

УДК 004.62

В.К. Гулаков, С.Б. Клепинин

## ИСПОЛЬЗОВАНИЕ ЛОКАЛИЗОВАННЫХ ФРАКТАЛОВ ДЛЯ КОДИРОВАНИЯ И РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Рассмотрен вопрос фрактального кодирования изображения для задач распознавания изображений. Проанализировано пространственное распределение соответствующих друг другу доменных и ранговых блоков, полученных в процессе кодирования. Предложен новый метод построения фрактального кода, призванный минимизировать влияние дефектов изображения на качество распознавания.

Ключевые слова: фракталы, системы итерируемых функций, распознавание изображений, фрактальный код изображения, локализованные фракталы.

Человеческий глаз способен различать лица достаточно хорошо с минимальными усилиями при разных условиях. Существует необходимость в разных областях экономики (например, охрана общественного порядка, медицина) получить автоматическую систему распознавания, работающую с подобной скоростью и точностью. За последние 20 лет было предложено большое количество алгоритмов решения этой задачи на основе нейронных сетей [1], скрытых марковских моделей [2], самоорганизующихся карт [3] и других способов.

В данной статье представлен улучшенный метод фрактального распознавания лиц. Для получения фрактального кода [4] изображение необходимо разбить на множество ранговых и доменных блоков. Для каждого рангового блока ищется наиболее подходящий (по некоторым критериям) доменный блок, который может быть подвержен аффинным преобразованиям. Также могут быть скорректированы контрастность и яркость. Сжимающее отображение, преобразующее доменный блок в ранговый, называется фрактальным кодом рангового блока. Фрактальный код изображения – это коллекция фрактальных кодов всех его ранговых блоков.

Распределение пар «доменный блок - ранговый блок» зависит от содержания изображения и используемого алгоритма фрактального кодирования. Некоторые методы ищут лучшее совпадение пар блоков, другие - первое совпадение, удовлетворяющее некоторому пороговому значению. Доменный и ранговый блоки могут принимать форму квадрата, прямоугольника, треугольника. Размер доменного блока может быть как фиксированным, так и переменным. Все эти параметры можно использовать для того, чтобы сделать фрактальный код нечувствительным к небольшим изменениям в изображении. Так как нет очевидной связи между расположением соответствующих ранговых и доменных блоков, то изменения в одной части изображения могут повлиять на фрактальный код другой части изображения.

В этой статье предлагается метод фрактального кодирования, позволяющий сделать фрактальный код одной части изображения нечувствительным к случайным изменениям других частей того же изображения.

**Локализованные фракталы.** Существует ли зависимость между позициями на изображении рангового блока и соответствующего ему во фрактальном коде доменного блока? Для ответа на этот вопрос мы построили фрактальный код для изображения в градациях серого размером 256 x 256 (использовался классический алгоритм фрактального кодирования на основе квадродерева и квадратичной метрики [5]). На рис. 1 показаны три случайным образом выбранные пары «ранговый блок - доменный блок». Как видно из рисунка, доменные блоки расположены на достаточном расстоянии от ранговых, интуитивно нельзя выявить логику взаимоположения. Далее для всех пар во фрактальном коде была вычислена разница между X-координатой центра рангового блока и центра соответствующего доменного блока. Распределение полученных величин представлено на рис. 2.

Из графика можно сделать вывод о том, что не существует предрасположенности к ассоциации доменных блоков из некоторой окрестности ранговых блоков. На рис. 3 представлена аналогичная картина распределения расстояний по X-координате для пар случайно выбранных точек в рамках одного и того же изображения.

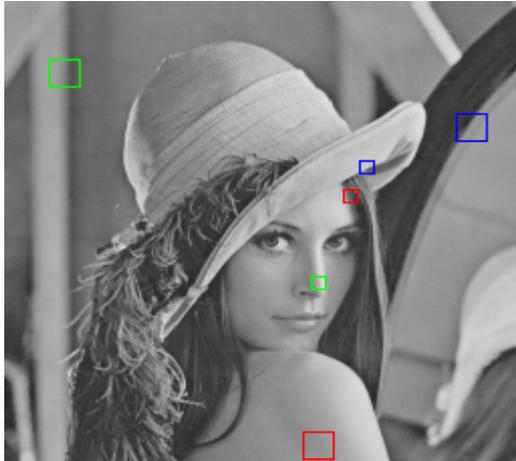


Рис. 1. Расположение трех произвольных пар «ранговый блок - доменный блок» на изображении Lena (пару составляют квадраты одного типа: меньшего размера – ранговый блок, большего – доменный)

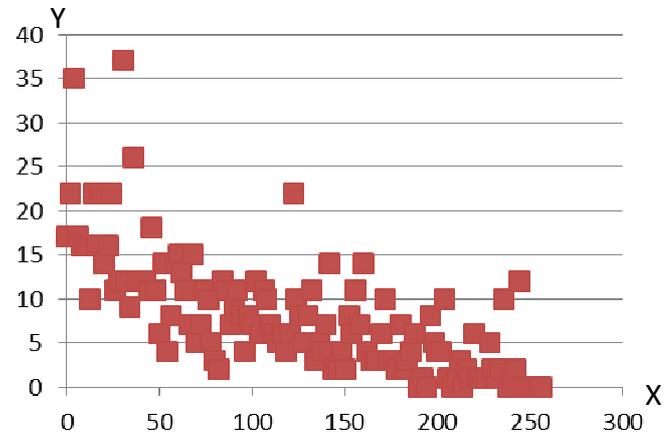


Рис. 2. Распределение расстояний между центрами ранговых и доменных блоков, составляющих фрактальный код (ось X – расстояние между центрами блоков в пикселях, ось Y – количество пар «ранговый блок - доменный блок»)

Ювал Фишер рассматривает фрактальное кодирование с позиции задачи сжатия изображений. Он также задаётся вопросом о наличии связи положений рангового и доменного блоков и приходит к выводу, что такой зависимости в общем случае нет [6]. С точки зрения алгоритмов сжатия изображений отсутствие таких закономерностей действительно может не играть большой роли. К сожалению, мы не можем сказать то же самое относительно задачи распознавания образов.

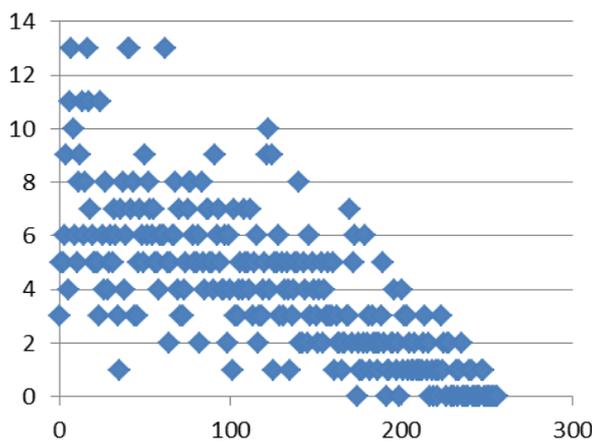


Рис. 3. Распределение расстояний между парами случайно выбранных точек изображения (количество пар совпадает с количеством ранговых блоков во фрактальном коде изображения Lena)

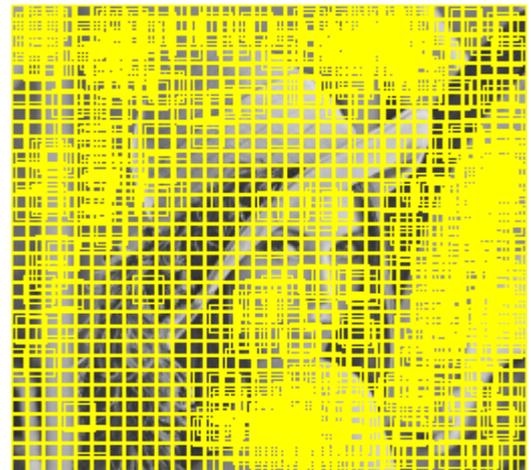


Рис. 4. Расположение доменных блоков фрактального кода на кодируемом изображении

Если связь между доменным и ранговым блоками случайна, то небольшие изменения в одной части изображения могут непредсказуемо повлиять на фрактальный код (тройка «ранговый блок, доменный блок, преобразование»). Риски такого искажения увеличиваются из-за следующих двух факторов:

- Один доменный блок может соответствовать нескольким ранговым блокам (при использовании разных преобразований), следовательно, дефект такого доменного блока приведет к изменению кода для всех соответствующих ему ранговых блоков.

- Доменные блоки, попавшие во фрактальный код, располагаются неравномерно на изображении. На рис. 4 представлена схема распределения доменных блоков, используемых во фрактальном коде тестового изображения Lena. Явно выделяются 3-4 области с более плотной сеткой. Можно сделать вывод, что изменение изображения в этих областях кардинальным образом повлияет на весь фрактальный код.

С другой стороны, Фишер говорит [6], что традиционные методы фрактального кодирования для определенных типов изображений имеют тенденцию к выбору пространственно близких доменного и рангового блоков, но это не всегда происходит. Эксперименты показывают, что для определенных типов изображений ранговые блоки из некоторого сегмента с большей вероятностью будут соответствовать доменным блокам из этого же сегмента. Это объясняется подобностью свойств (например, одинаковой текстурой). Такая особенность делает фрактальное кодирование устойчивым к небольшим изменениям или дефектам [7] (например, изменение настроения - в случае распознавания лиц - влечет только маленькие изменения вокруг губ, глаз, что не влияет на остальной фрактальный код).

Для усиления этого свойства предлагается новый способ фрактального кодирования, который выбирает доменные блоки только из некоторой окрестности ранговых блоков (на рис. 6 показан пример такого распределения). Это гарантирует, что любые перемены в определенном сегменте изображения повлияют только на фрактальный код этого сегмента и не отразятся на коде других частей, код различных областей изображения станет независимым. Получаемые таким образом фракталы называются *локализованными фракталами*. Полный фрактальный код изображения получается объединением всех локализованных фракталов. Однако получение полного фрактального кода не является целью для задачи распознавания образов. Достаточно найти локализованные фрактальные коды ключевых областей изображения и использовать только их для сравнения.

**Алгоритм распознавания образов на основе локализованных фракталов.** Принципиальная схема реализации системы фрактального распознавания образов представлена на рис. 5.

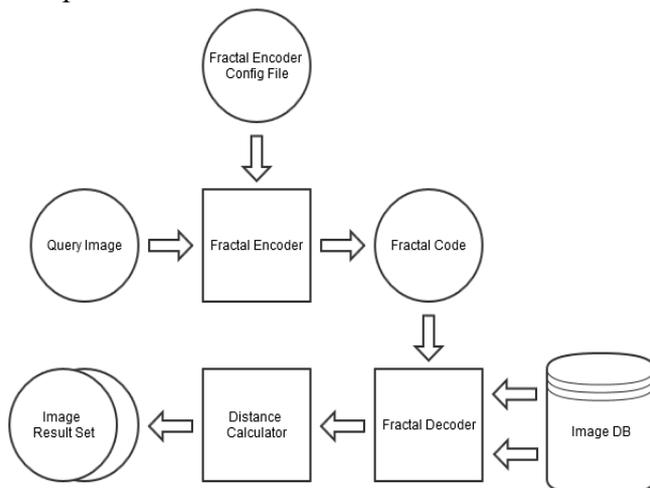


Рис. 5. Принципиальная схема системы распознавания образов на основе локализованных фракталов

Изначально искомое изображение (элемент Query Image на схеме) поступает в модуль (элемент Fractal Encoder), который вычисляет фрактальный код. В нашем случае это будет совокупность фрактальных кодов локализованных областей, а не полный код изображения. Как правило, параметры алгоритма (размер ранговых и доменных блоков, шаг доменных блоков, используемые трансформации и др.) хранятся в конфигурационном файле (блок Fractal Encoder Config File). После получения локализованных фрактальных кодов они применяются для каждого изображения из поисковой базы (элемент Image DB) в специальном модуле, который работает по принципу обычного фрактального декодера (элемент Fractal Decoder). Далее измеряется расстояние (в элементе Distance Calcula-

tor) между локализованными областями оригинального изображения из базы и его проекции, полученной после применения фрактального кода. Такой метод сравнения детально описан в работе [4]. Основная идея заключается в том, что чем ближе изначально изображение к тому, для которого построили фрактальный код, тем расстояние между оригиналом и первой проекцией, первой и второй проекциями будет меньше. Важно, чтобы метрика, используемая для измерения расстояний между проекциями, совпадала с метрикой, применяемой в процессе построения фрактального кода, при сравнениях ранговых и доменных блоков. В результате получается упорядоченное по сходству с искомым множество изображений (элемент Image Result Set).

Представленная на рис. 5 схема была реализована на языке программирования Java с применением JDK 1.7.0.51. В качестве базы изображений использовалась нормализованная база Yale Face Database (<http://vision.ucsd.edu/datasetsAll>). Рассмотрим более подробно шаги алгоритма:

1. Локализация фрактальных областей, таких как глаза, нос, губы и остальные части лица, вручную для одного произвольного изображения в нормализованной базе. Эта информация в дальнейшем будет использована для всех изображений в базе.

2. Подготовка конфигурационного файла (также ручной шаг), который содержит следующую информацию:

- Координаты локализованных фрактальных областей.
- Максимальный размер рангового блока.
- Максимальная глубина квадродерева.

Конфигурационный файл может быть любого удобного формата. В нашей реализации это обычный property-файл, содержащий набор «ключ - значение».

3. Нахождение фрактального кода для каждой локализованной области:

- Построение доменного пула. Покрываем фрактальную область сеткой пересекающихся доменных блоков квадратной формы. Размер доменного блока выбирается в два раза больше размера рангового блока, с которым он в дальнейшем будет сравниваться. Соответственно наш пул распадается на  $n$ -е количество слоев, в каждом из которых размер доменных блоков одинаковый. Число слоев равно максимальной глубине квадродерева, указанной в конфигурационном файле. Шаг, через который помещается следующий доменный блок, в 4 раза меньше размера блока. Далее каждый доменный блок урезается в два раза (чтобы точно совпадать по размеру с ранговым в момент сравнения) и трансформируется использованием набора аффинных преобразований:

- поворот на 90, 180, 270 градусов;
- отражение относительно вертикальной оси, горизонтальной оси, главной диагонали, второстепенной диагонали.

Полученные преобразованные блоки, как и оригинальный, помещаются в соответствующий слой доменного пула.

- Построение квадродерева ранговых блоков. Фрактальная область покрывается сеткой непересекающихся ранговых блоков. Размер берется из конфигурационного файла. Далее ранговый блок сравнивается со всеми блоками из слоя доменного пула, который соответствует размеру данного рангового блока. Находится наилучшее совпадение, запоминается. Затем ранговый блок разбивается на 4 дочерних, для которых повторяется процедура поиска наиболее подходящего доменного блока. Процедура останавливается, когда достигается максимальная глубина квадродерева. Далее необходимо определиться с наилучшим покрытием области оригинального рангового блока. Для этого полученное дерево обходится от вершины к листьям и сравниваются расстояние до найденного доменного блока родительской вершины и суммарное расстояние всех (четырех) ее детей. Выбирается наименьшее.

Используемая метрика:

$$Err = \sum_{i=1}^n (s \cdot d_i + o - r_i)^2,$$

где  $d_i$  – интенсивность  $i$ -точки доменного блока;  $r_i$  – интенсивность  $i$ -точки рангового блока;  $s$  – изменение контраста;  $o$  – сдвиг яркости.

Параметры яркости и контраста подбираются индивидуально для каждой пары «ранговый блок - доменный блок» по формулам (выведение было описано в [4])

$$s = \frac{\alpha}{\beta};$$

$$o = \bar{r} - \left(\frac{\alpha}{\beta}\right) \cdot \bar{d},$$

где

$$\bar{d} = \frac{1}{n} \cdot \sum_{i=1}^n d_i; \quad \bar{r} = \frac{1}{n} \cdot \sum_{i=1}^n r_i,$$

$$\alpha = \sum_{i=1}^n (d_i - \bar{d}) \cdot (r_i - \bar{r}); \quad \beta = \sum_{i=1}^n (d_i - \bar{d})^2.$$

4. Сохранение фрактальных кодов локализованных областей в файл. Опциональный шаг, присутствует в нашей реализации для возможности анализа фрактального кода. Используется CSV-формат. Файл содержит следующие столбцы:

- X-координата рангового блока.
- Y-координата рангового блока.
- Размер рангового блока.
- X-координата доменного блока.
- Y-координата доменного блока.
- Размер доменного блока.
- Контрастность.
- Яркость.
- Набор преобразований для доменного блока.
- Идентификатор локализованной фрактальной области.

Таким образом, каждая строчка в файле означает одно отображение рангового блока в доменный блок.

5. Применение полученного фрактального кода к изображениям из базы.

• Вырезаем из изображения область, соответствующую координатам и размеру доменного блока.

• Преобразуем полученный кусок, используя трансформации, указанные во фрактальном коде.

• Вставляем полученный блок, используя координаты рангового блока из фрактального кода.

Процедура применяется для всех ранговых блоков в локализованном фрактальном коде.

6. Измеряем расстояние (используется квадратичная метрика) между локализованными фрактальными областями на оригинальном изображении из базы и получившемся после применения фрактального кода. Суммируем расстояния всех фрактальных областей в рамках одного изображения.

7. На основе цифр, получившихся на предыдущем шаге, сортируем базу изображений в возрастающем порядке. Первые элементы упорядоченного множества и будут наиболее близкими к искомым изображениям (рис. 6).

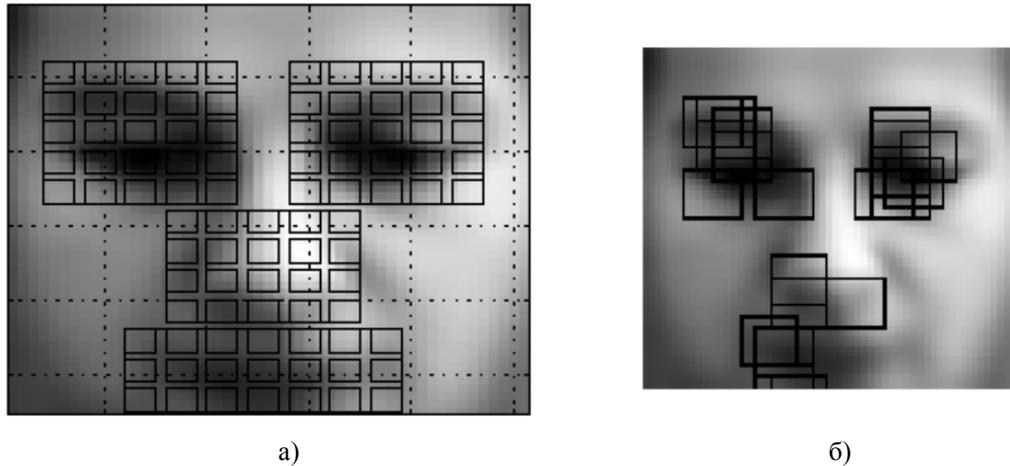


Рис. 6. Расположение ранговых и доменных блоков в локализованных фрактальных областях (показаны только ранговые блоки первого уровня, состояние до построения квадродерева): а – ранговые блоки; б - доменные блоки

Проводился эксперимент: на тестовые изображения (как в базе, так и искомые) наносились дефекты (затемнения, осветления, инородные фигуры), далее эти изображения подавались на вход реализациям классического алгоритма фрактального распознавания и алгоритма на основе локализованных фракталов. Оказалось, что классический алгоритм дает ошибку в ~20% случаев, в то время как алгоритм на основе локализованных фракталов - всего в ~5%.

Итак, был предложен новый метод фрактального кодирования, основанный на локализованных фракталах. Изображение разбивается на фрактальные области, коды для которых строятся независимо друг от друга. Для каждого рангового блока ищется доменный блок только из той же фрактальной области. Для увеличения доменного пула каждого локализованного фрактала мы используем перекрывающееся разбиение, варьируем размеры доменных блоков, используем перевернутые и отраженные версии каждого блока. В дальнейшем имеет смысл исследовать возможность сравнения изображений не по всем фрактальным областям, а по некоторым (например, улыбка человека почти не затрагивает носовую область, следовательно, ее можно использовать для минимизации ошибок при распознавании эмоций).

#### СПИСОК ЛИТЕРАТУРЫ

1. Lin, S. H. Face Recognition/Detection by Probabilistic Decision-Based Neural Network / S. H. Lin, S. Y. Kung // IEEE Trans. On Neural Networks. – 1997. - №8(1). – P.114-132.
2. Гулятьева, Т.А. Скрытые марковские модели с одномерной топологией в задаче распознавания лиц / Т.А. Гулятьева, А.А. Попов. - Новосибирск: НГТУ, 2006.
3. Kohonen, T. The Self-organizing Map / T. Kohonen // Proceedings of the IEEE. – 1990. - № 78(9). – P.1464-1480.
4. Гулаков, В.К. Методы распознавания изображений на основе фрактального кодирования / В.К. Гулаков, С.Б. Клепинин // Вестн. Брян. гос. техн. ун-та. – 2012. - №4(36). – С. 54 – 61.
5. Jacquin, A. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations / A. Jacquin // IEEE Transactions on image processing. – 1992. – №1(1). – P.18-30.
6. Fisher, Y. Fractal Image Compression: Theory and Application / Y. Fisher. - Springer-Ver, 1994.
7. Ebrahimpour, H. Face recognition using fractal codes / H. Ebrahimpour, V. Chandran, S. Sridharan // Proceedings of International Conference on Image Processing. – 2007. – P. 58–61.

Материал поступил в редколлегию 3.06.14.