

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет

«Московский институт электронной техники»

Факультет микроприборов и технической кибернетики

Кафедра высшей математики №1

Димаков Владислав Сергеевич

Бакалаврская работа

по направлению 01.03.04 «Прикладная математика»

«Обнаружение и слежение за объектами в реальном времени на основе  
самообучающегося классификатора»

Студент

\_\_\_\_\_

Димаков В.С.

Научный руководитель,

к.ф.-м.н., доцент кафедры ВМ-1

\_\_\_\_\_

Козлитин И.А.

Москва 2017

## Оглавление

Введение. ....	3
Актуальность проблемы. ....	3
Цели и задачи выпускной квалификационной работы. ....	3
Содержание работы. ....	4
Глава 1. Обнаружение и слежение за объектами видеопоследовательности. ....	5
1.1. Видеопоследовательность. ....	5
1.2. Задача обнаружения и слежения за объектами. ....	6
1.3. Сегментация изображений. ....	7
1.4. Вычисление пространственных параметров объектов. ....	10
1.5. Учет движения камеры в методе трекинга на основе вычитания фона. ....	11
1.6. Порог фоновой части в методе трекинга на основе вычитания фона. ....	16
Глава 2. Распознавание образов и классификация. ....	17
2.1. Распознавание образов. ....	17
2.2. Методы распознавания образов. ....	18
2.3. Случайные леса. ....	19
2.4. Обучение случайного леса в режиме реального времени. ....	24
Глава 3. Применение методов распознавания образов к задаче слежения. ....	28
3.1. Задача долгосрочного слежения за объектами. ....	28
3.2. Предлагаемый метод решения задачи. ....	29
3.3. Алгоритм работы детектора. ....	29
3.4. Признаки цифрового изображения. ....	30
Глава 4. Результаты экспериментов. ....	32
4.1. Тестовые данные для проведения экспериментов. ....	32
4.2. Результаты экспериментов. ....	32
Заключение. ....	40
Список использованной литературы. ....	41

## **Введение**

### **Актуальность проблемы**

В настоящее время цифровая обработка изображений является одной из наиболее актуальных и быстро развивающихся областей науки и техники. Это вызвано как многообразием практических задач, требующих анализа визуальной информации, так и все возрастающими возможностями оптики и вычислительной техники. Все многообразие задач цифровой обработки изображений можно разделить на две основные категории: задачи, связанные с обработкой отдельных статических изображений, и задачи, в которых исходным материалом для обработки служат видеопоследовательности. Изображения, входящие в видеопоследовательности, обычно бывают сформированы одним и тем же датчиком изображений, но в разные моменты времени, из-за чего видеопоследовательности содержат информацию об изменениях, происходящих в наблюдаемой сцене с течением времени [2].

Системы обработки и анализа видеоинформации все более интенсивно применяются в различных областях человеческой деятельности. Наиболее широкое распространение они получили при создании бортовых и стационарных систем, способных в режиме реального времени решать множество задач, связанных с автоматическим обнаружением, слежением и сопровождением объектов. Для таких систем примерами отслеживаемых объектов могут служить летящие самолеты, вертолеты, движущиеся по шоссе автомашины, суда и люди. В качестве фонового изображения могут выступать изображения лесов, домов, дорог или облаков. При этом информация о характеристиках объектов, которые требуется обнаруживать и отслеживать, как правило, бывает очень скудна и, в лучшем случае, включает в себя лишь приблизительные размеры объектов или их начальные координаты [2].

### **Цели и задачи выпускной квалификационной работы**

Целью данной работы является разработка и программная реализация метода, способного решать задачу долгосрочного слежения, предполагающего наличие минимальной начальной информации.

Основные задачи, решение которых необходимо для достижения поставленной цели:

- изучение классических методов решения задачи обнаружения и слежения за объектами видеопоследовательности;
- программная реализация и оценка эффективности работы трекера, основанного на вычитании изображения фона, а также разработка методов, способных улучшить результаты его работы;
- изучение классических методов решения задачи распознавания образов;
- программная реализация и оценка эффективности работы детектора, основанного на использовании случайного леса;
- разработка и программная реализация метода решения задачи долгосрочного слежения, предполагающего наличие минимальной начальной информации;
- оценка и анализ результатов работы предложенного метода на примере тестовых видеопоследовательностей.

## **Содержание работы**

В первой главе производится обзор классических методов решения задачи обнаружения и слежения за объектами видеопоследовательности. Предлагаются методы, способные улучшить результаты работы трекера, основанного на вычитании изображения фона, а также приводятся результаты их применения.

Во второй главе производится обзор классических методов решения задачи распознавания образов. Рассматривается классификатор, основанный на случайном лесе, а также его модификация, способная обучаться в режиме реального времени.

В третьей главе описывается применение методов распознавания образов к задаче обнаружения и слежения. Представлены теоретические основы разрабатываемого метода долгосрочного слежения.

В четвертой главе показаны результаты работы предложенного метода при различных изменениях сцены на кадрах тестовых видеопоследовательностей, представлен анализ полученных результатов. Приведена сравнительная характеристика разработанного метода с классическим методом обнаружения и слежения.

В заключении описаны перспективы дальнейшего развития метода, предложенного в рамках данной работы.

# Глава 1. Обнаружение и слежение за объектами видеопоследовательности

## 1.1. Видеопоследовательность

Одной из важнейших компонент системы обнаружения и слежения является датчик изображений, поскольку именно он служит главным, а зачастую и единственным источником информации, необходимой для решения задачи обнаружения и слежения. Объем информации, генерируемой датчиком, определяется количеством элементов светочувствительной матрицы датчика, количеством спектральных каналов наблюдения и частотой формирования кадров [2].

Наиболее широкое распространение в системах слежения получили одноканальные датчики видимого диапазона. Как правило, эти датчики формируют на выходе видеопоток, который представляет собой последовательность цифровых изображений.

Цифровым изображением называется массив данных, полученный путем аналого-цифрового преобразования сигнала, поступающего от датчика изображений. Существуют три основных типа представления цифровых изображений: растровая графика, векторная графика и фрактальная графика. Растровое изображение – это изображение, представляющее собой двумерную матрицу  $I$ , каждый элемент  $I(x, y)$  которой характеризует цвет соответствующего пикселя. Пикселем называется наименьший логический элемент цифрового изображения. Каждый пиксель растрового изображения может кодироваться различным количеством бит, что определяет количество возможных значений яркости. Разрешение цифрового изображения – это величина, определяющая количество пикселей изображения на единицу площади.

Изображение оптического поля, получаемое с помощью датчика изображений, подвергается дискретизации по пространственным координатам и по времени – с частотой поступления кадров. Таким образом, с датчика изображений на устройство анализа поступает последовательность матриц чисел  $\{I_t(x, y)\}$ , называемая видеопоследовательностью, каждый элемент которой представляет собой измерение яркости в соответствующей точке на каждом кадре. Здесь  $x$  и  $y$  – индексы изображения по пространственным координатам,  $t$  – номер кадра [2].

В данной работе использовались видеопоследовательности, состоящие из цифровых растровых одноканальных изображений с разрешением  $640 \times 480$  пикселей и числом бит на пиксель равным 8.

## **1.2. Задача обнаружения и слежения за объектами**

Одной из задач, решаемых системами обнаружения и слежения, является задача обнаружения движущихся или появляющихся на изображении объектов. После решения задачи первичного обнаружения объекта, производимого в автоматическом режиме или с помощью оператора, как правило, решается задача отслеживания траектории движения цели.

Существует два основных метода отслеживания траектории движения объекта. Первый метод, называемый слежением, предполагает оценку координат объекта на каждом кадре наблюдаемой видеопоследовательности без решения задачи управления перемещением датчика изображений. Последовательность координат при этом формирует траекторию движения объекта. Методы, решающие задачу слежения, также называются методами трекинга, а устройство, способное решать эту задачу – трекером. Второй метод, не рассматриваемый в данной работе, называется сопровождением и предусматривает наряду с определением координат объекта на каждом кадре еще и решение задачи управления перемещением датчика изображений. Цель такого управления, чаще всего, состоит в удержании объекта сопровождения в центре поля зрения видеокамеры или в исключении потери визуального контакта с объектом сопровождения вследствие движения датчика изображений и самого объекта в пространстве [2].

Решение задачи обнаружения и слежения осуществляется в несколько этапов. Целью начального этапа является выделение точек текущего кадра, принадлежащих объектам, то есть разделение изображения на сегменты, представляющие его смысловые области. Данный процесс называется сегментацией и может осуществляться различными методами. На последующих этапах, с учетом уже имеющейся информации, решаются задачи оценки координат и параметров объектов. Последним этапом является слежение, которое состоит в нахождении траектории движения объектов во времени, основываясь на анализе изменений координат объектов на кадрах видеопоследовательности.

### 1.3. Сегментация изображений

Сегментация изображения – разделение изображения на сегменты, то есть на ряд составляющих его смысловых областей. При этом предполагается, что точки каждой из смысловых областей обладают некоторой степенью общности, которая позволяет рассматривать их как одну область [2]. Результат сегментации обычно представляется в виде бинарного изображения  $BIN(x, y)$ , имеющего столько же элементов, сколько их содержит исходное изображение  $I(x, y)$ . В ходе сегментации всем точкам бинарного изображения, принадлежащим одному классу, присваивается одно и то же значение яркости, которое выбирается из множества  $\{0, 1\}$ . При этом значение 1 присваивается точкам, принадлежащим областям, в которых предположительно находится объект, а значение 0 – точкам, соответствующим областям, предположительно содержащим фон.

Существует большое число подходов к решению задачи сегментации, обладающих своими достоинствами и недостатками. Большинство из них основаны на анализе признаков точек изображения. Признаки точек – это скалярные или векторные величины, вычисляемые с помощью яркостного поля изображения, выбор которых решающим образом влияет на правильность сегментации [2].

#### 1.3.1. Метод пороговой сегментации

Наиболее простым методом сегментации является пороговая обработка. Результат пороговой обработки изображения  $I(x, y)$  может быть описан соотношением:

$$BIN(x, y) = \begin{cases} 1, & I(x, y) > T(x, y) \\ 0, & I(x, y) \leq T(x, y) \end{cases},$$

где  $BIN(x, y)$  – результат сегментации;  $T(x, y)$  – значение порога, которое в общем случае может быть различным для каждой точки изображения [2].

Одним из подходов к определению порогов является использование гистограмм яркостей изображений. Основная идея данного метода заключается в предположении, что изображение содержит известное число однородных по яркости классов точек, то есть таких классов, распределения вероятностей которых унимодальны. Кроме того, предполагается, что граничные участки между замкнутыми областями занимают сравнительно небольшую площадь изображения.

Поэтому на гистограмме яркостей им должны соответствовать межмодовые впадины, в которых и устанавливаются пороги сегментации [2].

Использование методов пороговой обработки обычно ограничено ситуациями, при которых выполняется предположение об унимодальности распределения яркостей объектов и фона. Это ограничение позволяет использовать методы пороговой обработки лишь в простейших случаях [2].

### **1.3.2. Метод сегментации на основе выделения границ**

Другим популярным методом сегментации является метод, основанный на выделении границ объектов. Этот метод базируется на предположении о существовании резкого перепада яркости между областями изображения, соответствующими различным смысловым участкам. Для выделения границ могут быть использованы различные подходы, основным из которых является пространственное дифференцирование. Данный подход основан на предположении о том, что граничные точки смысловых областей изображения  $I(x, y)$  имеют большую величину градиента

$$\nabla I(x, y) = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right).$$

Для того чтобы выделить границы областей, вначале вычисляют модуль градиента изображения  $|\nabla I(x, y)|$ , а полученное градиентное изображение подвергают пороговой обработке. После этого выделяются сегменты и производится их оценка по пространственным осям координат, а также рассчитываются коэффициенты, характеризующие форму и другие признаки, которые могут быть использованы для принятия решения о присутствии искомого объекта на изображении и оценки его координат [2].

Так как растровое изображение является дискретным, при его обработке для вычисления градиента обычно используют дискретные оценки производных яркости изображения, вычисленные по двум взаимно перпендикулярным направлениям.

При работе с реальными изображениями предположение о наличии резких границ между смысловыми областями часто не выполняется. Кроме того, смысловые области нередко содержат внутри себя резкие перепады яркости. Эти обстоятельства ведут к низкой эффективности применения методов выделения границ при обработке таких изображений. Также следует отметить, что методы выделения границ



показывают низкие результаты при работе с сильно зашумленными изображениями, что является следствием использования операций дискретного дифференцирования.

Часто для улучшения результатов работы метода применяют дополнительную обработку контурного изображения, которая включает в себя прослеживание и уточнение контуров, восстановление разрывов и удаление одиночных точек [2].

### 1.3.3. Метод сегментации на основе вычитания изображения фона

Данный метод основан на построении изображения фона с последующим вычитанием его из изображения текущего кадра. Изображение фона строится на основе модели фона, состоящей из двух частей: среднего фоновое изображение  $B$  и изображения средних абсолютных отклонений  $D$ .

Среднее фоновое изображение  $B$  формируется путем осреднения  $N$  последних кадров  $I_{t-N+1}, \dots, I_t$  ( $t = N, N + 1, \dots$ ) видеопоследовательности. При этом обновление среднего фоновое изображения новыми кадрами происходит только теми участками, где нет движущихся объектов. Пусть  $B_{t-1}$  – среднее фоновое изображение, сформированное к кадру  $I_{t-1}$ ;  $S[I, I^*]$  – оператор позволяющий получить изображение,  $I$  без движущихся объектов:

$$S[I(x, y), I^*(x, y)] = \begin{cases} I^*(x, y), & (x, y) - \text{точка движущегося объекта} \\ I(x, y), & \text{в противном случае} \end{cases}$$

Тогда среднее фоновое изображение для кадра  $I_t$  вычисляется по формуле:

$$B_t = (1 - \alpha) \cdot B_{t-1} + \alpha \cdot S[I_t, B_{t-1}], \quad \alpha = \frac{1}{N}, \quad B_1 = I_1$$

Изображение средних абсолютных отклонений  $D$  является характеристикой средней изменчивости фона и вычисляется путем осреднения  $N$  последних абсолютных отклонений кадров  $I_{t-N+1}, \dots, I_t$  от фоновых изображений  $B_{t-N+1}, \dots, B_t$ . Изображение средних абсолютных отклонений, так же как и среднее фоновое изображение, обновляется только теми участками, где нет движущихся объектов. Пусть  $D_{t-1}$  – изображение средних абсолютных отклонений, сформированное к кадру  $I_{t-1}$ . Тогда изображение средних абсолютных отклонений для кадра  $I_t$  вычисляется следующим образом:

$$D_t = (1 - \alpha) \cdot D_{t-1} + \alpha \cdot S[|I_t - B_t|, D_{t-1}], \quad \alpha = \frac{1}{N}, \quad D_1 = I_1$$

Для определения точек, принадлежащих движущимся объектам, используется предположение о том, что абсолютное отклонение значений движущихся точек на

текущем кадре от среднего фонового изображения превышает их значение на изображении средних абсолютных отклонений. То есть движущимися считаются такие точки  $(x, y)$ , для которых  $|I_t(x, y) - B_{t-1}(x, y)| > k_1 \cdot D_{t-1}(x, y)$ , где  $k_1 \geq 1$ .

Бинарное изображение  $BIN$  на кадре  $t$  вычисляется на основе полученной модели фона по формуле:

$$BIN_t(x, y) = \begin{cases} 1, & |I_t(x, y) - B_t(x, y)| > k_2 \cdot D_t(x, y) \\ 0, & \text{в противном случае} \end{cases}$$

где  $k_2 > k_1 \geq 1$ . Таким образом, бинарное изображение принимает значение, равное единице, в тех точках, которые изменяют свое положение от кадра к кадру.

#### **1.4. Вычисление пространственных параметров объектов**

Выше были рассмотрены алгоритмы сегментации изображений, результатом работы которых являлись бинарные изображения. Если сегментация изображения была выполнена довольно точно, то бинарное изображение содержит исчерпывающую информацию, позволяющую вычислить практически любые пространственные параметры наблюдаемых объектов [2].

Алгоритм вычисления пространственных параметров состоит из двух частей: выделения объектов на основе бинарного изображения и непосредственного вычисления пространственных параметров объектов.

Выделение объектов происходит следующим образом. Среди множества точек бинарного изображения находятся связные области, то есть такие области, которые представляют собой совокупности связных элементов бинарного изображения. Связными называются такие два соседних элемента в строке или столбце матрицы, соответствующей изображению, которые имеют одинаковое значение яркости. После чего каждая связная область окружается прямоугольным окном минимально возможного размера так, чтобы все элементы связной области попали в него. Таким образом, полученное окно окружает объект исходного изображения, представленный на бинарном изображении связной областью. Вычисление пространственных параметров объекта, таких как его размер и положение в пространстве, происходит на основе оценки координат построенного окна.

## 1.5. Учет движения камеры в методе трекинга на основе вычитания фона

При решении задачи обнаружения и слежения часто бывает невозможным обеспечить надежную механическую стабилизацию датчика изображений. Вследствие чего, датчик может изменять свое положение в пространстве из-за вибраций и небольших сдвигов. Очевидно, что при изменении пространственных координат датчика изображений, меняются и координаты объектов наблюдения на его матрице, из-за чего возникают геометрические преобразования изображений, входящих в видеопоследовательность. Стоит отметить, что для решения задачи обнаружения и слежения при движущемся датчике изображений, крайне важно как можно более точно оценить параметры геометрических преобразований, вызванных движением датчика [2].

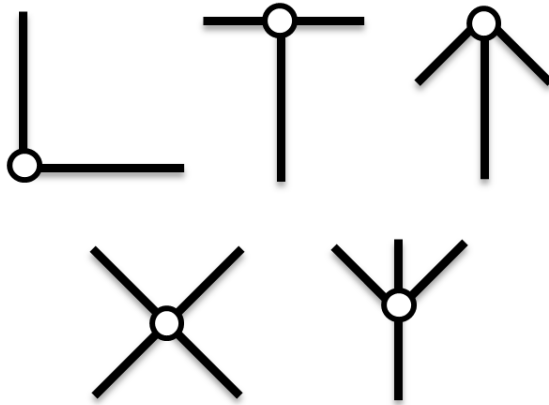
В рамках данной работы предлагается метод компенсации смещений кадров видеопоследовательности, вызванных движением датчика изображений в плоскости, параллельной наблюдаемой. При этом в алгоритме построения модели фона, используемой в методе сегментации, основанной на вычитании фонового изображения, необходимо произвести следующие изменения: формировать среднее фоновое изображение и изображение средних абсолютных отклонений с учетом взаимных смещений кадров видеопоследовательности.

Существует большое число подходов к оценке параметров геометрических преобразований изображений: корреляционный, спектральный, оценка на основе опорных элементов. Корреляционные и спектральные методы оценки опираются на использование яркостной информации изображения, поэтому их главным недостатком является низкая устойчивость к изменению освещенности в области наблюдения. Оценка на основе опорных элементов, использованная в предлагаемом методе компенсации смещений кадров, состоит в поиске опорных элементов на каждом кадре видеопоследовательности, определении их координат, а также нахождении параметров геометрических преобразований.

### 1.5.1. Опорные элементы

В качестве опорных элементов изображений в предлагаемом методе компенсации смещений кадров были использованы угловые особые точки. Особой точкой  $m$  называется такая точка изображения, окрестность которой  $\Omega(m)$  однозначно отличима от окрестности любой другой точки  $n \in \Omega^*(m)$ , где  $\Omega^* > \Omega$  [4].

Угловые особые точки – это особые точки, находящиеся на пересечении двух или более граней, определяющих границу между различными объектами изображения. Главным достоинством использования таких особых точек является то, что в их окрестности у градиента изображения преобладают два доминирующих направления, что позволяет относительно просто осуществлять их поиск.



(1)



(2)

**Рисунок 1.** Угловые особые точки. Характерные позиции угловых особых точек (1). Положения угловых особых точек на изображении, найденные детектором Харриса (2).

Одним из популярных детекторов угловых особых точек является детектор Харриса. В нем используется окно  $W$  размера  $n \times n$  пикселей с центром в точке  $(x^*, y^*)$ , а также сдвиг этого окна на величину  $(\Delta x, \Delta y)$ . Взвешенная сумма квадратов разностей между сдвинутым и исходным окном, то есть изменение окрестности точки  $(x^*, y^*)$ , при сдвиге на  $(\Delta x, \Delta y)$ , равна величине

$$E(\Delta x, \Delta y) = \sum_{(x_i, y_j) \in W} \omega(x_i, y_j) [I(x_i + \Delta x, y_j + \Delta y) - I(x_i, y_j)]^2,$$

где  $\omega(x_i, y_j)$  – весовая функция, в качестве которой обычно используется двумерный гауссиан [11]. Используя ряд Тейлора для вычисления значений интенсивностей сдвинутых точек  $I(x_i + \Delta x, y_j + \Delta y)$ , величину  $E(\Delta x, \Delta y)$  можно выразить следующим образом:

$$E(\Delta x, \Delta y) = [\Delta x \ \Delta y] \cdot M \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad M = \sum_{(x_i, y_j) \in W} \omega(x_i, y_j) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Угловая особая точка характеризуется большими изменениями функции  $E(\Delta x, \Delta y)$  по всем возможным направлениям  $(\Delta x, \Delta y)$ , что эквивалентно большим по модулю собственным значениям  $(\lambda_1, \lambda_2)$  матрицы  $M$ . Поскольку прямое вычисление

собственных значений матрицы является трудоёмкой задачей, Харрисом была предложена мера отклика:

$$R = \det M - k \cdot (\text{trace} M)^2 > k,$$

где  $\det M$  – определитель матрицы  $M$ ,  $\text{trace} M$  – след матрицы  $M$ ,  $k$  – эмпирическая константа [11].

После вычисления меры отклика  $R$  производится отсечение точек по этому порогу. Далее по окрестности заданного радиуса находятся локальные максимумы функции отклика, которые и выбираются в качестве угловых особых точек.

### 1.5.2. Оценка смещений опорных элементов

Для оценки смещений опорных элементов в предлагаемом методе компенсации смещений кадров был использован метод вычисления оптического потока. Оптическим потоком называется изображение видимого движения, представляющее собой сдвиг каждой точки между двумя изображениями. Суть вычисления оптического потока состоит в том, чтобы для каждой точки изображения  $I_1(x, y)$  найти такой сдвиг  $(\Delta x, \Delta y)$ , что точке на изображении  $I_1(x, y)$  соответствовала бы точка на изображении  $I_2(x + \Delta x, y + \Delta y)$  [10].

Одним из популярных методов вычисления оптического потока является метод Лукаса-Канаде. Данный метод основан на предположении, что значения пикселей переходят из кадра  $I_t$  в кадр  $I_{t+1}$  без изменений [10]. Таким образом, мы делаем допущение, что пиксели, относящиеся к одному и тому же объекту, могут смещаться от кадра к кадру, но их значения остаются неизменным, то есть

$$I_t(x, y) = I_{t+1}(x + \Delta x, y + \Delta y).$$

Также делается допущение, что между двумя кадрами видеопоследовательности проходит единичный интервал времени, то есть можно считать, что

$$I_t(x, y) - I_{t+1}(x, y) = \frac{\partial I_t(x, y)}{\partial t}.$$

Используя сделанные допущения и разложение  $I_t(x + \Delta x, y + \Delta y)$  в ряд Тейлора, можно получить следующее равенство:

$$\frac{\partial I_t(x, y)}{\partial t} + \frac{\partial I_t(x, y)}{\partial x} \Delta x + \frac{\partial I_t(x, y)}{\partial y} \Delta y = 0.$$

Предполагается, что для пикселей из фрагмента изображения  $W$  размером  $n \times n$  смещения одинаковы и равны  $(\Delta x, \Delta y)$ . Тогда вместо одного уравнения

получается система из  $n^2$  уравнений, которая в общем случае не имеет решения. Поэтому в качестве величины сдвига выбирается такое значение  $(\Delta x, \Delta y)$ , которое минимизирует ошибку

$$E(\Delta x, \Delta y) = \sum_{(x_i, y_j) \in W} \omega(x_i, y_j) \left[ \frac{\partial I_t(x_i, y_j)}{\partial t} + \frac{\partial I_t(x_i, y_j)}{\partial x} \Delta x + \frac{\partial I_t(x_i, y_j)}{\partial y} \Delta y \right]^2,$$

где  $\omega(x_i, y_j)$  – весовая функция, в качестве которой обычно используется двумерный гауссиан [10].

### 1.5.3. Смещение кадра

Оценка смещения текущего кадра видеопоследовательности относительно предыдущего производится на основе смещений опорных элементов, вычисляемых методом Лукаса-Канаде. При этом смещение всего кадра  $(\Delta x, \Delta y)$  определяется как медиана смещений опорных элементов.

После определения смещения кадра производится его параллельный перенос на величину  $\Delta x$  по горизонтали и на  $\Delta y$  по вертикали. Так как значения сдвигов, полученных данным образом, могут быть не только целочисленными, для нецелых частей сдвигов выполняется субпиксельное смещение с помощью билинейной интерполяции, используя следующую формулу:

$$I(x + u, y + v) = I_1 \cdot |(1 - u)(1 - v)| + I_2 \cdot |u(1 - v)| + I_3 \cdot |(1 - u)v| + I_4 \cdot |uv|,$$

где  $u$  и  $v$  – дробные части  $\Delta x$  и  $\Delta y$ ,  $I_1 = I(x, y)$ ,

$$I_2 = \begin{cases} I(x + 1, y), & u \geq 0 \\ I(x - 1, y), & v < 0 \end{cases} \quad I_4 = \begin{cases} I(x + 1, y + 1), & u \geq 0, v \geq 0 \\ I(x - 1, y + 1), & u < 0, v \geq 0 \\ I(x + 1, y - 1), & u \geq 0, v < 0 \\ I(x - 1, y - 1), & u < 0, v < 0 \end{cases}$$

$$I_3 = \begin{cases} I(x, y + 1), & u \geq 0 \\ I(x, y - 1), & v < 0 \end{cases}$$



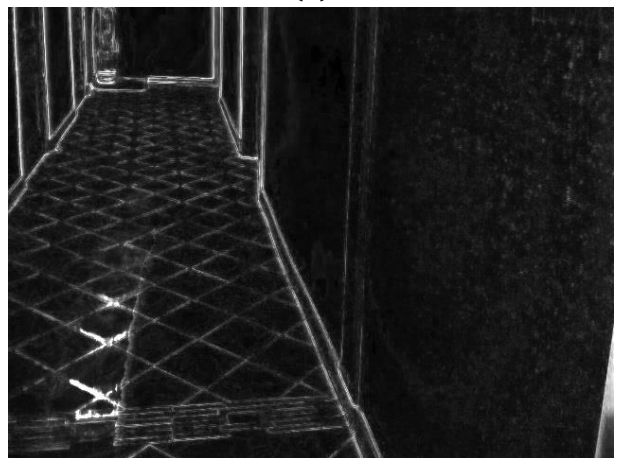
(1)



(2)



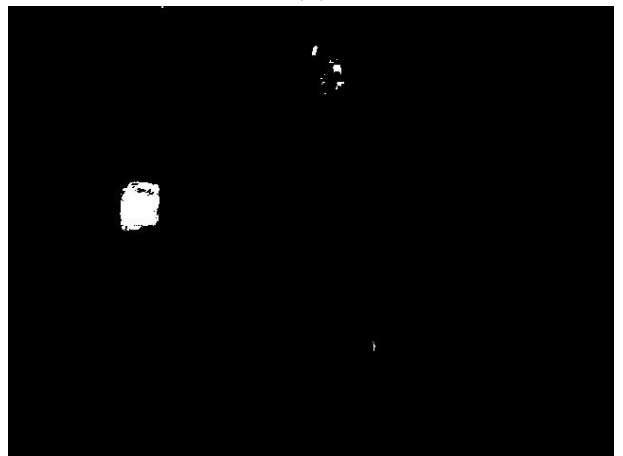
(3)



(4)



(5)



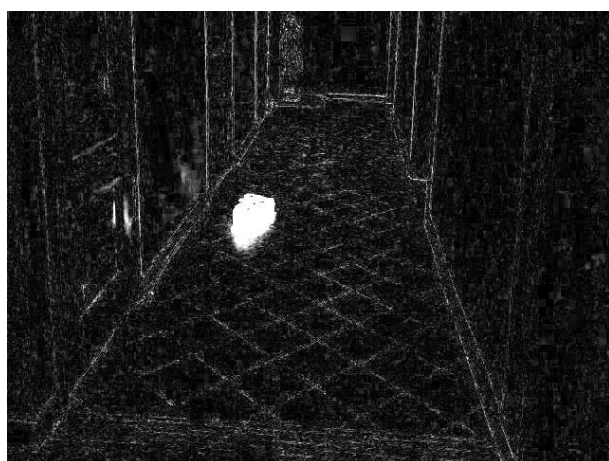
(6)

**Рисунок 2.** Результат учета движения камеры в методе трекинга на основе вычитания фона. Среднее фоновое изображение  $B_t$ : без учета движения камеры (1), с учетом (2). Масштабированное по яркости изображение средних абсолютных отклонений  $D_t$ : (3), (4). Бинарное изображение  $BIN_t$ , полученное в результате сегментации: (5), (6).

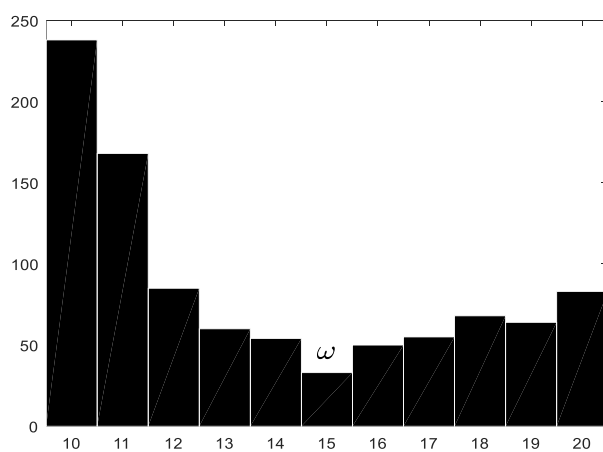
## 1.6. Порог фоновой части в методе трекинга на основе вычитания фона

Метод вычисления бинарного изображения, описанный в §1.3.3, обладает существенным недостатком. Из-за зашумленности исходного изображения  $I_t$  некоторые пиксели, принадлежащие фону, имеют большое отклонение от среднего фонового изображения, вследствие чего такие пиксели на бинарном изображении  $BIN_t$  имеют значения равные единице, что соответствует подвижному объекту.

Данный недостаток устраняется применением порога фоновой части для изображения  $BIN_t$ . Основная идея данного метода заключается в предположении, что изображение  $|I_t - B_t|$  содержит два относительно однородных по яркости класса точек, принадлежащих объекту и фону соответственно. Поэтому на гистограмме яркостей  $H_t$  им должна соответствовать межмодовая впадина, в которой и устанавливается порог фоновой части изображения.



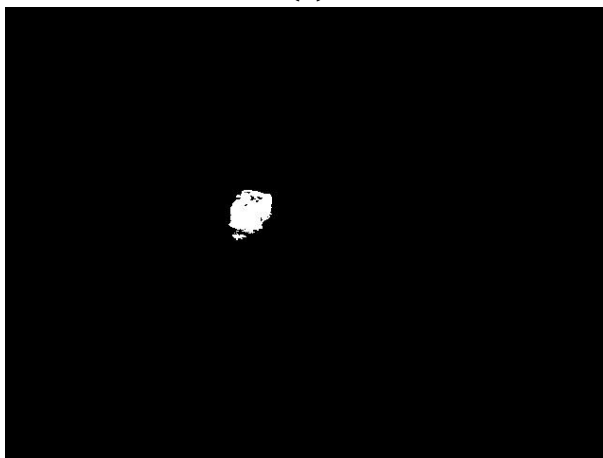
(1)



(2)



(3)



(4)

**Рисунок 3.** Результат применения порога фоновой части. Масштабированное по яркости изображение  $|I_t - B_t|$  (1). Гистограмма изображения  $|I_t - B_t|$  (2). Бинарные изображения, полученные в результате сегментации: без применения порога (3), с применением (4).



Применение порога фоновой части может быть описано следующим выражением:

$$\widetilde{BIN}_t(x, y) = \begin{cases} BIN_t(x, y), & |I_t(x, y) - B_t(x, y)| \geq \omega \\ 0, & \text{в противном случае} \end{cases}$$

где  $\widetilde{BIN}_t(x, y)$  – бинарное изображение, полученное с применением порога;  $\omega$  – порог фоновой части изображения, вычисляемый следующим образом:

$$\omega = \min_{v \in [v_0; 255]} \{H_t(v+1) \geq H_t(v)\},$$

$$v_0 = \begin{cases} 1, & H_t(0) > H_t(1) \\ \min_{v \in [1; 255]} \{H_t(v+1) < H_t(v)\}, & H_t(0) \leq H_t(1) \end{cases}$$

## Глава 2. Распознавание образов и классификация

### 2.1. Задача распознавания образов

Распознавание образов – это задача, целью которой является распределение объектов по нескольким категориям или классам. Распознаваемые объекты называются образами и описываются векторами признаков  $\mathbf{x}$ . Признаком называется некоторое количественное измерение объекта, которое может быть выполнено в различных шкалах: числовой, порядковой или номинальной. Множество всех возможных значений векторов признаков  $\mathbf{x}$  называется пространством образов  $X$ . Ответы распознавателя представляют собой вектора  $\mathbf{y}$  из некоторого пространства ответов  $Y$  [5].

Используя вероятностную постановку задачи классификации, задачу распознавания образов можно записать следующим образом. Пусть множество пар (объект, метка класса)  $X \times Y$  является вероятностным пространством с неизвестной вероятностной мерой  $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$ . Имеется конечная обучающая выборка наблюдений  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$ , сгенерированная согласно вероятностной мере  $P$ . Требуется построить алгоритм  $f: X \rightarrow Y$ , способный классифицировать произвольный объект  $\mathbf{x} \in X$ . Здесь  $\mathbf{x}_i = \{x_i^s\}_{s=1}^n$  – вектор, координаты которого представляют собой значения  $n$  признаков, измеренных на некотором объекте;  $y_i$  – метка класса,  $y_i \in \{\omega_j\}_{j=1}^c$ ;  $c$  – число классов [[5], [7]].

## **2.2. Методы распознавания образов**

В настоящее время существует множество различных подходов к решению задачи распознавания образов. Происхождение каждого из них связано с тем или иным представлением о том, как описываются образы и каким способом наиболее успешно можно решить задачу их распознавания [3].

### **2.2.1. Статистические методы распознавания образов**

В статистических методах распознавания предполагается, что объекты обучающей выборки и распознаваемые объекты принадлежат одной и той же генеральной совокупности. Считается, что существует совместное вероятностное распределение элементов данной генеральной совокупности по классам  $\{\omega_j\}_{j=1}^c$  и в пространстве образов  $X$ . Очевидно, что в случаях, когда такое распределение известно, существует оптимальное решение задачи определения принадлежности меток классов  $y_i$  объектам [3].

Предположим, что нам необходимо классифицировать некоторый объект, описываемый вектором признаков  $\mathbf{x}$ . Для каждого из классов  $\{\omega_j\}_{j=1}^c$  вычисляется условная вероятность принадлежности  $P(y|\mathbf{x})$ , где  $y \in \{\omega_j\}_{j=1}^c$ . Объект относится к тому классу, для которого условная вероятность принадлежности максимальна. Данное решающее правило, называемое байесовским, минимизирует вероятность ошибочной классификации [3]. На практике плотности распределения классов, как правило, не известны и оцениваются по обучающей выборке. В результате байесовское правило перестает быть оптимальным, так как восстановить плотности распределения удастся лишь с некоторой погрешностью.

### **2.2.2. Методы распознавания образов, основанные на построении разделяющих поверхностей**

В основе данных методов лежит геометрическое представление модели классов. Предполагается, что множеству объектов каждого класса соответствует определенная область в  $n$ -мерном пространстве образов  $X$ . Также предполагается, что данные области имеют достаточно простую форму и их можно разделить линейной, кусочно-линейно или квадратичной поверхностью (гиперплоскостью) [3].

Задача построения линейной разделяющей поверхности состоит в вычислении некоторой линейной относительно признаков функции:

$$f(\mathbf{x}) = a_1x_1 + a_2x_2 + \dots + a_nx_n + a_{n+1}$$

и использовании при классификации решающего правила:

$$y = \begin{cases} \omega_1, & f(\mathbf{x}) > 0 \\ \omega_2, & f(\mathbf{x}) < 0 \end{cases}.$$

Здесь для простоты считается, что имеются лишь два класса объектов;  $y = \omega_1$  означает отнесение объекта, признаковое описание которого представлено вектором  $\mathbf{x}$  к первому классу;  $y = \omega_2$  – отнесение ко второму.

Следует отметить, что методы классификации, основанные на построении линейных разделяющих поверхностей, эффективны лишь для относительно простых практических задач [3].

### **2.2.3. Методы распознавания образов, основанные на построении решающих деревьев и лесов**

Для данного класса методов распознавание объекта осуществляется путем прохождения по бинарному дереву из корня в некоторую терминальную вершину. Каждая терминальная вершина связана с одним из классов, к которому и относится распознаваемый объект. На пути от корня в каждой вершине дерева вычисляется значение определенной логической функции, в зависимости от которого происходит переход далее по дереву в левую или правую вершину следующего уровня [3].

Лес решений представляет собой множество деревьев, чьи решения комбинируются некоторым образом для получения окончательной классификации объекта. Обычно синтез решений отдельных деревьев осуществляется путем их взвешенного или невзвешенного голосования [7].

### **2.3. Случайный лес**

Случайный лес – это метод, предназначенный для решения задач классификации и регрессии, заключающийся в использовании ансамбля классификаторов, представляющих собой деревья решений. Метод сочетает в себе две основные идеи: метод баггинга и метод случайных подпространств [7].

### 2.3.1. Деревья решений

Пусть  $V$  – конечное непустое множество,  $E$  – конечное множество, состоящее из поименованных неупорядоченных пар элементов множества  $V$ . Совокупность множеств  $V$  и  $E$  называют неориентированным графом и обозначают  $G = (V, E)$ . Элементы  $v \in V$  называют вершинами, а элементы  $e \in E$  – ребрами. Если  $e$  – это неупорядоченная пара вершин  $v_i$  и  $v_j$ , то пишут  $e = v_i v_j$ , причем вершины  $v_i$  и  $v_j$  называют концами ребра  $e$  [6].

Пусть  $U$  – конечное множество, состоящее из поименованных упорядоченных пар элементов множества  $V$ . Тогда совокупность множеств  $V$  и  $U$  называют ориентированным графом (орграфом) и обозначают  $G = (V, U)$ . Элементы  $u \in U$  называют ориентированными ребрами или дугами. Если дуга  $u$  есть упорядоченная пара вершин  $v_i$  и  $v_j$ , то пишут  $u = (v_i, v_j)$ , причем вершину  $v_i$  называют началом дуги  $u$ , а вершину  $v_j$  ее концом. Число дуг, исходящих из вершины  $v_i$  орграфа  $G$ , называют степенью исхода вершины  $v_i$ , а число дуг, заходящих в вершину  $v_i$ , называют степенью захода вершины  $v_i$  [6].

Путем длины  $k$  на графе  $G$  из вершины  $v_0$  в вершину  $v_k$  называется последовательность  $v_0 e_1 v_1 \dots v_{k-1} e_k v_k$  или  $v_0 u_1 v_1 \dots v_{k-1} u_k v_k$  вершин и ребер графа, в которой  $e_i = v_{i-1} v_i$ ,  $u_i = (v_{i-1}, v_i)$ ,  $1 \leq i \leq k$  [6]. Связным называется граф, между любой парой вершин которого существует как минимум один путь. Циклом называется замкнутый путь ненулевой длины без повторяющихся ребер. Граф без циклов называется ациклическим.

Дерево – связный ациклический граф. Ориентированным деревом  $T$  называется ациклический связный орграф, в котором только одна вершина имеет нулевую степень захода, а все остальные вершины имеют степень захода 1. Вершина с нулевой степенью захода называется корнем дерева, а вершины с нулевой степенью исхода называются листьями или терминальными вершинами.

Для каждого листа в ориентированном дереве существует единственный путь из корня. Максимальная длина пути из корня в лист называется высотой дерева. Если из вершины  $a \in T$  ведет дуга в вершину  $b \in T$ , то  $a$  называется отцом, а  $b$  – сыном [6].

Деревом решений называться ориентированное дерево  $T$ , с каждой вершиной  $t$  которого связаны [1]:

1. Подмножество  $X_t \subset X$ ;

2. Подвыборка  $D_t \subset D$ , такая, что  $D_t = \{(\mathbf{x}, y) \in D \mid \mathbf{x} \in X_t\}$ ;
3. Функция (правило)  $f_t: X \rightarrow \{0, 1, \dots, k_t - 1\}$  определяющая разбиение множества  $X$  на  $k_t$  непересекающихся подмножеств. Здесь  $k_t \geq 2$  – количество сыновей вершины  $t$ .

Стоит отметить, что с корневой вершиной связывается все пространство образов  $X$  и вся выборка  $D$ . С листьями дерева не связываются никакие функции  $f_t$ .

Пусть  $t_i$  – вершина, являющаяся  $i$ -м сыном вершины  $t$ . Тогда множество  $X_t$  и функция  $f_t$  определяют подмножества  $X_{t_i}$  следующим образом:

$$X_{t_i} = X_t \cap \{\mathbf{x} \in X \mid f_t(\mathbf{x}) = i\}.$$

Цель построения дерева решений состоит в классификации векторов  $\mathbf{x}$  из распределения  $P(\mathbf{x})$ . Процесс принятия решений начинается с корневой вершины и состоит в последовательном применении правил  $f_t$ , связанных с вершинами дерева. Результатом этого процесса является определение листа  $t$  такого, что  $\mathbf{x} \in X_t$ , при этом вектор  $\mathbf{x}$  относится к классу, являющемуся мажорантным в подвыборке  $D_t$  [7].

### 2.3.2. Мера загрязненности вершины

Алгоритм построения дерева решений основан на идее рекурсивного разбиения обучающей выборки на две более однородные подвыборки с помощью одного из признаков. Обычно вместо меры однородности используется противоположная по смыслу мера загрязненности [7].

Пусть  $i(t)$  – мера загрязненности вершины  $t$ ,  $D_t$  – подвыборка, связанная с этой вершиной. Необходимо потребовать, чтобы загрязненность вершины была равна 0, если  $D_t$  содержит прецеденты только одного класса, и была максимальной в случае, если  $D_t$  содержит одинаковое число прецедентов каждого класса [7]. Одной из наиболее часто используемых является мера загрязненности вершины, формализованная в индексе Gini:

$$i(t) = 1 - \sum_{j=1}^c P^2(\omega_j),$$

где  $P(\omega_j)$  – доля примеров класса  $\omega_j$  в подвыборке  $D_t$ . Другой популярной мерой является мера загрязненности, основанная на понятии энтропии:

$$i(t) = - \sum_{j=1}^c P(\omega_j) \log_2 P(\omega_j).$$

Реже применяется мера загрязненности, основанная на частоте ошибочной классификации:

$$i(t) = 1 - \max_j P(\omega_j).$$

В работе [9] было показано, что выбор меры загрязненности не оказывает существенного влияния на точность классификации, поэтому, в силу простоты реализации, в качестве используемой в данной работе меры был выбран индекс Gini.

### 2.3.3. Оптимальное расщепление вершины

Правило разбиения множества  $X$ , связанное с каждой вершиной дерева решений, называется расщеплением. Бинарное расщепление вершины  $t$  можно рассматривать как функцию  $f_t: X \rightarrow \{0,1\}$ ,  $\mathbf{x} \in X$ , где в случае  $f_t(\mathbf{x}) = 0$  вектор  $\mathbf{x}$  относится к левому сыну вершины  $t$ , а в случае  $f_t(\mathbf{x}) = 1$  – к правому [7].

Расщепление подвыборки необходимо осуществлять таким образом, чтобы максимально уменьшить загрязненность. Уменьшение загрязненности вершины  $t$  для бинарных деревьев определяется как

$$\Delta i(t) = i(t) - P_L i(t_L) - P_R i(t_R),$$

где  $P_L$  и  $P_R$  – доли примеров подвыборки  $D_t$ , соответствующие левому и правому сыновьям ( $t_L$  и  $t_R$ ). Наилучшим расщеплением вершины  $t$  считается такое разбиение, которое максимизирует величину  $\Delta i(t)$  [7], то есть оптимальное расщепление необходимо выполняться таким образом, чтобы  $\Delta i(t) \rightarrow \max$ .

Определим величину  $\Delta i(t)$ , используя в качестве меры загрязненности вершины индекс Gini. Пусть  $L$  и  $R$  – размеры подвыборок, соответствующих сыновьям  $t_L$  и  $t_R$ ;  $l_j$  и  $r_j$  – число экземпляров класса  $\omega_j$  в сыновьях  $t_L$  и  $t_R$ . Тогда оптимальным расщеплением вершины  $t$  будем считать такое расщепление, что

$$\Delta i(t) = \frac{1}{L} \sum_{j=1}^c l_j^2 + \frac{1}{R} \sum_{j=1}^c r_j^2 \rightarrow \max.$$

### 2.3.4. Ансамбль деревьев решений

Ансамбль классификаторов представляет собой множество классификаторов, чьи решения комбинируются некоторым образом для получения окончательной классификации объектов. Обычно синтез решений отдельных классификаторов, составляющих ансамбль, осуществляется путем их голосования. Основным достоинством ансамбля перед отдельным классификатором является то, что при

соблюдении определенных условий, он обладает значительно более высокой точностью, а также устойчивостью к «зашумлению» обучающей выборки. Необходимым и достаточным условием высокой точности ансамбля классификаторов является то, что составляющие его отдельные классификаторы должны быть достаточно точны и различны, то есть должны совершать ошибки на разных примерах [7]. Для выполнения этого условия в случайных лесах применяются два метода: манипулирование примерами обучающей выборки и инъекция случайности в индуктивный алгоритм.

Для манипулирования примерами используется метод, называемый баггингом. Он основан на формировании обучающей выборки для каждого дерева ансамбля с помощью бутстрепа и использовании невзвешенного голосования при агрегировании решений отдельных классификаторов [8]. Бутстреп-выборкой называется случайная выборка с возвращением из исходной обучающей выборки.

Инъекция случайности в алгоритм построения случайных лесов происходит при построении каждого дерева решений на стадии расщепления вершин, где из всего множества признаков обучающей выборки случайно выбирается только их фиксированное число. Таким образом, на каждой итерации построения деревьев решений фактически используется метод случайных подпространств.

### **2.3.5. Алгоритм построения случайного леса, алгоритм классификации**

Пусть  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$  – обучающая выборка,  $B$  – число деревьев в ансамбле,  $m$  – число случайно отбираемых признаков на этапе расщепления вершин. Тогда алгоритм построения случайного леса может быть представлен следующим образом.

Для каждого дерева  $T_i$  ( $i = 1, 2, \dots, B$ ) выполнить:

1. Сформировать бутстреп выборку  $S_i$  размера  $l$  на основе исходной обучающей выборки  $D$ ;
2. По бутстреп выборке  $S_i$  построить неусеченное дерево решений  $T_i$  с минимальным количеством наблюдений в терминальных вершинах равным  $N_{min}$ , рекурсивно следуя следующему подалгоритму:
  - а) из исходного набора  $n$  признаков случайно выбрать  $m$  признаков;
  - б) из  $m$  признаков выбрать признак, который обеспечивает наилучшее расщепление;

- с) расщепить выборку, соответствующую обрабатываемой вершине, на две подвыборки и создать двух сыновей вершины.

Классификация примера  $\mathbf{x} \notin D$  осуществляется следующим образом. Пусть  $\omega_i^*(\mathbf{x}) \in \{\omega_j\}_{j=1}^C$  – класс, предсказанный деревом решений  $T_i$  для вектора  $\mathbf{x}$ . Тогда класс  $\omega_B^*(\mathbf{x})$ , предсказанный случайным лесом для вектора  $\mathbf{x}$ , определяется как мажорантный класс во множестве  $\{\omega_i^*(\mathbf{x})\}_{i=1}^B$  [7].

### 2.3.5. Оценка вероятности ошибочной классификации

Одно из достоинств случайных лесов состоит в том, что для оценки вероятности ошибочной классификации нет необходимости использовать кросс-проверку или тестовую выборку [7]. Оценка вероятности ошибочной классификации случайного леса осуществляется методом «Out-Of-Bag» (OOB), состоящем в следующем.

Известно, что каждая бутстреп выборка не содержит примерно 37% примеров исходной обучающей выборки [7]. Будем классифицировать некоторый вектор  $\mathbf{x} \in D$ , используя только те деревья случайного леса, которые строились по бутстреп выборкам, не содержащим  $\mathbf{x}$ . Частота ошибочно классифицированных примеров обучающей выборки при таком способе классификации представляет собой оценку вероятности ошибочной классификации случайного леса. Практика применения оценки OOB показала, что в случае, если количество деревьев в ансамбле достаточно велико, то эта оценка обладает высокой точностью [7].

Обычно оценку вероятности ошибочной классификации, найденную методом OOB, обозначают как  $OBE$  (Out-Of-Bag-Error). Также часто используют оценку OOB для отдельных деревьев ансамбля, обозначая ее  $OBE_T$ . Данная оценка определяется, как частота ошибочной классификации векторов  $\mathbf{x} \in D$ , не входящих в бутстреп выборку, на основе которой строилось дерево  $T$ . При этом в классификации участвует не весь ансамбль деревьев, а только само дерево  $T$ .

## 2.4. Обучение случайного леса в режиме реального времени

Обычно построение случайного леса осуществляется до начала его применения в качестве классификатора, то есть фазы обучения и тестирования разделены. Однако на практике обучающие данные часто не могут быть даны заранее в полном объеме, а



поступают последовательно. Так, при решении задачи трекинга, необходимо обучать классификатор и получать его прогнозы в режиме реального времени [14].

Случайный лес представляет собой ансамбль деревьев решений, объединённых посредством баггинга, из-за чего для реализации on-line обучения необходимо осуществлять как баггинг, так и построение самих деревьев решений в режиме реального времени.

#### 2.4.1. On-line баггинг

Допустим, что обучающая выборка  $D$  предоставлена заранее и имеет размер  $N$ . Тогда для построения каждого дерева случайного леса необходимо сформировать бутстреп выборку  $S$  размера  $N$ . Пусть случайная величина  $K_j$  ( $j = 1, \dots, N$ ) принимает значение 1, если пример  $(\mathbf{x}, y) \in D$  попал в выборку  $S$  на позицию  $j$ , и значение 0 в противном случае. Соответствующие вероятности равны:

$$p = P\{K_j = 1\} = \frac{1}{N}, \quad q = P\{K_j = 0\} = 1 - \frac{1}{N}.$$

Тогда  $\{K_j\}_{j=1}^N$  – конечная последовательность независимых случайных величин, имеющих одинаковое распределение Бернулли с параметрами  $p$  и  $q$ . Случайная величина  $K = K_1 + K_2 + \dots + K_N$ , являющаяся числом попаданий примера  $(\mathbf{x}, y)$  в выборку  $S$ , имеет биномиальное распределение с параметрами  $N$  и  $p$ :

$$K \sim Bin(N, p), \quad p = \frac{1}{N}.$$

Вероятность того, что пример  $(\mathbf{x}, y)$  попадет в выборку  $S$  ровно  $k$  раз можно найти следующим образом:

$$P\{K = k\} = \binom{N}{k} \cdot \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{N-k}, \quad \binom{N}{k} = \frac{N!}{(N-k)! k!}.$$

В случае последовательно поступающих данных, мы не можем определить размер обучающей выборки  $N$  заранее. Будем считать, что  $N \rightarrow \infty$ , тогда  $p \rightarrow 0$  и биномиальное распределение можно аппроксимировать распределением Пуассона с параметром  $\lambda = Np = 1$ :

$$K \sim Poisson(1).$$

Таким образом, при баггинге в on-line режиме, последовательно поступающие данные возможно моделировать при помощи распределения Пуассона [13]. Для этого необходимо обучать каждое дерево ансамбля на каждом поступающем примере  $K \in \xi$

раз. Здесь  $\xi = \{\xi_i\}_{i=1}^n$  – последовательность случайных величин, принимающих неотрицательные целочисленные значения, такая что для всякого  $k$  при  $n \rightarrow \infty$  выполняется условие

$$P\{\xi_i = k\} \sim Poisson(1).$$

То есть  $\xi$  – это последовательность распределений асимптотически равных распределению  $Poisson(1)$ .

#### 2.4.2. Построение деревьев решений в on-line режиме

Главной задачей при построении дерева решений в on-line режиме является выбор оптимальных моментов расщепления вершин. Правильность этого выбора крайне важна, так как надежность статистик, накапливаемых в последующих вершинах-сыновьях, напрямую зависит от выбранного момента расщепления вершины-отца. Для обеспечения правильности выбора момента расщепления необходимо соблюдать два условия [14]:

1. число примеров, прошедших через вершину должно быть достаточным, чтобы в ней находилась надежная статистика;
2. расщепление вершины должно быть произведено так, чтобы достигалась цель верной классификации.

Соблюдение первого условия обеспечивается введением в алгоритм расщепления вершины величины  $\alpha_{split}$ , определяющей минимальное число примеров, которые должны пройти через вершину до ее расщепления. Для выполнения второго условия вводится величина  $\beta_{split}$ , определяющая минимальное значение уменьшения загрязненности вершины. Численные значения  $\alpha_{split}$  и  $\beta_{split}$  подбираются так, чтобы минимизировать *OOBE*.

От дерева, построение которого производится в on-line режиме, также требуется возможность осуществления классификации сразу же после расщепления вершины, то есть без получения новых данных для накопления статистик [14].

Пусть  $t$  – вершина, удовлетворяющая условиям расщепления. При прохождении примеров через эту вершину, были накоплены статистики меток классов  $p_{t_L}(x_i)$  и  $p_{t_R}(x_i)$  левых и правых возможных сыновей вершины, разделенных по признакам  $x_i \in \mathbf{x}$ . Тогда при расщеплении вершины по признаку  $x^*$ , обеспечивающему наилучшее расщепление, происходит создание сыновей  $t_L$  и  $t_R$ ,

которым передаются значения статистик  $p_{t_L}(x^*)$  и  $p_{t_R}(x^*)$ . Таким образом, данный метод обеспечивает наличие накопленных статистик у создаваемых вершин-сыновей, что позволяет осуществлять классификацию без необходимости получения новых данных для накопления статистик.

### 2.4.3. Алгоритм on-line обучения случайного леса

Пусть  $(\mathbf{x}, y)$  – обучающий пример из последовательности поступающих данных,  $B$  – число деревьев в ансамбле. Тогда алгоритм on-line обучения случайного леса может быть представлен следующим образом.

Для каждого дерева  $T_i$  ( $i = 1, 2, \dots, B$ ) выполнить:

1.  $K \sim \text{Poisson}(1)$  раз обучить дерево  $T_i$  на примере  $(\mathbf{x}, y)$ ;
2. Для всех нерасщепленных вершин  $t$  дерева  $T_i$ , удовлетворяющих условиям расщепления, следовать следующему подалгоритму:
  - а) из исходного набора  $n$  признаков вектора  $\mathbf{x}$  случайно выбрать  $m$  признаков;
  - б) из  $m$  признаков выбрать признак  $x^*$ , который обеспечивает наилучшее расщепление;
  - с) создать двух сыновей вершины, передав им значения статистик  $p_{t_L}(x^*)$  и  $p_{t_R}(x^*)$ .

### 2.4.4. Адаптивное исключение деревьев из ансамбля

Часто при on-line обучении распределение обучающих примеров  $P(\mathbf{x})$  меняется с течением времени. Следовательно, необходимо иметь механизм, способный «отучивать» случайный лес от старого распределения. Этого можно достичь, исключая из ансамбля деревья, неспособные верно классифицировать примеры с изменившимся распределением [14].

Стоит отметить, что если ансамбль состоит из большого числа деревьев, то исключение из него одного дерева не приносит вреда точности классификации ансамбля в целом. Таким образом, непрерывное исключение деревьев, неспособных верно классифицировать примеры, гарантирует адаптацию ансамбля к изменяющемуся с течением времени распределению [14].

Рассмотрим обучающий пример  $(\mathbf{x}, y)$ . Моделирование последовательно поступающих данных при помощи пуассоновского распределения ведет к тому, что

некоторые деревья не участвуют в обучении на этом примере. Следовательно, на основе примера  $(\mathbf{x}, y)$  мы можем оценить как  $OOBE$  всего леса, так и  $OOBE_T$  дерева, не участвующего в обучении на этом примере.

Очевидно, что деревья, неспособные верно классифицировать примеры, имеют большие значения  $OOBE_T$ . Следовательно, на основе этих значений можно производить случайное исключение деревьев из ансамбля, причем вероятность исключения  $i$ -го дерева должна зависеть от  $OOBE_T(i)$  [14].

## **Глава 3. Применение методов распознавания образов к задаче слежения**

### **3.1. Задача долгосрочного слежения за объектами**

Задачей долгосрочного слежения называется такая задача, которая предполагает слежение за объектами в течение большого, возможно, бесконечного промежутка времени. При этом трекер, осуществляющий долгосрочное слежение должен корректно продолжать работу при сбое. Сбои работы трекера часто происходят в случае изменения окружающей обстановки сцены, перекрытия объекта слежения другими объектами или же в случае исчезновения отслеживаемого объекта из области наблюдения [12].

Базовые методы решения задачи слежения, описанные в главе §1, основаны на вычислении изменения координат объекта от кадра к кадру. Данные методы не позволяют решать задачу долгосрочного слежения непосредственно, так как не предполагают корректного поведения трекера в условиях сбоя.

Очевидно, что решение задачи долгосрочного слежения требует повторного нахождения отслеживаемого объекта в случае сбоя. С этой целью в алгоритм слежения часто интегрируют детектор, основанный на предварительно обученном классификаторе. Обучение таких детекторов производится на заранее подготовленных размеченных выборках, что не позволяет использовать их в том случае, если отслеживаемый объект не известен заранее. Другим недостатком таких методов является то, что фазы обучения и тестирования строго разделены, следовательно, изменения окружающей обстановки, не представленные в обучающей выборке, не могут стать частью модели [12]. Решением проблем, связанных с

предварительным обучением детектора, является использование классификатора, способного обучаться в режиме реального времени, описанного в §2.4.

Таким образом, целью данной работы является создание метода долгосрочного слежения за объектами, предполагающего наличие минимальной начальной информации. Под минимальной начальной информацией понимается то, что объект отслеживания не известен заранее и единственной информацией о нем являются его координаты, заданные пользователем на некотором кадре видеопоследовательности.

### 3.2. Предлагаемый метод решения задачи

В основе алгоритма долгосрочного слежения, разрабатываемого в рамках данной работы, лежит использование двух компонент: трекера, основанного на методе вычитания фона, и детектора, основанного на случайном лесе решений. В §1.5 и §1.6 была описана модификация метода вычитания фона, позволяющая компенсировать движение камеры и осуществить отсеечение нежелательных шумов. В §2.4 был представлен алгоритм построения случайного леса, обеспечивающий обучение в режиме реального времени.

Суть предлагаемого алгоритма заключается в следующем. Трекер отслеживает положение цели от кадра к кадру, при этом выбор объекта для отслеживания осуществляется пользователем на некотором кадре видеопоследовательности. В случае если трекер теряет цель, детектор начинает ее поиск, и в случае его удачного завершения, сообщает трекеру новые координаты отслеживаемого объекта. Таким образом, использование двух компонент делает метод устойчивым к сбоям.

### 3.3. Алгоритм работы детектора

Алгоритм работы детектора можно условно разделить на два этапа: обучение классификатора и поиск отслеживаемого объекта с его помощью.

Обучение классификатора осуществляется на последовательно поступающих примерах, формируемых трекером, представляющих собой последовательности вида  $S_T = \{(\mathbf{x}_j, y_j)\}_{j=1}^l$ , где  $l$  – число выделенных на кадре областей,  $y_j \in \{0,1\}$ ,  $T$  – номер дерева в ансамбле. Последовательность  $S_T$  можно представить следующим образом:

$$S_T = \{(\mathbf{x}_1, 1), (\mathbf{x}_2, 0), \dots, (\mathbf{x}_l, 0)\},$$

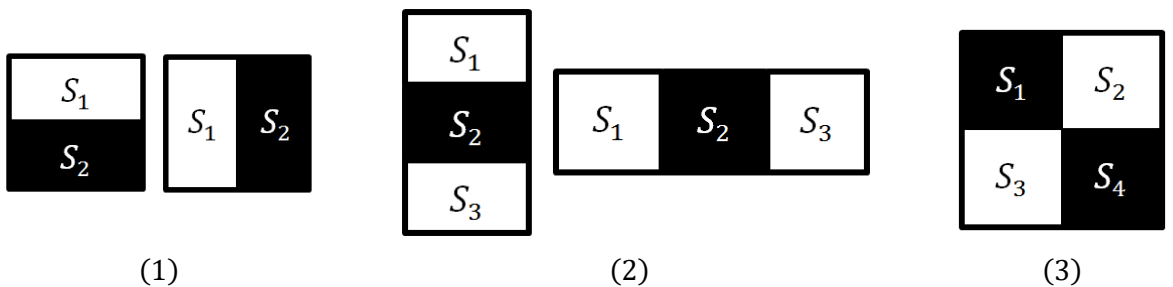
где  $(\mathbf{x}_1, 1)$  – пример соответствующий области цели,  $\{(\mathbf{x}_j, 0)\}_{j=2}^l$  – примеры, соответствующие областям, не являющимся областью отслеживаемого объекта.

Алгоритм построения примеров последовательности  $S_T$  следующий. Трекер выделяет на каждом кадре  $I$  одну область – прямоугольную область  $I_1$ , соответствующую цели. Изображение  $I \setminus I_1 = \{(x, y) \in I | (x, y) \notin I_1\}$  разбивается на  $l - 1$  непересекающихся прямоугольную область  $\{I_j\}_{j=2}^l$  произвольного размера. В областях  $\{I_j\}_{j=1}^l$  вычисляются значения векторов признаков  $\{\mathbf{x}_j\}_{j=1}^l$ , при этом значение метки класса для вектора  $\mathbf{x}_1$ , вычисленного в области  $I_1$ , принимается равным единице, а значение остальных меток классов – нулю.

В случае сбоя трекера, детектор осуществляет поиск отслеживаемого объекта следующим образом. На каждом кадре трекер, основанный на методе вычитания фона, осуществляет сегментацию изображения на области  $\{I_j\}_{j=1}^k$ , где  $k$  – число сегментов. Для каждого дерева ансамбля  $T$  вычисляется последовательность векторов признаков  $F_T = \{\mathbf{x}_j\}_{j=1}^k$ , где вектор  $\mathbf{x}_j$  соответствует области  $I_j$ . Классификатором принимается решение, какой из векторов признаков  $\mathbf{x}_m \in F_T$  соответствует отслеживаемому объекту, после чего координаты области  $I_m \in \{I_j\}_{j=1}^k$  выбираются в качестве новых координат цели, и процесс слежения трекером за целью запускается заново.

### 3.4. Признаки цифрового изображения

В качестве признаков, значения которых вычисляются в областях изображения  $I_j$ , были выбраны признаки Хаара. Признаки Хаара – это признаки цифрового изображения, используемые в распознавании образов. Своим названием они обязаны сходством с вейвлетами Хаара и были впервые описаны в работе [16]. Признак Хаара состоит из смежных прямоугольных областей, позиционируемых на изображении. Основываясь на числе смежных областей, выделяют несколько типов признаков Хаара: двух-прямоугольные, трех-прямоугольные и четырех-прямоугольные.



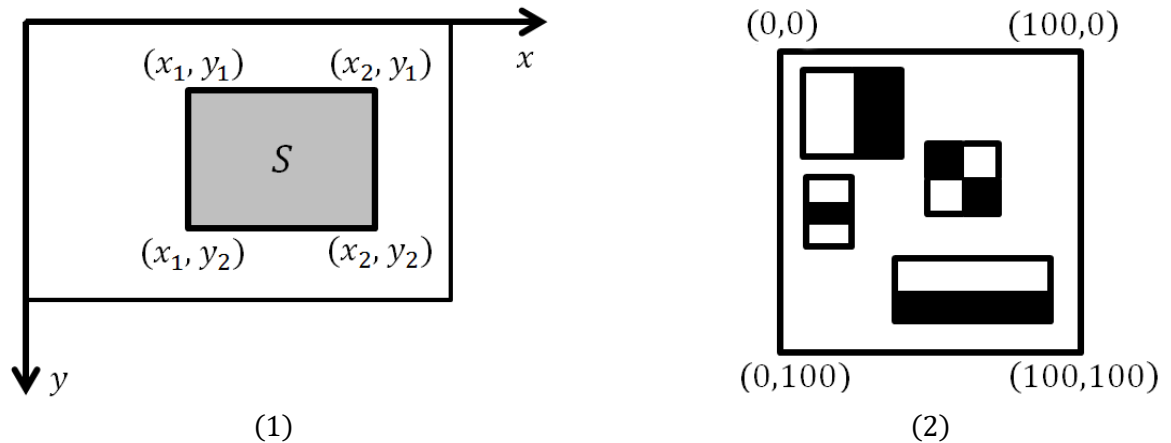
**Рисунок 4.** Типы признаков Хаара: двух-прямоугольные (1), трех-прямоугольные (2) и четырех-прямоугольные (3).

При вычислении значений смежных прямоугольных областей используется интегральное представление изображения, то есть матрица  $L$ , совпадающая по размерам с исходным изображением, каждый элемент которой представляет собой сумму интенсивностей всех пикселей, находящихся левее и выше данного элемента:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j).$$

Интегральную сумму произвольной прямоугольной области  $S(x_1, x_2, y_1, y_2)$  можно вычислить, используя интегральное представление изображения следующим образом:

$$L(S) = L(x_2, y_2) - L(x_1 - 1, y_2) - L(x_2, y_1 - 1) + L(x_1 - 1, y_1 - 1).$$



**Рисунок 5.** Положение прямоугольной области  $S$  (1). Шаблон для вычисления значений векторов признаков (2).

Значение признаков Хаара первой группы вычисляется, как разность между интегральными суммами пикселей в двух смежных прямоугольных областях:  $L(S_1) - L(S_2)$ . Для второй группы значение вычисляется, как интегральная сумма двух внешних областей, вычитаемая из суммы центральной области:  $L(S_2) - L(S_1) - L(S_3)$ . Значение третьей группы признаков вычисляется, как разность между суммами диагональных пар областей:  $L(S_1) + L(S_4) - L(S_2) - L(S_3)$  [16].

Для каждого дерева  $T$  ансамбля при инициализации классификатора строится шаблон, по которому впоследствии будут вычисляться значения векторов признаков  $\mathbf{x}_j$ . Этот шаблон представляет собой область фиксированного размера ( $100 \times 100$  пикселей), на которую позиционируются  $n$  признаков Хаара, причем размер

признаков, их координаты и тип выбираются случайным образом, чтобы обеспечить различность строящихся деревьев решений.

Рассмотрим вектор признаков  $\mathbf{x}_j = \{x_j^s\}_{s=1}^n$  для дерева  $T$ , вычисляемый в области  $I_j$ . Координаты этого вектора представляют собой значения  $n$  признаков Хаара, вычисляемых в области  $I_j$  по масштабированному шаблону. Шаблон масштабируется так, чтоб совпадать размером с областью  $I_j$ , что обеспечивает неизменность значений признаков при изменении размеров области. Значение признака  $x_j^s \in \mathbf{x}_j$  принимается равным единице в случае, если разность интегральных сумм смежных прямоугольных областей неотрицательна, и нулю – в противном случае.

## **Глава 4. Результаты экспериментов**

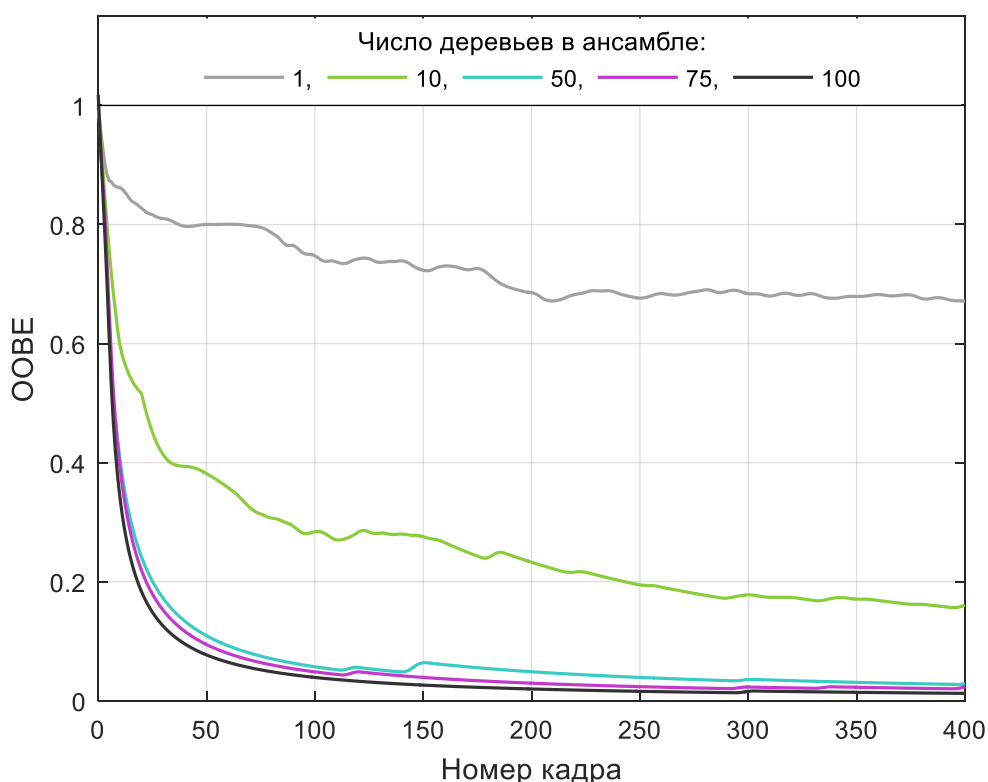
### **4.1. Тестовые данные для проведения экспериментов**

В качестве тестовых данных для проведения экспериментов, связанных с анализом характеристик рассматриваемых методов обнаружения и слежения, использовалась видеозапись движения радиоуправляемой модели машины в закрытом помещении. Данная видеозапись включала в себя временные промежутки с изменением окружающей обстановки сцены, перекрытием объекта слежения другими объектами и исчезновением отслеживаемого объекта из области наблюдения с последующим его возвращением.

### **4.2. Результаты экспериментов**

Важнейшими характеристиками любого метода обнаружения и слежения являются его точность и производительность. Вначале рассмотрим результаты обучения классификатора при относительно однородных обучающих примерах (кадрах). Под однородностью в данном контексте понимается то, что обучающие примеры принадлежат тем промежуткам видеопоследовательности, которые не включают в себя изменения окружающей обстановки сцены, перекрытия объекта слежения другими объектами или исчезновения отслеживаемого объекта из области наблюдения. То есть единственными изменениями, происходящими с отслеживаемым объектом, являются изменения его координат в пространстве, что также, вследствие перспективной проекции, влечет за собой изменения его размеров.

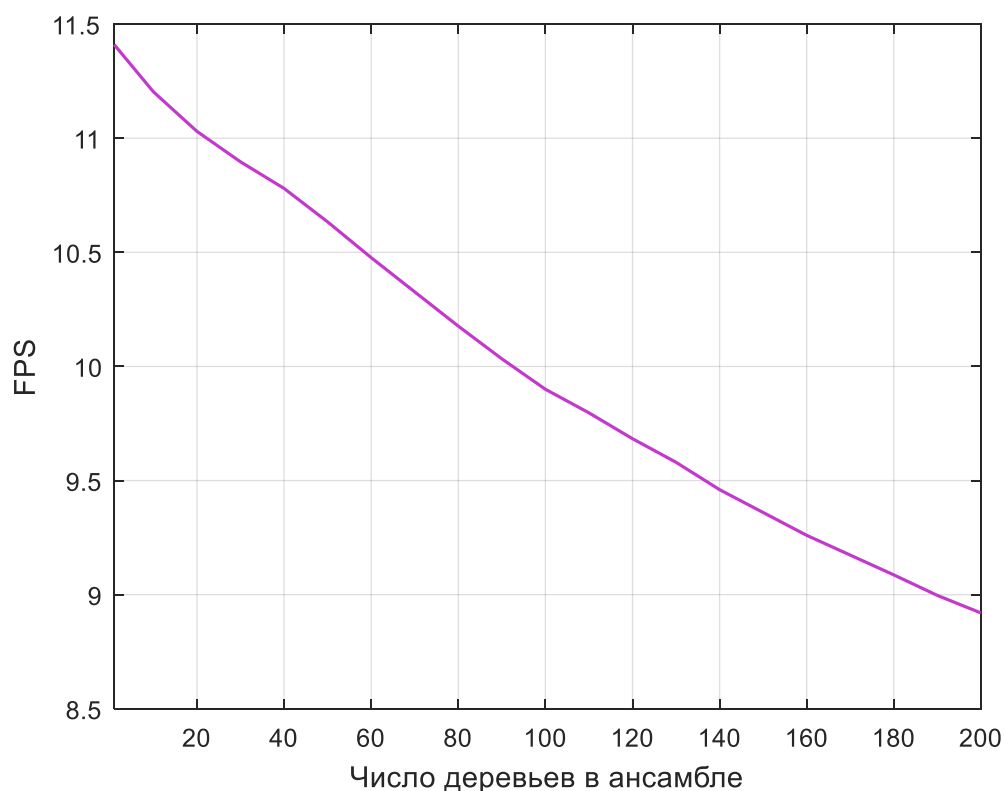




**Рисунок 6.** Зависимость вероятности ошибочной классификации (OOBE) от числа деревьев в ансамбле при обучении на однородных примерах.

Проанализировав рисунок 6, можно заключить, что обучение классификатора на однородных примерах приводит к монотонному уменьшению ошибки классификации от кадра к кадру. Также можно наблюдать, что вероятность ошибочной классификации обратно пропорциональна числу деревьев в ансамбле классификатора. Уменьшение вероятности ошибочной классификации с ростом числа деревьев в ансамбле происходит нелинейно. Так, для ансамблей, включающих в себя большое число деревьев (50,75,100), разница между их ошибками классификации с течением времени становится не столь заметной.

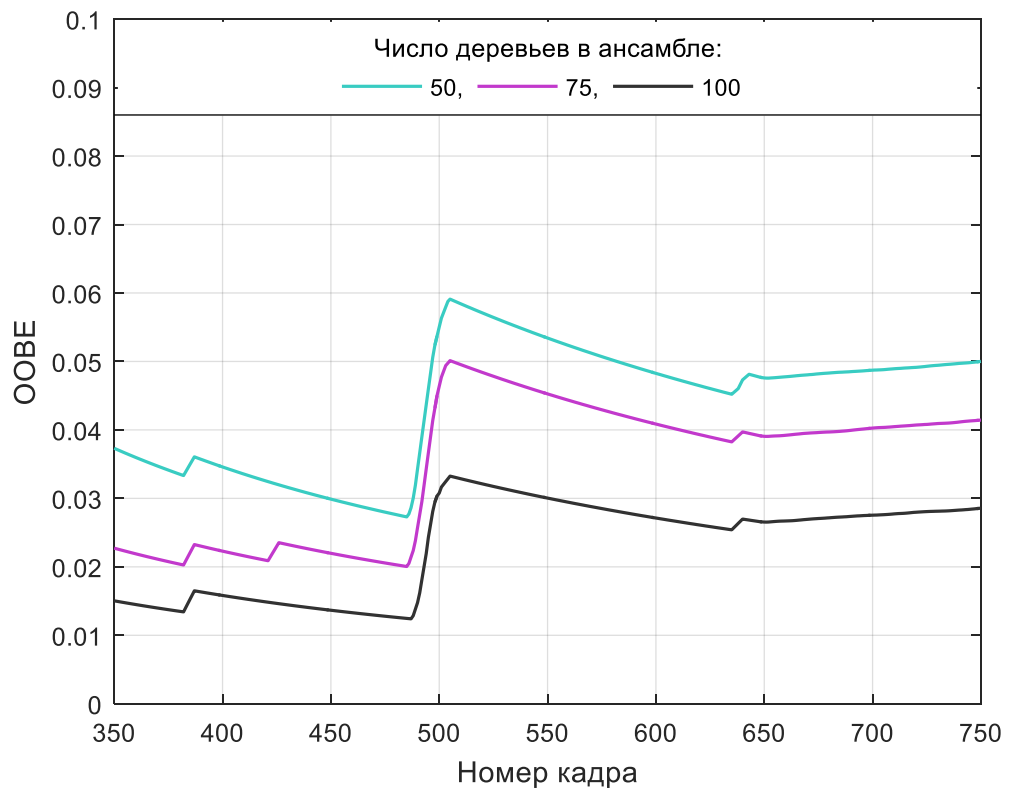
Для подбора оптимальных параметров классификатора проведем исследование зависимости производительности алгоритма классификации от числа деревьев, входящих в ансамбль.



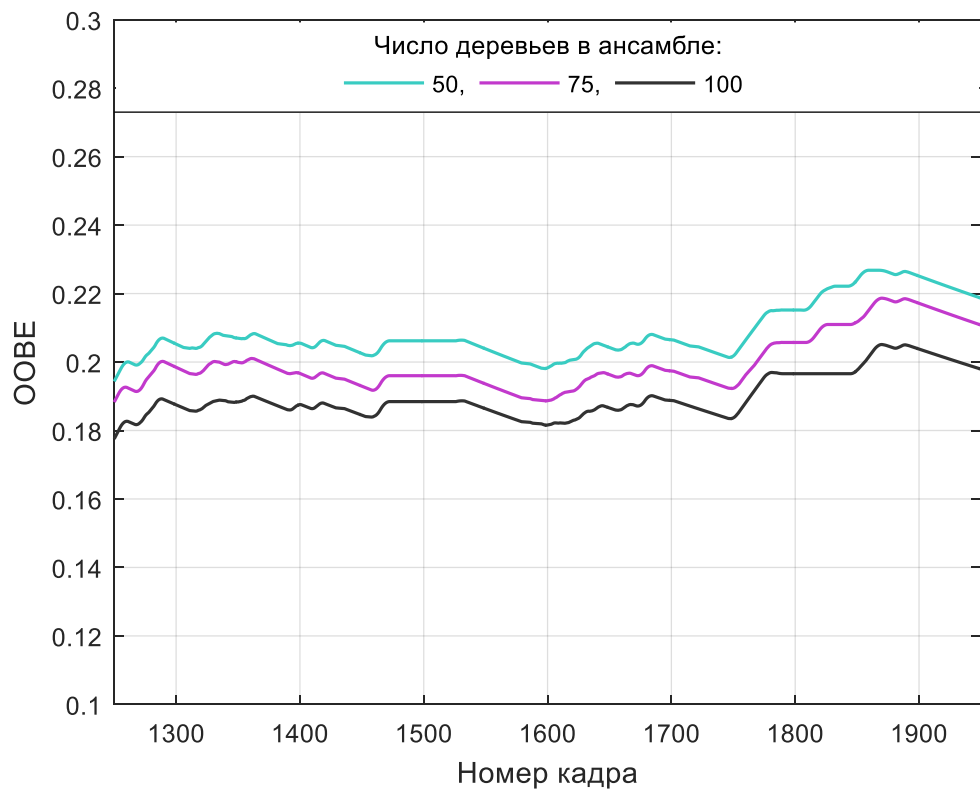
**Рисунок 7.** Зависимость производительности алгоритма классификации (число кадров, обрабатываемых в секунду) от числа деревьев в ансамбле.

Из рисунка 7 становится очевидным, что производительность алгоритма линейно зависит числа деревьев в ансамбле. Таким образом, при подборе параметров классификатора следует выбирать такое число деревьев в ансамбле, которое обеспечивает требуемую точность классификации при минимальном числе деревьев.

Ранее рассматривалось обучение классификатора на однородных примерах, что редко возможно при решении реальных задач обнаружения и слежения. Далее рассмотрим результаты обучения классификатора при изменении обстановки окружающей сцены.



**Рисунок 8.** Рост вероятности ошибочной классификации (OOBE) при изменении обстановки окружающей сцены, вызванной изменением освещенности.

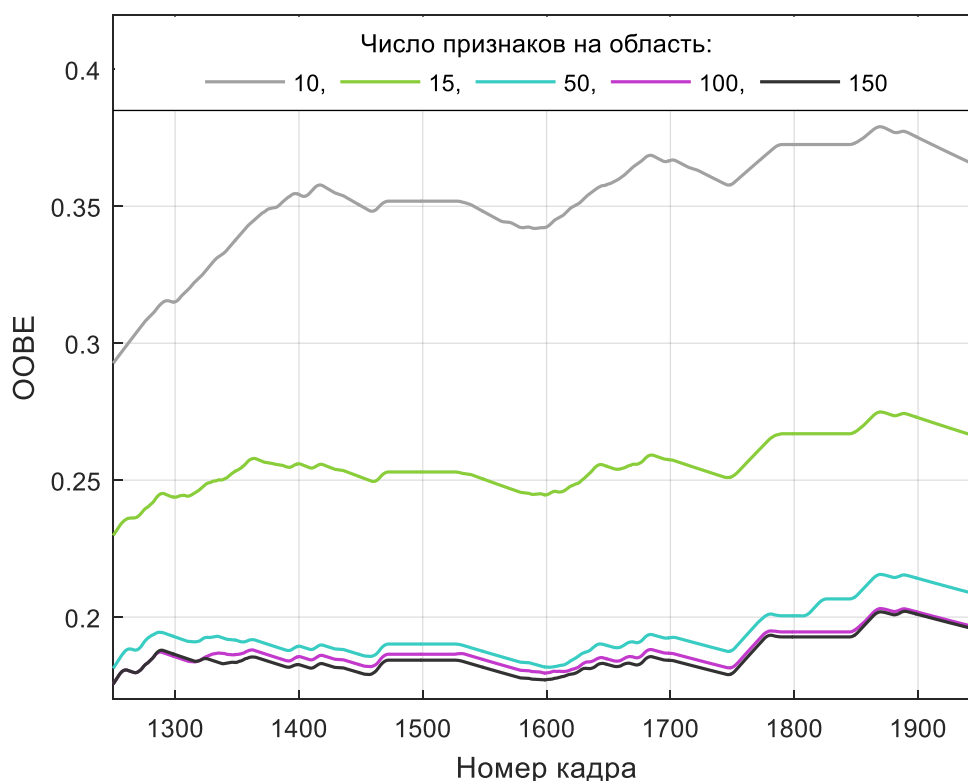


**Рисунок 9.** Изменение вероятности ошибочной классификации (OOBE) при длительном промежутке работы классификатора.

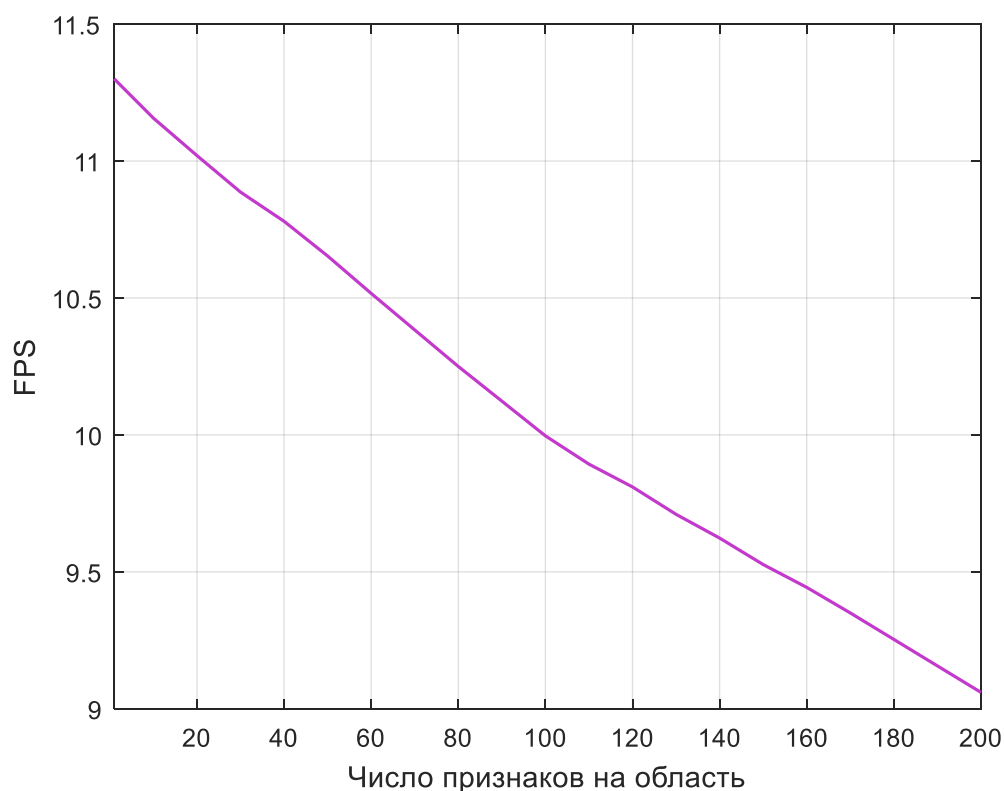
Из рисунка 8 видно, что при изменении обстановки окружающей сцены, происходит рост вероятности ошибочной классификации. Стоит отметить, что

полученная вероятность ошибки убывает с течением времени, так как классификатор начинает обучаться на примерах с новой обстановкой сцены. Также можно наблюдать, что классификатор не способен обеспечить понижение ошибки до начального уровня, так как для этого ему необходимо «отучить» информацию о старой обстановке сцены. Метод решения данной проблемы был предложен в §2.4.3 и будет рассмотрен далее. Проанализировав рисунок 9, можно заключить, что при длительном промежутке работы классификатора, вероятность его ошибки достигает некоторого предельного значения, которое сохраняется при изменении окружающей обстановки сцены и параметров отслеживаемого объекта, хотя иногда и испытывает небольшие скачки.

Еще одним ключевым параметром классификатора является число признаков, вычисляемых при его обучении в каждой из выделенных областей  $I_j$  (§3.3, §3.4). Для оптимального выбора этого параметра необходимо произвести как анализ ошибки классификации в зависимости от числа признаков, так и анализ производительности алгоритма классификации.



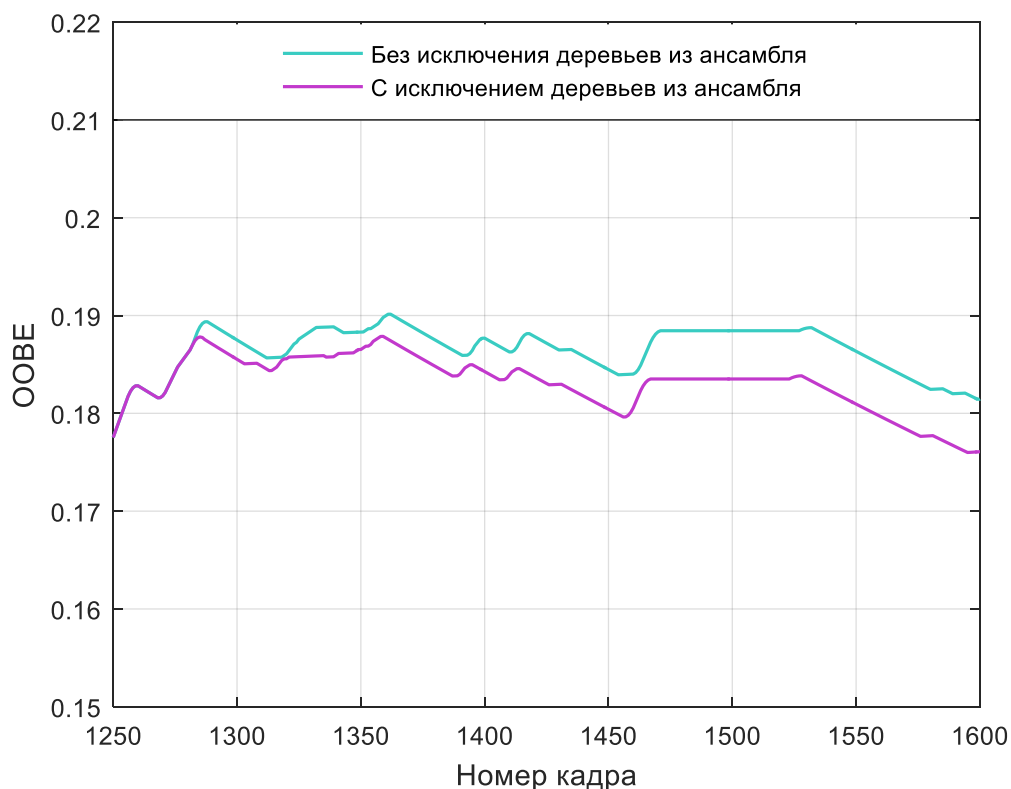
**Рисунок 10.** Зависимость вероятности ошибочной классификации (OOBE) от числа признаков, вычисляемых в каждой из выделенных областей.



**Рисунок 11.** Зависимость производительности алгоритма классификации (число кадров, обрабатываемых в секунду) от числа признаков, вычисляемых в каждой из выделенных областей.

На рисунке 10 можно наблюдать, что вероятность ошибочной классификации обратно пропорциональна числу признаков на область. Также можно заметить, что эта зависимость не линейна, и для классификаторов с большим числом признаков (100,200) разница в вероятностях ошибочной классификации почти неразличима. На рисунке 11 показано, что производительность алгоритма линейно зависима от числа признаков, поэтому при подборе параметров классификатора, число признаков на область необходимо выбирать так, чтобы обеспечить требуемую ошибку, используя при этом минимальное число признаков.

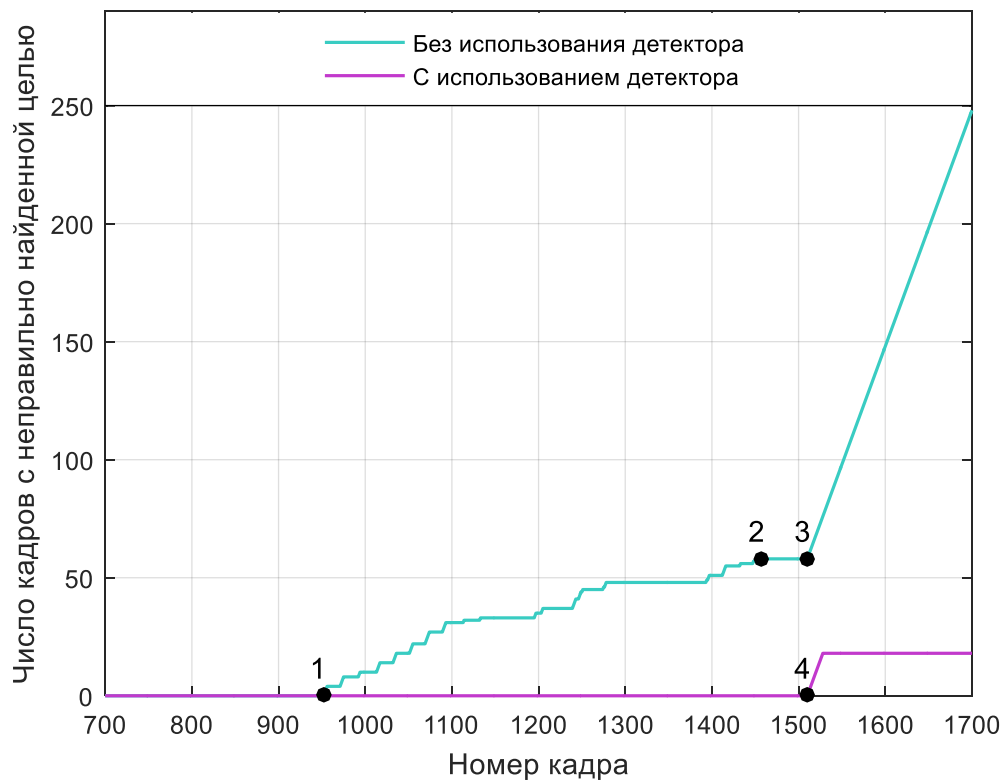
Далее рассмотрим результаты применения метода адаптивного исключения деревьев из ансамбля, предложенного в §2.4.3.



**Рисунок 12.** Изменение вероятности ошибочной классификации (OOBE) при использовании метода адаптивного исключения деревьев из ансамбля.

Из рисунка 12 видно, что использование метода адаптивного исключения деревьев из ансамбля, показало довольно высокие результаты. Предложенный метод позволяет адаптировать классификатор к изменениям примеров. Так, на графике можно наблюдать, что метод позволяет уменьшить величину скачков, вызванных изменениями обучающих примеров. Таким образом, данный метод позволяет снизить вероятность ошибочной классификации, адаптируя на протяжении времени ансамбль деревьев к изменениям примеров.

В заключение, проведем сравнительный анализ предложенного в §3 метода решения задачи долгосрочного слежения с классическим методом слежения без детектора, описанного в §1.3.3.



**Рисунок 13.** Сравнение результатов работы алгоритма: без использования детектора, с использованием детектора.

Из рисунка 13 видно, что предложенный в §3 метод решения задачи долгосрочного слежения доказал свою устойчивость к изменениям окружающей обстановки сцены, перекрытию объекта слежения другими объектами и исчезновению отслеживаемого объекта из области наблюдения. Рост ошибки классического метода решения задачи слежения (§1.3.3) на промежутке между точками 1 и 2 вызван неспособностью метода правильно определять координаты цели при изменении освещенности сцены и перекрытии объекта слежения другими объектами. Бесконечный рост ошибки, начинающийся с точки 3, вызван тем, что данный метод не способен находить цель заново в случае ее исчезновения из области наблюдения. Рост ошибки предлагаемого метода долгосрочного слежения (точка 4) вызван тем, что детектор не способен моментально находить объект, ранее исчезнувший из области наблюдения, из-за сильного изменения его размеров.

## Заключение

В рамках данной работы был проведен анализ эффективности методов обнаружения и слежения за объектами видеопоследовательности, а также был предложен метод, способный решать задачу долгосрочного слежения, предполагающего наличие минимальной начальной информации. Был проведен обзор существующих подходов к решению задачи обнаружения и слежения, а также обзор существующих методов решения задачи распознавания образов. Предложены модификации метода обнаружения и слежения, основанного на вычитании фонового изображения, способные улучшить результаты его работы. Проведена оценка эффективности применения метода распознавания образов к решению задачи обнаружения и слежения. Проанализированы характеристики предлагаемого метода долгосрочного слежения, а также выявлены его достоинства и недостатки.

Хотя предложенный метод и является законченным методом решения задачи долгосрочного слежения, показавшим высокие результаты и относительно низкую вычислительную сложность, существует множество путей его дальнейшего развития:

1. Улучшение качества исходного изображения путем предварительной фильтрации: подавление шумов, восстановление утраченных участков изображения, подчеркивания его важных деталей.
2. Использование модели фона, основанной на комбинации гауссианов [15].
3. Улучшение алгоритма сегментации: выделение точного контура цели, удаление из области цели теней и точек вызванных распространением световых лучей.
4. Компенсация перспективных преобразований при движении датчика изображений.
5. Использование большего числа траекторных признаков при слежении за объектом, например, средней скорости и ускорения.
6. Разбиение алгоритма выделения цели на две составные части (рост и усечение области) для более точного определения координат [12].
7. Использование сверточных нейронных сетей в качестве классификатора при решении задачи распознавания образов.



## Список использованной литературы

- [1] Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности: Справочное издание. – М.: Финансы и статистика, 1989. – 607 с.
- [2] Алпатов Б.А., Бабаян П.В., Балашов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. – М.: Радиотехника, 2008. – 176 с.
- [3] Журавлев Ю.И., Рязанов В.В., Сенько О.В. Распознавание. Математические методы. Программная система. Практические применения. – М.: Фазис, 2005. – 176 с.
- [4] Конушин А. Слежение за точечными особенностями сцены // Компьютерная графика и мультимедиа. 2003. №1(5).
- [5] Мерков А.Б. Распознавание образов: Введение в методы статистического обучения. – М.: Едиториал УРСС, 2011. – 254 с.
- [6] Олейник Т.А. Основы дискретной математики: теория и практика: уч. пособие. – М.: МИЭТ, 2010. – 252 с.
- [7] Чистяков С.П. Случайные леса: обзор // Труды Карельского научного центра РАН. №1. 2013. С. 117-136.
- [8] Breiman L. Bagging Predictors // Machine Learning. 1996. Vol. 24. P. 123-140.
- [9] Breiman L., Friedman R., Olshen R., Stone C. Classification and Regression Trees. – Belmont, California: Wadsworth International, 1984. – 342 p.
- [10] Fleet D., Weiss Y. Optical Flow Estimation // Handbook of Mathematical Models in Computer Vision. Springer US, 2006. P. 237-257.
- [11] Harris C., Stephen M. A combined corner and edge detector // Proceedings of the 4th Alvey Vision Conference, 1988. P. 147-151.
- [12] Kalal Z., Matas J., Mikolajczyk K. Online learning of robust object detectors during unstable tracking // 3rd On-line Learning for Computer Vision Workshop, 2009. P. 1417-1424.
- [13] Oza N., Russell S. Online bagging and boosting // Proceedings Artificial Intelligence and Statistics, 2001. P. 105-112.
- [14] Saffari A., Leistner C., Santner J., Godec M., Bischof H. On-line Random Forests // 3rd IEEE ICCV Workshop on On-line Computer Vision, 2009. P. 112-127.

- [15] Stauffer C., Grimson W.E.L. Adaptive background mixture models for real-time tracking // Computer Vision and Pattern Recognition. IEEE Computer Society Conference. 1999. Vol. 2. P. 246-252.
- [16] Viola P., Jones M. Robust Real-time Object Detection // International Journal of Computer Vision. 2004. Vol. 57. № 2. P. 137-154.