

## ПОВЫШЕНИЕ БЫСТРОДЕЙСТВИЯ АЛГОРИТМА ФРАКТАЛЬНОГО КОДИРОВАНИЯ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ

*Зыков А.Н., АО «Центр Судоремонта «Звёздочка», г. Северодвинск, e-mail: alexzikov@gmail.com.*

### INCREASING OF THE SPEED OF THE ALGORITHM OF FRACTAL CODING THE HALFTONE IMAGES

*Zykov A.N.*

*The offered to your attention work describes using Local Binary Pattern (LBP) in the task of fractal coding grayscale images. Based on local binary models, local binary patterns offer an effective way to analysis of the image and texture are its effective characteristic. This work is aimed at providing new ideas for reducing the time of fractal coding, in particular, using the classification of areas of an image on the basis of using operator the local binary patterns.*

**Key words:** image compression, fractal coding, image analysis, local binary patterns.

**Ключевые слова:** сжатие изображений, фрактальное кодирование, анализ изображений, локальные бинарные шаблоны.

#### Введение

В нашей жизни изображение представляет нам наиболее полную информацию об окружающем мире и поэтому является важным её носителем. В настоящее время, в связи с резким ростом количества цифровой техники, появилось огромное количество цифровой информации. Значительную долю объема цифровой информации занимают графические цифровые изображения. Поскольку цифровое изображение содержит большое количество данных, то оно должно быть эффективно сжато с тем, чтобы занимать меньше места при хранении, и его можно было передавать с более высокой скоростью по сети.

Основной целью кодирования изображений является представление цифровых изображений как можно минимальным количеством битов, сохраняя при этом уровень качества, необходимого для их использования. Методы сжатия изображений имеют важное значение в реальной жизни в сфере информационных технологий. Эффективные технологии кодирования помогающие убрать избыточность и сохранить при этом качество изображения находятся в центре внимания современных исследований. Фрактальное кодирование изображения [1] является одной из таких новых эффективных технологий кодирования.

Сегодня фрактальное кодирование изображений является одним из наиболее эффективных технологий сжатия графической информации с потерями и, как следствие, достаточно перспективным для использования при разработке компьютерных мультимедиа приложений и обработке цифрового видео. Технология фрактального кодирования представляет собой эффективный метод сжатия изображения, основанная на представлении изображения посредством сжимающих аффинных преобразований в пространстве изображений, для которых

*Описывается использование оператора локальных бинарных шаблонов (LBP – Local Binary Pattern) в задаче фрактального кодирования полутоновых изображений. Основываясь на локальной двоичной модели, локальные бинарные шаблоны предлагают эффективный способ анализа текстуры изображения и являются её эффективной характеристикой. Эта работа нацелена на предоставление новых идей для уменьшения времени фрактального кодирования, в частности, в использовании классификации областей изображения на основе использования оператора локальных бинарных шаблонов.*

неподвижная точка близка к исходному изображению. Этот метод привлек к себе большое внимание в последние годы из-за высокой степени сжатия, которая может быть достигнута при его применении и также скоростью восстановления исходного изображения.

Большим недостатком фрактального кодирования является время его кодирования. В 1989 А.Жаквин был предложен метод [2] автоматического кодирования на основе блоков изображения. Данный метод лёг в основу подавляющего большинства схем фрактального кодирования и имел важные последствия для фрактального кодирования изображений в будущем. Процесс кодирования состоит в использовании системы итерированных кусочно-определенных функций (PIFS – Partitioned Iterated Function Systems) в построении оператора, который будет представлять образ закодированного изображения. Если изображение представлено оператором, то его можно эффективно хранить и передавать. Большая часть времени, в построении такого оператора, уходит на поиск лучшего совпадения между частями подлежащего кодированию изображения. Для сокращения вычислительных затрат используют несколько подходов увеличения скорости кодирования. Одно из направлений состоит в классификации областей изображения, которая может значительно ускорить кодирование за счет уменьшения количества используемых областей, среди которых ведется поиск. Классификация происходит путем выделения определенных характеристических особенностей, описывающих области изображения. Выде-

ление небольшого количества особенностей, характеризующих области изображения, позволяет ускорить процесс фрактального кодирования. Тогда сравнение между областями изображения будет производиться на основании этих характеристик, а не по отдельным пикселям, что существенно сокращает объем работы. Эти характеристики могут быть получены на основании спектрального анализа Фурье, вейвлет-анализа, различных характеристик текстуры изображения.

Текстурный анализ является одним из основных в компьютерном зрении и распознавании образов. Текстура со структурной точки зрения состоит из элементов, которые организованы в соответствии с определенными правилами размещения. В данной работе описывается использование локальных бинарных шаблонов для классификации областей изображения в задаче фрактального кодирования изображений. Изображение кодируется методом поиска наилучшего соответствия между областями только с одними и теми же значениями, полученными с помощью оператора локальных бинарных шаблонов. Экспериментальные результаты показывают, что данная схема является компетентной как в скорости кодирования, так и в качестве восстановления.

**Схема фрактального кодирования изображений**

Рассмотрим классическую схему фрактального блочного кодирования изображений методом А. Jacquin. Возьмем полутоновое изображение  $I$  и представим его в виде двумерного прямоугольного массива пикселей:

$$I = i(x, y), \quad x \in [0, w), \quad y \in [0, h),$$

где  $w$  и  $h$  размеры изображения.

Фрактальное кодирование изображения подразумевает нахождение сжимающего аффинного преобразования  $w_i$  области изображения. Для его поиска необходимо выполнить следующие шаги:

Разбиваем изображение  $I$  на прямоугольные ранговые блоки  $R_i$ . Данные ранговые блоки не перекрываются и покрывают всю область изображения, они могут быть одного размера, но могут также иметь адаптивное разбиение с переменным размером блоков.

$$I = \cup R_i \quad i \in [1, n), \quad R_i \cap R_j = 0, \quad i \neq j$$

где  $n$  – количество ранговых блоков.

Покрываем изображение  $I$  множеством прямоугольных доменных блоков  $D_j$ . Эти доменные блоки могут перекрываться, иметь разный размер и их количество может достигать сотни или тысячи единиц.

$$D_j \subseteq I, \quad j \in [1, m), \quad D_i \cap D_j \neq 0, \quad i \neq j$$

где  $m$  – количество доменных блоков.

Для каждого рангового блока  $R_i$  мы находим сжимающее аффинное преобразование  $w_i$ , которое наилучшим образом аппроксимирует данный ранговый блок в доменный блок  $D_j$  (рис. 1).

$$w_i(D_j) \rightarrow R_i$$

При этом каждый ранговый блок  $R_i$  сравнивается со всеми доменными блоками  $D_j$  для нахождения наи-

лучшего совпадения между ними. Данные этих преобразований сохраняются для дальнейшего восстановления исходного изображения.

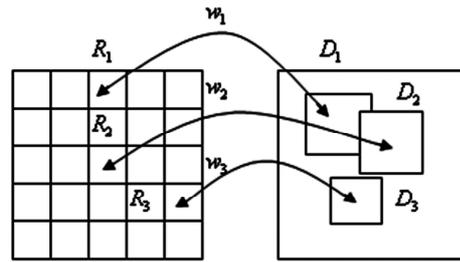


Рис. 1. Отображение ранговых блоков в доменные блоки

Сжимающее аффинное преобразование  $w_i$  – это сочетание геометрических преобразований и преобразований яркости и контраста области изображения. Матрица преобразования  $w_i$  между ранговым и доменным блоком выглядит следующим образом

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix},$$

где  $i \in [1, n)$ , коэффициенты  $a_i, b_i, c_i$  и  $d_i$  – геометрическое масштабирование и изометрическое преобразование между ранговым  $R_i$  и доменным  $D_j$  блоками,  $e_i$  и  $f_i$  – координаты смещения между ранговым  $R_i$  и доменным  $D_j$  блоками относительно  $x$  и  $y$  соответственно,  $s_i$  – разность контраста,  $o_i$  – разность яркости между пикселями рангового  $R_i$  и доменного  $D_j$ .

Для того чтобы аффинное отображение доменной области на ранговую область было сжимающим, размеры доменной области должны превышать размеры ранговой области. В процессе данного отображения каждый доменный блок уменьшается в размерах до рангового блока, он также может быть повернут или отражен для лучшего совпадения с ранговым блоком. Также вычисляется разность контраста и яркости между ними. Эти изменения представляют собой сжимающиеся аффинные преобразования доменных блоков (рис. 2).

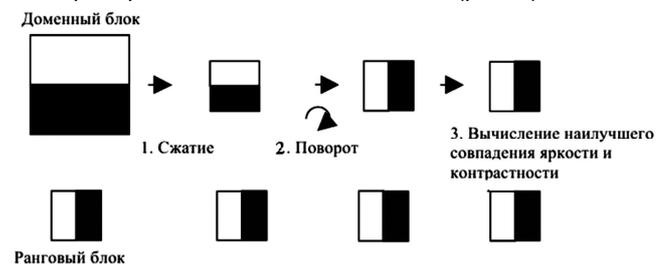


Рис. 2. Схема фрактального кодирования изображения

Для оценки ошибки приближения между ранговым  $R_i$  и доменным  $D_j$  блоками используют среднеквадратичную функцию [3]:

$$E = \sum_{i=1}^N \sum_{j=1}^M (s \cdot d_{i,j} + o - r_{i,j})^2, \quad (1)$$

где  $r_{i,j}$  и  $d_{i,j}$  – соответствующие значения пикселей

ранговой и доменной области,  $N$  и  $M$  – количество строк и столбцов в прямоугольной области.

Контрастность  $s$  и яркость  $o$  определяются следующим образом:

$$s = \frac{a}{b}, \quad o = \bar{r} - \left(\frac{a}{b}\right)\bar{d},$$

где

$$a = \sum_{i=1}^N \sum_{j=1}^M (d_{i,j} - \bar{d}) \cdot (r_{i,j} - \bar{r}),$$

$$b = \sum_{i=1}^N \sum_{j=1}^M (d_{i,j} - \bar{d})^2;$$

$$\bar{d} = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M d_{i,j};$$

$$\bar{r} = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M r_{i,j}.$$

Хорошее сжатие и качество восстановленного изображения зависят от возможности найти сжимающие аффинное преобразование между доменными и ранговыми блоками с минимальной ошибкой приближения. Совокупность сжимающих аффинных преобразований  $\tau$  представляет собой аффинный коллаж:

$$\tau = \bigcup_{i=1}^N w_i.$$

В ходе фрактального кодирования изображения, для каждой ранговой области, ищется доменная область, которая наилучшим образом аппроксимирует эту ранговую область:

$$E = \min(R_i, \tau(D)).$$

Параметры полученных сжимающих аффинных преобразований представляют собой кодовую книгу, необходимую для восстановления исходного изображения.

Для восстановления исходного изображения из кодовой книги необходимо произвести следующие шаги:

Возьмем произвольное исходное изображение – однотонное или шум. Размеры изображения не имеют значения.

Формируем второе изображение, которое является результатом отображения аффинного коллажа  $\tau$  на исходное изображение.

Меняем местами первое изображение на второе.

Процесс продолжается до тех пор, пока ошибка между этими двумя изображениями не будет меньше определенной величины. На практике для этого вполне достаточно 6-8 циклов.

### Локальные бинарные шаблоны

Оператор локальных бинарных шаблонов позволяет произвести классификацию областей изображения. Впервые данный оператор был предложен Т. Ojala, М. Pietikainen, Т. Maehraa [4] и в дальнейшем развит в работах [5, 6]. Данный оператор преобразует изображение в массив двоичных кодов, описывающих окрестности элементов изображения. Использование данного оператора теоретически простой и одновременно эф-

фективный метод анализа текстуры. Он показывает хорошие результаты во многих эмпирических исследованиях. Может быть представлен как подход, унифицирующий статистические и структурные методы анализа текстуры. Текстура, с его точки зрения, описывается микро-примитивами (textons) и статистикой их размещения.

Сущность оператора локальных бинарных шаблонов заключается в следующем. Рассмотрим окрестность 3x3 вокруг пикселя изображения. Значение центрального пикселя служит пороговой величиной для окружающих его соседних пикселей. Если величина соседнего пикселя имеет значение выше или равное значению центрального пикселя, то ему присваивается значение 1, а если его величина меньше значения центрального пикселя присваивается значение 0. В данной позиции пикселя  $(x_c, y_c)$  оператор локальных бинарных шаблонов определен как упорядоченный набор двоичных сравнений интенсивности пикселя между центральным пикселем и восемью смежными пикселями. Затем, например, по часовой стрелке с левого верхнего угла полученные значения соседних пикселей умножаются на биномиальные веса, приписываемые соответствующим пикселям, и суммируются для получения десятичного кода центрального пикселя (рис. 3).

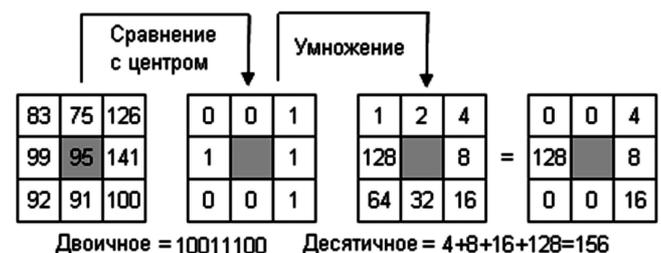


Рис. 3. Вычисление оператора локальных бинарных шаблонов

Окрестность вокруг центрального пикселя состоит из 8 соседних пикселей. Это дает возможность в общей сложности получить  $2^8 = 256$  различных двоичных кодов. Эта возможность зависит от относительных значений тона в центральном пикселе и соседних с ним смежных пикселях. Двоичные числа или их десятичный эквивалент могут быть ассоциированы с центральным пикселем и использоваться как характеристика локальной структуры текстуры изображения вокруг данного пикселя. Следует обратить внимание, что вес каждого бита в двоичном коде, полученном с помощью оператора локальных бинарных шаблонов, имеет одинаковую значимость, а последовательность из двух и более бит может иметь совершенно разный смысл.

Десятичная форма получившегося 8-битного двоичного кода (LBP кода) может быть выражена следующим образом:

$$LBP(x_c, y_c) = \sum_{n=0}^7 s(i_n - i_c) 2^n,$$

где  $i_c$  – значение центрального пикселя  $(x_c, y_c)$ ,  $i_n$  – значения восьми соседних пикселей.

Функция  $s(x)$  определена как:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases}$$

Данный код можно истолковать как индекс ядра структуры текстуры изображения размерностью  $3 \times 3$ , которое суммирует локальную пространственную структуру изображения. По определению оператор локальных бинарных шаблонов не зависит от любого монотонного преобразования тона изображения, которое сохраняет порядок интенсивности пикселя в локальной области и, следовательно, является менее чувствительным к изменениям освещенности. Оператор локальных бинарных шаблонов также инвариантен к масштабированию, вращению и переносу. На рис. 4 изображен циклический поворот двоичного кода центрального пикселя.

0	0	1	10011100 = 156	11001001 = 201
1		1	01001110 = 78	11100100 = 228
0	0	1	00100111 = 39	01110010 = 114
			10010011 = 147	00111001 = 57

Рис. 4. Циклический поворот двоичного кода

Благодаря этим своим отличительным свойствам и своими очень низкими вычислительным затратам, оператор локальных бинарных шаблонов становится довольно эффективным методом для анализа структуры текстуры изображения.

### Эксперимент

При фрактальном кодировании происходит поиск наилучшего соответствия между ранговым и доменным блоками. Этот процесс требует наибольших вычислительных затрат. Есть несколько подходов для сокращения вычислительных затрат и увеличения скорости кодирования. В настоящее время данные работы ведутся по нескольким направлениям исследований. Процесс классификации доменных областей одно из таких направлений исследований. Классификация позволяет значительно ускорить процесс кодирования за счет значительного уменьшения количества доменных областей используемых для поиска подобия.

Рассмотрим предложенную схему фрактального кодирования, использующую классификацию областей изображения с применением оператора локальных бинарных шаблонов.

#### Шаг 1. Предварительные вычисления.

Создание библиотеки ранговых областей.

Изображение разбивается на не перекрывающиеся области размером  $3 \times 3$  пикселя.

Для классификации каждой области, вычисляем двоичный (LBP) код её центрального пикселя, используя оператор локальных бинарных шаблонов.

Создадим библиотеку LBP кодов для каждой ранговой области.

Создание библиотеки доменных областей.

Уменьшим исходное изображение в два раза.

Вычисляем для каждого пикселя изображения, с помощью оператора локальных бинарных шаблонов, его LBP код. Полученные LBP коды будут классифицировать доменные области. Создадим библиотеку LBP кодов для каждой доменной области.

Для крайних и угловых пикселей недостающие соседние пиксели можно принять за 0 или дополнить циклически на противоположный край изображения.

#### Шаг 2. Кодирование ранговых областей.

Из библиотеки ранговых областей берем LBP код выбранной ранговой области.

Из библиотеки доменных областей выбираем только те области, у которых их LBP код равен LBP коду выбранной ранговой области. По формуле (1) производим вычисление ошибки приближения между ранговой областью и всеми доменными областями с тем же LBP кодом. Для доменной области с наименьшей ошибкой приближения сохраняем полученные коэффициенты преобразования для дальнейшего восстановления исходного изображения.

При отсутствии в библиотеке доменных областей области со значением LBP кода ранговой области производим циклический поворот LBP кода ранговой области (рис. 4) и повторяем поиск доменной области в библиотеке с новым LBP кодом. Если в результате полного циклического поворота, отсутствует полное соответствие между LBP кодами ранговой и доменной области, то можно взять доменные области с LBP кодом, близкие по величине LBP коду ранговой области. Находим параметры аффинного преобразования для всех ранговых блоков и сохраняем их в виде аффинного коллажа для последующего восстановления исходного изображения.

Теперь сравним сложность работы данного подхода с классическим методом фрактального кодирования. Предположим, что  $N$  – количество ранговых областей, а  $M$  – количество доменных областей. Исходя из классического алгоритма фрактального кодирования, чтобы найти доменную область, имеющую наименьшую ошибку приближения с ранговой областью, необходимо провести полный перебор всех доменных областей библиотеки. При этом сложность классического метода фрактального кодирования равняется  $O(N * M)$ . В подходе, описанном в данной работе, выбор доменной области ограничивается областями того же класса, что и класс рангового блока. При этом сложность метода фрактального кодирования с классификацией областей изображения с применением локальных бинарных шаблонов равняется  $O(N * \log_2 M)$ . Видно существенное увеличение скорости работы предложенного алгоритма фрактального кодирования на основе оператора локальных бинарных шаблонов.

Для оценки качества восстановленного изображения используем два критерия, наиболее часто используемые для измерения уровня искажений при сжатии изображений:

$PSNR$  – пиковое отношение сигнала к шуму [8]. Данный критерий является инженерным термином, означаящим соотношение между максимумом возможного значения сигнала и мощностью шума, искажающего значения сигнала. Проще всего его определить через среднеквадратичную ошибку MSE.

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (I_{i,j} - K_{i,j})^2,$$

где два монохромных изображения  $I$  и  $K$  размера  $m \times n$ , одно из которых считается зашумленным приближением другого:

$$PSNR = 20 \log_{10} \left( \frac{2^B - 1}{\sqrt{MSE}} \right),$$

где  $B$  разрядность бит изображения.

$SSIM$  – индекс структурного сходства [9]. Является одним из методов измерения схожести между двумя изображениями.  $SSIM$ -индекс это метод полного сопоставления, он проводит измерение качества на основе исходного изображения и учитывает «восприятие ошибки» благодаря учёту структурного изменения информации. Идея данного метода заключается в том, что пиксели имеют сильную взаимосвязь, особенно когда они близки пространственно:

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$

где  $\mu_x$  – среднее  $x$ ,  $\mu_y$  – среднее  $y$ ,  $\sigma_x^2$  – дисперсия  $x$ ,  $\sigma_y^2$  – дисперсия  $y$ ,  $\sigma_{xy}$  – ковариация  $x$  и  $y$ ,  $c_1 = (k_1L)^2$ ,  $c_2 = (k_2L)^2$  – две переменные,  $L = (2^{(bits \text{ per pixel})} - 1)$  – динамический диапазон пикселей,  $k_1 = 0,01$ ,  $k_2 = 0,03$  – константы.

Полученный  $SSIM$ -индекс лежит в пределах от  $-1$  до  $+1$ . Значение  $+1$  достигается только при полной аутентичности образцов.

При проведении экспериментов использовался пакет прикладных программ MatLab на базе компьютера с процессором Intel Pentium Dual-Core с тактовой частотой 1,8 ГГц и системной памятью 1 Гб.

Таблица 1. Сравнение методов фрактального кодирования

Изображение	Классический			LBP		
	$T_c$	PSNR	SSIM	$T_c$	PSNR	SSIM
Airplane	61,3	35,03	0,9879	1,8	32,19	0,9859
Baboon	61,2	29,66	0,9594	1,7	27,76	0,9497
Barbara	59,3	32,55	0,9833	1,8	31,04	0,9796
Cameraman	52,6	30,64	0,9780	1,6	28,86	0,9483
House	48,3	37,78	0,9875	1,4	34,57	0,9849
Lena	59,0	35,63	0,9900	1,4	32,35	0,9842
Peppers	60,2	33,34	0,9901	1,3	31,22	0,9834
Sailboat	56,4	33,54	0,9876	1,9	31,46	0,9845

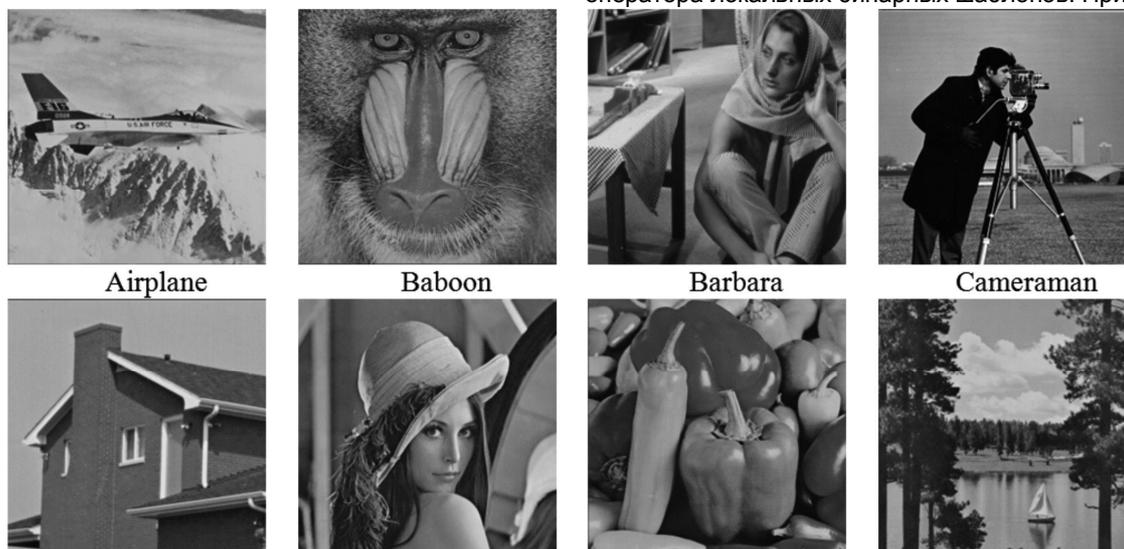


Рис. 5. Набор полутоновых изображений

Результаты экспериментов на полутоновых изображениях [7] (рис. 4), представленных ниже, приведены в табл. 1, где  $T_c$  – обозначает время работы алгоритма.

В результате проведенных экспериментальных исследований, можно сделать следующие выводы:

Использование локальных бинарных шаблонов в качестве классификатора областей изображения, позволяет значительно сократить количество сравнений между ранговыми и доменными областями.

Происходит значительное уменьшение времени работы алгоритма.

Погрешность кодирования сопоставима с погрешностью кодирования при классическом методе.

Незначительная потеря качества восстановленного изображения при использовании предложенного метода фрактального кодирования может быть компенсирована за счет увеличения количества просматриваемых доменных областей.

При расширении количества доменных областей, дополнительных вычислительных затрат почти не требуется, т.к. вычисление LBP кода выполняется достаточно легко.

### Заключение

В работе рассмотрено фрактальное кодирование изображения на основе классификации областей изображения оператором локальных бинарных шаблонов. Оператор локальных бинарных шаблонов представляет простой, эффективный метод классификации текстуры полутоновых изображений, инвариантный к масштабированию, вращению и переносу. Другое важное преимущество данного подхода – вычислительная простота и универсальный подход для анализа текстуры любых полутоновых изображений.

Применение данного подхода позволяет значительно уменьшить временные и сократить вычислительные расходы при фрактальном кодировании изображений и получить хорошие экспериментальные результаты.

Предложен новый алгоритм «неполного» перебора на основе двоичных значений, полученных с помощью оператора локальных бинарных шаблонов. При исполь-

звании этого алгоритма «с потерей качества» ошибка восстановления изображения мало уступает классическому алгоритму, а по скорости превосходит его.

Получено значительное увеличение скорости работы предложенного алгоритма фрактального кодирования при сопоставимой степени сжатия.

Из всего этого следует, что требуется дальнейшее изучение оператора локальных бинарных шаблонов при фрактальном кодировании изображений для его применения при обработке больших объемов графической информации.

#### Литература

1. Barnsley M., and Hurd L., *Fractal Image Compression* // Wellesley, MA: A.K.Peters, Ltd., 1993.
2. Jacquin A.E. *Fractal image coding: A review* // *Proceedings of the IEEE*, 1993. Vol. 81(10). P. 1451-1465.
3. Welstead S., *Fractal and Wavelet Image Compression Techniques* // *Tutorial Texts in Optical Engineering*, SPIE Publications, 1999. Vol 40.
4. Ojala T., Pietikainen M., Maenpaa T. *Multi resolution gray-scale and rotation invariant texture classification with local binary patterns* // *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, Vol.24, №.7. P. 971–987.
5. Ahonen T., Hadid A., and Pietikainen M. *Face recognition with local binary patterns* // *Proc. European Conf. Computer Vision*, 2004, P. 469–481.
6. Takala V., Ahonen T., Pietikainen M. *Block-based methods for image retrieval using local binary patterns* // *Scandinavian Conference on Image Analysis*, Joensuu, Finland, 2005, P. 882–891.
7. <https://sipi.usc.edu/database/database.php?volume=misc>.
8. <https://ru.wikipedia.org/wiki/PSNR>.
9. <https://ru.wikipedia.org/wiki/SSIM>.