

Санкт-Петербургский государственный университет

Кафедра системного программирования

Лялинов Илья Михайлович

Автомобильная система распознавания,  
слежения и определения расстояния до  
объекта в зоне движения

Курсовая работа

Научный руководитель:  
ст. преп. Смирнов М. Н.

Санкт-Петербург  
2018

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Постановка задачи</b>	<b>5</b>
<b>2. Обзор</b>	<b>6</b>
2.1. Распознавание (objects detection) . . . . .	6
2.2. Типы алгоритмов слежения . . . . .	8
2.3. Алгоритмы слежения библиотеки OpenCV . . . . .	10
2.4. Измерение расстояния до объекта с помощью монокулярного зрения . . . . .	12
<b>3. Реализация</b>	<b>15</b>
3.1. Инструменты . . . . .	15
3.2. Определение дистанции до объектов . . . . .	15
3.3. Модуль слежения . . . . .	16
3.4. Алгоритм . . . . .	17
3.5. Апробация . . . . .	18
<b>4. Заключение</b>	<b>20</b>
<b>Список литературы</b>	<b>21</b>

# Введение

По статистике Всемирной Организации Здравоохранения более 10 миллионов человек ежегодно получают травмы разной степени тяжести в результате дорожно-транспортных происшествий. Чтобы уменьшить количество пострадавших в результате аварий на дорогах, разрабатываются системы автоматического управления транспортным средством и продвинутые системы помощи водителям (ADAS). Самые продвинутые из таких систем автоматически применяют торможение в экстренных ситуациях, имеют подсистемы ночного зрения, помощи при парковке, предупреждают водителя о критических уровнях его сонливости или усталости. Основными возможностями таких систем являются распознавание (detection), слежение (tracking) и определение дистанции до движущихся объектов на дороге.

Процесс распознавания объекта состоит из локализации и классификации. Локализация - это процесс поиска точных координат целевого предмета на изображении. Классификация - это процесс разбиения множества объектов на несколько классов. Например, в нашем случае алгоритм классификации должен различать следующие виды объектов на дороге: дорожные знаки, пешеходы, мотоциклы, велосипеды, машины, грузовики, автобусы.

Задача слежения (tracking) представляет собой проблему обнаружения объекта на нескольких последовательных видеокдрах. Такое определение выглядит просто, однако в области компьютерного зрения и машинного обучения задача отслеживания подразумевает много концептуально похожих, но совершенно разных с точки зрения технической реализации подходов.

Существуют различные способы расчета расстояния до непрозрачного объекта. Для решения таких задач часто используют радар или лидар. Показания этих приборов отличаются высокой точностью и скоростью их получения. Однако радар и лидар – технически сложные механизмы. Следовательно, их использование сильно повышает общую стоимость автомобильной системы дорожного контроля. К тому же пример

человеческого зрения указывает на возможность существования других способов достижения приемлемой точности детекции и определения дистанции до дорожных объектов.

Видео-системы определения расстояния можно разделить на 2 типа: на основе стереозрения (с использованием видеоданных с 2 камер) и на основе монокулярного зрения (с использованием одной камеры). В первом случае используются кадры с двух камер. Расстояние между камерами обычно известно и не изменяется. Две точки двух изображений, соответствующие одной точке  $X$  в пространстве, называют стереопарой. Для каждой точки на первом изображении (с первой камеры) выполняется поиск парной ей точки на изображении со второй камеры. Затем, используя известные параметры камер, по стереопаре вычисляют пространственные координаты прообраза. Так строится карта глубин. Системы второго типа не так популярны и изучены. Они не требуют дополнительной камеры, поэтому алгоритмы определения расстояния на основе монокулярного зрения открывают путь к более простым и менее дорогим решениям рассматриваемой задачи. В данной работе описывается процесс реализации системы распознавания, слежения и определения расстояния до дорожных объектов с использованием видеоданных только одной камеры.

# 1. Постановка задачи

Конечной целью работы является реализация и апробация алгоритма распознавания, слежения и определения дистанции до объектов на дороге. Входными данными алгоритма должно быть видео с одной камеры. Для достижения обозначенной цели были выделены следующие подзадачи:

1. проанализировать способы распознавания объектов;
2. провести обзор различных типов и алгоритмов слежения;
3. проанализировать существующие решения задачи определения дистанции до объекта с использованием монокулярного зрения;
4. реализовать систему распознавания, слежения и определения расстояния с использованием видеоданных одной камеры;
5. провести апробацию системы на тестовых видеоданных;

## 2. Обзор

### 2.1. Распознавание (objects detection)

Задача локализации и классификации не является неизученной. Для ее решения существует множество алгоритмов, многие из которых основаны на методах машинного обучения.

#### **SSD (Single Shot Multibox Detector) [1]**

Для выполнения всех необходимых вычислений использует единую сверточную нейронную сеть. На тестовом наборе данных Pascal VOC 2007 алгоритм демонстрирует 74.3% mAP, обрабатывая 59 изображений в секунду. Скорость работы SSD делает возможным его использование в некоторых системах реального времени.

#### **Faster R-CNN [2]**

Алгоритм является улучшением метода Fast R-CNN. Ядром алгоритма также является сверточная нейронная сеть. В отличие от SSD здесь она применяется не только к самому изображению, но и ко многим его прямоугольным частям, что негативно сказывается на скорости работы всего метода. Сначала выделяются регионы, в которых есть распознаваемые объекты, а потом решается задача классификации. После выделения нужных прямоугольников, сверточная нейронная сеть генерирует для каждого из них вектор признаков. Далее эти данные обрабатываются заранее обученными классификаторами, которые определяют точный тип объекта. Алгоритм обрабатывает всего 5 кадров в секунду из тестового набора данных Pascal VOC 2007, показывая 76.4% mAP.

#### **YOLOv2, YOLOv3 (You Only Look Once) [3]**

Также является алгоритмом распознавания на основе сверточной нейронной сети. На изображении (рис. 1) показаны результаты работы

методов YOLOv2, SSD, Faster R-CNN на наборе данных Pascal VOC 2007. В YOLOv3 были добавлены дополнительные логические уровни (сверточные слои), улучшающие распознавание небольших объектов. Это повысило показатель mAP, но и общее время распознавания также увеличилось. Однако распознавание методом YOLOv3 все равно выполнялось быстрее, чем алгоритмами SSD или Faster R-CNN

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	<b>78.6</b>	40

Рис. 1: Сравнение фреймворков распознавания

## Сравнение и анализ методов

Алгоритм YOLOv3 по показателю mAP превосходит метод SSD, но уступает методу Faster R-CNN. Однако фреймворк Faster R-CNN обрабатывает не более 5 изображений в секунду. С использованием видеопроцессора NVIDIA Titan X алгоритм YOLOv3 обрабатывает изображения со скоростью 30 кадров в секунду, что делает возможным его применение в видеосистемах реального времени. Поэтому для решения поставленной задачи был выбран метод YOLOv3.

## 2.2. Типы алгоритмов слежения

Обычно выделяют три основных типа алгоритмов слежения: слежение за набором точек, представляющих объект, слежение за ядром объекта и слежение за силуэтом объекта.

### Слежение за особыми точками

Такой тип слежения подразумевает поиск соответствий целевого объекта представленного его особенными точками на нескольких последовательных кадрах. Задача особенно сложна из-за возможных окклюзий, ложных распознаваний, появления и исчезновения искомого объекта с видеокadra. Алгоритмы этого типа делятся на детерминированные и статистические. Детерминированный подход предполагает использование ограничений:

1. положение объекта в кадре изменяется не более чем на заранее заданное значение;
2. скорость близких точек должна быть примерно одинаковой;
3. определяется верхняя граница возможной скорости объекта;
4. определяется верхняя граница возможного изменения скорости объекта;

Каждый объект с предыдущего кадра сравнивается с объектом на новом кадре. Для решения задачи сопоставления часто используются жадные алгоритмы. Статистический подход решает задачу слежения, используя текущие данные о состоянии объекта (его скорость, координаты, ускорение, и т. д.). Такие алгоритмы явно обрабатывают шумовые помехи. Часто рассматриваются функции шума с нормальным распределением.

### Слежение за ядром объекта

Этот тип методов слежения основан на расчете движения объекта, ограниченного примитивной геометрической фигурой. Например,

эллипсом или прямоугольником. Выделяют 2 подвида таких алгоритмов. Один из них - слежение с использованием шаблона. Под шаблоном объекта здесь понимается матрица признаков, характерная для целевого объекта. С помощью готового шаблона объекта с кадра  $t - 1$  на кадре  $t$  ищется регион с таким же или похожим шаблоном. Часто для создания шаблона используется яркость или цвет пикселей. Также подходят градиенты изображения. Нередко образец составляется с учетом нескольких разных характеристик изображения. Второй тип слежения за ядром - слежение с использованием нескольких моделей вида объекта. В предыдущем подвиде слежения за ядром используемый шаблон менялся от кадра к кадру. То есть такие методы подразумевают, что объект и на следующих кадрах будет выглядеть похожим образом. Однако как только вид объекта меняется кардинально, слежение с помощью таких шаблонов работать перестает. Решить эту проблему можно заранее сгенерировав несколько различных шаблонов для отслеживаемого объекта. Примером такого подхода является метод машины опорных векторов, используемый для слежения.

### **Слежение за силуэтом объекта**

Некоторые объекты могут иметь сложную форму, которую невозможно описать с использованием простых геометрических фигур. Например, пальцы, рука или плечо. Слежение за силуэтом объекта предполагает использование точного описания формы целевого предмета. Различают 2 подвида таких методов: на основе сопоставления формы и слежение за контуром. Первый тип похож на слежение за ядром объекта с использованием сопоставления шаблона. Сначала составляется модель формы целевого объекта. Затем на основе этой модели производится поиск предмета на следующем кадре. При этом описание формы объекта всегда составляется по одному видеок cadру. Такой подход позволяет корректно обрабатывать незначительные изменения формы предмета. Слежение за контуром лучше обрабатывает ситуации резкого изменения формы объекта. Для поиска изменений контура необходимо, чтобы определенная часть объекта попадала с предыдущего кадра пере-

секалась с его ограничивающим регионом на следующем кадре. Есть 2 основных решения проблемы слежения за контуром. С помощью минимизации функционала энергии по контуру и с использованием математической модели состояния. Задача минимизации решается жадными алгоритмами или методом градиентного спуска.

## **2.3. Алгоритмы слежения библиотеки OpenCV**

### **SIFT (scale-invariant feature transform)**

Это не алгоритм сопровождения, а детектор особых точек на изображении. Имея 2 последовательных кадра, на первом из которых выделен объект, подлежащий слежению, можно найти особые точки и с помощью и сопоставив их определить положение целевого объекта на следующем кадре.

### **Medianflow**

Этот алгоритм пытается отследить движение объекта не только от кадра  $t-1$  к кадру  $t$ , но и наоборот. Для этого изображение делится на маленькие регионы и для них используется алгоритм поиска оптического потока. Медиана смещения всех точек объекта принимается за смещение его центра. Этот алгоритм хорошо определяет сбой процесса слежения и хорошо работает в случаях плавного движения объекта. Плохо обрабатывает случаи перекрытия и частичного исчезновения объекта.

### **Boosting**

Использует алгоритм бустинга AdaBoost. Является одним из первых алгоритмов слежения, реализованных в библиотеке OpenCV. Используемый классификатор рассматривает выделенный объект на первом кадре как положительный пример. А остальные регионы изображения, не пересекающиеся с регионом целевого предмета, как отрицательный. На следующем кадре происходит поиск объекта недалеко от его преды-

дущего положения. Найденный регион на новом кадре используется для обучения классификатора как еще один положительный пример.

### **MIL (multiple instance learning)**

Концептуально похож на алгоритм слежения Boosting. Различие заключается в использовании классификатором сразу нескольких пересекающихся регионов на новом кадре в качестве положительного примера для обучения.

### **TLD (tracking learning detection)**

Алгоритм состоит из трех частей: детекция, слежение и обучение. На каждом кадре запускается процесс распознавания и на основании результата происходит корректировка процесса слежения. Суть обучения заключается в оценке показаний детектора и распознавании его ошибок. Этот алгоритм хорошо работает в случаях частичного перекрытия объекта.

### **KCF (kernelized correlation filter)**

Авторы утверждают, что этот фреймворк слежения работает точнее и быстрее чем метод TLD. Алгоритм также как и MIL и Boosting использует онлайн обучение классификатора. Авторы ускорили и улучшили алгоритм обучения. Таким образом, алгоритм KCF выигрывает в точности и скорости слежения у методов MIL и Boosting.

### **Сравнение методов слежения OpenCV [4]**

Введем показатель  $C_{succ} = (r_t \cap r_g) / (r_t \cup r_g)$ , где  $r_t$  - регион предсказанный алгоритмом слежения,  $r_g$  - правдивый регион целевого объекта.

На рис. 2 можно видеть процент успешных кадров для разных алгоритмов слежения. Кадр считается “успешным”, если показатель  $C_{succ} > 0.5$ . Авторы исследования [5] не включили в сравнение алгоритм KCF, однако как мы уже отмечали, на тестовых данных этот алгоритм

работает точнее и быстрее чем методы Boosting и MIL. Медленнее всего работают методы SIFT и TLD.

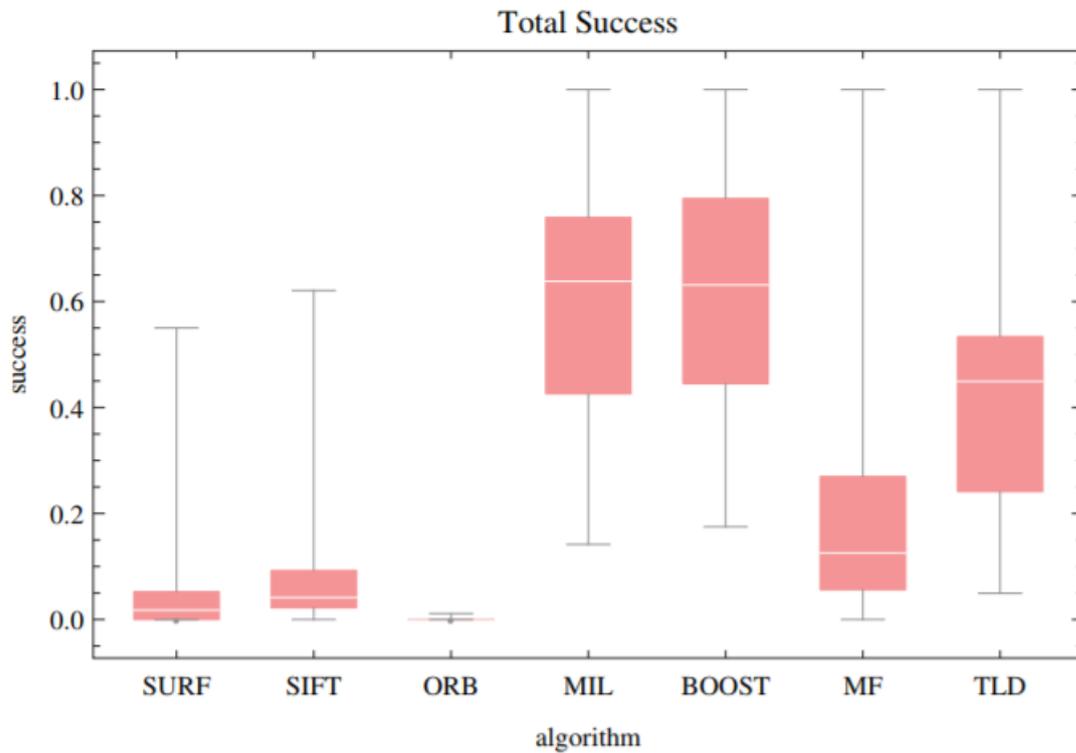


Рис. 2: Сравнение алгоритмов слежения, реализованных в OpenCV

## 2.4. Измерение расстояния до объекта с помощью монокулярного зрения

Существуют различные алгоритмы определения расстояния до объекта по монокулярного видео. Далее рассматриваются конкретные подходы решения задачи.

### Определение дистанции до объекта с использованием информации о его размере

Если известны реальные размеры предмета (высота или ширина), то, зная координаты объекта на изображении, с помощью формул проективной геометрии можно рассчитать расстояние до него. Недостатком такого подхода является увеличение ошибки при уменьшении объ-

екта. К тому же для корректной работы алгоритма длина и ширина объекта не должны меняться с течением времени. Также для использования этого метода необходимо заранее знать фокусное расстояние камеры.

### **Определение дистанции до поверхности с использованием логарифмического преобразования изображения [6]**

Для использования такого подхода определения дистанции до объекта необходимо выполнение двух условий: камера должна двигаться только вдоль оптической оси и расстояние между двумя позициями камеры (которые соответствуют двум видеокадрам) должно быть известно. Оба изображения переводятся из ортогональной системы координат в полярную. Затем из полярных координат получают логарифмически-полярные. Такое представление изображений упрощает операции его масштабирования и поворота. После этого на двух изображениях сопоставляются пары точек, которые соответствуют одной точке в пространстве. Теперь можно вычислить отношение размеров проекций объекта на 2 изображения, что делает возможным дальнейшее определение дистанции до объекта. Такой метод хорош, если форма объекта может меняться. Но как пишут авторы, этот метод не может быть использован в системах реального времени.

### **Определение дистанции и скорости объекта с помощью монокулярного зрения [7]**

Этот метод позволяет определять дистанцию до дорожного объекта по одному видеокадру. Для корректного расчета расстояний до объектов на дороге необходимо, чтобы оптическая ось камеры была параллельна плоскости дороги. Также надо знать высоту, на которой подвешена камера и ее фокусное расстояние. Для определения расстояния до объекта находится точка его соприкосновения с дорогой и используются некоторые формулы проективной геометрии. Вычислительная сложность такого способа позволяет использовать его в системах реального

времени.

## 3. Реализация

### 3.1. Инструменты

В качестве языка реализации был выбран язык C++. Для распознавания использовалась предобученная модель детектора YOLO-v3. Работа с изображениями и видео осуществлялась при помощи библиотеки компьютерного зрения OpenCV.

### 3.2. Определение дистанции до объектов

Для реализации был выбран метод из статьи [7]. Опишем его подробнее. Пусть плоскость дороги параллельна оптической оси камеры. И пусть расстояние от камеры до плоскости дороги известно и не изменяется со временем. Рассмотрим точку дороги на расстоянии  $Z$  от проекции камеры на плоскость дороги. Ей будет соответствовать точка изображения на высоте  $y$ . Тогда верно равенство (1):

$$y = \frac{f * H}{Z} \quad (1)$$

Где  $f$  - фокусное расстояние камеры, а  $H$  - длина перпендикуляра проведенного от камеры на плоскость дороги.

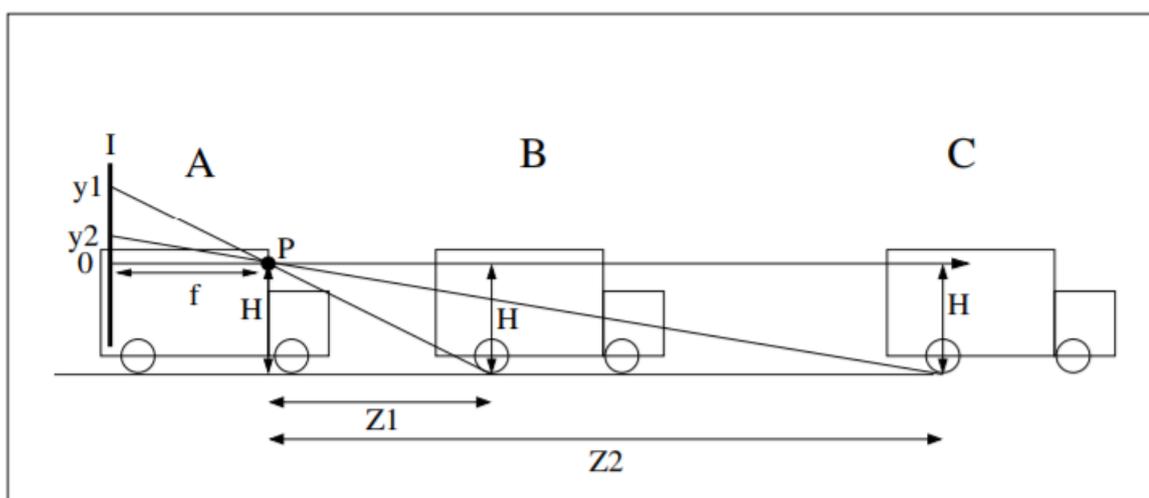


Рис. 3: Схема создания изображения

На рисунке показан принцип действия камеры-обскуры. Р - входное отверстие камеры, I - плоскость изображения, Н - высота камеры, у1 - проекция точки соприкосновения с дорогой автомобиля В, у2 - проекция точки соприкосновения с дорогой автомобиля С.

Соотношение (1) получается из равенства

$$\frac{y}{f} = \frac{H}{Z} \quad (2)$$

А (2) следует из подобия треугольников. Расстояние до автомобиля (Z1 или Z2 на рисунке) можно вычислить по формуле:

$$Z = \frac{f * H}{y} \quad (3)$$

Погрешность расстояния, вычисленного по формуле (3), квадратично зависит от реального расстояния между камерой и автомобилем.

### 3.3. Модуль слежения

Подсистема сопровождения объектов дополняет распознавание. Во-первых, алгоритмы слежения как правило быстрее алгоритмов распознавания. Использование методов сопровождения дает возможность не использовать операцию обнаружения на каждом видеокадре. Во-вторых, алгоритмы слежения могут обрабатывать случаи, когда методы распознавания не работают. Например, когда объект оказывается частично перекрытым. В-третьих, алгоритмы слежения строят траекторию движения объекта в кадре (сохраняют идентичность объекта на нескольких кадрах), в то время как алгоритмы распознавания работают только с одним изображением. Таким образом, методы сопровождения позволяют строить траекторию движения объекта в кадре и предсказывать его положение.

Для каждого нового найденного детектором объекта вычисляется его номер слежения. Мы предполагаем, что смещение объекта за 10 кадров невелико. Результаты работы детектора за последние 10 кадров объединяются и среди них ищется прямоугольник того же типа (маши-

на, автобус, и т. д.), расположенный рядом с текущим положением объекта в кадре. Если такой ограничивающий прямоугольник найден, то его `track_id` присваивается и новому объекту. В противном случае новый объект получает уникальный номер слежения. Затем для каждого объекта строится его траектория движения в кадре. Считается средний вектор смещения за 1 кадр. Далее по известному положению объекта в кадре и среднему вектору его смещения производится экстраполяция координат объекта на следующих кадрах.

### 3.4. Алгоритм

На вход система получает видеопоследовательность с одной камеры. Каждый видеокادر последовательно обрабатывается в цикле. Сначала для нового кадра запускается процесс распознавания. Результатом работы детектора является набор ограничивающих прямоугольников. Каждому такому прямоугольнику присваивается тип объекта (например, машина, автобус или грузовик) и определенная в процессе локализации и классификации вероятностная оценка. Используемая версия детектора распознает 80 разных типов объектов, поэтому после локализации и классификации лишние виды отбрасываются. Далее каждому найденному на кадре объекту сопоставляется номер `track_id`. Каждый объект на изображении уникально определяется парой натуральных чисел `object_id` (вид объекта), `track_id` (номер слежения). Результатом работы механизма слежения также является множество ограничивающих прямоугольников, для которых известен тип объекта и номер сопровождения (`track_id`). Результаты детектора и слежения сравниваются. Если детектор находит объект с `object_id` и `track_id`, который возвращается еще и фреймворком слежения, то правдивым прямоугольником считается результирующий прямоугольник, определенный детектором, а соответствующий ему результат механизма слежения отбрасывается. Так объединяются результаты работы детектора и алгоритма сопровождения. Далее набор прямоугольников детектора передается для обработки фреймворку слежения. В качестве точки со-

прикосновения объекта с дорогой выбирается левая нижняя вершина ограничивающего прямоугольника. После этого с использованием данных о фокусном расстоянии камеры и ее высоты над дорогой для всех объектов вычисляется расстояние до них. Затем описанный процесс повторяется для следующего кадра видеопоследовательности. Основные шаги алгоритма схематично показаны на рисунке (4).

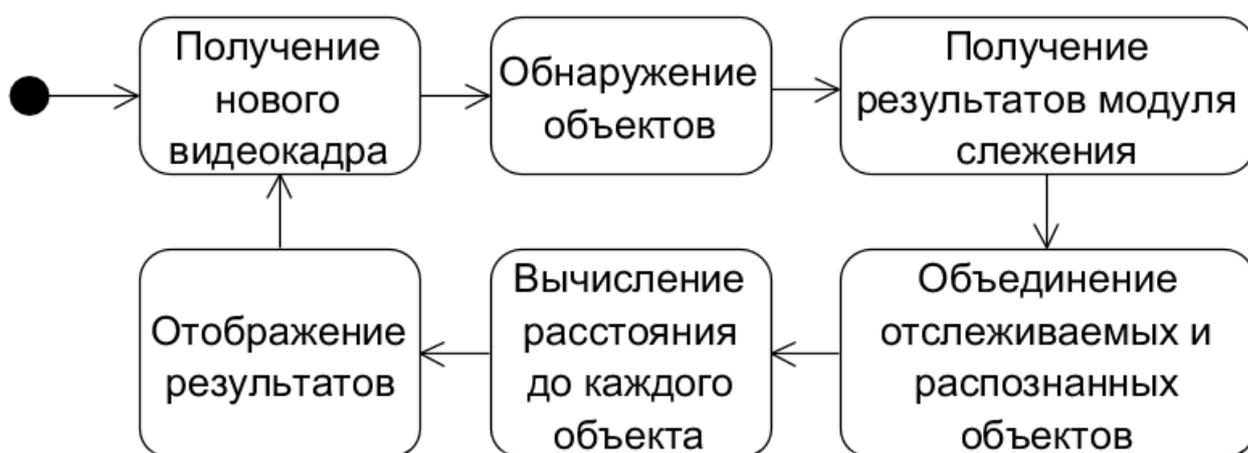


Рис. 4: Схема алгоритма

### 3.5. Апробация



Рис. 5: Результат 1



Рис. 6: Результат 2

Работа системы была проверена на видео из тестового набора KITTI [8] (рис. 5, 6). Результаты работы системы на тестовых данных выложены в общий доступ на youtube канале.

## 4. Заключение

В рамках данной работы были выполнены следующие задачи:

- проведен обзор и сравнение систем распознавания объектов по изображению;
- проведен обзор и сравнение подходов и алгоритмов решения задачи слежения;
- проведен обзор систем определения дистанции до объекта с использованием монокулярного зрения;
- разработан алгоритм распознавания, слежения и определения дистанции до объектов на дороге на основе монокулярного зрения;
- реализована система распознавания, слежения и определения дистанции до объектов на дороге на основе монокулярного зрения;
- работа системы проверена на примерах тестового набора данных KITTI;

## Список литературы

- [1] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. Ssd: Single shot multibox detector. — 2016.
- [2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. — 2016.
- [3] Yolo home page. — <https://pjreddie.com/darknet/yolo/>. — Accessed on 01.09.2018.
- [4] Opencv home page. — <https://opencv.org/>. — Accessed on 01.09.2018.
- [5] Peter Janku, Karel Koplik, Tomas Dulik, and Istvan Szabo. Comparison of tracking algorithms implemented in opencv // MATEC Web of Conferences. — 2003.
- [6] J. Venu Gopala Krishnan, Dr. N. Manoharan, Dr. B. Sheela Rani. Estimation of distance to texture surface using complex log mapping // Journal of Computer Applications. — 2010.
- [7] G. P. Stein, O. Mano, A. Shashua. Vision-based acc with a single camera: Bounds on range and range rate accuracy. — 2003.
- [8] Kitti dataset. — <http://www.cvlibs.net/datasets/kitti/>. — Accessed on 01.09.2018.