

Оглавление

Введение.....	4
Глава 1. Обзор существующих методов обнаружения объектов на изображении.....	7
1.1. Структура алгоритмов обнаружения.....	7
1.2. Обзор существующих методов	9
1.2.1. Методы предварительной обработки изображения	9
1.2.2. Методы построения признакового описания.....	13
1.2.3. Методы локализации объектов	17
1.3. Выводы по главе 1	21
Глава 2. Обзор теории активного восприятия: алгоритм и его практическое применение.....	22
2.1. Q -преобразование.....	22
2.2. U -преобразование.....	23
2.3. Алгебра групп	27
2.4. Выводы по главе 2.....	29
Глава 3. Реализация системы обнаружения объекта на изображении на основе теории активного восприятия.....	30
3.1. Предварительная обработка изображения.....	31
3.2. Формирование признакового описания	32
3.3. Принятие решения о локализации объекта на изображении.....	35
3.4. Выводы по главе 3	45
Глава 4. Вычислительный эксперимент.....	46
4.1. Описание эксперимента.....	46
4.2. Описание программного продукта	50
4.3. Анализ результатов эксперимента	53
4.4. Выводы по главе 4.....	66
Заключение	67
Список литературы	68

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Домнина Н.А.			Модель и алгоритмы обнаружения объектов на изображении с использованием локального признакового описания	Лит.	Лист	Листов
Провер.		Гай В.Е.					3	71
Реценз.						НГТУ им. Р.Е. Алексеева		
Н. Контр.								
Утверд.								

Введение

Актуальность темы. В настоящее время наблюдается быстрый рост объемов цифровых графических данных. Каждую секунду в сеть Интернет попадает множество фото и видео с разных концов света. Посещая музей, галерею, другой город или другую страну люди хотят запечатлеть каждый значимый момент этого события. В связи с этим число алгоритмов, применимых к фото- и видеосъемке с каждым днем все растет. Уже никого не удивляет, что при создании фото на экране отображается возраст и пол человека, что при создании группового фото выделяются лица всех людей. Гораздо меньше уделено внимания анализу достопримечательностей и произведений искусства.

Чтобы при наведении камеры отображалось название памятника или скульптуры нужно разработать быстрый алгоритм для использования его в реальном времени, а также устойчивый к повороту, шуму, смене освещения и масштаба. Эффективность этого алгоритма в наибольшей мере зависит от того, как изображение будет представлено для анализа. Здесь очень важно правильно выбрать так называемые «ключевые» точки и способ их описания.

Для реализации такого метода предлагается разработать алгоритм с использованием теории активного восприятия, которая отличается низкой вычислительной сложностью и высокой скоростью работы.

Целью данной работы является исследование методов обнаружения объектов на цифровых изображениях и разработка метода обнаружения с использованием локального признакового описания на основе теории активного восприятия.

Для достижения поставленной цели необходимо решить следующие **задачи**:

- Исследование и анализ существующих методов для решения поставленной задачи
- Создание информационной модели обнаружения объекта на изображении
- Создание алгоритма обнаружения объекта и реализация основывающийся на нем системы
- Проведение вычислительного эксперимента для установления корректности работы реализованной системы

Объектом исследования является цифровое изображение.

Предметом исследования являются модели и алгоритмы обнаружения объектов на изображении.

Для решения поставленных задач были использованы следующие **методы**:

- Метод цифровой обработки изображений
- Метод теории активного восприятия
- Метод формирования локального признакового описания
- Метод проведения вычислительного эксперимента, для которого был разработан программный продукт на языке программирования Python

Положения, выносимые на защиту:

1. Информационная модель обнаружения объектов на изображении, построенная на основе положений теории активного восприятия
2. Алгоритм обнаружения объектов на изображении на основе локального признакового описания объекта
3. Результаты вычислительного эксперимента

Научная новизна разработанного метода обнаружения объектов на изображении заключается в использовании новой комбинации подходов при реализации каждого из этапов решения данной задачи. На этапе подготовки изображения и формирования признакового описания используется теория активного восприятия. На этапе принятия решения используется евклидово расстояние для сравнения векторов признаков изображения-шаблона и изображения, в котором осуществляется поиск объекта, алгоритм k-means и алгоритм выборочного поиска для выделения сегментов интереса.

Теоретическая ценность работы заключается в разработанной информационной модели признакового описания объекта, а также предложенном алгоритме его локализации на изображении.

Практическая ценность работы заключается в созданном программном продукте, реализующем указанный алгоритм, а также в результатах проведенного вычислительного эксперимента.

Апробация данной работы была проведена в ходе защиты отчета по распределенной научно-исследовательской работе, который включал в себя основные положения проведенного исследования и анализ полученных результатов, а также была сделана публикация в Материалах XXV

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		5

Международной научно-технической конференции «Информационные системы и технологии» 2019.

Обоснованность и достоверность проведенного исследования доказывается проведением вычислительного эксперимента, анализом его результатов, и сравнением их с результатами существующих методов решения данной задачи.

Публикации:

1. Н.А.Домнина, В.Е.Гай «Модель и алгоритмы обнаружения объектов на изображении с использованием локального признакового описания». Нижний Новгород: Материалы XXV Международной научно-технической конференции «Информационные системы и технологии», 2019

Объем и структура работы

Выпускная квалификационная работа состоит из введения, четырех глав основной части, заключения и списка литературы. Общий объем работы – 71 страница. Диссертация содержит 43 рисунка, 13 таблиц, 26 формул. Список литературы включает в себя 34 наименования.

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		6

Глава 1. Обзор существующих методов обнаружения объектов на изображении

В настоящее время существует множество алгоритмов для обнаружения объектов на изображении. Некоторые из них известны своей надежностью, некоторые – скоростью работы, какие-то используются для широкого ряда задач, другие же узкоспециализированные. В данной главе будут рассмотрены наиболее распространённые алгоритмы, будут оценены их достоинства и недостатки.

1.1. Структура алгоритмов обнаружения

Существующие методы обнаружения объектов на изображении преимущественно включают в себя три этапа:

- Предварительная обработка изображения
- Формирование признакового описания
- Принятие решения о присутствии объекта на изображении, а также его локализация

Предварительная обработка – это подготовка изображения ко второму, основному, этапу. На этом этапе изображение преобразуется к виду, подходящему для вычисления признаков. В зависимости от метода, здесь может применяться фильтр, подавляющий шум, размытие, увеличение контрастности, нормализация цвета, выделение ключевых областей и т.д. В некоторых методах этот этап опускается.

Второй этап, формирование признакового описания, как следует из названия, нужен для создания итогового описания изображения. По этому описанию метод, применяемый на третьем этапе, будет сравнивать изображение-шаблон и изображение, на котором необходимо найти объект. Признаковое описание, в зависимости от алгоритма, может рассчитываться по всему изображению или по некоторым его ключевым областям, которые позволят однозначно описать объект. Такие ключевые области принято называть ключевыми точками, но по сути это именно области интереса. Признаковое описание – это вектор значений, описывающий объект. Наиболее распространены следующие типы признаков:

- Бинарный
- Номинальный (конечное множество)
- Порядковый (конечное упорядоченное множество)

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		7

- Количественный (множество действительных чисел)

Также в прикладных задачах могут встречаться признаки в виде числовых последовательностей, изображений, текстов, функций, графов, результатов запросов к базе данных и другие.

На этом этапе определяются возможности алгоритма: будет ли он устойчив к повороту изображения, шуму, изменению освещения, масштаба.

Третий этап заключается в принятии решения о локализации искомого объекта на исследуемом изображении. Для этого признаки (их еще называют дескрипторами) двух изображений сопоставляются. Чаще всего здесь используются методы сравнения, основанные на вычислении расстояний всех возможных пар дескрипторов. Выбор наилучшего метода зависит от типа признакового описания. Самые распространённые методы для вычисления расстояния – это Евклидова метрика и расстояние Хемминга. Чем меньше расстояние между двумя точками интереса, тем вероятнее верное сопоставление.

Структура системы обнаружения объектов:

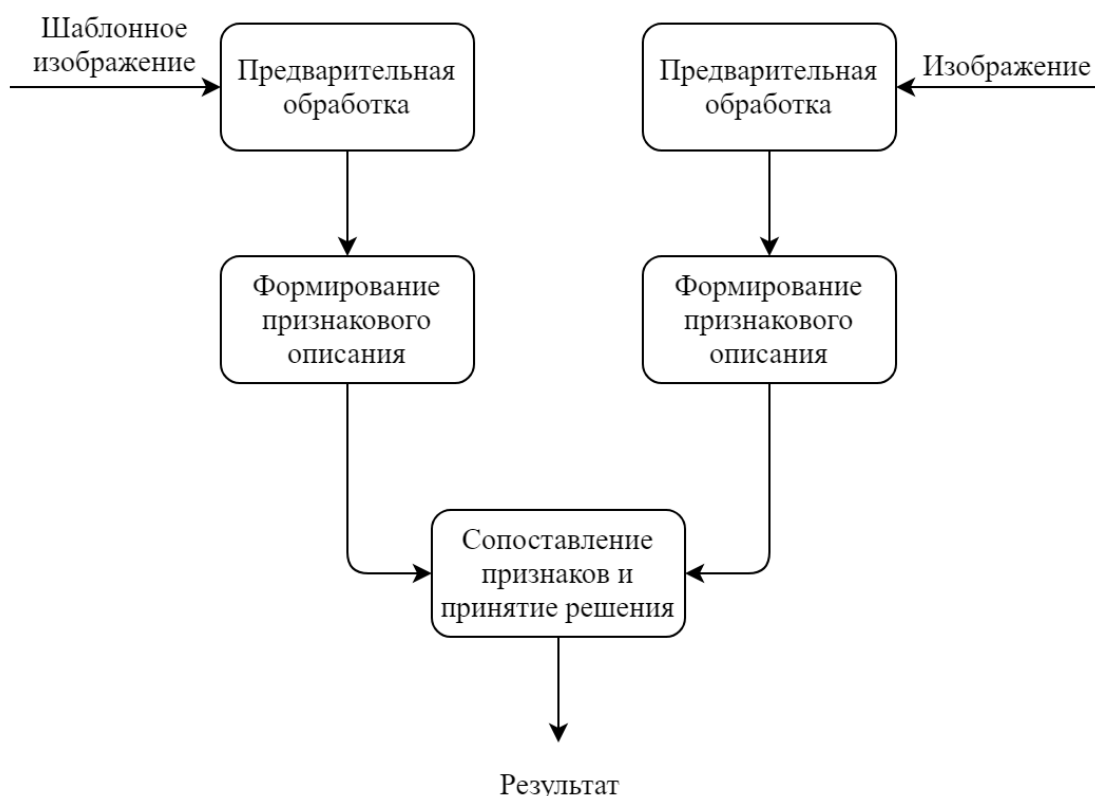


Рисунок 1.1. Структура системы обнаружения объектов

1.2. Обзор существующих методов

1.2.1. Методы предварительной обработки изображения

Предварительная обработка изображения, как правило, производится с целью избавления от дефектов изображения и повышения точности распознавания.

В наиболее распространённом в настоящее время методе для поиска объектов на изображении – SIFT [1], на этом этапе производится поиск точек интереса. В качестве них выступают так называемые «пятна» - это такие области, которые ярче или темнее, чем их окружение. Поскольку для низко контрастных изображений найти необходимое количество таких областей проблематично, перед поиском выполняется повышение контрастности изображения.

Основным моментом при поиске особых точек является построение пирамиды гауссианов и пирамиды разностей гауссианов. Гауссианом (или изображением, размытым гауссовым фильтром) является изображение, вычисляемое по следующей формуле:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (1.1)$$

где L – значение гауссиана в точке с координатами (x, y) , а σ – радиус размытия. G – гауссово ядро, I – значение исходного изображения, $*$ – операция свертки.

Разностью гауссианов называют изображение, полученное путем попиксельного вычитания одного гауссиана исходного изображения из гауссиана с другим радиусом размытия:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (1.2)$$

Масштабируемое пространство – это набор сглаженных некоторым фильтром версий исходного изображения. Доказано, что гауссово масштабируемое пространство является линейным, инвариантным относительно сдвигов, вращений, масштаба, не смещающим локальные экстремумы, и обладает свойством полугрупп.

Инвариантность относительно масштаба достигается за счет нахождения ключевых точек для исходного изображения, взятого в разных масштабах. Для этого строится пирамида гауссианов: все масштабируемое пространство

разбивается на некоторые участки – октавы, причем часть масштабируемого пространства, занимаемого следующей октавой, в два раза больше части, занимаемой предыдущей. При переходе от одной октавы к другой размеры изображения уменьшаются вдвое. Масштаб первого изображения следующей октавы равен масштабу изображения из предыдущей октавы с номером N .

Вместе с построением пирамиды гауссианов, строится пирамида разностей гауссианов, состоящая из разностей соседних изображений в пирамиде гауссианов. На рисунке 1.2 показаны пирамиды две этих пирамиды.

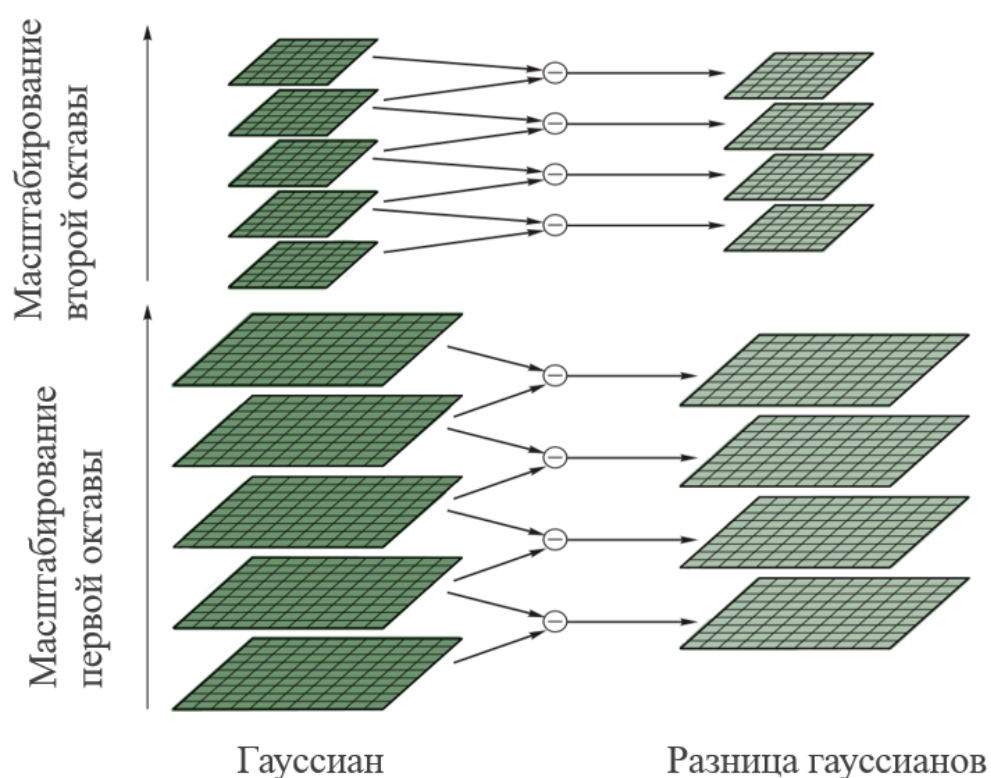


Рисунок 1.2. Пирамиды гауссианов и разницы гауссианов

После построения пирамид остается определить ключевые точки. Точка считается особой, если она является локальным экстремумом разности гауссианов. Для поиска экстремумов будем использовать метод, схематично изображенный на рисунке:

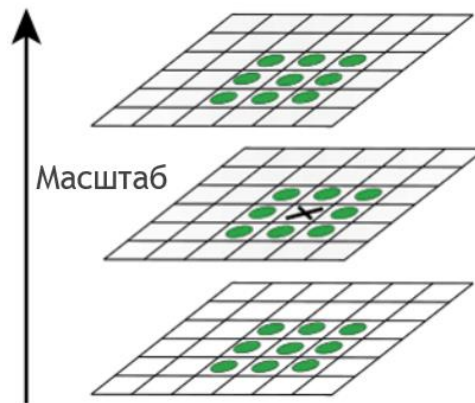


Рисунок 1.3. Поиск экстремумов

Каждая точка сравнивается с 26 соседями в текущей октаве. Точка считается точкой локального экстремума, если она больше или меньше всех соседей.

После того, как точка признана ключевой, нужно вычислить её ориентацию. Направление ключевой точки вычисляется исходя из направлений градиентов точек, соседних с особой. Его можно найти из гистограммы направлений. Эта гистограмма состоит из 36 компонент, которые равномерно покрывают промежуток в 360 градусов, и формируется она следующим образом: каждая точка окна (x, y) вносит вклад, равный $m * G(x, y, sigma)$, в ту компоненту гистограммы, которая покрывает промежуток, содержащий направление градиента $theta(x, y)$.

Направление ключевой точки лежит в промежутке, покрываемом максимальной компонентой гистограммы. Значения максимальной компоненты (max) и двух соседних с ней интерполируются параболой, и точка максимума этой параболы берётся в качестве направления ключевой точки. Если в гистограмме есть ещё компоненты с величинами не меньше $0.8 * max$, то они аналогично интерполируются и дополнительные направления приписываются ключевой точке.

В методе ASIFT [1], который является расширенным методом SIFT, после повышения контрастности для каждого изображения формируется несколько его копий, к каждой из которых применяется некоторое аффинное преобразование. Таким образом получается в своем роде несколько возможных фотографий данного «места» с разных точек обзора.

В работе [2] в качестве предварительной обработки изображения производится вычисление центрированных и нормализованных гистограмм для всех окон по изображению-шаблону и вычисление преобразований Фурье. Для

ускорения процесса обработки алгоритм использует прореживание. Локальная пороговая фильтрация гистограмм с последующим прореживанием результирующего изображения используется вместо классического пирамидального подхода с фильтрацией низких частот.

Для снижения уровня шума на изображениях разработано много методов и алгоритмов. Среди них можно выделить методы адаптивного сглаживания [4, 5], анизотропной диффузии [6, 7] и медианной фильтрации [4-9]. Последний обладает высоким быстродействием и позволяет снизить уровень помех на изображениях, при его работе не происходит усреднения по яркости, контуры объектов на изображении не размываются.

При медианной фильтрации значения отсчётов внутри окна фильтра сортируются в порядке возрастания (убывания). Значение, находящееся в середине упорядоченного списка, поступает на выход фильтра. В случае чётного числа отсчётов в окне выходное значение фильтра равно среднему значению двух отсчётов в середине упорядоченного списка. Окно перемещается вдоль фильтруемого сигнала, и вычисления повторяются.

В робототехнике для компьютерного зрения часто используется бинаризация изображений, которая позволяет выделять объекты в поле. Наиболее известны следующие алгоритмы бинаризации:

- Адаптивная бинаризация [4, 5]
- Метод Отсу [11]
- Методы кластеризации (k-means, EM-алгоритм) [12, 13]
- Обработка с постоянным порогом [4, 5]

Исследования показали, что метод Отсу дает лучшие результаты как по скорости, так и по качеству уменьшения шума. Данный алгоритм состоит из двух основных шагов.

Шаг 1. Автоматическое определение порога бинаризации по нормированной гистограмме яркостей изображения:

$$p_i = \frac{n_i}{N} \quad (1.3)$$

где N – общее число пикселей на изображении; n_i – число пикселей с уровнем яркости i .

Шаг 2. Процедура простого порогового разбиения изображения. Диапазон яркостей делится на два класса с помощью порогового значения уровня яркости t (целое значение от нуля до L). При этом каждому классу соответствуют:

$$\begin{aligned}\omega_0(t) &= \sum_{i=1}^t p_i, \\ \omega_1(t) &= 1 - \omega_0(t), \\ \mu_0(t) &= \sum_{i=1}^t \frac{ip_i}{\omega_0(t)}, \\ \mu_1(t) &= \sum_{i=t+1}^L \frac{ip_i}{\omega_1(t)},\end{aligned}\quad (1.4)$$

где $\omega_0(i)$ и $\omega_1(i)$ – относительная частота двух классов 0 и 1, разделенных порогом i ; $\mu_0(0)$ и $\mu_1(0)$ – средний уровень для каждого из двух классов 0 и 1 изображения, разделенных порогом i .

1.2.2. Методы построения признакового описания

Формирование признакового описания является самым важным этапом. От применяемых здесь алгоритмов зависит точность и скорость работы системы.

Метод формирования описания изображения на основе локальных бинарных шаблонов представлен в [14]. Он основан на вычислении знака разности между яркостями соседних отсчётов. Базовый оператор, применяемый к пикселю изображения, использует восемь пикселей окрестности, принимая значение интенсивности центрального пикселя в качестве порога:

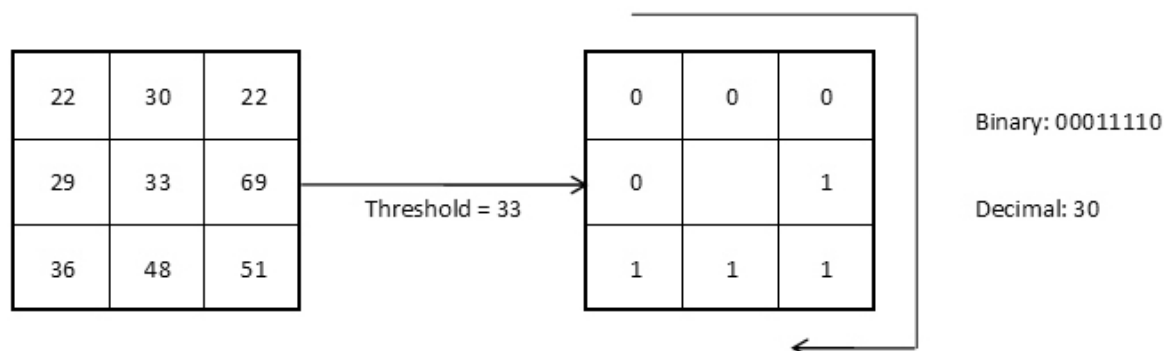


Рисунок 1.4. Базовый оператор локальных бинарных шаблонов

Пиксели со значением интенсивности большим или равным значению интенсивности центрального пикселя принимают значения равные «1», остальные

принимают значения равные «0». Таким образом, результатом применения базового оператора к пикселю изображения является восьмиразрядный бинарный код, который описывает окрестность этого пикселя.

Использование круговой окрестности и билинейной интерполяции значений интенсивностей пикселей позволяет построить локальный бинарный шаблон с произвольным количеством точек и радиусом.

Некоторые бинарные коды несут в себе больше информации, чем остальные. Так, локальный бинарный шаблон называется равномерным, если он содержит не более трех серий «0» и «1» (например, 00000000, 001110000 и 11100001). Равномерные ЛБШ определяют только важные локальные особенности изображения, такие как концы линий, грани, углы и пятна:

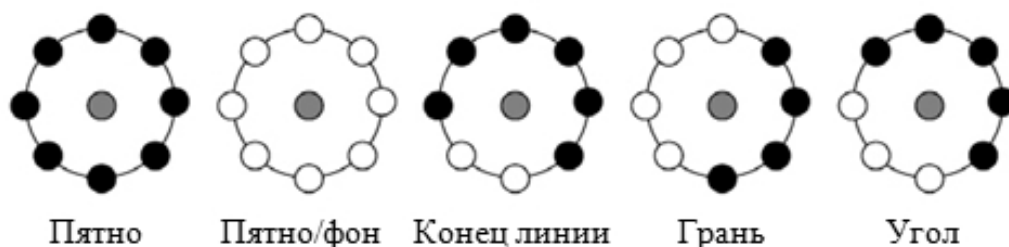


Рисунок 1.5. Примеры локальных особенностей

Данный метод не требователен к вычислительным ресурсам и устойчив к изменениям яркости, но не устойчив к шуму, вследствие использования пороговой операции. Известна модификация метода локальных бинарных шаблонов – метод локальных тернарных шаблонов [15]. Описание, формируемое на основе локальных тернарных шаблонов, устойчиво к шуму, но не устойчиво к изменению уровня яркости, так как алгоритм формирования признакового описания использует фиксированные predetermined пороги.

В методе SIFT [1] признаковое описание вычисляется по ключевым точкам, которые были найдены на предыдущем этапе. Такой выбор был сделан по той причине, что на маленькие области меньшее влияние оказывают эффекты искажений, а некоторые изменения (положение объекта, изменение сцены, перекрытие одного объекта другим, поворот) могут вообще не повлиять на дескриптор.

В данном методе дескриптором является вектор, который вычисляется на гауссиане, ближайшем по масштабу к ключевой точке, и исходя из градиентов в некотором окне ключевой точки. Перед вычислением дескриптора это окно

поворачивают на угол направления ключевой точки, чем и достигается инвариантность относительно поворота.

На рисунке 1.6 схематично показана часть изображения (слева) и полученный на её основе дескриптор (справа).

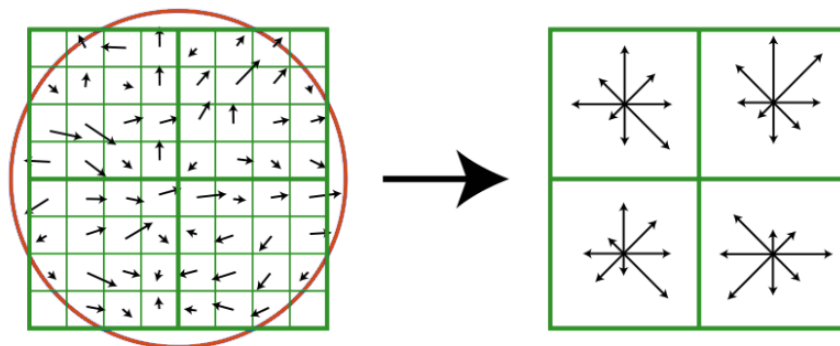


Рисунок 1.6. Часть изображения и полученный дескриптор

Каждая окрестность делится на четыре равных региона. Каждый пиксель в каждом регионе имеет направление градиента. Круг обозначает окно свертки с гауссовым ядром (аналогично окну для вычисления направления ключевой точки). Для этого ядра определяется σ , равное половине ширины окна дескриптора. В дальнейшем значение каждой точки окна дескриптора будет домножаться на значение гауссова ядра в этой точке, как на весовой коэффициент.

В построенном дескрипторе показаны компоненты гистограммы, вычисленные по каждому региону. В качестве ещё одних весовых коэффициентов берутся коэффициенты трилинейной интерполяции.

Каждому градиенту в окне дескриптора можно приписать три вещественные координаты (x, y, n) , где x – расстояние до градиента по горизонтали, y – расстояние по вертикали, n – расстояние до направления градиента в гистограмме (имеется ввиду соответствующая гистограмма дескриптора, в которую вносит вклад этот градиент). Коэффициент трилинейной интерполяции определяется для каждой координаты (x, y, n) градиента как $1-d$, где d равно расстоянию от координаты градиента до середины того единичного промежутка в который эта координата попала. Каждое вхождение градиента в гистограмму умножается на все три весовых коэффициента трилинейной интерполяции.

Полученный дескриптор нормализуется, после чего все его компоненты, значение которых больше 0.2, урезаются до значения 0.2 и затем дескриптор нормализуется ещё раз.

Построение дескрипторов в алгоритме SURF [15] основывается на тех же принципах, что и SIFT. SURF работает в несколько раз быстрее и является более устойчивым к различным преобразованиям изображения, но его недостаток проявляется на сильно размытых изображениях [16].

Для описания окрестности интересной точки используются отклики исходного изображения на вейвлеты Хаара:

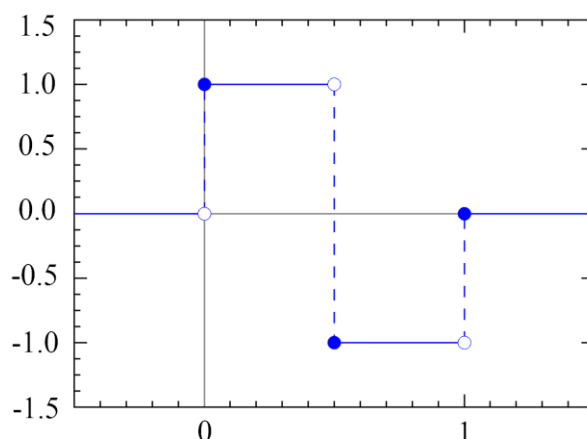


Рисунок 1.7. Вейвлеты Хаара

На первом шаге получения дескриптора вокруг точки интереса строится квадратная область, которую ориентируют по некоторому предпочтительному направлению. Затем область разделяется на квадратные сектора. В каждом из секторов в точках, принадлежащих регулярной сетке, вычисляются отклики на два вида вейвлетов — горизонтально и вертикально направленные. Отклики взвешиваются Гауссианом, суммируются по каждому сектору, и образуют первую часть дескриптора.

Вторая часть состоит из сумм модулей откликов. Это сделано для того, чтобы учитывать не только факт изменения яркости от точки к точке, но и сохранить информацию о направлении изменения.

К наиболее компактным дескрипторам относится дескриптор BRIEF [13]. Для вычисления дескриптора в точке сравниваются значения яркостей точек, расположенных в ее окрестности. Если яркость в точке p_{i1} больше, чем яркость в точке p_{i2} , то i -я компонента дескриптора принимает значение 1, в противном случае она становится равной нулю. Фрагмент, по которому вычисляются дескрипторы, предварительно сглаживается.

В статье было рассмотрено несколько способов выбора точек для бинарных сравнений. Как оказалось, один из лучших способов — выбирать точки случайным

образом Гауссовским распределением вокруг центрального пикселя. Эта случайная последовательность точек выбирается один раз и в дальнейшем не меняется.

BRIEF-дескрипторы чрезвычайно просты в вычислении, поскольку их значения равны результату сравнения двух чисел. Они также очень компактны, поскольку результат элементарного теста – это число 0 или 1, то есть один бит. В стандартной реализации для построения одного BRIEF-дескриптора требуется выполнить 256 сравнений, что дает итоговую длину 64 байта. Это очень мало, учитывая, что SIFT-дескриптор состоит из 128 действительных чисел, то есть занимает как минимум 512 байтов. Полученный бинарный дескриптор оказывается устойчив к сменам освещения, перспективному искажению, быстро вычисляется и сравнивается, но очень неустойчив к вращению в плоскости изображения.

1.2.3. Методы локализации объектов

Локализация объектов – последний этап, на котором принимается решение о том, найден объект на изображении или нет.

В методе BRIEF [13] для этой цели вычисляется расстояние Хэмминга [18] между двумя последовательностями битов. Расстояние Хэмминга – это количество различающихся позиций для строк с одинаковой длиной. Вычисляется по формуле:

$$\rho(d_i, d'_j) = \sum_{k=0}^{|K|} I(a_k \neq a'_k), \quad (1.5)$$

где d – дескриптор первого изображения, вектор из признаков a_k , d' – дескриптор второго изображения, вектор из признаков a'_k , $|K|$ – размерность вектора признаков.

Сейчас эта метрика широко используется для различных задач, таких как поиск близких дубликатов, распознавание образов, классификация документов, исправление ошибок, обнаружения вирусов и т.д.

Если дескрипторы представлены не в виде бинарных или символьных строк, то в качестве метрики может использовать Евклидово расстояние. Оно применимо для признаков, описываемых количественными переменными, и вычисляется по формуле:

$$\rho(d_i, d'_j) = \sum_{k=0}^{|K|} |a_k - a'_k|^2. \quad (1.6)$$

Евклидова метрика – наиболее естественная функция расстояния, возникающая в геометрии, отражающая интуитивные свойства расстояния между точками. Эта метрика используется для широкого ряда задач и является универсальной.

Вычисление вышеописанных метрик может быть финальным этапом в сопоставлении признаков. В этом случае есть два подхода:

1) Каждому дескриптору изображения-шаблона будет соответствовать столько дескрипторов тестируемого изображения, сколько совпадений будет найдено. Для того, чтобы считать дескрипторы совпавшими, можно задавать некоторый порог, который надо преодолеть.

2) Для каждого дескриптора изображения-шаблона необходимо найти наиболее подходящий дескриптор исследуемого изображения. Этот способ является более быстрым, но главный его недостаток – возможность поиска только одного объекта. Если искомым объектов на изображении будет несколько – алгоритм вероятнее всего даст сбой.

Другой вариант может содержать дополнительные шаги для локализации объекта. Если объект переместился на сцене или снят с другого ракурса, то можно выявить особенности, не относящиеся к общему объекту и тем самым уменьшить количество ложно определенных связей.

К одному из таких алгоритмов относится RANSAC [19]. Схема его работы заключается в циклическом повторении поиска матрицы трансформации H между случайно выбираемыми четырьмя особыми точками s_i на одном изображении и соответствующе им четырём точкам на втором:

$$s_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix}. \quad (1.7)$$

Лучшая матрица трансформации считается той, в которой достигнут минимум суммы отклонений всех особых точек изображений при преобразовании H , за заданное количество циклов. В итоговое множество включаются только те точки, отклонение которых не превосходит заданного порога.

Преимуществом алгоритма RANSAC является его способность дать надёжную оценку параметров модели, даже если в исходном наборе данных присутствует значительное количество выбросов.

Одним из недостатков метода является отсутствие верхней границы времени, необходимого для вычисления параметров модели. Если использовать в качестве некоторой границы времени максимальное число итераций, полученное решение может быть не оптимальным. Ещё одним недостатком является то, что для выполнения алгоритма необходимо задать конкретное пороговое значение.

После того, как все признаки сопоставлены и, возможно, применены дополнительные методы для достижения наилучшего результата, остается локализовать объект, т.е. указать область, где он найден. В данной работе для этого шага предлагается использовать один из методов разбиения изображения на сегменты, после чего сегмент, содержащий определенное количество верно сопоставленных точек, будет выделен, как искомый объект.

Большинство существующих алгоритмов сегментации можно разделить на два класса [20]:

1. Интерактивные – использующие пользовательские подсказки.
2. Автоматические – не требующие участия пользователя.

Есть также более узкоспециализированные типы сегментации [21], но в рамках данной работы они не представляют интереса.

Автоматические алгоритмы сегментации, в свою очередь, также делятся на два класса:

1. Выделение областей изображения с заданными свойствами, уникальными для конкретной предметной области.
2. Разбиение изображения на однородные области.

Второй класс алгоритмов автоматической сегментации не использует априорную информацию об изображении, но к свойствам получаемых сегментов предъявляются некоторые требования, такие как однородность текстуры внутри сегмента и непохожесть текстур смежных областей, гладкость границ сегментов и т.п. Одним из таких автоматических алгоритмов сегментации изображений является алгоритм Mean shift [22].

Основная идея Mean shift метода [23] заключается в том, что по входным данным (изображению) можно построить ядерную оценку для плотности вероятности распределения данных в пространстве признаков. Далее делается

естественное предположение о том, что локальные максимумы плотности вероятности соответствуют центрам сегментов. Таким образом, для работы этого алгоритмы нет необходимости заранее задавать количество сегментов. Рассмотрим его работу немного подробнее.

Предположим, дан набор точек $\{u_i\}$ в d -мерном пространстве, отобранный из некоторой большой совокупности, и что выбрано ядро K , имеющий параметр полосы пропускания h . Вместе эти данные служат для описания оценки плотности ядра:

$$f_k(u) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{u-u_i}{h}\right). \quad (1.8)$$

При этом функция ядра должна удовлетворять условиям:

$$\begin{aligned} \int K(u) du &= 1, \\ K(u) &= K(|u|). \end{aligned} \quad (1.9)$$

Первое требование нужно для того, чтобы оценка была нормализована, а второе связано с симметрией пространства. Две наиболее часто используемые функции ядра, которые удовлетворяют этим условиям, определяются как:

$$\begin{aligned} K(u) &= \frac{1}{2} \begin{cases} 1 & -1 \leq |u| \leq 1 \\ 0 & \text{иначе} \end{cases}, \\ K(u) &= \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}|u|^2}. \end{aligned} \quad (1.10)$$

Алгоритм сегментации Mean shift является практическим применением процесса поиска максимума функции плотности. Набор всех точек в пространстве, которые сходятся к одному и тому же максимуму функции плотности, называется множеством точек данного максимума. Считается, что точки, попавшие в одно и то же множество, относятся к одному и тому же сегменту. Таким образом, этот алгоритм сегментации позволяет разделять точки изображения на заранее неизвестное число сегментов.

Вторым из наиболее популярных алгоритмов сегментации является алгоритм k-means [24]. Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек сегментов от центров

этих сегментов. Это итеративный алгоритм, который делит данное множество пикселей на k сегментов точки, которые являются максимально приближенными к их центрам, а сама сегментация происходит за счет смещения этих же центров. Для этого алгоритма необходимо заранее указывать число сегментов, на которые будет разбито изображение.

Сначала для каждого сегмента случайным образом выбираются центры масс. После этого все объекты разбиваются на сегменты по принципу близости к центрам масс сегментов. На следующих итерациях вычисляются новые центры масс на основе сегментов, полученных на предыдущем этапе. С каждым шагом алгоритм стремится минимизировать суммарное квадратичное отклонение точек сегментов от центров этих сегментов:

$$V = \sum_{i=1}^k \sum_{j \in S_i} (x_j - \mu_i)^2, \quad (1.11)$$

где k – число сегментов; S_i – i -й сегмент; $i = \overline{1, k}$; μ_i – центр масс сегмента S_i ; x_j – j -й элемент сегмента S_i .

Алгоритм завершает свою работу, когда на одной из итераций не происходит изменение центров масс сегментов.

Алгоритм k-means имеет более высокую скорость работы нежели алгоритм Mean shift, но к его недостаткам можно отнести то, что число сегментов необходимо задавать как входной параметр алгоритма, и для каждого конкретного случая наиболее подходящее число этих сегментов может варьироваться.

1.3. Выводы по главе 1

В данной главе были рассмотрены известные методы для реализации каждого из этапов поиска объекта на изображении. Исследование показало, что в настоящее время существует множество методов для решения этой задачи, с каждым годом они совершенствуются, а также появляются новые разработки, что говорит об актуальности данной темы. При составлении обзора было получено представление о базовых принципах работы методов, об их достоинствах и недостатках. Проблемы, с которыми сталкивались при применении алгоритмов, были толчком для их улучшения, для разработки усовершенствованных версий. Эти проблемы также были изучены, что позволит частично их избежать при разработке системы.

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		21

Глава 2. Обзор теории активного восприятия: алгоритм и его практическое применение

Теория активного восприятия (ТАВ) [25] может использоваться для этапа предварительной обработки изображения и для этапа формирования системы признаков [26-31].

Она базируется на операции U -преобразования - последовательном применении к изображению операций интегрирования и дифференцирования. В результате этого формируется набор спектральных коэффициентов, который и является признаковым описанием изображения.

Теория активного восприятия также включает в себя так называемую «Алгебру групп». Метод используется для выявления зависимостей между спектральными коэффициентами, полученными в результате вычисления U -преобразования. Данные зависимости также могут быть использованы при формировании признакового описания изображения.

2.1. Q -преобразование

Данный этап служит для предварительной обработки изображения. Его результатом является так называемая «матрица визуальных масс» размером 4×4 элемента (2.1).

$$m[i, j] = \sum_{i=(i-1/4)N}^{(i/4)N} \sum_{l=(l-1/4)M}^{(l/4)M} I[k, l], i = \overline{1, 4}, j = \overline{1, 4}, \quad (2.1)$$

где I – изображение размером $N \times M$.

Для практического применения это означает деление области интереса (всего изображения или его части, в зависимости от поставленной задачи) на 16 частей и вычисление суммарного значения яркости в каждой области. Для этого в каждой области вычисляется сумма яркостей всех входящих в нее пикселей.

Пример выполнения Q -преобразования показан на рисунке 2.1.

Первым делом исходное изображение переводится в градации серого для понижения вычислительной сложности (рис.2.1(б)). Затем делится на 16 равных областей (рис.2.1(в)). Для каждой области вычисляется сумма яркостей, входящих в нее пикселей. Каждый пиксель может принимать значения 0..255. Чем больше значение, тем светлее пиксель.

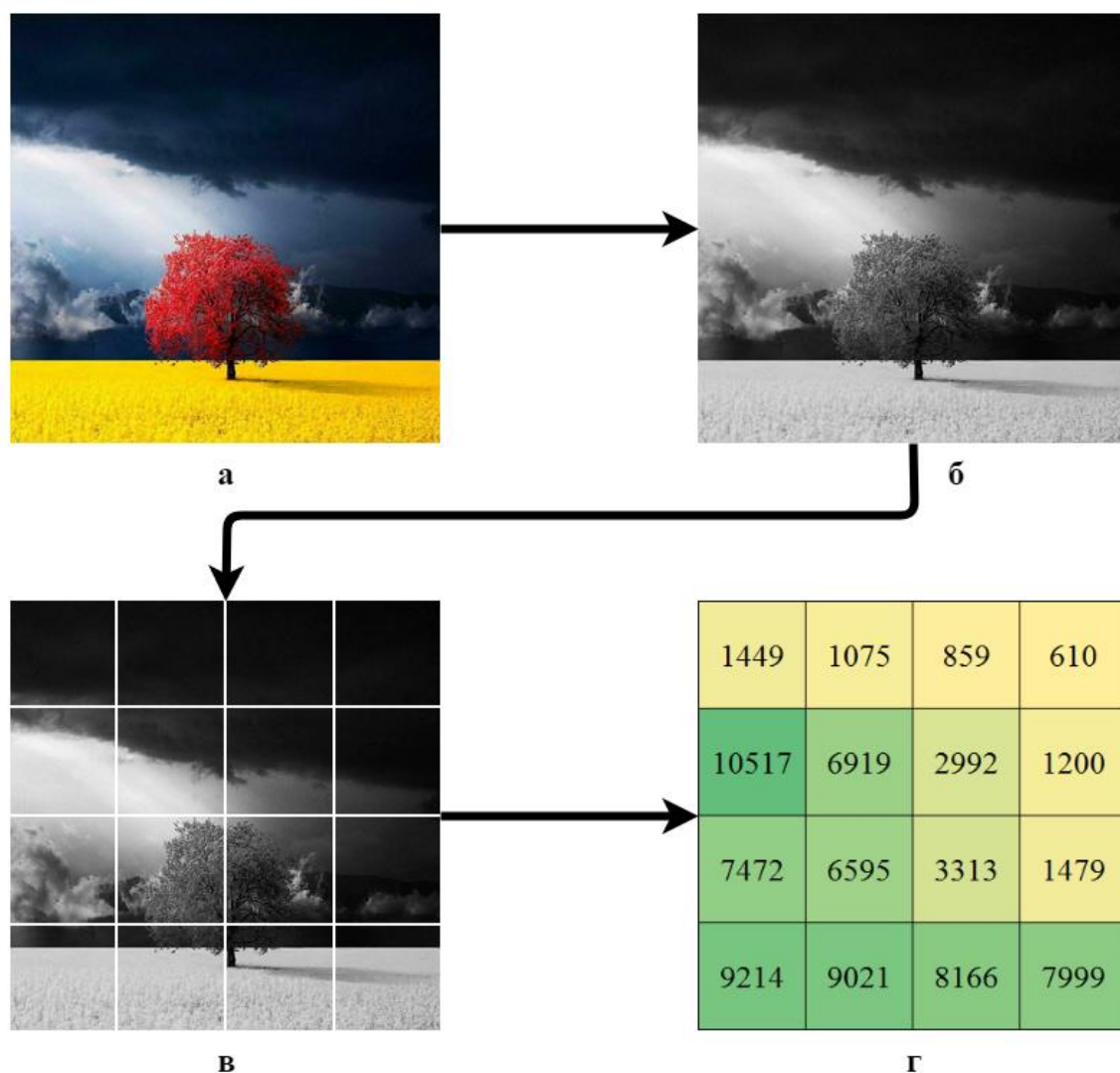


Рисунок 2.1. Выполнение Q -преобразования

Соответственно, чем светлее в целом область, тем значение визуальной массы этой области будет больше. В результате вычисления суммы пикселей по всем областям и получается матрица визуальных масс. Как можно видеть на рис.2.1(г), действительно, чем темнее область, тем меньше значение визуальной массы этой области. Для наглядности все значения на рис.2.1(г) были поделены на 100, в действительности значения, собранные по областям всего изображения целиком, значительно больше.

2.2. U -преобразование

На этом этапе к результату Q -преобразования применяется множество фильтров Уолша системы Хармута (см. рис.2.2). Специфика использования фильтров в данном методе заключается в том, что они применяются после выполнения Q -преобразования.

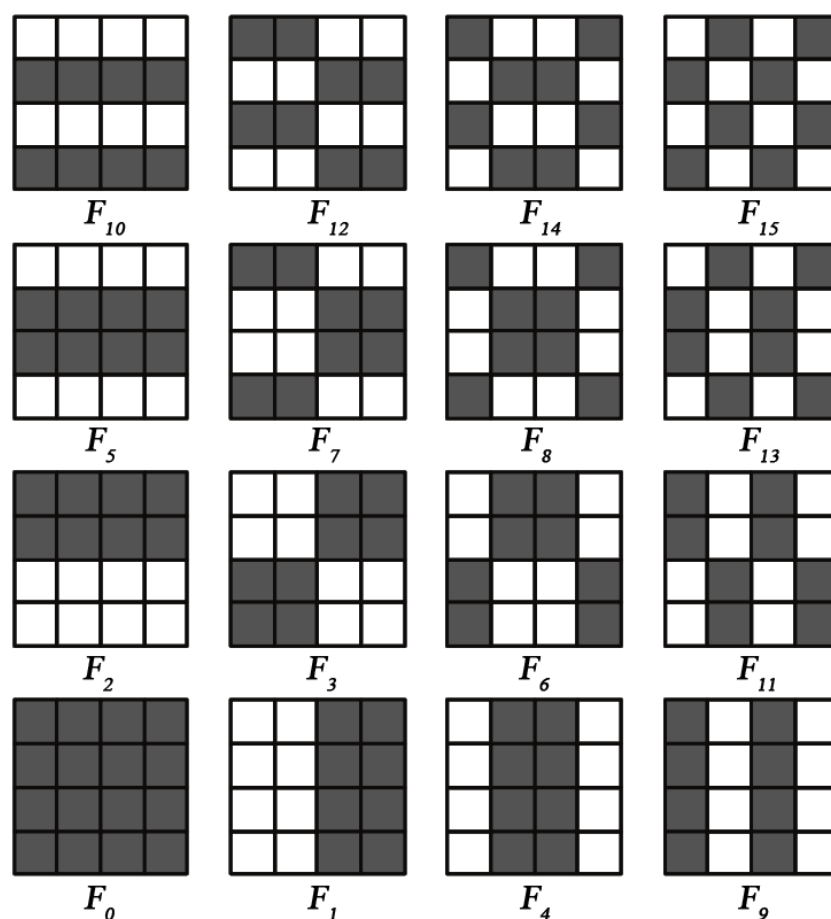


Рисунок 2.2. Фильтры Уолша системы Хармута

В результате этого формируется вектор спектральных коэффициентов:

$$\mu_i = \sum_{k=1}^4 \sum_{l=1}^4 m[k,l] * F_i[k,l], \quad (2.2)$$

где $F_i, i = \overline{1,16}$ – множество фильтров.

Размер каждого фильтра 4x4 элемента, также, как и матрица визуальных масс. На рисунке 2.2 можно видеть, что фильтры имеют темные и светлые области. Темная область принимает значение +1, светлая – значение -1. На рисунке 2.3 можно видеть представление одного из фильтров в таком числовом формате.

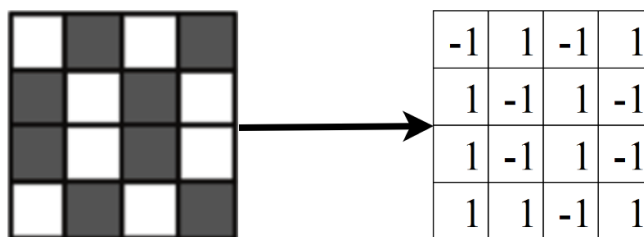


Рисунок 2.3. Представление фильтра №13 в числовом формате

В ходе этого этапа матрица визуальных масс, полученная на этапе выполнения Q -преобразования, умножается на рассмотренные фильтры. Всего это действие выполняется 16 раз, для каждого из фильтров.

На рисунке 2.4 показан пример умножения рассмотренной ранее матрицы на один из фильтров.

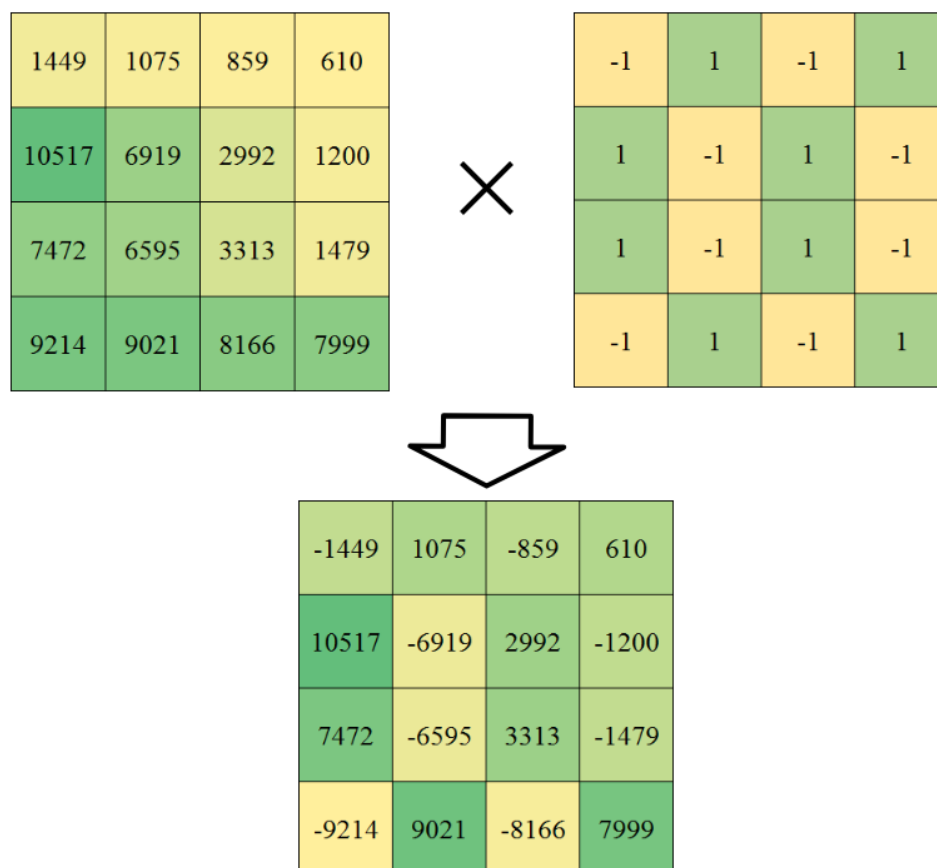


Рисунок 2.4. Умножение матрицы визуальных масс на фильтр №13

Для получения значения спектрального коэффициента по этому фильтру необходимо сложить все полученные после умножения значения:

$$-1449 + 1075 - 859 + 610 + 10517 - 6919 + 2992 - 1200 + 7472 - 6595 + 3313 - 1479 - 9214 + 9021 - 8166 + 7999 = 7118$$

Значение 7118 является итоговым для данного фильтра. По такому же принципу вычисляются значения по всем фильтрам, в итоге получаем 16 чисел, характеризующих изображение или его часть.

Для рассматриваемого изображения вектор спектральных коэффициентов приведен в таблице 2.1. Как можно видеть, наибольшее значение имеет спектральный коэффициент фильтра №0, т.к. все его компоненты имеют положительное значение. На втором месте фильтр №10, поскольку его

положительные компоненты охватывают наиболее светлую часть изображения, где значения пикселей максимальные. Фильтры №1 и №2 имеют минимальные значения, т.к. их положительные компоненты охватывают наибольшее число темных областей.

Таблица 2.1. Спектральные коэффициенты

Фильтр	Спектральный коэффициент
0	78887
1	-25645
2	-27637
3	-2954
4	-1000
5	9084
6	-2862
7	-19395
8	-695
9	2093
10	33176
11	2943
12	-4985
13	7118
14	-2664
15	2416

Таким образом, при выполнении U -преобразования был получен вектор спектральных коэффициентов, характеризующий изображение, который может быть использован в качестве признакового описания. В качестве примера рассматривалось его применение к полному изображению, но на практике возможно его использование и на частях изображения. Здесь есть несколько вариантов:

1) Применение U -преобразования на всем изображении, поделенном на сегменты. В этом случае задается некоторое «скользящее окно» - область выбранного размера, которая будет проходить по всему изображению. Изображение делится на области этого размера и по каждой такой области вычисляются спектральные коэффициенты. Такой подход более устойчив к изменению угла наклона объекта, повороту и изменению яркости.

2) Применение U -преобразования к заранее найденным ключевым областям. В этом случае на этапе предварительной обработки изображения производится

поиск ключевых точек любым из существующих методов, и затем, для каждой из этих точек выполняются Q - и U -преобразования.

В зависимости от разрабатываемого алгоритма этот этап может быть последним при формировании признакового описания или за ним может следовать этап под названием Алгебра групп.

2.3. Алгебра групп

Данный раздел предназначен для выявления зависимостей между спектральными коэффициентами, полученными на этапе вычисления U -преобразования. Для этого в алгебре групп вводятся понятия операторов, полных и замкнутых групп.

В ТАВ каждому фильтру F_i ставится в соответствие бинарный оператор V_i . В этом случае компоненте $\mu_i \neq 0$ вектора μ соответствует оператор V_i или \bar{V}_i в зависимости от знака компоненты, а вектору μ соответствует подмножество операторов из V_i , имеющих аналогичную фильтрам конструкцию, но разное значение элементов матрицы: +1 соответствует 1, а -1 соответствует 0 [26]. Таким образом, для уже рассмотренного фильтра F_{13} бинарный оператор V_{13} , который ему соответствует, будет иметь вид, представленный на рисунке:

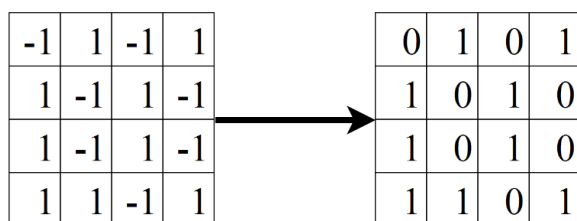


Рисунок 2.5. Бинарный оператор для фильтра №13

На основании полученных операторов формируются множества полных и замкнутых групп:

- Множество полных групп $P_n = \{P_{ni}\}$, $P_{ni} = \{V_i, V_i, V_k\}$ мощностью 35
- Множество замкнутых групп $P_s = \{P_{si}\}$, $P_{si} = \{V_i, V_i, V_k, V_r\}$ мощностью 105

С учётом возможных инверсий операторов, входящих в описание полной и замкнутой группы, всего существует 140 полных групп и 840 замкнутых.

Полные и замкнутые группы позволяют произвести спектрально-корреляционный анализ изображения. Полные группы выявляют корреляционные

связи между операторами, замкнутые группы – связи между полными группами. Иерархия операторов, полных и замкнутых групп представлена на рисунке:

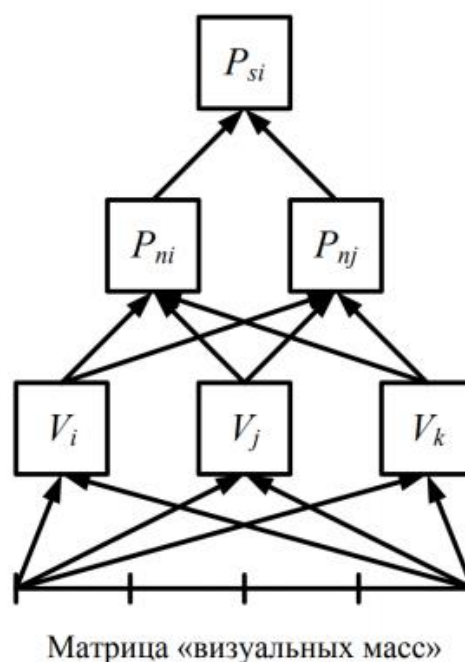


Рисунок 2.6. Иерархия операторов, полных и замкнутых групп

Каждому оператору или группе ставится в соответствие понятие образа оператора или группы. Образ оператора V_{13} представлен на рисунке 2.5. На основании подобных образов происходит вычисление масс операторов, полных и замкнутых групп. Пример вычисления массы замкнутой группы $V_1 V_3 V_7 \overline{V_{12}}$ по матрице визуальных масс уже рассмотренного изображения (рисунок 2.1) представлен на рисунке:

1449	1075	859	610
10517	6919	2992	1200
7472	6595	3313	1479
9214	9021	8166	7999

×

0	0	1	1
0	0	1	1
0	0	1	1
1	1	0	0

28688

Рисунок 2.7. Вычисление массы замкнутой группы

Как и при вычислении спектральных коэффициентов каждый элемент матрицы визуальных масс перемножается с соответствующим элементом образа группы.

Информация о наличии полных и замкнутых групп в спектре изображения или его сегмента, а также массы данных групп являются базисом, на котором формируется еще один вид признакового описания изображения, предлагаемый теорией активного восприятия.

Данный вид признакового описания является более вычислительно сложным по сравнению со спектральным представлением изображения, полученным в результате U -преобразования. Однако признаковое описание на основе алгебры групп несет в себе больше информации, в том числе о корреляционных связях между спектральными коэффициентами, что позволяет иметь более точное признаковое описание изображения.

2.4. Выводы по главе 2

В данной главе была рассмотрена теоретическая составляющая теории активного восприятия, а также практическое применение ее отдельных частей. Были представлены наглядные примеры ее работы, рассмотрены и проанализированы результаты. Был также затронут раздел Алгебра групп, который позволяет получать более точное признаковое описание, хоть и требует больших вычислительных затрат.

Глава 3. Реализация системы обнаружения объекта на изображении на основе теории активного восприятия

Теория активного восприятия позволяет реализовать как этап предварительной обработки изображения, так и этап формирования признакового описания. Ее можно использовать как единственный метод, а также в комбинации с уже существующими, которые будут реализовывать один из этапов. В данной главе описывается реализация системы с использованием теории активного восприятия в обоих этапах обнаружения объекта на изображении.

Информационная модель решения задачи локализации объекта на изображении с помощью теории активного восприятия представлена на рисунке 3.1. Далее в разделе будет подробно рассмотрен каждый из этапов, представленных в модели.



Рисунок 3.1. Информационная модель разрабатываемой системы

3.1. Предварительная обработка изображения

Разрабатываемая система принимает на вход изображения, закодированные согласно модели RGB. Каждый пиксель такого изображения представлен в виде числового вектора из трёх элементов, которые описывают яркости красного, зеленого и синего компонентов модели RGB. Для применения теории активного восприятия яркость пикселя должна быть представлена одним значением, поэтому необходимо представить исследуемое изображение в градациях серого.

Согласно рекомендации Международного союза электросвязи [32] изображение, представленное в виде компонент R , G и B , может быть переведено в изображение с компонентами Y , C_B и C_R , где Y – яркость изображения, а C_B и C_R – синяя и красная цветоразностные компоненты соответственно. Для нашей цели необходима только Y компонента, т.к. требуется только яркость изображения. Для получения этой компоненты в рекомендации представлена формула:

$$Y = 0.299R + 0.587G + 0.144B. \quad (3.1)$$

Если рассматривать трансляцию каждого пикселя изображения из RGB в $YC_B C_R$, то получим следующую функцию яркости:

$$I(x, y) = 0.299R_{xy} + 0.587G_{xy} + 0.144B_{xy}, x \in X, y \in Y, \quad (3.2)$$

где R_{xy} , G_{xy} , B_{xy} – яркости соответствующих компонент модели RGB для пикселя с координатами (x, y) ; X, Y – множества соответствующих координат изображения.

Матрица значений яркости изображения I формируется как набор значений функции $I(x, y)$ на всем множестве X и Y . Для вычисления признакового описания изображение будет поделено на области, что позволит добиться большей инвариантности к повороту объекта, наклону и смене освещения. Для этого деления выбирается некоторое «скользящее окно» размером $n \times n$, которое проходит по всему изображению и выделяет области интереса для дальнейшего вычисления по ним признакового описания. Таким образом изображение можно представить в виде набора областей:

$$P = \{P_{ij}\}, i \in X, j \in Y, \quad (3.3)$$

где P_{ij} – область изображения размером $n \times n$ пикселей; i, j – координаты левого верхнего угла области P_{ij} .

Для достижения большей степени детализации признакового описания изображения набор областей формируется с «перекрытием», т.е. разница между координатами (i,j) двух соседних областей по X и по Y меньше, чем размер области n .

Для изображения-шаблона на этом этапе выполняется еще несколько шагов. Чтобы добиться инвариантности к повороту и масштабу, после представления изображения в градациях серого, выполняется его масштабирование с коэффициентами 0.8, 0.6, 0.4, а также вращение на углы -30, -20, -10, 10, 20, 30

Таким образом, изображение представляется в виде набора областей, взятых как с исходного изображения, так и с трансформированного. Данное представление является подходящим для выполнения следующего этапа решения задачи – формирования признакового описания.

3.2. Формирование признакового описания

Признаковое описание для разрабатываемой системы представляет собой набор векторов спектральных коэффициентов изображения, полученных в результате выполнения U -преобразования. Оно является совокупностью признакового описания каждой из полученных на предыдущем этапе областей.

Признаковое описание для каждой области изображения P_{ij} можно представить в виде следующей формулы:

$$D_{ij} = \{d_k\}, k = \overline{1, L}, \quad (3.4)$$

где d_k – k -й коэффициент спектрального представления области изображения; L – количество фильтров, применяемых при вычислении U -преобразования.

Исходя из этого признаковое описание изображения в целом можно представить в виде набора признаковых описаний всех его областей:

$$D = \{D_{ij}\}, i \in X, j \in Y, \quad (3.5)$$

Для оптимизации затрачиваемой памяти и времени работы системы полученное признаковое описание сокращается. В итоговые дескрипторы изображения заносятся только те признаки, которые соответствуют ключевым областям. Будем считать область ключевой, если в ней есть перепад яркости, который соответствует контуру объекта.

Признаковое описание областей изображения, состоящее из спектральных коэффициентов U -преобразования, позволяет легко выявлять области с перепадами яркости. Для каждой области, по которой были вычислены спектральные коэффициенты, вычисляется их среднеквадратичное отклонение (СКО). Поскольку фильтр №0 полностью состоит из положительных значений, его результатом является сумма всех визуальных масс области, и при расчете СКО он не учитывается. Если брать в расчет этот фильтр, то вместо выделения границ получим выделение самых светлых участков на изображении.

Исходя из этого получаем формулу для вычисления СКО спектральных коэффициентов области P_{ij} :

$$s_{ij} = \sqrt{\frac{1}{n-1} \sum_1^n (d_i - d_{cp})^2}, \quad (3.6)$$

где $n = 15$ – количество учитываемых коэффициентов спектрального представления; d_i – i -й спектральный коэффициент; d_{cp} – среднее арифметическое спектральных коэффициентов рассматриваемой области без учета 0-го коэффициента.

После того как для каждой области было вычислено среднеквадратичное отклонение, из полученных значений находим максимальное. Для принятия решения, является ли рассматриваемая область ключевой, введем коэффициент, который в применении с максимальным значением СКО будет давать некий порог, уникальный для каждого изображения:

$$s_{ij} > kS_{\max}, \quad (3.7)$$

где k – задаваемый коэффициент отбора точек, S_{\max} – максимальное значение СКО по всем областям изображения.

Таким образом, если среднеквадратичное отклонение спектральных коэффициентов некоторой области больше, чем максимальное среднеквадратичное отклонение всех областей изображения, умноженное на коэффициент k , то эта область содержит ключевую точку.

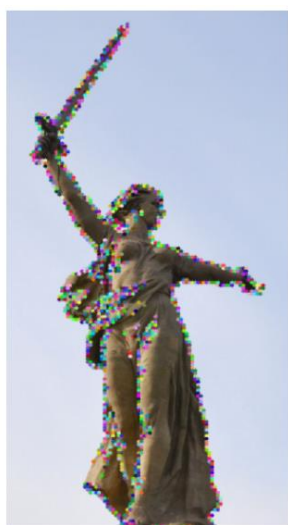
На рисунке 3.2 приведен пример найденных ключевых точек с разными значениями коэффициента:

а) Значение коэффициента равно 0.3, найдено 1137 ключевых точек

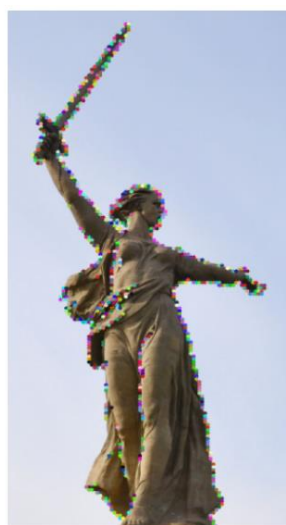
					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		33

б) Значение коэффициента равно 0.5, найдено 577 ключевых точек

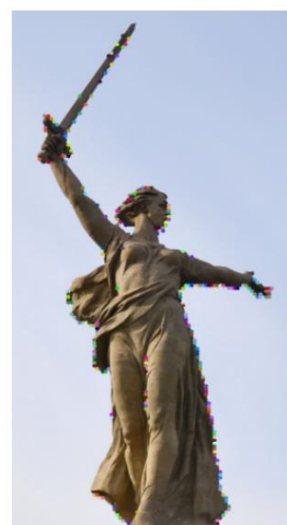
в) Значение коэффициента равно 0.7, найдено 215 ключевых точек



а



б



в

Рисунок 3.2. Результат поиска ключевых точек с разным значением коэффициента

Поиск ключевых точек выполнялся на изображении, представленном в градациях серого. Координаты найденных точек были графически представлены на цветном изображении для наглядности.

Из результатов видно, что увеличение коэффициента задает больший порог, что приводит к уменьшению количества точек, принимаемых за ключевые. Чем ниже его значение, тем меньший перепад яркости области нужен для того, чтобы считать область ключевой. Выставление этого значения в 0 сделает каждую рассматриваемую область ключевой, даже ту, где нет перепада яркости.

Как уже было сказано в п.3.1, чтобы добиться инвариантности к повороту и масштабу, изображение-шаблон масштабируется с коэффициентами 0.8, 0.6, 0.4, а также поворачивается на углы -30, -20, -10, 10, 20, 30. После каждого такого изменения по изображению заново вычисляются дескрипторы и выявляются ключевые точки. Исходя из этого, изображение-шаблон будет иметь больше ключевых точек, чем тестовое изображение:

$$(\text{все коэффициенты масштаба}) * (\text{все углы поворота}) = 4 * 7 = 28$$

Именно во столько раз, в теории, должно отличаться количество ключевых точек шаблонного изображения и тестового, но на практике оно отличается всего в 3-5 раз за счет нескольких вещей:

1. Чем меньше изображение, тем меньше на нем будет найдено ключевых точек.
2. Коэффициент отбора ключевых точек с уменьшением размера изображения, отбрасывает большее количество точек.

Таким образом, с использованием теории активного восприятия было реализовано вычисление признакового описания изображения, а также его сокращения с помощью выявления точек интереса. Это сокращение позволит уменьшить затрачиваемое время и используемую память на этапе принятия решения.

3.3. Принятие решения о локализации объекта на изображении

Последний этап служит для сопоставления дескрипторов исследуемого изображения и изображения-шаблона. В рамках данной работы для этой цели используется метрика под названием Евклидово расстояние, которая вычисляется по формуле:

$$\rho(d_i, d'_j) = \sum_{k=0}^{|K|} |a_k - a'_k|^2. \quad (3.8)$$

Как уже было отмечено в главе 1, Евклидова метрика – наиболее естественная функция расстояния, возникающая в геометрии, она отражает интуитивные свойства расстояния между точками.

Для каждой точки изображения-шаблона вычисляется расстояние до каждой из точек исследуемого изображения. В результате этого получаем вектор, который содержит 5 значений: вычисленное расстояние, координаты точки первого изображения и координаты точки второго изображения.

Ряд существующих методов (SIFT, SURF) на этом этапе находят только одну наиболее подходящую точку, в следствии чего такие методы не могут обнаружить 2 и более объекта, если они присутствуют на изображении. Сопоставление точки «один – ко всем» занимает немного больше времени, нежели если останавливать поиск на первой подходящей, но такой подход значительно расширяет функциональность метода, позволяя находить все присутствующие на изображении объекты.

На рисунке 3.3 представлен пример сопоставления особых точек двух изображений. В этом примере присутствует изменение освещения искомого

объекта. Как можно заметить, из-за этого изменения не все точки сопоставляются корректно, но все они ведут на требуемый объект.

После того, как точки были сопоставлены, необходимо локализовать объект, т.е. выделить его на исследуемом изображении. Для этой цели будем использовать алгоритм k-means, который был рассмотрен в главе 1. Поскольку алгоритм требует указать количество сегментов, на которое будет поделено изображение, один его вызов с некоторым заданным числом может быть не всегда эффективен. Для решения этой проблемы было решено поступить следующим образом.



Рисунок 3.3. Сопоставление дескрипторов особых точек

Первый вызов k-means осуществляется с минимальным числом сегментов – 2. Результат показан на рисунке:



Рисунок 3.4. Применение метода k-means

После этого используется вспомогательный метод выборочного поиска [33], который из полученных сегментов создаст набор областей, которые в дальнейшем будут проверяться на наличие объекта с помощью уже сопоставленных нами точек. Данный метод осуществляет выборочный поиск, используя 4 показателя сходства, основанные на совместимости цвета, текстуры, размера и формы.

Рассмотрим подробнее алгоритм. Выборочный поиск начинается с чрезмерного сегментирования изображения на основе интенсивности пикселей с использованием метода сегментации на основе графиков, предложенного Р. Felzenszwalb и D. Huttenlocher в [34].

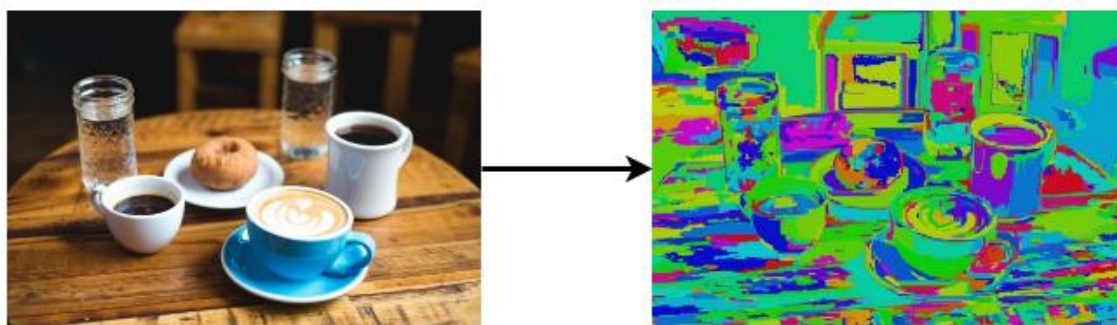


Рисунок 3.5. Чрезмерное сегментирование изображений

Полученные сегментированные части не могут использоваться для поиска регионов, т.к. каждый объект имеет 2 и более сегментированных частей. Алгоритм выборочного поиска принимает эти избыточные сегменты в качестве исходного ввода и выполняет следующие шаги:

1. Добавление всех ограничивающих рамок, соответствующих сегментированным частям, в список региональных предложений.
2. Группировка смежных сегментов на основе сходства.
3. Переход к шагу 1.

На каждой итерации большие сегменты формируются и добавляются в список региональных предложений. Следовательно, создаются предложения по регионам от меньших сегментов до больших сегментов в подходе снизу-вверх. Это является вычислением иерархической сегментации с использованием надсегментов Р. Felzenszwalb и D. Huttenlocher.

Рисунок 3.6 показывает начальные, средний и последний шаг процесса иерархической сегментации.

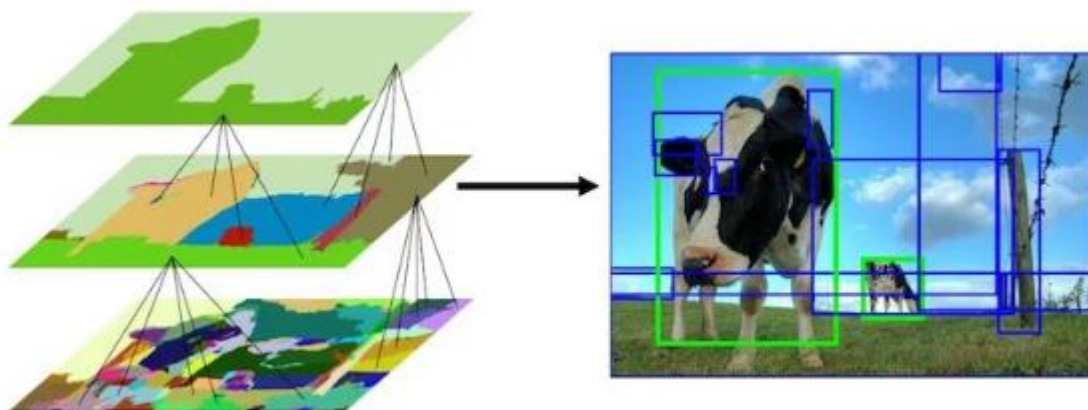


Рисунок 3.6. Иерархическая сегментация

Как уже было сказано ранее, метод используем 4 показателя сходства, на основе которых происходит группировка маленьких сегментов.

Сходство цвета. Цветовая гистограмма из 25 столбцов рассчитывается для каждого канала изображения, а гистограммы для всех каналов объединяются для получения цветового дескриптора. Поскольку на вход метода подается результат k-means, у нас есть только один канал, что повышает скорость вычислений. Цветовое сходство двух областей основано на пересечении гистограммы и может быть рассчитано как:

$$s_{color}(r_i, r_j) = \sum_{k=1}^n \min(c_i^k, c_j^k), \quad (3.9)$$

где r_i, r_j – две области или два сегмента изображения; c_i^k, c_j^k – значения гистограмм в цветовых дескрипторах этих сегментов.

Сходство текстуры. Особенности текстуры рассчитываются путем извлечения гауссовых производных в 8 ориентациях для каждого канала. Для каждой ориентации и для каждого цветового канала вычисляется гистограмма с 10 ячейками, в результате получается дескриптор пространственного объекта $10 * 8 * 3 = 240$. Для случая, когда на вход подается изображение с одним каналом, размер дескриптора составляет $10 * 8 = 80$. Сходство текстуры двух областей также рассчитывается с использованием пересечений гистограммы:

$$s_{texture}(r_i, r_j) = \sum_{k=1}^n \min(t_i^k, t_j^k), \quad (3.10)$$

где t_i^k, t_j^k – значения гистограмм в текстурных дескрипторах сегментов.

Сходство размера. Эта мера способствует слиянию небольших регионов. Если эта мера сходства не принимается во внимание, то один регион будет продолжать поглощать все меньшие смежные области один за другим. Размер сходства определяется как:

$$s_{size}(r_i, r_j) = 1 - \frac{size(r_i) + size(r_j)}{size(im)}, \quad (3.11)$$

где $size(im)$ – размер изображения в пикселях.

Сходство формы. Совместимость формы измеряет, насколько хорошо две области соответствуют друг другу. Если две области вписываются друг в друга, они объединяются, чтобы заполнить пробелы, и, если они даже не касаются друг друга, они не должны быть объединены. Совместимость формы определяется как:

$$s_{fill}(r_i, r_j) = 1 - \frac{size(BB_{ij}) - size(r_i) - size(r_j)}{size(im)}, \quad (3.12)$$

где размер $size(BB_{ij})$ - ограничительная рамка вокруг двух областей.

Окончательное сходство между двумя областями определяется как линейная комбинация вышеупомянутых сходств:

$$s(r_i, r_j) = a_1 s_{color} + a_2 s_{texture} + a_3 s_{size} + a_4 s_{fill}, \quad (3.13)$$

где $a_i \in 0,1$ обозначает, используется или нет мера сходства.

Таким образом, с помощью данного метода мы получаем набор областей для дальнейшего принятия решения о наличии на одной или нескольких найденных областях искомого объекта. На рисунке показаны все найденные области на тестовом изображении:



Рисунок 3.7. Применение выборочной сегментации

Из результатов видно, что интересующая область найдена. В ней содержится статуя целиком. Но также были обнаружены «лишние» области. Следующим шагом является поиск верной области, содержащей объект. Для этого, с использованием координат найденных областей и координат сопоставленных точек, подсчитывается количество таких точек в каждой из областей. Каждая область, содержащая не менее 30 совпадений ключевых точек, считается за искомый объект. Результат этого шага показан на рисунке:



Рисунок 3.8. Найденная область с искомым объектом

Итоговый результат работы программы не содержит вывода сопоставленных точек, в нем содержится только найденная область. Пример такого результата показан на рисунке:



Рисунок 3.9. Результат поиска объекта

Как показало тестирование, для 80% случаев рассмотренный подход полностью работоспособен. Для более сложных случаев, например, при высоком уровне шума, метод сегментации с разделением на 2 сегмента не дает корректных результатов. В этом случае предлагается следующий подход: если после первого вызова k-means и вспомогательного метода ни в одной из найденных областей нет требуемого количества сопоставленных точек, то вызов совершается второй раз, но уже с количеством сегментов, большим на 1. Если ситуация повторяется, то снова увеличиваем количество сегментов и вызываем k-means.

Как показала практика, за максимальное число сегментов можно взять 4, т.е. вызов k-means будет осуществлен не более 3 раз. Такой подход позволяет сократить время обработки изображения в 80% случаев в сравнении с методом Mean Shift. Если в каком-то случае необходимо вызвать k-means дважды, то время обработки получаем аналогичное методу Mean Shift. В случае, если k-means необходимо вызвать трижды, получаем более долгий поиск нежели при использовании метода Mean Shift.

Рассмотрим на примере один из сложных случаев. На рисунке 3.10 показано изображение-шаблон, т.е. тот объект, который необходимо найти.

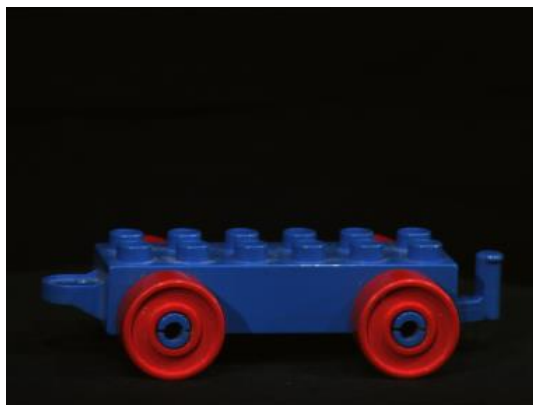


Рисунок 3.10. Изображение-шаблон

На рисунке 3.11 показан результат сегментации исследуемого изображения без наложения шума.

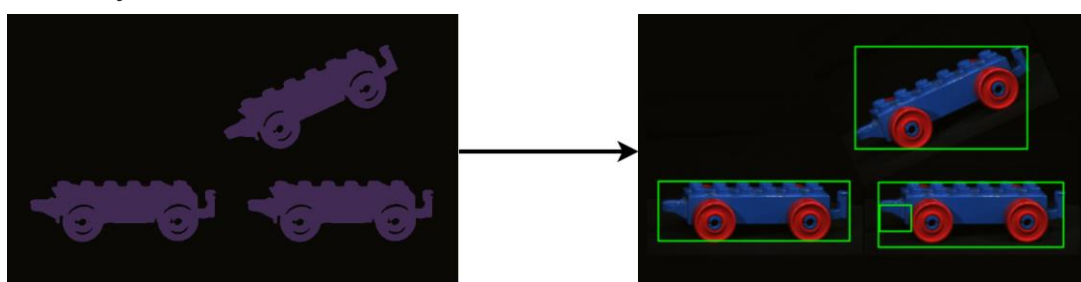


Рисунок 3.11. Результат сегментации без наложения шума, $K=2$

Видно, что в этом случае достаточно разового вызова метода сегментации при количестве сегментов равным 2: все необходимые объекты были выделены. Усложним задачу, наложив на изображение шум, и выполним сегментацию:

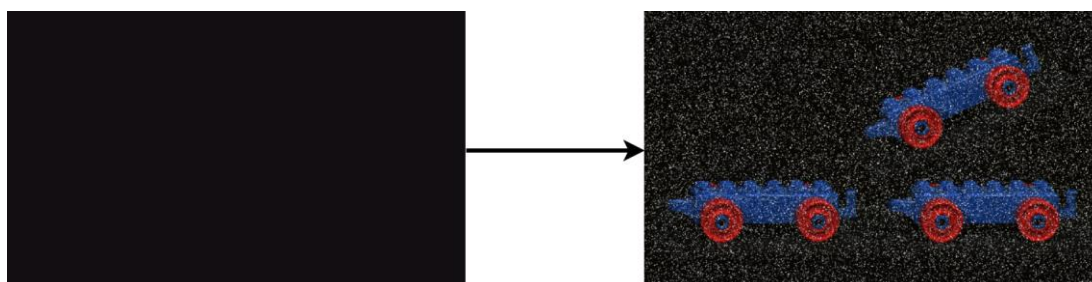


Рисунок 3.12. Результат сегментации с наложением шума, $K=2$

При зашумленном изображении с количеством сегментов равном 2 на нем не было найдено ни одного объекта, соответственно шаг вычисления количества совпавших точек в областях будет пропущен, и происходит переход к следующему вызову метода k-means с количеством сегментов равном 3:

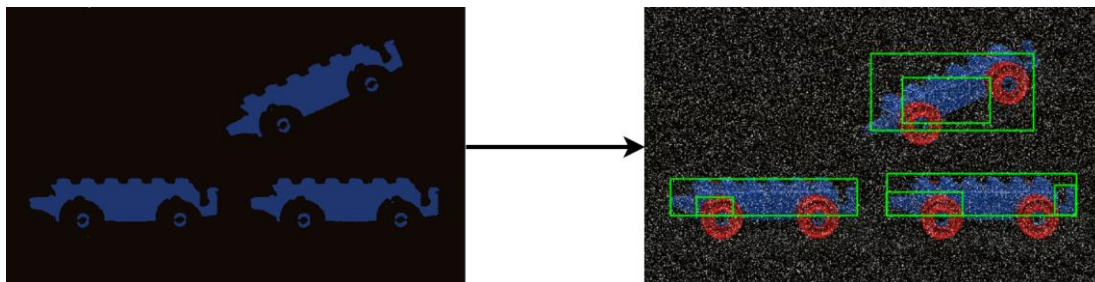


Рисунок 3.13. Результат сегментации с наложением шума, $K=3$

Как можно заметить, в этот раз объекты были найдены, но не целиком, только их часть. Если эти части наберут достаточно количество совпавших точек, то на этом поиск объекта будет завершен. В данном случае этого не происходит и k-means вызывается третий раз:

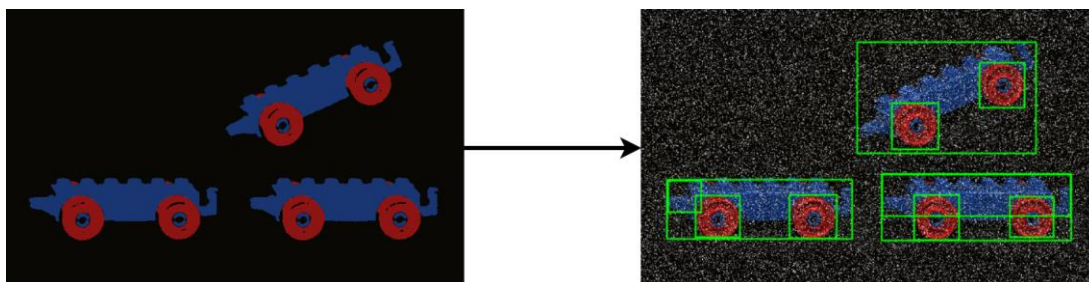


Рисунок 3.14. Результат сегментации с наложением шума, $K=4$

Здесь уже видно, что объекты найдены полностью. После подсчета совпавших точек в найденных областях получаем корректный результат:

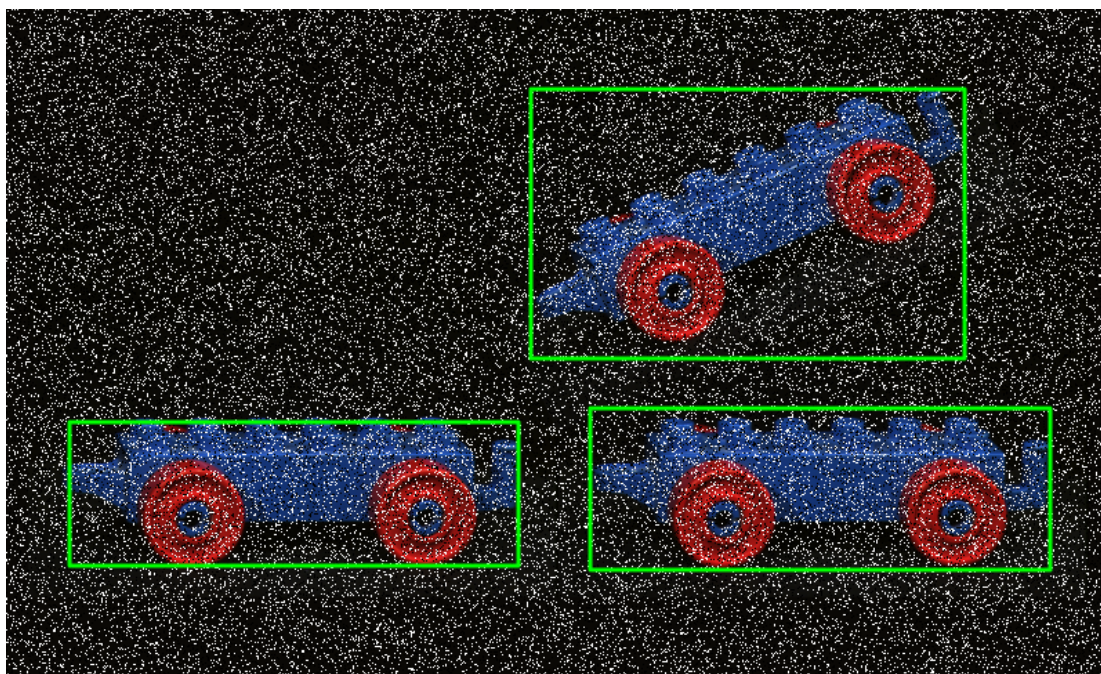


Рисунок 3.12. Результат поиска объектов

Также интересен случай, когда на тестовом изображении нет искомого объекта. Результат поиска такого объекта показан на рисунке:



Рисунок 3.15. Результат сопоставления точек

Как можно отметить, некоторые ключевые точки были сопоставлены как верно найденные, хотя видно, что это не так. Но даже в этом случае объект не будет локализован, т.к. необходимый порог (30 точек в области) не преодолен. Таким образом на рисунке 3.16 можно увидеть, что из всех найденных областей не была выбрана ни одна.

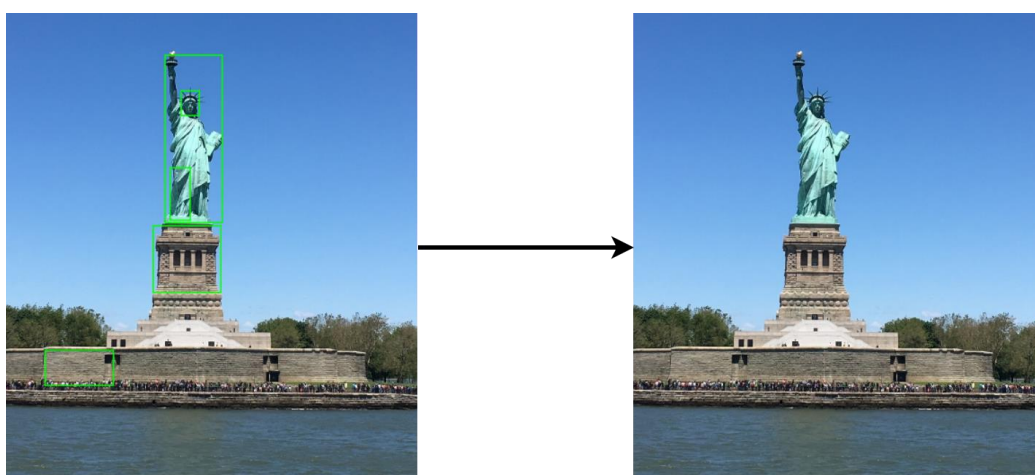


Рисунок 3.16. Результат локализации объекта

Можно видеть, что часть точек вообще указывает за пределы какой-либо области, а оставшихся не хватает для преодоления порога.

3.4. Выводы по главе 3

В данной главе был рассмотрен каждый из этапов обнаружения объекта на изображении с применением теории активного восприятия. Были указаны причины использования вспомогательных методов, а также описаны их алгоритмы.

Рассмотренные в главе 1 методы были исследованы на предмет применения их в разрабатываемой системе, и выбраны наиболее подходящие из них. Представлены графические результаты каждого из этапов, полученные с использованием разработанной по описанным алгоритмам системы.

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		45

Глава 4. Вычислительный эксперимент

Вычислительный эксперимент является очень важным этапом при разработке нового метода. Он позволяет определить работоспособность метода, выявить слабые места реализации, чтобы тщательней проработать их, сравнить разрабатываемый алгоритм с аналогами. В данной главе будут дано описание проводимого вычислительного эксперимента, и рассмотрены его результаты.

4.1. Описание эксперимента

Для проведения эксперимента использовалась база изображений 1000 предметов, созданная в Амстердамском университете – ALOI [35]. Эта база представляет собой открытую коллекцию цветных изображений, собранную в научных целях. Пример типов изображений представлен на рисунке:



Рисунок 4.1. Изображения из базы ALOI

Для каждого объекта в этой базе есть одно или несколько следующих видов изменений:

1. Угол обзора
2. Направление освещения
3. Цвет освещения

Рассмотрим подробнее каждое из изменений.

Направление освещения варьируется в 24 конфигурациях, которые представлены на рисунке:

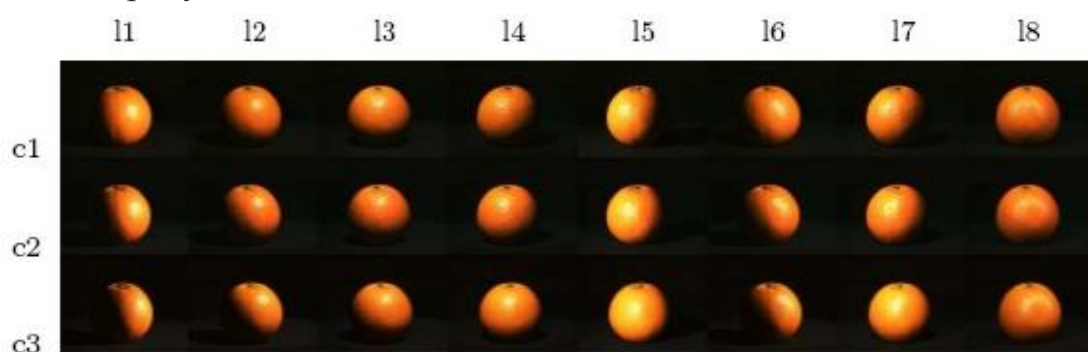


Рисунок 4.2. Изображение с разным направлением освещения

Изображение каждого объекта было получено с включением только одного из пяти источников света, что дало пять различных углов освещения (условия 11-15). При переключении камеры и повороте сцены к этой камере световой луч практически поворачивается на 15 (камера c2) и 30 градусов (камера c3) соответственно. Следовательно, аспект объектов, видимых каждой камерой, идентичен, но направление света сместилось на 15 и 30 градусов по азимуту.

В общей сложности это приводит к 15 различным углам освещения. Кроме того, комбинации освещения были использованы для освещения объекта. Включение двух источников света по бокам объекта дало наклонное освещение справа (условие 16) и слева (условие 17). Включение всех источников света (условие 18) дает своего рода полусферическое освещение, хотя оно ограничено более узким сектором освещения, чем истинное полушарие. Таким образом, в общей сложности было получено 24 различных условия освещения, условия c[1..3]l [1..8].

Цвет освещения варьируется в 12 конфигурациях, которые представлены на рисунке:

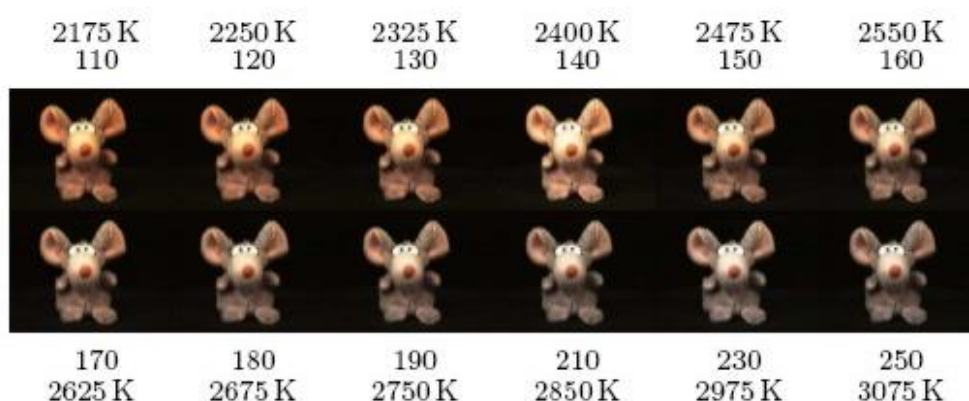


Рисунок 4.3. Изображение с разным цветом освещения

Каждый объект был записан при виде спереди, все пять ламп были включены. Цветовая температура подсветки изменяется с 2175К до 3075К. Камеры были уравновешены белым при 3075К, в результате чего объекты освещались в цвете от красноватого до белого в условиях i110, i120, ..., i250.

Угол обзора варьируется в 72 направлениях, представленных на рисунке:

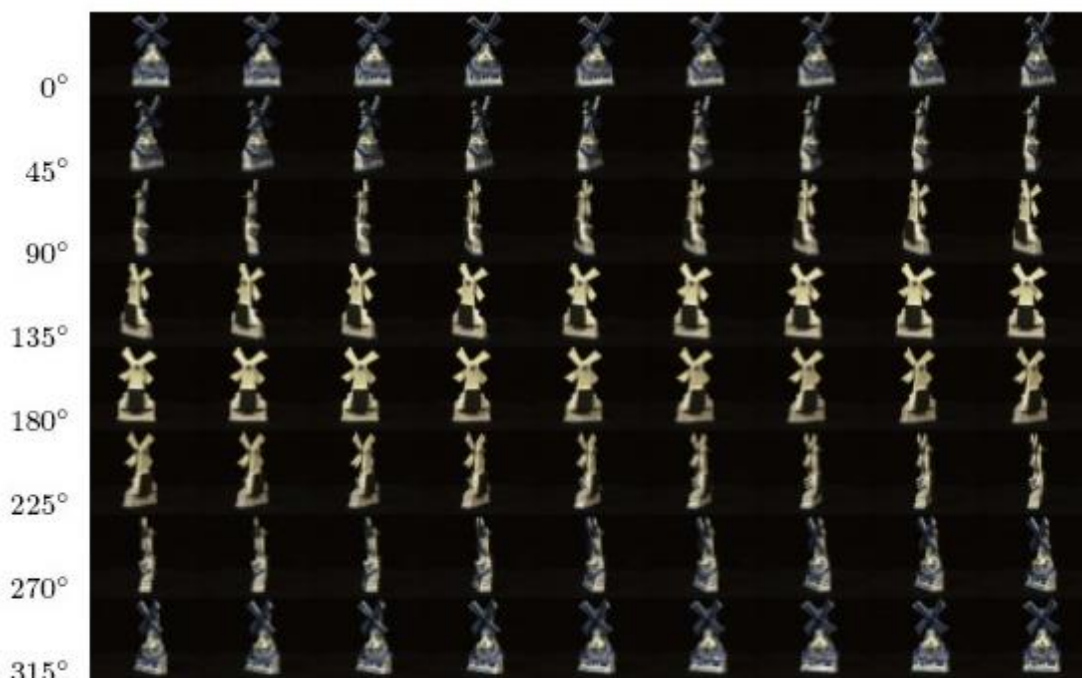


Рисунок 4.4. Изображение с разным углом поворота

Фронтальная камера использовалась для записи 72 аспектов объектов путем вращения объекта в плоскости с шагом 5 градусов. Углы поворота лежат в диапазоне 0..355.

Для проведения тестирования было выбрано 100 объектов с разными типами изменений. В дополнении к этому, с использованием этой базы были получены дополнительные изображения для проверки устойчивости алгоритма к шуму, масштабу и присутствию на исследуемом изображении несколько искомых объектов.

Для создания изображений разного масштаба использовались следующие коэффициенты: 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2.

Для создания шума на изображениях использовался алгоритм под названием «соль-перец», суть которого заключается в случайно возникающих черных и белых пикселей на изображении. Значения вероятностей для создания такого шума были использованы следующие: 0.02, 0.04, 0.06, 0.1, 0.2, 0.4. Пример изображения с наложенным шумом этих вероятностей представлен на рисунке:

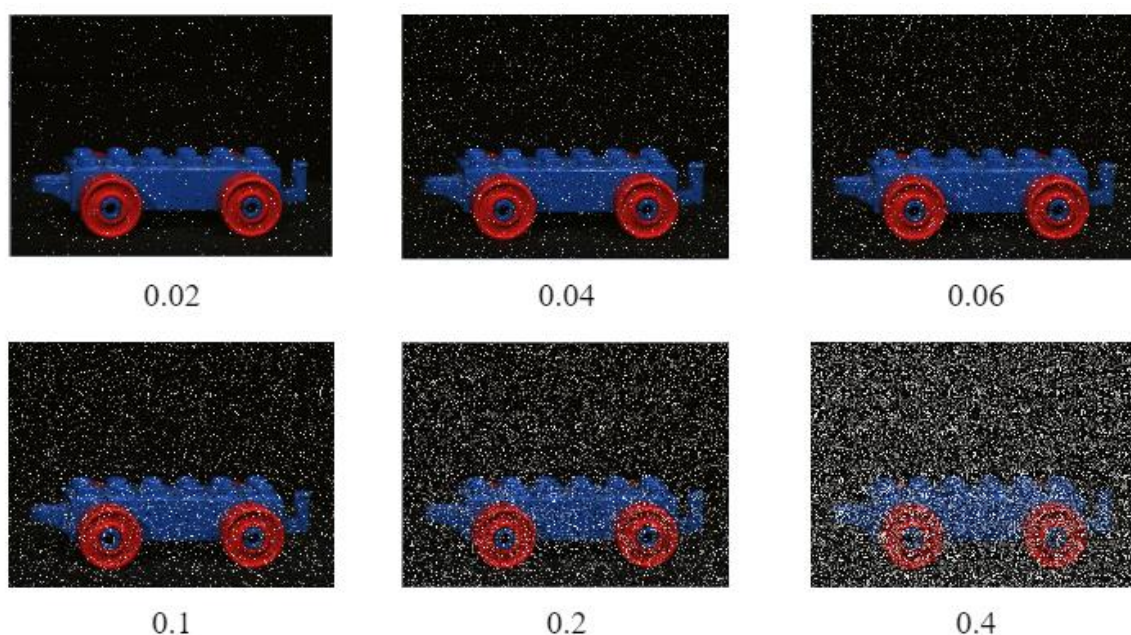


Рисунок 4.5. Изображение с разным уровнем шума

Пример синтезированного изображения с присутствием нескольких искоемых объектов представлен на рисунке:

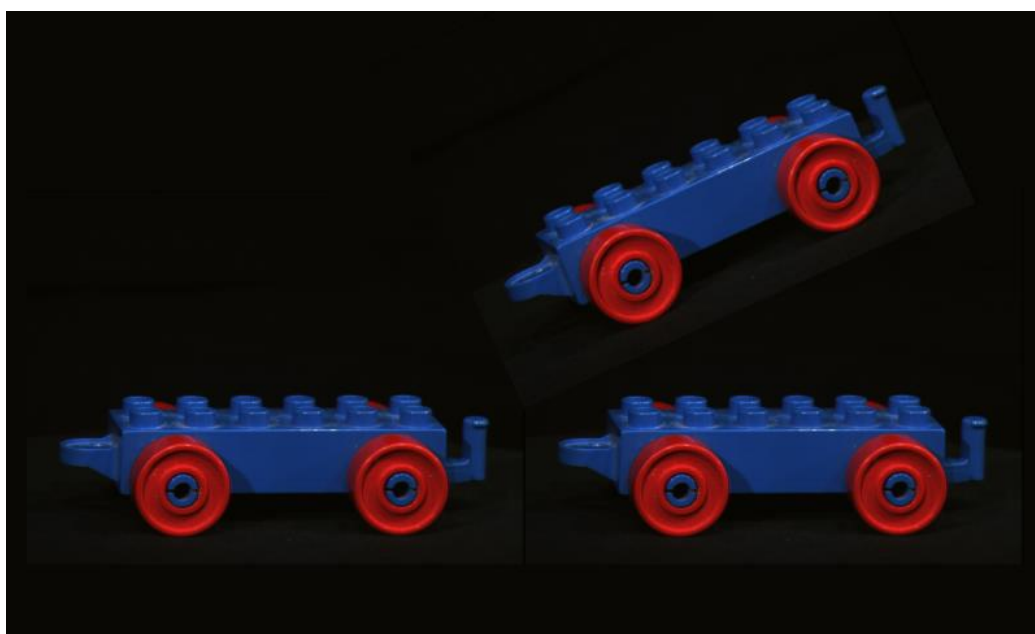


Рисунок 4.6. Изображение с несколькими объектами

Последней частью датасета для тестирования являются реальные изображения. Основная часть - это фотографии памятников, остальное – фотографии обычных предметов, но, в отличии от ALOI, это реальные фотографии с присутствием некоторого фона.

В итоге, получается база изображений 435 предметов, у каждого предмета есть шаблоны и изображения для поиска с разными типами изменений объекта.

4.2. Описание программного продукта

В рамках данной работы был разработан программный продукт на языке программирования Python. Данный язык был выбран по нескольким причинам:

1) Большой выбор сторонних библиотек. Так как язык является одним из самых популярных на сегодняшний день языков программирования, сообществом пишется для него множество сторонних модулей, которые значительно расширяют возможности языка.

2) Простой и четкий синтаксис. Написание программы на языке Python, как правило, занимает гораздо меньше времени, чем, например, на языке C++. Он очень удобен для реализации новых алгоритмов, когда надо проверить, работоспособно ли предполагаемое решение.

3) Кроссплатформенность. Скрипт, написанный на языке Python не имеет привязки к определенной платформе и может быть запущен без изменений как на Windows, так и на Linux.

Рассмотрим основные модули программы, реализующей поиск объекта на изображении.

sum_pixels (area) – функция для подготовки фрагмента изображения к обработке фильтрами Уолша. Делит фрагмент на блоки 4x4 и вычисляет по каждому блоку сумму яркостей, входящих в него пикселей

Входные параметры:

- area – фрагмент изображения, по которому необходимо вычислить сумму яркостей пикселей его блоков.

Выходные параметры:

- parts – список из 16 элементов, содержащий сумму яркостей пикселей каждой из частей, на которые был поделен входной фрагмент

apply_filters (brightness) – функция для вычисления 16 значений фильтров Уолша по уже вычисленным значениям яркости каждого из блоков фрагмента.

Входные параметры:

- brightness – список из 16 значений, содержащих сумму яркостей 16 блоков какого-либо фрагмента изображения.

Выходные параметры:

- full_results – список из 16 значений, содержащий результат вычисления 16 фильтров.

descriptors_calculation (img, template) – функция для вычисления дескрипторов и выделения ключевых точек

Входные параметры:

- **img** – входное изображение (шаблон или тестовое изображение).
- **template** – бинарное значение: 0 – тестовое изображение, 1 - шаблон

Выходные параметры:

- **img_kp** – изображение с найденными ключевыми точками
- **kp** – список с парами координат ключевых точек
- **des** – список с признаковым описанием, которое соответствует найденным ключевым точкам

compare (data_img_1, data_img_2) – функция для сопоставления ключевых точек двух изображений и локализации объекта

Входные параметры:

- **data_img_1** – список из трех элементов для описания изображения-шаблона:
 - Изображение с найденными ключевыми точками
 - Координаты ключевых точек
 - Дескрипторы ключевых точек
- **data_img_2** – список из трех элементов для описания исследуемого изображения:
 - Изображение с найденными ключевыми точками
 - Координаты ключевых точек
 - Дескрипторы ключевых точек

Выходные параметры: вывод на экран и в PNG-файл результата поиска объекта.

В результате работы программы на экран выводится изображение с локализованным объектом:

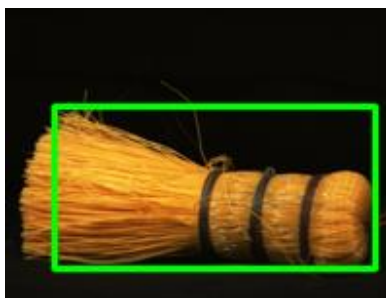


Рисунок 4.7. Результат работы программы: локализованный объект

Также в программе есть возможность включить вывод промежуточных результатов, таких как:

- 1) Все найденные сегменты (рисунок 4.8).
- 2) Результат сопоставления ключевых точек (рисунок 4.9).
- 3) Результат сопоставления ключевых точек и дальнейшая локализация объекта (рисунок 4.10).



Рисунок 4.8. Промежуточный результат: найденные сегменты



Рисунок 4.9. Промежуточный результат: сопоставление точек

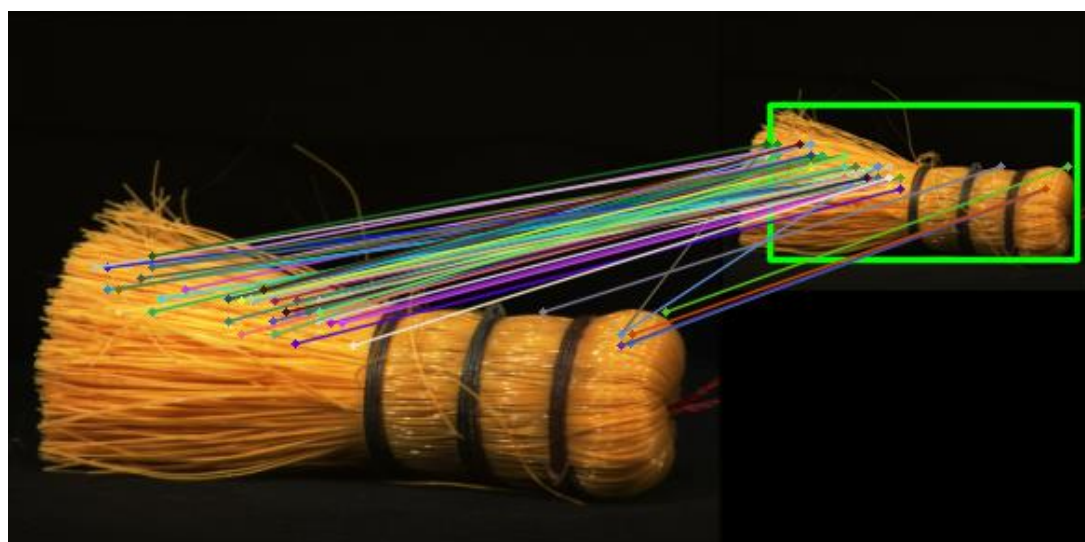


Рисунок 4.10. Промежуточный результат: локализация объекта

4.3. Анализ результатов эксперимента

Разделим результаты в зависимости от вида тестируемого изображения:

1. Изображения из базы ALOI. У тестового изображения относительно изображения-шаблона будет изменено:
 - а) Угол обзора
 - б) Направление освещения
 - в) Цвет освещения
2. Изображения, созданные с использованием базы ALOI:
 - а) Несколько искомых объектов на исследуемом изображении
 - б) Изображения с наложенным шумом
3. Реальные изображения. У тестового изображения относительно изображения-шаблона будет изменено:
 - а) Угол обзора
 - б) Освещение
 - в) Добавлен шум

В таблицах будут представлены результаты для разных значений параметров, которые используется в разрабатываемой системы. Для каждого типа тестируемых изображений будут также приведены результаты метода SIFT для сравнения точности и времени работы. По итогу будет сделан вывод, какие значения параметров дают наилучший результат.

Приведем описание этих параметров:

- $k_{кр}$ – коэффициент отбора ключевых точек; чем меньше значение, тем больше на изображении будет определено ключевых точек
- k_m – коэффициент отбора дескрипторов; чем больше значение, тем больше будет определено совпадений
- n – размер скользящего окна

Для каждой комбинации параметров будет указана точность локализации, а также затрачиваемое на локализацию время. Комбинации параметров, которые дают точность менее 80% были исключены из рассмотрения.

Результаты поделены на шесть видов тестируемых изображений:

1. Изображения из базы ALOI. Изменение угла обзора.
2. Изображения из базы ALOI. Изменение направления освещения.
3. Изображения из базы ALOI. Изменение цвета освещения.

4. Синтезированные изображения на основе изображений из базы ALOI. Тестовое изображение имеет два и более искомого объекта, которые могут быть повернуты, наклонены, а также иметь измененный угол обзора или масштаб.
5. Синтезированные изображения на основе изображений из базы ALOI. В дополнении к предыдущему виду тестирования накладывается шум.
6. Реальные изображения. В тестовых изображениях может быть изменен угол обзора, освещение, масштаб, а также присутствовать шум.

Рассмотрим результаты тестирования первого вида изображений (табл.4.1). Отдельные объекты на однородном фоне, у тестового изображения изменен угол обзора. Значение угла обзора лежат в диапазоне -30..30 градусов.

Таблица 4.1. Результаты тестирования первого вида изображений

$k_{кр}$	k_m	n	Точность	Время
0.5	1.4	16	90.00%	0.25 сек.
		32	86.67%	0.16 сек.
	1.6	16	96.67%	0.27 сек.
		32	93.33%	0.17 сек.
	1.8	16	93.33%	0.28 сек.
		32	90.00%	0.20 сек.
	2.0	16	93.33%	0.29 сек.
		32	86.67%	0.20 сек.
0.6	1.4	16	90.00%	0.20 сек.
		32	86.67%	0.11 сек.
	1.6	16	93.33%	0.21 сек.
		32	93.33%	0.11 сек.
	1.8	16	96.67%	0.23 сек.
		32	93.33%	0.12 сек.
	2.0	16	90.00%	0.24 сек.
		32	93.33%	0.13 сек.

Данный блок содержит 60 изображений, наилучшая точность локализации - 96.67%, это соответствует двум ошибкам. Наилучшее время обработки одного изображения – 0.23 секунды. Из результатов можно сделать вывод, что метод устойчив к изменению угла обзора объекта и может корректно его определить его в 58 случаях из 60. Результат, полученный с использованием SIFT, дает точность 98.3%, он корректно определяет объекты в 59 случаях из 60, среднее время – 0.12 секунд.

В дополнении к этому типу тестирования рассмотрим еще 80 изображений, которые соответствуют 10 объектам, у каждого из которых угол поворота изменяется в диапазоне -40..40 градусов с шагом 10. Результаты приведены в таблице:

Таблица 4.2. Результаты дополнительного тестирования второго вида изображений: угол обзора

Угол обзора	-40	-30	-20	-10	10	20	30	40
Точность	80%	90%	100%	100%	100%	100%	90%	70%

Исходя из этих результатов, можно сделать вывод, что изменение угла обзора не больше чем на ± 20 градусов, приводит к верно найденному объекту в 40 случаях из 40. Изменение угла обзора на ± 30 градусов дает уже менее точные результаты – верно найденных объектов 18 из 20. Изменение угла на ± 40 градусов дает еще менее точные результаты – 15 объектов из 20.

Таким образом, можно сказать, что разрабатываемая система поддерживает изменение угла поворота объекта не более чем на ± 30 градусов, что дает точность 96.67%. При угле поворота меньше или равному ± 20 градусам – система имеет точность 100%.

Метод SIFT имеет результаты, представленные в таблице:

Таблица 4.3. Результаты дополнительного тестирования второго вида изображений: угол обзора. Метод SIFT

Угол обзора	-40	-30	-20	-10	10	20	30	40
Точность	80%	100%	100%	100%	100%	100%	90%	70%

Существующий метод на таком же наборе изображений дает 100% точность при изменении угла обзора на ± 10 , ± 20 и -30 градусов. При изменении на $+30$

градусов метод совершил одну ошибку – распознал объект в 9 случаях из 10. Для ± 40 результаты аналогичны разрабатываемому методу – 15 объектов из 20.

Результаты тестирования второго вида изображений представлены в таблице 4.4. Отдельные объекты на однородном фоне, у тестового изображения изменено направление освещения.

Таблица 4.4. Результаты тестирования второго вида изображений

$k_{кр}$	k_m	n	Точность	Время
0.5	1.4	16	98.00%	0.24 сек.
		32	98.00%	0.15 сек.
	1.6	16	100.00%	0.26 сек.
		32	100.00%	0.17 сек.
	1.8	16	100.00%	0.28 сек.
		32	96.00%	0.21 сек.
	2.0	16	96.00%	0.27 сек.
		32	94.00%	0.20 сек.
0.6	1.4	16	92.00%	0.21 сек.
		32	90.00%	0.12 сек.
	1.6	16	100.00%	0.19 сек.
		32	92.00%	0.11 сек.
	1.8	16	100.00%	0.20 сек.
		32	96.00%	0.11 сек.
	2.0	16	96.00%	0.18 сек.
		32	92.00%	0.10 сек.

Данный блок содержит 50 изображений, наилучшая точность локализации - 100.00%. Наилучшее время обработки одного изображения – 0.17 секунд. Из результатов можно сделать вывод, что метод полностью устойчив к изменению освещению объекта и может корректно его определить его в 50 случаях из 50. Результат, полученный с использованием SIFT, также дает точность 100%, он безошибочно определяет объекты в 50 случаях из 50, среднее время – 0.11 секунд.

Результаты тестирования третьего вида изображений представлены в таблице 4.3. Отдельные объекты на однородном фоне, у тестового изображения изменен цвет освещения.

Таблица 4.3. Результаты тестирования третьего вида изображений

$k_{кр}$	k_m	n	Точность	Время
0.5	1.4	16	100.00%	0.25 сек.
		32	97.78%	0.15 сек.
	1.6	16	100.00%	0.24 сек.
		32	100.00%	0.16 сек.
	1.8	16	100.00%	0.24 сек.
		32	100.00%	0.18 сек.
	2.0	16	97.78%	0.23 сек.
		32	95.56%	0.17 сек.
0.6	1.4	16	97.78%	0.23 сек.
		32	95.56%	0.13 сек.
	1.6	16	100.00%	0.20 сек.
		32	97.78%	0.12 сек.
	1.8	16	100.00%	0.20 сек.
		32	100.00%	0.11 сек.
	2.0	16	100.00%	0.19 сек.
		32	97.78%	0.09 сек.

Данный блок содержит 45 изображений, наилучшая точность локализации - 100.00%. Наилучшее время обработки одного изображения – 0.11 секунд. Из результатов можно сделать вывод, что метод полностью устойчив к изменению цвета освещения объекта и может корректно его определить его в 45 случаях из 45. Результат, полученный с использованием SIFT, также дает точность 100%, он безошибочно определяет объекты в 45 случаях из 45, среднее время – 0.10 секунд.

Результаты тестирования четвертого вида изображений представлены в таблице 4.4. Отдельные объекты на однородном фоне, тестовое изображение имеет 2 и более искомых объекта, которые могут быть повернуты, наклонены, а также иметь измененный угол обзора или масштаб. Для этого тестирования представим результаты в ином виде – разделим столбец Точность на два:

- Точность (один объект) – столбец показывает процент случаев, в которых верно был найден хотя бы один объект на тестовом изображении
- Точность (все объекты) – столбец показывает процент случаев, в которых верно были найдены все объекты на тестовом изображении

Таблица 4.4. Результаты тестирования четвертого вида изображений

$k_{кр}$	k_m	n	Точность (один объект)	Точность (все объекты)	Время
0.5	1.4	16	100.00%	83.33%	0.25 сек.
		32	96.67%	86.67%	0.14 сек.
	1.6	16	100.00%	93.33%	0.23 сек.
		32	100.00%	90.00%	0.12 сек.
	1.8	16	100.00%	93.33%	0.24 сек.
		32	100.00%	86.67%	0.15 сек.
	2.0	16	96.67%	86.67%	0.22 сек.
		32	93.33%	83.33%	0.16 сек.
0.6	1.4	16	100.00%	93.33%	0.21 сек.
		32	96.67%	90.00%	0.14 сек.
	1.6	16	100.00%	93.33%	0.20 сек.
		32	100.00%	83.33%	0.14 сек.
	1.8	16	100.00%	90.00%	0.19 сек.
		32	96.67%	86.67%	0.12 сек.
	2.0	16	96.67%	86.67%	0.19 сек.
		32	93.33%	83.33%	0.10 сек.

Данный блок содержит 30 изображений, наилучшая точность локализации хотя бы одного объекта - 100.00%, всех объектов – 96.67%, что соответствует одной ошибке. Наилучшее время обработки одного изображения – 0.20 секунд. Из результатов можно сделать вывод, что метод может корректно определять 2 и более искомых объектов в 44 случаев из 45. Метод SIFT определяет один объект с точностью 100.00%, но он не может определить 2 и более объекта, поэтому здесь его точность – 0.00%. Пример работы метода SIFT с несколькими искомыми объектами:

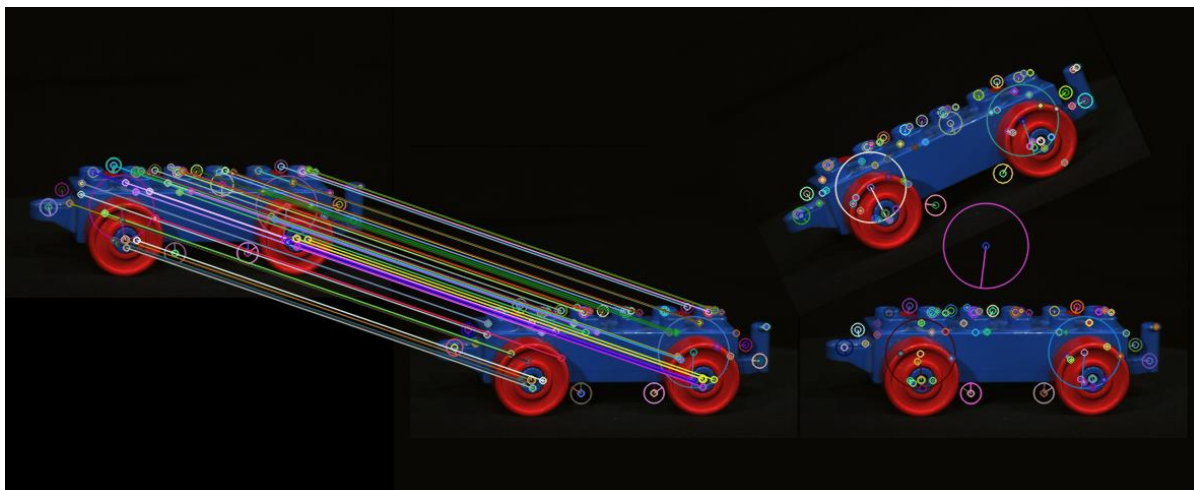


Рисунок 4.11. Результат работы метода SIFT с несколькими объектами

Метод SIFT корректно нашел ключевые точки на всех 3 объектах, но на этапе сопоставления дескрипторов, он сопоставил только максимально похожие. Для сравнения на рисунке 4.12 показан результат работы разрабатываемого метода на том же тестовом изображении.

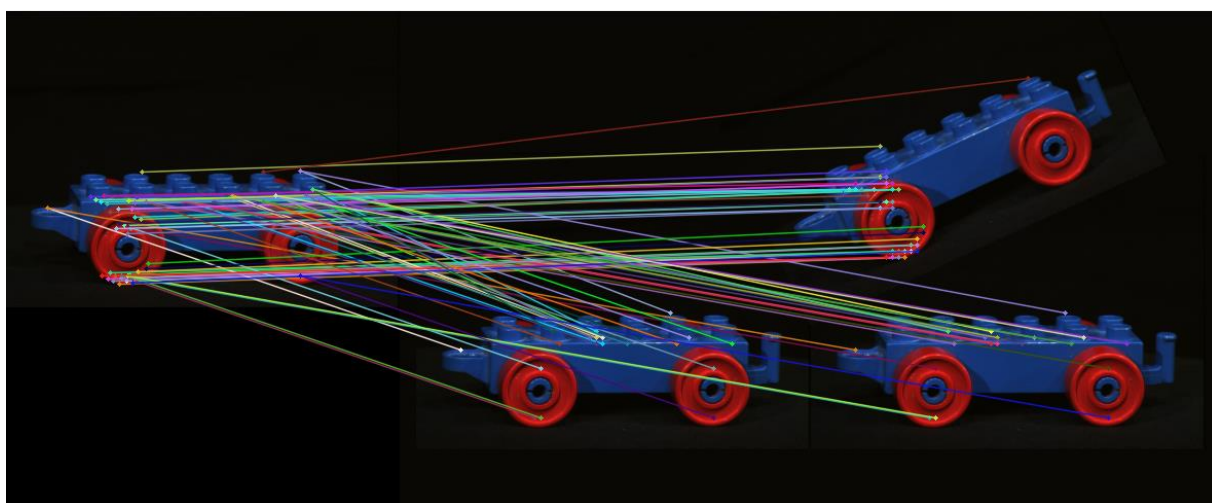


Рисунок 4.12. Результат работы разрабатываемого метода с несколькими объектами

Результаты тестирования пятого вида изображений представлены в таблице 4.5. Отдельные объекты на однородном фоне, тестовое изображение имеет 2 и более искомым объекта, которые могут быть повернуты, наклонены, а также иметь измененный угол обзора или масштаб. Дополнительно на тестовое изображение накладывается шум.

Таблица 4.5. Результаты тестирования пятого вида изображений

$k_{кр}$	k_m	n	Точность (один объект)	Точность (все объекты)	Время
0.5	1.4	16	93.33%	83.33%	0.24 сек.
		32	86.67%	80.00%	0.13 сек.
	1.6	16	86.67%	86.67%	0.24 сек.
		32	83.33%	83.33%	0.12 сек.
	1.8	16	93.33%	86.67%	0.23 сек.
		32	96.67%	90.00%	0.12 сек.
	2.0	16	86.67%	86.67%	0.22 сек.
		32	83.33%	80.00%	0.11 сек.
0.6	1.4	16	93.33%	86.67%	0.23 сек.
		32	93.33%	83.33%	0.12 сек.
	1.6	16	96.67%	90.00%	0.22 сек.
		32	93.33%	86.67%	0.12 сек.
	1.8	16	93.33%	83.33%	0.20 сек.
		32	86.67%	86.67%	0.10 сек.
	2.0	16	86.67%	86.67%	0.19 сек.
		32	83.33%	83.33%	0.08 сек.

Данный блок содержит 30 изображений, наилучшая точность локализации хотя бы одного объекта - 96.67%, всех объектов – 93.33%, что соответствует двум ошибкам. Наилучшее время обработки одного изображения – 0.12 секунд. Из результатов можно сделать вывод, что метод может корректно определять изображения, искаженные шумом, в 28 случаях из 30. Метод SIFT оказался очень

слабо устойчив к шуму, он определяет один объект с точностью 38.33%, и он также не может определить 2 и более объекта, поэтому здесь его точность – 0.00%.

В дополнении к этому виду тестирования рассмотрим еще 80 изображений, которые соответствуют 10 объектам, каждый из которых имеет следующие вероятности наложенного шума «соль-перец»: 0.02, 0.04, 0.06, 0.1, 0.2, 0.4, 0.5, 0.6. Результаты приведены в таблице:

Таблица 4.6. Результаты дополнительного тестирования пятого вида изображений: шум

Вероятность	0.02	0.04	0.06	0.1	0.2	0.4	0.5	0.6
Точность	100%	100%	100%	100%	90%	90%	60%	40%

Из таблицы видно, что вплоть до вероятности шума, равной 10%, точность определения объекта составляет 100%, вероятности 20% и 40% дают точность 90%, что соответствует 9 найденным объектам из 10. Начиная с вероятности 50% точность начинает резко падать и показывает 60% при вероятности шума 50% и 40% точности при вероятности шума 60%.

Результаты метода SIFT для такого же набора изображений показаны в таблице:

Таблица 4.7. Результаты дополнительного тестирования пятого вида изображений: шум. Метод SIFT

Вероятность	0.02	0.04	0.06	0.1	0.2	0.4	0.5	0.6
Точность	60%	50%	40%	40%	20%	10%	0%	0%

Таким образом, можно сделать вывод, что метод SIFT нельзя применять в ситуациях, когда есть даже небольшой шум. При вероятности шума 2% он уже показывает точность 60%, при вероятности 4% точность снижается до 50% и с повышением уровня шума только падает. Вероятно, причина этому – повышение контрастности на изображении на этапе предварительной обработки данных и отсутствие применения алгоритмов, понижающих шум.

Рассмотрим еще один дополнительный блок для пятого вида изображений, который включает в себя 80 изображений, которые соответствуют 10 объектам, каждый из которых имеет следующие коэффициенты масштабирования: 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2. Результаты приведены в таблице:

Таблица 4.8. Результаты дополнительного тестирования пятого вида изображений: масштаб

Масштаб	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
Точность	100%	100%	100%	100%	100%	100%	90%	50%

Исходя из полученных данных, можно сделать вывод, что алгоритм хорошо инвариантен к масштабу. С учетом того, что дескрипторы изображения-шаблона вычисляются только на трех масштабах исходного изображения (0.8, 0.6, 0.4), алгоритм также безошибочно детектирует объекты с коэффициентами масштаба 0.9, 0.7, 0.5. Для коэффициента равного 0.3 система допустила одну ошибку, для коэффициента 0.2 получили точность всего 50%, с настолько уменьшенным изображением система уже справиться не может.

Результаты метода SIFT для такого же набора изображений показаны в таблице:

Таблица 4.9. Результаты дополнительного тестирования пятого вида изображений: масштаб. Метод SIFT

Масштаб	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
Точность	100%	100%	100%	100%	100%	100%	100%	80%

Таким образом, получаем, что метод SIFT полностью инвариантен к масштабу. Его точность для коэффициентов от 0.9 до 0.3 составляет 100%, с коэффициентом масштабирования 0.2 метод корректно определил 8 объектов из 10, в то время как разрабатываемый «справился» всего с 5 объектами.

Результаты тестирования последнего, шестого, вида изображений представлены в таблице 4.10. В эту группу объединены все реальные тестовые изображения. В тестовых изображениях может быть изменен угол обзора, освещение, масштаб, а также присутствовать шум.

Таблица 4.10. Результаты тестирования шестого вида изображений

k_{kp}	k_m	n	Точность	Время
0.5	1.4	16	92.50%	0.25 сек.
		32	90.00%	0.17 сек.
	1.6	16	87.50%	0.23 сек.
		32	92.50%	0.16 сек.
	1.8	16	95.00%	0.24 сек.
		32	85.00%	0.19 сек.
	2.0	16	92.50%	0.22 сек.
		32	87.50%	0.17 сек.
0.6	1.4	16	92.50%	0.24 сек.
		32	95.00%	0.14 сек.
	1.6	16	92.50%	0.21 сек.
		32	92.50%	0.13 сек.
	1.8	16	95.00%	0.21 сек.
		32	92.50%	0.12 сек.
	2.0	16	92.50%	0.19 сек.
		32	87.50%	0.12 сек.

Данный блок содержит 40 изображений, наилучшая точность локализации - 95.00%, что соответствует двум ошибкам. Наилучшее время обработки одного изображения – 0.14 секунд. Из результатов можно сделать вывод, что метод локализовать объекты в реальных условиях в 38 случаев из 40. Результат, полученный с использованием SIFT, дает точность 85.0%, он безошибочно определяет объекты в 34 случаев из 40, среднее время – 0.10 секунд. Если рассматривать результаты метода SIFT без зашумленных изображений, то его точность равна 96.67%.

Проанализируем полученные результаты для выявления параметров, наиболее подходящих для каждого конкретного случая.

Для измененного угла обзора наилучшая точность была получена со следующими комбинациями параметров:

- $k_{kp} - 0.5, k_m - 1.6, n - 16.$
- $k_{kp} - 0.6, k_m - 1.8, n - 16.$

Для измененного направления освещения наилучшая точность была получена со следующими комбинациями параметров:

- $k_{kp} - 0.5, k_m - 1.6, n - 16.$
- $k_{kp} - 0.5, k_m - 1.6, n - 32.$
- $k_{kp} - 0.5, k_m - 1.8, n - 16.$
- $k_{kp} - 0.6, k_m - 1.6, n - 16.$
- $k_{kp} - 0.6, k_m - 1.8, n - 16.$

Для измененного цвета освещения наилучшая точность была получена со следующими комбинациями параметров:

- $k_{kp} - 0.5, k_m - 1.4, n - 16.$
- $k_{kp} - 0.5, k_m - 1.6, n - 16.$
- $k_{kp} - 0.5, k_m - 1.6, n - 32.$
- $k_{kp} - 0.5, k_m - 1.8, n - 16.$
- $k_{kp} - 0.5, k_m - 1.8, n - 32.$
- $k_{kp} - 0.6, k_m - 1.6, n - 16.$
- $k_{kp} - 0.6, k_m - 1.8, n - 16.$
- $k_{kp} - 0.6, k_m - 1.8, n - 32.$
- $k_{kp} - 0.6, k_m - 2.0, n - 16.$

Для синтезированных случаев, которые содержат несколько искомых объектов, каждый из которых может быть повернут, наклонен, а также иметь измененный угол обзора или масштаб наилучшая точность была получена со следующими комбинациями параметров:

- $k_{kp} - 0.5, k_m - 1.6, n - 16.$
- $k_{kp} - 0.5, k_m - 1.8, n - 16.$
- $k_{kp} - 0.6, k_m - 1.4, n - 16.$
- $k_{kp} - 0.6, k_m - 1.6, n - 16.$

Для синтезированных случаев, аналогичным предыдущим, но с наложенным шумом, наилучшая точность была получена со следующими комбинациями параметров:

- $k_{kp} - 0.5, k_m - 1.8, n - 32$.
- $k_{kp} - 0.6, k_m - 1.6, n - 16$.

Для реальных случаев наилучшая точность была получена со следующими комбинациями параметров:

- $k_{kp} - 0.5, k_m - 1.8, n - 16$.
- $k_{kp} - 0.6, k_m - 1.4, n - 32$.
- $k_{kp} - 0.6, k_m - 1.8, n - 16$.

Таким образом, для каждого конкретного случая были выявлены параметры, дающие наиболее высокую точность. Суммируя полученные результаты, можно определить параметры, которые дают лучшие результаты в большинстве случаев. Такими параметрами являются:

- $k_{kp} - 0.5, k_m - 1.8, n - 16$.

Среднее время обработки изображения – 0.24 секунды.

Анализируя результаты, полученные с использованием метода SIFT, можно сделать следующий вывод: алгоритм работает лучше разрабатываемого в нормальных условиях: он дает 100% точность при изменении угла поворота, направления и цвета освещения для изображений из базы и 96.67% точность для реальных изображений. Но если усложнить ему задачу, наложив шум, то его точность резко снижается до 63.33%. Также алгоритм не может работать с несколькими объектами: если на тестовом изображении есть 2 и более искомых объекта, метод локализует только один. Но при этом время работы метода SIFT вдвое меньше времени разрабатываемого метода (0.24 секунд против 0.10 секунд).

4.4. Выводы по главе 4

В данной главе было рассмотрено описание вычислительного эксперимента, проводимого с целью определить точность и время работы разрабатываемого метода, а также анализ его результатов и сравнение с наиболее точным из существующих методов. Было приведено подробное описание базы изображений, которая использовалась для тестирования, то, как она была модифицирована для проведения наиболее полного тестирования. В главе также был описан программный продукт, реализующий алгоритм поиска объектов на изображении, рассмотрены результаты его работы.

При анализе результатов тестирования были выбраны параметры, дающие наибольшую точность и наименьшее время работы. Полученная точность при сравнении с точностью существующих методов показала, что в нормальных условиях разрабатываемый метод работает немного хуже (– 3-5%). При наличии шума он дает намного большую точность (+ 30%). При наличии 2 и более искомых объектов на тестовом изображении существующий метод может локализовать только один, в то время как разрабатываемый метод с точностью 96.67% определит их все.

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		66

Заключение

В ходе выполнения выпускной квалификационной работы было произведено исследование существующих методов для поиска объекта на изображении, а также исследование методов, применяемых на каждом из этапов поиска. В рамках этого обзора методы были проанализированы, выделены их достоинства и недостатки, изучены проблемы, с которыми сталкивались авторы методов и люди, использующие их.

Основываясь на полученной информации был разработан новый алгоритм решения задачи поиска объекта на изображении, который заключается в новой комбинации подходов для решения задачи на разных ее этапах.

Для реализации метода была разработана программа на языке Python, были найдены и подготовлены тестовые изображения разной сложности для проверки работоспособности системы. В результате проведения экспериментов на этих тестовых изображениях и их анализа, был сделан вывод о корректной работе метода при решении задачи обнаружения объекта на изображении.

Точность работы системы в нормальных условиях оказалось сопоставимой с точностью существующих методов. Точность работы на зашумленных изображениях оказалась значительно выше, а также разработанный метод может поддерживать случаи с двумя и более искомыми объектами на изображении, локализуя их все, что существующие методы не поддерживают.

Таким образом, можно сделать вывод, что разработанный метод можно использовать на практике. Задача, поставленная в начале выполнения выпускной квалификационной работы, была выполнена в полном объеме, а цели достигнуты.

Список литературы

1. Oji R., Tajeripour F. Full Object Boundary Detection by Applying Scale Invariant Features in a Region Merging Segmentation Algorithm. // International Journal of Artificial Intelligence & Applications (IJAIA). – 2012. - Т. 3, №5. – С. 41-50.
2. Guoshen Yu, and Jean-Michel Morel, ASIFT: An Algorithm for Fully Affine Invariant Comparison, Image Processing On Line, 1 (2011), pp. 11–38
3. Д. Мирамонте-Харамилло, В.Х. Диас-Рамирес, В.И. Кобер, В.Н. Карнаухов. Алгоритм сопоставления изображений на основе скользящих гистограмм направленных градиентов. Электронный научный журнал: Информационные процессы, Том 14, № 1, 2014, стр. 56–63.
4. Писаревский А. Н. и др. Системы технического зрения (принципиальные основы, аппаратное и математическое обеспечение). Л.: Машиностроение, 1988. 424 с.
5. Мошкин В. И., Петров А. А., Титов В. С., Якушенков Ю. Г. Техническое зрение роботов. М.: Машиностроение, 1990. 272 с.
6. Анисимов Б. В., Курганов В. Д., Злобин В. К. Распознавание и цифровая обработка изображений: Учеб. пособие. М.: Высш. школа, 1983. 295 с.
7. Фурман Я. А. Цифровые методы обработки и распознавания бинарных изображений. Красноярск: Изд-во Красноярского ун-та, 1992. 248 с.
8. Фисенко В. Т., Фисенко Т. Ю. Компьютерные обработки и распознавание изображений: Учеб. пособие. СПб: СПбГУ ИТМО, 2008. 192 с.
9. Сойфер В. А.. Методы компьютерной обработки изображений. М.: Физматлит, 2003. 784 с.
10. Otsu N. A threshold selection method from gray-level histograms // IEEE Trans. Syst., Man. and Cybern. 1979. Vol. SMC-9. P. 62—66.
11. Shi J. and Malik J. Normalized Cuts and Image Segmentation // Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1997. P. 731—737.

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		68

12. Kanungo T., Mount D. M., Netanyahu N., Piatko C., Silverman R. and Wu A Y. An efficient k-means clustering: Analysis and implementation // Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2002. P. 881—892.
13. RLBP: Robust Local Binary Pattern / J. Chen [и др.] // Proceedings of the British Machine Vision Conference. — BMVA Press, 2013.
14. Sharma G. Local Higher-Order Statistics (LHS) for Texture Categorization and Facial Analysis / G. Sharma, S. ul Hussain, F. Jurie // Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII / под ред. A. Fitzgibbon [и др.]. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. — С. 1—12.
15. Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features. Computer Vision Image Understanding, 2008, vol. 110, no. 3, pp. 346–359.
16. Khan, N. SIFT and SURF Performance Evaluation Against Various Image Deformations on Benchmark Dataset / N. Khan, B. McCane, G. Wyvill // Proc. of Digital Image Computing Techniques and Applications (DICTA). - 2011. - P. 501-506.
17. Calonder M., Lepetit V., Strecha C., Fua P. BRIEF: Binary Robust Independent Elementary Features // Proc. European Conference on Computer Vision. – 2010. – P. 778–792.
18. M. Minsky and S. Papert. Perceptrons. MIT Press, Cambridge, MA, 1969.
19. Martin A. Fischler and Robert C. Bolles (June 1981). “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. Comm. Of the ACM. 24: 381—395.
20. Шапиро Л., Стокман Дж. Компьютерное зрение. – М.: Бином. Лаборатория знаний, 2006. – 752 с.
21. Zhang Y. Advances in Image And Video Segmentation. – USA: IRM Press., 2006.

					ВКР-НГТУ-09.04.01-(М17-ИВТ-3)-005-2019(ПЗ)	Лист
Изм.	Лист	№ докум.	Подп.	Дата		69

22. Comaniciu D., Meer P. Mean shift: A robust approach towards feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence, 24: 603–619, 2002.
23. Kim, E., Wang, W., Li, H., Huang, X.: A parallel annealing method for automatic color cervigram image segmentation. In: Medical Image Computing and Computer Assisted Intervention, MICCAI-GRID 2009 HPC Workshop (2009)
24. Вежневц А., Баринаова О. Методы сегментации изображений: автоматическая сегментация // Компьютерная графика и мультимедиа. 2006. № 4.
25. Утробин В.А. Элементы теории активного восприятия изображений // Труды Нижегородского государственного технического университета им. Р.Е. Алексеева. - 2010. - Т. 81, №2. - С. 61-69.
26. Утробин В. А. Информационные модели системы зрительного восприятия для задач компьютерной обработки изображений. — Нижний Новгород: НГТУ им. Р. Е. Алексеева, 2001. — С. 234.
27. Утробин В. А. Компьютерная обработка изображений. Анализ и синтез. — Н. Новгород: НГТУ, 2003. — С. 228. 63 Инв. № подл Подп. и дата Взам. инв. № Инв. № дубл. Подп. и дата Лист ВКР-НГТУ-09.04.01-(М14 ИВТ-3)-001-2016 (ПЗ) Изм. № докум. Подп. Дат
28. Утробин В. А. Компьютерная обработка изображений. Принятие решений в пространстве эталонов. — Нижний Новгород: НГТУ им. Р. Е. Алексеева, 2004. — С. 221.
29. Утробин В. А. Физические интерпретации элементов алгебры изображения // Успехи физических наук. — 2004. — Т. 174, № 10. — С. 1089–1104.
30. Утробин В. А. Компьютерная обработка изображений. Информационные модели этапа понимания. — Н. Новгород: НГТУ, 2006. — С. 247.

31. Утробин В. А. Элементы теории активного восприятия изображений // Труды Нижегородского государственного технического университета им. Р.Е. Алексеева. — 2010. — Т. 81, № 2. — С. 61–69.
32. Recommendation ITU-T T.871 (05/2011), International Telecommunication Union – Information technology – Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)
33. Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.
34. P. Felzenszwalb, D. Huttenlocher, Efficient Graph-Based Image Segmentation // International Journal of Computer Vision, Vol. 59, No. 2, September, 2004
35. J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, The Amsterdam library of object images, Int. J. Comput. Vision, 61(1), 103-112, January, 2005