

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(**Н И У « Б е л Г У »**)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

**КАФЕДРА ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ
СИСТЕМ И ТЕХНОЛОГИЙ**

**ИССЛЕДОВАНИЕ МЕТОДОВ ОБНАРУЖЕНИЯ ДВИЖУЩИХСЯ
ОБЪЕКТОВ НА ВИДЕО-ДАННЫХ**

Выпускная квалификационная работа
обучающегося по направлению подготовки
11.04.02 Инфокоммуникационные технологии и системы связи,
магистерская программа «Системы и устройства радиотехники и связи»
очной формы обучения, группы 07001636
Балабановой Натальи Сергеевны

Научный руководитель
канд. техн. наук,
доцент кафедры
Информационно-
телекоммуникационных
систем и технологий
НИУ «БелГУ» Заливин А.Н.

Рецензент
канд. техн. наук,
начальник отдела
программного обеспечения
информационных средств
ООО "НПП "ЭИТ" БелГУ"
Соловьев Виктор Иванович

БЕЛГОРОД 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ОБРАБОТКА ИЗОБРАЖЕНИЙ.....	5
1.1 ПРЕДОБРАБОТКА ИЗОБРАЖЕНИЙ	5
1.2 ПОСТОБРАБОТКА ИЗОБРАЖЕНИЙ.....	6
1.3 МАТЕМАТИЧЕСКАЯ МОРФОЛОГИЯ	7
ГЛАВА 2. МЕТОДЫ ОБНАРУЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ	9
2.1 ОБЗОР МЕТОДОВ ОБНАРУЖЕНИЯ ДВИЖУЩЕГОСЯ ОБЪЕКТА	9
2.2 МЕТОДЫ ОБНАРУЖЕНИЯ ОБЪЕКТОВ, ОСНОВАННЫЕ НА ВЫЧИТАНИИ И УСРЕДНЕНИИ ФОНА.....	10
2.3 МЕТОДЫ ОБНАРУЖЕНИЯ ОБЪЕКТОВ, ОСНОВАННЫЕ НА ОЦЕНКЕ ОПТИЧЕСКОГО ПОТОКА.....	29
2.3.1 <i>Оценка оптического потока</i>	30
2.3.2 <i>Методы определения оптического потока</i>	32
ГЛАВА 3. МЕТОДЫ СОПРОВОЖДЕНИЯ ОБЪЕКТОВ.....	37
3.1 КЛАССИФИКАЦИЯ МЕТОДОВ СОПРОВОЖДЕНИЯ.....	37
3.2 МЕТОДЫ СОПРОВОЖДЕНИЯ ОСОБЫХ ТОЧЕК.....	39
ГЛАВА 4. РАЗРАБОТКА АЛГОРИТМОВ ОБНАРУЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ	45
ГЛАВА 5. ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ	54
5.1 КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ	54
5.2 ПЛАНИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ	56
5.3 ОПИСАНИЕ РЕЗУЛЬТАТОВ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ.....	58
ГЛАВА 6. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	63
ЗАКЛЮЧЕНИЕ	66
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	68
ПРИЛОЖЕНИЕ А.....	77

ВВЕДЕНИЕ

Компьютерное зрение, обработка изображений и распознавание образов являются обширной областью науки и широко применяются в таких направлениях, как дистанционное зондирование, медицинская диагностика, взаимодействие человек-компьютер и т.д. Также компьютерное зрение нашло применение в видеоаналитике.

Видеоаналитика представляет собой технологию, которая использует методы компьютерного зрения для автоматизированного получения различных данных с видеокамер. Данные формируются на основе анализа последовательности изображений, поступающих в режиме реального времени или из архивных записей. Видеоаналитика определяется, как программное обеспечение для работы с видеоконтентом [1]. Данное программное обеспечение основано на множестве алгоритмов машинного зрения, позволяющих вести видеомониторинг и производить анализ данных без прямого участия человека.

Видеоаналитика автоматизирует функции средств охраны:

- обнаружение;
- слежение;
- распознавание;
- прогнозирование.

Обнаружение движущихся объектов из видеопоследовательностей является ключевым вопросом системы визуального наблюдения. Важно качественно сегментировать области движения для дальнейшего распознавания, отслеживания, а также понимания поведения объекта.

Из-за повышения требований к точности измерений при сборе аналитической информации в системах видеоаналитики должна повышаться достоверность обнаружения подвижных объектов при определении их местоположения. Это говорит о том, что задача эффективного обнаружения движущихся объектов является актуальной.

Целью данной выпускной квалификационной работы является исследование методов вычитания фона для задачи обнаружения движущихся объектов и оценка их показателей качества.

Основные задачи, решаемые в выпускной квалификационной работе, для достижения поставленной цели:

- Произвести обзор современных методов обнаружения движущихся объектов на видео-данных;
- Произвести выбор методов обнаружения движущихся объектов на видео-данных;
- Произвести разработку алгоритмов для выбранных методов;
- Произвести их реализацию в виде программного комплекса;
- Произвести экспериментальные исследования выбранных методов;
- Произвести оценку полученных результатов вычислительных экспериментов.

ГЛАВА 1. ОБРАБОТКА ИЗОБРАЖЕНИЙ

Цель методов обнаружения подвижных объектов на видео-данных состоит в обнаружении значимых изменений, появляющихся на видео, при игнорировании малозначительных. К числу малозначительных изменений можно отнести шум на изображении, его уровень может меняться от кадра к кадру. В целях уменьшения влияния шума на результат определения движущихся объектов используют предобработку и постобработку изображений.

1.1 Предобработка изображений

Предобработка изображений состоит из таких операций, как геометрические и яркостные корректировки [2]. Также иногда используется информация о расстоянии до объекта (глубине).

Корректировка яркости.

На практике применяются несколько методов компенсации изменения освещенности между кадрами, возникающие из-за изменений яркости или позиции источника света сцены. Одним из таких методов является метод нормализации значений яркости пикселей изображения. Он применяется для получения конкретного значения математического ожидания и дисперсии [3, 4].

По данному методу могут быть нормализованы не только текущий кадр, но и фоновое изображение (получение нулевого математического ожидания с единичной дисперсией) [2]. Однако, несмотря на получение лучших результатов для локальных изменений освещенности, существует вероятность появления дополнительных артефактов [2]. Алгоритмы, которые используют нормализацию дают плохие результаты для темных областей изображения.

Информация о глубине (расстоянии до объектов).

Использование этих данных позволяет уменьшить количество проблем при обнаружении подвижного объекта, так как они в наименьшей степени подвержены искажению при изменении условий освещённости на видеоданных. Однако методы, которые для анализа используют информацию о расстоянии, являются неэффективными для малоконтрастных областей или областей, которые недоступны для всех камер. Эффективность данного метода также падает при выделении переднего плана (подвижных объектов), которые находятся в непосредственной близости от фона (рука на стене, ноги на полу и т.д.).

1.2 Постобработка изображений

Разделение объектов.

Разделение обнаруженных и выделенных объектов, которые находятся на достаточном расстоянии, не является сложной задачей. Ее решение сводится к вводу простого порога. Трудной задачей представляется разделение находящихся рядом или пересекающихся объектов [5]. Использование того или иного метода обусловлено конкретной ситуацией и требуемой точностью выделения объекта.

Для разделения подвижных объектов используются подходы, основанные на построении сети (дерева графов), не только в загруженном видео, но и в режиме реального времени [6, 7].

Выделение границ (контура).

Точность выделения области движущегося объекта определяет точность выделения границ данного объекта. Можно выделить некоторые ошибки, возникающие при выделении области движущегося объекта: шум, ложное обнаружение (ошибка первого рода) и пропуск цели (ошибка второго

рода). Для решения данных проблем на стадии обнаружения используют метод соответствующей фильтрации [8].

Существует множество причин появления шумов, оказывающих неизбежное влияние на результат выделения подвижного объекта. Некоторые из них – изменение освещенности, изменение положения камеры, цветовой шум, шум системы. Для устранения шума применяется адаптивная фильтрация. Однако для получения более точной маски объекта используется сигма-дельта фильтрация. Она позволяет уменьшить влияние нестабильности фона и шумов, которые вызваны небольшим движением камеры или фона.

1.3 Математическая морфология

Математическая морфология – это анализ изображения с точки зрения его формы. Применение данного подхода над изображениями помогает изменить форму объектов, которые содержатся на изображении [9].

Математическая морфология применяется в различных системах, работающих над обработкой изображений, на различных этапах. Данный подход также используется для следующих целей:

- улучшение визуальных характеристик изображения (яркость, контрастность и т.д.);
- восстановление испорченных изображений, например, реставрация фотоснимков;
- обнаружение контуров;
- снижение уровня шума.

Метод математической морфологии может использоваться для различных категорий изображений: цветных, чёрно-белых изображений и для изображений в оттенках серого. Для этих трёх случаев формально данный подход определяется несколько по-разному. Будет рассмотрен случай бинарных чёрно-белых растров. Так как его применение над данной

категорией изображений позволяет также провести дополнительную фильтрацию шума в маске движения. Этот процесс считается эффективным, поскольку маска состоит из нулей и единиц. Существует большое количество операций математической морфологии: перенос, наращивание, эрозия, размыкание, а также замыкание [10].

Замыкание бинарного изображения A структурным элементом B обозначается $A \bullet B$ определяется как:

$$A \bullet B = (A \oplus B) - B \quad (1.1)$$

Данная операция «закрывает» небольшие внутренние «дырки» в изображении, и убирает углубления по краям области.

Размыканием бинарного изображения A структурным элементом B обозначается $A \circ B$ и определяется как:

$$A \circ B = (A - B) \oplus B \quad (1.2)$$

Размыкание отсеивает все объекты, которые оказываются меньше, чем структурный элемент, но в тоже время помогает избежать сильного уменьшения размера объектов. Также размыкание идеально подходит для удаления линий, толщина которых меньше, чем диаметр структурного элемента. Данная операция делает контуры объектов более гладкими.

ГЛАВА 2. МЕТОДЫ ОБНАРУЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ

2.1 Обзор методов обнаружения движущегося объекта

Для решения задач по обнаружению подвижных объектов применяются различные подходы. Условно их можно разделить на 2 класса:

1. методы, базирующиеся на временных различиях (метод вычисления оптического потока);
2. моделирование и вычитание заднего фона.

Основой первых научных трудов по обнаружению подвижных объектов, написанных в конце 70-х годов XX века, являлось различие соседних видеокадров. В данных методах главными правилами были так называемые аспекты статистических гипотез. В 90-х годах XX века в основном развивались алгоритмы, которые основывались на изменении положения пикселей изображения. Применялся не только любой отдельный пиксель, но и блоки пикселей. В 90-х годах XX века был предложен комплексный метод опознавания [11], ну а в 2005 году [12] предложен метод «от противного», базирующийся на принципе Гельмгольца. Целью этого метода было обнаружение видеокадра, где не было подвижного объекта.

Для обнаружения изменений на переднем плане методом вычитания фона используется некоторая модель. Данная модель фона обновляется через определенный интервал времени. В качестве простейшей модели может использоваться первый видеокадр [13]. При вычитании данного изображения из последующих подвижный объект представляется достаточно грубо. Эта схема вычитания может использоваться только с простыми сценами, где присутствует однородный контраст и незначительные изменения.

Выделение фона может осуществляться различными методами и алгоритмами с помощью статистических функций: функции среднего, медианы, а также с помощью распределения Гаусса.

Также для обнаружения подвижных объектов применяются методы, использующие в качестве области анализа частотную область, которые базируются на вейвлет-преобразованиях. Иногда вейвлет-преобразование используется для оценки фона на базе предыдущих кадров видеопоследовательности, также для выделения движущегося объекта и его расположения [14]. Существует и другой способ, базирующийся на дробной статистике низкого порядка [15].

2.2 Методы обнаружения объектов, основанные на вычитании и усреднении фона

Вычитание фона (background subtraction) из кадра видео является простым методом, который применяется в задаче выделения движущегося объекта [16]. Процедура вычитания фона предполагает, что существует модель фона, построенная для данного видео:

$$F = \{F(x, y), 0 \leq x < width, 0 \leq y < height\}, \quad (2.1)$$

где *width* - ширина кадра, *height* - высота кадра.

В идеальных условиях такая модель представляет собой сцену без подвижных объектов. Однако необходимо ее регулярно обновлять, для того чтобы учитывать изменение условий освещенности и настроек камеры, таких как поворот, наклон и изменение фокусного расстояния. Чаще всего для поддержания актуальности фона используют процедуру обновления модели фона с течением времени [17].

Процедура вычитания модели фона состоит из двух этапов:

- Вычитание фонового изображения из текущего кадра видеопоследовательности. На данном этапе осуществляется попиксельное вычитание интенсивностей кадра видео и фонового изображения

$$D_k(x, y) = \text{abs}(I_k(x, y) - F(x, y)), k = \overline{1, N}, \quad (2.2)$$

где $I_k(x, y)$ - интенсивность определенного кадра.

- Разделение пикселей по принадлежности к фону и объекту, то есть построение бинарного изображения (маски). Считается, что пиксель относится к объекту и имеет белый цвет в маске, если разность интенсивности фона и текущего кадра для данного пикселя превышает некоторое пороговое значение, в противном случае, принимается, что пиксель принадлежит фону

$$M_k(x, y) = \begin{cases} 255, D_k(x, y) \geq \tau \\ 0, D_k(x, y) < \tau \end{cases}, k = \overline{1, N} \quad (2.3)$$

От качества построенной модели фона зависит качество определения положения подвижного объекта с помощью метода вычитания фона [18]. Методы вычитания фона делятся на две группы в зависимости от механизма построения фонового изображения:

1. Нерекурсивные методы.

При обновлении модели фона данные методы используют для текущего кадра данные об интенсивностях пикселей некоторого набора предыдущих моделей фона [19] (либо кадров) и текущего кадра. Можно выделить следующие методы, которые относятся к нерекурсивным:

- *Метод вычитания текущего и предыдущего кадра* [20]. Данный метод предполагает, что для определенного кадра I_k модель фона F_k

совпадает с предыдущим кадром, т.е. $F_k = I_{k-1}$. Из этого следует, что на первом этапе алгоритма вычитания фона вычисляется разница двух последовательно идущих кадров видеопоследовательности

$$D_k(x, y) = abs(I_k(x, y) - F_k(x, y)) = abs(I_k(x, y) - I_{k-1}(x, y)), k = \overline{2, N} \quad (2.4)$$

- *Метод усреднения некоторого количества предшествующих кадров* [21]. В данном случае количество кадров, которое используется для построения модели фона, обозначается как s . Тогда модель фона F_k для кадра I_k выражается формулой:

$$F_k(x, y) = \frac{1}{s} \sum_{j=0}^{s-1} I_{k-j}(x, y) \quad (2.5)$$

- *Метод определения медианы фиксированного количества предшествующих кадров* [22]. Также как и в методе усреднения предполагается, что s – число кадров, на основании которых будет обновляться модель фона. Тогда модель фона F_k определяется по формуле:

$$F_k(x, y) = med_{j=0, s-1} \{I_{k-j}(x, y)\} \quad (2.6)$$

- *Метод, основанный на использовании минимаксного фильтра*. В данном методе используется обучающая последовательность без подвижных объектов для определения трех значений каждого пикселя фонового изображения: минимальная яркость $Z_{i,j_{\min}}^0$, максимальная яркость $Z_{i,j_{\max}}^0$, а также максимальное изменение яркости между соседними кадрами $\Delta Z_{i,j}^0$ [23]. Данные параметры вычисляются по формулам:

$$z_{i,j_{\min}}^0 = \min z_{i,j}^0, z_{i,j_{\max}}^0 = \max z_{i,j}^0, \Delta z_{i,j}^0 = \max |z_{i,j}^t - z_{i,j}^{t-1}| \quad (2.7)$$

Данные значения определяются по параметрам нескольких кадров и обновляются для моделей фона с некоторым интервалом времени.

Главное преимущество нерекурсивных методов – это простота реализации и скорость обновления моделей фона при переходе от одного кадра к другому. Однако точность работы данных методов зависит также и от скорости объектов [24]. Чем медленнее движется объект, тем хуже он определяется. Методы данной группы весьма неэффективны при наличии движущегося фона (листва деревьев, струящаяся вода и так далее).

Для сглаживания указанных эффектов применяют так называемое α -смешивание. В данной процедуре обновленная модель фона для кадра I_k представляется выпуклой оболочкой модели фона F_{k-1} . Выражается формулой:

$$F_k(x, y) = \alpha I_k(x, y) + (1 - \alpha) F_{k-1}(x, y) \quad (2.8)$$

2. Рекурсивные методы.

Методы этого класса для обновления модели фона используют данные об интенсивностях пикселей только текущего кадра. К рекурсивным методам относятся: гистограммный метод, метод представления модели фона смесью Гауссовых распределений (Gaussian mixture model) [25], метод "шифровальной" книги (codebook) [26], метод извлечения визуального фона (Visual Background Extractor, ViBe) [27]. Некоторые методы рассмотрены более подробно.

- *Гистограммный метод.* Данный метод разбивает всё цветовое пространство на отдельные бины. В качестве данного пространства используются различные значения: в полутоновом изображении – отрезок

изменения интенсивности, а в цветном изображении – трёхмерный куб. Гистограмма строится для всех изображений в видеопоследовательности.

При использовании гистограммного метода осуществляется анализ всех пикселей изображения [28]. Исходя из значения интенсивности/цвета, которое наблюдается в пикселе, значение соответствующего бина гистограммы увеличивается на единицу. Если величина определенного бина меньше фиксированного порогового значения, то пиксели, которые входят в состав этого бина, принадлежат фону. В противном случае, считается, что пиксели принадлежат объекту. Главной проблемой гистограммного метода является необходимость использования дополнительной памяти. Также выполняется множество операций обращения к памяти в процессе реализации алгоритма.

- *Смесь Гауссовых распределений (Gaussian Mixture Model - GMM)* [29, 30]. Построение фона по данному методу предполагает, что для любого пикселя (x_0, y_0) изображения I_k известна история изменения его интенсивности/цвета на всех предыдущих кадрах $\{X_1, X_2, \dots, X_k\} = \{I_j(x_0, y_0), j = \overline{1, k}\}$. Тогда вероятность наблюдения значения X_k , определяется как смесь Гауссовых распределений:

$$P(X_k) = \sum_{j=1}^s \omega_j^k N(X_k | \mu_j^k, \Sigma_j^k) \quad (2.9)$$

где ω_j^k – вес j -ого распределения Гаусса для кадра с номером k ,

μ_j^k – математическое ожидание,

Σ_j^k – среднеквадратичное отклонение,

$N(X_k | \mu_j^k, \Sigma_j^k)$ – функция плотности нормального распределения,

которая выражается формулой:

$$N(X_k | \mu_j^k, \Sigma_j^k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_j^k|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_k - \mu_j^k)^T (\Sigma_j^k)^{-1} (X_k - \mu_j^k)} \quad (2.10)$$

Существует предположение о том, что составляющие цвета являются независимыми и обладают одинаковой среднеквадратической погрешностью [31]. Поэтому матрицу ковариации можно записать, как

$$\Sigma_j^k = (\sigma_j^k)^2 E \quad (2.11)$$

где E – единичная матрица.

Из этого утверждения следует, что существует возможность уменьшения вычислительной трудоемкости метода за счет отсутствия необходимости вычислять матрицу, обратную к матрице ковариации Σ_j^k . Таким образом, задано распределение наблюдаемых значений цвета для каждого пикселя [32]. Новое значение цвета можно представить как одну из основных компонент полученной смеси Гауссовых распределений. И посредством этого могут проводиться обновления параметров модели. Построенные распределения сортируются в порядке уменьшения величины:

$$r_j^k = \frac{\omega_j^k}{\sigma_j^k} \quad (2.12)$$

При данной сортировке пиксель фона отвечает распределению с большим весом и малой дисперсией.

Первые B^k распределений, как правило, относятся к цвету фоновых пикселей, когда выполняется условие:

$$B^k = \arg \min_b \left\{ \sum_{j=1}^b \omega_j^k > T \right\} \quad (2.13)$$

где T – пороговое значение, параметр модели.

Когда появляется следующий кадр I_{k+1} , для каждого пикселя изображения выполняется следующий тест: применение расстояния Махаланобиса [33] определяет, какому распределению соответствует полученное значение.

$$\sqrt{(X_{k+1} - \mu_j^k)^T (\Sigma_j^k)^{-1} (X_{k+1} - \mu_j^k)} < 2.5\sigma_j^k \quad (2.14)$$

Тогда возможны две ситуации:

1. При условии, что нашлось соответствующее распределение Гаусса, то в зависимости от того, определяет ли оно распределение фоновых пикселей (входит в группу из B^k распределений) или нет, текущий пиксель распределяется к фоновому или определяется, как принадлежащий объекту.

2. В случае, если ни одно распределение не соответствует распределению Гаусса, удовлетворяющего условию (2.14), то считается, что пиксель принадлежит объекту.

По данному правилу формируется двумерная маска.

Прежде, чем приступить к работе над следующим кадром, необходимо обновить параметры распределений: математическое ожидание μ_j^k и среднеквадратичное отклонение σ_j^k . Обновление выполняется по-разному в зависимости от результата нахождения соответствующего распределения для цвета текущего пикселя [34]:

1. Соответствие обнаружено. В данном случае весовые коэффициенты, составляющие смесь Гауссовых распределений, которым соответствует X_{k+1} , и параметры распределений пересчитываются по формулам:

$$\omega_j^{k+1} = (1 - \alpha)\omega_j^k + \alpha \quad (2.15)$$

$$\mu_j^{k+1} = (1 - \rho)\mu_j^k + \rho X_{k+1} \quad (2.16)$$

$$(\sigma_j^{k+1})^2 = (1 - \rho)(\sigma_j^k)^2 + \rho(X_{k+1} - \mu_j^{k+1})(X_{k+1} - \mu_j^{k+1})^T \quad (2.17)$$

где α - заданная константа,

$$\rho = \alpha N(X_{k+1} | \mu_j^k, \Sigma_j^k).$$

В случае, если для всех распределений X_{k+1} не соответствует, параметры не изменяются, однако пересчитываются коэффициенты ω_j^{k+1} по следующей формуле:

$$\omega_j^{k+1} = (1 - \alpha)\omega_j^k \quad (2.18)$$

2. Соответствие не обнаружено. В таком случае последнее (в смысле введенного отношения порядка) распределение Гаусса замещается распределением с новыми параметрами. Математическое ожидание приравнивается текущему значению цвета пикселя:

$$\mu_s^{k+1} = X_{k+1} \quad (2.19)$$

Дисперсия $(\sigma_s^{k+1})^2$ выбирается максимально возможной, а вес ω_s^{k+1} - минимально допустимым [35].

Количество распределений выбирается исходя из сложности фона и имеющихся вычислительных мощностей (в [25] предлагается использовать от 3 до 5 Гауссовых распределений).

- **ViVe** - это метод, использующий несколько образцов фона для создания фоновых моделей. В данном методе вводятся новые механизмы обновления [36]. ViVe характеризуется следующими правилами:

- Работа алгоритма ведется на уровне пикселей. Так работают многие эффективные алгоритмы по классификации объектов [37]. Информация на уровне пикселей является более предпочтительной для агрегированных функций, поскольку процесс сегментации является локальным. А обработка пикселей позволяет расширить возможности реализации данного алгоритма для многих аппаратных архитектур.

- Алгоритм сохраняет предыдущие значения для каждого пикселя, а не создает статичную модель [38]. Для этого существует две причины. Во-первых, потому, что статистическая значимость предыдущих значений намного выше значений, измерения которых не проводились. Во-вторых, выбор конкретной модели приводит к смещению данных к этой модели. Например, если предположить, что функция распределения вероятности пикселя является гауссовой, то метод пытается определить ее среднее значение и дисперсию. Он не проверяет релевантность самой модели, даже если распределение представляется бимодальным [39].

- Данный алгоритм предполагает, что количество собранных образцов фона должно быть относительно небольшим [40]. При увеличении числа образцов соответственно увеличивается вычислительная нагрузка и использование памяти. В ViVe сохраняется 20 фоновых образцов для каждого пикселя [41].

- В алгоритме Vibe предусматривается механизм обеспечения пространственной согласованности. При использовании пиксельных методов обычно игнорируются соседние значения пикселей в их модели [42]. Это позволяет обеспечить довольно четкую возможность обнаружения, но не гарантирует, что соседние пиксели будут согласованы. ViVe предлагает новый механизм, который состоит в том, чтобы фоновые значения находились также и в модели соседних пикселей. В процессе обработки

после того заменяется значение модели текущим значением пикселя, политика обновления также распределяет это значение в модель одного из соседних пикселей [43].

- Алгоритм предполагает, что процесс принятия решения принадлежит ли пиксель фону, должен быть прост, так как для подходов на основе пиксельных вычислений вычислительная нагрузка прямо пропорциональна процессу принятия решения.

Классификация пикселей

Большинство методов вычитания фона основано на функции плотности вероятности (ФПВ) либо на статистических параметрах основного процесса генерации фона [44]. Однако выбор конкретной формы ФПВ неизбежно приводит к смещению данных к этой модели. Но на самом деле нет необходимости вычислять ФПВ, пока не произведена соответствующая фоновая сегментация. Альтернативой данному подходу является повышение с течением времени статистической значимости, и одним из способов является построение модели с реальными наблюдаемыми значениями пикселей. Основопологающее предположение состоит в том, что данная процедура является эффективной со стохастической точки зрения, поскольку наблюдаемые значения должны иметь более высокую вероятность повторного наблюдения, по сравнению со значениями, которые еще не использовались [45].

При рассмотрении процедуры вычитания фона как проблемы классификации, то можно классифицировать новое значение пикселя относительно ранее наблюдаемых значений, сравнивая их. Важным отличием алгоритма Vibe от других является следующее свойство: когда новое значение пикселя сравнивается с фоновыми выборками, оно должно быть близко к нескольким выборкам, которые классифицируются как фоновые [46].

Предполагается, что $v(p)$ – значение признака, например, компоненты RGB, некоторого пикселя, расположенного в точке p на изображении, и v_i -

фоновая функция с индексом i . Каждый фоновый пиксель в точке p затем моделируется набором из N фоновых выборок, взятых в предыдущих кадрах.

$$M(p) = \{v_1, v_2, \dots, v_N\} \quad (2.20)$$

Для классификации нового значения функции пикселя $v(p)$ производится его сравнение со всеми значениями, содержащимися в модели $\mathcal{M}(p)$ относительно функции расстояния d и порога T [47].

```

1  int width; // width of the image
2  int height; // height of the image
3  byte image[width*height]; // current image
4  byte segmentationMap[width*height]; // classification result
5
6  int numberOfSamples = 20; // number of samples per pixel
7  int requiredMatches = 2; // #_min
8  int distanceThreshold = 20;
9
10 byte samples[width*height][numberOfSamples]; // background model
11
12 for (int p = 0; p < width*height; p++) {
13     int count=0, index=0, distance=0;
14
15     // counts the matches and stops when requiredMatches are found
16     while ( (count < requiredMatches) && (index < numberOfSamples) ) {
17         distance = getDistanceBetween(image[p], samples[p][index]);
18         if (distance < distanceThreshold) {
19             count++; }
20         index++;
21     }
22
23     // pixel classification
24     if (count < requiredMatches)
25         segmentationMap[p] = FOREGROUND_LABEL;
26     else
27         segmentationMap[p] = BACKGROUND_LABEL;
28 }

```

Рисунок 2.1 – Алгоритм Vibe

Чаще всего в качестве расстояния d используют евклидово расстояние, но также может быть применена любая метрика, которая измеряет совпадение между двумя значениями. В работе [48] проведено сравнение нескольких показателей для изображений RGB. Результаты показали следующее: четыре из шести тестируемых показателей, включая общее

евклидово расстояние, дают одинаковые результаты классификации. Во многих системах видеонаблюдения показатель $v(p)$ является либо яркостью, либо компонентами RGB. Для яркости значение расстояния d уменьшается до абсолютной разницы между значением пикселя и значением образца фона. Важно отметить, что для каждого пикселя процесс классификации останавливается при обнаружении совпадений; однако при непараметрических методах нет необходимости производить сравнения показателя $v(p)$ с каждым образцом [49].

Обновление

В алгоритме Vibe при классификации происходит сравнение текущего значения функции пикселя в момент времени t $v^t(p)$ непосредственно с образцами, содержащимися в фоновой модели $\mathcal{M}^{t-1}(p)$, построенной в момент времени $t-1$.

Однако на видео-данных могут появляться такие ситуации, как внезапное изменение освещения, присутствие новых статических объектов в сцене или изменение фона. Они могут быть решены только в том случае, если процесс обновления включает механизмы, способные справиться с динамическими изменениями в сцене. Чаще используется следующий подход к обновлению: производится замена старых значений после нескольких кадров или через определенный период времени (обычно около нескольких секунд). Несмотря на универсальное применение, данный метод не является оптимальным. В алгоритме Vibe предлагается использование другого метода, основанного на случайных заменах, которые с течением времени улучшают производительность всего алгоритма.

В Vibe разработан новый способ обновления, который включает в себя три важных механизма [50]:

- политика обновления без памяти, согласно которой обеспечивается плавное удаление устаревших значений, хранящихся в моделях фоновых пикселей;

- используется случайная временную подвыборка для расширения временных окон, охватываемых моделями фоновых пикселей;
- механизм, распространяющий выборки фоновых пикселей для обеспечения пространственной согласованности и позволяющий адаптировать модели фонового пикселя, которые маскируются на переднем плане.

Политика обновления без памяти

Многие методы, основанные на выборках, используют политику FIFO (first input, first output) для обновления моделей фона.

В алгоритме Vibe заменяются значения, выбранные случайным образом, в соответствии с равномерной функцией плотности вероятности. Если одни и те же видео-данные обрабатывается несколько раз, все результаты будут практически идентичными.

Алгоритм обновления Vibe приведен на рисунке 2.2. Ядро предлагаемой политики обновления описывается в строках 18-19: один образец модели выбирается случайным образом и заменяется текущим значением [47]. Следует отметить, что данный процесс применим к скалярному значению, к вектору цвета RGB или даже более сложным векторам объектов.

```

1  int width;           // width of the image
2  int height;        // height of the image
3  byte image[width*height]; // current image
4  byte updatingMask[*height]; // updating mask (1==updating allowed)
5
6  int subsamplingFactor = 16; // amount of random subsampling
7
8  int numberOfSamples = 20; // number of samples per pixel
9  byte samples[width*height][numberOfSamples]; // background model
10
11 for (int p = 0; p < width*height; p++)
12     if (updatingMask[p] == 1) { // updating is allowed
13
14         // eventually updates the model of p (in-place updating)
15         int randomNumber = getRandomIntegerIn(1, subsamplingFactor);
16         if (randomNumber == 1) { // random subsampling
17             // randomly selects a sample in the model to be replaced
18             int randomNumber = getRandomIntegerIn(0, numberOfSamples - 1);
19             samples[p][randomNumber] = image[p];
20         }
21
22         // eventually diffuses in a neighboring model (spatial diffusion)
23         int randomNumber = getRandomIntegerIn(1, subsamplingFactor);
24         if (randomNumber == 1) { // random subsampling
25             // chooses a neighboring pixel randomly
26             q = getPixelLocationFromTheNeighborhoodOf(p);
27             // diffuses the current value in the model of q
28             // (uncomment the following check to inhibit diffusion across the
29             // border of the updating mask)
30             // if (updatingMask[q] == 1) {
31                 int randomNumber = getRandomIntegerIn(0, numberOfSamples - 1);
32                 samples[q][randomNumber] = image[p];
33             // }
34         }
35     } // end of "if"
36 } // end of "for"

```

Рисунок 2.2 – Алгоритм Vibe

Процесс случайного обновления противоречит убеждению, что метод вычитания фона должен быть полностью детерминированным и предсказуемым. Потому как, нет причин отдавать предпочтение методу обновления, который сначала производит замену наиболее устаревшего образца, так как это резко сокращает временное окно. Кроме того, может случиться так, что динамический фон будет наблюдаться на более длинном участке, чем значения временного окна. Из-за этого модель не сможет охарактеризовать данный фон. При обновляющей политике алгоритма Vibe

прошлое не «заказывается»; можно сказать, что прошлое не влияет на будущее. Это свойство называется свойством без памяти.

В различных методах рассматривались пиксели по отдельности. Однако политика случайного обновления ввела пространственную неоднородность. Из-за этого в Vibe рассматривается пространственная окрестность пикселя, а не только сам пиксель. Для этого используются два условия [50]:

- 1) все пиксели обрабатываются по-разному,
- 2) диффузная информация локальна.

Случайный выбор индекса, который определяет, какие значения следует заменить, является первым методом обработки пикселей.

Временная подвыборка

Использование процедуры случайного замещения позволяет при сохранении образцов фона охватывать большое, теоретически бесконечное временное окно с ограниченным количеством выборок. Чтобы еще больше расширить размер окна времени, можно воспользоваться случайной выборкой времени. Основная идея заключается в том, что фон представляет собой медленную переменную информацию, за исключением случаев внезапного изменения глобального освещения или сокращения сцены, которые требуют правильной обработки на уровне кадра. Поэтому нет необходимости обновлять каждую фоновую модель для каждого нового кадра. При уменьшении частоты обновления фона срок службы фоновых выборок существенно увеличивается и обеспечивается пространственная неоднородность, поскольку решение об обновлении или отсутствии зависит от пикселя. Как и при наличии периодических или псевдопериодических фоновых движений, использование фиксированных интервалов подвыборки может препятствовать надлежащей адаптации фоновой модели к этим движениям, поэтому используется случайная политика подвыборки [51].

В большинстве случаев используется коэффициент поддискретизации во времени, обозначенный, как φ . Это значение берется равным 16: значение

фонового пикселя имеет 1 шанс в 16 выбранных для обновления своей пиксельной модели. Для некоторых конкретных сценариев можно настроить этот параметр, чтобы отрегулировать длину окна времени, охватываемого моделью пикселя. Например, меньший коэффициент поддискретизации, должен оказаться более эффективным при первых кадрах видеопоследовательности, когда на изображении много движения или при дрожании камеры. Это позволяет ускорить обновление фоновой модели.

- XCS-LBP

Алгоритм вычитания фона может столкнуться с несколькими сложными ситуациями, такими как изменения освещения, динамический фон, плохая погода, дрожание камеры, шум и тени. Для устранения этих ситуаций были разработаны несколько методов извлечения признаков. Наиболее широко используются цветовые особенности, но при наличии изменений освещения, теней их эффективность ограничена. В последнее время большое количество локальных текстурных дескрипторов привлекло большое внимание в задаче фонового моделирования. Среди них локальный бинарный шаблон (LBP), так как вычисления осуществляются просто и быстро [52].

LBP представляет собой описание окрестности пикселя изображения в двоичном представлении. Базовый оператор LBP, применяемый к пикселю изображения, использует восемь пикселей окрестности, принимая значение интенсивности центрального пикселя в качестве порога (рисунок 2.1). Пиксели со значением интенсивности большим или равным значению интенсивности центрального пикселя принимают значения равные «1», остальные принимают значения равные «0». Таким образом, результатом применения базового оператора LBP к пикселю изображения является восьмиразрядный бинарный код, который описывает окрестность этого пикселя [53].

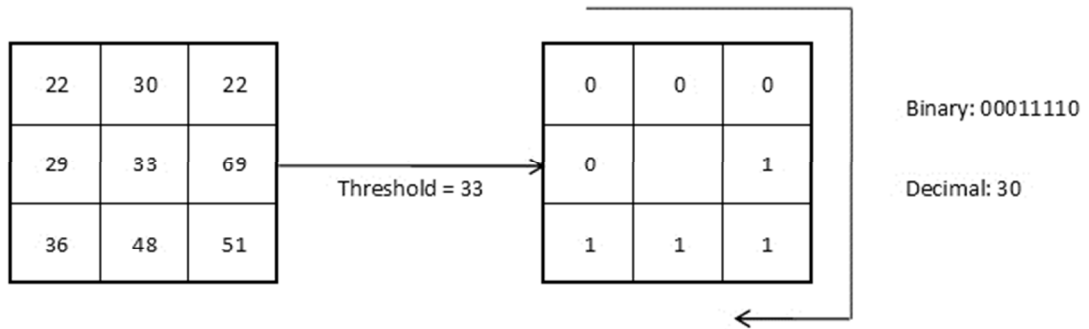


Рисунок 2.3 — Базовый оператор LBP

LBP кодирует локальные примитивы, такие как изогнутые края, пятна, плоские области и т. д. В контексте вычитания фона как текущее изображение, так и изображение, представляющее фоновую модель, кодируются таким образом, что они становятся текстурным представлением сцены. Пусть пиксель, рассматриваемый как центральный пиксель $c = (x_c; y_c)$ локальной окрестности, состоящий из P равномерно расположенных пикселей по кругу радиуса R . Оператор LBP, примененный к c , может быть выражен как:

$$LBP_{P,R}(c) = \sum_{i=0}^{P-1} s(g_i - g_c) 2^i \quad (2.21)$$

где g_c - значение яркости центрального пикселя c ,

g_i - значение яркости для каждого соседнего пикселя,

s - функция порога, определяемая как:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

Из (1) легко показать, что число двоичных членов, подлежащих суммированию, $\sum_{i=0}^{P-1} 2^i = 2^P - 1$, так что длина полученной гистограммы равна 2^P . Основная идея CS-LBP [54] заключается в сравнении уровней яркости пар пикселей в центрированных симметричных направлениях, а не

сравнения центрального пикселя с его соседями. Предполагая четное число P соседних пикселей, оператор CS-LBP задается:

$$CS - LBP_{P,R}(c) = \sum_{i=0}^{(P/2)-1} s(g_i - g_{i+(P/2)})2^i \quad (2.23)$$

где g_i и $g_{i+(P/2)}$ – значения яркости центров симметричных пар пикселей, s – функция порога, определяемая как:

$$s(x) = \begin{cases} 1 & \text{if } x > T \\ 0 & \text{otherwise} \end{cases} \quad (2.24)$$

где T – пользовательский порог.

Так как уровни яркости нормированы в пределах $[0,1]$, рекомендуется использовать небольшое значение.

Также существует расширенный оператор CS-LBP, который сравнивая яркости пар центров симметричных пикселей так, чтобы полученная гистограмма была короткой, но с учетом центрального пикселя. Данная комбинация делает полученный дескриптор менее чувствительным к шуму. Новый вариант LBP, называемый XCS-LBP [55] (eXtended CS-LBP), выражается как:

$$XCS - LBP_{P,R}(c) = \sum_{i=0}^{(P/2)-1} s(g_1(i,c) - g_2(i,c))2^i \quad (2.25)$$

Здесь пороговая функция s , которая используется для определения типов локального перехода шаблона, определяется как характеристическая функция:

$$s(x_1 + x_2) = \begin{cases} 1 & \text{if } (x_1 + x_2) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.26)$$

В данной формуле $g_1(i; c)$ и $g_2(i; c)$ определяются следующим образом:

$$\begin{cases} g_1(i, c) = (g_i - g_{i+(P/2)}) + g_c \\ g_2(i, c) = (g_i - g_c)(g_{i+(P/2)} - g_c) \end{cases} \quad (2.27)$$

с теми же обозначениями, что и в уравнениях (2.21) и (2.23).

Стоит отметить, что пороговая функция не нуждается в пользовательском пороговом значении, в отличие от CS-LBP.

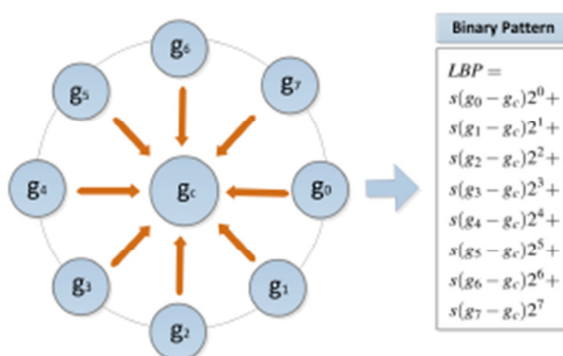


Рисунок 2.4 – LBP дескриптор

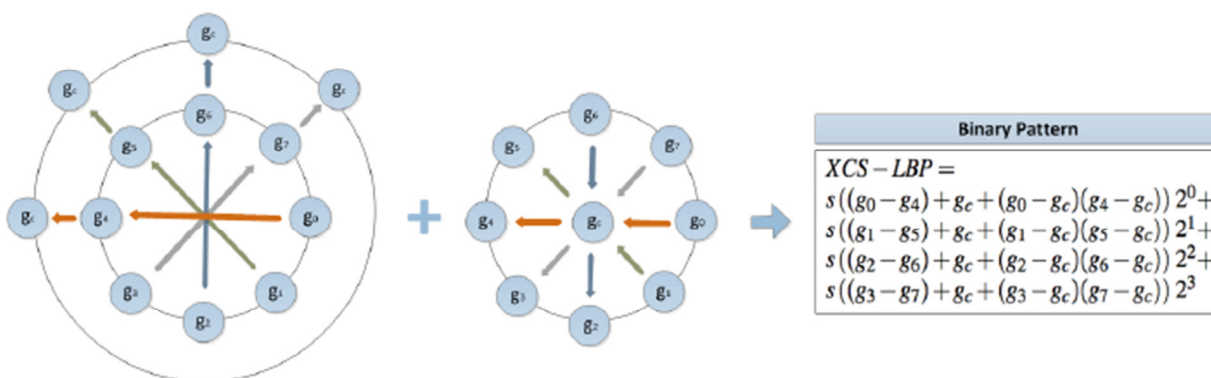


Рисунок 2.5 – XCS-LBP дескриптор

Вычисление исходного LBP для окрестности размера $P=8$ проиллюстрировано на рисунке 2, и вычисление предлагаемого XCS-LBP показано на рисунке 3. Обратите внимание на соответствующие длины кода 8 и 4, которые приводят к соответствующим сжатиям изображения.

Предлагаемый XCS-LBP создает более короткую гистограмму, чем LBP, но он извлекает больше деталей изображения, чем CS-LBP, так как он учитывает значение яркости центрального пикселя и опирается на новую стратегию сравнения соседних пикселей [56]. Поскольку он также более устойчив к шуму, чем LBP и CS-LBP, предлагаемый дескриптор более эффективен для фонового моделирования и вычитания.

2.3 Методы обнаружения объектов, основанные на оценке оптического потока

Оптическим потоком является изображение видимого движения объекта, краев сцены или поверхностей, которое возникает в результате перемещения наблюдателя (глаз или камеры) относительно сцены. Алгоритмы, в основе которых лежит оптический поток (регистрация движения, сегментация объектов) используют данные движения объектов, краев, а также поверхностей. Оптический поток определяет поле скоростей, так как сдвиг точки между двумя изображениями можно назвать мгновенной скоростью. Отсюда следует, что при осуществлении операции вычисления оптического потока находятся вектора (скорости) для оценки движения объекта [57].

Использование данного подхода предполагает наличие следующих условий:

- яркость каждой точки объекта остается постоянной с течением времени;
- ближайшие точки, принадлежащие одному объекту, двигаются с одинаковой, или близкой к таковой, скоростью.

2.3.1 Оценка оптического потока

Методы обнаружения подвижных объектов, в основе которых лежит вычисление оптического потока, определяют движение между двумя кадрами в каждом пикселе. Кадры берутся в момент времени t и $t + \delta t$. Эти методы относятся к дифференциальным, так как они основаны на приближении сигнала отрезком ряда Тейлора. Поэтому данные методы вычисляют частные производные по времени и пространственным координатам.

В случае размерности $2D+t$ (при большей размерности аналогично) пиксель в позиции (x, y, t) с интенсивностью $I(x, y, t)$ за один кадр будет перемещен на δx , δy и δt . Вследствие того, что интенсивность точки не изменяется, получается:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (2.28)$$

Так как перемещение мало, при помощи ряд Тейлора выводится уравнение (2.29):

$$I(x + \delta x, y + \delta y, t + \delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t \quad (2.29)$$

Из данных равенств следует:

$$\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0 \quad (2.30)$$

или

$$\frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \frac{\delta t}{\delta t} = 0 \quad (2.31)$$

Исходя из равенства (2.31) можно получить:

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0 \quad (2.32)$$

где V_x, V_y — компоненты скорости оптического потока в $I(x, y, t)$,

$\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \frac{\partial I}{\partial t}$ — производные изображения в (x, y, t) в соответствующих

направлениях.

Таким образом:

$$I_x V_x + I_y V_y = -I_t \quad (2.33)$$

или

$$\nabla I^T \cdot \bar{V} = -I_t \quad (2.34)$$

Уравнение (2.34) – это уравнение оптического потока [58]. Данное равенство содержит две неизвестных V_x, V_y и не может быть однозначно разрешено. Следовательно, необходимо ввести дополнительное предположение. Например, пусть оптический поток меняется плавно от кадра к кадру, т. е. для всех пикселей p из окрестности (x, y) оптического потока смещение будет постоянным.

2.3.2 Методы определения оптического потока

Существует множество разных методов определения оптического потока:

- Блочные методы — поиск местоположения определенных областей (блоков) текущего кадра на предыдущем кадре.
- Фазовая корреляция — инверсия нормализованного перекрестного спектра.
- Дифференциальные методы оценки оптического потока, базирующиеся на частных производных сигнала:
 1. Алгоритм Лукаса — Канаде
 2. Horn–Schunck
 3. Buxton–Buxton

Метод Лукаса–Канаде.

Алгоритм Лукаса-Канаде является одним из методов, применяющихся для вычисления оптического потока. Данный алгоритм используется достаточно широко в задачах оценки движения объекта. Метод Лукаса-Канаде обрабатывает пиксели в окрестности конкретной точки, поэтому его называют локальным подходом вычисления оптического потока [59].

Алгоритм предполагает, что:

1. Смещение точек на двух кадрах видеопоследовательности незначительное.
2. Смещение точек в окрестности некоторой точки p одинаково.

Существует предположение: $D = \{q_1, q_2, \dots, q_n\}$ — набор точек в окрестности точки p . Тогда, можно записать систему уравнений

$$\begin{cases} I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2) \\ \dots \\ I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n) \end{cases} \quad (2.35)$$

где q_1, q_2, \dots, q_n — точки внутри окна поиска,

$I_x(q_i), I_y(q_i), I_t(q_i)$ - частные производные изображения I по x, y и времени t , вычисляемые в точке q_i в текущий момент времени.

Эти выражения можно переписать в матричной форме $Av=b$, где

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \quad (2.36)$$

Эта система уравнений обычно сильно избыточна, поэтому обычно ее решают методом наименьших квадратов. В результате получается

$$A^T Av = A^T b \quad (2.37)$$

Из уравнения 2.37 можно вывести

$$v = (A^T A)^{-1} A^T b, \quad (2.38)$$

где A^T - транспонированная матрица A .

Выражение 2.38 можно записать в матричном виде

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (2.39)$$

В основном набор точек в окрестности точки p ограничен окном, размер которого $N \times M$. Основная проблема заключается в прямой

зависимости между количеством уравнений в системе (2.40) и числом точек в окрестности определенной точки. То есть, чем больше количества точек в окрестности точки, тем больше уравнений будет содержать система.

Для решения проблемы используется взвешенный метод наименьших квадратов. При определении весовых коэффициентов пикселей на изображении применяются функция $W(x, y)$ [60]. Исходя из данного метода, для нахождения решения необходимо минимизировать невязку:

$$\begin{aligned} E(V) &= \sum_{x,y \in D} W(x, y) [I(x, y, t) - I(x + \delta x, y + \delta y, t + \delta t)]^2 = \\ &= \sum_{x,y \in D} W(x, y) \left(\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} \right)^2 \end{aligned} \quad (2.40)$$

где $V = (V_x, V_y)$

$W(x, y)$ - весовые коэффициенты, присвоенные пикселям q_i .

Чтобы найти минимум $E(V)$ применяют метод наименьших квадратов, с нахождением её частных производных по V_x и V_y :

$$\begin{aligned} \frac{\partial E(V)}{\partial V_x} &= 2 \sum_{x,y \in D} W(x, y) \left(\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} \right) \frac{\partial I}{\partial x} = \\ &= 2 \sum_{x,y \in D} W(x, y) \left[\left(\frac{\partial I}{\partial x} \right)^2 V_x + \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} V_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right] \end{aligned} \quad (2.41)$$

$$\begin{aligned} \frac{\partial E(V)}{\partial V_y} &= 2 \sum_{x,y \in D} W(x, y) \left(\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} \right) \frac{\partial I}{\partial y} = \\ &= 2 \sum_{x,y \in D} W(x, y) \left[\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} V_x + \left(\frac{\partial I}{\partial y} \right)^2 V_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right] \end{aligned} \quad (2.42)$$

Данные уравнения можно переписать в более компактной форме и приравняем к нулю:

$$\begin{cases} \sum_{x,y \in D} W(x,y) \left[\left(\frac{\partial I}{\partial x} \right)^2 V_x + \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} V_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right] = 0 \\ \sum_{x,y \in D} W(x,y) \left[\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} V_x + \left(\frac{\partial I}{\partial y} \right)^2 V_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right] = 0 \end{cases} \quad (2.43)$$

Система уравнений (2.43) может быть представлена в матричной форме:

$$\begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} + \begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right) \end{bmatrix} = 0 \quad (2.44)$$

Пусть

$$A = \begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix} \quad V = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \quad (2.45)$$

$$B = \begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right) \end{bmatrix}$$

Тогда

$$A \cdot V + B = 0 \quad (2.46)$$

ИЛИ

$$V = A^{-1}B \quad (2.47)$$

Выражение 2.47 можно записать как систему уравнений

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right) \end{bmatrix} \quad (2.48)$$

В настоящее время методы вычисления оптического потока становятся все популярнее. Их применяют при определении направления движения объектов; определении расстояния между объектами с помощью анализа оптического потока кадров, полученных с двух камер (стереозрение). С помощью методов оптического потока может быть проведено наиболее аккуратное выделение движущихся объектов, даже в случае перемещения камеры.

Вывод по главе: рассмотренные методы, которые являются наиболее популярными при решении задачи обнаружения движущихся объектов в системах видеоаналитики. Следовательно, требуется их исследование на предмет оценки эффективности работы алгоритмов при появлении на видеопоследовательности различных нестабильностей, таких как условия плохой погоды, нестационарная камера, динамический фон, тень, дрожание камеры.

ГЛАВА 3. МЕТОДЫ СОПРОВОЖДЕНИЯ ОБЪЕКТОВ

Сопровождение (трекинг) подвижных объектов является одной из часто применяемых задач в системах видеонаблюдения, системах слежения, анализа видео и т.д. В качестве входных данных алгоритмов сопровождения может использоваться видеопоследовательность (последовательность изображений) I_1, I_2, \dots, I_N с большим объемом информации, которую необходимо анализировать и обрабатывать.

Задача сопровождения заключается в построении траектории движения подвижных объектов на входной последовательности видео-кадров. Предположим, что *положение объекта* на изображении с номером k обозначается P_k . Тогда *траектория* движения объекта – это последовательность различных положений $P_s, P_{s+1}, \dots, P_{s+l-1}$, где s – номер первого кадра, на котором был обнаружен объект, l – общее количество кадров последовательности, на которых наблюдается данный объект. В зависимости от используемого метода сопровождения положение объекта определяется разными способами (с помощью координат и размеров сторон окаймляющего прямоугольника, координат центра масс контура и т.д.).

3.1 Классификация методов сопровождения

Существуют различные подходы решения задачи сопровождения подвижных объектов [61]. Их подразделяют на 3 группы.

1. *Методы сопровождения особых точек* (point tracking). В данных методах расположение нескольких характерных точек определяет положение объекта. Некоторый объект на последовательных кадрах может быть представлен, как набор соответствующих пар точек. Выделяют также 2 подгруппы:

- Детерминистские методы [62] используют для анализа качественные эвристики движения (небольшое изменение скорости, неизменность расстояния в трехмерном пространстве между парой точек, принадлежащих объекту). В данных методах в целях решения задачи сопровождения объектов проводится минимизация функции соответствия наборов точек. К данной группе методов относятся методы на основе вычисления плотного и разреженного оптического потока, а также методы сопоставления (matching) дескрипторов ключевых точек.

- Вероятностные методы используют так называемое пространство состояний. Считается, что подвижный объект обладает определенным внутренним состоянием, измеряемым на каждом кадре. Для оценки следующего состояния объекта данные методы максимально обобщают все измерения, то есть определяют новое состояние, когда уже существует набор измерений состояний на предыдущих кадрах. К данной группе методов можно отнести методы на базе фильтра Кальмана [63-64] и фильтра частиц (particle filter) [65, 66].

2. *Методы сопровождения компонент (kernel tracking)*. В данном случае компонентой считается форма объекта. В простейшем случае за компоненту принимается двумерный шаблон: прямоугольник или овал. В более сложных случаях шаблон – это трехмерная модель объекта, которая спроецирована на плоскость изображения. В общих случаях методы этой группы используются в случаях определения движения обычным смещением, поворотом или аффинным преобразованием. Трекингом компонент является итеративная процедура локализации, которая основана на максимизации определенного критерия подобия. На практике данная процедура реализуется с помощью сдвига среднего (mean shift) [67] и его непрерывной модификации (Continuous Adaptive Mean Shift, CAM Shift) [68].

3. *Методы сопровождения силуэта (silhouette tracking)*. Силуэт может быть представлен как контур или набор простых связанных геометрических примитивов. Задача трекеров силуэта состоит в определении некоторой

области на каждом кадре, в которой находится объект. Это осуществляется с применением модели силуэта объекта, которая построена на основании предыдущих кадров. Выделяют две подгруппы данных методов: методы сопоставления и сопровождения фрагментов изображения, содержащих объект; и методы сопровождения контура.

3.2 Методы сопровождения особых точек

На практике часто используются детерминистские методы сопровождения особых точек. Наличие определенных выделенных точек, представляющих объект наиболее точно, является главным условием использования данных методов. Для выделения данных точек используются специальные дескрипторы и детекторы. Когда детектор обнаруживает данную точку, с помощью дескриптора строится вектор признаков, характерных именно для этой точки.

Процедура сопровождения особых точек может состоять из следующих этапов [69]:

1. Используемый детектор (LoG, SURF, SIFT, и т.д.) находит особые точки на предыдущем и текущем кадрах.
2. Определяются дескрипторы для найденных точек – n -мерных векторов-описателей.
3. Полученные на текущем и предыдущем кадре дескрипторы сопоставляются друг с другом с помощью вычисления некоторой метрики (норма L_2 , кросс-корреляция и т.д.).

Применение фильтра Калмана для оценки положения объекта

В теории управления задачу сопровождения следует рассматривать как оценку состояния системы на основании последовательности зашумленных измерений [70]. Примем, что существует модель объекта, которая

наблюдается в зашумленном пространстве кадра. В данном случае x – модель объекта, z – наблюдение объекта. В общем случае x, z – вектора признаков, которые могут быть разной размерности. При сопровождении объекта данные признаки могут использоваться двумя способами:

1. Последовательность наблюдений z_1, z_2, \dots используется в целях уточнения базовой модели x . Однако, модель может изменяться с течением времени, поэтому новое наблюдение z_k может представлять новую оценку модели x_k .

2. Полученная оценка модели x_k применяется в целях предсказания модели x_{k+1} и наблюдения z_{k+1} .

В результате имеем механизм обратной связи: наблюдаем z_k , оцениваем x_k , предсказываем x_{k+1} и z_{k+1} , обновляем x_{k+1} на основании z_{k+1} . Фильтры Калмана и фильтры частиц основаны на данном механизме.

Фильтр Калмана используется в работе с дискретизированными по времени линейными динамическими системами. Данные системы моделируются с помощью цепей Маркова с использованием линейных операторов и слагаемых с нормальным распределением. В каждый дискретный момент времени линейный оператор воздействует на состояние, при этом переводя его в другое состояние посредством добавления некоторой случайной величины в виде нормального шума и вектора управления, который моделирует воздействие управляющего сигнала.

В данном случае модель обратной связи описывается следующими формулами:

$$x_{k+1} = F_k x_k + w_k \quad (3.1)$$

$$z_k = H_k x_k + v_k \quad (3.2)$$

где F_k – матрица преобразования состояния системы;

w_k — белый шум с нормальным распределением $N(0, Q_k)$, математическим ожиданием, равным 0, и матрицей ковариации Q_k ;

H_k — матрица связи модели и наблюдения;

v_k — белый шум с нормальным распределением $N(0, R_k)$.

В основе алгоритма лежат две повторяющиеся фазы: фаза предсказания и фаза коррекции.

При работе первой фазы происходит предсказание (экстраполяция) значений переменных состояния с использованием оценки состояния предыдущего шага и их неопределенности. Эта оценка также называется априорной, так как она дается до выполнения измерений и основывается только на математической модели.

На данном шаге вычисляется априорная оценка состояния \tilde{x}_k^- и наблюдения z_k .

$$\tilde{x}_k^- = F_{k-1}x_{k-1} + w_{k-1} \quad (3.3)$$

$$z_k = H_k \tilde{x}_k^- + v_k \quad (3.4)$$

Во второй фазе уточняются результаты экстраполяции при помощи некоторых измерений. В данном случае возможна некоторая погрешность. Эта оценка называется апостериорной.

Определяется матричный коэффициент Кальмана K_k и строится апостериорная оценка состояния \tilde{x}_k^+ .

$$K_k = P_k^- H_k^T (R_k + H_k P_k^- H_k^T)^{-1} \quad (3.5)$$

$$\tilde{x}_k^+ = \tilde{x}_k^- + K_k(z_k - H_k \tilde{x}_k^-) \quad (3.6)$$

где P_k^- - предсказанная матрица ковариации ошибок измерения состояния.

$$P_k^- = F_k P_{k-1}^+ F_k^T + Q_{k-1} \quad (3.7)$$

где

$$P_k^+ = (E - K_k H_k) P_k^- \quad (3.8)$$

При классической работе алгоритма данные фазы чередуются: предсказание осуществляется на основе результатов корректировки с прошлой итерации, а при корректировке уточняются результаты фазы экстраполяции. Но иногда фазу коррекции можно пропустить и предсказание будет происходить на основе не уточненной оценки. Такая ситуация может возникнуть если информация с измерительных датчиков на данном шаге не доступна.

Применение фильтра частиц к задаче сопровождения

Для получения более точной оценки состояния системы необходимо уйти от предположения, что шум имеет Гауссово распределение. В данном случае предполагается наличие мультимодального распределения шума. Тогда для моделирования таких систем используются фильтры частиц. *Фильтры частиц* являются более общим подходом к сопровождению подвижных объектов с использованием вероятностных методов [71, 72].

Базовым алгоритмом фильтрации частиц является алгоритм *воспроизведения условной плотности* (CONDitional DENSity propagATIOn, CONDENSATION) [72]. Он является основой для большинства алгоритмов данной группы, используемых в компьютерном зрении.

Рассчитывается плотность распределения вероятности исходя из предположения, что система может находиться в состояниях

$X_t = \{x_1, x_2, \dots, x_t\}$, в момент времени t . Последовательность наблюдений обозначается $Z_t = \{z_1, z_2, \dots, z_t\}$. Также выполняется условие Марковской цепи - состояние x_t зависит только от предыдущего состояния x_{t-1} . То есть получаем систему с независимым набором наблюдений. При фильтрации частиц распределение вероятности представляется, как набор взвешенных выборок – частиц, появление которых регулируется посредством введения весов. Функция плотности вероятности для состояния x_t при заданном наборе наблюдений Z_t определяется посредством множества S_t . Оно задает приближенное распределение $p(x_t|Z_t)$

$$S_t = \left\{ (s_i^t \pi_i^t, i = \overline{1, N}, \sum_{i=1}^N \pi_i^t = 1) \right\} \quad (3.9)$$

Основной задачей является построение метода восстановления множества S_t на основании S_{t-1} . Данный алгоритм можно представить как последовательность этапов [71, 72]:

1. Построена коллекция взвешенных выборок в момент времени $t - 1$:

$$S_{t-1} = \left\{ (s_i^{t-1} \pi_i^{t-1}, i = \overline{1, N}, \sum_{i=1}^N \pi_i^{t-1} = 1) \right\} \quad (3.10)$$

Дополнительно определяются интегральные веса по формуле:

$$c_i = c_{i-1} + \pi_i^{t-1}, i = \overline{1, N}, c_0 = 0 \quad (3.11)$$

2. Определяется n -ый экземпляр выборки S_t . Для этого случайно выбирается число r из отрезка $[0,1]$, и вычисляется $j = \arg \min_i \{c_i > r\}$. Далее получается текущая оценка состояния s_j^{t-1} .

3. Выполняется *предсказание* следующего состояния. Предсказывается аналогично фильтру Кальмана, однако нет ограничений, связанных с линейностью системы и видом распределения шума.

$$s_n^t = F_{t-1}s_j^{t-1} + w_{t-1} \quad (3.12)$$

4. Выполняется *коррекция*. Вычисляется вес полученного экземпляра с помощью текущего наблюдения z_t и его распределения по формуле:

$$\pi_n^t = p(z_t | x_t = s_n^t) \quad (3.13)$$

5. Строится множество частиц S_t , за счет повтора шагов 2 – 4 N -раз.

6. Нормализуется последовательность весов π_i^t так, чтобы $\sum_{i=1}^N \pi_i^t = 1$.

7. Вычисляется наилучшая оценка для состояния x_t (линейная свертка полученного набора экземпляров выборки). То есть фактически определяется некоторая средняя частица.

$$x_t = \sum_{i=1}^N \pi_i^t s_i^t \quad (3.14)$$

К преимуществам подхода можно отнести высокую вычислительную эффективность.

ГЛАВА 4. РАЗРАБОТКА АЛГОРИТМОВ ОБНАРУЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ

Для разработки были выбраны следующие алгоритмы:

- метод вычитания фона;
- Vibe;
- метод вычитания фона с использованием смеси Гауссовых распределений (GMM);
- метод оптического потока (алгоритм Лукаса-Канаде).

Алгоритм вычитания фона:

Разработан алгоритм вычитания фона, словесное описание которого состоит в следующем.

1. Ввод видео-данных
2. Первый кадр видео принимается за фоновое изображение без движущегося объекта;
3. Исследование следующего кадра;
4. Находится разность текущего кадра видео и фонового изображения

$$D_k(x, y) = \text{abs}(I_k(x, y) - F(x, y)), k = \overline{1, N}, \quad (4.1)$$

где $I_k(x, y)$ - интенсивность определенного кадра.

5. Определяется истинность выражения (больше ли полученная разность выбранного порога):

$$D_k(x, y) > \tau \quad (4.2)$$

- если истинно, то есть $(D_k(x, y) > \tau)$, то пиксель определяется, как принадлежащий объекту

- если ложно, то есть $(D_k(x,y) < \tau)$, то пиксель определяется, как принадлежащий фону

6. Конец

Блок-схема алгоритма

Блок-схема алгоритма вычитания фона, приведена на рис. 4.1.



Рисунок 4.1 – Блок-схема алгоритма вычитания фона

Алгоритм Vibe:

Разработан алгоритм вычитания фона Vibe, словесное описание которого состоит в следующем.

1. Ввод видео-данных
2. Создается модель из N различных значений, которая соответствует каждому пикселю

$$M(p) = \{v_1, v_2, \dots, v_N\} \quad (4.3)$$

3. Начальная инициализация

Модель каждого пикселя заполняется значениями случайных пикселей из окрестности размером 3x3.

4. Вычитание фона

Сравнивается новое значение пикселя со всеми значениями в модели фона для данного пикселя.

$$count > T \quad (4.3)$$

Если количество совпадений (*count*) больше некоторого значения (порога *T*), то пиксель считается фоновым.

5. Обновление модели

Если пиксель принадлежит фону, то заменяем одно значение в модели данного пикселя и одного из соседних на рандомное значение.

Блок-схема алгоритма

Блок-схема алгоритма вычитания фона Vibe приведена на рис. 4.2.

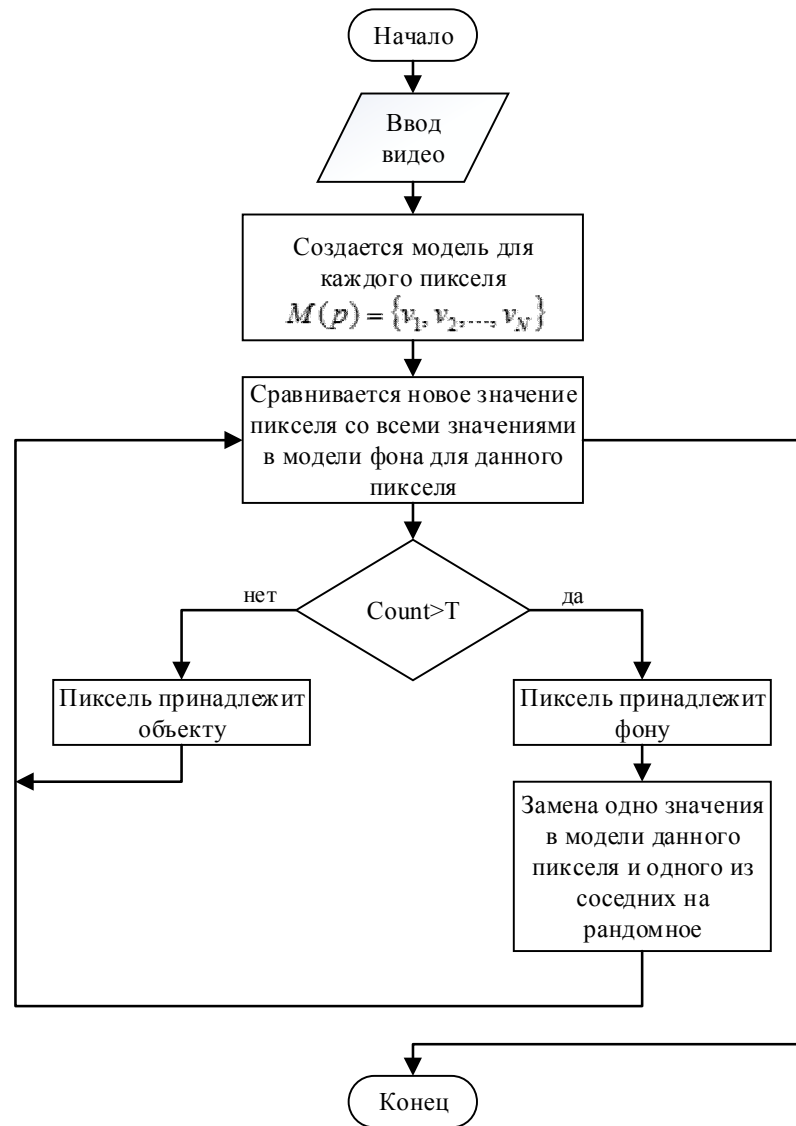


Рисунок 4.2 – Блок-схема алгоритма Vibe

Алгоритм метода вычитания фона с использованием смеси Гауссовых распределений (GMM):

Разработан алгоритм вычитания фона, основанный на смеси Гауссовых распределений, словесное описание которого состоит в следующем.

1. Ввод видео-данных
2. Определяется распределение наблюдаемых значений цвета для каждого пикселя. Новое значение цвета можно представить как одну из основных компонент полученной смеси Гауссовых распределений.
3. Первые B^k распределений относятся к цвету фоновых пикселей, когда выполняется следующее условие

$$B^k = \arg \min_b \left\{ \sum_{j=1}^b \omega_j^k > T \right\} \quad (4.4)$$

где T – пороговое значение, параметр модели.

4. Исследование следующего кадра;

5. Для каждого пикселя изображения выполняется тест, с помощью которого при использовании расстояния Махаланобиса, определяется, какому распределению соответствует полученное значение.

$$\sqrt{(X_{k+1} - \mu_j^k)^T (\Sigma_j^k)^{-1} (X_{k+1} - \mu_j^k)} < 2.5 \sigma_j^k \quad (4.5)$$

6. Проверяется обнаружено ли соответствие полученного значения и распределения Гаусса, соответствующего фону:

- если соответствие обнаружено, то пиксель определяется как фоновый;
- если соответствие не обнаружено, то принимается, что пиксель принадлежит объекту.

7. Проверяется обнаружено ли соответствие распределения Гаусса для цвета текущего пикселя:

- если соответствие обнаружено, то параметры распределений обновляются по следующим формулам:

$$\mu_j^{k+1} = (1 - \rho) \mu_j^k + \rho X_{k+1} \quad (4.6)$$

$$(\sigma_j^{k+1})^2 = (1 - \rho) (\sigma_j^k)^2 + \rho (X_{k+1} - \mu_j^{k+1}) (X_{k+1} - \mu_j^{k+1})^T \quad (4.7)$$

где $\rho = \alpha N(X_{k+1} | \mu_j^k, \Sigma_j^k)$.

- если соответствие не обнаружено, то математическое ожидание уравнивается текущему значению цвета пикселя. Дисперсия $(\sigma_s^{k+1})^2$ выбирается максимально возможной, а вес ω_s^{k+1} - минимально допустимым.

Блок-схема алгоритма

Блок-схема алгоритма вычитания фона, основанного на смеси гауссовых распределений, приведена на рис. 4.3.

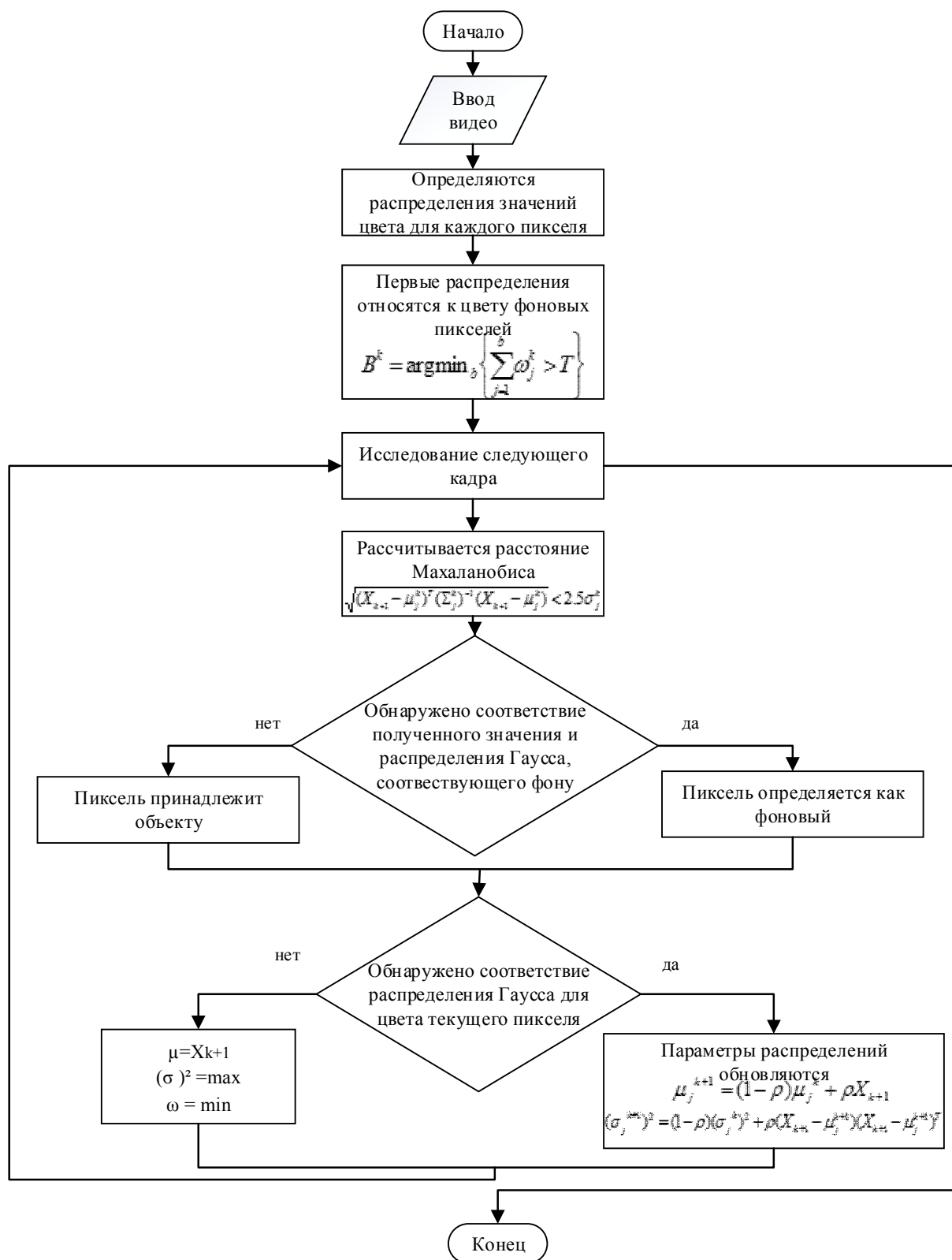


Рисунок 4.3 – Блок-схема алгоритма вычитания фона, основанного на смеси Гауссовых распределений

Алгоритм Лукаса-Канаде:

Разработан алгоритм Лукаса-Канаде для оценки оптического потока, словесное описание которого состоит в следующем.

1. Ввод видео-данных
2. Находятся частные производные изображения I по x, y и времени t , вычисляемые в точке q_i в текущий момент времени
3. Составляется матрица A , состоящая из частных производных изображения I по x, y , вычисляемые в точке q_i

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \quad (4.8)$$

4. Составляется матрица b , состоящая из частных производных изображения I по времени t , вычисляемые в точке q_i

$$b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \quad (4.9)$$

5. Находится вектор движения

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (4.10)$$

Блок-схема алгоритма

Блок-схема алгоритма Лукаса-Канаде для оценки оптического потока, приведена на рис. 4.4.

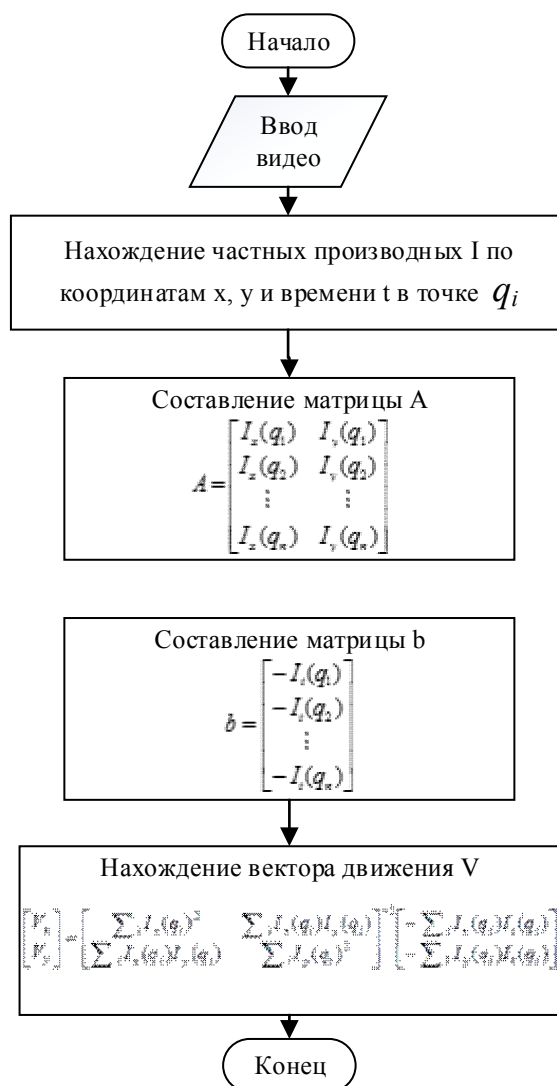


Рисунок 4.4 – Блок-схема алгоритма Лукаса-Канаде

ГЛАВА 5. ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ

При оценке эффективности разработанных алгоритмов обнаружения движущихся объектов интерес представляет оценка показателей качества алгоритма. Можно выделить такие показатели, как полнота и точность, специфичность, процент неправильной классификации и F-мера.

Точность (P) и полнота (R) являются метриками которые используются при оценке большей части алгоритмов извлечения информации. Иногда они используются сами по себе, иногда в качестве базиса для производных метрик, таких как F-мера или R-Precision. Суть точности и полноты очень проста.

Точность системы в пределах класса – это доля истинно положительных срабатываний алгоритма от общего числа срабатываний. Полнота системы – доля истинно положительных срабатываний алгоритма от общего числа событий, которое требовалось обнаружить.

Специфичность (истинно отрицательная пропорция) отражает долю отрицательных результатов, которые правильно идентифицированы как таковые.

F-мера сводит к одному числу такие основополагающие метрики, как точность и полнота. Она представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

5.1 Концептуальные основы экспериментальных исследований

Целью проведения вычислительных экспериментов для разработанных алгоритмов обнаружения движущихся объектов является оценка

эффективности работы алгоритмов при различных условиях видеосъемки.

Эксперимент

Для проведения вычислительного эксперимента используются видеоданные с разными условиями видеосъемки.

Входное видео обрабатывается одним из исследуемых алгоритмов, на выходе которого должны быть определены подвижные объекты.

Далее собирается статистическая информация работы алгоритма.

Вычисляется число истинно-положительных срабатываний (TP), т.е. нет ошибки в обнаружении движущегося объекта, и число истинно-отрицательных срабатываний (TN).

Определяется число ложно-положительных срабатываний (FP) (ошибка I рода) или число ложных тревог (например, обнаружено движение ветвей деревьев) и число ложно-отрицательных срабатываний (FN) (ошибка II рода) подвижный объект был в кадре, но не был обнаружен.

На основе собранных данных вычисляются следующие метрики:

1. Полнота – доля истинно положительных срабатываний алгоритма от общего числа событий, которое требовалось обнаружить:

$$R = \frac{TP}{TP + FN} \quad (5.1)$$

2. Точность – доля истинно положительных срабатываний алгоритма от общего числа срабатываний:

$$P = \frac{TP}{TP + FP} \quad (5.2)$$

3. Специфичность.

$$S = \frac{TN}{TN + FP} \quad (5.3)$$

4. Процент неправильной классификации.

$$PWC = 100 \times \frac{FN + FP}{TP + FN + FP + TN} \quad (5.4)$$

5. F-мера.

$$F1 = \frac{2 \times P \times R}{P + R} \quad (5.5)$$

5.2 Планирование вычислительных экспериментов

В качестве исходных видео-данных будут использоваться видео с различными условиями (рис. 5.1): базовые условия, условия плохой погоды, нестационарная камера, динамический фон, тень, дрожание камеры.



а



б



В



Г



Д



е

Рисунок 5.1 – Исходные кадры видео-данных

- а) базовые условия;**
- б) тень;**
- в) условия плохой погоды;**
- г) динамический фон;**
- д) дрожание камеры;**
- е) нестационарная камера**

Эксперимент.

1. Ввод данных
2. Применяя один из разработанных алгоритмов над входным видео, получить на выходе видео с определенным движущимися объектами.
3. Вычислить число истинно-положительных срабатываний (TP), истинно-отрицательных срабатываний (TN).
4. Определить число ложно-положительных срабатываний (FP) и число ложно-отрицательных срабатываний (FN).

5. На основании полученных данных вычислить значения метрик (5.1), (5.2), (5.3), (5.4) и (5.5).

Повторить описанный эксперимент для всех видео-данных (рис. 5.1), а также для всех разработанных алгоритмов.

5.3 Описание результатов вычислительных экспериментов

Для оценки эффективности работы алгоритмов обнаружения движущихся объектов использовались видео-данные, кадры которых представлены на рисунке 5.1.

Результаты работы алгоритмов (кадры видеопоследовательностей с обнаруженными объектами) представлены в приложении А.

В ходе проведения экспериментов были получены значения метрик для различных алгоритмов при разных условиях съемки. Результаты представлены в таблицах 5.1-5.6.

Таблица 5.1 – Значения метрик различных алгоритмов при базовых условиях

Метрики Методы	R	P	S	PWC	F1
Вычитание фона (BS)	0.6028	0.5971	0.8873	3.4482	0.5999
Vibe	0.7628	0.6530	0.9868	3.0410	0.7036
GMM	0.7535	0.6028	0.9687	3.3599	0.6698
Алгоритм Лукаса-Канаде (LK)	0.6289	0.7291	0.9567	3.3647	0.6753

Таблица 5.2 – Значения метрик различных алгоритмов при появлении тени

Метрики Методы	R	P	S	PWC	F1
Вычитание фона (BS)	0.5680	0.5531	0.8432	4.4238	0.5605
Vibe	0.6147	0.6387	0.9273	4.3089	0.6265
GMM	0.5983	0.5987	0.8723	4.3297	0.5985
Алгоритм Лукаса-Канаде (LK)	0.5828	0.6128	0.8127	4.3597	0.5974

Таблица 5.3 – Значения метрик различных алгоритмов в условиях плохой погоды

Метрики Методы	R	P	S	PWC	F1
Вычитание фона (BS)	0.4289	0.5077	0.8322	4.3547	0.4650
Vibe	0.5385	0.5939	0.9187	4.1897	0.5648
GMM	0.5522	0.5922	0.8510	4.6847	0.5715
Алгоритм Лукаса-Канаде (LK)	0.5229	0.7034	0.8016	4.6049	0.5998

Таблица 5.4 – Значения метрик различных алгоритмов при динамическом фоне

Метрики Методы	R	P	S	PWC	F1
Вычитание фона (BS)	0.4692	0.5847	0.8694	4.2971	0.5206
Vibe	0.5291	0.6292	0.9833	3.3392	0.5748
GMM	0.7322	0.5786	0.9235	3.5527	0.6464
Алгоритм Лукаса-Канаде (LK)	0.6148	0.6428	0.8869	4.1287	0.6284

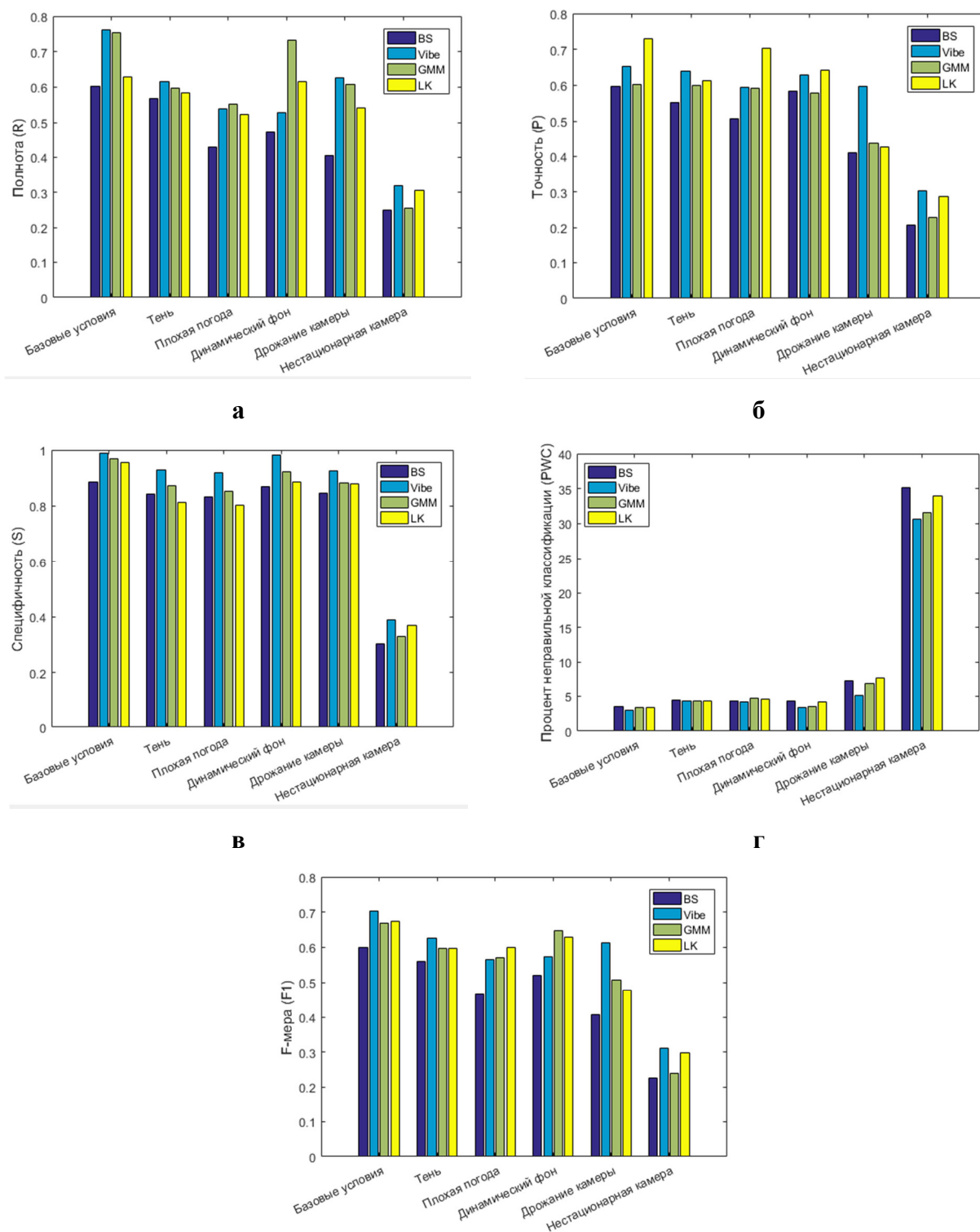
Таблица 5.5 – Значения метрик различных алгоритмов при дрожании камеры

Метрики Методы	R	P	S	PWC	F1
Вычитание фона (BS)	0.4028	0.4087	0.8469	7.3347	0.4057
Vibe	0.6274	0.5982	0.9266	5.1073	0.6125
GMM	0.6079	0.4367	0.8833	6.9971	0.5083
Алгоритм Лукаса-Канаде (LK)	0.5423	0.4256	0.8799	7.6971	0.4769

Таблица 5.6 – Значения метрик различных алгоритмов при нестационарной камере

Метрики Методы	R	P	S	PWC	F1
Вычитание фона (BS)	0.2491	0.2073	0.3017	35.2297	0.2263
Vibe	0.3182	0.3029	0.3897	30.6471	0.3104
GMM	0.2547	0.2280	0.3289	31.5479	0.2406
Алгоритм Лукаса-Канаде (LK)	0.3071	0.2879	0.3697	33.9971	0.2972

Для более удобного анализа были построены диаграммы полученных метрик при различных условиях видеосъемки.



д

Рисунок 5.2 – Диаграммы полученных результатов

а) полнота;

б) точность;

в) специфичность;

г) процент неправильной классификации;

д) F-мера;

По данным диаграммам можно сделать следующие выводы:

1) наиболее эффективным алгоритмом обнаружения подвижных объектов по исследуемым критериям является алгоритм Vibe, так как для классификации пикселя используется сравнение не с одной постоянной моделью фона, а с несколькими образцами модели фона (20) для данного пикселя;

2) при появлении нестабильностей: плохая погода, динамический фон, дрожание камеры, эффективность алгоритмов падает.

Тени, отбрасываемые объектами переднего плана, усложняют процесс обработки изображения. Перекрывающиеся тени, например, препятствуют разделению объектов друг от друга и их идентификации. Наиболее эффективным при появлении тени оказался алгоритм Vibe.

Некоторые части сцены могут содержать незначительное движение (динамический фон), например: брызги фонтана, движения облаков, покачивание ветвей деревьев, волны на побережье и т. д. Такое движение необходимо рассматривать как фон, а не как движущийся объект. При данных условиях наилучший результат из всех алгоритмов показал алгоритм вычитания фона, использующий смесь Гауссовых распределений для моделирования фонового изображения.

3) при нестационарной камере алгоритмы показывают отрицательные результаты. Как видно по рисунку 5.2 (г) процент неправильной классификации резко возрастает, по сравнению с значениями данного критерия при других условиях;

4) наименее эффективным алгоритмом является алгоритм вычитания фона, однако он также является наименее ресурсоемким, поэтому его часто применяют в системах реального времени.

ГЛАВА 6. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Был разработан пользовательский интерфейс программной поддержки обнаружения движущихся объектов в системе Matlab.

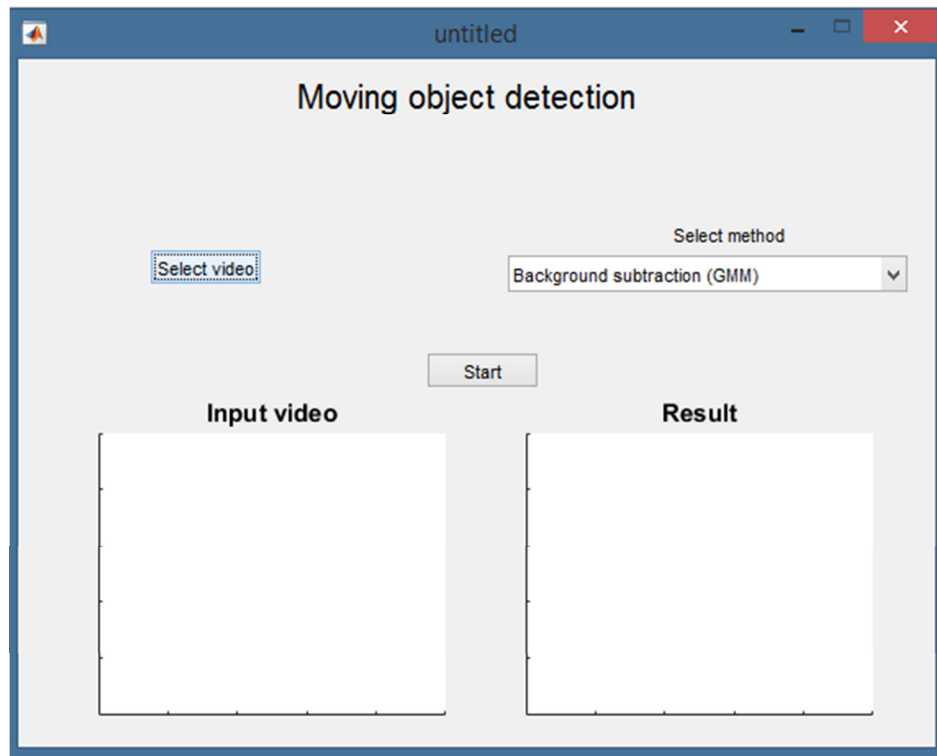


Рисунок 6.1 - Пользовательский интерфейс программной поддержки обнаружения движущихся объектов

Интерфейс программной поддержки содержит управляющие кнопки, с помощью которых можно выбрать необходимое видео и метод обнаружения подвижных объектов (рис. 6.2). Также имеется кнопка «Start», запускающая процесс обработки видео.

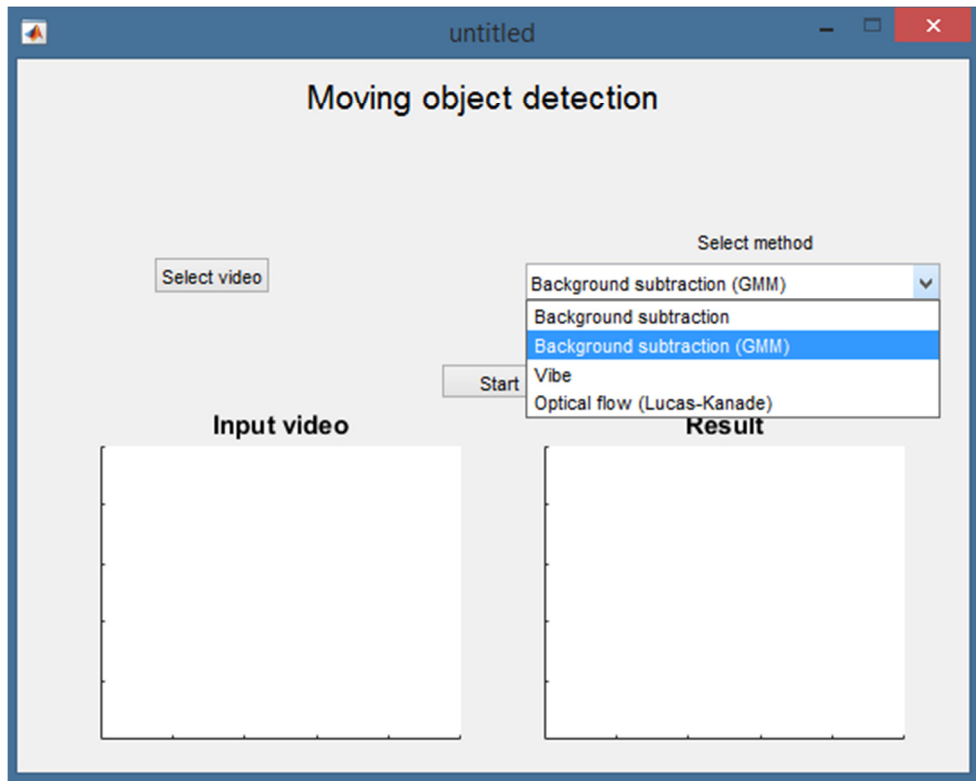


Рисунок 6.2 – Выбор метода обнаружения подвижных объектов

После нажатия кнопки «Start» будет запущен процесс обнаружения подвижных объектов посредством выбранного алгоритма. После чего на экране будет отображено исходное видео и полученный результат работы алгоритма (рис. 6.3).

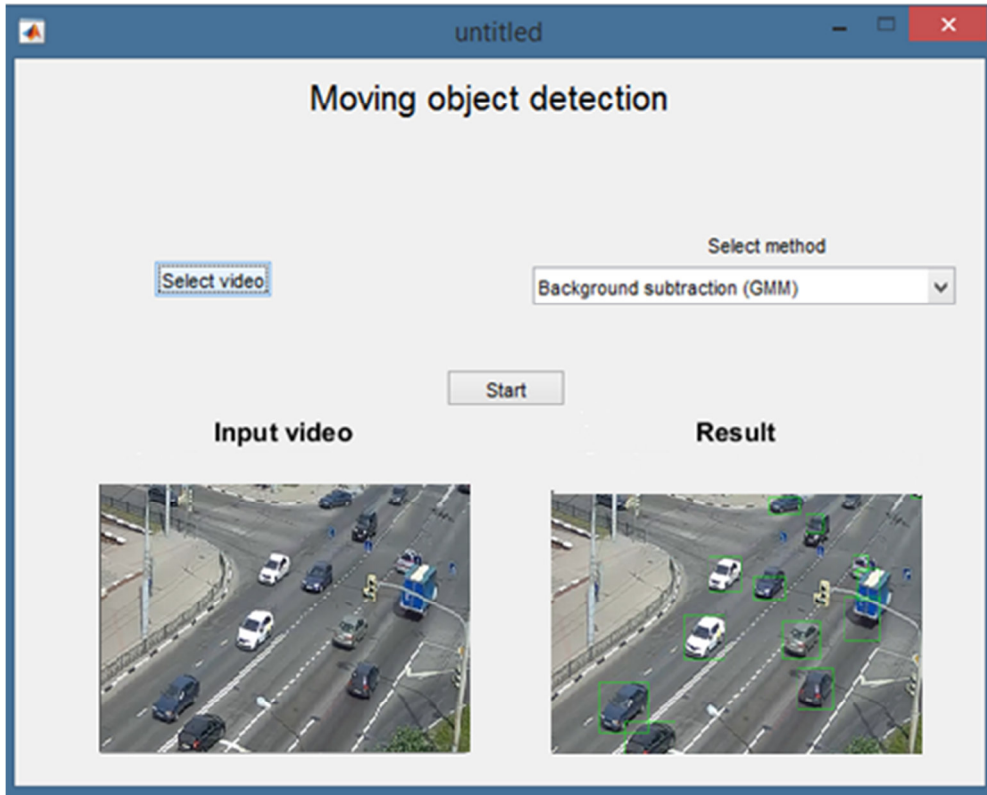


Рисунок 6.3 – Результат работы пользовательского интерфейса программной поддержки обнаружения движущихся объектов

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были получены следующие результаты:

1. Рассмотрены существующие методы обнаружения движущихся объектов в системах видеоаналитики. В результате чего выбраны наиболее популярные из них для последующего анализа.

2. Разработаны следующие алгоритмы обнаружения подвижных объектов:

- алгоритм вычитания фона;
- Vibe;
- алгоритм вычитания фона, использующий смесь Гауссовых распределений (GMM) для моделирования фонового изображения;
- алгоритм Лукаса-Канаде.

3. Проведены исследования разработанных алгоритмов на предмет оценки эффективности их работы при появлении на видеопоследовательности различных нестабильностей, таких как условия плохой погоды, нестационарная камера, динамический фон, тень, дрожание камеры.

4. Разработана программная поддержка исследуемых алгоритмов обнаружения движущихся объектов на видео-данных.

Основные результаты выпускной квалификационной работы апробированы в 4 печатных работах, двух конкурсах ПАО «Ростелеком» и на следующих конференциях:

1. Ежегодная межвузовской научно-технической конференции студентов, аспирантов и молодых специалистов имени Е.В. Арменского, Московский институт электроники и математики им. А.Н. Тихонова НИУ ВШЭ 17.02.2017г.-01.03.2017г.

2. VI Конгресс молодых ученых, Санкт-Петербургский национальный исследовательский университете информационных технологий, механики и оптики (Университет ИТМО) 18–21 апреля 2017г.

3. I Молодежная научно-практическая конференция с международным участием «Естественнонаучные, инженерные и экономические исследования в технике, промышленности, медицине и сельском хозяйстве», НИУ «БелГУ», 20.04.2017г.-21.04.2017г.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Видеоаналитика. Машинное зрение [Электронный ресурс] / www.tadviser.ru - российский интернет-портал и аналитическое агентство / URL:[http://www.tadviser.ru/index.php/Статья:Видеоаналитика_\(машинное_зрение\)](http://www.tadviser.ru/index.php/Статья:Видеоаналитика_(машинное_зрение)) (дата обращения 25.05.2018)
2. Radke, R.J. Image Change Detection Algorithms: A systematic survey / R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam // IEEE Transactions on Image Processing. – 2005. – V. 14(3). – P. 294-307.
3. Dai, X. The effects of image misregistration on the accuracy of remotely sensed change detection / X. Dai, S. Khorram // IEEE Trans. Geoscience Remote Sensing. – 1998. – V. 36(5). – P. 1566-1577.
4. Lillestrand, R. Techniques for change detection // IEEE Trans. On Computers. – 1972. – V. 21(7). – P. 654-659.
5. Li, J. A Video-based Algorithm for Moving Objects Detection at Signalized Intersection / J. Li, C. Shao, C. Dong, D. Zhao, Y. Liu // World Academy of Science, Engineering and Technology. International Journal of Computer, Electrical, Automation, Control and Information Engineering. – 2010. – V. 4(6). – P. 1081-1086.
6. Stojkoska, B. N-Queens-based Algorithm for Moving Object Detection in Distributed Wireless Sensor Networks / B. Stojkoska, D. Davcev, V. Trajkovic // Journal of Computing and Information Technology - CIT 16. – 2008. – V. 4. – P. 325-332.
7. Vu, T.D. Grid-based localization and local mapping with moving object detection and tracking / T.D. Vu, J. Burlet, O. Aycard // Journal of information Fusion. – 2010. – P. 58-62.
8. Vargas, M. An Enhanced Background Estimation Algorithm for Vehicle Detection in Urban Traffic Scenes / M. Vargas, J.M. Milla, S.L. Toral, F. Barrero // Vehicular Technology, IEEE Transactions. – 2010. – V. 59(8). – P. 3694-3709.

9. Evans A.N. Digital video preprocessing with multi-dimensional attribute morphology / N.Young and A.N. Evans // Proceedings IEE Visual Information Engineering Conference (VIE) July 2003. – Guildford, Surrey, UK, 2003. – pp.89 - 92.
10. Морфологические операции на бинарных изображениях [Электронный ресурс] / www.wiki.technicalvision.ru - профессиональный Wiki ресурс «Техническое зрение» / URL: http://wiki.technicalvision.ru/index.php/Морфологические_операции_на_бинарных_изображениях (дата обращения 20.04.2018)
11. Rosin P. Thresholding for change detection / P. Rosin // Sixth International Conference. 04 Jan 1998-07 Jan 1998 – Bombay, 1998. – P. 274-279.
12. Veit T. A maximality principle applied to a contrario motion detection / Veit T., Cao F., Bouthemy P. // IEEE International Conference on Image Processing. 14 Sept. 2005. – Genova, 2005.
13. Скрипкина, А.А. Обзор методов обнаружения движущегося объекта по видеоизображениям [Текст] / А.А. Скрипкина // Перспективы развития информационных технологий. – 2011. – № 3-1. – С. 126-127.
14. Töreyn, B.U. Moving Object Detection in Wavelet Compressed Video / B.U. Töreyn, A. Enis Çetin, A. Aksay, M.B. Akhan // Signal Processing: Image Communication, EURASIP. – 2005. – V. 20. – P. 255-264.
15. Bağcı, M. Moving object detection using adaptive subband decomposition and fractional lower order statistics in video sequences / M. Bağcı, Y. Yardımcı, A.E. Çetin // Signal Process. International Journal of Signal Processing. – 2002. – P. 1942-1947.
16. Nister D. Preemptive RANSAC for live structure and motion estimation / D.Nister // Nince IEEE International Conference on Computer Vision 13-16 Oct. 2003. – Nice, 2003. – P.199-206.
17. Cucchiara, R. Detecting moving objects, ghosts, and shadows in video streams / R. Cucchiara, C. Grana, M. Piccardi, A. Prati // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 25(10). – 2003. – P. 1337-1342.

18. Bouwmans, T. Recent Advanced Statistical Background Modeling for Foreground Detection / T. Bouwmans // *A Systematic Survey. – Recent Patents on Computer Science.* – 4(3). – 2011. – P. 147-176.
19. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход [Текст] / Д. Форсайт, Ж. Понс. – М.: Изд. д. Вильямс, 2004. – 465с.
20. Brutzer, S. Evaluation of background subtraction techniques for video surveillance / S. Brutzer, B. Höferlin, G. Heidemann // *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 20-25 June 2011.* – Providence, RI, USA. – P. 1937-1944.
21. Hofmann, M Background Segmentation with Feedback: The Pixel-Based Adaptive Segmenter / M. Hofmann, P. Tiefenbacher, G. Rigoll // *IEEE Computer Society Conference 16-21 June 2012.* – Providence, RI, USA. – P. 28-43.
22. Jodoin, P-M Statistical Background Subtraction Using Spatial Cues / P.-M. Jodoin, M. Mignotte, J. Konrad // *IEEE Transactions on Circuits and Systems for Video Technology.* – 17(12). – 2007. – P. 1758-1763.
23. Haritaoglu, I. W4: Who? When? Where? What? A real time system for detecting and tracking people / I. Haritaoglu, D. Harwood, LS. Davis // *Third Face and Gesture Recog Conf. 14-16 Apr 1998.* – Nara, 1998. – P. 222-227.
24. Bouwmans, T Statistical Background Modeling for Foreground Detection: A Survey / T. Bouwmans, F. El Baf, B. Vachon // *In Handbook of Pattern Recognition and Computer Vision (volume 4).* – World Scientific Publishing. – January 2010. – P. 181-199.
25. Stauffer, C. Adaptive background mixture models for real-time tracking / Chris Stauffer, W. Eric L. Grimson // *Conference on Computer Vision and Pattern Recognition 23-25 June 1999.* – Ft. Collins, CO, USA, 1999. – P. 2246-2252.
26. Kim, K Real-time foreground-background segmentation using codebook model / K. Kim, T. Chalidabhongse, D. Harwood, L. Davis // *Real-Time Imaging.* – 11(3) – 2005. – P. 172-185.

27. Barnich, O. ViBe: A universal background subtraction algorithm for video sequences / O. Barnich, M. V. Droogenbroeck // IEEE Transactions on Image Processing. June 2011 – 2011. P.1709-1724.
28. Lee, P.H. Real-time pedestrian and vehicle detection in video using 3D cues / P.H. Lee, T.H. Chiu, Y.L. Lin, Y.P. Hung // IEEE international conference on Multimedia and Expo. June 28 2009-July 3 2009. – New York, 2009. – P. 614-617.
29. Bouwmans, T. Background Modeling using Mixture of Gaussians for Foreground Detection / T. Bouwmans, F. El Baf, B. Vachon // A Survey Recent Patents on Computer Science. – 2008. – V. 1(3). – P. 219-237.
30. Stauffer, C Adaptive background mixture models for real-time tracking / C. Stauffer, E. Grimson // IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). – 1999. – V. 2. – P. 246-252.
31. Parks, D Evaluation of Background Subtraction Algorithms with Post-Processing / D. Parks, S. Fels // IEEE International Conference on Advanced Video and Signal Based Surveillance. – 2008 – P. 192-199.
32. Porikli, F Bayesian Background Modeling for Foreground Detection / F. Porikli, O. Tuzel // ACM international workshop on Video surveillance & sensor networks. – 2005. – P. 55-58.
33. Lin H.-H. Learning a Scene Background Model via Classification / H.-H. Lin, T.-L. Liu, J.-C. Chuang // IEEE Signal Processing Magazine. – 2009. – V. 57(5). – P. 1641-1654.
34. Sun, L Multimode Spatiotemporal Background Modeling for Complex Scenes / L. Sun, Q. De Neyer, C. De Vleeschouwer // European Signal Processing Conference (EUSIPCO) 27-31 Aug. 2012. – Bucharest, Romania. – P. 165-169.
35. Elgammal, A. Non-parametric Model for Background Subtraction / A. Elgammal, D. Harwood, L. Davis // Proceedings of the 6th European Conference on Computer Vision. – Part II. –2000. – P. 751-767.
36. Droogenbroeck, M. ViBe: A Disruptive Method for Background Subtraction. In Background Modeling and Foreground Detection for Video

Surveillance / M. Van Droogenbroeck, O. Barnich // Chapman and Hall/CRC. – July 2014. – P. 7.1-7.23.

37. Manzanera, A. A new motion detection algorithm based on Σ - Δ background estimation / A. Manzanera, J. Richefeu // Pattern Recognition Letters – V. 28(3). – 2007. – P. 320-328.

38. Barnich, O. ViBe: A universal background subtraction algorithm for video sequences / O. Barnich, M. Van Droogenbroeck // In IEEE Transactions on Image Processing. – 20(6). – 2011. – P. 1709-1724.

39. Manzanera, A. Local Jet Feature Space Framework for Image Processing and Representation / A. Manzanera // International Conference on Signal Image Technology & Internet-Based Systems 28 Nov.-1 Dec. 2011. – Dijon, France. – P. 261-268.

40. Schick, A. Improving Foreground Segmentation with Probabilistic Superpixel Markov Random Fields / A. Schick, M. Bauml, R. Stiefelhagen // IEEE Computer Society Conference 16-21 June 2012. – Providence, RI, USA. – P. 27 – 31.

41. Droogenbroeck, M. Background Subtraction: Experiments and Improvements for ViBe / M. Van Droogenbroeck, O. Paquot // IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 16-21 June 2012. – Providence, Rhode Island, USA. – P. 32-37.

42. Srivastava, A. On Advances in Statistical Modeling of Natural Images / A. Srivastava, A. Lee, E. Simoncelli, S.-C. Zhu // Journal of Mathematical Imaging and Vision. – V. 18(1). – 2003. – P. 17-33.

43. Shoushtarian, B. A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking / B. Shoushtarian, H. Bez // Pattern Recognition Letters. – V. 26(1). – 2005. – P. 5-26.

44. Wren, C. Pfunder: Real-Time Tracking of the Human Body / C. Wren, A. Azarbayejani, T. Darrell, A. Pentland // IEEE Transactions on Pattern Analysis and Machine Intelligence. – V. 19(7). – 1997. – P. 780-785.

45. Wang, H. A consensus-based method for tracking: Modelling background scenario and foreground appearance / H. Wang, D. Suter // *Pattern Recognition*. – V. 40(3). – 2007. – P. 1091-1105.
46. Barnich, O. ViBe: a powerful random technique to estimate the background in video sequences / O. Barnich, M. Van Droogenbroeck. // In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, 19-24 April 2009. – Taipei, Taiwan. – P. 945-948.
47. Visual background extractor [Текст]: пат. EP 2 015 252 B1 Europe: G06T 7/20 / авторы и заявители М. Van Droogenbroeck, О. Barnich; заявл. 2007-07-08; опубл. 17-Feb-2010.
48. Benezeth, Y. Review and evaluation of commonly-implemented background subtraction algorithms / Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, C. Rosenberger // *IEEE International Conference on Pattern Recognition (ICPR)* 16 Dec 2008. – Tampa, United States. – P. 35-38.
49. Kryjak, T. Real-time Implementation of the ViBe Foreground Object Segmentation Algorithm / T. Kryjak, M. Gorgon // *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS)* 8-11 Sept. 2013. – Krakow, Poland. – P. 591-596.
50. Visual background extractor [Текст]: пат. JP 2011 4699564 B2 Japan: G06T1/00, G06T7/20 / авторы и заявители М. Van Droogenbroeck, О. Barnich; заявл. 2010-11-14; опубл. 15-Jun-2011.
51. Dike, A. K. ViBe: A Universal Background Subtraction Algorithm for Video Sequences / Aarti K Dike, D.R.Deshmukh // *International Journal of Advance Engineering and Research Development (IJAERD)*. – Volume 4. – Issue 7. – July-2017. – P. 2348-6406.
52. Liao, S. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes / S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, S. Li // In *IEEE Int. Conf. on Computer Vision and Pattern Recognition* 13-18 June 2010. – San Francisco, CA, USA. – P. 1301–1306.

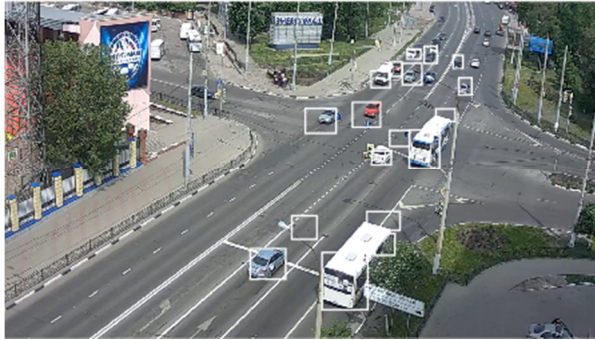
53. Bilodeau, G.-A. Change detection in feature space using local binary similarity patterns / G.-A. Bilodeau, J.-P. Jodoin, N. Saunier // In Int. Conf. on Computer and Robot Vision 28-31 May 2013. – Regina, SK, Canada. – P. 106–112.
54. Xue, G. Hybrid center-symmetric local pattern for dynamic background subtraction / G. Xue, L. Song, J. Sun, M. Wu // In IEEE International Conference on Multimedia and Expo 11-15 July 2011. – Barcelona, Spain. – P. 1–6.
55. Xue, G. Dynamic background subtraction based on spatial extended center-symmetric local binary pattern / G. Xue, L. Song, J. Sun // In IEEE International Conference on Multimedia and Expo 19-23 July 2010. – Suntec City, Singapore. – P. 1050–1054.
56. Silva, C. An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos / C. Silva, T. Bouwmans, C. Frélicot // International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2015. – Mar 2015. – Berlin, Germany. –P. 33-40.
57. Fleet, D. J. Optical flow estimation / David J. Fleet, Y. Weiss // Mathematical Models in Computer Vision: The Handbook – 2005. – C. 15 – P. 239–258.
58. Оптический поток [Электронный ресурс] / www.ru.wikipedia.org – свободная энциклопедия / URL: https://ru.wikipedia.org/wiki/Оптический_поток (дата обращения 22.04.2018)
59. Performance of optical flow techniques / J. L. Barron, D. J. Fleet, S. S. Beauchemin, T. A. Burkitt // IEEE Computer Society Conference 15-18 Jun 1992. – Champaign, 1992 – P. 236 - 242.
60. Lucas, B. D. An Iterative Image Registration Technique with an Application to Stereo Vision / B. D. Lucas, T. Kanade // Proc. of 7th International Joint Conference on Artificial Intelligence, August 1981. – Vancouver, 1981. – P. 674-679.

61. Yilmaz, A. Object tracking: A survey / A.Yilmaz, O.Javed, M. Shah // ACM Computing Surveys. – 2006. – V. 38. – № 4. – P. 33-37.
62. Veenman, C. Resolving motion correspondence for densely moving points / C. Veenman, M. Reinders, E. Backer // IEEE Trans. Pattern Analysis Machine Intelligence. – 2001. – V.23. – № 1. – P. 54-72.
63. Salarpour, A. Vehicle tracking using Kalman filter and features / A. Salarpour, A. Salarpour, M. Fathi, Dezfoulan MirHossein // Signal & Image Processing: An International Journal (SIPIJ). – 2011. – V.2. – №2. – P. 44-48.
64. Dan, S. A Tracking Algorithm Based on SIFT and Kalman Filter / S. Dan, Zh Baojun, T. Linbo // In Proceedings The 2nd International Conference on Computer Application and System Modeling. – 2012. – P.1563-1566.
65. Isard, M. Condensation - conditional density propagation for visual tracking / M. Isard, A. Blake // Int. J. Computer Vision. – 1998. – V.29. – № 1. – P. 5-28.
66. Gustafsson, F. Particle Filters for Positioning, Navigation and Tracking / F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, Jonas Jansson, R. Karlsson, P.J. Nordlund // IEEE Transactions on Signal Processing. – 2002. – V.2. – Issue 2. – P. 425-437.
67. Comaniciu, D. Real-time tracking of non-rigid objects using mean shift / D. Comaniciu, V. Ramesh, P. Meer // In Proceedings of the CVPR'00. – 2000. – V.2. – P. 142-149.
68. Exner, D. Fast and robust CAMShift tracking / D. Exner, E. Bruns, D. Kurz, A. Grundhofer // In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. – 2010. – P. 9-16.
69. Gauglitz, S. Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking / S. Gauglitz, T. Holerer, M. Turk // Int J Computer Vision. – 2011. – P. 335–360.
70. Sonka, M. Image Processing, Analysis and Machine Vision / M. Sonka, V. Hlavac, R. Boyle // 3rd Edition. — Thomson. – 2008. – XXV. – P. 829.

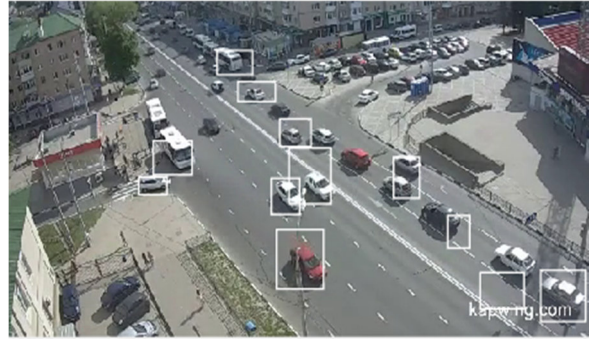
71. Almeida, A. Real-Time Tracking of Moving Objects Using Particle Filters / A. Almeida, J. Almeida, R. Araujo // IEEE ISIE 2005. – 20-23 June 2005. – Dubrovnik, Croatia. – P. 1327-1332.

72. Isard, M. Condensation - conditional density propagation for visual tracking / M. Isard, A. Blake // International Journal of Computer Vision. – August 1998. – V. 29. – Issue 1. – P. 5–28.

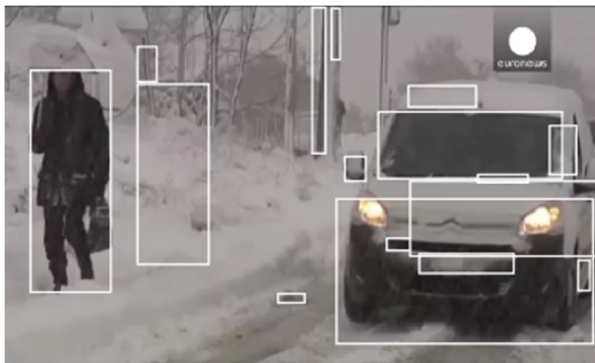
ПРИЛОЖЕНИЕ А



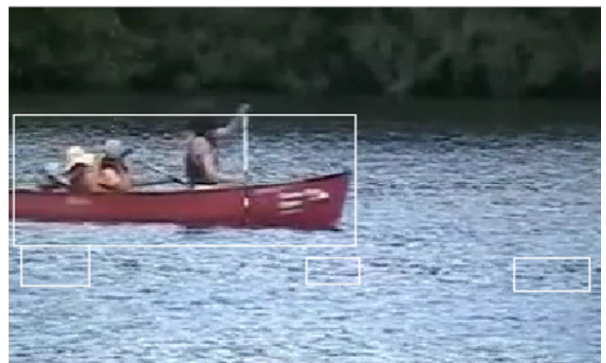
а



б



в



г



д



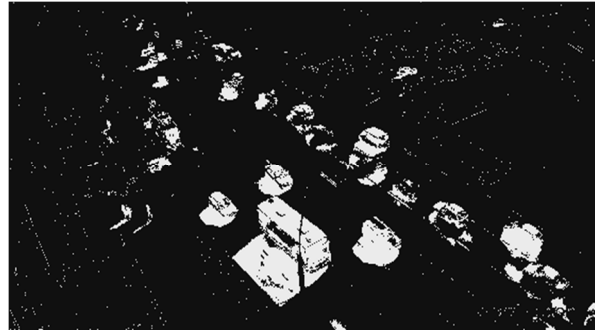
е

Рисунок 1 – Результаты работы алгоритма вычитания фона

- а) базовые условия;
- б) тень;
- в) условия плохой погоды;
- г) динамический фон;
- д) дрожание камеры;
- е) нестационарная камера



а



б



в



г



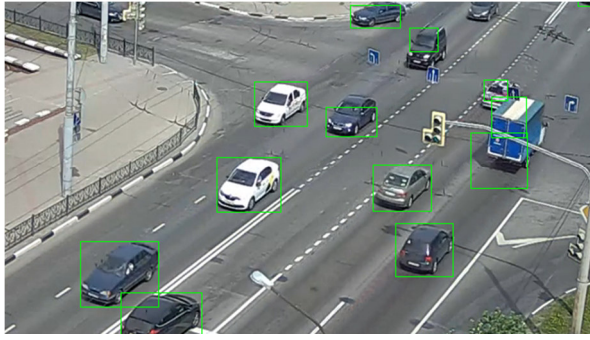
д



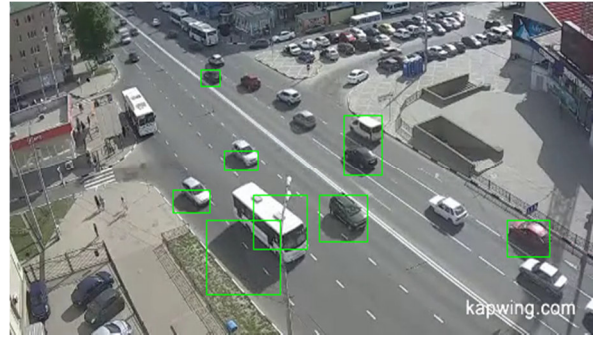
е

Рисунок 2 – Результаты работы алгоритма Vibe

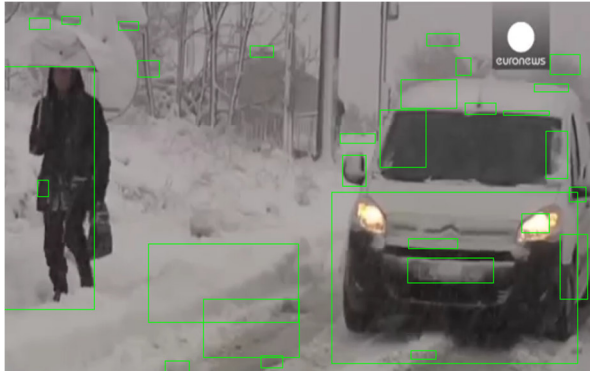
- а) базовые условия;
- б) тень;
- в) условия плохой погоды;
- г) динамический фон;
- д) дрожание камеры;
- е) нестационарная камера



а



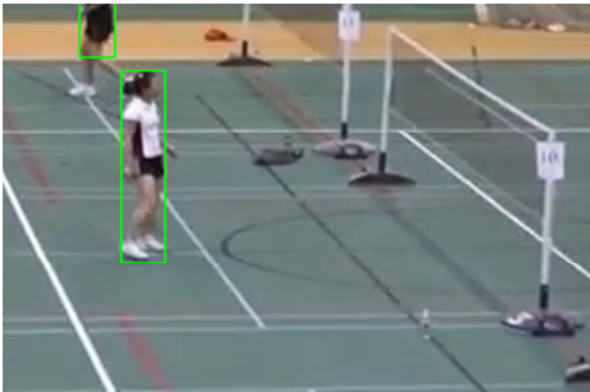
б



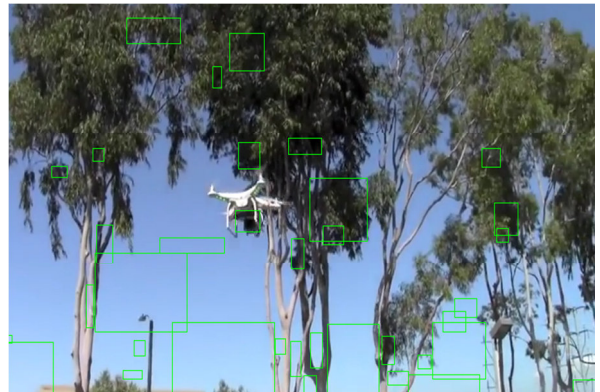
в



г



д



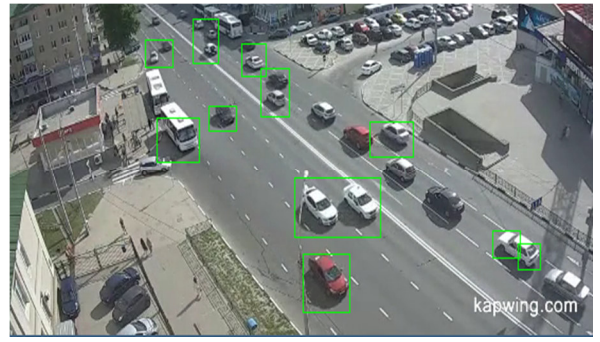
е

Рисунок 3 – Результаты работы метода вычитания фона с использованием смеси Гауссовых распределений (GMM)

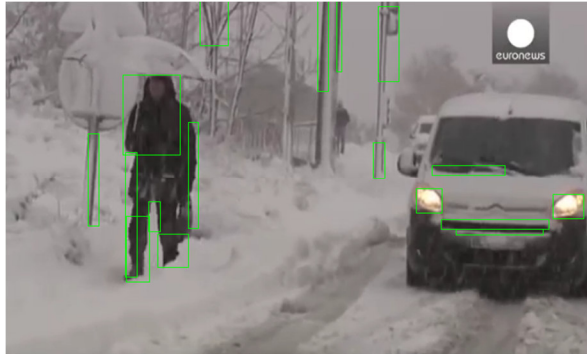
- а) базовые условия;**
- б) тень;**
- в) условия плохой погоды;**
- г) динамический фон;**
- д) дрожание камеры;**
- е) нестационарная камера**



а



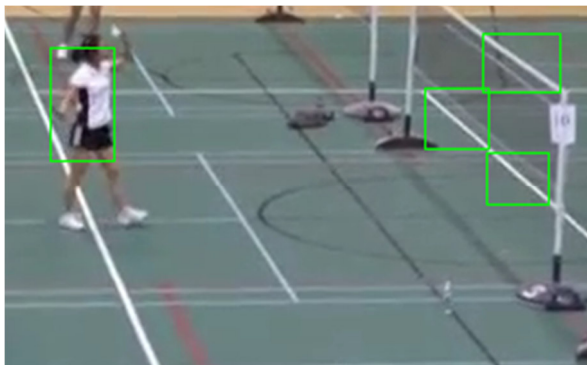
б



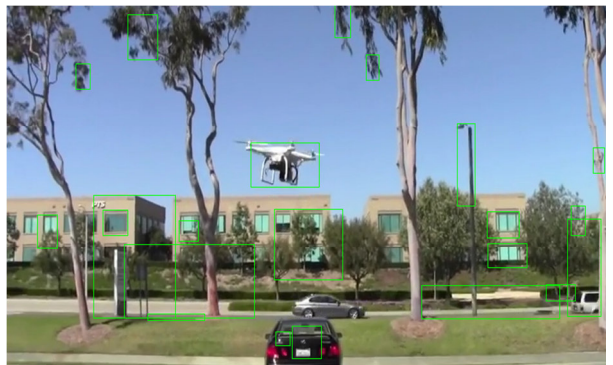
в



г



д



е

Рисунок 4 – Результаты работы метода оптического потока (алгоритм Лукаса-Канаде)

- а) базовые условия;**
- б) тень;**
- в) условия плохой погоды;**
- г) динамический фон;**
- д) дрожание камеры;**
- е) нестационарная камера**