

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра Дифференциальных уравнений
Специализация «Компьютерная математика»

АВТАЕВ Александр Сергеевич

Теория вейвлетов и ее применение к поиску изображений

Магистерская диссертация

Руководитель:

МАЛЕВИЧ Александр Эрнестович,
кандидат физико-математических наук,
доцент

«Допустить к защите»

Зав. кафедрой ДУ,

проф. Громак В.И.

«___» _____ 2004 г.

Минск, 2004

Содержание

Содержание	2
Введение. Контекстный поиск изображений.	3
Обзор проблемы	3
Существующие решения.....	4
Краткое содержание работы	4
Цели работы.....	5
Алгоритм университета Вашингтона.....	6
Описание.....	6
Построение метрики	7
Построение индекса	9
Формирование запроса	10
Исходный код.....	10
Примеры запросов	11
Алгоритм «особых» точек	12
Описание.....	12
Извлечение «особых» точек.....	12
Влияние диапазона на качество особых точек	14
Влияние начального уровня на качество особых точек.....	15
Влияние вейвлета на качество особых точек	15
Замечание к качеству «особых» точек.....	15
Цветовые особенности	15
Замечание к выбору цветового пространства	16
Особенности формы	17
Построение метрики	18
Построение индекса	18
Формирование запроса	19
Исходный код.....	19
Тестирование алгоритма	20
Заключение.....	23
Достигнутые результаты.....	23
Возможные направления дальнейших исследований.....	23
Использованные источники информации.....	24
Приложение 1. Системы поиска	25
Приложение 2. Основы теории вейвлетов	26
Вейвлет-преобразование сигнала.....	26
Основные принципы вейвлет-преобразования.....	26
Непрерывное вейвлет-преобразование.	26
Кратномасштабное представление функций	28
Представление функций при помощи вейвлетов.....	29
Дискретное вейвлет-преобразование.....	31
Вейвлет-преобразование для двумерного случая.....	32
Приложение 3. Примеры запросов.....	34
Приложение 4. Особые точки.....	38
Вейвлеты Хаара.	38
Вейвлеты D4.	39
Влияние начального уровня.....	41
Приложение 5. Особенности формы	43
Повороты	43
Разные картинки одного класса	45
Совершенно разные картинки	47
Приложение 6. Примеры запросов.....	49

Введение. Контекстный поиск изображений.

Обзор проблемы

Роль цифровых данных значительно возросла в последние 10 лет в связи с увеличивающимся распространением цифровой техники. Возрастает количество и доступность цифровых фотоаппаратов, сканеров, настольных издательских систем, настольных баз данных. Постоянно увеличивается пропускная способность каналов сети Интернет. Все это приводит к значительному росту количества обрабатываемых данных, однако алгоритмы обработки и поиска не успевают обеспечить адекватную реакцию на столь значительное увеличение объема доступных цифровых материалов. Особенно сильно это ощущается в области средств и алгоритмов поиска в мультимедийных базах данных. На данный момент поиск в большей части таких баз производится с применением принципов и алгоритмов, разработанных несколько десятков лет назад для другого типа данных, что приводит к неудовлетворительным результатам поиска. Большая часть имеющихся на данный момент алгоритмов не учитывает особенности восприятия человеком изображений, и базируется на разметке баз изображений текстовыми тэгами и набором атрибутов, по которым в дальнейшем можно производить поиск с использованием классических алгоритмов поиска строки.

Восприятие человеком информации является трудноформализуемой задачей, и результаты поиска с использованием существующих алгоритмов поиска слабо коррелируют с тем, что хотел бы видеть пользователь, осуществляющий поиск. Данная работа посвящена исследованию алгоритмов контекстного поиска изображений. Цель данных алгоритмов — максимально приблизить результаты поиска изображений к результатам ручного отбора, проведенного человеком.

Зачастую, с увеличением размера базы изображений традиционные алгоритмы исчерпывают свои возможности. Например, для ускорения поиска можно просматривать уменьшенные копии изображений. Поиск изображения среди сотен уменьшенных копий еще является посильной задачей для пользователя, однако увеличение порядка размера базы до тысяч изображений делает данную задачу практически невозможной. Поэтому при существенном расширении базы данных поиск методом полного перебора изображений теряет свою эффективность.

Одной из распространенных стратегий поиска является индексация по ключевым словам и атрибутам. Данный подход имеет свои преимущества, однако выдвигает ряд требований, исполнение которых порой становится чрезвычайно затруднительным. Первой причиной является, собственно необходимость разметки базы изображений ключевыми словами, которая является весьма затратной процедурой. Второй причиной является то, что некоторые визуальные аспекты изображения в принципе являются трудноописуемыми или неопишуемыми вообще. Далее, пользователю может быть неизвестно, какие аспекты внесены в индекс, а какие нет. Стоит отметить, что некоторые классы изображений (например, текстуры) вообще с большим трудом поддается разметке ключевыми словами. Третьей причиной является неоднозначность восприятия изображения и, как следствие из этого, поиск по базе изображений, размеченной тэгами одним человеком для другого значительно менее эффективен.

Изложенные выше соображения привели к росту исследований в области контекстного поиска изображений. Контекстный поиск изображений имеет ряд других названий, среди которых «поиск по содержимому», «поиск по образцу», «поиск по эскизу», «поиск по методу подобия». Стоит отметить, что контекстный поиск является альтернативой традиционным методам поиска и может применяться совместно с ними.

Поиск изображения исходя из содержания искомого изображения находит применение во многих областях, включая архитектуру, телевидение, средства мультимедиа, графический дизайн, историю искусств, криминологию, геологию,

медицину. Дизайнер может потребовать найти изображения, похожие на данное, для использования в качестве альтернативы или на замену. В криминологии данные алгоритмы востребованы при поиске фотографии по фотороботу. Также для правоохранительных органов актуальным является поиск фрагмента или целого изображения в записи камер наблюдения. Как правило, при этом используется фотография грубого разрешения. Архитектор может использовать данный метод поиска для отбора решений, которые являются удачными для данной местности, а также для поиска аналогов задумываемого или исполненного строения. Данная задача также является актуальной и в домашней сфере. Увеличившаяся доступность цифровой аппаратуры привела к значительному росту домашних видео- и фото- архивов. Контекстный поиск призван качественно улучшить результаты поиска в цифровых библиотеках [1]. Во всех упомянутых областях совершенствование технологии формирования изображения является важной и общепризнанной проблемой.

Существующие решения

За последние 10 лет в свете усиления интереса к данной проблеме появилось несколько систем контекстного поиска изображений. Стоит упомянуть наиболее известные системы, базирующиеся на поиске с использованием ключевых слов:

1. HotBot (<http://hotbot.lycos.com>)
2. NBCi (<http://www.snap.com>)
3. Yahoo! Image Search (<http://search.yahoo.com/images>)
4. Lycos multimedia searcher (<http://multimedia.lycos.com>)
5. AltaVista Images Search Center (<http://www.altavista.com/image>)
6. Google Image Search (<http://images.google.com>)
7. PICSearch (<http://www.picsearch.com/>)

Наиболее старой и известной системой является система контекстного поиска QVIC (Query By Image Content), разработанная компанией IBM. Некоторые исследователи [4] отмечают, что данная система достаточно успешна в работе с цветом и текстурой, однако недостаточно точна при работе с формой изображения. Приложение 1 данной работы содержит более широкий список систем, осуществляющих контекстный поиск. Большинство данных систем являются результатом исследований и скорее полигоном для испытания идей исследователя, нежели законченным продуктом. Преимущественно этим обосновывается то, что алгоритмы, лежащие в основе данных систем обыгрывают один или два аспекта контекстного поиска. Большая часть этих систем использует цветовые и текстурные особенности, небольшая часть использует пространственные особенности, т.е. местоположение (в некоторых случаях распределение) особенности на картинке. Поиски по цветовым особенностям, как правило, дают неплохие результаты, несколько хуже обстоит дело с поиском по текстурным особенностям. Исключением являются случаи присутствия в базе некой превалирующей текстуры. Поиск по форме изображения на данный момент дает плохие результаты в большинстве данных систем.

Краткое содержание работы

В данной работе предлагается алгоритм, позволяющий извлекать информацию о форме изображенного объекта, рассматриваются варианты использования этой информации при построении индекса, также рассматривается алгоритм контекстного поиска, предложенный ранее исследователями университета Вашингтона (США). Рассмотренные алгоритмы активно используют особенности вейвлет-разложения изображения для извлечения набора особенностей, в дальнейшем используемых при индексировании и поиске.

Цели работы

Данная работа ставит целью исследование алгоритмов контекстного поиска изображений и разработку нового алгоритма, позволяющего использовать форму объекта при поиске.

Алгоритм университета Вашингтона

Описание

Одним из наиболее известных алгоритмов контекстного поиска является алгоритм исследователей из университета Вашингтона [5,6]. Алгоритм позволяет проводить поиск в базе изображений по представленному пользователем рисованному эскизу или рисунку грубого разрешения.

Главными причинами рассмотрения данного алгоритма является иллюстрация базовой идеи использования кратномасштабного представления с целью извлечения информации для индексирования, а также чрезвычайно эффективный способ построения индекса, предложенный для организации и хранения извлеченных особенностей. Также данный алгоритм сравнительно прост для понимания и реализации.

Наиболее очевидным способом измерения расстояния между двумя изображениями является использование метрики L_1 или L_2 . Однако данные метрики удовлетворительно работают только с полностью идентичными изображениями. Для запроса в грубом разрешении эти метрики дают очень большие ошибки, связанные с искажениями, внесенными при уменьшении изображения или сканировании. Еще большие ошибки данные метрики вносят при использовании в качестве запроса рисованного эскиза. Метод поиска должен учитывать такие возможные искажения, как изменение цвета, низкое разрешение, нечеткость контура изображения-запроса, сдвиги и, возможно, повороты. Таким образом, входные данные рассматриваются как некое приближение к искомому изображению.

Для упорядочивания базы изображений определяется метрика, использующая укороченные квантованные версии вейвлет-разложений, называемые сигнатурами. Эти сигнатуры содержат только наиболее важную информацию об изображении. Предлагаемая метрика сравнивает количество общих значимых коэффициентов в разложении изображения-запроса и потенциального целевого изображения. Стоит отметить также эффективную организацию индекса, позволяющую производить чрезвычайно быстрый поиск.

Наиболее популярные из использованных ранее методов поиска включали индексирование цветовых гистограмм и текстуры. Некоторые системы позволяли указывать некое сочетание фиксированного набора параметров поиска. Например, система QBIC [7] позволяет формировать запрос на основе целого ряда параметров, характеризующих различные визуальные аспекты изображения. Пользователь может задать процентный состав определенного цвета изображения, конкретную текстуру, некоторые особенности формы и даже эскиз главных контуров изображения. Пользователь также может указывать для каждого из использованных атрибутов относительный вес. Также система QBIC позволяет аннотировать изображения и проводить поиск по аннотациям, а также по фиксированному набору атрибутов.

В рассматриваемом алгоритме, напротив, основной упор делается на возможность совершать поиск непосредственно по изображению-запросу без необходимости указывать дополнительные атрибуты и без дальнейших указаний пользователя базе данных.

Как уже упоминалось, рассматриваемый алгоритм использует вейвлет-разложение изображения-запроса, что позволяет проводить запросы к базе данных при переменном разрешении, в том числе потенциально отличном от разрешения целевого изображения. Кроме того, данный алгоритм теоретически позволяет извлекать сигнатуры непосредственно из сжатых версий изображения. Данная возможность стала особенно актуальна после принятия комитетом ISO стандарта 15444, более известного как JPEG2000 [8]. Для использования данной возможности необходимо провести дополнительные исследования, поскольку в JPEG2000 применяется другой набор фильтров для разложения изображения перед сжатием. Однако стоит учесть, что основной

идеей алгоритма является использование кратномасштабного представления, а не использование специфически сконструированного базиса вейвлетов для разложения.

Построение метрики

Рассмотрим задачу вычисления расстояния между изображением-запросом Q и потенциальным целевым изображением J . Наиболее очевидным выбором являются L_1 или L_2 нормы:

$$\|Q - J\|_1 = \sum_{i,j} |Q(i, j) - J(i, j)|$$
$$\|Q - J\|_2 = \left(\sum_{i,j} (Q(i, j) - J(i, j))^2 \right)^{1/2}$$

Однако данные метрики не только чрезвычайно медленно считаются, но и весьма неэффективны, поскольку не учитывают неточности и «вольности» изображения-запроса.

Для построения данной метрики предлагается использовать двумерное вейвлет-разложение (см. [9], Приложение 2) и, в частности, кратномасштабное представление изображения.

Стандартным и наиболее распространенным цветовым пространством является RGB. Однако существует ряд других цветовых пространств, которые являются, в некоторых случаях, более эффективными для использования при запросах. В результате экспериментов, разработчики алгоритма использовали цветовое пространство YIQ, наиболее известное по применению его в телевизионном формате NTSC, который являющимся стандартом телевидения в США. Данное цветовое пространство интересно, в частности, неравномерностью распределения информации по цветовым каналам. Наиболее значительную часть информации (яркость) несет канал Y. Данная особенность используется позднее при подборе весовых коэффициентов.

На данный момент при обработке изображений активно используется несколько базисов вейвлетов. Наиболее простыми и быстрее всего вычисляемыми являются вейвлеты Хаара. В приложении к рассматриваемому алгоритму они также имеют преимущество в хорошем представлении больших равномерно окрашенных участков изображения, которых много в типичном запросе-эскизе.

В целях повышения эффективности метрики авторы уменьшают изображение до размера 128x128 пикселей и уже уменьшенное изображение подвергают процедуре вейвлет-преобразования. После этого каждый цветовой канал изображения представляет собой последовательность из 16384 коэффициентов. Данная последовательность все еще очень велика для непосредственного использования при построении индекса, поэтому предлагается использовать усеченную версию данного разложения, содержащую менее ста наиболее значимых коэффициентов. Более того, уменьшение числа хранимых коэффициентов оказывает положительное влияние на производительность и качество работы метрики.

Далее, каждый коэффициент после разложения представляет собой вещественное число. Авторы предлагают квантовать данное значение до двух уровней: +1 и -1. Данный прием, как и усечение, позволяет увеличить эффективность поиска, а также применить специальную технику построения индекса, позволяющую значительно ускорить поиск.

Введем некоторые обозначения. Пусть Q обозначает один цветовой канал изображения-запроса, а J — потенциального целевого изображения. $Q(0,0)$ и $J(0,0)$ представляют собой коэффициенты масштабирующих функций, соответствующие среднему значению интенсивности данного цветового канала. Далее, $Q1$ и $J1$ будут обозначать усеченные квантованные версии матриц Q и J . Значения данных матриц могут принимать значения -1,0,+1. Для удобства примем $Q1(0,0)=0$, $J1(0,0)=0$.

Подходящая для использования метрика выражается формулой:

$$\|Q - J\| = w_{0,0} |Q_{0,0} - J_{0,0}| + \sum_{i,j} w_{i,j} |Q1(i,j) - J1(i,j)|,$$

где $w_{i,j}$ являются весовыми коэффициентами. Данную метрику можно упростить с целью повышения эффективности ее расчета без потери функциональности. После упрощения данная метрика потеряет право называться метрикой, однако это не влияет на ее работоспособность.

Для уменьшения числа коэффициентов, которые необходимо определить экспериментальным путем, сгруппируем их по уровням разложения. Далее, будем рассматривать только те коэффициенты вейвлет-разложения запроса, которые не равны нулю. Преимущество подобного подхода заключается в том, что не детализированный запрос сопоставляется детализированному изображению, и наоборот, детализированный запрос не поставится в соответствие изображению с отсутствующей деталью. После этого упрощения получаем:

$$\|Q - J\| = w_0 |Q(0,0) - J(0,0)| + \sum_{i,j:Q1(i,j) \neq 0} w_{bin(i,j)} (Q1(i,j) - J1(i,j))$$

Далее разность между коэффициентами можно заменить неравенством, т.е. метрика приобретает вид:

$$\|Q - J\| = w_0 |Q(0,0) - J(0,0)| + \sum_{i,j:Q1(i,j) \neq 0} w_{bin(i,j)} (Q1(i,j) \neq J1(i,j))$$

где оператор \neq принимает численное значение 1 при выполнении неравенства.

Далее, поскольку операция сравнения производится быстрее, чем неравенства, и поскольку искомое множество изображений составляет всего лишь часть базы, производится замена неравенства на равенство следующим образом и второе слагаемое предыдущей формулы принимает вид:

$$\sum_{i,j:Q1(i,j) \neq 0} w_{bin(i,j)} - \sum_{i,j:Q1(i,j) \neq 0} w_{bin(i,j)} (Q1(i,j) = J1(i,j))$$

Поскольку первое слагаемое не зависит от J, мы можем им пренебречь в целях ускорения расчета. На окончательный результат это не влияет. Таким образом, после ряда упрощений, выражение приобрело вид:

$$w_0 |Q(0,0) - J(0,0)| - \sum_{i,j:Q1(i,j) \neq 0} w_{bin(i,j)} (Q1(i,j) = J1(i,j))$$

Весовые коэффициенты дают возможность настроить данную метрику на конкретную базу изображений и цветовое пространство. Для уменьшения числа коэффициентов предлагается группировать вейвлет-коэффициенты по диапазонам разложения следующим образом, используя функцию bin:

$$bin(i, j) = \min(\max(level(i), level(j)), 5)$$

где $level(i)$ обозначает уровень, на который попадает коэффициент. В данном случае он равняется $\log_2(i)$. Ограничение в 5 является следствием принятия размера картинки для индексирования в 128x128 пикселей. Если брать более точное разрешение, то можно брать уровни разложения ниже, однако большого улучшения качества метрики это не дает. Ниже на рисунке 1 цветом показано, как именно группируются коэффициенты. Такая группировка обусловлена кратномасштабным представлением изображения, при котором наиболее важные коэффициенты, представляющие более грубую версию изображения, располагаются в верхнем левом углу.

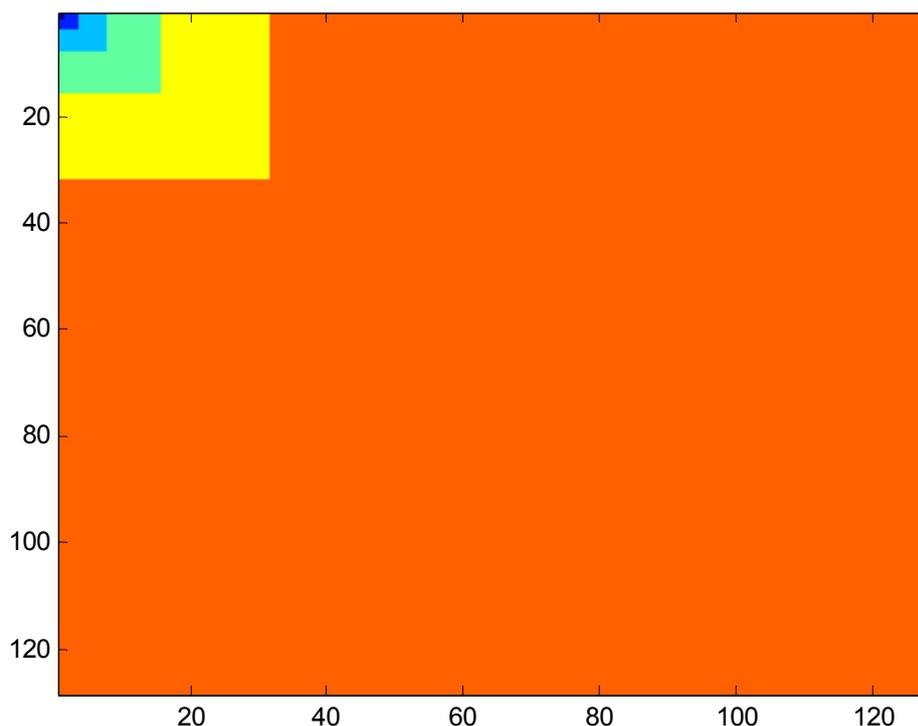


Рисунок 1

Стоит отдельно отметить данные весовые коэффициенты. Они являются одновременно достоинством и недостатком данного алгоритма. Достоинством, поскольку метрика является настраиваемой благодаря ним. Недостатком, поскольку настраивать все же необходимо. Авторы предлагают некий набор коэффициентов (см. таблицу 1), разработанный с использованием статистических методов на их модельной базе изображений.

Данные коэффициенты равны:

Таблица 1

b	Нарисованные запросы			Сканированные запросы		
	W _y	W _i	W _q	W _y	W _i	W _q
1	4.04	15.14	22.62	5.00	19.21	34.37
2	0.78	0.92	0.40	0.83	1.26	0.36
3	0.46	0.53	0.63	1.01	0.44	0.45
4	0.42	0.26	0.25	0.52	0.53	0.14
5	0.41	0.14	0.15	0.47	0.28	0.18
6	0.32	0.07	0.38	0.30	0.14	0.27

Построение индекса

Рассматриваемый алгоритм выгодно отличается от многих других интересным построением индекса, которое стало возможным благодаря квантованию коэффициентов, описанному выше. Построение индекса начинается с извлечения изображения из файла, затем его размер изменяется до 128x128 пикселей, после чего необходимо провести преобразование цветового пространства из RGB в YIQ. В таком виде изображение подвергается вейвлет-преобразованию, затем коэффициент J(0,0) сохраняется в индексе. Далее коэффициенты усекаются и квантуются. После этапа квантования и усечения у нас имеется данные о знаке и местоположении оставшихся коэффициентов. Эта информация объединяется в 6 индексных массивов — по одному на каждый цветовой канал (3 штуки) и знак коэффициента (2 знака). Иными словами, например, массив Y+ длиной 16384 в

ячейке X содержит номера изображений, коэффициент X в разложении цветового канала Y которых большой положительный.

Формирование запроса

Начальный этап формирования запроса совпадает с этапом формирования индекса для каждой отдельной картинке. Сначала извлекается изображение из файла либо, в случае эскиза, из редактора, затем его размер изменяется до 128x128 пикселей, после чего проводится преобразование цветового пространства из RGB в YIQ. В таком виде изображение подвергается вейвлет-преобразованию, затем коэффициент $J(0,0)$ оставляется, другие коэффициенты усекаются и квантуются. После этапа квантования и усечения у нас имеются данные о знаке и местоположении оставшихся коэффициентов.

Инициализация поиска состоит в заполнении массива оценок файлов нулями. Далее массив оценок начинает накапливать оценки. Первая часть оценки состоит из взвешенной разности средних цветов, т.е. $w_0(Q(0,0) - J(0,0))$. Эти оценки суммируются по каждому цветовому каналу. Далее по каждому цветовому каналу производится корректировка оценок в сторону уменьшения следующим образом. Для каждого ненулевого коэффициента запроса находится список файлов, имеющих коэффициент в этом же цветовом канале такого же знака. Оценка каждого файла из списка уменьшается на число $w_{bin(i,j)}$. Окончательный этап — отбор необходимого количества наиболее релевантных изображений. Ими являются изображения с наименьшими оценками.

Исходный код

Ниже представлен наиболее существенный участок кода функции запроса (синтаксис MATLAB, используется Wavelet Toolbox):

```
for j=1:3
    %decompose color plane
    [c,s]=wavedec2(img(:,:,j),maxlev,wFam);
    %extract CoeffCount of biggest coefficients and their coordinates
    midColor=c(1);
    absC=sort(abs(c(2:end)));
    coeffLimit=absC(length(absC)-CoeffCount);
    sig=[];
    for k=2:length(c)
        if abs(c(k))>coeffLimit
            sig=[sig sign(c(k))*k];
        end;
        if length(sig)==CoeffCount
            break;
        end;
    end;

    %colors
    for i=1:filesCount
        scores(i)=scores(i)+weights(1,j)*abs(midColor - Index.midColors(j,i));
    end;

    plInd=(j-1)*2+1;
    mnInd=(j-1)*2+2;
    %update scores for every files that have particular this coefficient
    for k=1:length(sig)
        if sig(k)>0
            fIdx=Index.idx{plInd, sig(k)};
            for i=1:length(fIdx)
                scores(fIdx(i))=scores(fIdx(i))-weights(binFunctionFlatIndex(sig(k), ThSize),j);
            end;
        else
            fIdx=Index.idx{mnInd, -sig(k)};
            for i=1:length(fIdx)
                scores(fIdx(i))=scores(fIdx(i))-weights(binFunctionFlatIndex(-sig(k), ThSize),j);
            end;
        end;
    end;
end;
end;
```

```
[scores, scIdx] = sort(scores);  
scIdx = scIdx(1:ResultCount);
```

Примеры запросов

В приложении 3 приведены примеры запросов к базе данных из 858 изображений нескольких классов разной степени однородности, в том числе один класс изображений с различными по характеру изображениями.

Верхнее левое изображение на каждой иллюстрации является запросом. Второе изображение во всех иллюстрациях также является копией запроса, т.е. алгоритм уверенно его находит.

Некоторые иллюстрации стоит прокомментировать отдельно:



На данной иллюстрации (рис. 2) первое изображение является запросом, второе — его копией, третье и четвертое — чуть-чуть измененными копиями. Пятое, шестое и седьмое изображения являются копиями запроса, повернутыми по часовой стрелке на 5, 10 и 20 градусов. Это показывает, что алгоритм достаточно уверенно распознает данный тип искажений, однако изображение с поворотом в 30 градусов по часовой, имевшееся в базе, алгоритм уже не нашел.

Рисунок 2



Данная иллюстрация (рис. 3) является примером сбоя алгоритма. Несмотря на совпадение цветовой гаммы изображений, алгоритм очень плохо «почувствовал» форму изображения. На данном примере это видно весьма наглядно. Как видно даже на этом рисунке, изображения более похожие (на субъективный человеческий взгляд) в базе присутствуют, но даже в запросе они ранжированы неверно, что свидетельствует об их включении в список по цвету.

Рисунок 3

Алгоритм «особых» точек

Описание

Данный алгоритм основан на идее «особых» (salient) точек [10]. Особые точки представляют собой точки изображения, в которых происходит сильное изменение изображения. Человеческий глаз выделяет такие точки автоматически, важность информации, которую они несут, является очевидной (см. примеры ниже).

Ниже приведен пример особых точек изображения. Как видно на рисунках 4 и 5, точки образуют четко различимый контур основного объекта изображения.



Рисунок 4



Рисунок 5

Главная идея состоит в том, что особые точки позволяют выделить форму объекта, а точнее, его локальные особенности. Применяя эти точки в качестве основы или компоненты метрики можно получить изображения, имеющие те же очертания.

Извлечение «особых» точек

Справедливости ради стоит заметить, что данный алгоритм не является единственным алгоритмом извлечения информации об особенностях изображения. Давно известны так называемые «детекторы Харриса» — алгоритм, позволяющий извлечь информации об «углах» в изображении [11]. Данные детекторы разрабатывались для применения в робототехнике и имеют определенные недостатки, непосредственно

связанные с их происхождением — они плохо работают с естественными¹ изображениями.

В силу значимости вклада, вносимого в метрику особыми точками, алгоритм их извлечения будет рассмотрен подробно и подвергнут некоторым улучшениям.

Основой для работы данного алгоритма является кратномасштабное представление изображения, пример которого изображен на рисунке 6. Уровни грубого разрешения вейвлет-разложения сохраняют только наиболее важную информацию об изображении, причем важность этой информации легко измеряется абсолютной величиной коэффициента. Каждый коэффициент в грубом разрешении соответствует целому региону изображения в более четком разрешении. Идея состоит в нахождении точек высокого разрешения, соответствующих глобальному изменению в низком разрешении.

Рассмотрим изображение на разных уровнях разложения. Обозначим интересующие нас уровни $1/2^j$, где $j \leq -1$. Обозначим уровень разложения $W_{2^j} f$, где f — изображение. Для разложения нужно использовать ортогональные вейвлеты с компактным носителем. Это обеспечивает два существенных для алгоритма момента: представление изображения получается полное и избыточное, а во-вторых, в силу компактности носителя, можно точно вычислить множество точек, на основе которых произведен расчет каждого конкретного коэффициента.

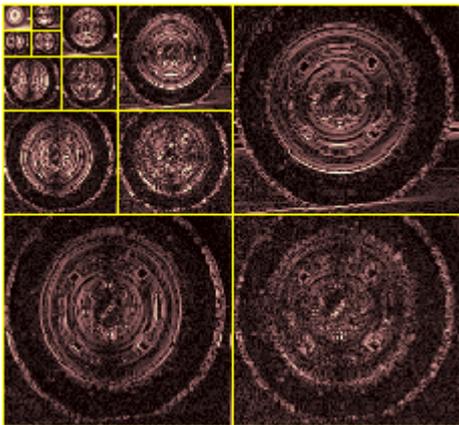


Рисунок 6

Возьмем уровень разложения j и коэффициент $f(n)$. Мы можем вычислить, какие коэффициенты на уровне $j+1$ вычислены на основе того же множества точек, что и коэффициент $f(n)$. Назовем их «дочерними» и обозначим $C(W_{2^j} f(n))$. Коэффициент $f(n)$ представляет изменение набора точек сигнала. «Дочерние» коэффициенты представляют изменение некоего подмножества точек вышеупомянутого набора точек. Наиболее «особым» множеством является то, которое соответствует максимальному по абсолютному значению коэффициенту на предыдущем уровне j . Рассмотрим данный максимум и возьмем его максимальный «дочерний». Повторяя

данный процесс, мы доберемся до уровня $1/2$, который представляет последний уровень перед изображением. У нас будут отобраны некоторые коэффициенты, представляющие множества точек изображения. Среди данных точек мы выберем точку с наибольшим градиентом. Примем за значение «особенности» данной точки сумму абсолютных значений отобранных нами коэффициентов:

$$saliency = \sum_{k=1}^{-j} |C^{(k)}(W_{2^j} f(n))|$$

Точка, в которой происходит некое глобальное изменение будет иметь большое значение «особенности», поскольку в него внесли вклад большие коэффициенты. Менее значительное изменение также приведет к извлечению точки, но с меньшим значением «особенности».

Для простоты изложения и понимания выше изложен одномерный случай извлечения особых точек. В двумерном случае в нашем распоряжении имеется три поддиапазона уточняющих коэффициентов, обозначаемых в разных источниках как HL, LH, HH, или H, V, D. Рассмотрим отбор по разным диапазонам.

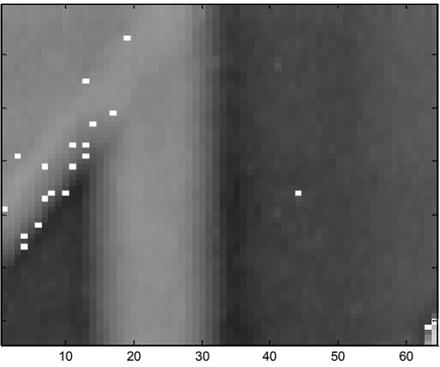
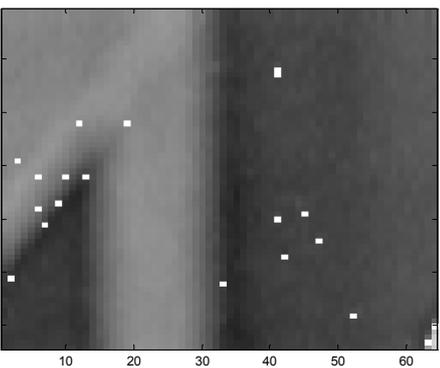
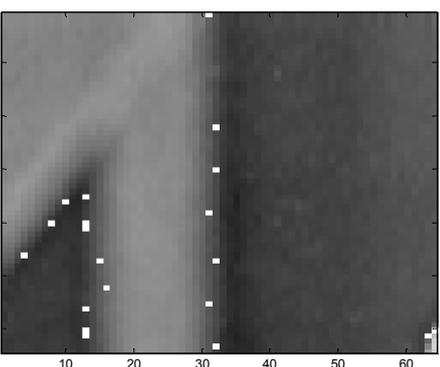
¹ Естественное изображение — изображение природного объекта, например ландшафт, цветок, животное. Изображение, например, строний, таковым не является.

Влияние диапазона на качество особых точек

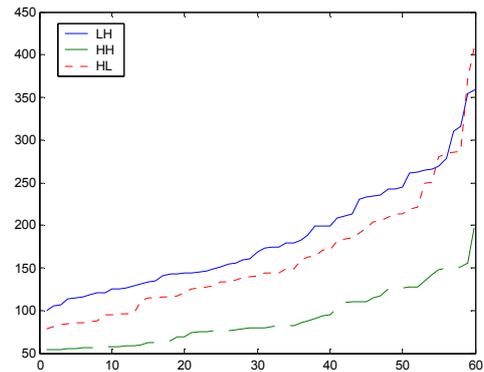
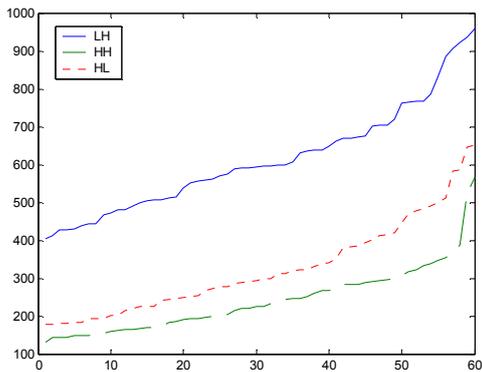
В случае одномерного сигнала алгоритм работает по одному диапазону — detail-коэффициентам разложения. В случае двумерного сигнала есть три варианта — HL, LH, HH диапазоны. Разницу в получаемых результатах иллюстрируют рисунки. Особые точки выделены белым цветом.

На трех рисунках ниже крупным планом показаны особые точки, извлеченные алгоритмом. Более подробные версии изображений можно посмотреть в Приложении 4.

Для улучшения результатов отбора можно как прибегнуть к самому простому приему — отобрать лучшие точки по разным диапазонам и скомбинировать их.

	<p>Особые точки по диапазону LH (верхний правый). Данный диапазон устремляет точки к горизонтальным линиям.</p>
	<p>Особые точки по диапазону HH (нижний правый) Данный диапазон устремляет точки как к горизонтальным линиям, так и к вертикальным линиям.</p>
	<p>Особые точки по диапазону HL (нижний левый). Данный диапазон устремляет точки к вертикальным линиям.</p>

Ниже приведены графики значений особенности (saliency) для трех диапазонов по выборке из 60 лучших точек:



Левый рисунок Lena и правый рисунок Daubechies из набора стандартных изображений WaveLab. На графиках заметны довольно резкие скачки вверх возле правого края графика — у лучших точек.

Влияние начального уровня на качество особых точек

Для вычисления «особых» точек необходимо выбрать начальный уровень, с которого начнется их отбор. Уровни грубого разрешения быстрее приводят к результату, поскольку уменьшается число точек для расчета. Уровни высокого разрешения увеличивают число точек для расчета значения saliency. Кроме влияния на количество отбираемых точек, это оказывает и влияние на качество «особых» точек. В приложении 4 можно оценить влияние уровня на качество особых точек.

Влияние вейвлета на качество особых точек

Разложение может быть проведено с использованием нескольких базисов. Быстрее всего считается базис Хаара. Для вычисления точек подходит и базис Добеши. Наибольшее влияние здесь оказывает длина носителя, как величина, определяющая размер множества «дочерних» точек. С увеличением длины носителя точки быстрее стремятся к своим предельным значениям. Это влияет как на количество доступных в итоге точек, так и на их качество. В приложении 4 можно сравнить извлеченные «особые» точки с помощью вейвлета Добеши 4 и Хаара.

Замечание к качеству «особых» точек

Несмотря на проанализированные выше вариации алгоритма, выводы о том, какой диапазон или начальный уровень брать сделать сложно в виду субъективности такого параметра как качество «особых» точек. Тем не менее, определенные выводы, исходя из приведенных примеров, сделать можно.

Цветовые особенности

«Особые» точки являются базой алгоритма, на которой вычисляются какие-то отличительные свойства (features) картинки, которые входят в индекс и составляют сигнатуру данной картинки. Цвет является один из важнейших параметров восприятия изображения, поэтому одним из свойств, включаемых в сигнатуру, стали цветовые особенности картинки. Они активно используются всеми алгоритмами — в форме гистограмм, распределений, либо каким-либо иным образом обработанные. Например, алгоритм университета Вашингтона использует среднее значение цвета каждого канала как одну из составляющих метрики. Подавляющее число алгоритмов, тем не менее, использует глобальную информацию о цвете изображения. В силу доступности нам «особых» точек изображения, сконцентрированных в локально важных местах изображения, мы будем использовать цветовые особенности локально. Для каждой особой

точки будет извлекаться три числа на каждый цветовой канал. Эти три числа являются тремя моментами — первого, второго и третьего порядков.

$$\mu_i = \frac{1}{N} \sum_{j=1}^N I_{ij}$$

$$\sigma_i = \left(\frac{1}{N} \sum_{j=1}^N (I_{ij} - \mu_i)^2 \right)^{1/2}$$

$$s_i = \left(\frac{1}{N} \sum_{j=1}^N (I_{ij} - \mu_i)^3 \right)^{1/3}$$

Таким образом, наша будущая метрика будет содержать локальную информацию о цвете изображения. Данные числа будут извлекаться из квадрата соседних точек размером 3x3 пикселя. В силу большей перцепционной однородности цветового пространства HSV [12], мы будем проводить расчеты именно в нем.

Замечание к выбору цветового пространства

Выбор цветового пространства является достаточно важным параметром, влияющим на работу алгоритма. Следующие иллюстрации и графики аргументируют в пользу пространства HSV. Цветовое пространство практически не влияет на выбор особых точек при условии хорошей контрастности картинки (рис. 8), однако достаточно сильно влияет при условии плохой контрастности изображения (рис. 7).

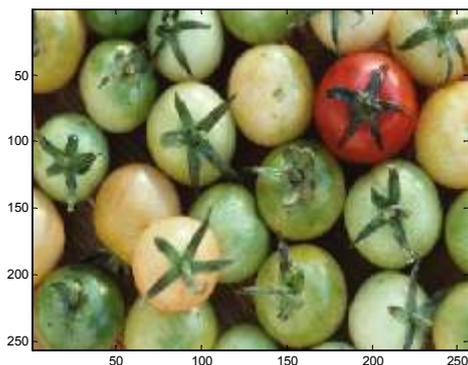


Рисунок 7
«Проблемная» картинка (правый верхний помидор красного цвета, остальные — зеленого).

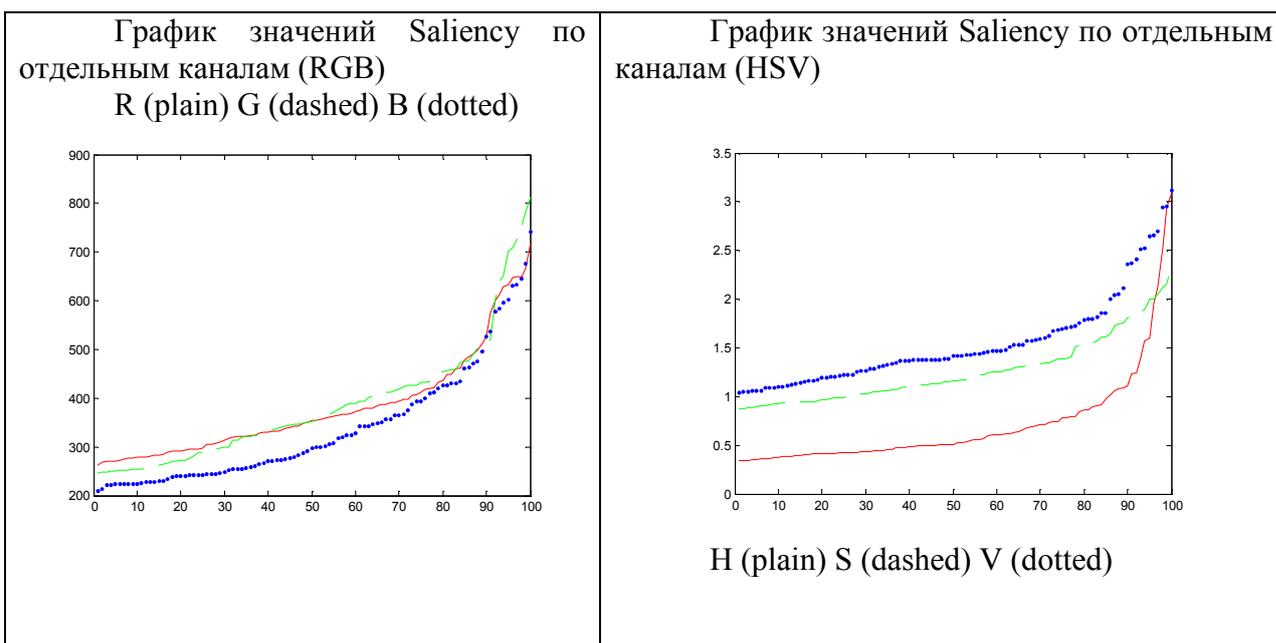
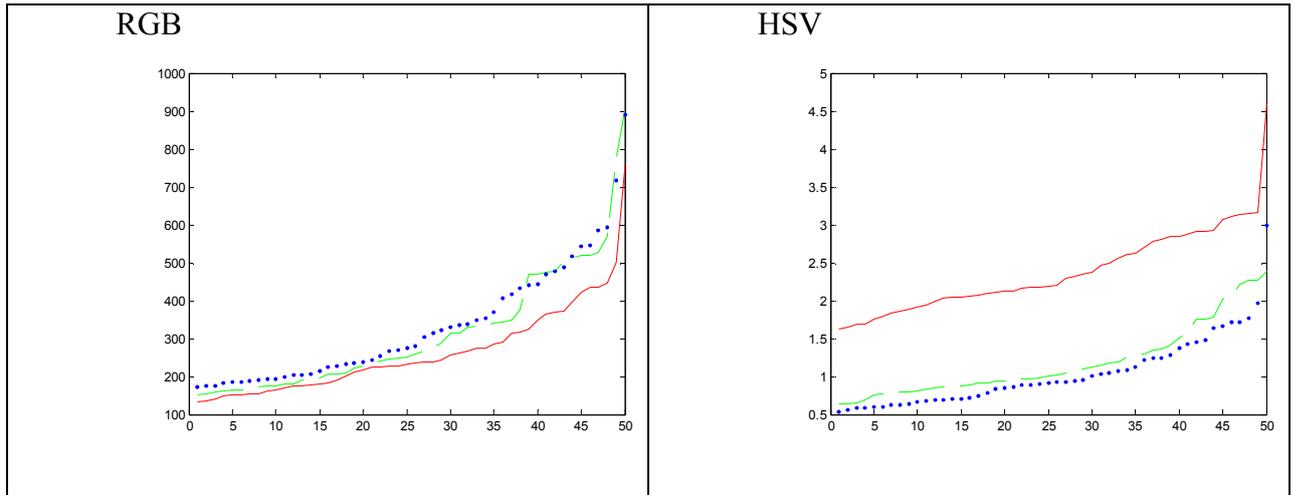
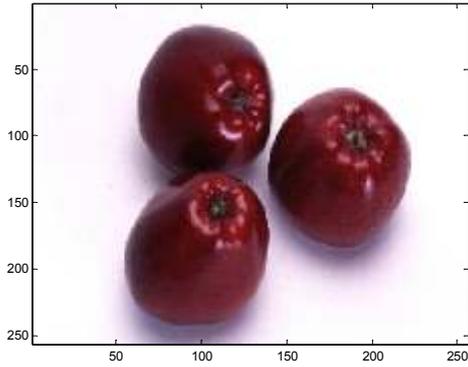


Рисунок 8
«Нормальная» картинка.



Графики, демонстрирующие значения «особенности» для рисунка 7, более ровные и в целом возрастают более быстро ближе к правой границе графика, где и располагаются наиболее «особенные» точки.

Особенности формы

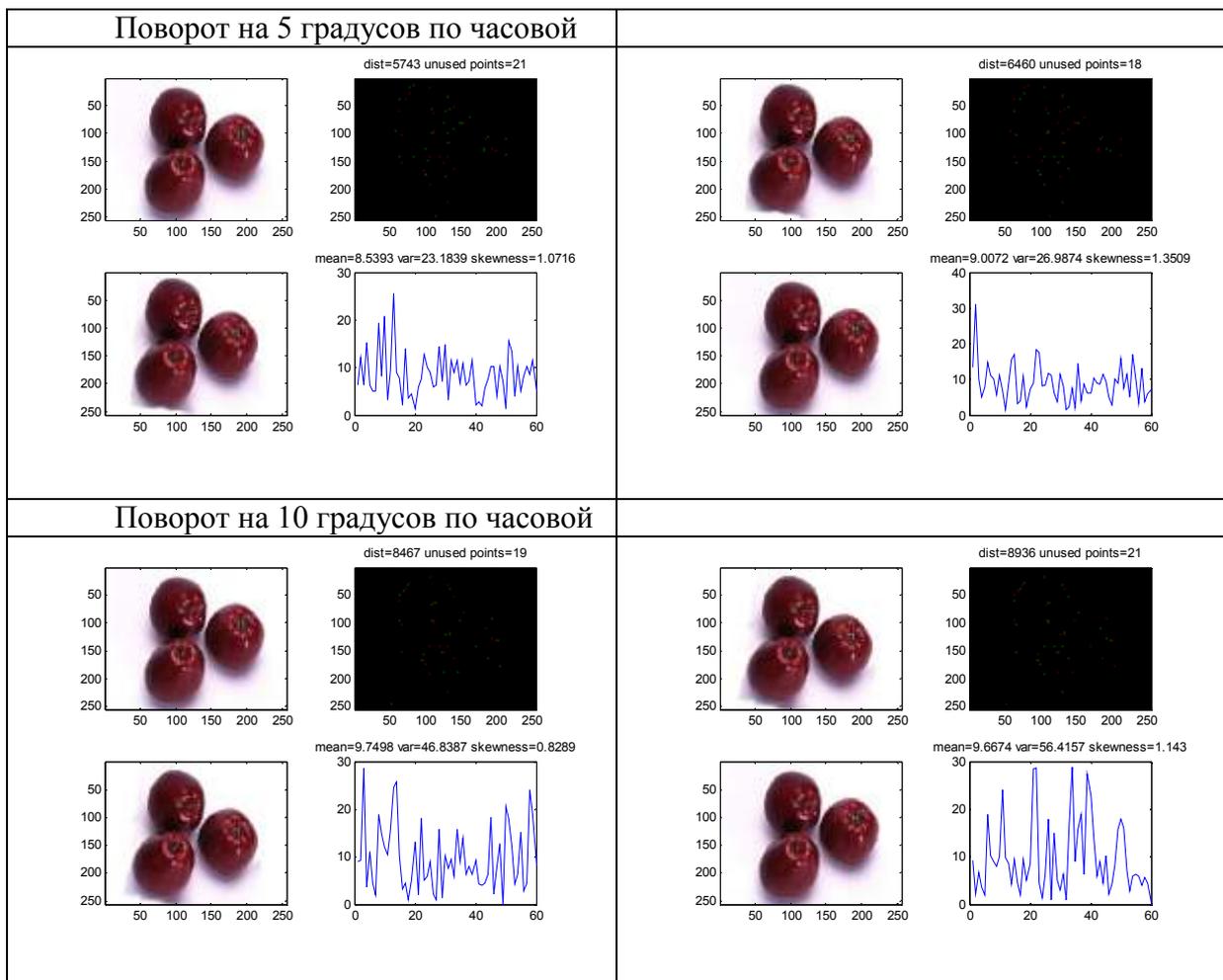
Форма объекта также является достаточно важным параметром отбора изображений. Используя координаты особых точек изображения, мы можем произвести индексацию изображений также по схожести формы. Рассмотрим две картинки — запрос и потенциальный результат. Расстояния между наборами «особых» точек можно посчитать как сумму расстояний от каждой точки до ближайшей точки другой карты. Безусловно, данная сумма не является метрикой в строгом смысле этого слова — данная «метрика» не является симметричной. Тем не менее, она достаточно хорошо сравнивает картинки с объектами схожей формы.

Рассмотрим устойчивость относительно поворота, как одного из видов искажений. Для этого сравним изображение исходное и его копию, повернутую на небольшой угол. Левый столбец содержит изображения, исходное и модифицированное. Правый столбец содержит карты особых точек и графики расстояний от каждой точки первой картинки до ближайшей «особой» точки другой картинки. Над картами точек отображены следующие параметры:

Dist — расстояние

Unused points — количество точек, не участвовавших в расчете.

Mean, var, skewness — соответствующие моменты.



Можно извлекать несколько «уровней» данной особенности. Первая и главная — это непосредственно расстояние. Оно показывает, насколько сильно две карты особых точек отличаются «в общем». Вторым параметром можно взять достаточно показательный параметр — число точек, оставшихся вне расчета. Третьим, четвертым и пятым параметров можно взять значения mean, var, skewness. Данные значения также характеризуют расстояние. Довольно интересна взаимосвязь первого и второго момента при сдвиге изображения, однако данный аспект оставлен для будущих исследований.

Более подробные графические данные содержатся в приложении 5.

Построение метрики

Метрика для данного алгоритма является составной. В качестве составляющих используются цветовые особенности и особенности формы, взвешенные с определяемыми пользователем коэффициентами. Введение коэффициентов делает данную метрику настраиваемой, однако указание их необязательно, можно воспользоваться предлагаемыми по умолчанию значениями, которые придают цвету и форме одинаковый приоритет. Цветовая составляющая считается как обычное расстояние между векторами, в которые собраны цветовые особенности каждой картинке в «особых» точках. Расчет составляющей формы описан выше.

Построение индекса

Данный алгоритм использует обычный индекс, т.е. особенности картинок записываются подряд в одномерный массив, по которому в дальнейшем осуществляется поиск.

Формирование запроса

Этап формирования запроса начинается с вычисления «особых» точек изображения. Далее для каждой «особой» точки рассчитываются цветовые особенности. После происходит инициализация массива оценок нулями. В основном цикле запроса считается расстояние до каждой картинке как

$$D_j = \sum_i W_i S_j(f_i)$$

где j — индекс картинке в массиве, W — массив весов, i — номер особенности (т.е. 1 для цвета и 2 для формы).

После расчета оценок алгоритм отбирает N наиболее релевантных картинок, т.е. имеющих наименьшие оценки.

Исходный код

Ниже приведена наиболее существенная часть программы запроса изображения (синтаксис MATLAB, использован Wavelet Toolbox).

```
disp('decomposing query...');
%decompose query
img=imread(QueryFileName);
%resize to thumbnail
img=imresize(img, [ThSize ThSize]);
%extract points
salIdx=ExtractSalientPoints(img, salPtsCount);
%convert to HSV
img=rgb2hsv(img);

features=zeros(salPtsCount,9);
%for each point extract color moments
for j=1:length(salIdx)
    %compute neighbour indices
    [r,c]=ind2sub([ThSize, ThSize], salIdx(j));
    r=[(r-1)*ones(1,3), r*ones(1,3), (r+1)*ones(1,3)];
    c=[c-1, c, c+1];
    c=[c, c, c];
    r(r>256)=r(r>256)-256;
    r(r<1)=r(r<1)+256;
    c(c>256)=c(c>256)-256;
    c(c<1)=c(c<1)+256;
    ind=sub2ind([ThSize, ThSize], r, c);
    for k=1:3
        colorPlane=img(:,:,k);
        features(j,3*(k-1)+1) = mean(colorPlane(ind));
        features(j,3*(k-1)+2) = var(colorPlane(ind));
        features(j,3*(k-1)+3) = skewness(colorPlane(ind));
    end;
end;
scores=zeros(5, filesCount);

disp('scoring...');
for i=1:filesCount
    disp([num2str(i), ' of ', num2str(filesCount)]);
    for j=1:salPtsCount
        dist=features(j,:)-squeeze(Index.features(i,j,:));
        dist=dist*dist';
        scores(1,i)=scores(1,i)+dist;
    end;
    [dist, unused, m,v]=MapDistance(salIdx, Index.salP(i,:), ThSize);
    scores(2,i)=dist;
    scores(3,i)=unused;
    scores(4,i)=m;
    scores(5,i)=v;
end;

%scale every score and compose them
singleScores=zeros(1, filesCount);
for i=1:length(Weights)
    scores(i,:)=scores(i, :)/max(scores(i, :));
    singleScores=Weights(i)*scores(i, :);
end;
```

Тестирование алгоритма

Для тестирования данного алгоритма была составлена база картинок в количестве 213 штук, которая состояла из кластеров изображений — групп изображений, похожих друг на друга, каждый кластер представлял разный класс изображений. В качестве запроса бралась одна картинка из базы. В силу небольшого размера базы каждый запрос содержит некоторое количество (часто до половины) не релевантных изображений.



Рисунок 9

Рисунок 9 иллюстрирует запрос к базе данных. В качестве запроса выступило изображение с весьма четко выделяемой формой. Алгоритм выбрал и правильно ранжировал все изображения данного кластера. Изображения искажены в связи с небольшим размером эскизов (thumbnail), сохраненных в базе данных и принудительным растягиванием в MATLAB на этапе построения графика.

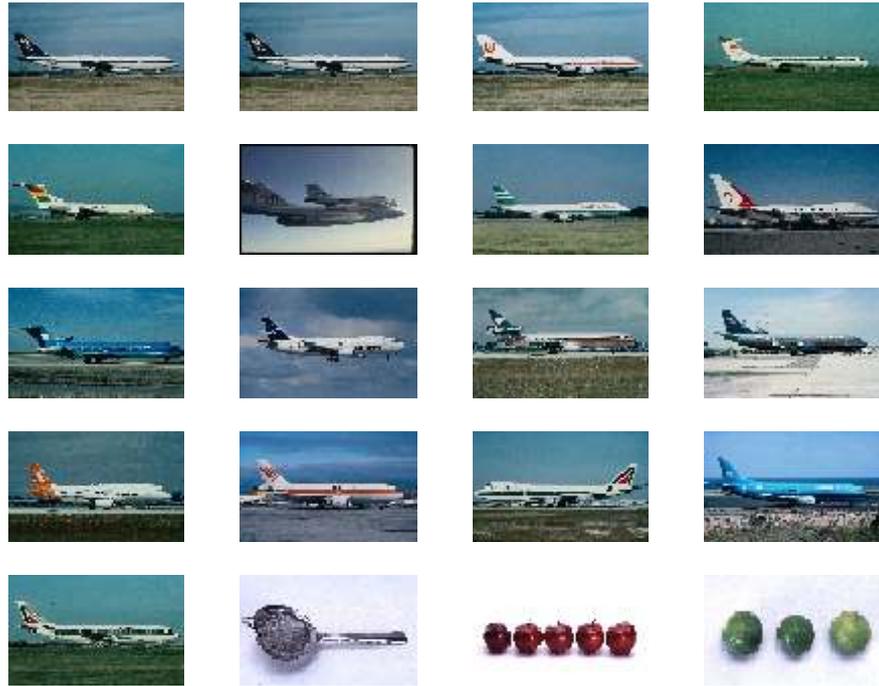


Рисунок 10

Рисунок 10 иллюстрирует запрос с достаточно хорошо выделяемой формой, но уже с присутствием текстурированного фона. В данном случае алгоритм также правильно выбрал все изображения кластера, правильно их ранжировав.

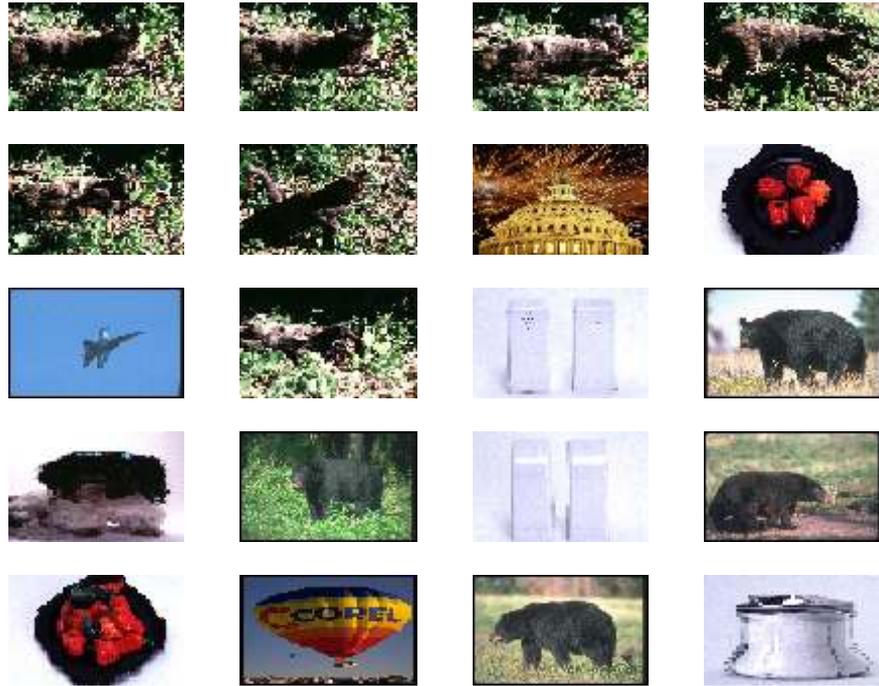


Рисунок 11

Рисунок 11 иллюстрирует запрос очень трудно выделяемой формой объекта. Тем не менее, алгоритм нашел почти всех представителей данного кластера изображений (не хватает одного), с небольшими погрешностями в ранжировании.

Дополнительные примеры запросов содержатся в приложении 6.

Заключение

Достигнутые результаты

В данной работе проанализирована идея извлечения «особых» точек, разработаны алгоритмы извлечения двух характеристик изображения на ее основе, цветовой и особенности формы. С использованием данных характеристик построен новый алгоритм поиска, использующий кратномасштабное представление изображения и учитывающий форму изображения.

Возможные направления дальнейших исследований

Для дальнейших исследований представляются перспективным следующие направления:

- Иное построение индекса, возможно аналогичное алгоритму университета Вашингтона, либо описываемому в [13].
- Принципиально иная визуализация результата, подобно методике, описанной в [14].
- Оптимизация алгоритма с целью увеличения скорости расчетов.
- Разработка и применение более объективной методики оценки релевантности результатов поиска [15,16].
- Более строгое исследование влияния начального уровня и вейвлета на качество особых точек.
- Рассмотрение текстурных особенностей.
- Расширенное исследование влияния цветового пространства на качество «особых» точек [17].

Использованные источники информации

1. Digital Libraries, Value, and Productivity. Go Wiederhold
<http://www.cacr.caltech.edu/~slombey/work/KB/libs/>
2. Content-Based Image Retrieval Systems. Peter L. Stanchev.
3. Content-Based Image Retrieval Systems: A Survey. Remco C. Veltkamp
4. State-of-the-Art in Shape Matching. Remco C. Veltkamp, Michiel Hagedoorn
5. Fast Multiresolution Image Querying. C.E. Jacobs, A. Finkelstein, D.H. Salesin.
6. Вейвлеты в компьютерной графике. Э. Столниц, Т. ДеРоуз, Д. Салезин
7. The QBIC project: Querying images by content using color, texture and shape. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin.
<http://www.jpeg.org/jpeg2000/>
8. An Introduction to Wavelets. Charles K. Chui.
9. Content-Based Image Retrieval Using Wavelet-based Salient Points. Q. Tian, N. Sebe, M.S. Lew, E. Loupiaz, T.S. Huang
10. Local grayvalue invariant for image retrieval. C. Schmid, R. Mohr
11. An Efficient Color Representation for Image Retrieval. Y. Deng, B. S. Manjunath, C. Kenney, Michael S. Moore, Hyundoo Shin.
12. Scalable Color Image Indexing And Retrieval Using Vector Wavelets. Elif Albuz, Erturk Kocalar, Ashfaq A. Khokhar
13. Spatial Visualization for Content-Based Image Retrieval. Baback Moghaddam, Qi Tian, Thomas Huang
14. Multi-class relevance feedback content-based image retrieval. Jing Peng
15. Relevance Feedback Techniques in Interactive Content-Based Image Retrieval. Yong Rui, Thomas S. Huang, and Sharad Mehrotra
16. An Efficient Color Representation for Image Retrieval. Yining Deng, B. S. Manjunath, Charles Kenney, Michael S. Moore and Hyundoo Shin
- 17.

Приложение 1. Системы поиска

Список систем контекстного поиска изображений

1. ADL (Alexandria Digital Library). Разработчик — University of California, Santa Barbara.
2. Amore (Advanced Multimedia Oriented Retrieval Engine). Разработчик C & C Research Laboratories NEC USA, Inc.
3. Berkeley Digital Library Project. Разработчик University of California, Berkeley.
4. Blobworld. Разработчик Computer Science Division, University of California, Berkeley.
5. CANDID (Comparison Algorithm for Navigating Digital Image Databases). Разработчик Computer Research and Applications Group, Los Alamos National Laboratory, USA
6. C-bird (Content-Based Image Retrieval from Digital libraries). Разработчик School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada.
7. Chabot. Разработчик Department of Computer Science, University of California, Berkeley, CA, USA.
8. CBVQ (Content-Based Visual Query). Разработчик Image and Advanced Television Lab, Columbia University, NY.
9. DrawSearch. Разработчик Department of Electrical and Electronic Engineering, Technical University of Bari, Italy.
10. Excalibur Visual RetrievalWare. Разработчик Excalibur Technologies.
11. FIR (Formula Image Retrieval). Разработчик Fraunhofer Institute (по заказу Computer Graphics, Darmstadt, Germany)
12. FOCUS (Fast Object Color-based Query System). Разработчик Department of Computer Science, University of Massachusetts, Amherst, MA.
13. ImageFinder. Разработчик Attrasoftware Inc.
14. ImageMiner. Разработчик Technologie-Zentrum Informatik, University of Bremen, Germany.
15. ImageRETRO (Image RETrieval by Reduction and Overview). Разработчик Department of Computer Science, University of Amsterdam, The Netherlands.
16. ImageRover. Разработчик Department of Computer Science, Boston University, MA.
17. ImageScape. Разработчик Department of Computer Science, Leiden University, The Netherlands.
18. MARS (Multimedia Analysis and Retrieval System). Разработчик Department of Computer Science, University of Illinois at Urbana-Champaign.
19. MetaSEEk. Разработчик Image and Advanced Television Lab, Columbia University, NY, USA.
20. Photobook. Разработчик Vision and Modeling Group, MIT Media Laboratory, Cambridge, MA.
21. PicToSeek. Разработчик Department of Computer Science, University of Amsterdam, The Netherlands
22. QBIC (Query By Image Content). Разработчик IBM Almaden Research Center, San Jose, CA
23. VisualSEEk. Разработчик Image and Advanced Television Lab, Columbia University, NY.
24. WebSEEk. Разработчик Image and Advanced Television Lab, Columbia University, NY

Приложение 2. Основы теории вейвлетов

Вейвлет-преобразование сигнала

Вейвлеты (wavelets-короткая волна) — функции определенной формы, локализованные по оси аргументов (независимых переменных), инвариантные к сдвигу и линейные к операции масштабирования (сжатия/растяжения). Они создаются с помощью специальных базисных функций, которые определяют их вид и свойства. По локализации во временном и частотном представлении вейвлеты занимают промежуточное положение между гармоническими (синусоидальными) функциями, локализованными по частоте, и функцией Дирака, локализованной во времени. Впервые этот термин использовали Гроссман и Морле (A.Grossmann, J.Morlet) при анализе свойств сейсмических и акустических сигналов. В отечественной литературе их называют также всплесками.

Основная область применения вейвлет-преобразования — анализ и обработка сигналов и функций, нестационарных во времени или неоднородных в пространстве, когда результаты анализа должны содержать не только общую частотную характеристику сигнала (распределение энергии сигнала по частотным составляющим), но и сведения об определенных локальных координатах, на которых себя проявляют те или иные группы частотных составляющих или на которых происходят быстрые изменения частотных составляющих сигнала. По сравнению с разложением сигналов на ряды Фурье, вейвлеты способны с гораздо более высокой точностью представлять локальные особенности сигналов, вплоть до разрывов 1-го рода (скачков). В отличие от преобразований Фурье, вейвлет-преобразование одномерных сигналов обеспечивает двумерную развертку, при которой частота и координата рассматриваются как независимые переменные, что дает возможность анализа сигналов сразу в двух пространствах.

Семейства вейвлетов во временной или частотной области используются для представления сигналов и функций в виде суперпозиций вейвлетов на разных уровнях декомпозиции (разложения) сигналов.

Основные принципы вейвлет-преобразования.

Гармонические базисные функции преобразования Фурье предельно локализованы в частотной области и не локализованы во временной (определены во всем временном интервале от $-\infty$ до ∞). Их противоположностью являются импульсные базисные функции типа импульсов Кронекера, которые предельно локализованы во временной области и "размыты" по всему частотному диапазону. Вейвлеты по локализации в этих двух представлениях можно рассматривать как функции, занимающие промежуточное положение между гармоническими и импульсными функциями.

Одна из главных идей вейвлет-представления сигналов заключается в разделении функций приближения к сигналу на две группы: аппроксимирующую — грубую, с достаточно медленной временной динамикой изменений, и детализирующую — с локальной и быстрой динамикой изменений на фоне плавной динамики, с последующим их дроблением и детализацией на других уровнях декомпозиции сигналов. Это возможно как во временной, так и в частотной областях.

Непрерывное вейвлет-преобразование.

В общей основе вейвлет-преобразования лежит использование двух непрерывных взаимозависимых и интегрируемых по оси t функций:

- Вейвлет-функции $\psi(t)$, как функции времени с нулевым значением интеграла и частотным фурье-образом $\Psi(\omega)$. Этой функцией, которую кратко обычно и называют вейвлетом, выделяются детали сигнала и его локальные особенности.
- Масштабирующей функции $\phi(t)$, как временной функции с единичным

значением интеграла, с помощью которой выполняется грубое приближение (аппроксимация) сигнала.

Базисными ψ -функциями пространства $L^2(\mathbb{R})$ могут быть самые различные функции, ограниченные по времени и местоположению на временной оси, и имеющие спектральный образ, в определенной степени локализованный на частотной оси. Как и для рядов Фурье, ортонормированное базисное пространство целесообразно конструировать из одной исходной базовой функции $\psi(t)$, норма которой должна быть равна 1. Для перекрытия локальной временной функцией вейвлета всей временной оси $\mathbb{R}(-\infty, \infty)$ пространства $L^2(\mathbb{R})$ используется операция сдвига (смещения по временной оси): $\psi_b(t) = \psi(t-b)$. Для перекрытия всего частотного диапазона пространства $L^2(\mathbb{R})$ используется операция временного масштабирования вейвлета: $\psi_a(t) = |a|^{-1/2} \psi(t/a)$. Это обозначает, что если временной образ вейвлета будет сужаться (при уменьшении значения коэффициента a), то его "средняя частота" будет повышаться, а его частотный образ перемещаться на более высокие частоты. С использованием этих операций базис функционального пространства $L^2(\mathbb{R})$ образуется путем масштабных преобразований и сдвигов вейвлета $\psi(t)$ с непрерывными значениями базисных параметров — масштабного коэффициента a и параметра сдвига b :

$$\psi_{ab}(t) = |a|^{-1/2} \psi[(t-b)/a], \quad a, b \in \mathbb{R}, \quad \psi(t) \in L^2(\mathbb{R}).$$

Нетрудно убедиться, что нормы вейвлетов $\psi_{ab}(t)$ равны норме $\psi(t)$, что обеспечивает нормировочный множитель $|a|^{-1/2}$, и если базовый вейвлет $\psi(t)$ нормировать на 1, то все функции вейвлетов также будут нормированными. Если при этом выполняется требование ортогональности функций, т.е.:

$$(\psi_{ik}(t), \psi_{nm}(t)) = \delta_{in} \cdot \delta_{km},$$

то функции $\psi_{ab}(t)$ будут представлять собой ортонормированный базис пространства $L^2(\mathbb{R})$, и прямое интегральное вейвлет-преобразование сигнала выполняется по аналогии с преобразованием Фурье:

$$c(a,b) = \int_{-\infty}^{\infty} s(t) \psi_{ab}(t) dt,$$

и представляет собой интегральное значение скалярного произведения сигнала на вейвлет-функцию заданного типа. Интегрирование обычно выполняется по пространству $L^2(\mathbb{R})$.

Так как форма базисных функций $\psi_{ab}(t)$ зафиксирована, то вся информация о сигнале переносится на значения коэффициентов $c(a,b)$. Точность обратного интегрального вейвлет-преобразования зависит от выбора базисного вейвлета и способа построения базиса, т.е. от значений базисных параметров a, b . Строго теоретически вейвлет может считаться базисной функцией $L^2(\mathbb{R})$ только в случае его ортонормированности. Для практических целей часто бывает вполне достаточно устойчивости и ортогональности системы разложения функций. Под устойчивостью в данном случае понимается достаточно точная реконструкция любых произвольных сигналов. Для ортонормированных вейвлетов обратное вейвлет-преобразование записывается с помощью того же базиса, что и прямое:

$$s(t) = C_{\psi}^{-1} \int_{\mathbb{R}} a^{-2} c(a,b) \psi_{ab}(t) da db.$$

где C_{ψ} - нормализующий коэффициент:

$$C_{\psi} = \int_{-\infty}^{\infty} |\Psi(\omega)|^2 |\omega|^{-1} d\omega < \infty.$$

Условие конечности C_{ψ} ограничивает класс функций, которые можно использовать в качестве вейвлетов. В частности, при $\omega=0$ значение $\Psi(\omega)$ также должно быть равно нулю, а, следовательно, функция $\psi(t)$ должна иметь нулевое значение нулевого момента (интеграл функции по \mathbb{R} должен быть нулевым). Однако это означает также, что не для всех сигналов возможна их точная реконструкция вейвлетом $\psi_{ab}(t)$. Условия точной

реконструкции сигналов будут рассмотрены при описании кратномасштабного анализа. Кроме того, далеко не все типы вейвлетов могут гарантировать реконструкцию сигналов, как таковую. Однако применение и таких вейвлетов может быть полезно для анализа особенностей сигналов, как дополнительного метода к другим методам анализа и обработки данных.

Кратномасштабное представление функций

При анализе сигналов часто полезно представить сигнал в виде совокупности его последовательных приближений. Например, при передаче изображения можно сначала передать грубую его версию, а затем последовательно ее уточнять. Такая стратегия передачи имеет выгоды, например, при осуществлении выбора изображений из некоторой базы данных, когда необходимо быстро просмотреть большое количество картинок, сначала быстро показывается большое количество небольших версий грубого разрешения (thumbnail view), затем, после выбора пользователя, необходимые изображения улучшаются.

Теория кратномасштабного анализа базируется на теории функциональных пространств. Под кратномасштабным анализом понимается описание пространства $L^2(R)$ через иерархические вложенные подпространства V_m , которые не пересекаются и объединение которых дает нам в пределе $L^2(R)$, то есть

$$\dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \dots$$

$$\bigcap_{m \in \mathbb{Z}} V_m = \{0\}, \quad \overline{\bigcup_{m \in \mathbb{Z}} V_m} = L^2(R) \quad (1)$$

Далее, эти пространства имеют следующее свойство: для любой функции $f(x) \in V_m$ ее сжатая версия будет принадлежать пространству V_{m-1} ,

$$f(x) \in V_m \Leftrightarrow f(2x) \in V_{m-1} \quad (2)$$

Последнее свойство кратномасштабного анализа — существует такая функция $\phi(x) \in V_0$, что ее сдвиги $\phi_{0,n}(x) = \phi(x-n), n \in \mathbb{Z}$ образуют ортонормированный базис пространства V_0 .

Так как функции $\phi_{0,n}(x)$ образуют ортонормированный базис пространства V_0 , то функции

$$\phi_{m,n}(x) = 2^{-m/2} \phi(2^{-m}x - n) \quad (3)$$

образуют ортонормированный базис пространства V_m . Эти базисные функции называются масштабирующими, так как они создают масштабированные версии функций в $L^2(R)$. Из кратномасштабного анализа, определенного выше, следует, что функция $f(x)$ в $L^2(R)$ может быть представлена множеством последовательных ее приближений $f_m(x)$ в V_m . Другими словами, функция $f(x)$ есть предел аппроксимаций $f_m(x) \in V_m$ при m стремящемся к минус бесконечности:

$$f(x) = \lim_{m \rightarrow -\infty} f_m(x) \quad (4)$$

Отсюда появляется возможность анализа функции или сигнала на различных уровнях разрешения, или масштаба. Переменная m называется масштабным коэффициентом, или уровнем анализа. Если значение m велико, то функция в V_m есть грубая аппроксимация $f(x)$, и детали отсутствуют. При малых значениях m имеет место точная аппроксимация. Из определения кратко-масштабного анализа следует, что все функции в V_m могут быть представлены как линейная комбинация масштабирующих функций. В действительности, $f_m(x)$ есть ортогональная проекция $f(x)$ на V_m ,

$$f_m(x) = \sum_n \langle \phi_{m,n}(x), f(x) \rangle \phi_{m,n}(x) = \sum_n c_{m,n} \phi_{m,n}(x) \quad (5)$$

Т.к. $\phi(x) = \phi_{0,0}(x) \in V_0 \subset V_{-1}$, можно записать

$$\phi_{0,0}(x) = 2^{1/2} \sum_n h_n \phi_{-1,n}(x) = 2 \sum_n h_n \phi(2x - n), \quad (6)$$

где h_n — некоторая последовательность. Последнее равенство является одним из основных в теории вейвлет-анализа и имеет различные названия в литературе. Будем называть его далее масштабирующим уравнением.

Функция $\phi(x)$ и последовательность h_n тесно связаны между собой. Выведем соответствующие отношения. Из масштабирующего уравнения можно получить

$$\phi_{m+1,k}(x) = 2^{1/2} \sum_p h_p \phi_{m,p-2k}(x) = 2^{(-m+1)/2} \sum_p h_p \phi(2^{-m}x - (p-2k)) \quad (7)$$

Выполним операцию скалярного произведения $\phi_{m,n-2k}(x)$ с обеих сторон полученного выше равенства:

$$\begin{aligned} \langle \phi_{m+1,k}(x), \phi_{m,n-2k}(x) \rangle &= \langle 2^{1/2} \sum_p h_p \phi_{m,p-2k}(x) \phi_{m,n-2k}(x) \rangle = \\ &= 2^{1/2} \sum_p h_p \langle \phi_{m,p-2k}(x) \phi_{m,n-2k}(x) \rangle = 2^{1/2} h_n \end{aligned} \quad (8)$$

Отметим, что это равенство выполняется для любого m . Далее, если переписать (6) в частотной области, можно получить

$$\Phi(\omega) = H\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right) \quad (9)$$

При рекурсивном повторении формулы (9) получается выражение

$$\Phi(\omega) = \prod_{m=1}^{\infty} H\left(\frac{\omega}{2^m}\right) \quad (10)$$

Итак, последовательность h_n тесно связана с масштабирующей функцией. Кроме того, из концепции кратномасштабного анализа вытекают следующие свойства. Во-первых, интегрируя масштабирующее уравнение по всей числовой оси x , можно получить

$$\sum_n h_n = 1 \quad (11)$$

так как для построения кратномасштабного анализа среднее значение функции $\phi(x)$ не должно быть равно нулю. Данное равенство эквивалентно тому, что $H(0)=1$. Во-вторых, в силу ортонормальности базисных функций

$$\delta_{0,k} = \langle \phi_{0,0}(x), \phi_{0,k}(x) \rangle = 2 \sum_n h_n h_{n+2k} \quad (12)$$

Третье свойство последовательности h_n сформулируем в спектральной области. Из записи условия ортонормальности функций $\phi_{n,m}(x)$ в области спектра

$$\sum_k |\Phi(\omega + 2k\pi)|^2 = 1 \quad (13)$$

можно получить следующее выражение:

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 1 \quad (14)$$

Данное равенство эквивалентно тому, что $H(\pi)=0$. перейдем к простейшему примеру множества масштабирующих функций, образующих $L^2(R)$.

Представление функций при помощи вейвлетов

Область $L^2(R)$ построена из множества «колец», которые есть разность между двумя соседними пространствами. Эти разностные пространства обозначаются через W_m и определяются как ортогональные дополнения областей V_m до V_{m-1} :

$$V_{m-1} = V_m \oplus W_m, \quad \bigcap_{m \in \mathbb{Z}} W_m = \{0\}, \quad \bigcap_{m \in \mathbb{Z}} V_m = L^2(R) \quad (15)$$

Пусть $\psi(x) = \psi_{0,0}(x)$ есть базисная функция W_0 . Т.к. $\psi_{0,0}(x) \in W_0 \subset V_{-1}$, можно записать

$$\psi_{0,0}(x) = 2^{1/2} \sum_n g_n \phi_{-1,n}(x) \quad (16)$$

для некоторой последовательности g_n . Определим семейство вейвлет-функций:

$$\psi_{m,n}(x) = 2^{-m/2} \psi(2^{-m}x - n) \quad (17)$$

Функции $\psi_{m,n}(x)$ образуют ортонормированный базис $L^2(R)$. Существуют строгие зависимости между $\psi(x), \phi(x), g_n, h_n$. Вначале получим формулу, аналогичную (10). Перепишем (16) для частотной области:

$$\Psi(\omega) = G\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right) \quad (18)$$

заменяем $\Phi(\omega)$ бесконечным произведением (10) и получим

$$\Psi(\omega) = G\left(\frac{\omega}{2}\right) \prod_{m=2}^{\infty} \left(\frac{\omega}{2^m}\right) \quad (19)$$

Теперь получим выражения, связывающие последовательности g_n и h_n . Так как W_m есть ортогональное дополнение V_m функции $\psi_{0,0}(x)$ и $\phi_{0,0}(x)$ должны быть ортогональны, и из (6) и (16) следует, что

$$0 = \langle \phi_{0,0}, \psi_{0,0} \rangle = 2 \sum_n \sum_p h_n g_p \langle \phi_{-1,n}, \phi_{-1,p} \rangle = 2 \sum_n h_n g_n \quad (20)$$

Легко увидеть, что выбор

$$g_n = (-1)^n h_{-n+2t+1} \quad (21)$$

будет корректен для всех $t \in \mathbb{Z}$. Эквивалент (21) в частотной области представляется в виде

$$G(\omega) = -H(-\omega + \pi) e^{-i\omega(2t+1)}. \quad (22)$$

С учетом этого из (18) получим

$$\Psi(\omega) = e^{-i\omega/2} H\left(-\frac{\omega}{2} + \pi\right) \Phi\left(\frac{\omega}{2}\right), \quad (23)$$

где без потери общности выбрано $t = 0$.

Наконец отметим, что и функция $\psi(x)$ и последовательность g_n имеют нулевое среднее. Этот факт легко проверить, подставляя $\omega = 0$ в (22) и (23) и используя свойство $H(\pi) = 0$:

$$\Psi(\omega) = \int_{-\infty}^{\infty} \psi(x) dx = 0 \quad (24)$$

и

$$G(0) = \sum_n g_n = 0 \quad (25)$$

Определение функций вейвлетов позволяет нам записать любую функцию $f(x) \in L^2(R)$ в виде суммы проекций на $W_j, j \in \mathbb{Z}$:

$$f(x) = \sum_{j=-\infty}^{\infty} e_j(x), \quad (26)$$

где

$$e_j(x) = \sum_k \langle \psi_{j,k}(x), f(x) \rangle \psi_{j,k}(x) \quad (27)$$

Если осуществлять анализ функции вплоть до некоторого масштаба m , то $f(x)$ будет представлена суммой ее грубой аппроксимации $f_m(x) \in V_m$ и множества деталей $e_j(x) \in W_j$:

$$\begin{aligned}
f(x) &= f_m(x) + \sum_{j=-\infty}^m e_j(x) = \sum_n \langle \phi_{m,n}(x), f(x) \rangle \phi_{m,n}(x) + \sum_{j=-\infty}^m \sum_k \langle \psi_{j,k}(x), f(x) \rangle \psi_{j,k}(x) = \\
&= \sum_n c_{m,n}(x) \phi_{m,n}(x) + \sum_{j=-\infty}^m d_{j,k}(x) \psi_{j,k}(x)
\end{aligned} \tag{28}$$

Дискретное вейвлет-преобразование

В нашем случае сигналом является изображение, т.е. дискретный сигнал, поэтому рассмотрим дискретную версию непрерывного преобразования. К сожалению, формулы для вейвлет-преобразования и рядов вейвлетов дискретного времени нельзя получить простой дискретизацией соответствующих формул для непрерывного времени. Также невозможно определить кратномасштабный анализ для дискретных сигналов, так как не существует базисных функций, масштабированные и смещенные версии которых давали бы нам базис для последовательностей бесконечной длины.

Попробуем вывести формулы для DWT из формул кратномасштабного анализа. Пусть имеется некоторая непрерывная функция $f_0(x) \in V_0$. Дискретный сигнал c_n представим как последовательность коэффициентов при масштабирующих функциях, по которым раскладывается $f_0(x)$:

$$f_0(x) = \sum_n c_{0,n} \phi_{0,n}(x) \tag{29}$$

где $c_{0,n} = c_n$. Другими словами, мы интерпретируем наш сигнал как последовательность коэффициентов разложения, полученную в ходе кратномасштабного анализа функции $f_0(x)$. Тогда мы можем вычислить аппроксимации этой функции, принадлежащие пространствам V_1, V_2, \dots . Пространства V_{-1}, V_{-2}, \dots не имеют значения при данной интерпретации.

Согласно концепции кратномасштабного анализа функция $f_0(x)$ декомпозируется на две функции $f_1(x) \in V_1$ и $e_1(x) \in W_1$:

$$f_0(x) = f_1(x) + e_1(x) = \sum_n c_{1,n}(x) \phi_{1,n}(x) + \sum_k d_{1,k}(x) \psi_{1,k}(x) \tag{30}$$

Таким образом, получили две новые последовательности $c_{1,n}$ и $d_{1,n}$. Этот процесс может быть продолжен по $f_1(x)$ и функция $f_0(x)$ (а также и последовательность c_n) будет представлена совокупностью коэффициентов $d_{m,n}$, $m \in Z^+, n \in Z$.

Итак, концепция рядов вейвлетов дискретного времени определена. Однако вычисления пока зависят от непрерывных функций $\psi(x)$ и $\phi(x)$. Поэтому покажем, как вычисления рядов вейвлетов дискретного времени могут быть выполнены с использованием операций только над дискретными сигналами.

С учетом того, что масштабирующая функция образует базис соответствующего пространства, из (8) можно получить

$$\begin{aligned}
c_{1,k} &= \langle \phi_{1,k}(x), f_1(x) \rangle = \langle \phi_{1,k}(x), f_0(x) - e_1(x) \rangle = \langle \phi_{1,k}(x), \sum_n c_{0,n}(x) \phi_{0,n}(x) \rangle = \\
&= \sum_{n \in Z} c_{0,n} \langle \phi_{1,k}(x) \phi_{0,n}(x) \rangle = 2^{1/2} \sum_{n \in Z} c_{0,n} h_{n+2k}
\end{aligned} \tag{31}$$

Так что оказывается возможным итеративное вычисление коэффициентов $c_{j,k}$ и $d_{j,k}$ без непосредственного использования функций $\psi(x)$ и $\phi(x)$. По аналогии с (30) можно записать для произвольного j

$$c_{j,k} = 2^{1/2} \sum_n c_{j-1,n} h_{n+2k}, \tag{32}$$

$$d_{j,k} = 2^{1/2} \sum_n c_{j-1,n} g_{n+2k}, \tag{33}$$

получив, таким образом, полностью дискретный процесс декомпозиции. Последовательности h_n и g_n называются фильтрами. Отметим, что $c_{j,k}$ и $d_{j,k}$ имеют

«половинную» длину по сравнению с $c_{j-1,k}$ (хотя, конечно, на данном этапе все последовательности бесконечны). Таким образом, не вводится избыточности.

Обратный процесс заключается в получении c_{j-1} из c_j и d_j :

$$\begin{aligned}
c_{j-1,n} &= \langle \phi_{j-1,n}(x), f_{j-1}(x) \rangle = \langle \phi_{j-1,n}, f_j(x) + e_j(x) \rangle = \\
&= \langle \phi_{j-1,n}, \sum_k c_{j,k}(x) \phi_{j,k}(x) \rangle + \langle \phi_{j-1,n}, \sum_k d_{j,k}(x) \psi_{j,k}(x) \rangle = \\
&= \sum_k c_{j,k} \langle \phi_{j-1,n}(x), \phi_{j,k}(x) \rangle + \sum_k d_{j,k} \langle \phi_{j-1,n}(x), \psi_{j,k}(x) \rangle = \\
&= 2^{1/2} \sum_k c_{j,k} h_{n+2k} + 2^{1/2} \sum_k d_{j,k} g_{n+2k}
\end{aligned} \tag{34}$$

Отметим, что в данном случае суммирование производится по другим переменным по сравнению с формулами (31) и (32). Длина последовательности c_{j-1} вдвое больше длины последовательности c_j или d_j .

Подставляя (31) и (32) в (33), получаем следующие ограничения на фильтры h_n и g_n :

$$2 \sum_k (h_{n+2k} h_{p+2k} + g_{n+2k} g_{p+2k}) = \delta_{n,p} \tag{35}$$

$$2 \sum_k h_{n+2k} h_{p+2k} + 2 \sum_k g_{n+2k} g_{p+2k} = \delta_{k,p} \tag{36}$$

$$2 \sum_k h_{n+2k} h_{p+2k} = 0 \tag{37}$$

Выражение (34) для временной области эквивалентно выражениям (14) и (22) для частотной. Равенства (35) и (36) уже появлялись ранее, но в менее общей форме ((12) и (20), соответственно).

На практике рядов вейвлетов дискретного времени должны применяться к сигналам конечной длины. Таким образом, его необходимо модифицировать, чтобы из сигнала какой-то длины получать последовательность коэффициентов той же длины. Получившееся преобразование называется дискретное вейвлет-преобразование.

Вейвлет-преобразование для двумерного случая

При работе с графикой необходимо обрабатывать двумерные массивы данных. Многомасштабный анализ должен строиться в пространстве функций двух переменных. Существует несколько способов обобщить изложенные ранее конструкции на функции нескольких (двух) переменных.

Самый простой и широко распространенный путь — тензорное произведение одномерных вейвлет-функций. В качестве двумерной масштабирующей функции берется

$$\Phi(t,s) = \varphi(t)\varphi(s).$$

Вместо одного вейвлета возникает три:

$$\Psi_{LH}(t,s) = \varphi(t)\psi(s), \Psi_{HL}(t,s) = \psi(t)\varphi(s), \Psi_{HH}(t,s) = \psi(t)\psi(s).$$

(L означает низкую частоту, H — высокую частоту). Пространства V_j порождаются сдвигами масштабирующей функции на одном и том же масштабе:

$$V_j = \{2^j \Phi(2^j t - k, 2^j s - m)\},$$

пространства деталей имеют вид:

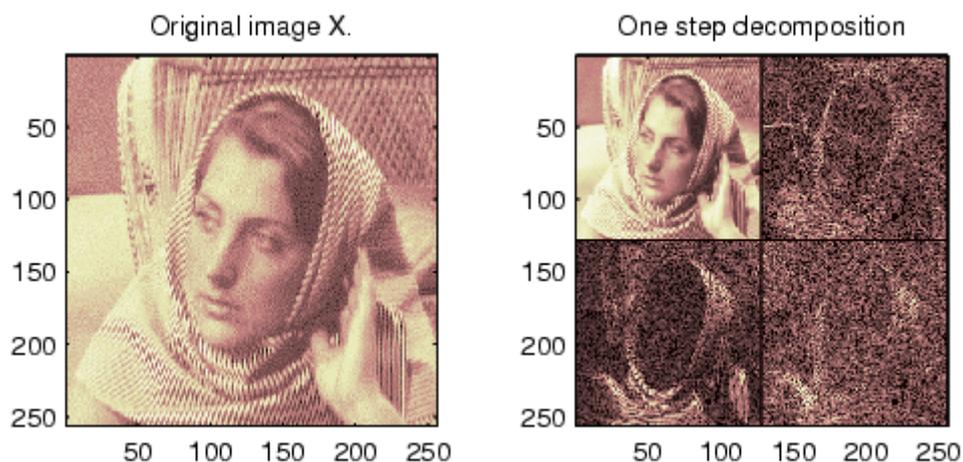
$$W^{LH}_j = \{2^j \Psi_{LH}(2^j t - k, 2^j s - m)\}, W^{HL}_j = \{2^j \Psi_{HL}(2^j t - k, 2^j s - m)\},$$

$$W^{HH}_j = \{2^j \Psi_{HH}(2^j t - k, 2^j s - m)\}$$

Соответствующие проекции находятся применением фильтров (в очевидных обозначениях):

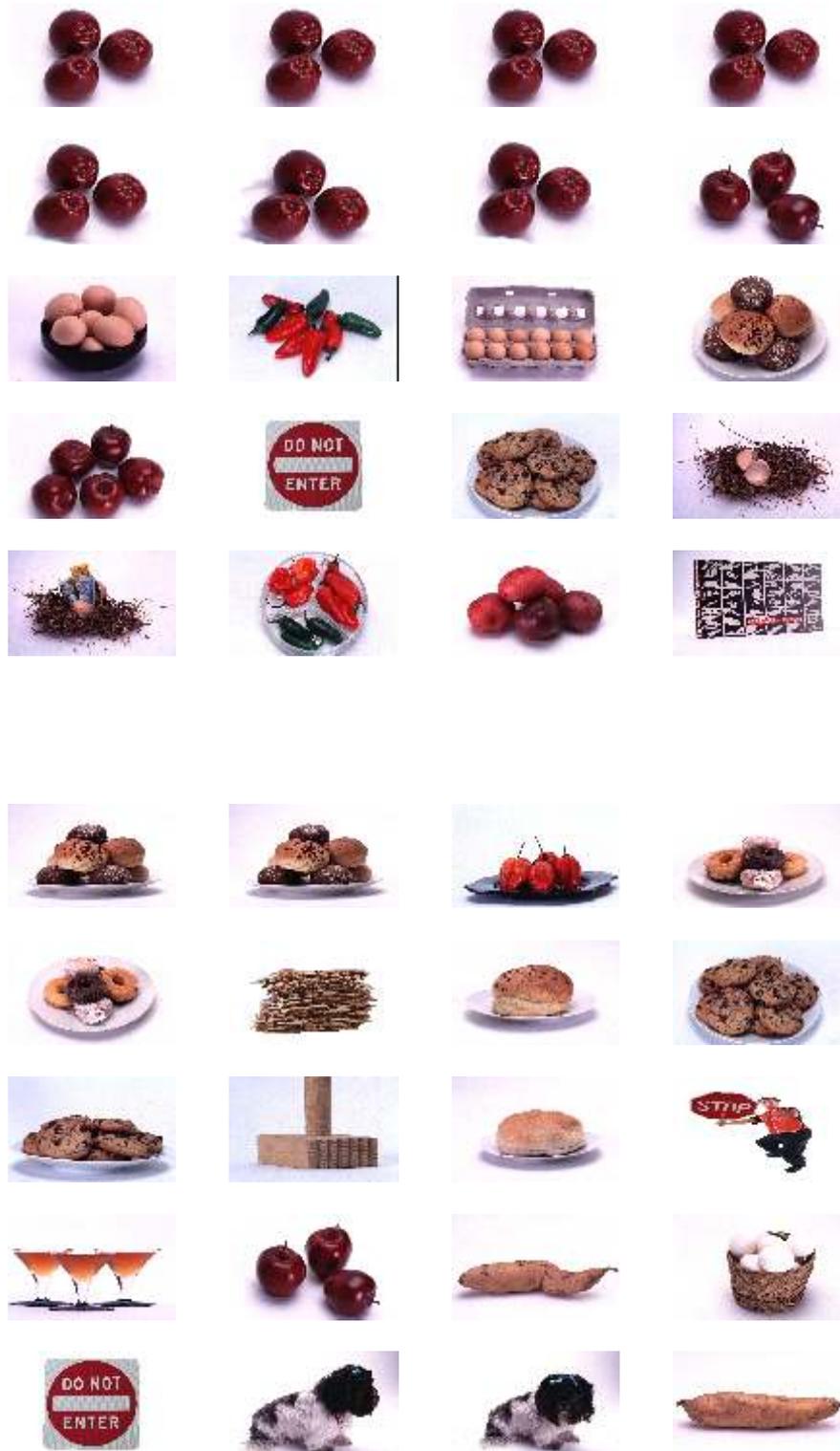
$$x \mapsto \{H_t \otimes H_s(x), H_t \otimes G_s(x), G_t \otimes H_s(x), G_t \otimes G_s(x)\},$$

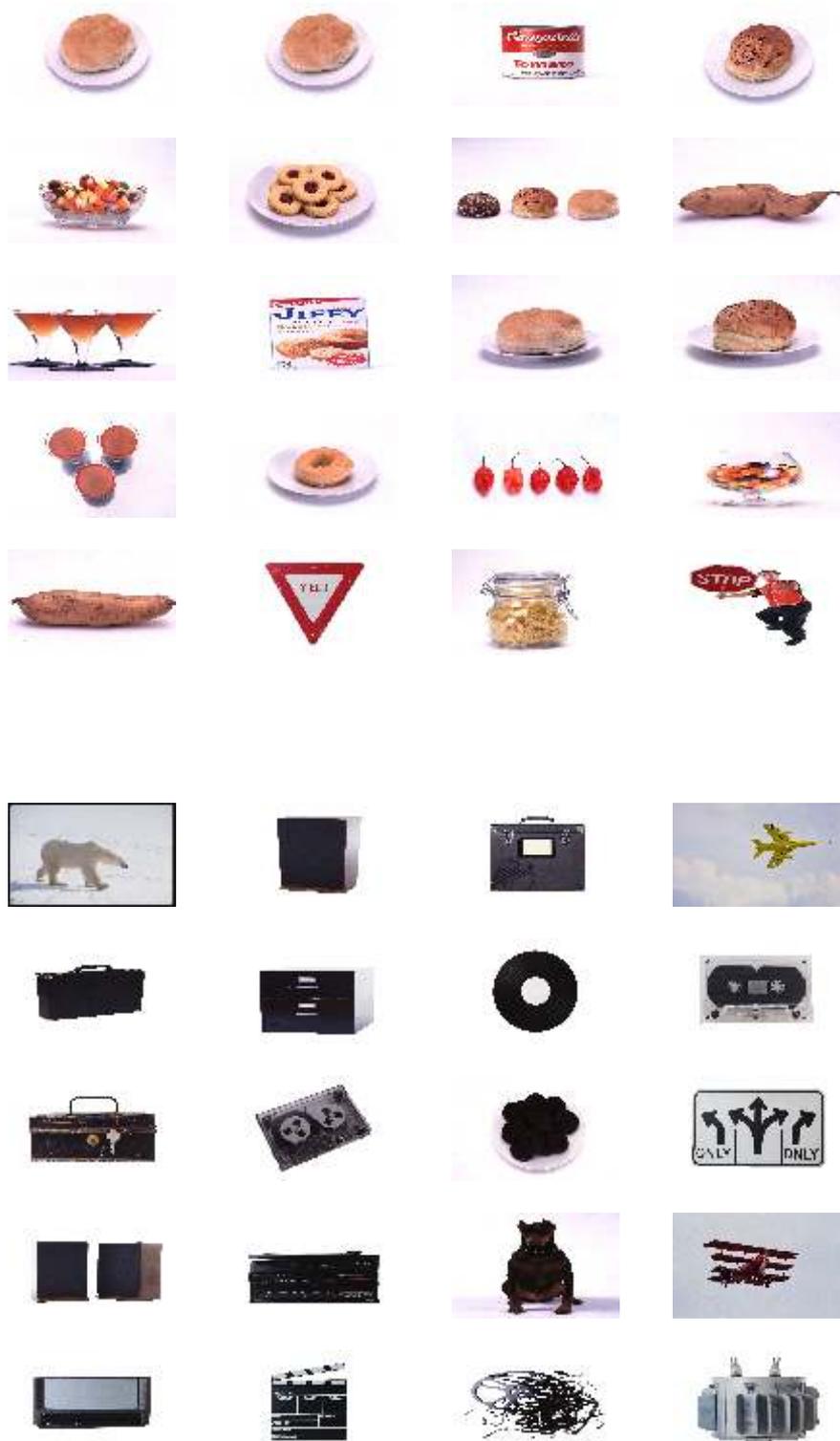
где x — двумерный сигнал. Классическая схема Маллата предполагает рекурсивное применение той же процедуры к низкочастотной составляющей. На рисунке показан результат одного шага этого процесса:



На рисунке более ярким цветом обозначены коэффициенты большей амплитуды. Четко видно, что их положение указывает на резкие перепады яркости. Такие перепады являются наиболее информативными при беглом просмотре любого изображения. Вейвлет-представление позволяет их локализовать путем последовательного уточнения, начиная с более крупных масштабов. Кроме того, коэффициенты проекции на различные пространства деталей отвечают за перепады яркости различной ориентации: например, если фильтр высоких частот применялся к строчкам, то в соответствующем квадранте ярче выделены вертикальные перепады.

Приложение 3. Примеры запросов









Приложение 4. Особые точки

Вейвлеты Хаара.

Особые точки, извлеченные по диапазону LH (60 штук)



Особые точки, извлеченные по диапазону HL (60 штук)



Особые точки, извлеченные по диапазону НН (60 штук)



Вейвлеты D4.

Особые точки, извлеченные по диапазону LH (60 штук)



Особые точки, извлеченные по диапазону HL (60 штук)



Особые точки, извлеченные по диапазону HH (60 штук)



Влияние начального уровня

Уровень 2



Уровень 3



Уровень 4

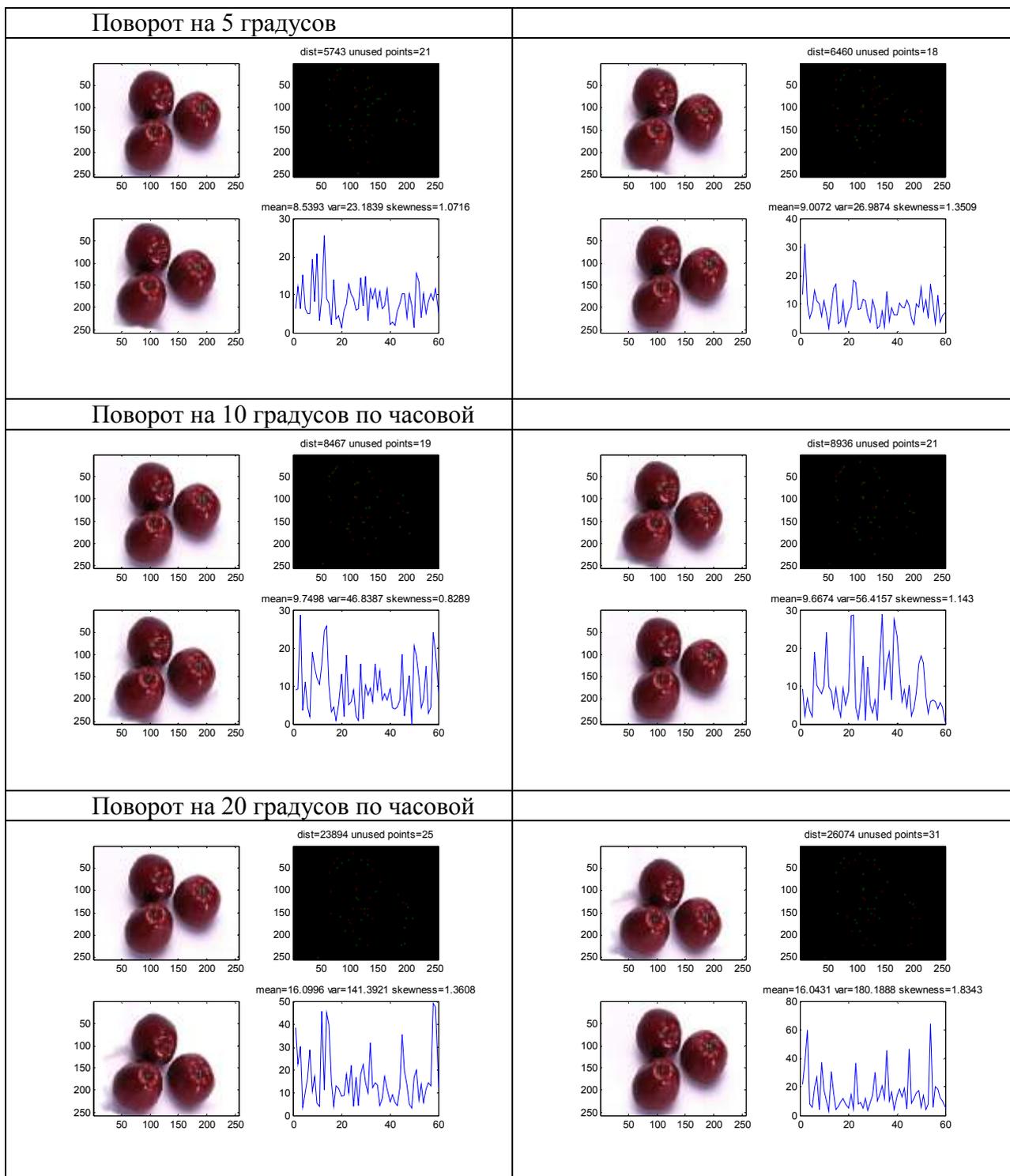


Уровень 5

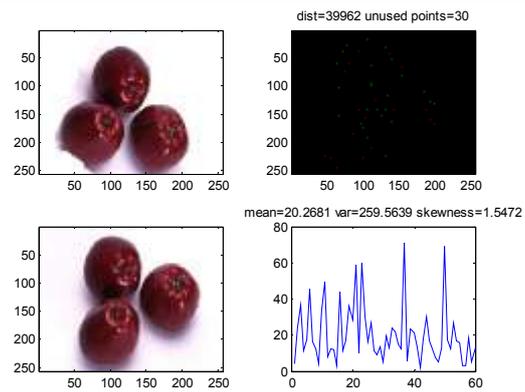
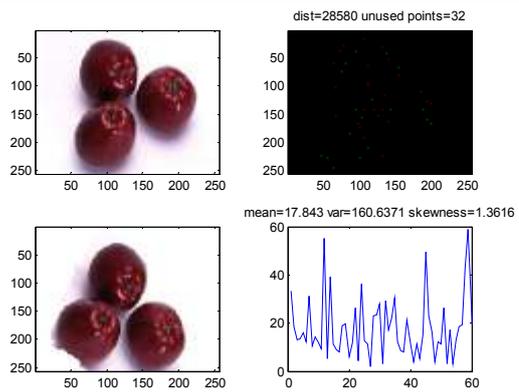


Приложение 5. Особенности формы

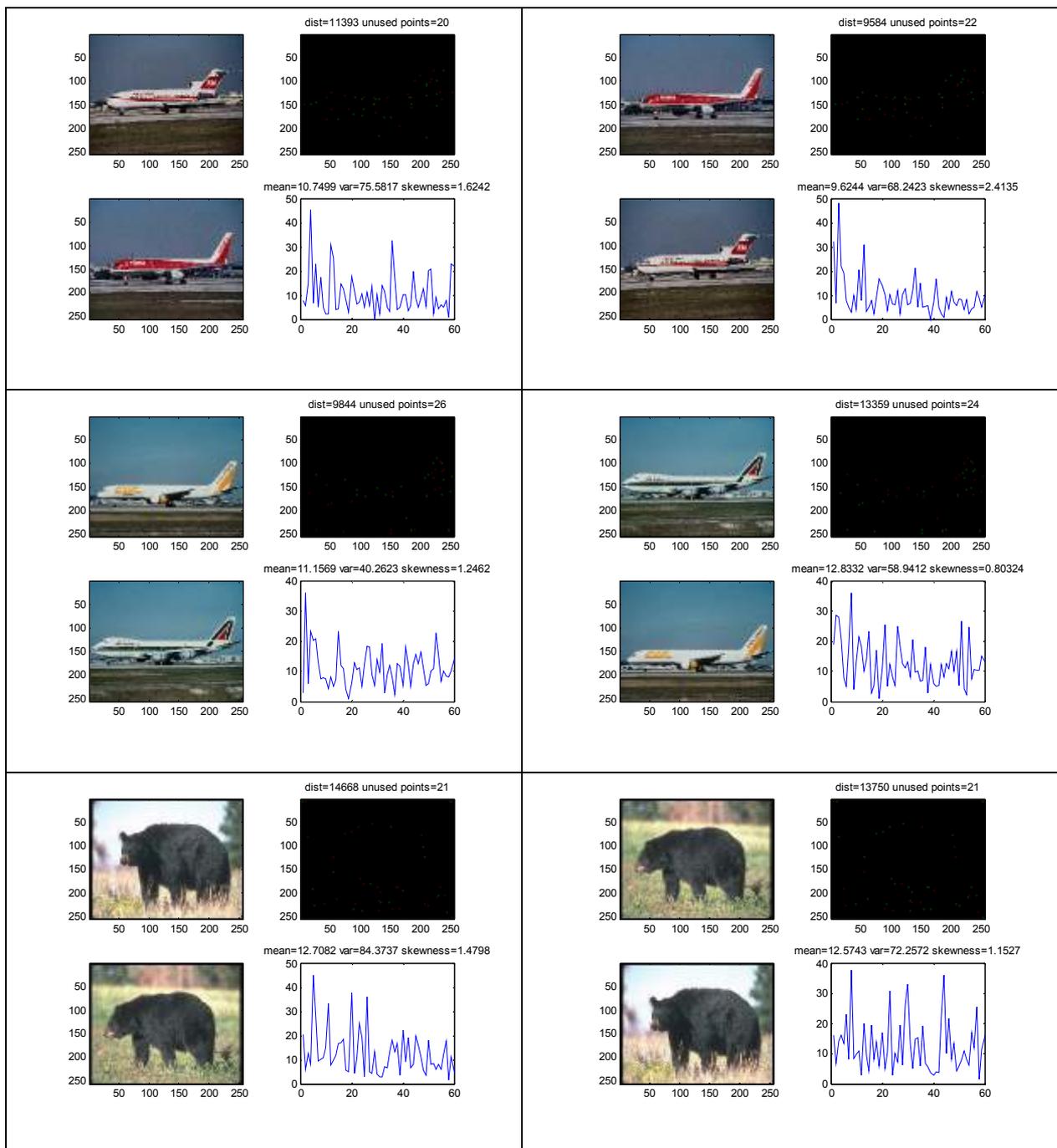
Повороты

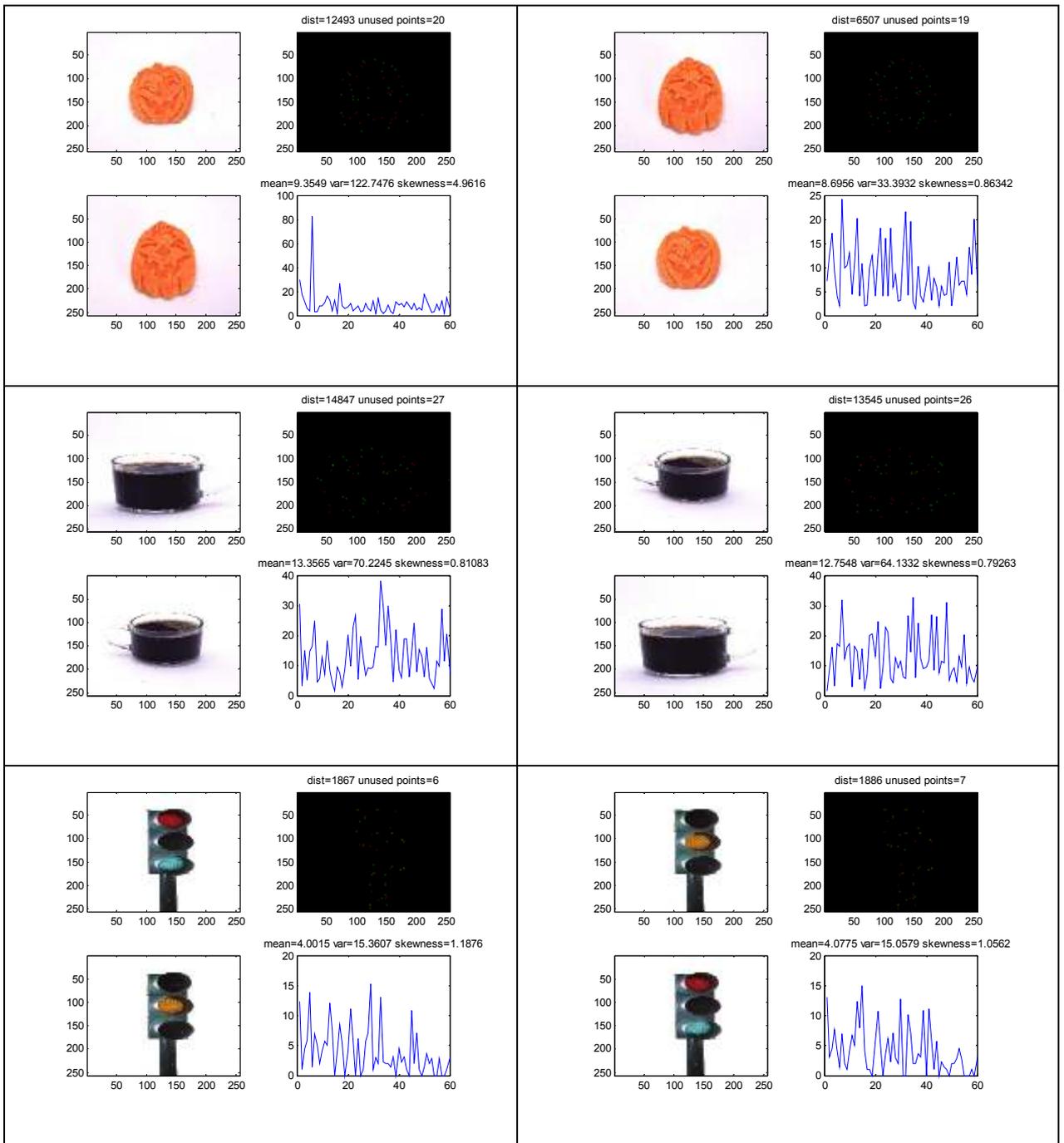


Поворот на 30 градусов

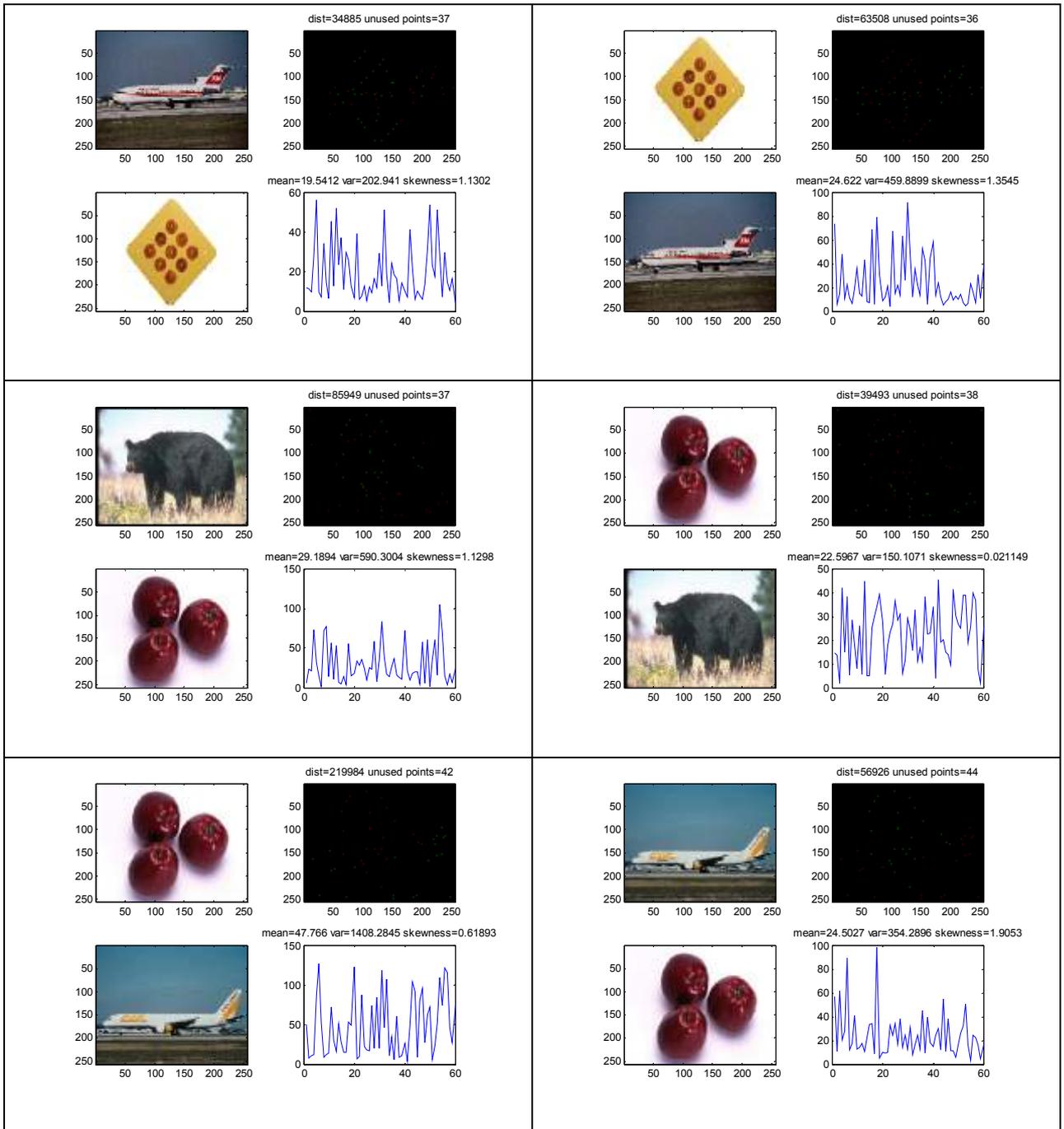


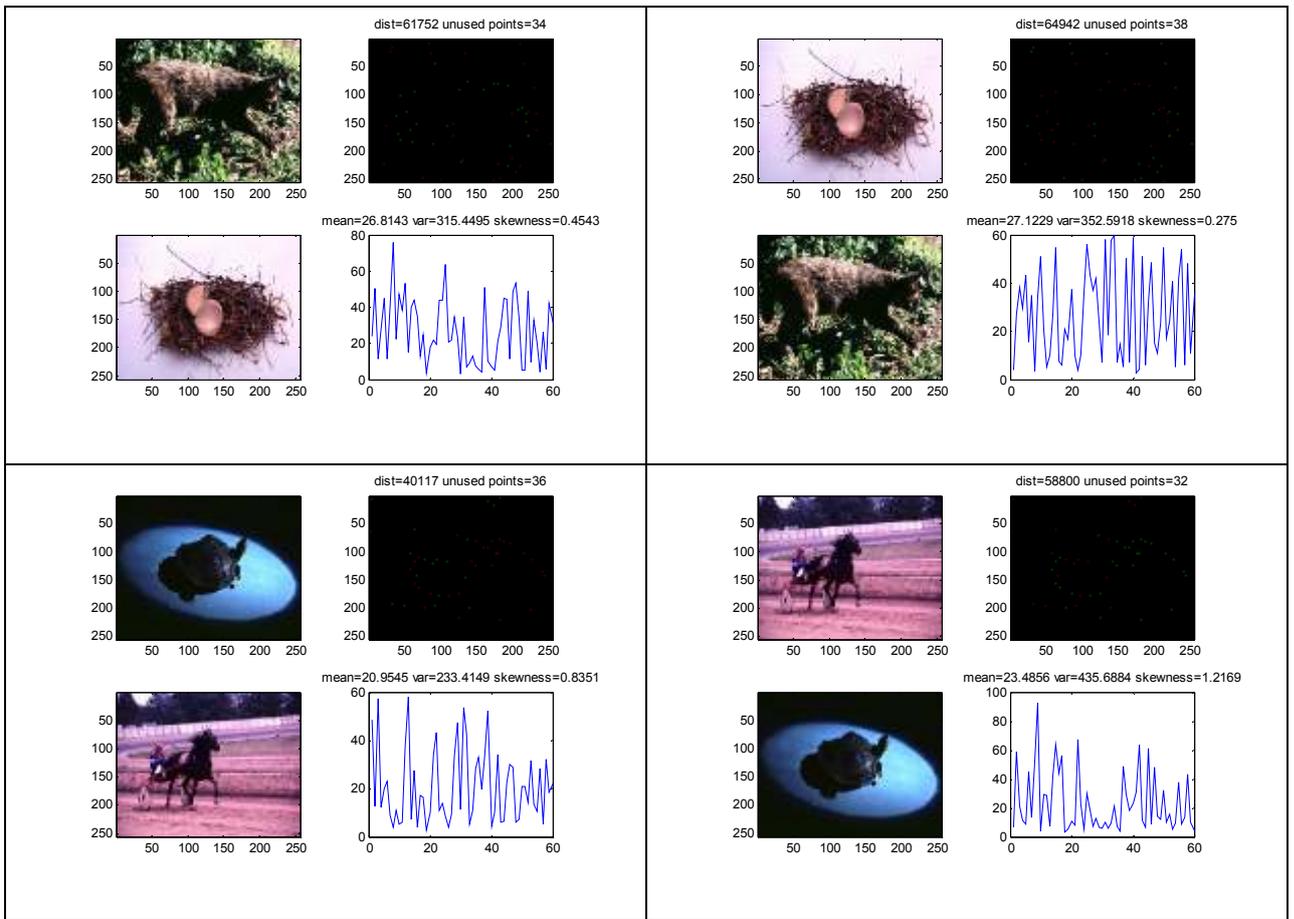
Разные картинки одного класса





Совершенно разные картинки





Приложение 6. Примеры запросов

Приведено несколько примеров запросов к разным классам изображений.

