

УДК 004.92

ПОИСК ИЗОБРАЖЕНИЯ ПО ИЗОБРАЖЕНИЮ-ЗАПРОСУ С ПОМОЩЬЮ ФРАКТАЛЬНОГО КОДИРОВАНИЯ

А.В.Савкина

АННОТАЦИЯ

Для эффективного решения задачи поиска по изображениям-образцам и каталогизации изображений разработана система поиска изображений по изображению-образцу на основе фрактального разложения изображений, а также алгоритмы и формат фрактального кодирования изображения методом квадродерева. Основные конструктивные и технико-эксплуатационные показатели полученных результатов: хранение большого количества изображений (ограничено размером физических носителей БД), малое время доступа к хранимым изображениям. Для функционирования разработанной системы необходимо наличие ОС Windows (не ниже версии Windows 2000), библиотеки MATLAB Runtime и Microsoft SQL Server, средства разработки MATLAB 2006 и Borland Delphi¹. Система выполняет фрактальное кодирование изображения перед помещением его образа в виде квадродерева в БД, с последующим анализом квадродерева в процессе осуществления поиска. В статье, представленной на 9 страницах формата А4, содержится 6 рисунков.

КЛЮЧЕВЫЕ СЛОВА: ИТЕРИРУЕМАЯ ФУНКЦИЯ, ФРАКТАЛ, ХРА-
НЕНИЕ ИЗОБРАЖЕНИЯ, ПОИСК ПО ОБРАЗЦУ, СРАВНЕНИЕ.

Поиск изображений порождает некоторые сложные проблемы. Как правило, очертить сферу поиска сложно, если не дана некоторая модель содержимого коллекции. Механизм навигации – это необходимый элемент взаимодействия человека с коллекцией, но только в том случае, если коллекция соответствующим образом организована. В идеале средства навигации должны выполнять следующее:

- отображать сходные изображения (это означает, что изображения могут выглядеть похожими, иметь сходный внешний вид, располагаться рядом в коллекции, иметь сходное содержимое и т.д.) так, чтобы их сходство было очевидным;
- отображать представление кластеров изображений, чтобы пользователь мог представить, что имеется в данной коллекции (например, сходные кластеры могут располагаться рядом, большие кластеры могут выглядеть большими и т.д.);
- обеспечивать некоторую разновидность взаимодействия, чтобы коллекцию можно было просматривать с различным уровнем детализации (возможно, посетителю требуется только элемент определенного кластера или некоторое обобщение всех изображений рядом с набором кластеров) и перемещаться по коллекции в различных направлениях.

Средства навигации и поиска естественно дополняют одно другое. Пользователь может вначале исследовать коллекцию, а затем провести поиск. После

поиска пользователь, возможно, захочет посмотреть на объекты, находящиеся «рядом» с любой позицией, выданной средством поиска.

Построение полезных средств навигации также требует эффективного понятия сходства изображений. Как правило, считается, что пользователь может предоставить примерное изображение того, что ему требуется, или заполнить электронную форму для поиска этого первого изображения, а затем использовать средство навигации для перемещения по коллекции, щелкая на образцах, предложенных средством. Предлагается сходство изображений формулировать через сравнение квадродеревьев, полученных в результате фрактального кодирования изображений.

Для этого необходимо разработать алгоритмы фрактального кодирования изображения и форму представления данных, полученных в результате кодирования. Это позволит использовать для анализа изображения не все данные об изображении, а только наиболее значимую его часть, что позволит решить задачу поиска более эффективным образом как с точки зрения скорости поиска, так и с точки зрения релевантности результатов поиска.

При фрактальном кодировании изображений мы пытаемся найти множество сжимающих преобразований, которые отображают доменные блоки (которые могут перекрываться) в множество ранговых блоков, которые покрывают изображение². Ранговые блоки могут быть одинакового размера, но удобнее используется адаптивное разбиение с переменным размером блоков. Ранговые блоки, показанные на рисунке справа, являются результатом разбиения методом квадродерева.

В качестве базового алгоритма фрактального кодирования изображений выберем следующий:

1. Разбиваем изображение f на неперекрывающиеся ранговые блоки $\{R_i\}$. Для упрощения реализации алгоритма в качестве ранговых блоков возьмем прямоугольники. Блоки R_i могут быть равными, но чаще используется адаптивное разбиение с переменным размером блоков. Это дает возможность плотно заполнять ранговыми блоками маленького размера части изображения, содержащие мелкие детали.
2. Покрываем изображение последовательностью доменных блоков, возможно перекрывающихся. Домены могут быть разных размеров, и обычно их количество исчисляется сотнями и тысячами. Ниже обсуждаются схемы построения множества доменных блоков.
3. Для каждого рангового блока находим домен и соответствующее преобразование, которое наилучшим образом покрывает ранговый блок. Обычно это аффинное преобразование вида $w_i(f)(x, y)$. Настраиваем параметры преобразования, такие как контрастность и яркость, для наилучшего соответствия.
4. Если достаточно точного соответствия не получилось, то разбиваем ранговые блоки на меньшие ранговые блоки. Продолжаем этот процесс до тех пор, пока или не добьемся приемлемого соответствия, или размер ранговых блоков не достигнет некоторого заранее определенного предела.

На рисунке 1 представлена блок-схема процесса фрактального кодирования.

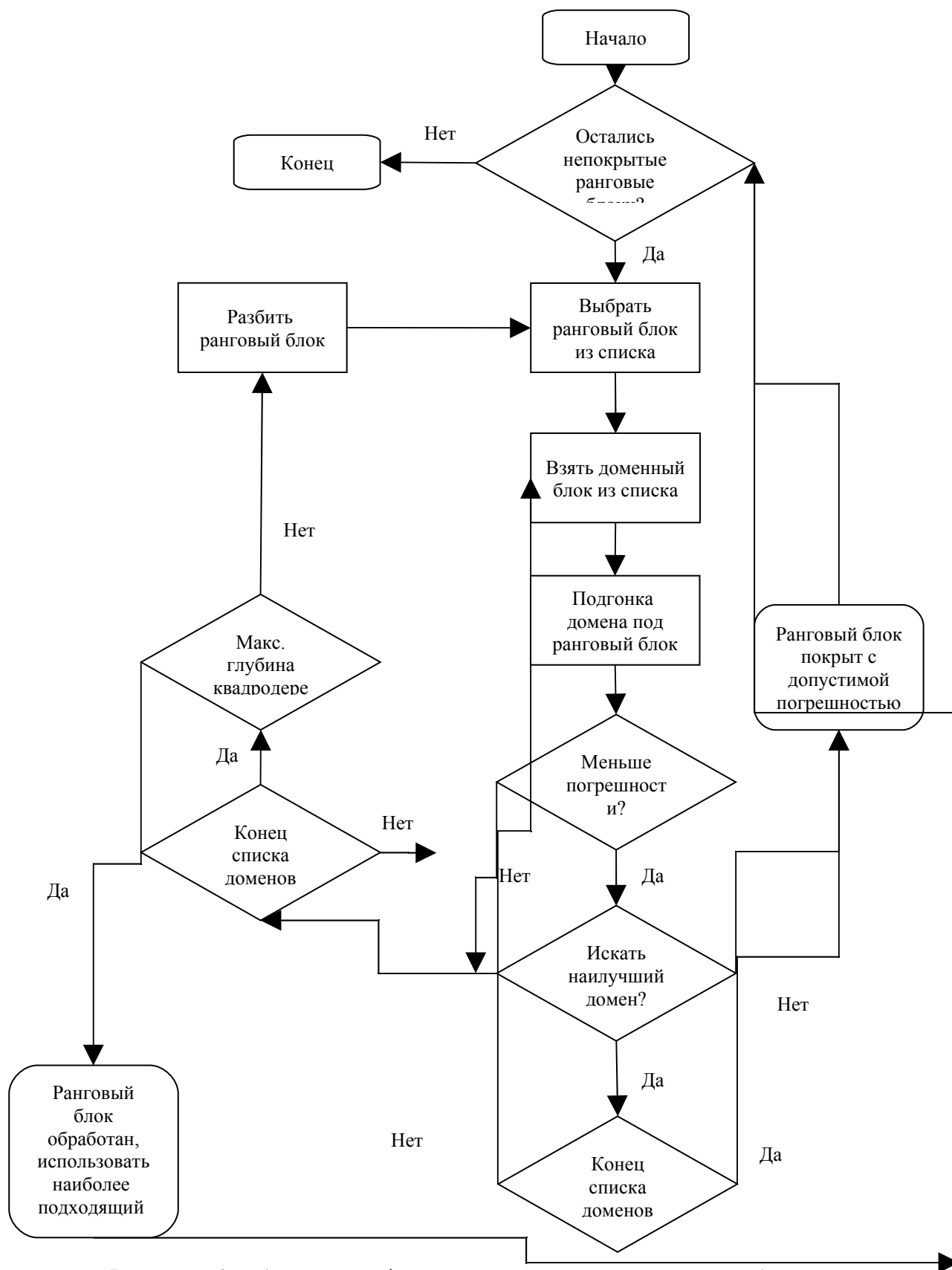


Рисунок 1 – Алгоритм фрактального кодирования изображения

При подгонке доменных блоков к ранговым блокам алгоритм реализует пространственную составляющую аффинных преобразований, включающую операции параллельного переноса, поворота и сжатия. Сжатие уменьшает размер доменного блока до размеров рангового блока. Эта операция дополняется простым усреднением по строкам и столбцам.

Шаг 3 требует наибольших вычислений. Для каждого рангового блока R_i алгоритм ищет домен D_i пространственное преобразование w'_i контрастность s_i и яркость o_i , такие, чтобы $w_i(f)$ было близко к изображению f в блоке R_i . То есть мы ищем w_i такое, чтобы величина

$$\int_{R_i} |w_i(f)(x, y) - f(x, y)|^2 dx dy$$

была небольшой. Для оцифрованных изображений этот интеграл заменяется суммированием по пикселям. Если после нахождения наилучшего w_i , величина оказывается все еще больше некоторой заранее определенной погрешности, то адаптивная схема разбиения на блоки разбивает ранговые блоки на меньшие ранговые блоки и затем поиск оптимального преобразования повторяется для этих меньших блоков. Этот процесс продолжается до тех пор, пока величина интеграла не станет меньше допустимой погрешности или пока не достигается заранее определенный минимальный размер рангового блока.

«Код» закодированного фрактальными методами изображения – это список, содержащий сведения о каждом ранговом блоке. А именно: расположение рангового блока, домен (обычно определяемый при помощи индекса), который отображается в этот ранговый блок, и параметры, описывающие преобразование доменного блока в ранговый. Таким образом, коэффициент сжатия зависит от количества ранговых блоков, а также от эффективности хранения информации о каждом ранговом блоке. Большое количество ранговых блоков обеспечивает высокое качество декодированного изображения, но за счет степени сжатия.

Для каждого рангового блока нам нужно найти доменный блок, который эффективно отображается в этот ранговый блок. Для того чтобы отображение было сжимающим, домен должен быть больше рангового блока. Хорошее сжатие зависит от возможности найти хорошее соответствие между доменными и ранговыми блоками без необходимости дополнительного разбиения ранговых блоков. Слишком дробное разбиение ранговых областей приводит к слишком большому их количеству, а это ухудшает коэффициент сжатия. В идеале нужно иметь континуум размеров и вариантов расположения доменных блоков и выбирать из него подходящие для каждого рангового блока. К сожалению, вычислительные издержки поиска среди стольких вариантов слишком велики. Задача определения системы доменов – это компромисс между необходимостью, чтобы множество доменов было достаточно большим для обеспечения возможности подбора наилучшего варианта соответствия ранговому блоку и, в то же время, достаточно маленьким, чтобы процесс поиска мог быть осуществлен за приемлемое время.

Один из методов разбиения изображения на ранговые блоки – это метод квадродерева. Вначале производится грубое разбиение, скажем деления целого изображения на четыре прямоугольника. Для каждого рангового блока ал-

горитм пытается найти домен и соответствующее сжимающее отображение, которое наилучшим образом покрывает ранговый блок. Для того чтобы обеспечить сжатие, те ранговые блоки, которые оказываются больше самого большого из доменов, разбивают на меньшие ранговые блоки. Контрастность и яркость вычисляются методом наименьших квадратов, что обеспечивает максимально возможное соответствие. Затем, если покрытие оказывается в пределах допустимой погрешности, то считается, что этот ранговый блок покрыт, и алгоритм переходит к следующему ранговому блоку. Если отклонение не укладывается в пределы допустимой погрешности, то алгоритм проверяет, была ли достигнута максимальная глубина квадродерева. Если максимальная глубина квадродерева не была достигнута, то алгоритм разбивает блок на четыре меньших ранговых блока и поиск оптимальных доменов и преобразований начинается заново для этих новых ранговых блоков. Процесс завершается, когда все ранговые блоки оказываются покрытыми – или с помощью такого подбора домена и преобразования, который обеспечивает отклонение в пределах допустимой погрешности, или путем достижения максимальной глубины квадродерева (рисунок 2).

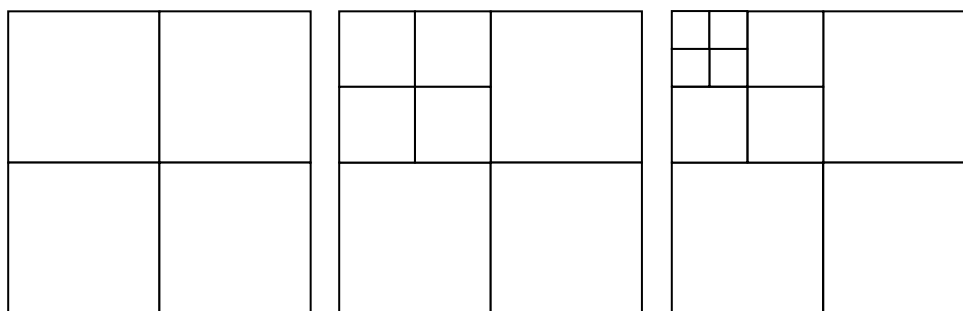


Рисунок 2 – Построение квадродерева

Заметим, что когда мы уменьшаем величину допустимой погрешности, это приводит к увеличению числа ранговых блоков, и увеличение глубины квадродерева также приводит к увеличению их числа. На рисунке показан эффект использования меньшей допустимой погрешности и большей глубины квадродерева. В каждом случае большее количество ранговых блоков означает худшую компрессию (а иногда и отсутствие компрессии вообще), но обычно лучшее качество изображения.

Для реализации алгоритма разбиения методом квадродерева необходим способ слежения за ранговыми блоками, которые получаются в результате разбиения. Алгоритм использует представленную ниже схему управления списком, при которой каждому ранговому блоку единственным образом сопоставляется индекс в квадродереве. Рассмотрим ее работу на простом примере с трехуровневым квадродеревом. Мы начнем со списка из четырех векторов, длина каждого из которых равняется максимальной глубине квадродерева, в данном случае, трем. Первая компонента первого вектора полагается равной 1, вторая компонента второго вектора – равной 2 и т.д. Таким образом, начальный список состоит из четырех векторов: $(1,0,0)$; $(2,0,0)$; $(3,0,0)$; $(4,0,0)$.

Этот список соответствует начальному разбиению, или первому уровню

квадродерева, показанному на рисунке слева. Принято считать, что список из четырех исходных векторов сопоставляется четырем ранговым блокам, расположенным по часовой стрелке, начиная с левого верхнего. Когда алгоритм осуществляет деление рангового блока, входной вектор, соответствующий этому ранговому блоку, заменяется на четыре новых входных вектора. Компоненты векторов, индексы которых меньше номера данного уровня квадродерева, остаются неизменными, а компоненты, соответствующие данному уровню, получают значения 1, ..., 4. Так, например, список, соответствующий второму уровню разбиения, показанному на рисунке в центре, будет таким: (1,1,0); (1,2,0); (1,3,0); (1,4,0); (2,0,0); (3,0,0) (4,0,0).

На рисунке 3 показан полученный по такой схеме третий уровень разбиения и соответствующий ему список индексов. Например, заштрихованный блок имеет индекс 3, 2, 4.

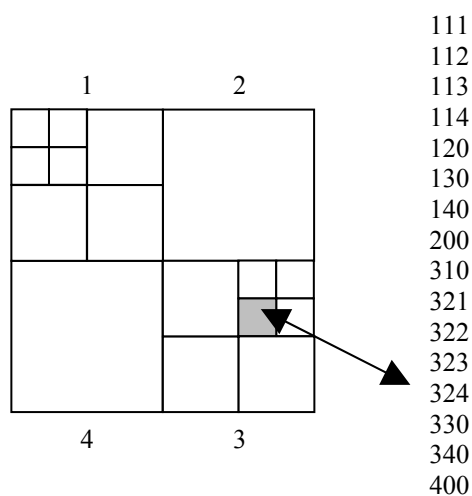


Рисунок 3 – Представление квадродерева

Главный вычислительный шаг во фрактальном кодировании – это сравнение доменной и ранговой области. Для каждого рангового блока алгоритм сравнивает варианты преобразования всех доменов к этому ранговому блоку. Это аффинные преобразования, пространственная составляющая которых жестко ограничена параллельным переносом, сжатием и одним из восьми вариантов ориентации. Варианты ориентации включают четыре поворота на 90° и зеркальное отражение в каждой ориентации.

Поиск соответствия между доменными и ранговой областями (будем называть его доменно-ранговым сопоставлением), реализованный в алгоритме, – это трехшаговый процесс; схема его представлена на рисунке 4. Во-первых, к выбранному домену применяется один из восьми (или меньше) базовых поворотов/отражений. Во-вторых, вращаемая доменная область сжимается, чтобы соответствовать размеру ранговой области. Заметим, что на практике ранговая область должна быть меньше доменной, для того чтобы суммарное отображение было сжимающим. И, наконец, методом наименьших квадратов вычисляются оптимальные параметры яркости и контрастности.

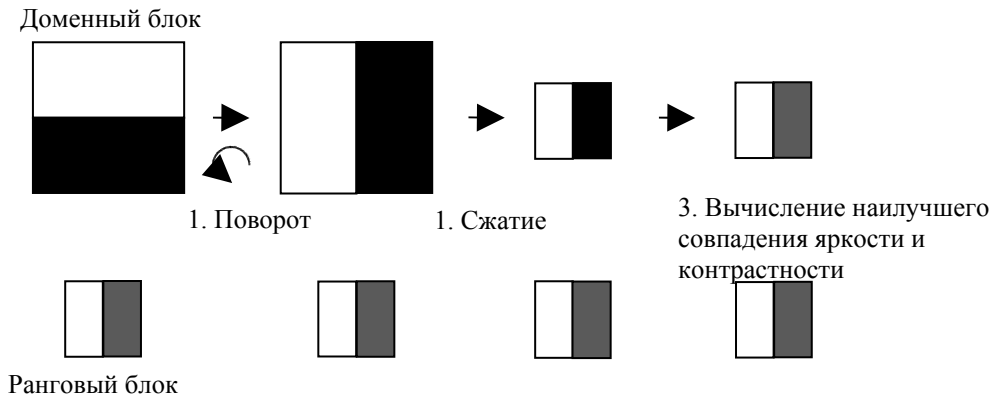


Рисунок 4 – Подбор рангового блока для доменного

Чтобы найти оптимальную контрастность s и яркость o , нам нужно найти значения s и o , которые бы минимизировали выражение

$$\sum_i \sum_j (sd_{ij} + o - r_{ij})^2$$

Здесь $\{d_{ij}\}$ и $\{r_{ij}\}$ – это соответственно значения пикселей доменной и ранговой области. Эти пиксели находятся в прямоугольных массивах с M строками и N столбцами (размер домена уже был сжат для соответствия ранговой области в этой точке).

Решением является

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

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

где

$$\alpha = \sum_i \sum_j (d_{ij} - \bar{d})(r_{ij} - \bar{r})$$

$$\beta = \sum_i \sum_j (d_{ij} - \bar{d})^2$$

$$\bar{d} = \frac{1}{NM} \sum_i \sum_j d_{ij}$$

$$\bar{r} = \frac{1}{NM} \sum_i \sum_j r_{ij}$$

Изображение декодируется путем итеративного применения преобразования W к произвольному начальному изображению g , где

$$W(g)(x, y) = w_i(q)(x, y); (x, y) \in R_i$$

Если преобразования $\{w_i\}$ были выбраны корректно, то итерация $W^{0n}(g)$ будет близка к исходному изображению f при некотором приемлемом значении n . В соответствии с теоремой о сжимающих отображениях, итерации будут сходиться независимо от выбора начального изображения³.

В результате был разработан и исследован алгоритм фрактального кодирования изображения, представляющий фрактальное разложение в виде квадродерева, выполнена реализация алгоритма в системе Delphi, создано приложение для поиска изображения по изображению-запросу в системе программирования Delphi (рисунки 5, 6), приложение реализует хранение эталонных изображений в реляционной СУБД.

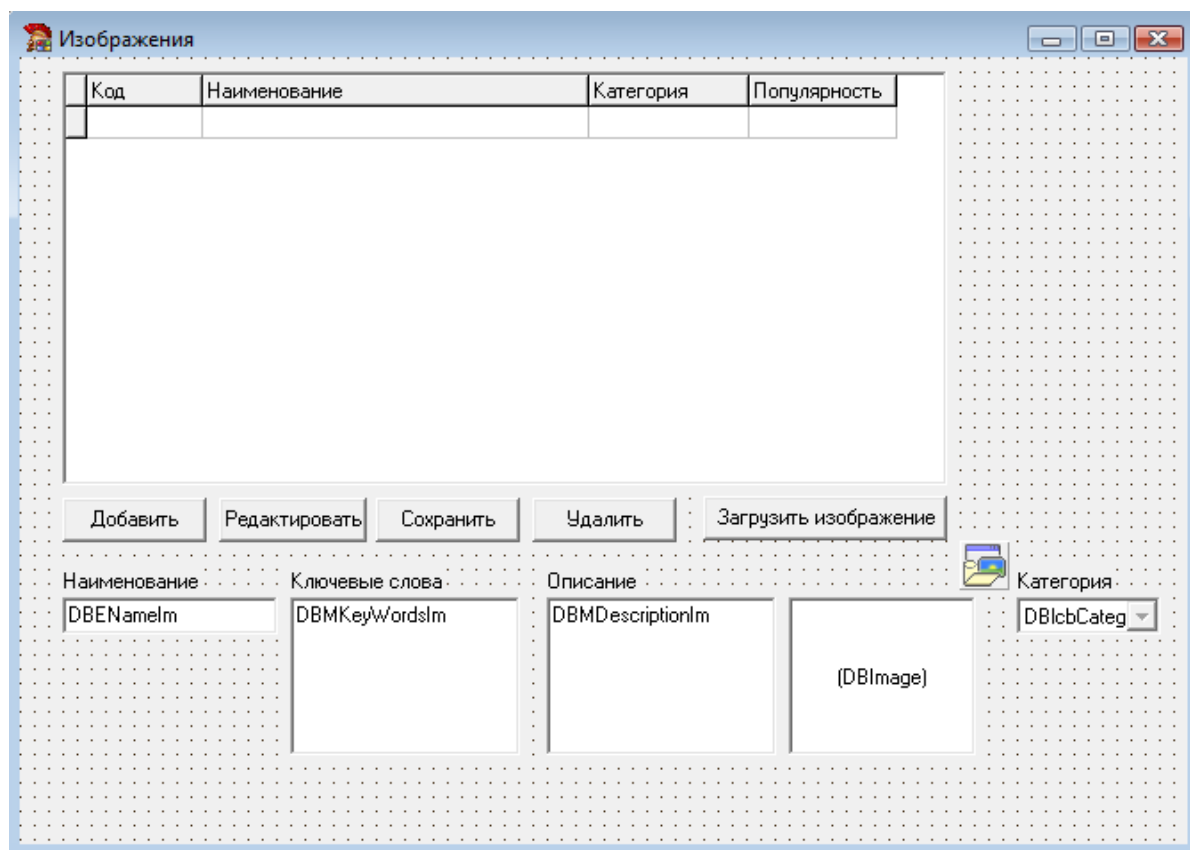


Рисунок 5 – Форма управления изображениями в БД

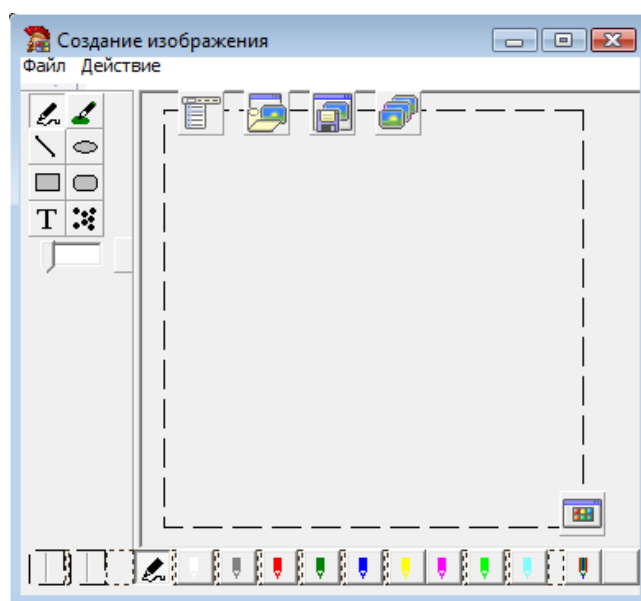


Рисунок 6 – Форма формирования изображения-запроса

Методы фрактального кодирования изображений в настоящее время яв-

ляются бурно развивающейся областью компьютерной графики, работа выполнена с учетом современных достижений в области обработки и хранения изображений.

ПРИМЕЧАНИЯ

¹ Архангельский А.Я. Программирование в Delphi 7 / А.Я. Архангельский. Тверь, 2003. С. 1152.

² Федер Е. Фракталы. Пер. с англ. / Е. Федер. М., 1991. С. 254.

³ Уэлстид С. Фракталы и вейвлеты для сжатия изображения в действии : учебное пособ. / С. Уэлстид. М., 2003. С. 320.