



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование кафедры)

УТВЕРЖДАЮ  
Завкафедрой «Прикладная  
математика и информатика»

\_\_\_\_\_ А.В. Очеповский  
(подпись) (И.О. Фамилия)  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**  
**на выполнение бакалаврской работы**

Студент Ватолин Владимир Алексеевич

1. Тема Разработка алгоритма для локализации движущихся объектов на изображении

2. Срок сдачи студентом законченной выпускной квалификационной работы  
21 июня 2019 г.

3. Исходные данные к выпускной квалификационной работе: получение потокового видео с веб-камеры, алгоритм размытия изображения, алгоритм поиска различий между двух кадров, алгоритм поиска контура объектов.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов)

Введение

1 Анализ состояния вопроса

2 Разработка алгоритма для локализации движущихся объектов на изображении

3 Практическая реализация предложенных решений

Заключение

Список используемой литературы

5. Ориентировочный перечень графического и иллюстративного материала: набор формул, объясняющих математический аппарат алгоритма; результаты тестирования работы алгоритма на реальном наборе данных; диаграммы, поясняющие результат работы алгоритма; формы окна программы.
6. Консультанты по разделам \_\_\_\_\_
7. Дата выдачи задания «15» декабря 2018 г.

Заказчик, директор центра

*IT Student*

\_\_\_\_\_  
(подпись)

**А.В. Очеповский**

\_\_\_\_\_  
(И.О. Фамилия)

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_  
(подпись)

**В.С. Климов**

\_\_\_\_\_  
(И.О. Фамилия)

Задание принял к исполнению

\_\_\_\_\_  
(подпись)

**В.А. Ватолин**

\_\_\_\_\_  
(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование кафедры)

УТВЕРЖДАЮ  
Завкафедрой «Прикладная  
математика и информатика»

\_\_\_\_\_ А.В. Очеповский  
(подпись) (И.О. Фамилия)  
«\_\_\_» \_\_\_\_\_ 20\_\_ г.

**КАЛЕНДАРНЫЙ ПЛАН  
выполнения бакалаврской работы**

Студента Ватолин Владимир Алексеевич

по теме «Разработка алгоритма для локализации движущихся объектов на изображении»

Наименование раздела работы	Плановый срок выполнения раздела	Фактически й срок выполнения раздела	Отметка о выполнении	Подпись руководителя
Анализ предметной области	1.10.2018	1.10.2018		
Разработка алгоритма для локализации движущихся объектов на изображении	12.12.2018	12.12.2018		
Практическая реализация предложенных	17.01.2019	17.01.2019		

решений				
Подготовка пояснительной записки	25.02.2019	25.02.2019		
Подготовка презентации к защите	25.02.2019	25.02.2019		
Предзащита дипломной работы	10.06.2019	10.06.2019		
Проверка ВКР в системе «Антиплагиат. ВУЗ»	10.06.2019	10.06.2019		
Сдача пояснительной записки и отзыва на кафедру	17.06.2019	17.06.2019		
Защита дипломной работы				

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_  
(подпись)

**В.С. Климов**  
\_\_\_\_\_  
(И.О. Фамилия)

Задание принял к исполнению

\_\_\_\_\_  
(подпись)

**В.А. Ватолин**  
\_\_\_\_\_  
(И.О. Фамилия)

## АННОТАЦИЯ

Тема бакалаврской работы: «Разработка алгоритма для локализации движущихся объектов на изображении».

В данной бакалаврской работе исследуются способы локализации движущихся объектов на изображении с использованием методов машинного зрения.

В работе предложен алгоритм локализации движущихся объектов на изображении основанный на последовательном выполнении следующих шагов: использование фильтра Гаусса для удаление артефактов изображений, получаемых с веб-камеры, расчет различий между начальным изображением и текущим для обнаружение пикселей изменивших свое значение, применение пороговой функции для фильтрации изменений, объединение пикселей в контур для локализации объекта. Разработано программное обеспечение, реализующее предложенные алгоритмы.

При подготовке исследований, представленных в рамках данной бакалаврской работы, была опубликована 1 статья.

Бакалаврской работа состоит из введения, трёх глав, заключения и списка литературы.

Во введении описывается актуальность проводимого исследования, дается краткая характеристика бакалаврской работы.

В первой главе проводится обзор методов машинного зрения.

Во второй главе описывается разработка предложенного алгоритма для локализации движущихся объектов на изображении.

В третьей главе описывается программное обеспечение, реализующее предложенные подходы, а также приводятся примеры работы алгоритма.

В заключении представлены выводы по проделанной работе.

В работе присутствуют 88 таблиц, 88 рисунков. Список литературы состоит из 20 литературных источников. Общий объем выпускной квалификационной работы составляет 40 страниц.

## **ABSTRACT**

The topic of the given graduation work is "Development of the algorithm for localization of moving objects in the image."

We examine methods of localizing moving objects in an image using computer vision methods are investigated.

We first discuss algorithm for localizing moving objects in an image based on the sequential execution of the following steps: using a Gaussian filter to remove image artifacts taken from a webcam, calculating the differences between the initial image and the current one for detecting pixels that have changed their value, using the threshold function to filter changes, combining pixels into the contour to localize the object. Developed software that implements the proposed algorithms.

In preparing the studies presented in this bachelor's work, 1 article was published.

The graduation work consists of introduction, three chapters, conclusion and list of references.

The introduction describes the relevance of the research, provides a brief description of the bachelor's work.

The first chapter reviews methods of machine vision.

The second chapter describes the development of the proposed algorithm for the localization of moving objects in the image.

The third chapter describes the software that implements the proposed approaches, and also provides examples of the operation of the algorithm.

In the conclusion presents conclusions on the work done.

In the work there are 88 tables, 88 figures. References consists of 20 references. The total amount of final qualifying work is 40 pages.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	5
1 АНАЛИЗ СОСТОЯНИЯ ВОПРОСА.....	7
2 РАЗРАБОТКА АЛГОРИТМА ЛОКАЛИЗАЦИИ ДВИЖУЩИХСЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ.....	15
2.1 Описание алгоритма .....	15
2.2 Подготовка изображения к анализу .....	17
2.3 Субтрактор фона .....	20
2.4 Детектор границ объектов.....	22
2.5 Анализ контуров объектов .....	27
3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННЫХ РЕШЕНИЙ .....	28
3.1 Описание программного обеспечения.....	28
3.2 Тестирование алгоритма .....	30
ЗАКЛЮЧЕНИЕ .....	35
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	37



## ВВЕДЕНИЕ

Разработка систем компьютерного зрения часто связана с решением задач локализации движущихся объектов на изображении. Например, при контроле за обстановкой на дороге необходимо отслеживать перемещение пешеходов и находящихся транспортных средств, в охранных системах контроль может осуществляться путем наблюдения за отсутствием движущихся объектов на охраняемой территории.

В настоящее время проблемами разработки алгоритмов для локализации движущихся объектов на изображении занимаются такие ученые как Cédric Demonceaux, Djemâa Kachi-Akkouche, Bin Nong Sang, Yuehuan Wang, Qingqing Zheng, Xiaotao Zhu, Qian Huang, Feng Ye, Fan Liu, Shufang Xu, Yanfang Wang, Horst Wildenauer, Philipp Blauensteiner, Allan Hanbury, Martin Kampel, Zhao-Yi Wei, Dah-Jye Lee, David Jilk, Robert Schoenberger и др [1-20].

В настоящее время существуют алгоритмы, позволяющие отделять движущиеся объекты на изображении от заднего фона. Такие алгоритмы называются субтракторами фона. Наиболее известные субтракторы – это MOG, MOG2, GNG.

Их использование для решение практических задач ограничено из-за следующих недостатков в работе алгоритмов:

- Алгоритмы не различают движущиеся объекты друг от друга. Т.е. не реализована возможность отдельной локализации движущихся объектов.
- В алгоритмах реализована возможность адаптивного расчёта модели фона. Это означает, что если движущийся объект на некоторое время остановится, то он будет считаться фоном, и не будет отображаться в маске переднего плана. Т.е. в данных алгоритмах не реализована возможность отслеживания объектов, на короткое время приостановивших своё движение.

Исследования, проведенные в рамках данной бакалаврской работы направлены на преодоление этих недостатков. Поэтому целью работы

является – разработка алгоритма для локализации движущихся объектов на изображении.

Для преодоления этих недостатков в работе предложен свой алгоритм для локализации движущихся объектов на изображении. Принцип его работы следующий. С веб-камеры считывается изображение, которое затем подготавливается к анализу, путем сглаживания цифровых шумов и снижения размера признакового пространства путем конвертации цветовой модели из RGB в Gray. Затем производится расчет маски движущихся объектов на основе данных текущего кадра и изображения фона. Для того, чтобы отделить одни движущиеся объекты от других к маске применяется детектор границ объектов (детектор Canny). Полученные контуры объектов фильтруются для отбрасывания, чтобы не учитывать контуры, полученные от помех. Полученные контуры локализуются с помощью прямоугольной рамки на изображении.

Для апробации данного алгоритма на языке программирования Python было разработано приложение, реализующее на практике предложенные решения.

При подготовке бакалаврской работы была опубликована статья по теме интеллектуального анализа данных. Результаты работы были доложены на V Международной научно-практической конференции (школы-семинара) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук».

## 1 АНАЛИЗ СОСТОЯНИЯ ВОПРОСА

При создании систем компьютерного зрения разработчикам часто требуется решать задачу обнаружения и трассировки движущихся объектов на изображении. Например, в системах контроля, за соблюдением правил дорожного движения необходимо при анализе изображения, получаемого со статичной камеры, определять движущиеся автомобили. Для охраны режимных объектов видеоряд, получаемый с камер наблюдения, также анализируется с целью определения наличия движения в охраняемых помещениях. На промышленных объектах также требуется решать задачи наличия движения в опасных для здоровья работников пространствах (например, перед запуском автоматизированной окрасочной камеры требуется убедиться в отсутствии в ней движения, т.е. людей).

Для решения задач обнаружения движения на кадре видеоряда чаще всего применяются субтракторы фона, которые отделяют неподвижный фон от объектов переднего плана, находящихся в движении.

Под субтрактором фона понимается алгоритм, анализирующий несколько последовательных кадров видеоряда с целью выделения модели и фона и применение ее к текущему кадру с целью выделения маски переднего плана. Маска переднего плана – это матрица с той же размерностью, что и исходное изображение. Данная маска является бинарной, т.е. ее принимают только значения или 0, или 1. Компонент данной матрицы принимает значение 0, когда субтрактор определяет, что пиксель исходного кадра, находящего по этим координатам, относится к фону. А значение 1 присваивается компоненту матрицы маски, когда субтрактор определяет, что пиксель исходного кадра, находящего по этим координатам, относится к объекту переднего плана.

Маску объекта переднего плана обычно представляют в виде черно-белого изображения, обозначая белым цветом пиксели переднего плана, а черным цветом пиксели фона.

Отличительной особенностью функционирования субтракторов фона является динамический пересчет модели фона на каждом кадре изображения. Это означает, что если движущийся объект, присутствующий в видео ряде, остановится и будет находиться в неподвижном состоянии, то через некоторое время субтрактор фона будет считать данный объект элементом фона.

Наиболее распространенными субтракторами фона являются алгоритмы MOG, MOG2 и GNG. Рассмотрим их основные отличия и способы реализации.

Проанализируем особенности данных субтракторов фона, приведем примеры программного кода для их использования и на основе тестового видео ряда покажем отличия в результатах их работы.

Работа алгоритма MOG (авторы P. KadewTraKuPong и R. Bowden) основана на использовании смеси Гауссовских распределений. Он использует метод моделирования каждого фонового пикселя с помощью смеси  $K$ -гауссовых распределений (при  $K = 3 - 5$ ). Весы смеси представляют временные пропорции, которые зависят от того, как долго цвета пикселей остаются на изображении. Вероятные цвета пикселей фона — это те, которые остаются дольше всех и более статичны.

Программный код для применения субтрактор фона MOG к тестовому видеоряду написанный на языке программирования Python представлен на рисунке 1.1.

```

import numpy as np
import cv2 as cv

cap = cv.VideoCapture('vtest.avi')

fgbg = cv.bgsegm.createBackgroundSubtractorMOG()

while(1):
    ret, frame = cap.read()

    fgmask = fgbg.apply(frame)

    cv.imshow('frame', fgmask)
    k = cv.waitKey(30) & 0xff
    if k == 27:
        break

cap.release()
cv.destroyAllWindows()

```

Рисунок 1.1 – Применение субтрактора фона MOG к тестовому видеоряду (на языке программирования Python)

Алгоритм MOG2 (автор Z.Zivkovic) тоже основан на использовании смеси Гауссовских распределений. Важной особенностью этого алгоритма является то, что он выбирает соответствующее количество гауссовского распределения для каждого пикселя. Он обеспечивает лучшую адаптацию к изменениям связанным, например переменной освещения объектов.

Другой особенностью алгоритм MOG2 является добавление в матрицу маски третьего типа значений. Помимо значения 0, обозначающего пиксель фона и значения 1, обозначающего пиксель объекта переднего плана, был добавлен вариант значения 2, обозначающий пиксель тени объекта. Тени объектов при визуализации автор алгоритма предлагает выделять серым цветом.

Программный код для применения субтрактор фона MOG2 к тестовому видеоряду написанный на языке программирования Python представлен на рисунке 1.2.

```

import numpy as np
import cv2 as cv

cap = cv.VideoCapture('vtest.avi')
fgbg = cv.bgsegm.createBackgroundSubtractorMOG()

while(1):
    ret, frame = cap.read()

    fgmask = fgbg.apply(frame)

    cv.imshow('frame', fgmask)
    k = cv.waitKey(30) & 0xff
    if k == 27:
        break

cap.release()
cv.destroyAllWindows()

```

Рисунок 1.2 – Применение субтрактора фона MOG2 к тестовому видеоряду  
(на языке программирования Python)

Алгоритм GMG основан на объединении статистической оценки фонового изображения и попиксельной байесовской сегментации. Данный алгоритм предложен Andrew B. Godbehere, Akihiro Matsukawa, Ken Goldber в 2012 году.

Он использует относительно длинную последовательность из 120 кадров для моделирования фона изображения. В нем используется вероятностный алгоритм сегментации переднего плана, который идентифицирует возможные объекты переднего плана с использованием байесовского вывода. Вероятностные оценки являются адаптивными: новые изменения в кадре имеют более сильный вес, чем, чем старые. Для устранения нежелательного шума выполняются операций морфологического преобразования, такие как закрытие и открытие.

Программный код для применения субтрактор фона GMG к тестовому видеоряду написанный на языке программирования Python представлен на рисунке 1.3.

```

import numpy as np
import cv2 as cv

cap = cv.VideoCapture('vtest.avi')

kernel = cv.getStructuringElement(cv.MORPH_ELLIPSE, (3,3))
fgbg = cv.bgsegm.createBackgroundSubtractorGMG()

while(1):
    ret, frame = cap.read()

    fgmask = fgbg.apply(frame)
    fgmask = cv.morphologyEx(fgmask, cv.MORPH_OPEN, kernel)

    cv.imshow('frame', fgmask)
    k = cv.waitKey(30) & 0xff
    if k == 27:
        break

cap.release()
cv.destroyAllWindows()

```


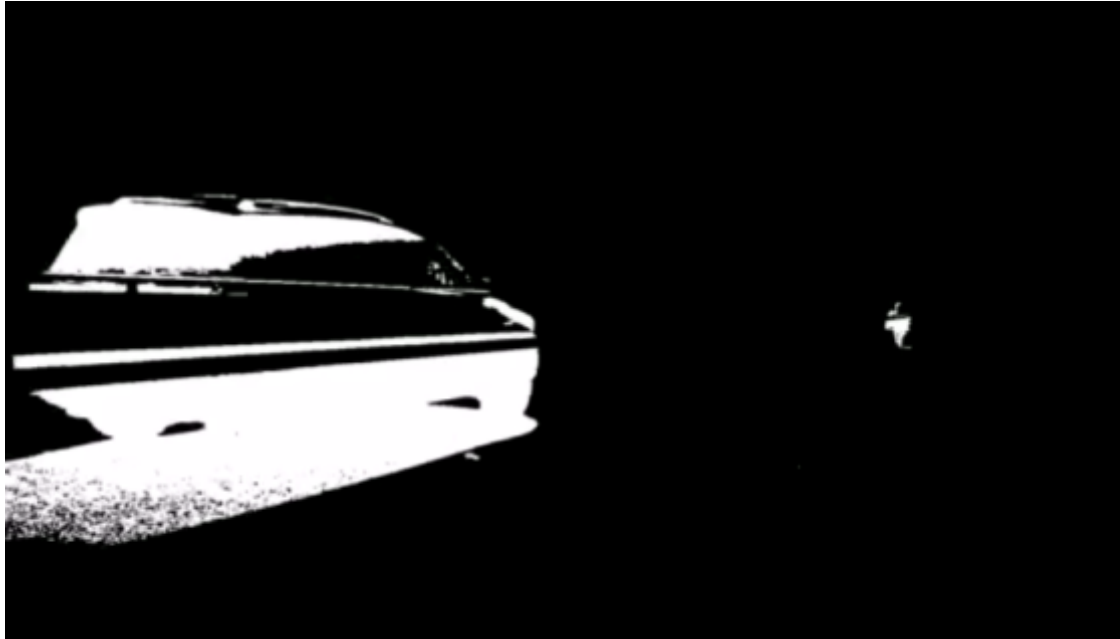
Рисунок 1.3 – Применение субтрактора фона GMG к тестовому видеоряду (на языке программирования Python)

Сравнение результатов работы субтракторов фона проводилось на тестовом видеоряде, который демонстрирует участок дороги, сняты с неподвижной камеры. Видеоряд содержит в себе изображения движущихся транспортных средств.

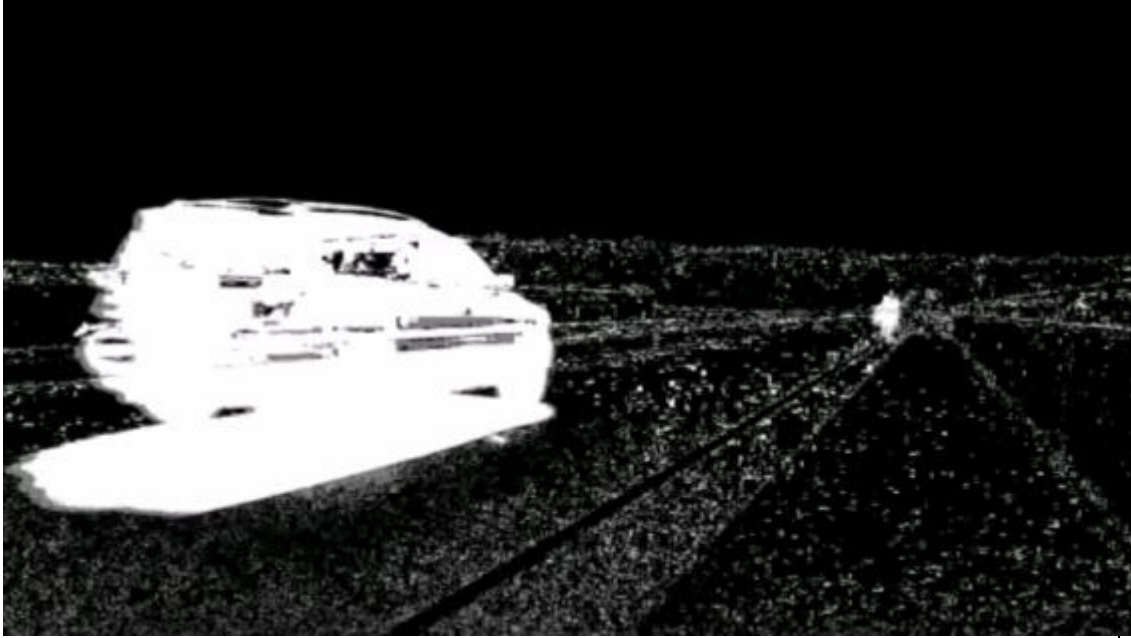

Сравнение результатов использования субтракторов фона на тестовом видеоряде представлено в таблице 1.

Как видно из тестов, все алгоритмы удовлетворительно справились с задачей выделения пикселей движущихся объектов на изображении. При этом маски движущихся объектов, вычисляемые по алгоритмам MOG, MOG2 и GMG различны.

Таблица 1.1 – Сравнение результатов использования субтракторов фона на тестовом видеоряде

Описание	Изображение
Исходный кадр видео ряда	
Субтрактор MOG	



<p style="text-align: center;">Субтрактор MOG2</p>	
<p style="text-align: center;">Субтрактор GMG</p>	

У алгоритмов MOG, MOG2, GMG, есть ряд существенных недостатков, ограничивающих область их практического применения:

1. Данные алгоритмы способны выделять пиксели движущихся объектов, но не различают движущиеся объекты друг от друга. Т.е. в данных алгоритмах не реализована возможность отдельной локализации движущихся объектов.

2. В алгоритмах реализована возможность адаптивного расчёта модели фона. Это означает, что если движущийся объект на некоторое время остановится, то он будет считаться фоном, и не будет отображаться в маске переднего плана. Т.е. в данных алгоритмах не реализована возможность отслеживания объектов, приостановивших своё движение.

Данная бакалаврская работа направлена на преодоление описанных выше недостатков.

Таким образом, актуальной является цель исследования - разработка алгоритма для локализации движущихся объектов на изображении.

Поставленную цель предполагается достигнуть путем решения следующих задач:

1. Провести анализ существующих исследований в области локализации объектов на изображении.
2. Разработать алгоритм для локализации движущихся объектов на изображении.
3. Разработать программное обеспечение, реализующее предложенный алгоритм.
4. Протестировать работу алгоритма и оценить его эффективность.

## **2 РАЗРАБОТКА АЛГОРИТМА ЛОКАЛИЗАЦИИ ДВИЖУЩИХСЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ**

### **2.1 Описание алгоритма**

Предложенный алгоритм локализации движущихся объектов на изображении состоит из следующих этапов (рисунок 2.1):

- Первый этап – считывание текущего изображения с веб-камеры и подготовка его к анализу.
- Второй этап – расчет маски переднего плана, для отделения движущихся объектов от фона.
- Третий этап – выделение границ объектов в отдельные контуры и отсеивание малозначимых контуров, образованных от помех.
- Четвертый этап – локализация объектов на основе оставшихся контуров.

Первый этап предполагает выполнение действий, направленных на подготовку изображения к анализу, и включает в себя преобразование цветовой модели изображения из RGB в Gray, а также сглаживание цифровых шумов за счет применения фильтра Гаусса.

Второй этап предполагает вычисление маски переднего плана, содержащей в себе движущиеся объекты, имеющиеся на изображении. Для этого осуществляется попиксельное вычитание яркостей изображения текущего кадра и изображения фона. Результат вычитания сравнивается с пороговым значением. Если разница яркостей пикселей больше порогового значения, то считается, что данный пиксель относится к движущемуся объекту, если разница меньше порога – то пиксель относится к фону. По результатам выполнения данного этапа мы получаем маску переднего плана, представляющую из себя бинарное изображение.

На третьем этапе маска переднего плана подвергается анализу с помощью детектора границ Канни. На выходе мы получаем множество контуров движущихся объектов.

Затем мы проводим анализ всех найденных контуров: если площадь контура мала, то его появление связано с помехами на изображении, и он исключается из рассмотрения. Если площадь контура выше порогового значения, то контур относится к одному из движущихся объектов.

На четвертом этапе на основе найденных контуров производится локализация движущихся объектов на изображении.

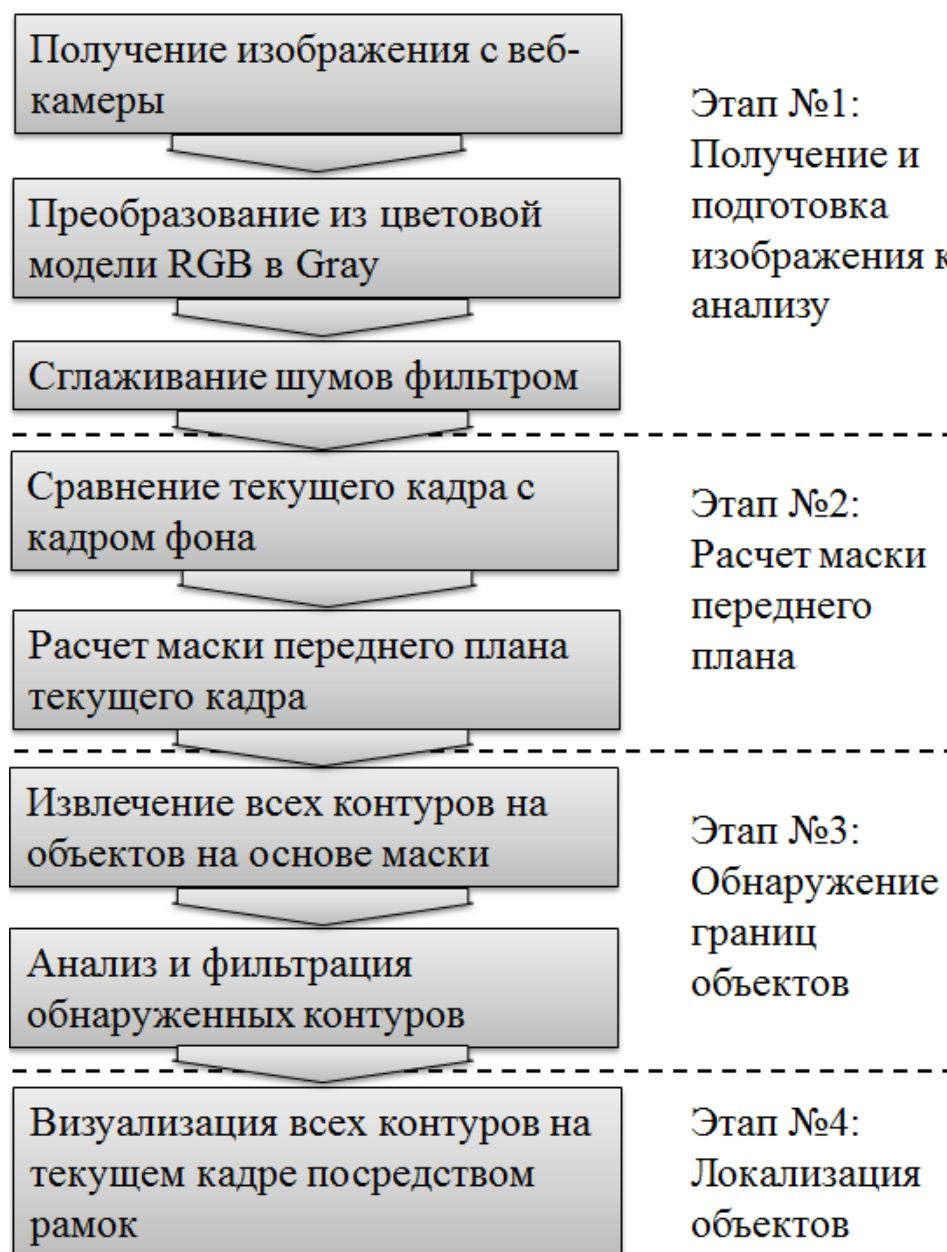


Рисунок 2.1 – Основные этапы работы алгоритма локализации движущихся объектов

## 2.2 Подготовка изображения к анализу

Изображения, получаемые из различных цифровых источников – веб-камер, фотоаппаратов, камер мобильных телефонов обычно представлены с использованием цифровой модели RGB. Данная цветовая модель является оптимальной при демонстрации таких изображений на экранах цифровых устройств, так как для отображения пикселя нужного цвета на экране применяется управления яркостью трех светодиодов (красного, зеленого и синего).

Существуют и другие цветовые модели, предназначенные для своих определенных целей. Например, в типографии распространено использование четырехцветной модели CMYK, так как при печати используются чернила четырех цветов: голубого (Cyan), пурпурного (Magenta), желтого (Yellow) и черного (Black) цвета.

При реализации систем компьютерного зрения обычной практикой является конвертирование изображений в цветовую модель Gray, где каждый пиксель кодируется единственным значением – яркостью пикселя. Такая практика связана с тем, что если у каждого пикселя расположенного по координатам  $x, y$  только один параметр, то изображение можно представить как обычную функцию  $img\_gray(x,y)$ , которая имеет одно значение – яркость пикселя. Работать с такой функцией с точки зрения анализа изображений намного проще, по сравнению с функцией  $img(x,y)$  (изображение в цветовой модели RGB), возвращающей вектор из трех значений – значение компонентов цвета R, G, B (2.1):

$$img(x, y) = \begin{pmatrix} (R_{1,1}; G_{1,1}; B_{1,1}) & \dots & (R_{m,1}; G_{m,1}; B_{m,1}) \\ \dots & \dots & \dots \\ (R_{1,n}; G_{1,n}; B_{1,n}) & \dots & (R_{m,n}; G_{m,n}; B_{m,n}) \end{pmatrix}, \quad (2.1)$$

где  $x$  и  $y$  – координаты пикселей изображения ( $x=1 \dots m, y=1 \dots n$ ),  $m$  – ширина изображения в пикселях,  $n$  – высота изображения в пикселях,  $R, G, B$  – компоненты цвета пикселей ( $R \in [0; 255], G \in [0; 255], B \in [0; 255]$ ).

Существуют разные стандарты и рекомендации по переводу изображений из цветовой модели RGB в Gray ( $img\_gray(x,y)$  – изображение, переведенное в оттенки серого), например:

- Рекомендация «ITU-R Recommendation BT.601», предлагает конвертирование цветовой модели производить по формуле (2.2):

$$img\_gray(x,y) = 0,2989 \cdot R_{x,y} + 0,5870 \cdot G_{x,y} + 0,1140 \cdot B_{x,y}, \quad (2.2)$$

- Рекомендация «ITU-R Recommendation BT.709», предлагает вариант конвертирование цветовой модели, представленный на (2.3):

$$img\_gray(x,y) = 0,2126 \cdot R_{x,y} + 0,7152 \cdot G_{x,y} + 0,0722 \cdot B_{x,y}, \quad (2.3)$$

- Рекомендация «ITU-R Recommendation BT.2100», предлагает вариант конвертирование цветовой модели, представленный на (2.4):

$$img\_gray(x,y) = 0,2627 \cdot R_{x,y} + 0,6780 \cdot G_{x,y} + 0,0593 \cdot B_{x,y}, \quad (2.4)$$

В нашем случае, мы будем пользоваться формулой (2.2). При этом как показала практика, использование других формул не влияет на результат работы предложенного в исследовании алгоритма.

Стоит отметить, что обычно цифровые изображения получают с помощью фотосенсоров, которыми оснащены web-камеры и фотокамеры. Фотосенсоры искажают изображения, внося в них дефекты, называемые цифровым шумом. При анализе изображений цифровой шум вносит значительную погрешность в результаты работы алгоритмов машинного зрения. Поэтому цифровые шумы необходимо сглаживать, используя, например, фильтр размытия по Гауссу.

Применение фильтра заключается в выполнении следующих вычислений (2.5)

$$\begin{aligned} img\_gauss(x,y) &= w * img\_gray(x,y) = \\ &= \sum_{s=-a}^a \sum_{t=-b}^b [w(s,t) \times img\_gray(x-s,y-t)]', \end{aligned} \quad (2.5)$$

где  $w$  – матрица свертки,  $img\_gray(x,y)$  – исходное изображение к которому применяется матрица свертки,  $img\_gauss(x,y)$  – изображение с результатом

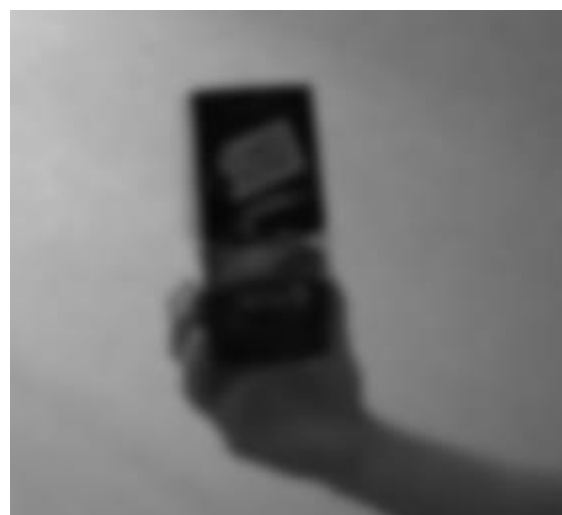
применения матрицы свертки,  $a$ ,  $b$  – коэффициенты, которые зависят от размеров матрицы ядра свертки (для матрицы свертки 3 на 3:  $a=1$ ,  $b=1$ , для матрицы 5 на 5:  $a=2$ ,  $b=2$  и т.д.)



а)



б)



в)

Рисунок 2.2 – Подготовка исходного кадра к анализу: а – изображение  $img(x, y)$ , б – изображение  $img\_gray(x, y)$ , в – изображение  $img\_gauss(x, y)$

Для ускорения расчётов используют матрицы, значения которых заранее рассчитанные по формуле нормального распределения Гаусса (2.6):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.6)$$

где  $\mu$  – математическое ожидание,  $\sigma$  – среднеквадратическое отклонение.

В нашем случае, для фильтрации изображения используется стандартная матрица свертки размером 5 на 5, представленная на (2.7):

$$w = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix} \quad (2.7)$$

Пример применения фильтра Гаусса к изображению представлен на рисунке 2.2.

### 2.3 Субтрактор фона

Маска переднего плана – это матрица с той же размерностью, что и исходное изображение.

Маска переднего плана рассчитывается на основе изображения текущего кадра и модели фона путем попиксельного вычитания одного из другого. (рисунок 2.3) (2.8).

$$mask\_temp(x, y) = |img(x, y) - fon(x, y)|, \quad (2.8)$$

где  $img(x, y)$  – текущий кадр,  $fon(x, y)$  – модель фона.

Затем полученные по модули разности пиксели сравниваются с фиксированным значением (значение порога  $T$ ) и если разность больше, чем значение порога, то пиксель по данным координатам считается относящимся к объекту, если разность меньше значения порога, то пиксель считается относящимся к фону.



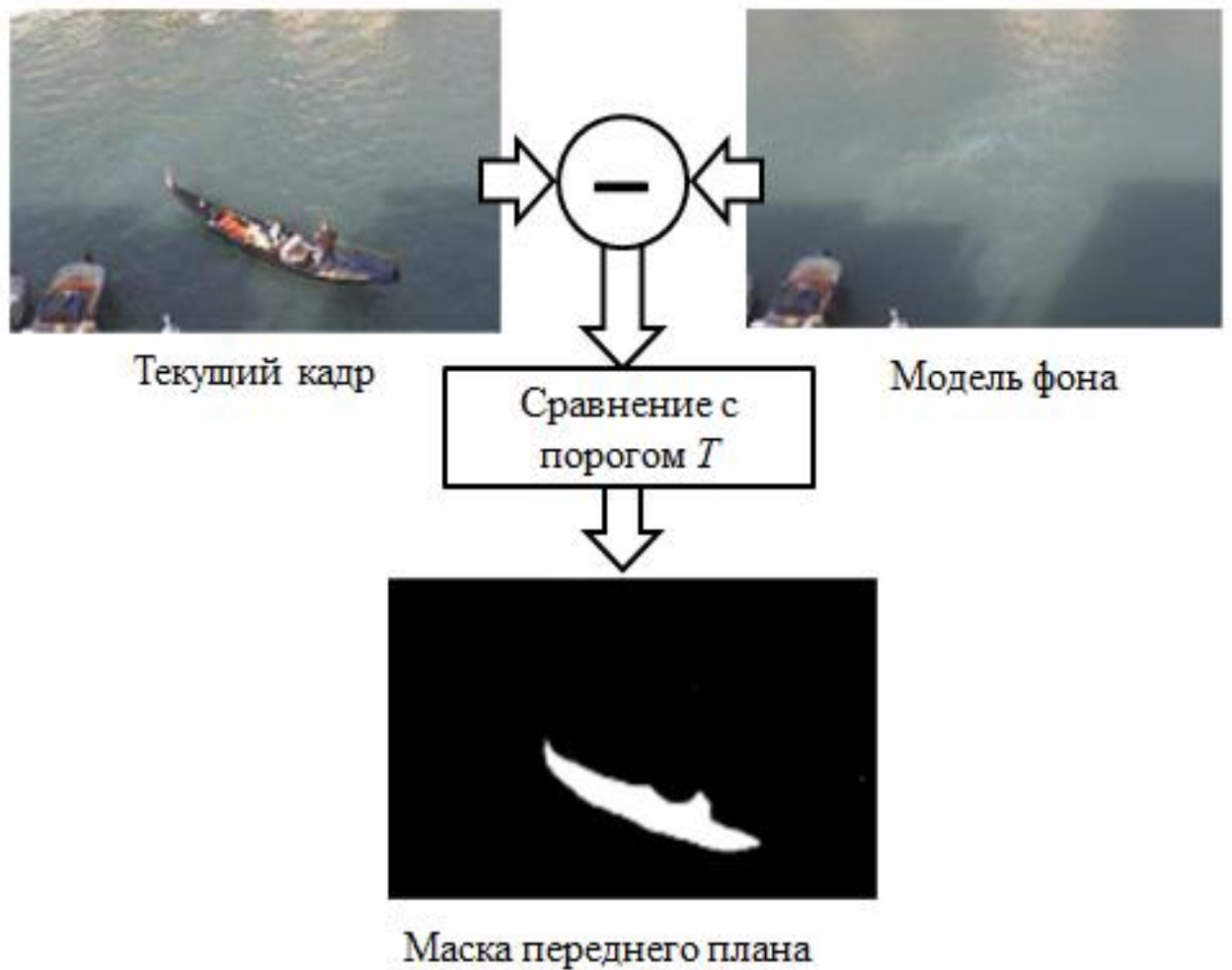


Рисунок 2.3 – Схема расчета маски переднего плана

С применение пороговой фильтрации показано на (2.9):

$$mask(x, y) = \begin{cases} 1 & \text{if } mask\_temp \geq T \\ 0 & \text{if } mask\_temp < T \end{cases}, \quad (2.9)$$

Маску переднего плана удобно визуализировать, представив ее в виде черно-белого изображения.

Пример расчета маски переднего плана с использованием предложенного подхода представлен на рисунке 2.4.

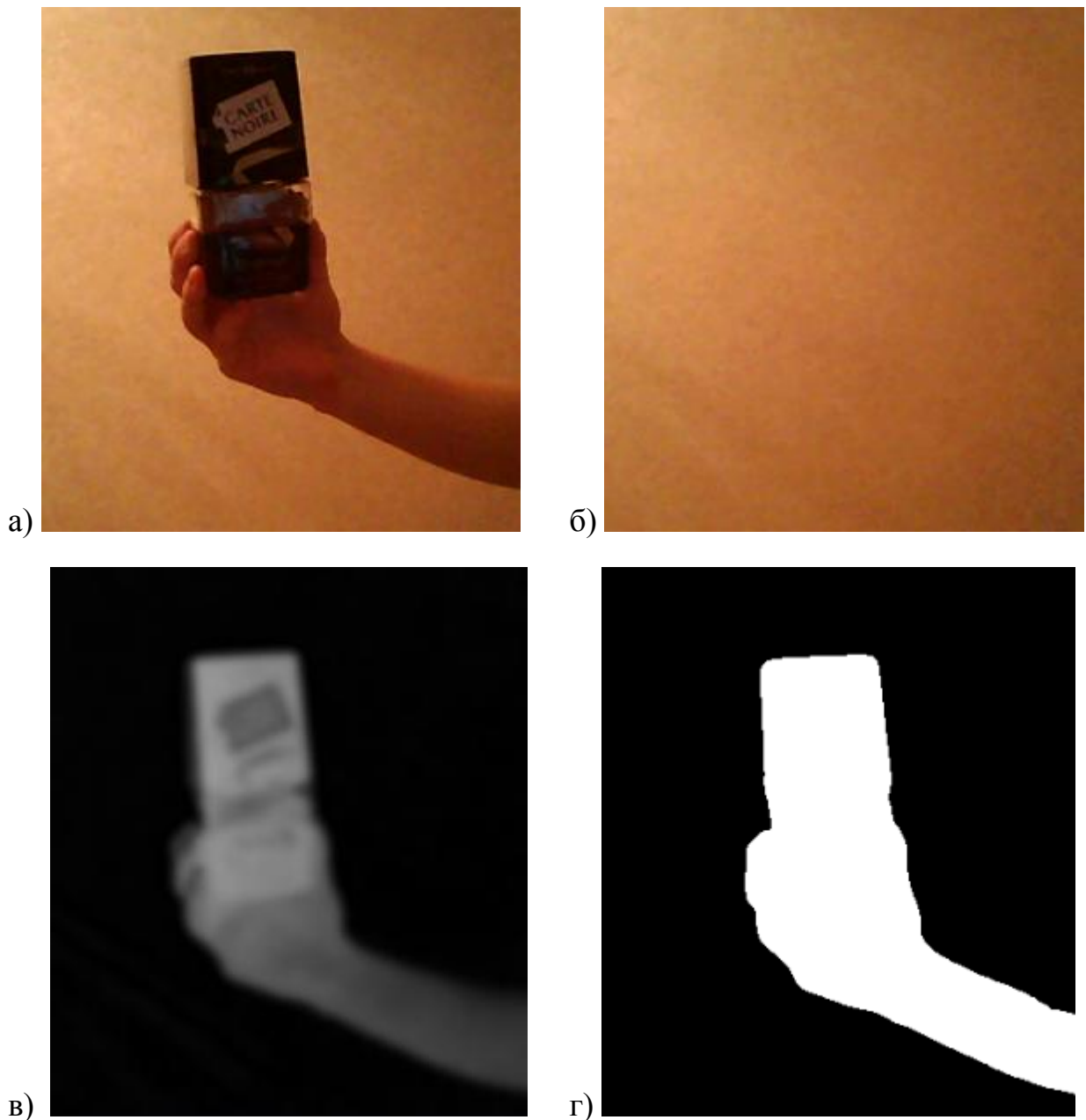


Рисунок 2.4 – Пример работы субтрактора фона: а – изображение  $img\_gauss(x,y)$ , б – изображение  $fon\_gauss(x,y)$ , в – изображение  $mask\_temp(x,y)$ , г – изображение  $mask(x,y)$

## 2.4 Детектор границ объектов

Теперь, когда у нас имеется маска переднего плана (бинарное изображение  $mask(x,y)$ ) нам требуется отделить изображения движущихся объектов друг от друга. Это предлагается сделать путем выделения контуров объектов на основе анализа изображения маски переднего плана  $mask(x,y)$  с

помощью детектора границ Канни. На выходе такой обработки мы ожидаем получить множество замкнутых непересекающихся контуров, в котором каждый контур соответствует своему объекту.

Работа метода заключается в поиске кривых на изображениях, вдоль которых происходит резкое изменение яркости пикселей.

Детектор границ Канни определяет места резкого изменения яркости пикселей с помощью градиента. При программной реализации расчета градиента с точки зрения оптимизации применяют дискретный дифференциальный оператор Собеля, позволяющий вычислять приближенное значение градиента.

При этом оператор Собеля основан на свертке изображения целочисленными фильтрами в двух направлениях – вертикальном и горизонтальном.

Ядра горизонтального фильтра  $Xg$  и вертикального фильтра  $Yg$  выглядят как (2.10):

$$\left\{ \begin{array}{l} Xg = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \\ Yg = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \end{array} \right. \quad (2.10)$$

Расчет значений градиента осуществляется для каждой позиции пикселя с помощью двух вложенных циклов. Первый цикл с индексом  $iX$  управляет за перемещение по оси X, второй цикл  $iY$  отвечает за перемещение по оси Y (2.11):

$$\begin{array}{l} \text{for } iY \in 2 \dots \text{rows}(\text{mask}(x, y)) - 2 \text{ do} \\ \quad \text{for } iX \in 2 \dots \text{cols}(\text{mask}(x, y)) - 2 \text{ do ,} \\ \quad \quad \langle \text{расчет градиента} \rangle \end{array} \quad (2.11)$$

где функция  $\text{rows}()$  определяет высоту исходного изображения в пикселях, а функция  $\text{cols}()$  – ширину изображения в пикселях.

Внутри цикла *<расчет градиента>* для расчета выполняются следующие действия. На первом этапе происходит выделение фрагмента *part\_img\_gauss* изображения *img\_gauss* размером 3 на 3 пикселя. Для этого используется функция *submatrix()* (2.12):

$$part\_img\_gauss = submatrix(img\_gauss, iY - 1, iY + 1, iX - 1, iX + 1) \quad (2.12)$$

Затем рассчитывается значение градиента *GX* в горизонтальном направлении и значение градиента *GY* в вертикальном направлении (2.13):

$$\begin{cases} GX = \sum_{y=1}^3 \sum_{x=1}^3 (mask(x, y) \cdot MGx(x, y)) \\ GY = \sum_{y=1}^3 \sum_{x=1}^3 (Mask(x, y) \cdot MGy(x, y)) \end{cases} \quad (2.13)$$

Теперь с учетом (2.11) и (2.12) можно рассчитать значение градиента с учетом горизонтальной и вертикальной составляющей следующим образом (2.14):

$$G = \sqrt{\left[ \sum_{y=1}^3 \sum_{x=1}^3 (Mask(x, y) \cdot Yg(x, y)) \right]^2 + \left[ \sum_{y=1}^3 \sum_{x=1}^3 (Mask(x, y) \cdot Xg(x, y)) \right]^2} \quad (2.14)$$

Угол вектора градиента квантируется по 45 градусам, т.е. все значения углов округляются до ближайшего значения из ряда (0°, 45°, ..., 315°). Это необходимо для последующей фильтрации изображения с целью более точного выделения границ объектов (2.15):

$$\theta = round \arctan 2(Gx, Gy) / 45^\circ \cdot 45^\circ - 90^\circ \quad (2.15)$$

На этом заканчивается текущая итерация цикла *<расчет градиента>* и осуществляется переход к следующей итерации с другой парой значений *iX* и *iY*.

После завершения выполнения цикла мы получаем две матрицы: первая содержит значения градиента для каждого пикселя изображения, а вторая угол вектора градиента для каждого пикселя изображения (2.16).

$$\begin{cases} img\_edge(iX, iY) = G \\ img\_angle(iX, iY) = \theta \end{cases} \quad (2.16)$$

Следующим этапом фильтрации изображения является подавление не-максимумов. Его идея заключается в том, что границами объявляются только те пиксели, в которых достигается максимум градиента в направлении вектора градиента.

На рисунке 2.5 показан принцип действия подавления не максимумов. При этом сначала определяется наиболее часто встречающееся направление градиента (на показанном примере это угол  $0^\circ$ ). При данном направлении вектора осуществляется поиск максимальных значений градиента в вертикальном направлении (в каждом столбце пикселей). В примере пиксели с максимальным значением градиента показаны белой рамкой. Эти максимальные значения градиента остаются, а остальные подавляются.

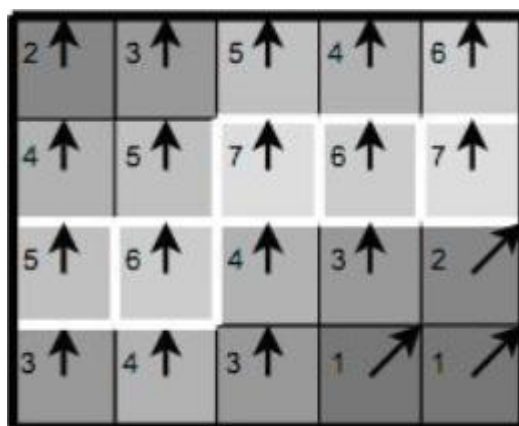


Рисунок 2.5 – Подавление не-максимумов (стрелками показано направление векторов), цифры – значение градиента.

Результат использования детектора границ Канни к изображению маски переднего плана представлен на рисунке 2.6

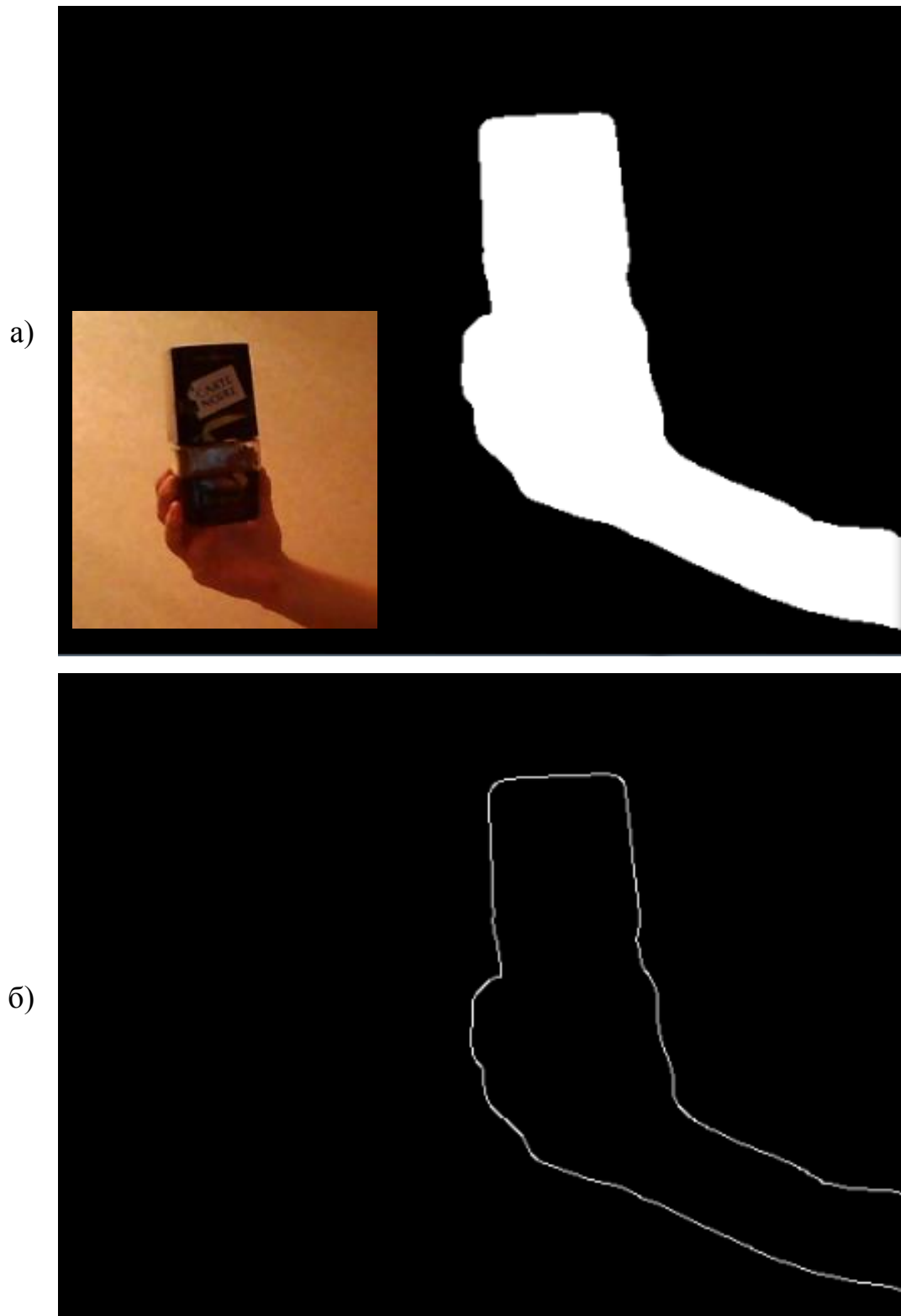


Рисунок 2.6 – Выделение контуров объектов: а – изображение  $mask(x,y)$ , б – изображение  $img\_edge(x,y)$

## 2.5 Анализ контуров объектов

С помощью детектора границ Канни изображение маски переднего плана было преобразовано во множество контуров *contours*.

Множество контуров содержит в себе:

- контуры движущихся объектов, которые необходимо локализовать на изображении;
- контуры, вызванные помехами и дефектами, присутствующих на изображении. Данные контуры не представляют интерес и их надо отсеивать.

Контуры помех и дефектов легко отсеять, так они обладают малыми размерами. Поэтому для их исключения из рассмотрения достаточно в цикле перебрать все контуры и сравнить площадь, очерчиваемую контуром с пороговым значением (в нашем случае пороговое значение – 10000 пикселей).

Если рассматриваемый контур очерчивает площадь больше, чем данное значение, то данный контур относится к искомому объекту. Для локализации такого объекта производится поиск максимального и минимального значения ключевых точек контура по осям X и Y. После этого, по данным координатам строится прямоугольник, ограничивающий область с искомым объектом.

Псевдокод алгоритма анализа контуров представлен на рисунке 2.7.

```
conturs ← findContours(img _ edge(x, y))  
for cont ∈ conturs do  
  if contourArea(cont) ≥ 10000 then  
    x _ max ← find.Xmax(cont)  
    x _ min ← find.Xmin(cont)  
    y _ max ← find.Ymax(cont)  
    y _ min ← find.Ymin(cont)  
    draw.rectangle(x _ min, y _ min, x _ max, y _ max)
```

Рисунок 2.7 – Псевдокод алгоритма анализа контуров объектов

## 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННЫХ РЕШЕНИЙ

### 3.1 Описание программного обеспечения

Для апробации на практике предложенных решений на языке Python было разработано программное обеспечение, обладающее следующими функциональными возможностями:

- получение изображения с помощью веб-камеры;
- локализация движущихся объектов на изображении с помощью описанного в бакалаврской работе алгоритма;
- обновление модели фона по нажатию горячей клавиши;
- наглядная визуализация каждого из этапов работы алгоритма.

При запуске программного обеспечения происходит инициализация и активирование веб-камеры. В случае положительной инициализации веб-камеры на экране отобразятся несколько окон, в которых будут отображены результаты работы различных этапов локализации движущихся объектов на изображении:

В окне “Color frame” (рисунок 3.1, поз. 1) отображается оригинальное изображение, получаемое с веб-камеры, которое затем подвергает обработке. В окне “Gray frame” (рисунок 3.1, поз. 2) демонстрируется изображение после перевода из цветовой модели RGB в Gray, а также после применения фильтра Гаусса (для сглаживания цифровых шумов). В окне “Gray frame” (рисунок 3.1, поз. 3) демонстрируется результат попиксельного вычитания из текущего кадра изображения фона. В окне “Threshold Frame” (рисунок 3.1, поз. 4) демонстрируется результат расчета маски переднего фона. В окне “Canny Frame” (рисунок 3.1, поз 5) демонстрируется результат применения к маске переднего фона детектора границ Канни. Результат анализа выделенных контуров – локализация движущихся объектов. Результат локализации объектов отображается в окне “Color frame” в виде рамок (рисунок 3.1, поз 6).



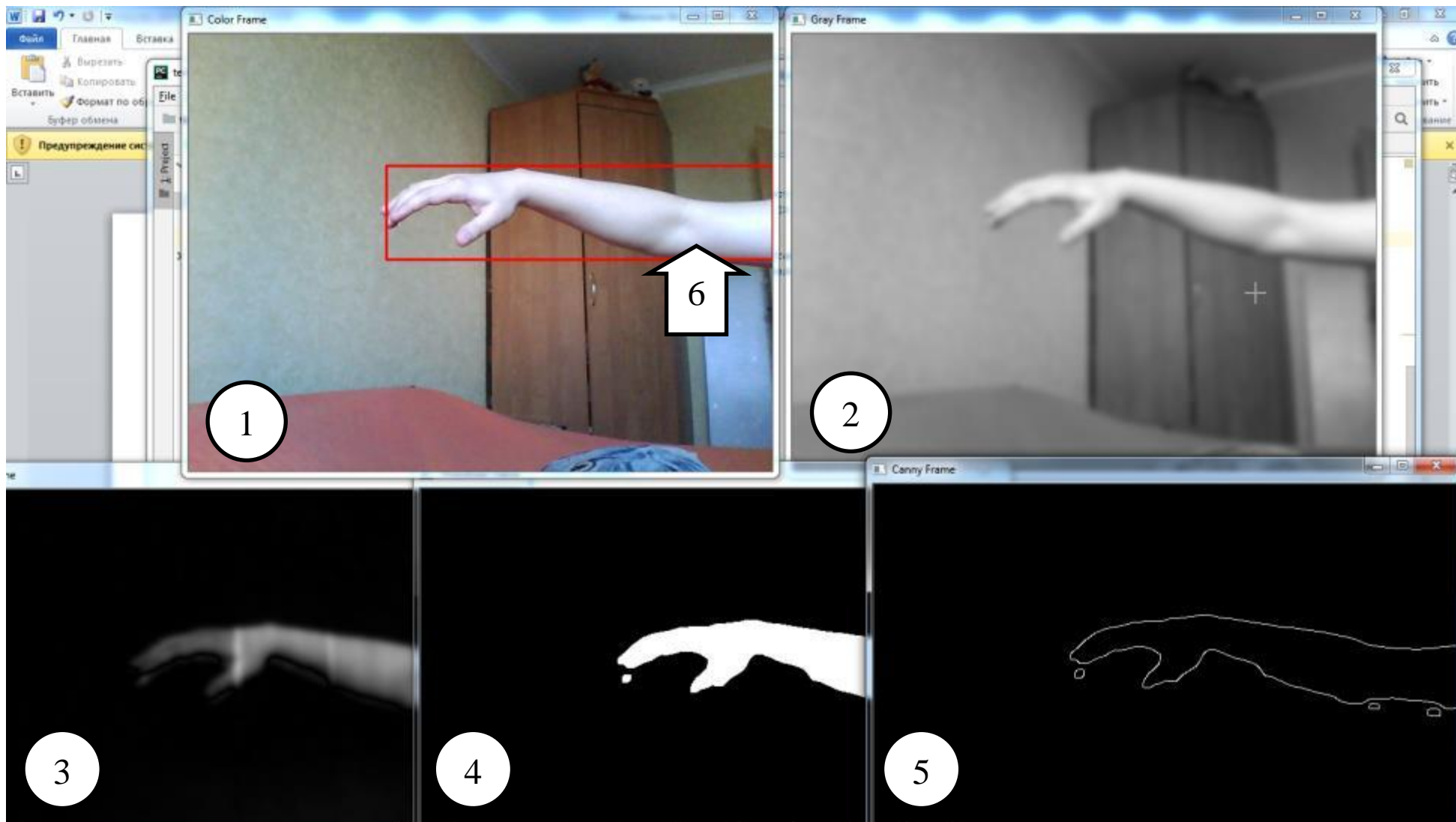


Рисунок 3.1 – Основные окна разработанного программного обеспечения

### 3.2 Тестирование алгоритма

Тестирование результатов работы алгоритма проводилась на данных видеоряда, получаемых с веб-камеры ноутбука. При этом веб-камера имеет низкое разрешение (640x480 пикселей) и технически несовершенный сенсор, генерирующий большое количество цифрового шума на изображении. Это является дополнительными возмущающими факторами, усложняющими задачу правильной работы алгоритма.

Работа алгоритма проверялась на примере двух ситуаций:

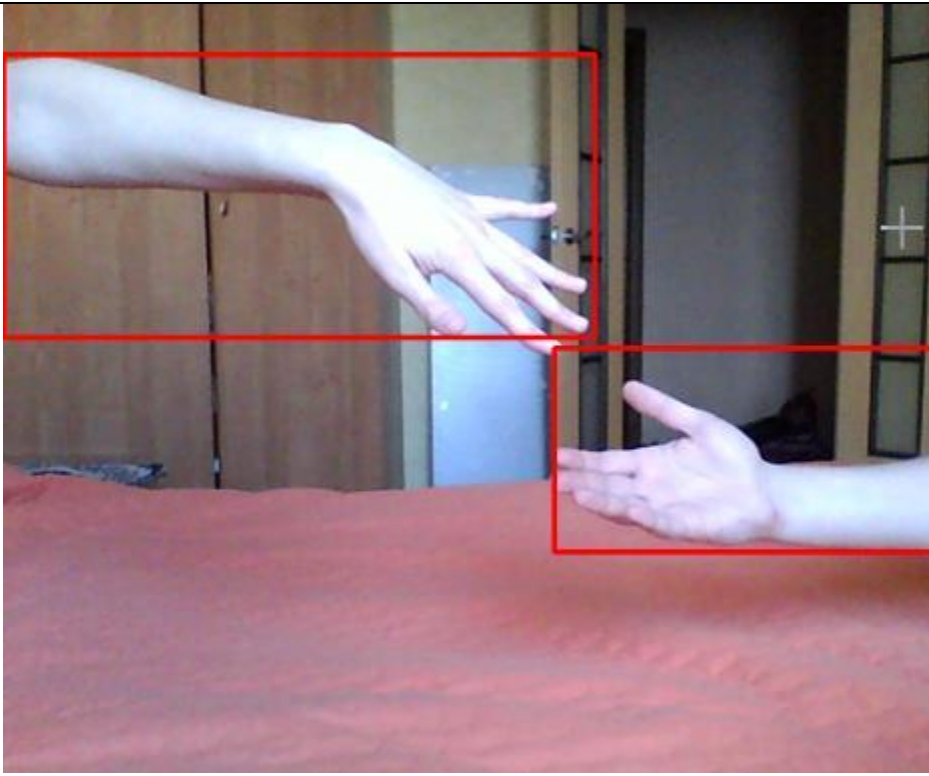

1. Присутствие на изображении несколько движущихся объектов (в данном случае – две руки). В этом случае от алгоритма требуется уверенное распознавание и локализация всех движущихся объектов отдельно друг от друга.

2. Присутствие на изображении объекта, находившегося в движении, а находящегося в состоянии покоя в текущий момент времени. В этом случае от алгоритма требуется локализация объекта, приостановившего свое движение. При этом сохраняется требование отдельной локализации объектов друг от друга.

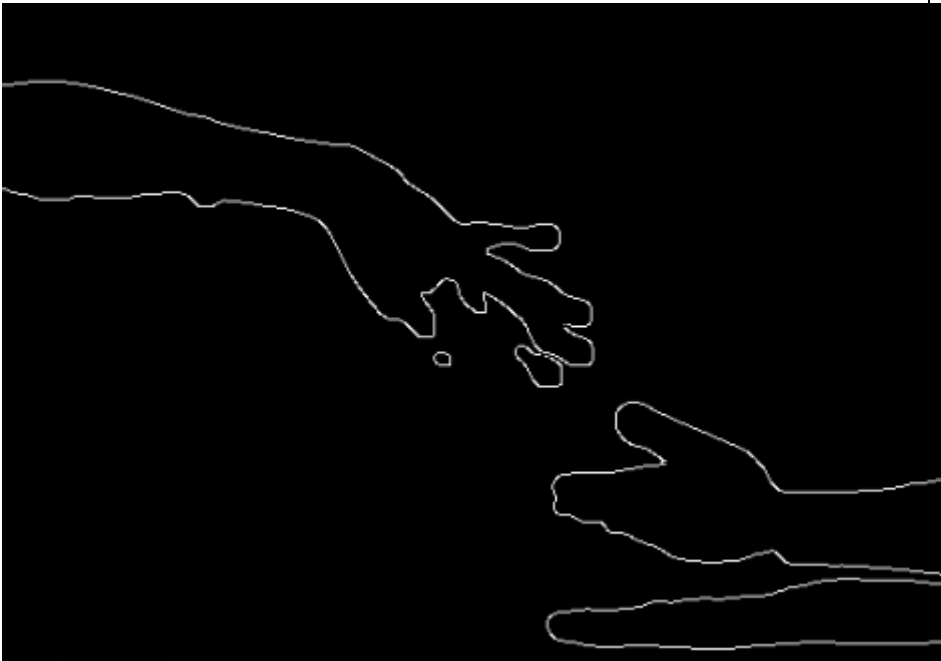
Результаты тестирования алгоритма в рамках первой ситуации представлены в таблице 1. Как видно из рисунков, алгоритм успешно справился с задачей локализации двух независимо движущихся объектов.

Результата тестирования алгоритма в рамках второй ситуации представлены на рисунке 3.2. Здесь роль объекта, приостановившего свое движение, выполняет банка кофе, в роль движущегося объекта – рука. Как видно и рисунков, алгоритм успешно работает и локализует объекты и в этом случае.

Таблица 3.1 – Применение алгоритма при локализации двух независимо движущихся объектов (рук)

Описание	Изображение
<p>Исходное изображение (текущий кадр), полученное с веб-камеры. Красными прямоугольниками отмечен результат работы алгоритма – найденные движущиеся объекты.</p>	 <p>The image shows two hands in a room. The hand on the left is extended horizontally, and the hand on the right is positioned lower. Two red rectangular bounding boxes are drawn around each hand, indicating the algorithm's detection of moving objects. The background includes a wooden door and a window with a cross-shaped frame.</p>
<p>Текущий кадр, переведенный в цветовую модель Gray. Для повышения точности работы алгоритма производится сглаживание цифровых шумов фильтром Гаусса (эффект размытия)</p>	 <p>This is a grayscale version of the image above. The image has been processed with a Gaussian blur filter, which softens the edges and reduces digital noise. The two hands are still visible, but the details are less sharp compared to the original color image.</p>

<p>Промежуточный результат при формировании маски переднего плана – попиксельное вычитание яркостей текущего кадра и изображения фона</p>	
<p>Формирование маски переднего плана, которая осуществляется с помощью пороговой функции.</p>	

<p>Результат применения к маске переднего плана детектора границ Канни. Получившиеся контуры отсеиваются с учетом их длины. Если площадь, занимаемая контуром больше 10000 пикселей, то он выделяется в красную рамку</p>	
---	--

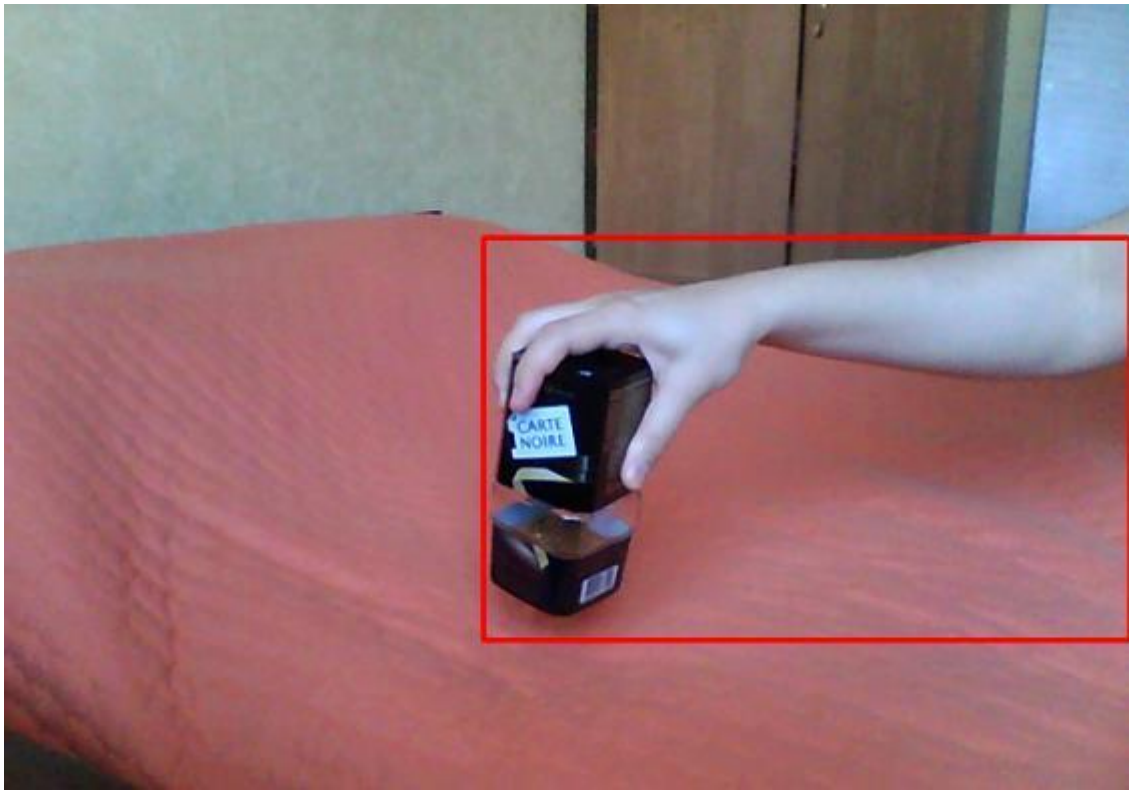
Как видно из результатов тестирования:

1. Предложенный алгоритм не только выделяет движущиеся объекты, но и различает объекты друг от друга. Таким образом, в предложенном алгоритме реализована возможность отдельной локализации движущихся объектов.

2. Предложенный алгоритм локализации, обладает способностью локализации объектов, которые находились в движении и в текущий момент времени находятся в состоянии покоя. Т.е. в данном алгоритме реализована возможность отслеживания объектов, приостановивших своё движение.

Таким образом, были преодолены недостатки современных алгоритмов MOG, MOG2 и GNG.

а)



б)



Рисунок 3.2 – Применение алгоритма при локализации покоящегося объекта, находившегося до этого в движении (банка кофе): а – в движении, б – в состоянии покоя

## ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы были сделаны следующие выводы:

1. Анализ литературных источников показал, что при разработке систем компьютерного зрения разработчикам часто требуется решать задачу обнаружения и трассировки движущихся объектов на изображении. Поэтому развитие алгоритмов локализации движущихся объектов на изображении является актуальной задачей.

2. Анализ литературных источников показал, что наиболее распространенными алгоритмами локализации движущихся объектов являются MOG, MOG2 и GNG. Но обладают недостатками – невозможность раздельной локализации движущихся объектов на одном изображении и не возможность учета локализации объектов, приостановивших свое движение.

3. Для преодоления этих недостатков разработан свой алгоритм локализации движущихся объектов, который основан на подготовке изображения к анализу путем преобразования цветовой модели изображения из RGB в Gray, сглаживание цифровых шумов за счет применения фильтра Гаусса, вычисление маски переднего плана путем попиксельного вычитания яркостей изображения текущего кадра и изображения фона и сравнение результата с пороговым значением, применения к маске переднего плана детектора границ Канни для обнаружения контуров движущихся объектов, анализ всех найденных контуров для фильтрации контуров связанных с шумами и дефектами и локализация объектов, путем ограничения контуров рамками.

4. Для апробации данного алгоритма на языке программирования Python было разработано приложение, реализующее на практике предложенные решения.

5. Предложенное программное обеспечение было протестировано на данных видеоряда, получаемого с веб камеры ноутбука. Подробно результаты тестирования рассматриваются в третьей главе. Результаты тестирования на практике подтверждают состоятельность предложенных решений.

6. При подготовке бакалаврской работы была опубликована статья по теме интеллектуального анализа данных. Результаты работы были доложены на V Международной научно-практической конференции (школы-семинара) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук».



## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Demonceaux, C. Motion Detection Using Wavelet Analysis and Hierarchical Markov Models [Text] / Cédric Demonceaux, Djemâa Kachi-Akkouche // International Workshop on Spatial Coherence for Visual Motion Analysis – First International Workshop, SCVMA 2004, Prague, Czech Republic, May 15, 2004. Revised Papers: Spatial Coherence for Visual Motion Analysis. – Springer-Verlag Berlin Heidelberg 2006. –pp. 64-75
2. Sun, B. Motion Detection Based on Biological Correlation Model [Text] / Bin Nong Sang, Yuehuan Wang, Qingqing Zheng // International Symposium on Neural Networks – 7th International Symposium on Neural Networks, ISNN 2010, Shanghai, China, June 6-9, 2010, Proceedings, Part II: Advances in Neural Networks - ISNN 2010. – Springer-Verlag Berlin Heidelberg 2010. – pp. 214-221
3. Zhu, X. Motion-Compensated Deinterlacing Based on Scene Change Detection [Text] / Xiaotao Zhu, Qian Huang, Feng Ye, Fan Liu, Shufang Xu, Yanfang Wang // Pacific Rim Conference on Multimedia – 18th Pacific-Rim Conference on Multimedia, Harbin, China, September 28-29, 2017, Revised Selected Papers, Part II: Advances in Multimedia Information Processing – PCM 2017. – Springer International Publishing AG, part of Springer Nature 2018. – pp. 397-406
4. Wildenauer, H. Motion Detection Using an Improved Colour Model [Text] / Horst Wildenauer, Philipp Blauensteiner, Allan Hanbury, Martin Kampel // International Symposium on Visual Computing – Second International Symposium, ISVC 2006 Lake Tahoe, NV, USA, November 6-8, 2006. Proceedings, Part II: Advances in Visual Computing. – Springer-Verlag Berlin Heidelberg 2006. – pp. 607-616
5. Ngan, P. Motion detection in temporal clutter [Text] / Phillip M. Ngan // Asian Conference on Computer Vision – Third Asian Conference on Computer Vision Hong Kong, China, January 8–10, 1998 Proceedings, Volume II: Computer Vision — ACCV'98. – Springer-Verlag Berlin Heidelberg 1997. – pp. 615-622

6. Wei, Zh. Motion Projection for Floating Object Detection [Text] / Zhao-Yi Wei, Dah-Jye Lee, David Jilk, Robert Schoenberger // International Symposium on Visual Computing – Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings, Part II ISVC 2007: Advances in Visual Computing. – Springer-Verlag Berlin Heidelberg 2007. – pp. 152-161
7. Liang, H. A Motion Detection Model Inspired by Hippocampal Function and Its FPGA Implementation [Text] / Haichao Liang, Takashi Morie // International Conference on Neural Information Processing – 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part III: Neural Information Processing. – Springer-Verlag Berlin Heidelberg 2011. – pp. 522-529
8. Wang, Zh. Object Detection with Multiple Motion Models [Text] / Zhijie Wang, Hong Zhang // Asian Conference on Computer Vision – 9th Asian Conference on Computer Vision, Xi'an, September 23-27, 2009, Revised Selected Papers, Part III: Computer Vision – ACCV 2009. – Springer-Verlag Berlin Heidelberg 2010. – pp. 183-192
9. Latecki, L. Activity and Motion Detection Based on Measuring Texture Change [Text] / Longin Jan Latecki, Roland Mieziako, Dragoljub Pokrajac // International Workshop on Machine Learning and Data Mining in Pattern Recognition – 4th International Conference, MLDM 2005, Leipzig, Germany, July 9-11, 2005. Proceedings: Machine Learning and Data Mining in Pattern Recognition. – Springer-Verlag Berlin Heidelberg 2005. – pp. 476-486
10. Gopalakrishnan, V. Sustained Observability for Salient Motion Detection [Text] / Viswanath Gopalakrishnan, Yiqun Hu, Deepu Rajan // Asian Conference on Computer Vision – 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III: Computer Vision – ACCV 2010. – Springer-Verlag Berlin Heidelberg 2011. – pp. 732-743

11. Pichaikuppan, V. Change Detection in the Presence of Motion Blur and Rolling Shutter Effect [Text] / Vijay Rengarajan Angarai Pichaikuppan, Rajagopalan Ambasamudram Narayanan, Aravind Rangarajan // European Conference on Computer Vision – 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII: Computer Vision – ECCV 2014. – Springer International Publishing Switzerland 2014. – pp. 123-137
12. Peng, Zh. Vibe Motion Target Detection Algorithm Based on Lab Color Space [Text] / Zhiyong Peng, Faliang Chang, Wenhui Dong // Chinese Conference on Image and Graphics Technologies – 10th Chinese Conference, IGTA 2015, Beijing, China, June 19-20, 2015, Proceedings: Advances in Image and Graphics Technologies. – Springer-Verlag Berlin Heidelberg 2015. – pp. 45-54
13. Czúni, L. Independent Motion Detection in the Light of the Aperture Problem [Text] / László Czúni, Mónika Gál // International Conference Image Analysis and Recognition – 9th International Conference, ICIAR 2012, Aveiro, Portugal, June 25-27, 2012. Proceedings, Part I: Image Analysis and Recognition. – Springer-Verlag Berlin Heidelberg 2012. – pp. 200-207
14. Blanco, C. Aerial Moving Target Detection Based on Motion Vector Field Analysis [Text] / Carlos R. del-Blanco, Fernando Jaureguizar, Luis Salgado, Narciso García // International Conference on Advanced Concepts for Intelligent Vision Systems – 9th International Conference, ACIVS 2007, Delft, The Netherlands, August 28-31, 2007. Proceedings: Advanced Concepts for Intelligent Vision Systems. – Springer-Verlag Berlin Heidelberg 2007. – pp. 990-1001
15. Park, M. Efficient Shot Boundary Detection for Action Movies Using Blockwise Motion-Based Features [Text] / Min-Ho Park, Rae-Hong Park, Sang Wook Lee // International Symposium on Visual Computing – First International Symposium, ISVC 2005, Lake Tahoe, NV, USA, December 5-7, 2005. Proceedings: Advances in Visual Computing. – Springer-Verlag Berlin Heidelberg 2005. – pp. 478-485

16. Chakraborty, M. Real-Time Image-Based Motion Detection Using Color and Structure [Text] / Manali Chakraborty, Olac Fuentes // International Conference Image Analysis and Recognition – 6th International Conference, ICIAR 2009, Halifax, Canada, July 6-8, 2009. Proceedings: Image Analysis and Recognition. – Springer-Verlag Berlin Heidelberg 2009. – pp. 678-686
17. Jiménez, P. Background Pixel Classification for Motion Detection in Video Image Sequences [Text] / P.Gil Jiménez, S. Maldonado-Bascón, R. Gil-Pita, H. Gómez-Moreno // International Work-Conference on Artificial Neural Networks – 7th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2003 Maó, Menorca, Spain, June 3–6, 2003 Proceedings, Part I: Computational Methods in Neural Modeling. – Springer-Verlag Berlin Heidelberg 2003. – pp. 718-725
18. Schulte, S. Noise Reduction of Video Sequences Using Fuzzy Logic Motion Detection [Text] / Stefan Schulte, Vladimir Zlokolica, Aleksandra Pizurica, Wilfried Philips, Etienne Kerre // International Conference on Advanced Concepts for Intelligent Vision Systems – 7th International Conference, ACIVS 2005, Antwerp, Belgium, September 20-23, 2005. Proceedings: Advanced Concepts for Intelligent Vision Systems. – Springer-Verlag Berlin Heidelberg 2005. – pp. 666-673
19. Saif, A. Adaptive Motion Pattern Analysis for Machine Vision Based Moving Detection from UAV Aerial Images [Text] / A. F. M. Saifuddin Saif, Anton Satria Prabuwono, Zainal Rasyid Mahayuddin // International Visual Informatics Conference – Third International Visual Informatics Conference, IVIC 2013, Selangor, Malaysia, November 13-15, 2013. Proceedings: Advances in Visual Informatics. – Springer International Publishing Switzerland 2013. – pp. 104-114
20. Fablet, R. Automatic Detection and Tracking of Human Motion with a View-Based Representation [Text] / Ronan Fablet, Michael J. Black // European Conference on Computer Vision – 7th European Conference on Computer Vision

Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I: Computer Vision —  
ECCV 2002. – Springer-Verlag Berlin Heidelberg 2002. – pp. 476-491