

**Санкт-Петербургский государственный университет**

**Математико-механический факультет**

*Кафедра системного программирования*

# Поиск изображений по признаку самоподобных локальных дескрипторов

Дипломная работа

Гладышевой Юлии Сергеевны

545 группа

Научный руководитель ..... к. ф.-м. н. Н. С. Васильева

Рецензент ..... асп. каф. сист. прогр. А. В. Бондарев

«Допустить к защите» ..... д. ф.-м. н. А. Н. Терехов

Заведующий кафедрой

Санкт-Петербург

2011

**Saint-Petersburg State University**  
**Mathematics & Mechanics Faculty**

*Software Engineering Department*

# Image retrieval method based on matching local self-similarities

Graduate paper by

Yulia Gladisheva

545 group

Scientific advisor ..... PhD N. S. Vassilieva

Reviewer ..... Postgraduate A. V. Bondarev

«Approved by» ..... Professor A. N. Terekhov

Head of Department

Saint-Petersburg

2011

## Оглавление

1. Введение .....	1
2. Постановка задачи .....	4
3. Обзор предметной области и существующих подходов .....	5
4. Поиск изображений по признаку самоподобия локальных дескрипторов	11
4.1. Общее описание подхода .....	11
4.2. Вычисление самоподобных локальных дескрипторов изображения ....	13
4.2. Адаптация метода поиска по признаку самоподобия локальных дескрипторов .....	16
5. Экспериментальное сравнение методов .....	20
5.1. Описание методов поиска, участвовавших в экспериментальном сравнении .....	20
5.2. Описание тестовой коллекции .....	21
5.3. Используемые метрики оценки .....	22
5.4. Анализ результатов экспериментов .....	23
6. Заключение .....	28
7. Список литературы .....	29

# 1. Введение

На протяжении последних десятилетий задаче поиска изображений уделяется большое внимание. Прежде всего, это связано с быстрым ростом объема мультимедийной информации, и как следствие, с возникновением задач поиска и навигации по ней. Одним из подходов к решению задачи поиска визуально похожих изображений является поиск по содержанию. Он основывается на анализе численных характеристик пикселей изображения, то есть учитывает цветовые и текстурные характеристики, форму объектов, композицию сцены и т.д.

Несмотря на разнообразие предлагаемых алгоритмов решения данной задачи, качество поиска на сегодняшний день оставляет желать лучшего. Это связано прежде всего с «семантическим разрывом», являющимся главной проблемой поиска изображений по содержанию. «Семантический разрыв» заключается в том, что люди и машины сравнивают изображения абсолютно по-разному: человек сравнивает смысловую наполненность (семантику), а машины основываются на сравнении низкоуровневых признаков, описывающих визуальные характеристики изображения (цвет, текстуру, форму объектов и т.д.).

Задача поиска изображений по содержанию является актуальной и ее решение потенциально применимо во многих областях. Например, это поиск в сети Интернет. Поисковые системы обычно не учитывают содержание изображения и осуществляют поиск по ключевым словам, построенным по контексту изображения. Использование методов поиска по содержанию могло бы предоставить возможность поиска по большему объему данных, а также повысить его качество. Поиск по содержанию применим в медицинской диагностике заболеваний, например, при постановке диагноза по снимку, а также для защиты авторских прав, в промышленности и т.д.

Методы поиска по содержанию основаны на представлении изображения в виде векторов признаков или дескрипторов. Вектором признаков (дескриптором) называется набор численных параметров, описывающих характеристики изображения, например, такие как цвет, текстуру и т.д. Вектора признаков принимают значения в пространстве признаков. Если на таком пространстве задать меру, то можно сравнивать изображения друг с другом, вычисляя расстояние между соответствующими векторами признаков.

Можно выделить два основных подхода к поиску изображений по содержанию:

- поиск по глобальным дескрипторам;
- поиск по локальным дескрипторам.

Под глобальными дескрипторами подразумеваются вектора признаков, полученные при анализе всего изображения в целом. Обычно в таких методах каждая точка изображения вносит вклад в значение дескриптора. Поиск по глобальным дескрипторам применяется при решении задачи поиска по общему подобию, которую можно сформулировать как нахождение по запросу-образцу визуально и семантически похожих изображений с точки зрения человека. Например, поиск по изображению заката других изображений заката. В данном случае важно сходство изображений в целом, а не их фрагментов. На сегодняшний день данные алгоритмы характеризуются невысоким уровнем качества поиска.

Локальные дескрипторы - это вектора признаков, которые описывают не все изображение в целом, а какие-то его значимые части. Методы поиска по локальным дескрипторам применяются в основном для решения задач поиска нечетких дубликатов и поиска заданного фрагмента на изображениях коллекции. Дубликатами считаются изображения одной и той же сцены или объекта, сделанные в разных условиях или разного качества. В частности, дубликатами считаются изображения одной и той же сцены в разном масштабе или снятые с разных точек, с различием в освещении или с незначительными изменениями фона. При решении данной задачи важно обнаружить сходство отдельных частей изображений, поэтому для данных целей обычно применяются локальные дескрипторы, описывающие особенности областей изображений. На сегодняшний день методы поиска по локальным дескрипторам дают более точные результаты по сравнению с методами поиска по глобальным дескрипторам.

Можно выделить работу М. Irani & Е. Shetchman [1], в которой предложенные локальные дескрипторы потенциально подходят для решения задачи поиска по общему подобию. Авторы работы [1] проводят успешные эксперименты, показывающие эффективность применения данных дескрипторов для решения задач распознавания объектов и поиска фрагментов в коллекциях изображений и видео без предварительного обучения. Алгоритм М. Irani & Е. Shetchman [1] основан на наблюдении, что небольшие фрагменты изображения имеют свойство повторяться на нем в том же или другом масштабе (исследования данного наблюдения описаны в работе [2]). Информация об описанной повторяемости (самоподобии) в пределах некоторой области формирует так называемую геометрическую разметку области изображения. С помощью данной

геометрической разметки формируются самоподобные локальные дескрипторы, которые в дальнейшем используются для сравнения изображений.

Задачами данной дипломной работы являются:

- 1) проверка применимости данного подхода к решению задачи поиска по общему подобию, в частности,
  - реализация подсчета дескрипторов описанным методом;
  - адаптация описанного подхода к поиску изображений по общему подобию;
- 2) сравнение результатов поиска по признаку самоподобия локальных дескрипторов с результатами методов поиска изображений по глобальным дескрипторам.

## **2. Постановка задачи**

В рамках данной дипломной работы были поставлены следующие задачи:

1. Реализация алгоритма вычисления признаков самоподобия локальных дескрипторов
2. Адаптация существующего метода поиска фрагментов изображений к задаче поиска изображений по запросу-образцу
3. Экспериментальное сравнение данного метода с поиском изображений по глобальным дескрипторам – наиболее распространенным подходом к решению задачи поиска изображений по содержанию
4. Анализ результатов

### 3. Обзор предметной области и существующих подходов

Все алгоритмы поиска изображений по содержанию можно разделить на методы, основанные на поиске по глобальным и по локальным дескрипторам.

#### Глобальные дескрипторы

Глобальные дескрипторы описывают изображение в целом и представляют его в виде векторов признаков. Обычно каждая точка вносит вклад в значение дескриптора.

При поиске по коллекции цветных изображений для человека одной из наиболее значимых характеристик является цвет. К тому же он инвариантен относительно расположения объектов и размера изображения, что упрощает его анализ. Самое распространенное представление цвета - это **цветовая гистограмма** [3] (гистограмма распределения цветов). Подход заключается в том, что цветовое пространство разбивается на промежутки и для каждого промежутка вычисляется доля пикселей из данного промежутка. Процесс разбиения цветового пространства на ограниченное количество цветовых диапазонов называется квантованием. Сложность данного подхода заключается в построении такого разбиения, чтобы цвета из одного диапазона были плохо различимы человеком, а цвета из разных диапазонов, наоборот, различались. Пример гистограммы с квантованием в цветовом пространстве RGB изображен на Рисунке 1.



Рисунок 1. Цветовая гистограмма в пространстве RGB

В простейшем случае, и в частности в данном примере, не учитывается пространственное расположение цветов на изображении. Решением является разбиение изображения на фиксированные блоки и вычисление цветовой гистограммы отдельно для каждого блока. Однако в данном решении возникает проблема в подборе размеров блоков. Еще одним решением проблемы учета пространственного расположения цветов является модификация классической цветовой гистограммы HistSP [4]. Главная идея данного метода заключается в том, что для каждого ненулевого элемента гистограммы



вычисляется центр масс пикселей соответствующего цвета, его координаты сохраняются в числе элементов вектора признаков. Пример такой гистограммы изображен на Рисунке 2.

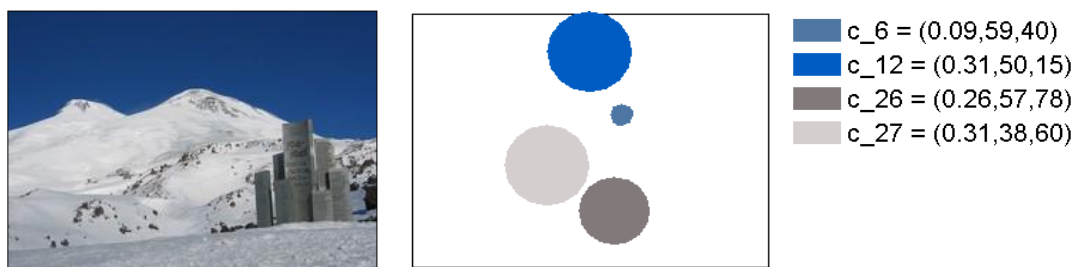


Рисунок 2. Цветовая гистограмма HistSP

Альтернативная модель представления цвета изображения – это **цветовые моменты**, предложенные Stricker M., Orengo M. в [5,6]. Авторы рассматривают распределения отдельных цветовых каналов как части трехмерного распределения. Вводится пять фиксированных областей: центральная область в виде эллипса и четыре боковые области (Рисунок 2). Для каждой области вычисляется математическое ожидание по каждому из цветовых каналов и попарные ковариации распределений каналов. Для каждого из пикселей вычисляется его степень принадлежности к области: чем ближе к границе области, тем меньше степень принадлежности к ней. Это значение регулирует вклад цвета соответствующего пикселя в общую оценку распределения цвета области.

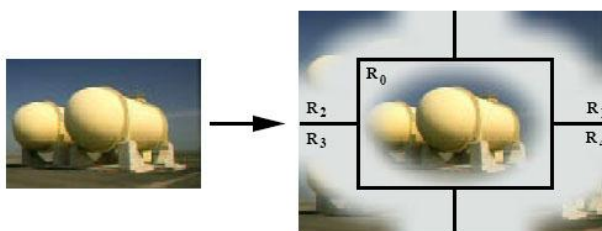


Рисунок 2. «Нечеткие области» изображения

Еще одной значимой характеристикой изображения является текстура. Она описывает структуру объектов на изображении, и определяется по распределению уровня яркости изображения.

Одним из наиболее известных представлений текстурных характеристик изображения являются матрицы смежности (матрицы совместного распределения яркости на изображении - Grey Level Co-occurrence Matrices, GLCM) [7]. Матрица смежности зависит от разницы яркостей в соседних пикселях. По ней можно вычислить значения

различных статистических показателей, например, таких как энтропия (степень неоднородности), контраст, показатель однородности, показатель гладкости и др.

Еще одними из известных текстурных признаков являются признаки Тамуры [8], которые были выделены с учетом особенностей зрительного восприятия человека. Они включают в себя зернистость, контрастность, направленность, линейность, регулярность и грубость. Из них были определены три наиболее подходящих при решении задачи поиска изображений – это зернистость, контрастность и направленность. Если собрать эти значения в одно изображение, в котором red, green, blue каналы будут заменены на зернистость, контрастность и направленность соответственно, то из полученного изображения можно вычислить 3d текстурную гистограмму. Если будем разбивать на промежутки зернистость, контрастность и направленность, то получим 3d гистограмму, аналогичную цветовой 3d гистограмме на Рисунке 1, только с другими осями координат.

Для рассмотрения текстуры изображения в различных масштабах, можно использовать вейвлет-анализ, который заключается в разложении сигнала по базисным функциям. Базисные функции (вейвлеты) строятся на основе порождающего вейвлета с помощью сдвига и масштабирования. Берется исходное изображение и строится первая проекция сигнала (свертка с первой базисной функцией), потом вычисляется разность исходного сигнала с полученным и строится вторая проекция этой разности (свертка со второй базисной функцией), и т.д. Причем, каждая базисная функция является сдвигом предыдущей, растянутой в  $2^n$  раз ( $n$  характеризует масштаб). Таким образом, в итоге получаем грубую версию изображения. Такие базисные функции обычно называют фильтрами. Одними из эффективных и используемых фильтров являются фильтры Габора [9], ICA-фильтры (Independent Component Analysis, ICA [10]). ICA-фильтры получены путем анализа обучающего множества изображений. Данные фильтры являются локальными и подобны фильтрам Габора, однако в отличие от них носят естественный характер и отражают основные направления текстуры изображений, по которым они строились. Также проводились исследования, показывающие, что способ построения ICA-фильтров схож с процессом зрительного обучения человека.

Еще одним важным признаком для сравнения изображений является форма объектов. Простейшими признаками являются центр тяжести фигур, площадь, направление главной оси и т.д. Существуют и более сложные методы, представляющие фигуры более детально, их можно разделить на два класса:

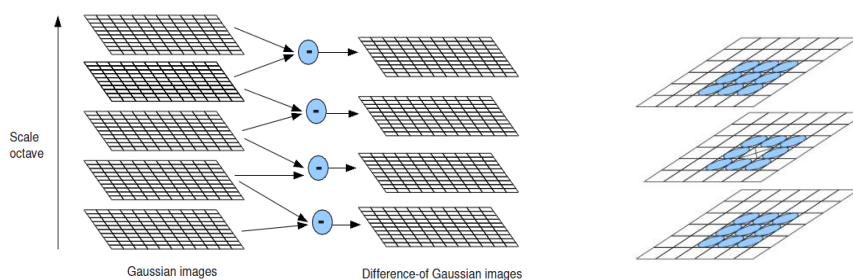
- внешнее представление, основанное на информации о контуре фигуры – дескрипторы границ, к ним относятся разные виды сигнатур, дескрипторы Фурье, вейвлет-дескрипторы, цепные коды и т.д.;
- внутреннее представление, основанное на информации о фигуре в целом – дескрипторы областей, к ним относятся, например, инварианты моментов и т.д.

Более подробно цветовые признаки, текстурные признаки и признаки формы описаны в [4].

### **Локальные дескрипторы**

Локальные дескрипторы представляют собой вектора признаков, построенные по отдельным фрагментам изображения. То есть они не описывают все изображение, а содержат информацию только о выбранных некоторым способом фрагментах. Самыми известными локальными дескрипторами являются SIFT [11] (Scale Invariant Feature Transform), SURF [12] (Speeded Up Robust Features), PCA-SIFT [13] (PCA – Principal Component Analysis) и т.д.

Метод SIFT [11] (Scale Invariant Feature Transform) обнаруживает и описывает локальные особенности изображения. Получаемые с помощью него признаки инвариантны относительно масштаба и поворота, устойчивы к ряду аффинных преобразований, шуму, изменению в освещении. Данный алгоритм можно разделить на две части: определение «точек интереса» (key points, points of interest, salient points) и построение дескрипторов окрестностей данных точек. Существует несколько способов определения точек интереса. Алгоритм, предложенный в рамках SIFT, один из самых известных. Он заключается в использовании пирамиды Гаусса [14], которая строится для изображения. Далее изображения приводятся к одному размеру, и вычисляется их разность (DoG, difference-of-Gaussian images), как показано на Рисунке 3. Причем в качестве кандидатов точек интереса выбираются только те пиксели, которые сильно отличаются от остальных, это делается, например, путем сравнения каждого пикселя изображения с несколькими соседними данного масштаба, с несколькими соответствующими соседями в большем и меньшем масштабе, этот процесс изображен на Рисунке 3. Пиксель выбирается как точка интереса только в том случае, если его яркость является экстремумом.



*Рисунок 3. Процесс нахождения точек интереса*

Далее для каждой такой точки интереса вычисляется локальный дескриптор, характеризующий направление градиентов в пикселях некоторой окрестности. Главным минусом SIFT дескрипторов является их высокая размерность и большое количество на изображении. PCA-SIFT [13] (PCA, Principal Component Analysis – анализ главных компонент) дескриптор – одна из вариаций SIFT, в которой уменьшается размерность дескриптора с помощью анализа главных компонент. Это достигается с помощью нахождения пространства собственных векторов, на которое впоследствии проецируются вектора признаков.

Альтернативным подходом является SURF [12] (Speeded Up Robust Features), который в несколько раз быстрее SIFT. В данном подходе для ускорения поиска точек интереса используются интегральные изображения [15]. Значение в каждой точке интегрального изображения вычисляется как сумма значения в данной точке и значений всех точек, которые находятся выше и левее данной. С помощью интегральных изображений за константное время вычисляются так называемые прямоугольные фильтры [16], которые состоят из нескольких прямоугольных областей. SURF в несколько раз быстрее SIFT, менее чувствителен к шуму, к повороту, но чувствителен к изменению освещения или угла, под которым был сделан снимок.

Глобальные и локальные дескрипторы обычно используются для решения разных задач. Глобальные дескрипторы в основном применяются для решения задачи общего поиска изображений по содержанию, то есть для поиска по запросу-образцу визуально и семантически похожих изображений. В данном случае важно все изображение в целом, а не отдельные его области, поэтому для решения данной задачи подходят глобальные дескрипторы, характеризующие все изображение. Локальные дескрипторы обычно применяются для решения задачи поиска нечетких дубликатов. Дубликатами считаются

изображения одной и той же сцены или объекта, сделанные в разных условиях или разного качества, в частности, изображения одной и той же сцены в разном масштабе или снятые с разных точек, с различием в освещении или с незначительными изменениями фона. При решении данной задачи важно обнаружить сходство отдельных частей изображений, для данных целей обычно применяются локальные дескрипторы, описывающие особенности областей изображений.

Все вышеописанные локальные дескрипторы используют общую парадигму, заключающуюся в нахождении точек интереса и построении для каждой из них дескриптора, описывающего ее окрестность. Принципиально другой подход описан в работе [1]. Он основан на свойстве повторяемости (самоподобии) фрагментов на изображении, то есть на наблюдении, что небольшие фрагменты изображения имеют свойство повторяться на нем в том же или другом масштабе. Информация о такой повторяемости в пределах некоторой области изображения формирует так называемую геометрическую разметку. С помощью данной геометрической разметки формируются самоподобные локальные дескрипторы. Причем, даже если изображения имеют разную текстуру, цвет и др., но их геометрические разметки похожи, то дескрипторы тоже будут похожи. Эксперименты, проведенные авторами работы, показали применимость данных дескрипторов для решения задач распознавания объектов и поиска фрагментов в коллекциях изображений и видео без предварительного обучения. Также стоит заметить, что большинство других локальных дескрипторов строятся только для точек интереса и, следовательно, не описывают все изображение в целом. Самоподобные локальные дескрипторы изначально строятся для каждой точки изображения, после чего производится фильтрация, даже после которой самоподобные дескрипторы образуют более плотное множество по сравнению с другими описанными локальными дескрипторами.

## 4. Поиск изображений по признаку самоподобия локальных дескрипторов

### 4.1. Общее описание подхода

Существует множество методов представления изображения в виде векторов признаков, часть которых описана в разделе 3. Но такие методы, например, неприменимы для случая на Рисунке 5. Изображения на Рисунке 5 не обладают общими свойствами, такими как цвет, текстура, границы и т.д., но человек ясно видит, что они представляют один и тот же объект - сердце. Похожими с точки зрения машины их делает тот факт, что на каждом из них локальные фрагменты (под фрагментом подразумевается небольшая часть изображения, например, квадрат 5x5 пикселей) в определенной степени похожи на фрагменты в некоторой окрестности этого фрагмента. Эта информация о степени схожести (самоподобия) формирует геометрическую разметку области изображения. Другими словами, данные изображения похожи именно этими геометрическими разметками, в то время как соответствующие фрагменты на отдельных изображениях могут быть не похожи между собой.

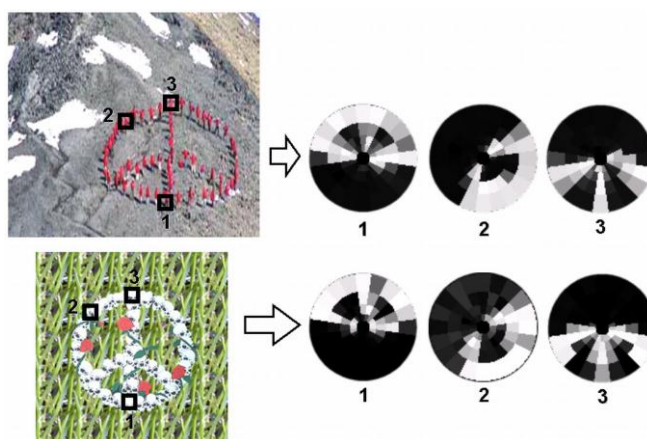


*Рисунок 5. Данные изображения одного и того же объекта (сердца) не обладают общими свойствами (цвет, текстура, границы), но обладают схожими геометрическими разметками.*

Подход, основанный на этом наблюдении, описан в статье M. Irani & E. Shtechman [1], авторы которой проводят успешные эксперименты, показывающие эффективность

применения данных дескрипторов для решения задач распознавания объектов и поиска фрагментов в коллекциях изображений и видео без предварительного обучения.

Для каждого изображения в качестве дескрипторов берется множество таких геометрических разметок, задаваемых фрагментами этого изображения. Два изображения считаются похожими, если дескрипторы одного похожи по значению и относительному расположению на дескрипторы другого. Сравнимые изображения могут быть разного размера, это не имеет значения, так как для изображения рассматриваются геометрические разметки, задаваемые фрагментами разного масштаба. Другими словами, если есть два изображения, на которых присутствует один объект разного размера, то они будут считаться похожими. Заметим, что соответствующие друг другу точки на изображениях могут визуально сильно отличаться, как показано на Рисунке 6, но их дескрипторы будут похожи. Процесс вычисления дескрипторов рассматривается в разделе 4.1.2.



*Рисунок 6. Соответствующие локальные самоподобные дескрипторы. Отмечены соответствующие точки (1,2,3) на изображениях и их дескрипторы. (Рисунок взят из работы [1])*

## 4.2. Вычисление самоподобных локальных дескрипторов изображения

Самоподобный локальный дескриптор содержит в себе информацию о цвете области, о границах объектов из данной области, о повторяющихся шаблонах и текстуре. Более точно, дескриптор описывает самоподобие цвета области, границ, текстуры, повторяющихся шаблонов, которые формируют геометрическую разметку области. Причем, дескриптор содержит информацию именно о самоподобии: например, близкими могут быть дескрипторы гомогенной области и области с четко выраженной текстурой, если их геометрические разметки похожи. Самоподобный локальный дескриптор  $\mathbf{d}_q$  считается для каждого пикселя изображения  $\mathbf{q}$ . На Рисунке 7 изображен процесс его вычисления.

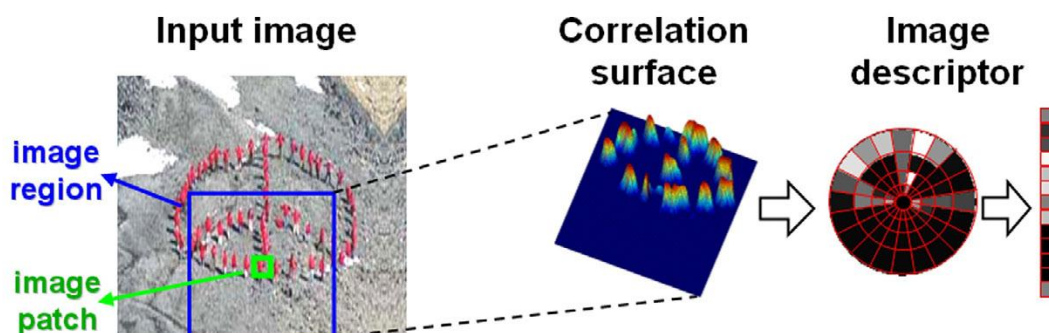


Рисунок 7. Вычисление самоподобного локального дескриптора. (Рисунок взят из работы [1])

Путем сравнения фрагмента (небольшого квадрата размера  $2*\text{patchRadius}$  x  $2*\text{patchRadius}$ ) с центром в  $\mathbf{q}$  с большим регионом (квадрат размера  $2*\text{regionRadius}$  x  $2*\text{regionRadius}$ ) с центром в  $\mathbf{q}$  получаем локальную корреляционную поверхность (local correlation surface), которая изображена на Рисунке 7. Далее локальная корреляционная поверхность преобразуется в круг с секторами (binned log-polar). Таким образом, получаем дескриптор, который хранится в виде вектора. Заметим, что для дескриптора такого вида характерно увеличение неясности положения с удалением от центра, что допускает небольшие отклонения (отклонения в масштабе, в ориентации и сдвиг).

Рассмотрим процесс вычисления дескриптора подробнее. Сравниваем фрагмент с центром в  $\mathbf{q}$  с фрагментами, которые находятся в некотором регионе с центром в  $\mathbf{q}$ , для



сравнения используем сумму квадратов расстояний (SSD, Sum of Square Differences) между цветами фрагментов (в качестве цветового пространства берется CIELab [17]). Нормализуем полученную поверхность расстояний  $SSD_q(x,y)$  и преобразуем ее в локальную корреляционную поверхность  $S_q(x,y)$ :

$$S_q(x, y) = e^{-\frac{SSD_q(x,y)}{\max(var_{noise}, var_{auto})}}$$

где  $var_{noise}$  – константа, которая указывает допустимые вариации (в цвете, освещении или допустимый шум);  $var_{auto}(q)$  – зависит от контраста фрагмента и его структуры, это максимальная разница между фрагментом с центром в  $q$  и фрагментами в центрах  $q_1, \dots, q_8$ , где  $q_1, \dots, q_8$  соседи пикселя  $q$ .

Далее локальная корреляционная поверхность переводится в полярные логарифмические координаты с центром в  $q$  и делится на 80 ячеек: 20 углов и 4 радиальных интервала. В каждой ячейке выбирается максимальное корреляционное значение. Полученные значения в ячейках, записанные в виде вектора, формируют локальный самоподобный дескриптор. Нормализуем вектор таким образом, чтобы значения всех его элементов лежали в промежутке [0,1].

На Рисунке 6 изображены самоподобные локальные дескрипторы, вычисленные для нескольких соответствующих друг другу точек (1,2,3) на двух различных изображениях одного и того же объекта. Несмотря на различные фотометрические свойства изображений, дескрипторы соответствующих точек похожи.

Самоподобный локальный дескриптор обладает следующими свойствами:

- Дескриптор вычисляется для некоторой области изображения, а не для всего изображения, то есть не является глобальным.
- Так как в каждой ячейке выбирается максимальное корреляционное значение, то дескриптор становится нечувствительным к точной позиции наиболее похожего фрагмента внутри ячейки. Также благодаря увеличению размера ячеек с увеличением радиуса допускаются негрубые различия.
- Вычисляются дескрипторы для разных масштабов, то есть для каждого изображения строится пирамида Гаусса [15] и для каждого уровня пирамиды вычисляются дескрипторы.
- Цвет, границы, сложные текстуры обрабатываются единообразным способом.

Не все дескрипторы считаются информативными, поэтому производится фильтрация. Фильтруем дескрипторы, которые не содержат полезной информации о самоподобии, то есть в которых центральный фрагмент не похож на остальные фрагменты в рассматриваемом регионе. То есть неинформативным считается дескриптор, все элементы которого ниже определенного порога.

Также фильтруем дескрипторы, в которых центральный фрагмент похож на все фрагменты из рассматриваемого региона. Эти области соответствуют гомогенным областям изображения (например, областям похожего цвета или с похожей текстурой). Данная фильтрация производится с помощью меры разреженности (sparseness measure [18]):

$$sparseness(x) = \frac{\sqrt{n} - \frac{\sum_i |x_i|}{\sqrt{\sum_i x_i^2}}}{\sqrt{n} - 1}$$

Чем больше элементы похожи между собой, тем ближе значение функции **sparseness** к нулю. Таким образом, для фильтрации гомогенных областей достаточно ограничить некоторым порогом **sparseness<sub>threshold</sub>** значение функции **sparseness**. Другими словами все дескрипторы, для которых **sparseness < sparseness<sub>threshold</sub>** считаются неинформативными и отфильтровываются.

## 4.2. Адаптация метода поиска по признаку самоподобия локальных дескрипторов

Необходимость адаптации существующего метода поиска фрагмента на изображениях, описанного в [1], возникла из-за того, что в исходном методе в качестве запроса выступала одна группа дескрипторов, которую пытались обнаружить на изображениях рассматриваемой коллекции, в то время как для метода поиска изображений по содержанию запрос представляется в виде набора групп дескрипторов.

Исходный алгоритм поиска фрагмента на изображении заключается в следующем. Пусть  $F$  – фрагмент-запрос, которой пытаемся найти в некотором изображении  $G$ . Для фрагмента-запроса  $F$  вычисляются дескрипторы всех уровней пирамиды Гаусса [14], они формируют группу дескрипторов  $d^1, d^2, \dots, d^n$  с координатами  $l^1, l^2, \dots, l^n$  и центром группы  $c$ . Далее с помощью алгоритма поиска похожей группы дескрипторов на изображении, описанного в [19], в каждой точке изображения  $G$  пытаемся найти группу дескрипторов, похожую на  $d^1, d^2, \dots, d^n$  по значениям и относительному расположению. В результате получается вероятностная карта (Рисунок 8) такого же размера как изображение  $G$ , которая соответствует вероятности нахождения фрагмента  $F$  в каждой точке изображения  $G$ .

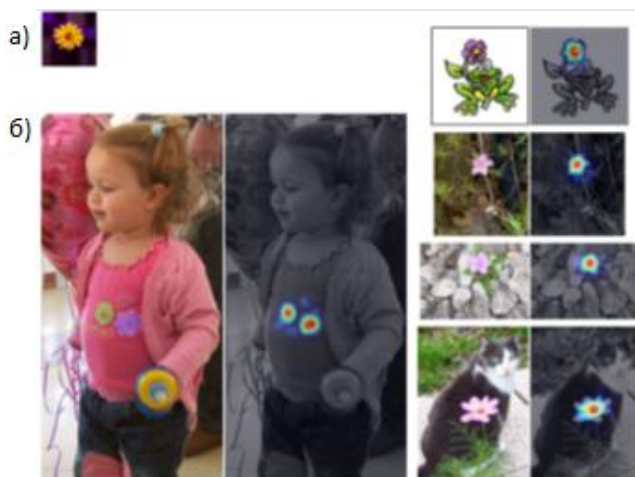


Рисунок 8. а) фрагмент-запрос (шаблон); б) изображения, на которых пытались найти фрагмент-запрос, и соответствующие им вероятностные карты. (Рисунок взят из работы [1])

Описанный исходный алгоритм метода поиска фрагментов изображений был адаптирован к задаче поиска изображений по запросу-образцу. Его можно представить следующим псевдокодом.

```
public double getSimilarity (queryImg, img){  
    /* сумма показателей подобия групп */  
    double groupSimilaritySum = 0.0;  
    /* список центров групп изображения-запроса */  
    List<Point> centerOfGroupList = getCenterOfGroupList(groupStep, queryImg);  
    /* список информативных групп-запросов */  
    List<List<Descriptor>> queryGroupList =  
        getQueryGroupList(queryImg, centerOfGroupList, groupRadius);  
    for (List<Descriptor> queryGroup : queryGroupList){  
        groupSimilaritySum += getSimilarityOfGroups(img, queryGroup);  
    }  
    return groupSimilaritySum / queryGroupList.length;  
}  
  
private double getSimilarityOfGroups(Image img, List<Descriptor> queryGroup){  
    /* список из k дескрипторов наиболее похожих на случайно выбранный дескриптор  
    из queryGroup */  
    List<Descriptor> similarToFirstDscrsList = getSimilarToFirstDscr(queryGroup, img);  
    /* формируем k групп-кандидатов */  
    List<SimilarGroup> similarGroupCandidateList =  
        formSimilarGroupCandidateList(similarToFirstDscrsList);  
    /* заполняем группы-кандидатов остальными похожими дескрипторами */  
    findOtherSimilarDscrs(similarGroupCandidateList, queryGroup, img);  
    SimilarGroup mostSimilarGroup = getMostSimilarGroup(similarGroupCandidateList);  
    return mostSimilarGroup.getSimilarity();  
}
```

Пусть есть изображение-запрос, вычисляем у него дескрипторы нулевого уровня пирамиды Гаусса [14]. Перебираем точки изображения-запроса с шагом **groupStep** пикселей по ширине и по высоте, в каждой точке рассматриваем квадрат размера **2\*groupRadius** x **2\*groupRadius** с центром в данной точке **c**. Все дескрипторы изображения-запроса, центры которых попали в данную область, формируют группу-запрос дескрипторов **d<sup>1</sup>, d<sup>2</sup>,.. d<sup>n</sup>** с координатами **l<sup>1</sup>, l<sup>2</sup>,.. l<sup>n</sup>** и центром группы **c**. Рассматриваются только информативные группы, то есть группы, содержащие не менее трех дескрипторов. Пример такой группы-запроса изображен на Рисунке 9.

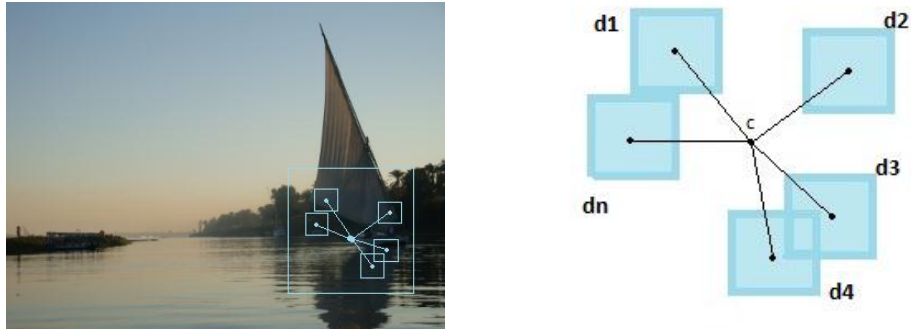


Рисунок 9. Группа дескрипторов **d<sup>1</sup>, d<sup>2</sup>,.. d<sup>n</sup>** с центром в точке **c**

Берем первый дескриптор из группы-запроса **d<sup>1</sup>**. Считаем (считываем из базы) дескрипторы всех уровней пирамиды Гаусса для изображения, с которым сравниваем запрос, среди них ищем похожие на **d<sup>1</sup>**. Мера подобия по значению дескрипторов считается по следующей формуле, взятой из работы [19]:

$$P(d_y^i, d_x^i) = e^{-(d_y^i - d_x^i)^T * S_D^{-1} * (d_y^i - d_x^i)}$$

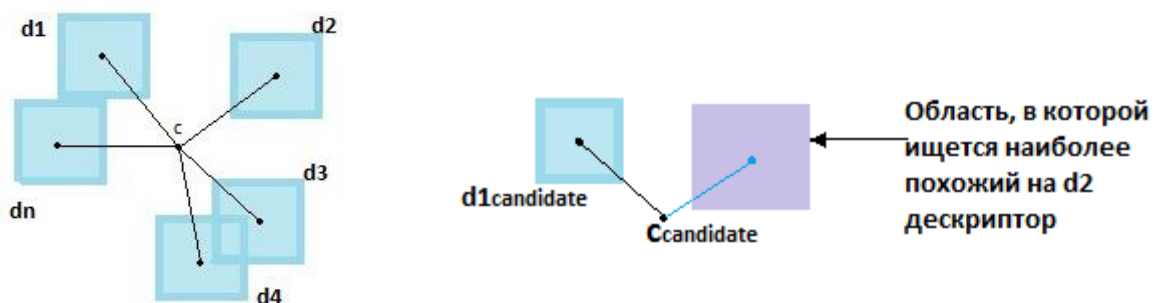
где **S<sub>D</sub>** – константная матрица ковариаций (диагональная, квадратная **RxR**, где **R** – это размерность вектора дескриптора), которая определяет допустимое отклонение значений дескрипторов. Если обозначить диагональные элементы **a<sub>1</sub>,..,a<sub>R</sub>**, то чем ближе диагональный элемент **a<sub>i</sub>** к нулю, тем важнее, чтобы **i**-ые элементы дескрипторов совпадали.

Выбираем первые **k** наиболее похожих по значению на **d<sup>1</sup>** дескрипторов, тем самым формируем **k** похожих групп-кандидатов, содержащих на данном шаге по одному дескриптору. Каждый из данных дескрипторов задает масштаб для всей группы-

кандидата, то есть далее для каждой группы-кандидата ищем похожие дескрипторы только такого же уровня пирамиды Гаусса, как и первый дескриптор, задающий группу.

Зная координаты центра группы на изображении-запросе, координаты дескриптора  $\mathbf{d}^1$  и координаты похожих на  $\mathbf{d}^1$  дескрипторов  $\mathbf{d}^1_{\text{candidate}_1}, \dots, \mathbf{d}^1_{\text{candidate}_20}$  можно предположить, где находятся центры групп-кандидатов  $\mathbf{C}_{\text{candidate}_1}, \dots, \mathbf{C}_{\text{candidate}_20}$ .

Для того чтобы найти похожие дескрипторы для следующего дескриптора  $\mathbf{d}^p$  из группы  $\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^n$  изображения-запроса, область поиска ограничивается следующим образом: зная как располагается дескриптор  $\mathbf{d}^p$  относительно центра  $\mathbf{c}$  группы-запроса, предполагаем, что дескриптор  $\mathbf{d}^p_{\text{candidate}}$ , похожий на  $\mathbf{d}^p$ , располагается относительно  $\mathbf{C}_{\text{candidate}}$  похожим образом. То есть ищем похожие на  $\mathbf{d}^p$  дескрипторы только среди тех, которые располагаются внутри области радиуса **locationDeviation** пикселей, центр которой соответствует ожидаемому центру  $\mathbf{d}^p_{\text{candidate}}$ . Выбираем наиболее похожий по значению дескриптор и пересчитываем центр группы-кандидата (среднее между центрами  $\mathbf{d}^1_{\text{candidate}}, \dots, \mathbf{d}^k_{\text{candidate}}$ ) и показатель подобия группы (среднее между подобиями всех дескрипторов  $\mathbf{d}^1_{\text{candidate}}, \dots, \mathbf{d}^k_{\text{candidate}}$ ).



*Рисунок 10. Процесс нахождения дескриптора похожего по значению и относительно расположению.*

Аналогично ищем дескрипторы, похожие на следующий дескриптор из группы-запроса, и т.д. В результате получаем  $\mathbf{k}$  похожих групп-кандидатов, выбираем группу с наибольшим подобием.

Продельваем описанные действия со всеми группами изображения-запроса. В качестве показателя подобия изображений берем сумму показателей подобия групп, деленную на количество информативных групп на изображении-запросе.

## 5. Экспериментальное сравнение методов

### 5.1. Описание методов поиска, участвовавших в экспериментальном сравнении

В данной дипломной работе производилось сравнение метода поиска по признаку самоподобия локальных дескрипторов с методами поиска по глобальным дескрипторам, в частности с методом цветowych моментов ColorMoments [5,6], с модифицированным методом гистограмм HistSP [4], с методом поиска по текстуре ICA, использующим ICA фильтры [14]. Данные методы были описаны в разделе 3, они были выбраны для сравнения, так как по результатам экспериментов, описанных в [4] показали лучший результат.

В методе HistSP [4] использовалось цветовое пространство HSI\* [4], разбитое на цветовые диапазоны с применением схемы равномерного квантования с граничными условиями [4] с параметрами <12, 6, 8>: не считая граничные диапазоны, ось оттенков (H) была разбита на 12 равных диапазонов, ось насыщенности (S) на 6, ось интенсивности (I) на 8. Для оценки подобия двух изображений используется функция  $D_{\text{HistSP}}$  [4].

В методе цветowych моментов [5,6] использовалось цветовое пространство CIELab [17]. Для сравнения изображений использовалась взвешенная манхэттенская метрика с различными весами для различных областей и различных параметров распределения цветов.

В адаптированном методе поиска по признаку самоподобия локальных дескрипторов использовалось цветовое пространство CIELab [17] и следующие значения параметров: **patchRadius** = 5, **regionRadius** = 40, **var<sub>noise</sub>** = 210, **groupRadius** = 25, **groupStep** = 20, **k** = 20, **locationDeviation** = 5.

Метод поиска изображений по признаку самоподобия локальных дескрипторов был реализован на языке java, в среде разработки netbeans, с использованием СУБД Microsoft SQL Server.

## 5.2. Описание тестовой коллекции

Для оценки адаптированного метода поиска по признаку самоподобных локальных дескрипторов был проведен ряд экспериментов, сравнивающих его с методами поиска по глобальным дескрипторам.

Эксперимент проводился на базе из 1000 изображений, построенной на кафедре системного программирования математико-механического факультета следующим образом: 100 изображений отобраны и разбиты вручную на 16 групп так, что изображения из одной группы очень похожи друг на друга, после этого в базу было добавлено 900 произвольных изображений, которые никак не размечались.

В качестве запросов использовались только размеченные изображения. Изображение считалось релевантным ответом на запрос, если оно принадлежало той же группе, что и изображение запрос. Все остальные изображения, включая 900 неразмеченных изображений, считались нерелевантными ответами.



### 5.3. Используемые метрики оценки

Для оценки качества работы медов использовались следующие оценки:

$$\text{Точность} = \frac{\text{количество найденных релевантных изображений}}{\text{количество найденных изображений}};$$

$$\text{Полнота} = \frac{\text{количество найденных релевантных изображений}}{\text{количество релевантных изображений}}.$$

Точность вычисляется как отношение найденных релевантных изображений к общему числу найденных изображений. Точность характеризует способность системы выдавать в списке только релевантные изображения. Точность на уровне N означает, что рассматриваются только первые N изображений из упорядоченного списка похожих изображений.

Полнота вычисляется как отношение найденных релевантных изображений к общему количеству релевантных изображений. Полнота характеризует способность системы находить нужные изображения и не учитывает количество выдаваемых нерелевантных изображений. Полнота на уровне N означает, что рассматриваются только первые N изображений из упорядоченного списка похожих изображений.

Для каждой пары изображений (image\_1, image\_2), где image\_1 – изображение-запрос из 100 размеченных изображений, image\_2 – одно из 1000 изображений из базы, вычислялась мера подобия. Далее для каждого изображения-запроса выбиралось N наиболее похожих изображений, и считались метрики оценки.

## 5.4. Анализ результатов экспериментов

Результаты экспериментов по сравнению методов поиска по глобальным дескрипторам с методом поиска по признаку самоподобия локальных дескрипторов представлены в виде графиков на Рисунках 11, 12.

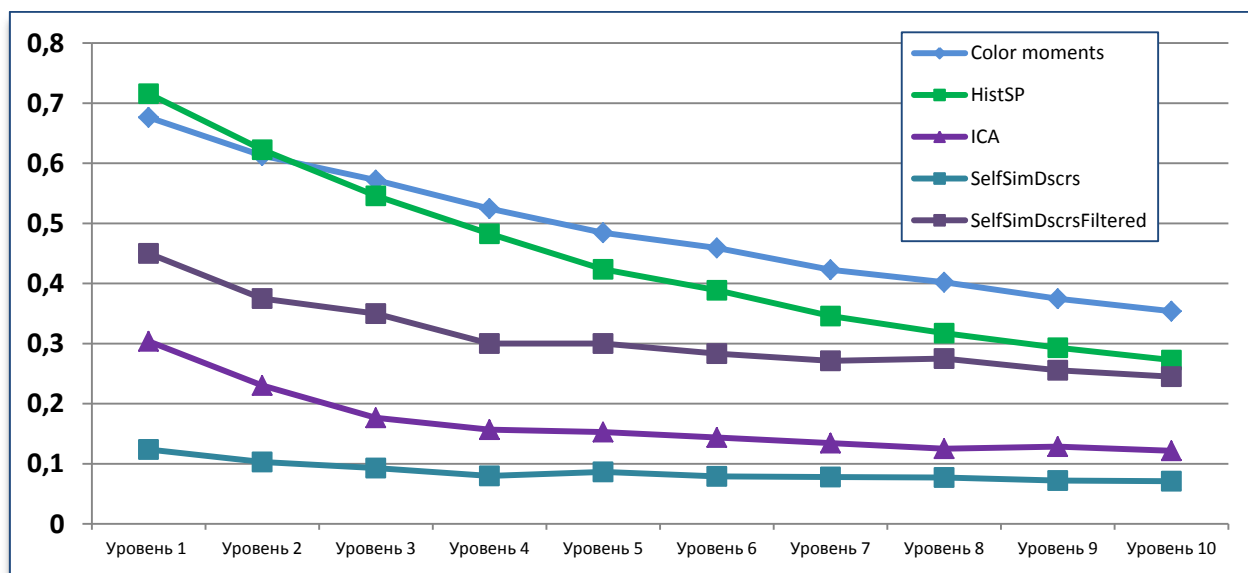


Рисунок 11. Точность методов поиска на уровнях от 1 до 10

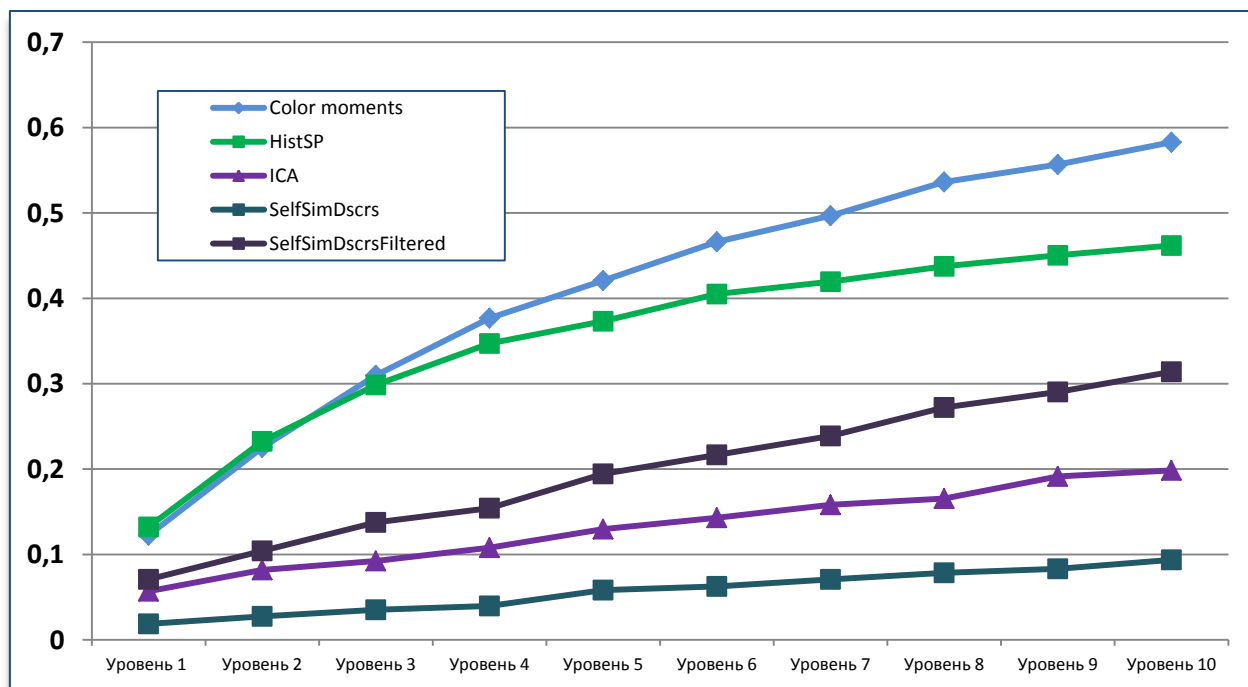


Рисунок 12. Полнота методов поиска на уровнях от 1 до 10

На Рисунках 11, 12 Color moments, HistSP, ICA – это методы поиска по глобальным дескрипторам. SelfSimDscrs – это адаптированный метод поиска по признаку самоподобия локальных дескрипторов.

При анализе результатов было замечено, что чем больше количество дескрипторов, тем выше качество результатов, и что большое количество запросов описывается малым количеством дескрипторов. Точность и полнота для тех пар, в которых количество дескрипторов более 600 была приемлема, однако среди 1000 изображений в базе оказалось всего 79 изображений с количеством дескрипторов более 600, скорее всего, именно поэтому качество поиска оказалось неудовлетворительным. Поэтому в качестве изображений-запросов были отобраны изображения, которые имеют больше 600 дескрипторов. Таких изображений-запросов оказалось 20 из 100. Результаты работы метода после отбора представлены на Рисунках 11, 12 под названием SelfSimDscrsFiltered. Точность и полнота увеличились примерно в три раза, из чего можно сделать предположение, что данный метод является конкурентоспособным при условии, что изображения имеют достаточное количество дескрипторов.

На Рисунке 13 приведен пример, когда на изображение-запрос «поле» (группа fields), выдается список похожих изображений (порядок справа налево по уменьшению схожести). Причем, для всех этих изображений количество дескрипторов больше 600.



Рисунок 13. Пример «хорошего» ответа на запрос

Можно предположить, что для изображений из группы «поля» ответы наиболее точные и полные, так как, во-первых, изображения из данной группы имеют большое количество дескрипторов (более 600), во-вторых, это достаточно характерные дескрипторы, которые в большинстве своем располагаются вблизи границы неба и земли, как показано на Рисунке 13 а. При этом дескрипторы соответствующих фрагментов на изображениях похожи.

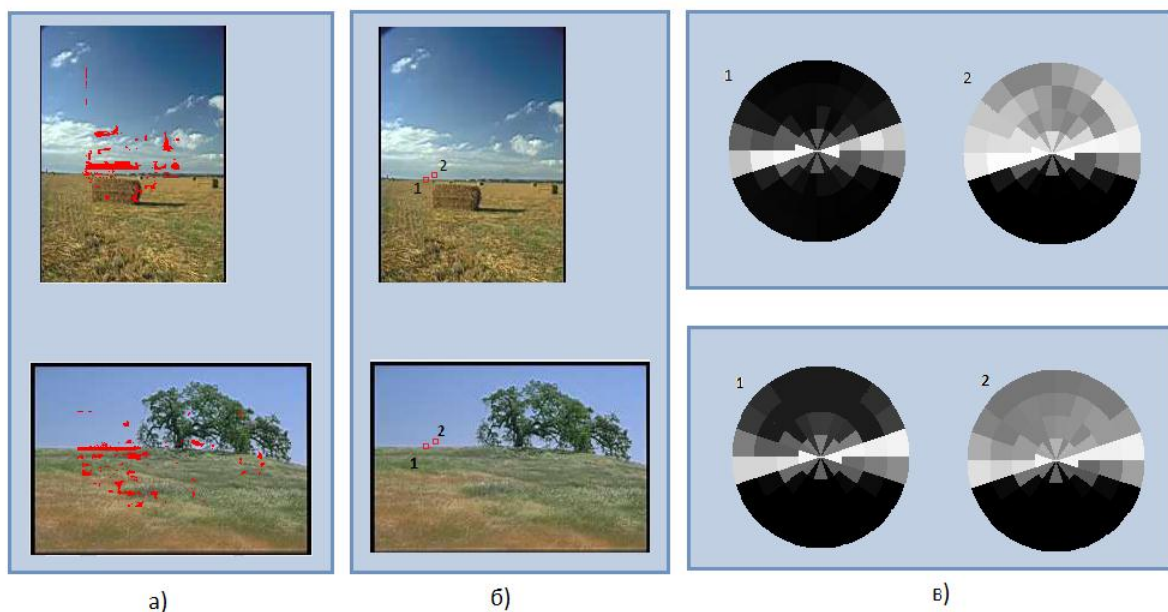


Рисунок 14. а) центры информативных дескрипторов; б) примеры фрагментов 1, 2 на изображениях; в) соответствующие фрагментам 1, 2 дескрипторы 1,2

Пример «плохого» ответа на запрос изображен на Рисунке 15. Данное изображение «побережья» (из группы coastal) имеет такие характерные черты, как линия горизонта, небо, вода и камни. В списке ответов на запрос нет релевантных ответов (все изображения либо из других групп, либо не являются размеченными). Но в них отчасти есть похожие черты: на всех присутствует горизонтальная линия, небо. Дескрипторы, описывающие камни на изображении-запросе считаются похожими с дескрипторами гор на втором изображении и с дескрипторами куста на третьем.

Запрос



Список ответов на запрос



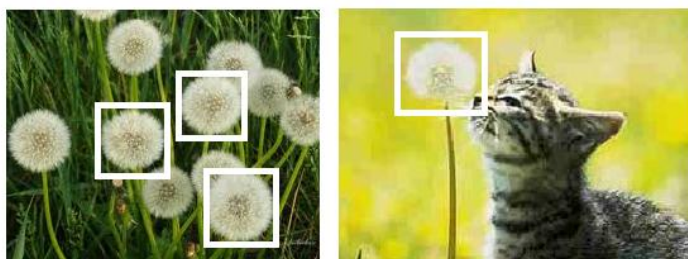
*Рисунок 15. Пример «плохого» ответа на запрос*

В результате экспериментов можно сделать вывод, что метод поиска изображений по признаку самоподобия локальных дескрипторов не превосходит по точности и полноте методы поиска по глобальным дескрипторам.

Одной из возможных причин таких результатов является недостаточное количество дескрипторов на изображениях коллекции, поэтому является целесообразным проверка данного метода на другой коллекции.

Также одной из возможных причин является то, что информативные дескрипторы имеют свойство группироваться, то есть располагаться близко друг к другу. Причем, соседние дескрипторы несут в себе похожую информацию. Предложением является либо их разбиение на небольшие «группировки» и выбор в каждой из них представителя, либо вычисление дескрипторов с небольшим шагом, что позволит избежать появления плотного множества дескрипторов.

Еще одной проблемой является «склеивание» групп, которое изображено на Рисунке 16.



*Рисунок 16. Пример, когда для нескольких групп изображения-запроса находится одна и та же похожая группа*

Под явлением «склеивания» групп подразумевается случай, когда для нескольких групп запросов находится одна и та же похожая группа. При текущей реализации алгоритма это никак не влияет на результат сравнения. Для решения данной проблемы предлагается ввести приоритетный выбор групп. То есть когда находится похожая группа, если она свободная, то она помечается, если нет, то выбирается группа с наибольшим показателем подобия.

## 6. Заключение

При выполнении данной работы были получены следующие результаты:

1. Предложен и реализован адаптированный метод поиска изображения по признаку самоподобия локальных дескрипторов
2. Адаптированный метод поиска по признаку самоподобия локальных дескрипторов был интегрирован в прототип системы поиска изображений по содержанию PhotoFinder, разрабатываемый на математико-механическом факультете СПбГУ
3. Проведено экспериментальное сравнение метода поиска по признаку самоподобия локальных дескрипторов с методами поиска по глобальным дескрипторам, показывающее превосходство методов поиска по глобальным дескрипторам
4. Произведен анализ результатов
5. Предложены идеи модификации алгоритма поиска по признаку самоподобия локальных дескрипторов

## 7. Список литературы

- [1] M. Irani, E. Shechtman, Matchilg Local Self-Similarities across Images and Videos // IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007
- [2] D. Glasner, S. Bagon, M. Irani, Super-Resolution from a Single Image // International Conference on Computer Vision (ICCV), 2009
- [3] [http://en.wikipedia.org/wiki/Color\\_histogram](http://en.wikipedia.org/wiki/Color_histogram)
- [4] Н. Васильева, Построение и комбинирование признаков в задаче поиска изображений по содержанию, 2010
- [5] M. Stricker, M. Orengo, Similarity of Color Images // SPIE Conference, 1995
- [6] M. Stricker, A. Dimai, Spectral Covariance and Fuzzy Regions for Image Indexing // Machine Vision and Applications, 1997
- [7] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification // IEEE Transactions on Systems, Man and Cybernetics, 1973
- [8] H. Tamura, S. Mori, T. Yamawaki, Textural features corresponding to visual perception // IEEE Transactions on Systems, Man and Cybernetics, 1978
- [9] [http://en.wikipedia.org/wiki/Gabor\\_filter](http://en.wikipedia.org/wiki/Gabor_filter)
- [10] A. J. Bell, T.J. Sejnowsky, The “independent components” of natural scenes are edge filters // Vision Research, 1997
- [11] L. David, Object recognition from local scale-invariant features // Proceedings of the International Conference on Computer Vision
- [12] H. Bay, T. Tuytelaars, L. V. Gool, SURF: Speeded Up Robust Features // Computer Vision and image Understanding, 2008
- [13] Y. Ke, R. Sukthankar, PCA-SIFT: A More Distinctive Representation for Local Image Descriptors
- [14] [http://en.wikipedia.org/wiki/Gaussian\\_pyramid](http://en.wikipedia.org/wiki/Gaussian_pyramid)
- [15] [http://en.wikipedia.org/wiki/Integral\\_image](http://en.wikipedia.org/wiki/Integral_image)
- [16] K. G. Derpanis, Integral image-based representations, 2007.
- [17] [http://en.wikipedia.org/wiki/Lab\\_color\\_space](http://en.wikipedia.org/wiki/Lab_color_space)
- [18] P. O. Hoyer, Non-negative matrix factorization with sparseness constraints // JMLR, 2004
- [19] O. Boiman, M. Irani, Detecting irregularities in images and in video // ICCV, Beijing, 2005