

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,

доцент, кандидат физико-математических наук \_\_\_\_\_ 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. ....	18
2.1. Распознавания образов. ....	18
2.2. Виды классификаторов. ....	18
2.3. Случайные леса. ....	18
2.4. Обучение случайного леса в режиме реального времени. ....	23
Глава 3. Применение методов распознавания образов к задаче слежения. ....	27
3.1. Задача долгосрочного слежения за объектами. ....	27
3.2. Предлагаемый метод решения задачи. ....	28
3.3. Алгоритм работы детектора. ....	29
3.4. Признаки цифрового изображения. ....	30
Глава 4. Результаты экспериментов. ....	31
4.1. Тестовые данные для проведения экспериментов. ....	31
4.2. Результаты экспериментов. ....	32
Заключение. ....	39
Список использованной литературы. ....	40

## **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.

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 и сопровождаемых 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 сопровождать, как правило, бывает очень скудна и, в лучшем случае, включает в себя лишь приблизительные размеры объектов, или их начальные координаты.

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

Целью работы является исследование и анализ эффективности методов обнаружения и слежения за объектами видеопоследовательности. В рамках данной работы будет предложен метод, способный решать задачу долгосрочного слежения, предполагающего наличие минимальной начальной информации.

Основные задачи, решение которых необходимо для достижения поставленной цели:

- производство обзора существующих подходов к решению задачи обнаружения и слежения за объектами видеопоследовательности, обзор существующих методов решения задачи распознавания образов;
- исследование эффективности метода обнаружения и слежения, основанного на вычитании фонового изображения, а также предложение способов улучшения результатов его работы;
- исследование эффективности метода распознавания образов с помощью классификатора, основанного на использовании случайного леса решений, способного обучаться в режиме реального времени.
- исследование эффективности применения методов распознавания образов к задаче обнаружения и слежения;
- разработка программного модуля для тестирования исследуемых алгоритмов, подбора их оптимальных параметров и оценки результатов их работы на основе тестовых видеопоследовательностей.

### **Содержание работы.**

В первой главе производится обзор существующих подходов к решению задачи обнаружения и слежения. Также предлагаются модификации метода, основанного на вычитании фонового изображения, способные улучшить результаты его работы.

Во второй главе производится обзор существующих подходов к решению задачи распознавания образов и классификации. Описывается классификатор, основанный на случайном лесе решений, а также его модификация, способная обучаться в режиме реального времени.

В третьей главе описывается применение методов распознавания образов к задаче обнаружения и слежения.

В четвертой главе представляются результаты работы алгоритма распознавания образов, и производится их сравнение при различных параметрах модели. Также представляются результаты работы метода долгосрочного слежения, предлагаемого 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 поставщиком 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.

Наибольшее распространение в системах слежения получили одноканальные датчики видимого диапазона. Как правило, эти датчики формируют на выходе цифровой 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 способа 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.

В данной работе будут использованы 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.

Отслеживание траектории может производиться двумя основными методами. Первый метод, называемый слежением, предполагает оценивание координат объекта на каждом кадре наблюдаемой видеопоследовательности без решения задачи управления перемещением датчика изображения. 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 решения 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 или самого объекта в пространстве.

Решение задачи обнаружения и слежения осуществляется в несколько этапов. Целью начального этапа является выделение принадлежащих объектам точек текущего кадра, 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 измерения координат и оценки параметров объектов. Последним этапом является слежение, которое состоит в оценке траектории движения объектов во времени, основываясь на анализе изменений координат сегментов на кадрах видеопоследовательности. При этом для

оценки траектории используются тракторные признаки, такие как близость сегментов на соседних кадрах видеопоследовательности и изменение соотношений их сторон.

### 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. Результат сегментации принято представлять в виде бинарного изображения  $BIN(x, y)$ , имеющего столько же элементов, сколько их содержит исходное изображение. В ходе сегментации всем точкам бинарного изображения, принадлежащим одному классу, присваивается одно и то же значение яркости. Значения яркостей выбираются из множества  $\{0, 1\}$ , при этом значение 1 присваивается точкам, принадлежащим областям, в которых предположительно находится объект, а значению 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.

#### 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 в каждой точке изображения.

Существует множество подходов к определению порогов, например, использование гистограмм яркости или использование моментов изображения. Основная идея метода заключается в предположении, что изображение содержит известное число однородных по яркости классов точек, то есть таких классов, распределения вероятностей которых унимодальны. Кроме того, считается, что

граничные участки между замкнутыми областями занимают сравнительно небольшую площадь изображения. Поэтому на гистограмме им должны соответствовать межмодовые впадины, в которых и устанавливаются пороги сегментации.

Использование методов пороговой обработки обычно ограничено ситуациями, при которых выполняется предположение об унимодальности распределения яркостей объектов и фона, при этом гистограммы объекта и фона должны слабо перекрываться. Эти ограничения позволяют использовать методы пороговой обработки лишь в простейших случаях.

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

Другим популярным методом сегментации является метод, основанный на выделении границ объектов. Этот метод основывается 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 имеют большую величину градиента функции  $I(x, y)$ :

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

Для того чтобы выделить границы областей, вначале вычисляют модуль градиента изображения

$$|\nabla I(x, y)| = \sqrt{\left( \frac{\partial I(x, y)}{\partial x} \right)^2 + \left( \frac{\partial I(x, y)}{\partial y} \right)^2},$$

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.



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

Применение методов выделения границ не всегда дает хороший результат сразу, вследствие чего часто применяется дополнительная обработка контурного изображения, которая включает прослеживание и уточнение контуров, восстановление разрывов в контурах и удаление одиночных точек.

К сожалению, 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 заметить, что методы выделения границ показывают низкие результаты при работе с сильно зашумленными изображениями, что является следствием использования операций дискретного дифференцирования.

### 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 фонового изображения  $B$  и изображения средних абсолютных отклонений  $D$ .

Среднее фоновое изображение  $B$  формируется путем осреднения  $N$  последних кадров  $I_{t-N}, \dots, I_t$  ( $t = N + 1, N + 2, \dots$ ) видеопоследовательности. При этом обновление среднего фонового изображения новыми кадрами происходит только теми участками, где нет движущихся объектов. Пусть  $B_{t-1}$  – среднее фоновое изображение, сформированное к кадру  $I_{t-1}$ ;  $S[I, \hat{I}]$  – оператор позволяющий получить изображение,  $I$  без движущихся объектов:

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

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  $I_{t-N}, \dots, I_t$  от фоновых изображений  $B_{t-N}, \dots, B_t$ . 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 по следующей формуле:

$$D_j = (1 - \alpha) \cdot D_{j-1} + \alpha \cdot S[|I_j - B_j|, D_{j-1}], \quad \alpha = \frac{1}{N}, \quad D_1 = I_1$$

Для определения точек, принадлежащих движущемуся объекту, используется предположение о том, что для движущихся точек, абсолютное отклонение их значений 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  $|I_j(x, y) - B_{j-1}(x, y)| > k_1 \cdot D_{j-1}(x, y)$ , где  $k_1 \geq 1$ .

Бинарное изображение  $BIN_t(x, y)$  на кадре  $t$  вычисляется на основе полученной модели фона следующим образом:

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

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.

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

Выделение объектов на основе бинарного изображения 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 полученного 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 этого, датчик может изменять свое положение в пространстве из-за вибраций и небольших сдвигов. Очевидно, что при изменении пространственных координат датчика изображений, меняются и координаты объекта наблюдения на матрице датчика, из-за чего в процессе наблюдения возникают геометрические преобразования изображений, такие как смещение, изменение масштаба и поворот на некоторый угол.

Стоит отметить, что для решения задачи выделения подвижных объектов при движущемся датчике, очень важно как можно более точно оценить параметры геометрических преобразований, вызванных движением датчика.

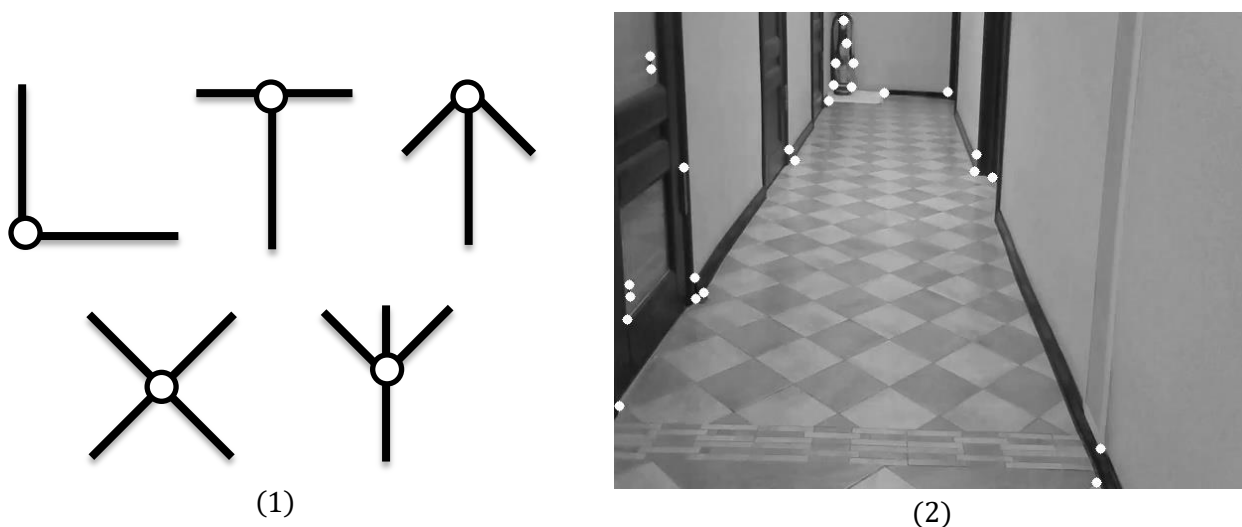
В рамках данной работы будет предложен метод компенсации смещений кадров видеопоследовательности, вызванных движением датчика изображений в плоскости, параллельной плоскости наблюдения. При этом в модели фона, используемой в методе сегментации, основанной на вычитании фонового изображения, необходимо произвести следующие модификации: формировать среднее фоновое изображение и изображение средних абсолютных отклонений с учетом взаимных смещений кадров видеопоследовательности. Следует отметить, что смещение изображений модели фона необходимо выполнять так, чтобы область

наблюдения на этих изображениях максимально соответствовала области наблюдения на текущем кадре.

Существует большое число методов оценки параметров геометрических преобразований: корреляционный, спектральный, оценка на основе опорных элементов. Корреляционные и спектральные методы оценки опираются на использование яркостной информации изображения. Оценка на основе опорных элементов, использованная в предлагаемом методе, состоит в поиске опорных объектов на каждом кадре видеопоследовательности, определении их координат, а также нахождении параметров геометрических преобразований.

### 1.5.1. Опорные элементы.

В качестве опорных элементов изображения в предлагаемом методе используются угловые особые точки, найденные на первом кадре видеопоследовательности. Особой точкой  $m$  или точечной особенностью изображения называется точка изображения, окрестность которой  $\Omega(m)$  однозначно отличима от любой другой окрестности точки  $n \in \Omega'(m)$ , где  $\Omega' > \Omega$ . 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 области вокруг них у градиента изображения преобладают два доминирующих направления, что позволяет относительно просто осуществлять их поиск.



**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 [3]. В нем используется окно  $W$  размера  $n \times n$  пикселей с центром в точке  $(x_c, y_c)$ , 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  $(x_c, y_c)$  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. 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 напрямую считать собственные значения является трудоёмкой задачей, Харрисом была предложена мера отклика:

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

где  $\det M$  – определитель матрицы  $M$ ,  $\text{trace} M$  – ее след,  $k \in [0.04; 0.06]$  – эмпирическая константа.

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 предложенном методе 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  $I_2(x + \Delta x, y + \Delta y)$ .

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. 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 такая величина  $(\Delta x, \Delta y)$ , которая 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.

### 1.5.3. Смещение кадра.

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

После оценки 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.



(1)



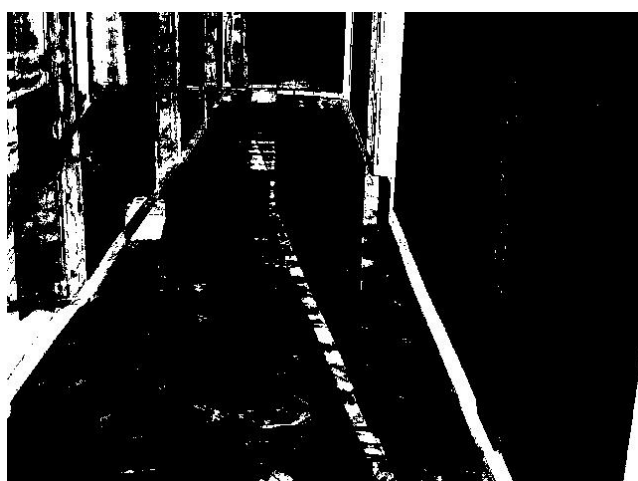
(2)



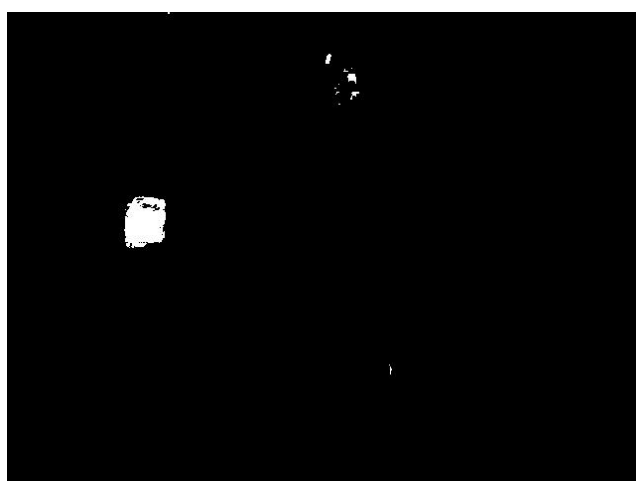
(3)



(4)



(5)



(6)

**Рисунок 2.** Результат учета движения камеры. Среднее фоновое изображение: без учета движения камеры (1), с учетом (2). Масштабированное по яркости изображение средних абсолютных отклонений: (3), (4). Бинарное изображение, 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 метода, основанного на использовании порога фоновой части для изображения  $|I_t - B_t|$ . 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. Кроме того, считается, что граничные участки между областями разных классов занимают сравнительно небольшую площадь изображения. Поэтому на гистограмме яркостей им должна соответствовать межмодовая впадина, в которой и устанавливается порог фоновой части изображения. Применение порога фоновой части может быть описано следующим выражением:

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

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

$$\omega = \min\{v \in [v_0; 255] \mid H(k + 1) \geq H(k)\},$$

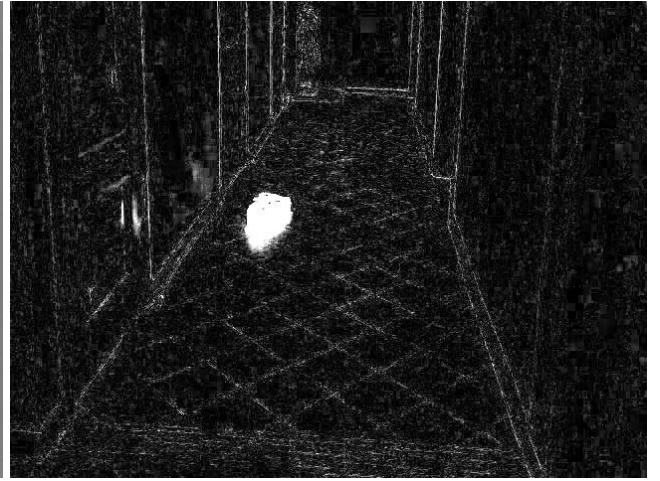
где  $H(i)$ ,  $i = 0, 1, \dots, 255$  – гистограмма изображения  $|I_t - B_t|$ ,

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

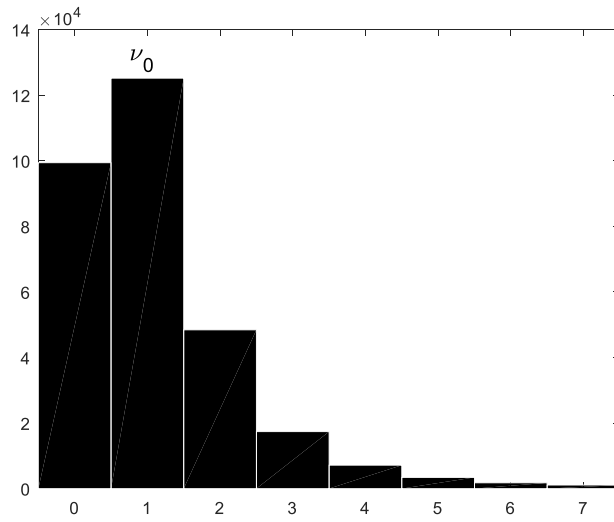




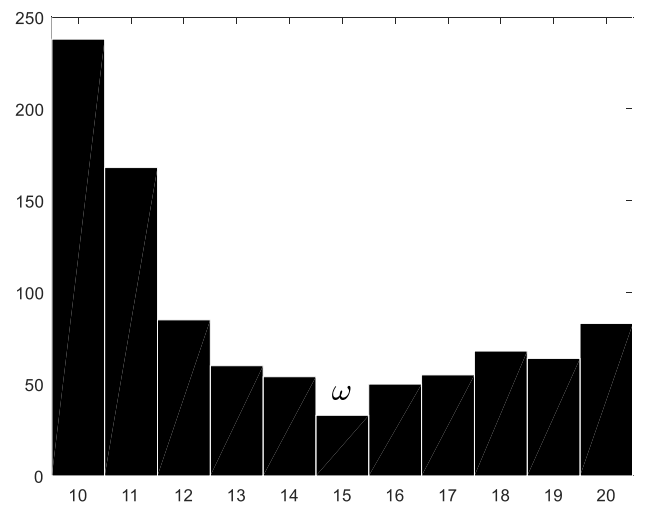
(1)



(2)



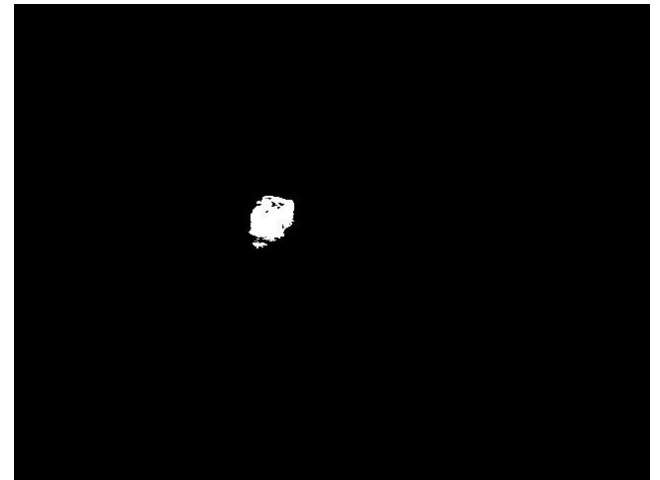
(3)



(4)



(5)



(6)

**Рисунок 3.** Результат применения порога фоновой части. Изображение  $I_t$ , полученное с датчика изображений (1). Масштабированное по яркости изображение  $|I_t - B_t|$  (2). Гистограмма изображения  $|I_t - B_t|$ : (3), (4). Бинарные изображения, полученные в результате сегментации: без применения порога (5), с применением (6).

## Глава 2. Распознавание образов и классификация.

### 2.1. Распознавания образов.

Распознавание образов – это научная дисциплина, целью которой является классификация объектов по нескольким категориям или классам. Распознаваемые объекты называются образами и представляют собой вектора 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  $Y$ .

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,  $y_i \in \{\omega_1, \omega_2, \dots, \omega_c\}$ ;  $c \geq 2$  – число классов.

### 2.2. Виды классификаторов.

### 2.3. Случайные леса.

Случайный лес – статистический 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.

#### 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  $e$ .

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  $v_i$ .

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  $e_i = v_{i-1} v_i$ ,  $u_i = (v_{i-1}, v_i)$ ,  $1 \leq i \leq k$ . 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.

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

1. Подмножество  $X_t \subset X$ ;
2. Подвыборка  $D_t \subset D$ , такая, что  $D_t = \{(x, y) \in D \mid x \in X_t\}$ ;
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 потомков вершины  $t$ .

Заметим, 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:

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

Цель построения дерева решений состоит в классификации векторов  $x$  из распределения  $P(x)$ . 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  $x \in X_t$ , при этом вектор  $x$  относится к классу, являющемуся мажорантным в подвыборке  $D_t$ .

### 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.

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. 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 [1] 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 загрязненности был выбран индекс Gini.

### 2.3.3. Оптимальное расщепление вершин.

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, а в случае  $f_t(\mathbf{x}) = 1$  – к правому.

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 разбиение, которое максимизирует величину  $\Delta i(t)$ , 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  $N$  – размер подвыборки  $D_t$ ;  $L$  и  $R$  – размеры подвыборок, соответствующих потомкам  $t_L$  и  $t_R$ ;  $l_j$  и  $r_j$  – число экземпляров класса  $\omega_j$  в потомках  $t_L$  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 наблюдений. Обычно синтез решений отдельных классификаторов, составляющих ансамбль, осуществляется путем их голосования. Основным достоинством ансамбля классификаторов перед отдельным классификатором является то, что при соблюдении определенных условий, он обладает значительно более

высокой точностью, а также устойчив к «зашумлению» обучающей выборки. Необходимым и достаточным условием высокой точности ансамбля классификаторов является то, чтобы составляющие его отдельные классификаторы были достаточно точны и различны, то есть совершали ошибки на различных примерах. Для выполнения этого условия в случайных лесах применяются два метода: манипулирование примерами обучающей выборки и инъекция случайности в индуктивный алгоритм.

Для манипулирования примерами обучающей выборки 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. Бутстрепом называется случайная выборка с возвращением из исходной обучающей выборки, имеющая тот же объем, что и обучающая выборка.

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 разделения вершин. Тогда алгоритм построения случайного леса может быть представлен в следующем виде:

Для  $i = 1, 2, \dots, B$  выполнить:

1. Сформировать бутстреп выборку  $S_i$  размера  $l$  по исходной обучающей выборке  $D$ ;
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:
  - a) Из исходного набора  $n$  признаков случайно выбрать  $m$  признаков;
  - b) Из  $m$  признаков выбрать признак, который обеспечивает наилучшее расщепление;

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

Классификация примера  $\mathbf{x} \notin D$  осуществляется следующим образом. Пусть  $\hat{\omega}_i(\mathbf{x}) \in \{\omega_j\}_{j=1}^C$  – класс, предсказанный деревом решений  $T_i$  для вектора  $\mathbf{x}$ , то есть  $T_i(\mathbf{x}) = \hat{\omega}_i(\mathbf{x})$ . Тогда  $\hat{\omega}_B(\mathbf{x})$  – класс наиболее часто встречающийся во множестве  $\{\hat{\omega}_i(\mathbf{x})\}_{i=1}^B$ , то есть класс предсказанный случайным лесом для вектора  $\mathbf{x}$ .

### 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. 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 (поскольку выборка с возвращением, то некоторые наблюдения в нее не попадают, а некоторые попадают несколько раз). 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 векторов обучающей выборки при таком способе классификации представляет собой оценку вероятности ошибочной классификации случайного леса методом OOB. Практика применения оценки OOB показала, что в случае, если количество деревьев достаточно велико, то эта оценка обладает высокой точностью.

Обычно оценку вероятности ошибочной классификации, найденную методом OOB, обозначают как OOBЕ (Out-Of-Bag-Error). Также часто используют оценку OOB для отдельных деревьев ансамбля, обозначая ее  $OOBE_T$ . 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 случайные леса обучаются до начала их тестирования, то есть фазы обучения и тестирования разделены. Однако на практике обучающие данные часто не

могут быть даны заранее, а поступают последовательно. Например, при решении задачи обнаружения и слежения за объектами видеопоследовательности, необходимо обучать классификатор и получать его прогнозы в режиме реального времени (on-line).

Так как случайный лес представляет собой ансамбль деревьев решений, объединённых посредством баггинга, то для реализации on-line обучения случайного леса необходимо осуществлять как баггинг, так и построение самих деревьев решений в on-line режиме.

#### 2.4.1. On-line баггинг.

Допустим, что обучающая выборка  $D$  дана заранее и имеет размер  $N$ . Тогда для построения каждого дерева случайного леса необходимо создать бутстреп выборку  $S$  размера  $N$ . Пусть случайная величина  $K_j$  ( $j = 1, \dots, N$ ) принимает значение 1, если пример  $(\mathbf{x}, y)$  попал в выборку  $S$  на позицию  $j$  и значение 0 в противоположном 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 параметром  $p$ . 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  $p$ , то есть

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

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

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

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

$$K \sim \text{Poisson}(1) \text{ при } N \rightarrow \infty.$$

Таким образом, при баггинге в on-line режиме, последовательно поступающие данные возможно моделировать при помощи пуассоновского распределения. Для



этого необходимо обновлять каждое дерево  $T_i$  на каждом примере  $K \in \xi$  раз подряд, где  $\xi = \{\xi_1, \xi_2, \dots\}$  – последовательность случайных величин, принимающих неотрицательные целочисленные значения, такая что для всякого  $k$  выполняется условие

$$P\{\xi_n = k\} \sim \text{Poisson}(1) \text{ при } n \rightarrow \infty.$$

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

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

Главной проблемой при построении дерева решений в on-line режиме является правильность выбора моментов разделения вершин. Правильность этого выбора крайне важна, так как надежность статистики, накапливаемой в последующих вершинах-сыновьях, напрямую зависит от правильности выбора момента разделения вершины-отца.

При on-line обучении статистика в вершине накапливается с течением времени и для обеспечения правильности выбора момента разделения, необходимо соблюдать два условия:

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

Для соблюдения этих условий введем в алгоритм разделения вершины две пороговых величины  $\alpha_{split}$  и  $\beta_{split}$ , численные значения которых подбираются экспериментально так, чтобы минимизировать ошибку классификации  $OUBE$ . Здесь  $\alpha_{split}$  – минимальное число примеров, прошедших через вершину до ее разделения;  $\beta_{split}$  – минимальное значение уменьшения загрязненности вершины такое, что  $\exists s \in S(t): \Delta i(s) > \beta_{split}$ , где  $S(t)$  – набор случайно выбранных признаков для вершины  $t$ .

От дерева, построение которого производится в on-line режиме, также требуется возможность произведения классификации сразу же после разделения вершины, то есть до получения новых данных.

Пусть  $t$  – вершина, удовлетворяющая условиям разделения с пороговыми величинами  $\alpha_{split}$  и  $\beta_{split}$ . При прохождении примеров через эту вершину, были накоплены статистики  $\mathbf{p}_{t_L}(s) = \{p_{t_L}^j(i)\}_{j=1}^c$  и  $\mathbf{p}_{t_R}(s) = \{p_{t_R}^j(i)\}_{j=1}^c$ , представляющие собой статистики меток классов  $\{\omega_j\}_{j=1}^c$  левых и правых возможных сыновей вершины  $t$ , разделенных по признаку  $s$ . Тогда при разделении вершины, из всего числа признаков  $n$  выбирается случайный набор  $S(t)$  из  $m$  признаков, после чего из набора  $S(t)$  выбирается признак  $\hat{s}$ , обеспечивающий наилучшее расщепление, и создаются вершины-сыновья  $t_L$  и  $t_R$ , разделенные по признаку  $\hat{s}$ . После этого значения статистик  $\mathbf{p}_{t_L}(\hat{s})$  и  $\mathbf{p}_{t_R}(\hat{s})$  передаются левому и правому сыновьям соответственно. Таким образом, новые вершины появляются с уже накопленной статистикой, что позволяет производить классификацию без необходимости получения новых данных.

#### 2.4.3. Адаптивное исключение деревьев из ансамбля.

Часто при on-line обучении распределение примеров в обучающей выборке  $P(\mathbf{x})$  меняется на протяжении времени. Следовательно, необходимо иметь механизм, способный «отучивать» случайный лес от старой информации. Этого можно достичь, исключая из ансамбля деревья, неспособные верно классифицировать примеры с изменившимся распределением. 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 классифицировать примеры с изменившимся распределением, гарантирует адаптацию классификатора к изменяющимся условиям на протяжении времени.

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, то есть для этих деревьев число обновлений на примере  $(\mathbf{x}, y)$  равно нулю. Следовательно, на примере  $(\mathbf{x}, y)$  мы можем оценить как  $OUBE$  для всего леса, так и  $OUBE_T$  для дерева, не участвующего в обучении по этому примеру.

Очевидно, что деревья, неспособные верно классифицировать примеры, имеют большое значение  $OUBE_T$ . Следовательно, на основе значений  $OUBE_T$  можно

производить случайное исключение деревьев из ансамбля, причем вероятность исключения -го дерева должна зависеть от  $OBE_T(i)$ .

#### 2.4.4. Алгоритм on-line обучения.

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 в следующем виде:

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

1. Найти число обновлений  $K_i$  дерева  $T_i$  на примере  $(\mathbf{x}, y)$ .  $K_i \sim \text{Poisson}(1)$ .
2. Если  $K_i > 0$ , то  $K_i$  раз обновить дерево  $T_i$  на примере  $(\mathbf{x}, y)$ . Для всех неразделенных вершин  $t$  дерева  $T_i$ , для которых выполнены условия: число примеров в вершине не менее  $\alpha_{split}$  и  $\exists s \in S(t): \Delta i(s) > \beta_{split}$  – следовать следующему подалгоритму:
  - а) Из множества  $S(t)$  выбрать признак  $\hat{s}$ , который обеспечивает наилучшее расщепление;
  - б) Разделить вершину  $t$  по признаку  $\hat{s}$ , то есть построить ее сыновей  $t_L(\hat{s})$  и  $t_R(\hat{s})$ ;
  - в) Передать значения статистик  $\mathbf{p}_{t_L}(\hat{s})$  и  $\mathbf{p}_{t_R}(\hat{s})$  левому и правому сыновьям.
3. Если  $K_i = 0$ , то оценить  $OBE_T(i)$ .

### 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.

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. 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:

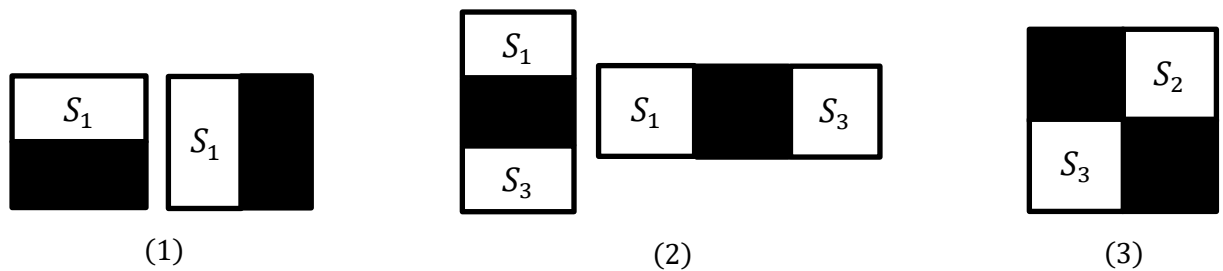
$$S_T = \{(\mathbf{x}_1, 1), (\mathbf{x}_2, 0), \dots, (\mathbf{x}_l, 0)\},$$

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 [2]. 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.



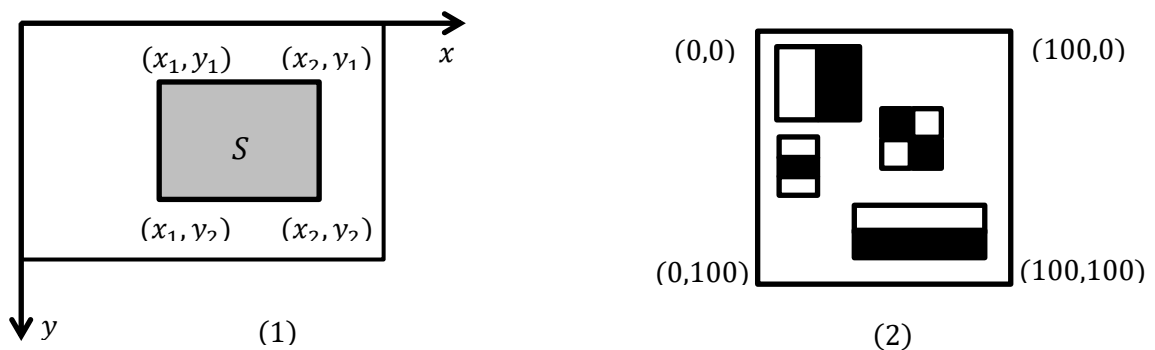
**Рисунок 4.** Типы признаков Хаара: двух-прямоугольные (1), трех-прямоугольные (2) и четырех-прямоугольные (3).

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:  $L(S_1) + L(S_4) - L(S_2) - L(S_3)$ .

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 противоположном случае.

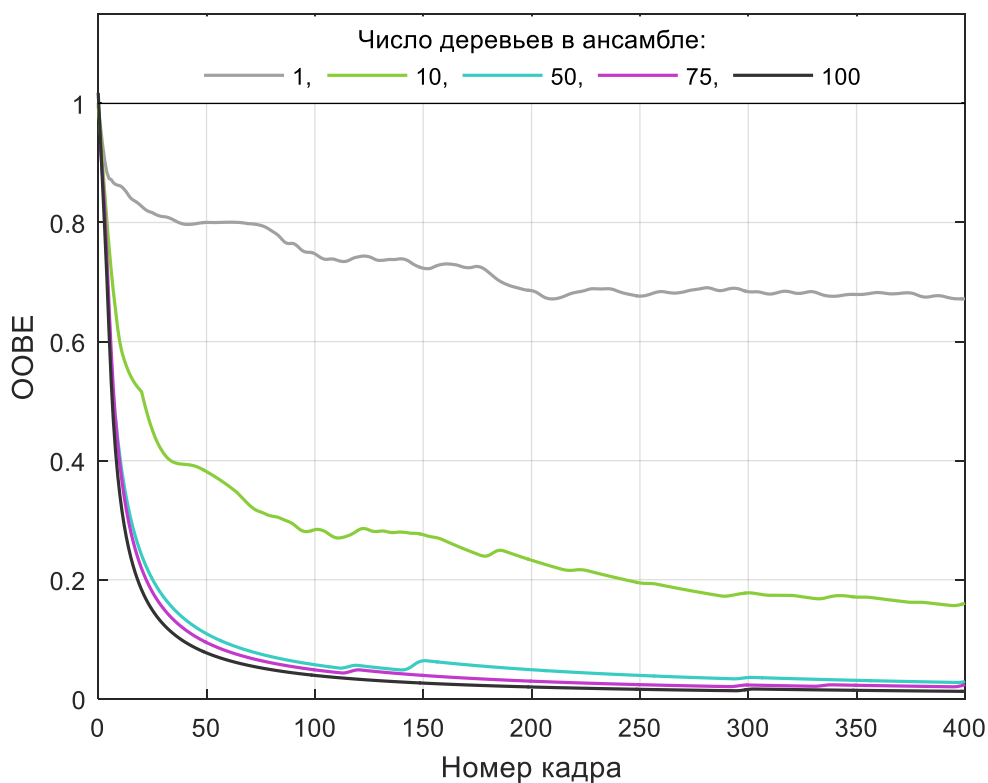
## **Глава 4. Результаты экспериментов.**

### **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.



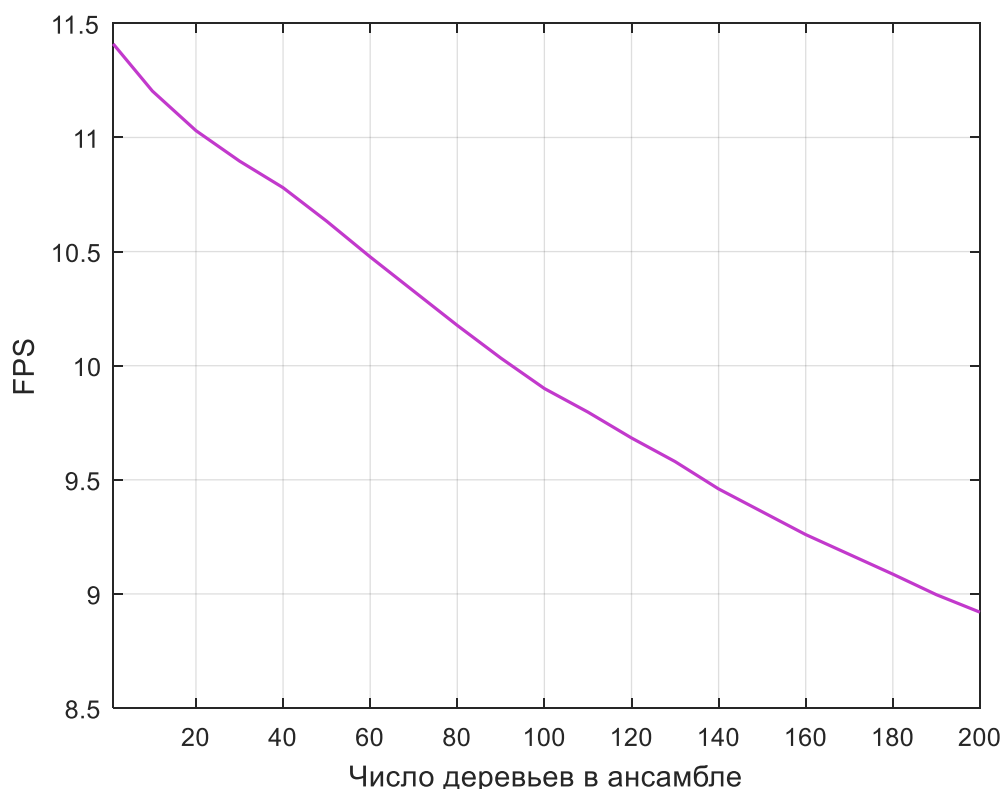
**Рисунок 6.** Зависимость вероятности ошибочной классификации (OOBE) от числа деревьев в ансамбле при обучении на однородных примерах.

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.

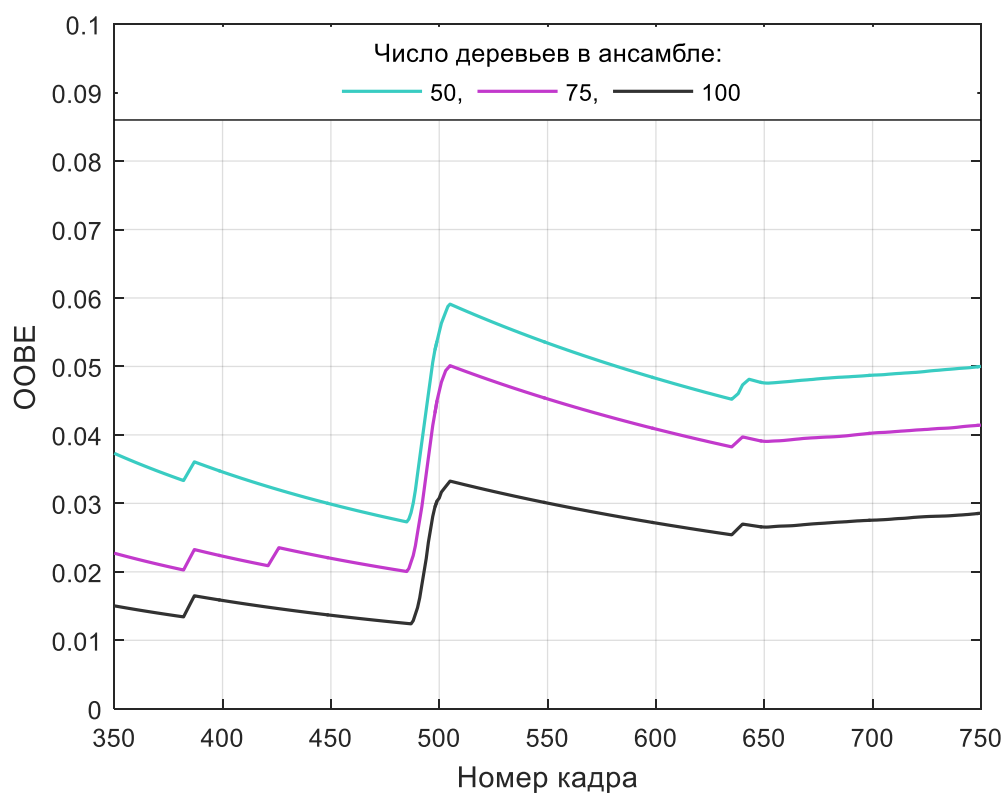
Для подбора оптимальных параметров классификатора проведем исследование зависимости производительности алгоритма классификации от числа деревьев, входящих в ансамбль.



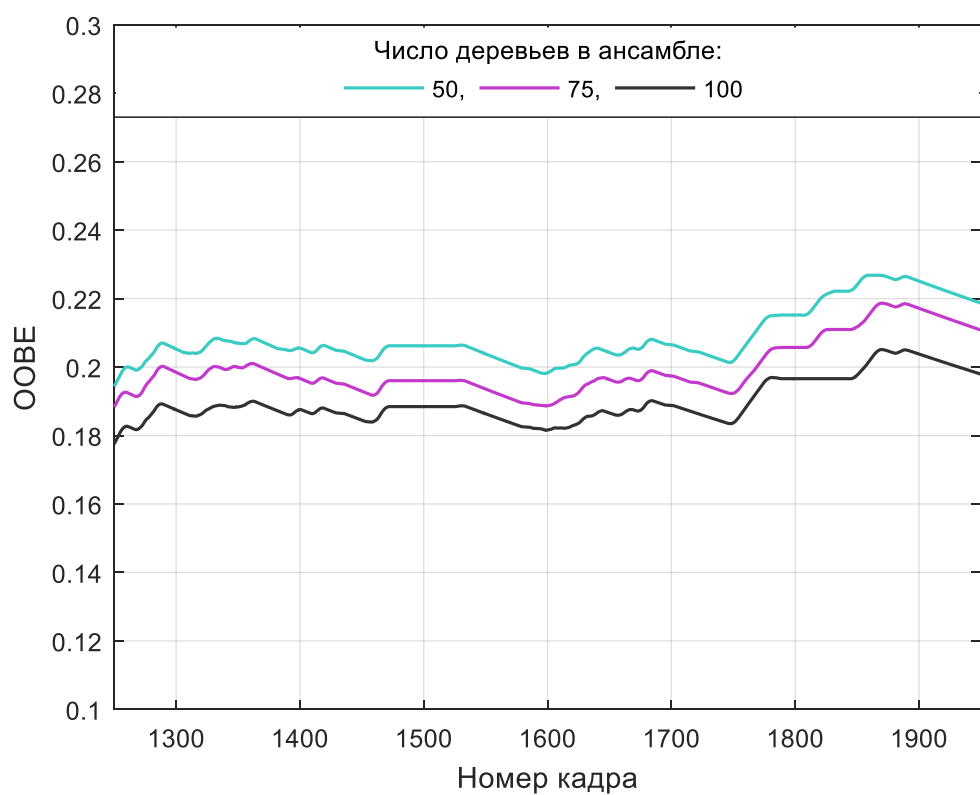
**Рисунок 7.** Зависимость производительности алгоритма классификации (число кадров, обрабатываемых в секунду) от числа деревьев в ансамбле.

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.



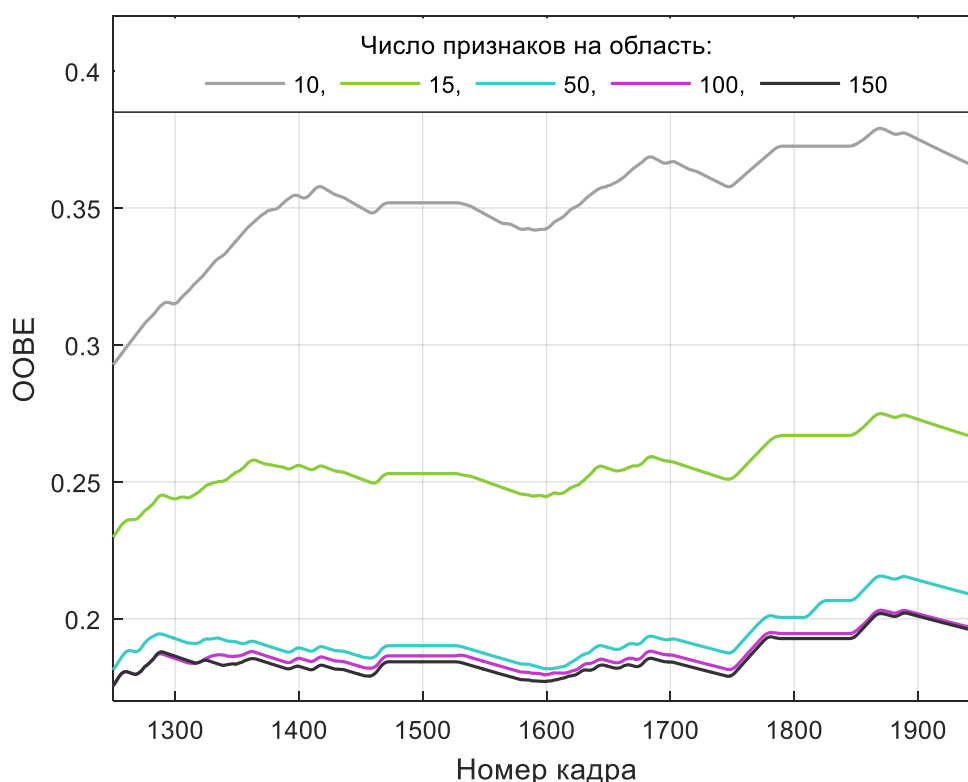
**Рисунок 8.** Рост вероятности ошибочной классификации (OOBE) при изменении обстановки окружающей сцены, вызванной изменением освещенности.



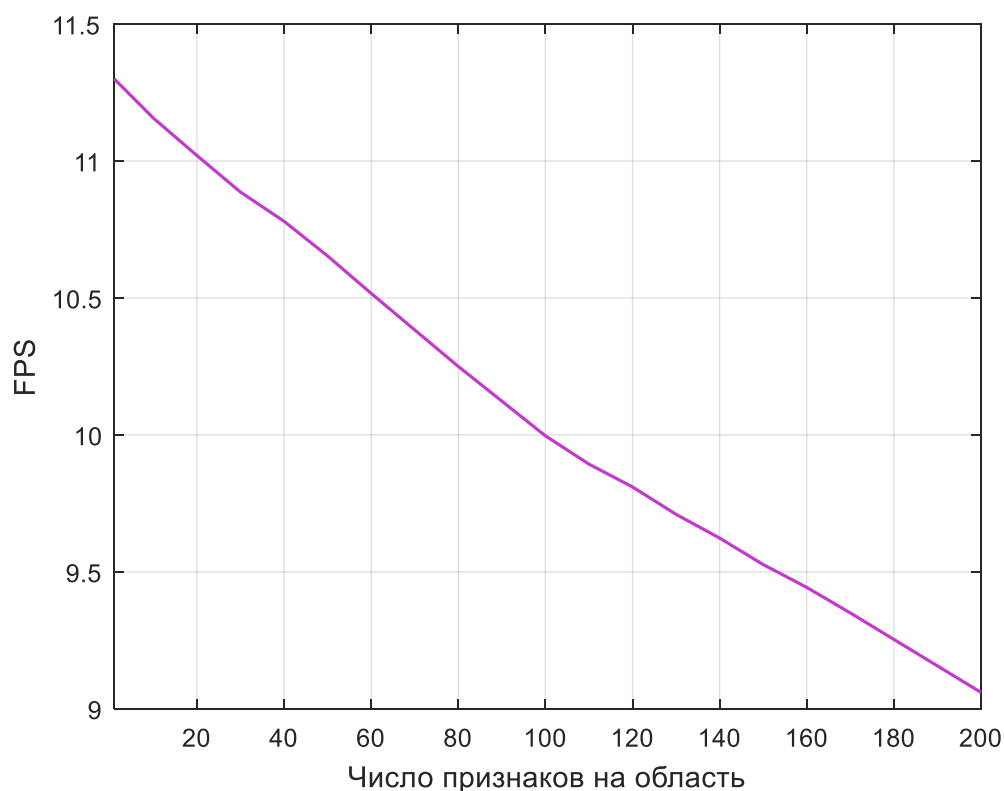
**Рисунок 9.** Изменение вероятности ошибочной классификации (OOBE) при длительном промежутке работы классификатора.

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.



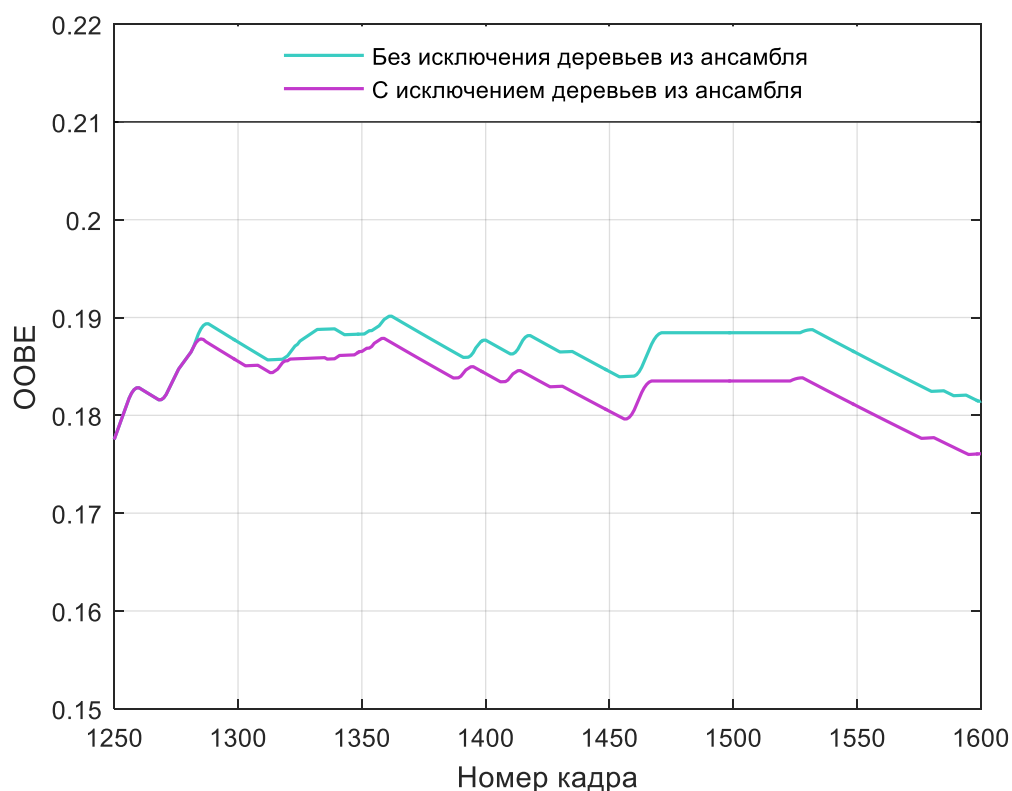
**Рисунок 10.** Зависимость вероятности ошибочной классификации (OOBE) от числа признаков, вычисляемых в каждой из выделенных областей.



**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.

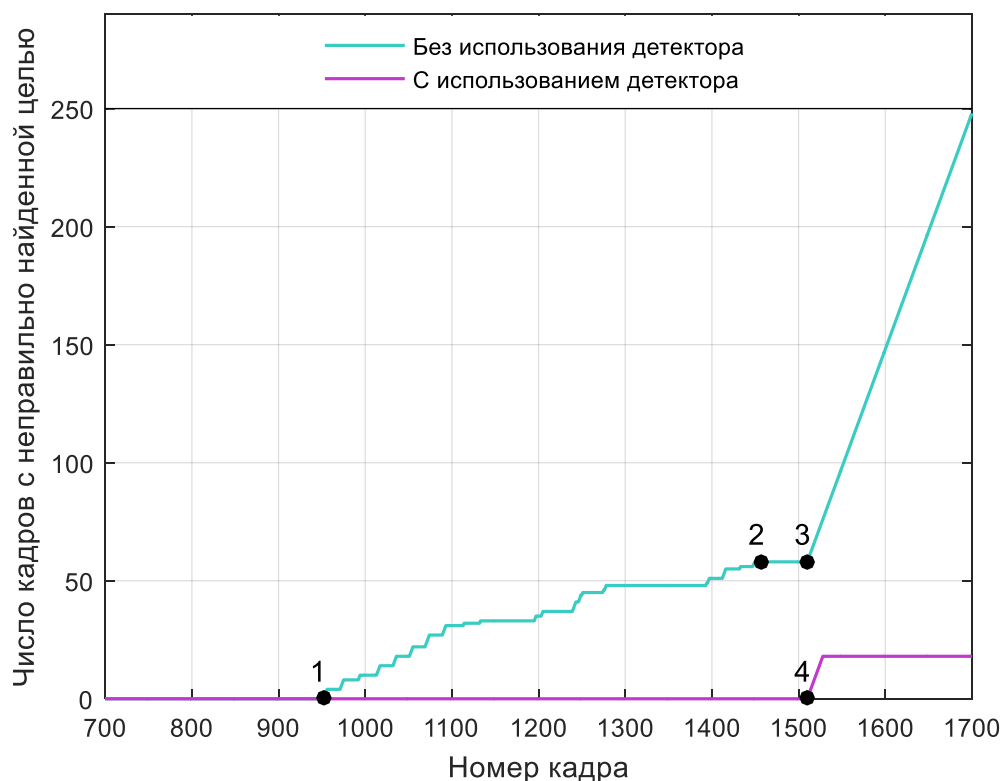
Далее рассмотрим результаты применения метода адаптивного исключения деревьев из ансамбля, предложенного в §2.4.3.



**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 [статья].
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 бóльшего 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 [статья].
7. Использование сверточных нейронных сетей в качестве классификатора при решении задачи распознавания образов.

## Список использованной литературы.

[1]

21. *Breiman L., Friedman R., Olshen R., Stone C.* Classification and Regression Trees. Belmont, California: Wadsworth International, 1984. 342 p.

[2] Viola Jones

[3] (<http://www.bmva.org/bmvc/1988/avc-88-023.pdf>)