,41	33,80
9	0,245	4,069	12,97	37,00	32,39
8	0,222	4,497	17,13	35,79	31,18
7	0,201	4,970	23,31	34,45	29,84
6	0,199	5,006	40,10	32,10	27,49
5	0,187	5,333	46,79	31,43	26,82
4	0,179	5,556	51,48	31,01	26,40
3	0,174	5,746	58,72	30,44	25,83
2	0,164	6,064	75,88	29,33	24,72
1	0,160	6,243	96,81	28,27	23,66
0	0,158	6,316	105,61	27,89	23,28

Таблица 4. Некоторые оценки PNG-сжатия изображения «4.1.05.bmp».

Число разрядов	Число цветов	Коэффициент сжатия k	Фактор сжатия c	MSE	$PSNR$	SNR
24	16777216	0,609136	1,641668	0	inf	inf
8	256	0,191079	5,233434	10,80395	37,79498	33,18098
8	128	0,15489	6,45619	16,1552	36,04768	31,43368
8	64	0,111638	8,957504	24,60652	34,22031	29,6063
8	32	0,091278	10,95549	34,69823	32,72773	28,11373
8	16	0,040465	24,71249	50,41061	31,10559	26,49158
8	8	0,026284	38,04643	140,4574	26,65536	22,04136
8	4	0,01436	69,63952	252,7536	24,10383	19,48983
8	2	0,007144	139,973	569,2647	20,57766	15,96366

Таблица 5. Некоторые оценки GIF-сжатия изображения «4.1.05.bmp».

Число цветов	Коэффициент сжатия k	Фактор сжатия c	MSE	$PSNR$	SNR
128	0,171243	5,839653	16,91316	35,84856	31,23455
64	0,123583	8,091754	26,3118	33,9293	29,3153
32	0,100335	9,966653	37,71211	32,36599	27,75199

Среди изображений полученных различными способами сжатия выберем по одному, имеющему допустимое для визуального восприятия качество при минимальном значении коэффициента сжатия. Несмотря на то, что данный выбор носит субъективный характер, он вполне допустим среди прочих сравнений, т.к. позволяет учесть фактор специфики восприятия изображения человеком. Изображения (рис. 9) и численная оценка сжатия различными методами (таблица 6) позволяют произвести сопоставление результатов.

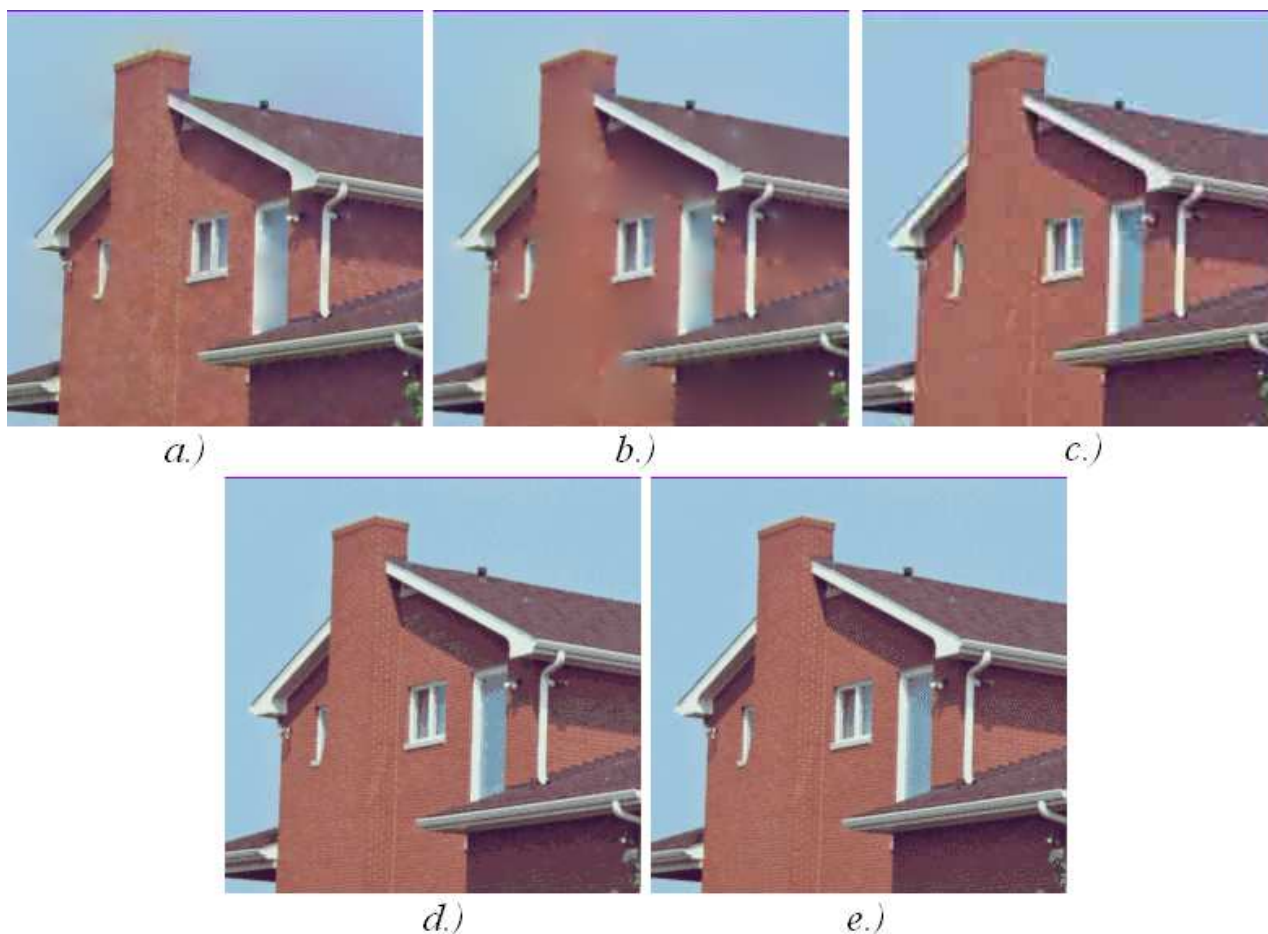


Рис. 9. Изображения, полученные различными способами сжатия (таблица 6):
 а.) – сжатие на основе анализа дифференциальной структуры ($\Delta = 39$ таблица 2);
 б.) – сжатие на основе анализа дифференциальной структуры с предварительной свёрткой;
 с.) – JPEG-сжатие (качество 0 таблица 3);
 д.) – PNG-сжатие (разрядность 8, число цветов 32 таблица 4);
 е.) – GIF-сжатие (32 цвета таблица 5).

Таблица 6. Сравнение некоторых результатов сжатия изображения «4.1.05.bmp», различными способами.

изображение (рис. 9)	Коэффициент сжатия k	Фактор сжатия c	MSE	$PSNR$	SNR
a.	0,193	5,183	52,1964	30,95	26,34
b.	0,156	6,391	76,36	29,30	24,69
c.	0,158	6,316	105,61	27,89	23,28
d.	0,091	10,955	34,69	32,72	28,11
e.	0,100	9,967	37,712	32,37	27,75

Визуальное (рис. 9) и численное (таблица 6) сравнение полученных результатов показывают, что сжатие на основе анализа дифференциальной структуры позволяет добиться сжатия сопоставимого с JPEG при сопоставимом качестве. При этом, качество восстановленного изображения, сжимаемого с помощью анализа дифференциальной структуры, имеет лучшие численные оценки (MSE, PSNR, SNR), что может быть особенно полезно для дальнейших машинных преобразований, например с целью распознавания образов.

Интересны результаты применения свёртки (рис. 10), (таблица 7) с ФРТ

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$
 которая является низкочастотным однородным фильтром

«скользящего среднего» с усилением. Применение свёртки позволило более эффективно выделить границы крупных объектов и практически исключить шумовую составляющую. Исключение высокочастотных составляющих и шума позволяет получить более протяжённые цепочки исключённых из паттерна элементов, что в свою очередь дополнительно способствует сжатию. Недостатком применения свёртки является возможная потеря высокочастотных элементов, например, небольших объектов, тонких линий и т.д.

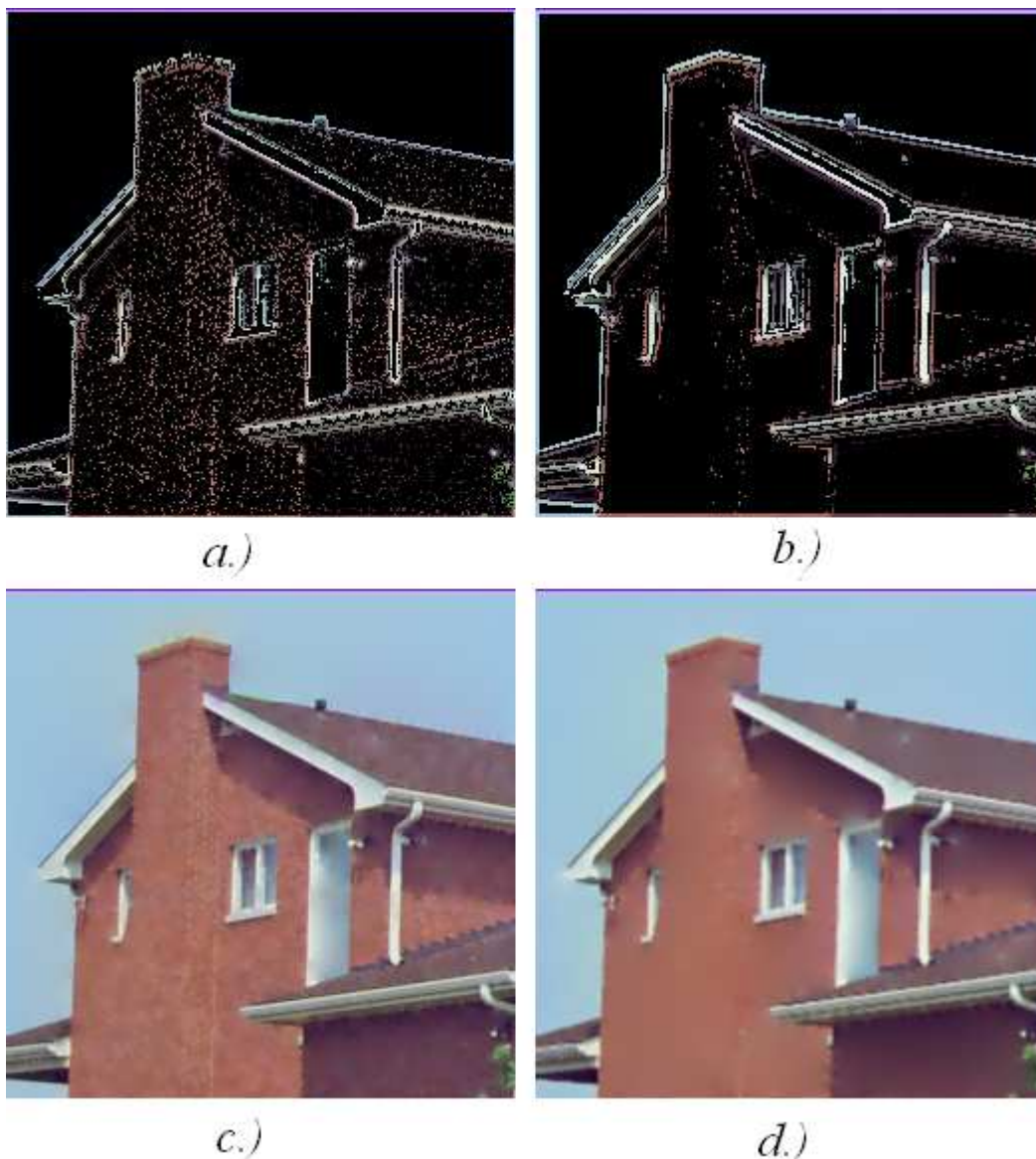


Рис.10. Паттерны и восстановленные изображения.

a. – паттерн получен без применения свёртки; *b.* – паттерн получен с применением свёртки; *c.* – изображение восстановленное по паттерну (*a.*); *d.* – изображение восстановленное по паттерну (*b.*).

Таблица 7. Результаты оценки влияния свёртки на сжатие с помощью анализа дифференциальной структуры на примере изображения «4.1.05.bmp».

Применение свёртки	Число исключённых элементов из паттерна	Коэффициент сжатия k	Фактор сжатия c	MSE	$PSNR$	SNR
нет	166086	0,184	5,434	56,114	30,640	26,026
да	166119	0,172	5,829	66,427	29,907	25,293

Результаты сравнения (таблица 7) и (рис. 10) показывают, что применение свёртки позволяет получить более компактный вид сжатого файла при лучшем визуальном качестве восстановленного изображения.

Дополнительные исследования на значительном числе тестовых изображений показали, что разработанный метод сжатия в большинстве случаев по численным оценкам (MSE , $PSNR$, SNR) показывает результаты превосходящие (при равных коэффициентах сжатия) полученные распространённым методом сжатия JPEG и JPEG2000. При этом качество визуального восприятия (при равной степени сжатия) получается сравнимо (с JPEG и JPEG2000) для контрастных изображений, с чётко выделенными границами объектов и значительными градиентными или однотонными пространствами. Для малоконтрастных изображений, с маловыраженными границами объектов визуальное качество получается ниже. Проявляется снижение качества в характерном размытии границ, и потери малоконтрастных фрагментов изображения.

Полученные результаты показывают конкурентоспособность способа сжатия с помощью анализа дифференциальной структуры. При этом необходимо отметить, что способ сжатия изображений на основе анализа дифференциальной структуры имеет дополнительный потенциал. При сжатии паттерна в данных исследованиях использовался код Хаффмена, известно, что, например, арифметическое кодирование (применяемое в JPEG2000) позволяет производить более эффективное сжатие. Дополнительно повысить эффективность сжатия возможно, используя, для формирования паттерна и восстановления сигнала, вместо уравнения Лапласа, уравнение Пуассона. Причём в правой части уравнения Пуассона можно учесть спектральные особенности сигнала, при помощи вейвлет или Фурье преобразования. Такой подход позволит наряду с дифференциальной учитывать и частотную структуру сигнала.

Литература

1. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников. / Стивен Смит; пер. с англ. А. Ю. Линовича, С В. Витязева, И. С. Гусинского. – М.: Додэка-XXI, 2011. – 720 с.:ил.
2. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. 2-е издание, испр. М.: Техносфера, 2009. – 856 с.: ил.
3. Солонина А.И., Улахович Д.А., Арбузов С.М., Соловьёва Е.Б. Основы цифровой обработки сигналов: курс лекций. 2-е издание испр. и перераб.– СПб.: БХВ-Петербург, 2005.–768 с.: ил.
4. Лайонс Р. Цифровая обработка сигналов: Второе издание. Пер. с англ.– М.:ООО «Бином-Пресс», 2009.–656 с.:ил.
5. Чобану М. Многомерные многоскоростные системы обработки сигналов. М.: Техносфера, 2009. – 480 с.
6. D. Saupe, R. Hamzaoui, H. Hartenstein. Fractal image compression - An introductory overview, in: Fractal Models for Image Synthesis, Compression, and Analysis, D. Saupe, J. Hart (eds.), ACM SIGGRAPH'96 Course Notes.
7. Скиена С. Алгоритмы. Руководство по разработке. – 2-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2011.
8. Сэлмон Д. Сжатие данных, изображений и звука. М.: Техносфера, 2006. – 386 с.
9. Пискунов Н.С. Дифференциальное и интегральное исчисления. В 2-х т. Т.II: – М.: Интеграл-Пресс, 2005. – 544 с.].
10. Ильин В.А., Поздняк Э.Г. Основы математического анализа: В 2 ч. Часть II.– 3-е изд.– М.: Наука. Физматлит, 1998.– 448 с.
11. Серов А.В. Эфирное цифровое телевидение DVB-T/H/–СПб.: БХВ-Петербург, 2010.–464 с.:ил.

12. Бахвалов Н.С., Воеводин В.В. Современные проблемы вычислительной математики и математическо-го моделирования: в 2 томах. – Т. 1. Вычислительная математика. – М.: Наука, 2005. – 343 с.
13. Формалёв В.Ф., Ревезников Д.Л. Численные методы. – М.: Физматлит, 2004. – 400 с.
14. Гришенцев А.Ю., Коробейников А.Г. Улучшение сходимости метода конечных разностей с помощью вычисления промежуточного решения. Научно-технический вестник информационных технологий, механики и оптики, 2012, № 3 (79).–С. 124-127.
15. Информационные технологии./ Обработка цифровых сигналов, 2012. // [Электронный ресурс]. URL: <http://www.moveinfo.ru/>
16. Технологии CUDA. NVIDIA Corporation, 2012. // [Электронный ресурс]. URL: http://www.nvidia.ru/object/cuda_home_new_ru.html.
17. University of Southern California./ Signal and Image Processing Institute, 2012. // [Электронный ресурс]. URL: <http://sipi.usc.edu/database/>