

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
SUMMARY OF A GRADUATION THESIS**

Обучающийся / Student Смирнова Валентина Сергеевна

Факультет/институт/клUSTER/ Faculty/Institute/Cluster факультет информационных технологий и программирования

Группа/Group M42381c

Направление подготовки/ Subject area 01.04.02 Прикладная математика и информатика

Образовательная программа / Educational program Программирование и искусственный интеллект 2020

Язык реализации ОП / Language of the educational program Русский

Статус ОП / Status of educational program

Квалификация/ Degree level Магистр

Тема ВКР/ Thesis topic Определение скорости ТС безрадарным методом

Руководитель ВКР/ Thesis supervisor Фильченков Андрей Александрович, кандидат физико-математических наук, Университет ИТМО, факультет информационных технологий и программирования, доцент (квалификационная категория "ординарный доцент")

Консультант не из ИТМО / Third-party consultant Евдокимов Лев Александрович, ООО "Софт Импульс", Директор, Бакалавр, Бакалавр

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
DESCRIPTION OF THE GRADUATION THESIS**

Цель исследования / Research goal

Разработать алгоритм определения скорости безрадарным методом

Задачи, решаемые в ВКР / Research tasks

а) реализация rc-парсера (выходного формата прибора); б) генерация синтетического датасета; в) разработка алгоритма определения скорости; г) внедрение алгоритма в тестовый прибор;

Краткая характеристика полученных результатов / Short summary of results/findings

Удалось разработать и внедрить алгоритм определения скорости безрадарным методом с допустимой погрешностью.

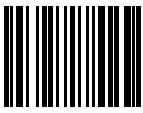
Обучающийся/Student

Документ подписан	
Смирнова Валентина Сергеевна	
27.05.2022	
(эл. подпись/ signature)	

Смирнова
Валентина
Сергеевна
(Фамилия И.О./ name

and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Фильченков Андрей Александрович	
26.05.2022	

(эл. подпись/ signature)

Фильченков
Андрей
Александрович

(Фамилия И.О./ name
and surname)

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
GRADUATION THESIS

Определение скорости ТС безрадарным методом

Обучающийся / Student Смирнова Валентина Сергеевна

Факультет/институт/клластер/ Faculty/Institute/Cluster факультет информационных технологий и программирования

Группа/Group M42381c

Направление подготовки/ Subject area 01.04.02 Прикладная математика и информатика

Образовательная программа / Educational program Программирование и искусственный интеллект 2020

Язык реализации ОП / Language of the educational program Русский

Статус ОП / Status of educational program

Квалификация/ Degree level Магистр

Руководитель ВКР/ Thesis supervisor Фильченков Андрей Александрович, кандидат физико-математических наук, Университет ИТМО, факультет информационных технологий и программирования, доцент (квалификационная категория "ординарный доцент")

Консультант не из ИТМО / Third-party consultant Евдокимов Лев Александрович, ООО "Софт Импульс", Директор, Бакалавр, Бакалавр

Обучающийся/Student

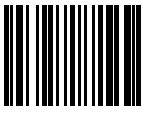
Документ подписан	
Смирнова Валентина Сергеевна	
27.05.2022	

(эл. подпись/ signature)

Смирнова
Валентина
Сергеевна

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Фильченков Андрей Александрович	
26.05.2022	

(эл. подпись/ signature)

Фильченков
Андрей
Александрович

(Фамилия И.О./ name
and surname)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. Обзор возможностей комплекса «КОРДОН.Про»М	8
1.1. Определения и ключевые понятия	8
1.2. Содержание RC-файла	8
1.3. Возможности прибора	12
1.4. Входные данные для исследования	15
Выводы по главе 1	15
2. Обзор существующих решений	17
2.1. Базовая геометрическая оптика	17
2.2. Компьютерное зрение	18
Выводы по главе 2	20
3. Предложенное решение	22
3.1. Сбор входных данных. RC-парсер	22
3.2. Объект мониторинга	23
3.3. Алгоритм определения скорости	24
3.4. Генерация синтетических данных	27
3.4.1. Объект разработки	28
3.4.2. GLTF-формат	28
3.4.3. Gplay	30
Выводы по главе 3	31
4. Результаты экспериментов	32
4.1. Данные для анализа	32
4.2. Матчинг радарных данных	32
4.3. Оценка погрешности	33
4.4. Результаты	33
4.4.1. Реальные данные	33
4.4.2. Синтетические данные	33
Выводы по главе 4	34
ЗАКЛЮЧЕНИЕ	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36
ПРИЛОЖЕНИЕ А. Описания форматов	38
ПРИЛОЖЕНИЕ Б. Результаты	41

ВВЕДЕНИЕ

На сегодняшний день фиксация нарушений правил дорожного движения почти полностью автоматизирована и осуществляется посредством специальных приборов, в их числе приборы производства компании «Симикон»:

- Комплекс «КОРДОН-КРОСС»
- Комплекс «КОРДОН.ПРО»М
- Комплекс «КОРДОН.ПРО»В
- Комплекс «ПАРКОН-А»
- Комплекс «КОРДОН-М»
- Система контроля «ЗПИ»
- Приспособление «МИРАЖ»
- Комплекс «КОРДОН-ТЕМП»
- Комплекс «КОРДОН-М”КР»
- Видеорегистратор «ГРОМ-1»
- Имитатор «ИС-24/3»
- Имитатор «ИС-24»Д



Рисунок 1 – Комплекс «КОРДОН.Про»М в стационарном режиме — на столбе.

Для фиксации нарушений скоростного режима приборы в обязательном порядке оборудованы радаром и камерой. Радар служит для измерения скорости транспортного средства с точностью до 1 км/ч, а камера — для распознавания Государственного регистрационного знака (далее — номер или номерной знак) и предоставления доказательства нарушения. Полученная с камеры и радара информация обрабатывается на приборе и формируется коллаж с информацией о проезде. После чего прибор, в соответствии с ограничениями на участке, принимает решение о том, был ли данный проезд нарушением или нет и отправляет нарушения в соответствующие органы.



Рисунок 2 – Комплекс «КОРДОН.Про»М в передвижном режиме — на триноге.

Основные приборы, используемые для фиксации скоростных нарушений — комплексы семейства «КОРДОН», далее в работе под «приборами» будем подразумевать именно их. КОРДОНЫ оборудованы радаром, камерой, ИК-проектором и непосредственно железом, на котором производятся вычисления и хранятся нарушения. Данные приборы могут работать в 3 режимах: стационарный (Рисунок 1), передвижной (Рисунок 2) и мобильный (Рисунок 3). В работе мы будем иметь в виду стационарный режим. Далее можно будет масштабировать предложенное решение на оставшиеся режимы. Технические характеристики прибора представлены в таблице 1.



Рисунок 3 – Комплекс «КОРДОН.Про»М в мобильном режиме — на заднем сидении.

Так как использование радара в приборах — достаточно затратно, а наличие камеры в приборе обязательно для распознавания номера, формирования коллажа и предоставления доказательства нарушения, то закономерно возникает задача определения скорости транспортного средства безрадарным методом исключительно по данным с камеры прибора, другими словами — оценка скорости транспортного средства по видео.

Таблица 1 – Технические характеристики комплекса «КОРДОН.Про»М.

ПАРАМЕТР	ЗНАЧЕНИЕ
Диапазон измеряемых скоростей	2 - 300 км/ч
Пределы допускаемой абсолютной погрешности измерений скорости	± 1,0 км/ч
Пределы допускаемой абсолютной погрешности синхронизации внутренней шкалы времени с UTC(SU)	± 5 мкс
Потребляемая мощность (при положительных температурах)	не более 25 Вт
Масса датчика	не более 6,0 кг
Габаритные размеры	не более 460×180×280 мм

ГЛАВА 1. ОБЗОР ВОЗМОЖНОСТЕЙ КОМПЛЕКСА «КОРДОН.ПРО»М

В данной главе мы введём ключевые понятия, подробно рассмотрим технологии, по которым на сегодняшний день работает прибор семейства «КОРДОН» и рассмотрим существующие решения для определения скорости объекта по видео.

1.1. Определения и ключевые понятия

Для начала введем определения, ключевые понятия и некоторые сокращения, которые будут использоваться в работе:

- **транспортное средство (ТС)** — любой участник дорожного движения, обладающий Государственным регистрационным знаком (автомобиль, мотоцикл, автобус, грузовик)
- **Государственный регистрационный знак (ГРЗ)** — номерная пластина на транспортном средстве, соответствующая ГОСТу Р 50577-2018 «Знаки Государственные Регистрационные Транспортных Средств»
- **правила дорожного движения (ПДД)** — свод правил, регулирующих обязанности участников дорожного движения (водителей транспортных средств, пассажиров, пешеходов и так далее), а также технические требования, предъявляемые к транспортным средствам, для обеспечения безопасности дорожного движения, актуальные на момент написания работы
- **нарушение** — любое нарушение правил дорожного движения, однако в данной работе под «нарушением» будем понимать нарушение скоростного режима на рассматриваемом участке

Также необходимо определить, что выход прибора — не просто видеофайл, а специальный формат .rc, который содержит в себе всю информацию с прибора, как, например, битовые представления кадров, радарные данные, проекционная матрица камеры, время, информация о распознанных номерах, данные о GPS, карты глубины, данные о настройках камеры и другое. В данной работе мы будем обрабатывать именно этот формат, поэтому, рассмотрим его подробнее.

1.2. Содержание RC-файла

RC-файл представляет из себя битовый файл, состоящий из frame-ов (рисунок 4). Каждый фрейм описывает 1 кадр из записи. В начале каждого фрейма

находятся 2 поля: `magic` и `length`, за ними идёт непосредственно информация о фрейме — `frame data`. Поле `magic` служит исключительно для проверки корректности чтения, оно имеет константное значение `0x11223307` для всех фреймов. Поле `length` определяет длину `frame data`(включая `magic` и `length`), то есть сколько последующих бит будет относиться к текущему фрейму.

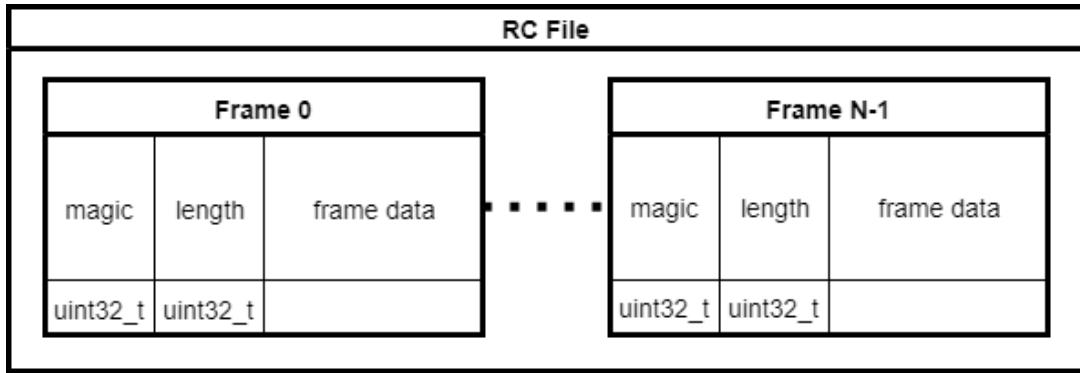


Рисунок 4 – Представление RC-файла.

Каждый фрейм содержит в себе набор данных соответствующих типов, схема представлена на рисунке 5. В начале каждого фрагмента данных присутствует 2 поля: `id` и `length`. Поле `id` определяет тип читаемых данных во фрагменте, это может быть примитивный тип, как, например, `timestamp` или целые структуры, как, например, `sCameraData`. Подробное описание каждого типа представлено в приложении А.3. Поле `length` аналогично одноимённому полю в RC описывает длину последующей последовательности.

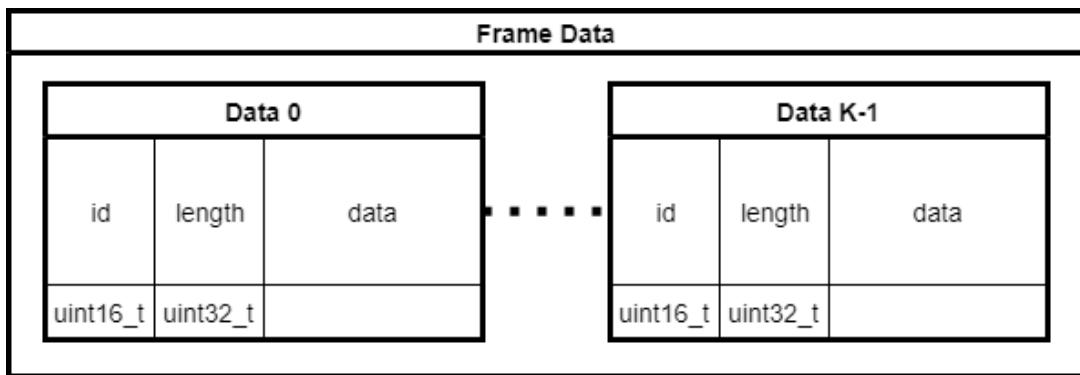


Рисунок 5 – Представление frame-a.

В таблицах А.1 и А.2 приложения представлено подробное описание к рисункам 4 и 5 соответственно. Теперь подробнее остановимся на 2 типах данных из таблицы А.3 — информации о распознанных номерах и радарных целях.

Результаты распознавания номера Результаты распознавания номера соответствуют ID=5 из таблицы А.3 и представляют собой массив типа TRectNumber. Реализация структуры описана в листинге 1.

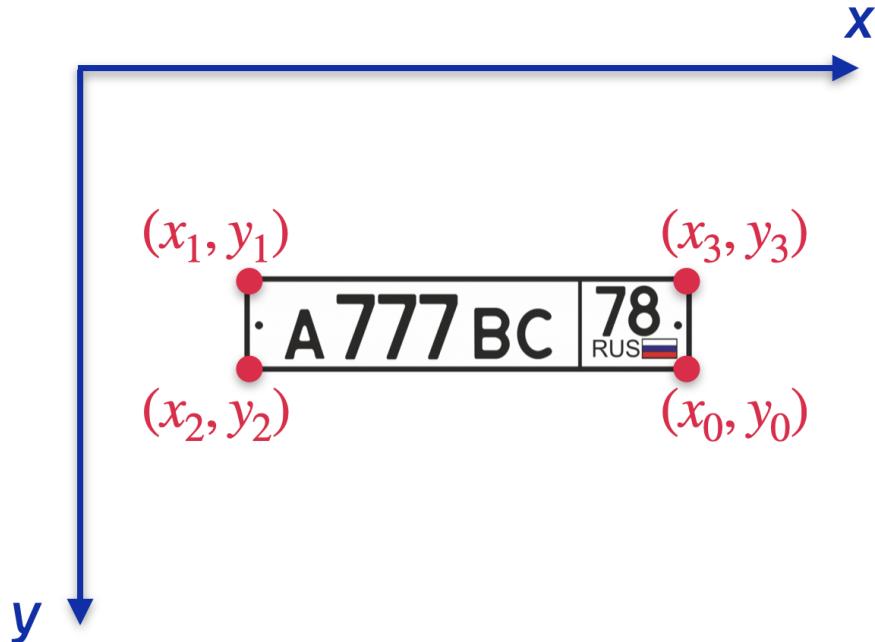


Рисунок 6 – Координаты номера в структуре TRectNumber.

Листинг 1 – Реализация структуры TRectNumber.

```
#define MAX_SYMBOL_NUM (16)

typedef int16_t TInt3[4];

typedef struct TRectNumber
{
    int numFormat;
    int n_symbols;
    uint16_t text16[MAX_SYMBOL_NUM];

    int16_t allCert;
    int16_t certList[MAX_SYMBOL_NUM];
    TInt3 x, y;

} TRectNumber;
```

Здесь numFormat — тип номерной рамки, складывается из страны и формата (однострочный/двустрочный), n_symbols — количество распо-

знанных символов, `text16` — непосредственно распознанный текст в кодировке `UINT-16`, `allCert` — общая вероятность распознавания, `certList` — массив вероятностей распознавания каждого символа, `x` и `y` — массивы координат распознанного номера, порядок отображён на рисунке 6.

Цели с радара Результаты распознавания номера соответствуют `ID=8` из таблицы А.3 и представляют собой массив типа `umrr_target`. Реализация структуры описана в листинге 2.

Листинг 2 – Реализация структуры `umrr_target`.

```
class umrr_target
{
public:
    std::string doString() const;
    double absSpeed() const;

    int id;
    double x;
    double y;
    double xspeed;
    double yspeed;
    double len;

    // calc values
    double imgx;
    double imgy;
    int numw;

    double imgx_left;
    double imgx_right;

    double imgy_top;
    double imgy_bottom;
} __attribute__((__packed__));
```

Здесь `id` — идентификационный номер радарной цели от 0 до 63 (количество одновременно отслеживаемых целей), `x` — координата `x` в плоскости дороги, `y` — координата `y` в плоскости дороги, `xspeed` — мгновенная скорость по оси `x`, `yspeed` — мгновенная скорость по оси `y`, `len` — длина транспортного средства. Система радарных координат располагается в основании столба, на котором установлен прибор, и считается по правилу правой руки, визуализация представлена на рисунке 7.

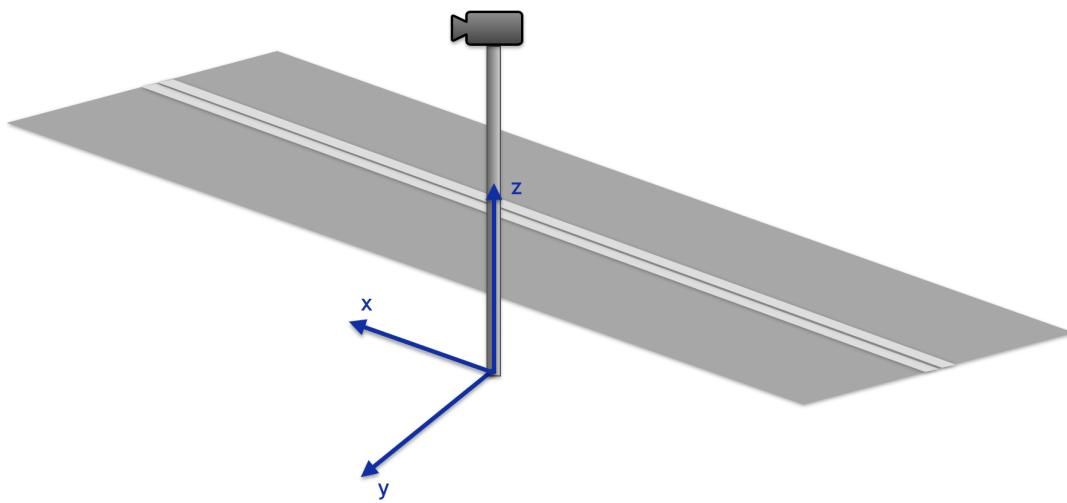


Рисунок 7 – Расположение системы радарных координат.

1.3. Возможности прибора

Для того, чтобы начать разработку нового решения, необходимо разобраться с тем, как устроен работающий прибор. Во введении мы определили основные характеристики и возможности прибора. Теперь чуть более подробно рассмотрим основные функции и алгоритм его работы, чтобы определить, какую информацию можно использовать для разработки. Ниже представлен перечень всех функций и возможностей универсального комплекса «КОРДОН.Про»М:

Автоматическая фотовидеофиксация

- Автоматическая фотовидеофиксация нарушений ПДД в зоне контроля до шести полос движения одновременно в обоих направлениях.
- Измерение скоростей в диапазоне от 2 до 300 км/ч.
- Возможность измерения средней скорости на протяженных участках совместно с любыми другими комплексами семейства «Кордон».
- Возможность фиксации нарушения «Непредоставление преимущества пешеходу» при наличии дополнительного блока «Мера».
- Возможность дооснащения обзорной камерой для работы со знаками переменной информации.
- Отдельные пороги скорости для разных полос движения и для ТС категорий «В», «С» и «D».
- Автоматическое сохранение фотоматериалов и видеоролика по каждому зафиксированному нарушению.

- Модуль ГЛОНАСС/GPS с автоматической коррекцией системного времени комплекса.

Распознавание номерных знаков и розыск ТС

- Автоматическое распознавание номерных знаков многих стран мира, включая двустрочные номера и российские ГРЗ нового образца по ГОСТ Р 50577-2018.
- Возможность включения и выключения распознавания ГРЗ тех или иных государств.
- Проверка распознанных номеров по базам данных с оповещением оператора и голосовым озвучиванием номера.
- Специальный режим работы «Перехват» для розыска ТС с отсутствующими или нечитаемыми номерными знаками.

Классификация ТС

- Автоматическое определение типа ТС и их классификация по четырём основным категориям (легковые, грузовые, автобусы, среднегабаритные).
- Автоматическое присвоение соответствующей категории ТС порога скорости по ПДД.
- Автоматический контроль запрета движения ТС для заданной категории (грузовые, автобусы и т.д.) по отдельным полосам или по дороге в целом.

Видеонаблюдение

- Трансляция видеопотока с высоким разрешением по протоколу RTSP.
- Ведение непрерывной видеозаписи с возможностью скачивания видеоролика по заданному промежутку времени.

Передача данных

- Передача данных на сервер ЦОД по зашифрованным проводным или беспроводным каналам связи (3G/4G).
- Автоматическое переключение на резервные каналы связи (Wi-Fi, 4G) при сбоях или отказе основного канала.
- Возможность параллельной передачи данных с комплекса на различные серверы.

Защита и безопасность

- Защита данных и встроенного ПО от несанкционированных изменений.
- Экспортируемые данные защищены ЭЦП.
- Ведение журнала событий и действий пользователя комплекса.
- Передача уведомлений по SMS и электронной почте о зафиксированных фактах физических воздействий на прибор (удары, вибрация).
- Возможность защиты от огнестрельного оружия с использованием бронированного кожуха, сертифицированного на пулестойкость по классам «Ер2» (пистолеты СПС и ТТ) и «С1» (охотниче ружье со свинцовой пулей).

Телеметрия и диагностика

- Самодиагностика, удаленная диагностика.
- Автоматическое отслеживание параметров комплекса и передача телеметрической информации в режиме реального времени.

Установка

- Автоматическая проверка правильности монтажа комплекса.
- Поворотный кронштейн для быстрой стационарной установки на опоре.
- Удобный и простой веб-интерфейс для настройки.
- Различные варианты подключения к сетям электропитания. Возможность подключения к осветительной сети и обеспечения бесперебойной работы комплекса от АКБ.

Работа в ночное время

- Встроенная инфракрасная подсветка для работы в ночное время.
- Дополнительный внешний ИК- прожектор для гарантированного определения марки ТС по изображению.

Статистика

- Сбор статистических данных об интенсивности транспортного потока.
- Построение интерактивных графиков по выбранным статистическим параметрам.
- Анализ зафиксированных нарушений ПДД с разбивкой по видам нарушений и величине превышения скорости.

1.4. Входные данные для исследования

Из секции 1.3 и описания RC-файла в секции 1.2 можно вынести следующую полезную информацию, которая будет использоваться в дальнейшем в работе:

- а) из информации об установке прибора в стационарном режиме:
 - 1) высоту столба — H
 - 2) расстояние до дороги — R
 - 3) угол между направлением движения и направлении камеры — α
- б) из информации о самом приборе:
 - 1) физические размеры матрицы — высота h_m и ширина w_m в миллиметрах
 - 2) разрешение — высота h_m и ширина w_m в пикселях
 - 3) фокусное расстояние — f
- в) из RC-файла:
 - 1) проекционная матрица камеры 3×4 — `matrix`
 - 2) данные по распознанным номерам — `licnum`
 - 3) радарные данные — `radar_targets`

Выводы по главе 1

В первой главе мы подробно разобрали устройство и возможности прибора на примере комплекса «КОРДОН.Про»М. Стоит отметить, что примерно все приборы семейства «КОРДОН» обладают схожими характеристиками и возможностями. Например, любой прибор можно дооборудовать цветной камерой, дополнительным прожектором, источником питания или бронированным кожухом. Принципиальное отличие «КОРДОН.Про»М и «КОРДОН.Про»В в наличии радара, в «В» версии он отсутствует, что удешевляет производство, однако использовать такой прибор для фиксации скоростных нарушений пока не возможно. Данный факт непосредственно подтверждает актуальность внедрения решения данной работы.

Кроме того, в данной главе мы определили необходимые для вычислений переменные и структуры, которые будут использованы в последующих главах. Также, был подробно рассмотрен основной формат данных для исследования — RC-файл, его содержание, подробное описание каждого типа данных и содержащихся в них структур. Стоит отметить, что данный формат будет использоваться как в качестве тренировочных данных (имеются в виду

записи с различных приборов в разных городах и при разных условиях в формате .rc), так и для внедрения решения непосредственно в реальный прибор.

ГЛАВА 2. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

На сегодняшний день существует несколько подходов к решению задачи определения скорости по видео. В основном, используются технологии компьютерного зрения и линейная алгебра. В данной главе мы рассмотрим наиболее перспективные подходы, их преимущества и недостатки и наметим план решения.

2.1. Базовая геометрическая оптика

Один из самых простых подходов [2] основывается на базовой геометрической оптике [4]. Вспомним некоторые базовые определения:

- а) *фокус собирающей линзы* — это место, где все проходящие через линзу лучи света пересекаются;
- б) *фокусное расстояние собирающей линзы* f — это отрезок от принятого центра линзы до её фокуса;
- в) *объектом* называется физическое тело, участвующее в оптической системе. Обычно объект располагается перед линзой;
- г) *изображением* называется проекция объекта на экран или матрицу (располагается после линзы);
- д) *расстоянием до объекта* d называется отрезок между объектом и линзой;
- е) *расстоянием до изображения* f называется отрезок между изображением и линзой.

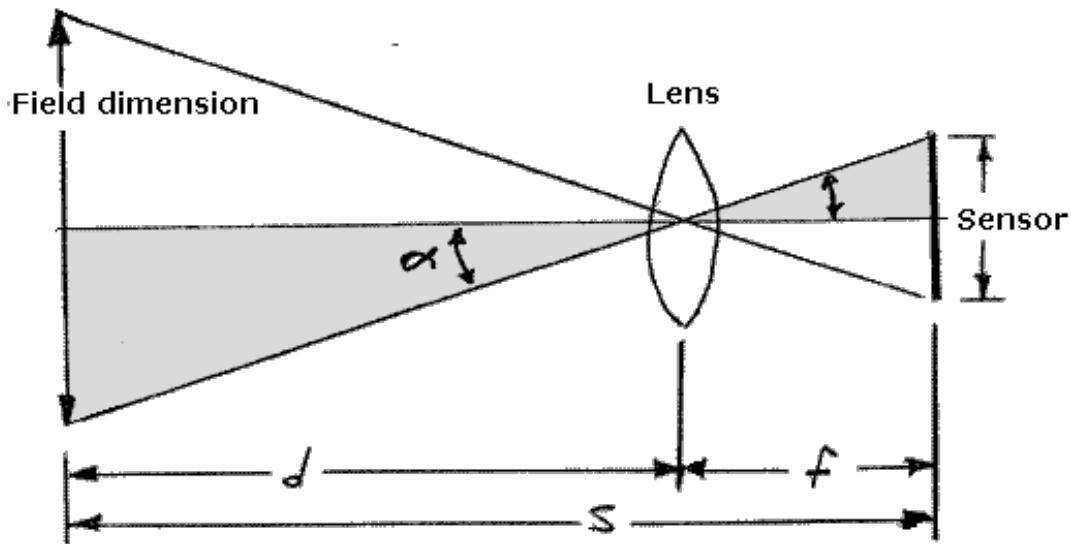


Рисунок 8 – Схема собирающей линзы.

Из законов базовой оптики известно, что зная физические размеры объекта и экрана (в нашем случае в качестве экрана выступает матрица камеры), также, зная фокусное расстояние линзы, можно найти расстояние до объекта [1]. На рисунке 8 представлена схема собирающей линзы, а формулой 1 описывается выше упомянутое отношение в общем виде.

$$\frac{\text{Sensor_dimension(mm)}}{\text{Focal_length(mm)}} = \frac{\text{Field_dimension(mm)}}{\text{Distance_to_Field(mm)}} \quad (1)$$

Если подставить вместо размера экрана (*Field_dimension*) размер объекта, то можно получить выражения 2 и 3:

$$\text{Object_height_on_sensor(mm)} = \frac{\text{Sensor_height(mm)} \cdot \text{Object_height(pixels)}}{\text{Sensor_height(pixels)}} \quad (2)$$

$$\text{Object_height_on_sensor(mm)} = \frac{\text{Real_Object_height(mm)} \cdot \text{Focal_length(mm)}}{\text{Distance_to_object(mm)}} \quad (3)$$

Путём простейшей подстановки из данных формул можно вывести формулу нахождения расстояния от камеры до объекта 4.

$$\text{Distance_to_object(mm)} = \frac{\text{Real_Object_height(mm)} \cdot \text{Focal_length(mm)} \cdot \text{Sensor_height(mm)}}{\text{Sensor_height(mm)} \cdot \text{Object_height(pixels)}} \quad (4)$$

В статье [2] описано решение, основанное именно на этом подходе и погрешность составляет 4%. Если переводить это на километры, то погрешность составит от 1 до 12 км/ч при заявленном диапазоне скоростей от 0 до 300 км/ч, что больше допустимой погрешности, поэтому данное решение недостаточно точное для данного исследования.

2.2. Компьютерное зрение

Остальные перспективные решения [6] основываются на работе с изображениями, а именно разбиением транспортного средства на составляющие и привязкой к одной из них.

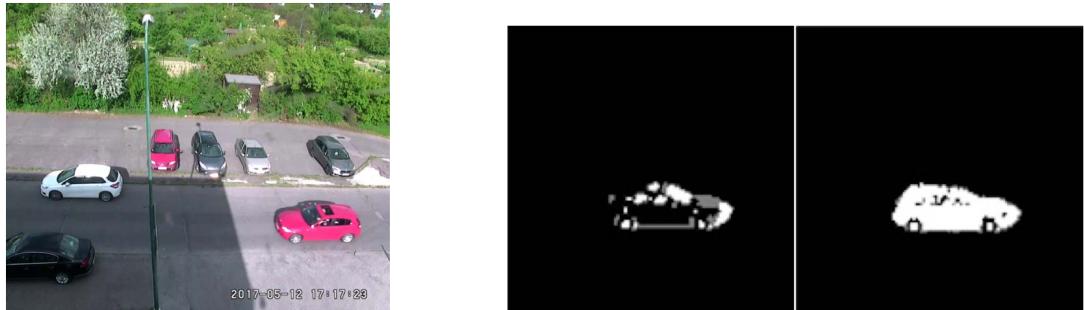


Рисунок 9 – Вырезка из источника [9], детект сбоку.

Например, в источнике [9] скорость автомобиля определяется сбоку (рисунок), то есть применены допущения, что камера установлена перпендикулярно дороге, что в реальном мире невозможно, поэтому данное решение также нам не подходит.



Рисунок 10 – Вырезка из источника [10], детект спереди под углом.

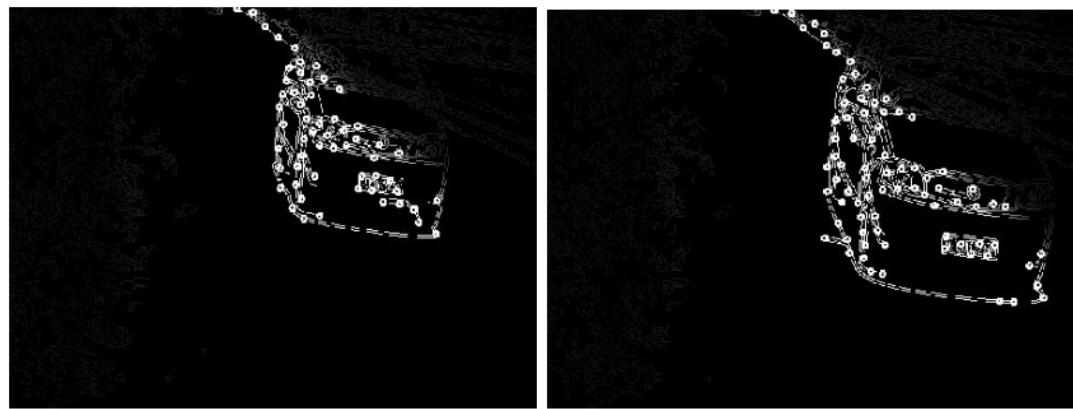


Рисунок 11 – Вырезка из источника [10], разбиение автомобиля на точки.

В другом источнике [10] машина разбивается на точки кривых и детект [7] ведётся по крайней из этих точек. Очевидно, что из-за сложной формы автомобиля, цвета и бликов, крайние точки могут меняться, что вызовет боль-

шую погрешность. Не смотря на подходящий для нашей работы ракурс и заявленную точность определения скорости до 2 км/ч, на это число нельзя ориентироваться, так как исследование [10] проводилось всего на нескольких видео и данных недостаточно для сбора подобной статистики. Также в этой статье были приняты некоторые допущения при вычислениях, как например замена поправочного коэффициента камеры на константу, зависимую от координат пикселя, что также недопустимо.

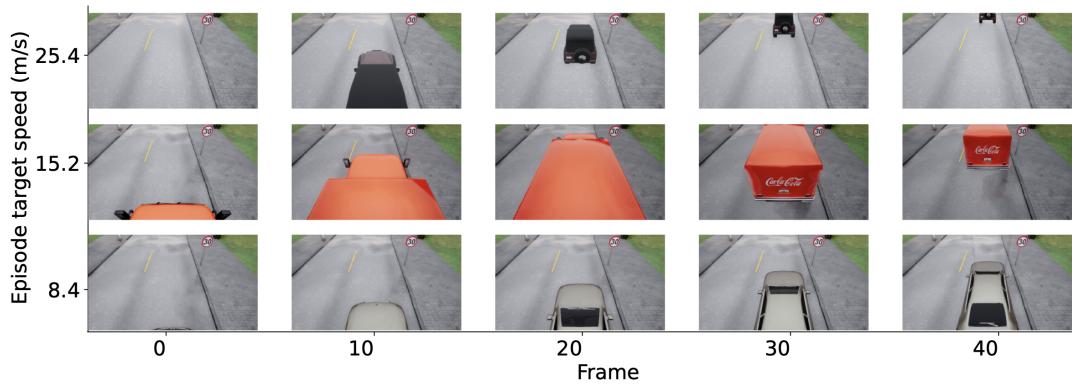


Рисунок 12 – Вырезка из источника [5], пример синтетических данных.

Ещё один популярный подход [5] основан на расположении камеры непосредственно над дорогой. Из данного положения, принятые некоторые допущения, исходя из которых детекция транспортного средства может проводится исключительно по прямой траектории по одной полосе движения. Во-первых, расположение камеры над дорогой не является универсальным решением, так как далеко не на всех участках дороги это возможно. Во-вторых, мы хотим детектировать до 30 целей одновременно, а на одной полосе это невозможно. Однако так как данное решение основано на синтетических данных (рисунок), то есть видеопоток является вручную сгенерированной 3D сценой, возьмём эту технику на заметку.

Выводы по главе 2

В данной главе мы рассмотрели существующие подходы к решению задачи нахождения скорости по видео. Максимально достигнутая точность определения скорости — 2 км/ч, однако нельзя полагаться на данный результат, так как в статье [10] присутствуют некоторые неточности и не произведено достаточное количество экспериментов для получения статистически значимого результата.

Наиболее правдоподобный подход обеспечивает погрешность до 7 км/ч, однако он основывается на расположении камеры непосредственно над движущимся транспортным средством, что не позволит измерять скорость на многолосной дороге (как заявлено в характеристиках нашего прибора) и при перестройках, а только на одной полосе.

Кроме того, в нескольких из рассмотренных подходов используются синтетические данные, что готовит о том, что подобный подход к формированию датасета имеет место быть.

ГЛАВА 3. ПРЕДЛОЖЕННОЕ РЕШЕНИЕ

В данной главе мы разобьём задачу определения скорости по видео на более мелкие подзадачи и приведём решение каждой из них.

3.1. Сбор входных данных. RC-парсер

В главе 1 мы вывели некоторый список входных данных, связанных с характеристиками прибора, которые необходимо определить отдельно для каждого RC-файла, например, фокусное расстояние, размеры матрицы камеры, высота установки прибора. Для этого соберём некоторый набор реальных данных, записанных на приборах при различных условиях. Для этого воспользуемся базой заранее записанных файлов, рисунок 13.

Название записи	Тип прибора	Матрица	Фокус	Положение	Расстояние от дороги	Высота установки	Ширина полосы	Кол-во полос	Ширина разделительной полосы	Угол поворота радара	Автобусы	Грузовики	Мотоциклы/Спецтехника	Дополнител
1527685066_2018.05.30_15-57-46.rc Скачать RC	Стационарный	Imx249	50	Дорога справа	1,4	7,7	3,7	4	0	-13	1	0	0	
1531482918_2018.07.13_14-55-18.rc Скачать RC	Стационарный	Imx249	50	Дорога справа	0,5	7,9	3,7	4	0	-8	0	1	0	синхронная запись в период 1531482920_2018.07.13_1

Рисунок 13 – WEB-интерфейс базы с RC-файлами.

Так как работать с RC-файлами достаточно неудобно, возникла задача извлечения релевантных для исследования данных и преобразования их в удобный формат. Для этого, на языке Python был реализован парсер RC-формата в привычный JSON (пример представлен на листинге 3). Подробнее остановимся на его содержании.

Поля `seconds` и `microseconds` описывают время в тиках прибора с начала UNIXTime. Поле `licenums` представляет собой массив с информацией о распознанных номерах. Поле `radar_targets` описывает радарные цели. Важно отметить, что в текущем состоянии не известно, какая радарная

Листинг 3 – Пример JSON-файла.

```
{
    "frames": [
        {
            "seconds": 1596722143,
            "microseconds": 379221,
            "licnums": [
                {
                    "format": 42139649,
                    "text": "МТВ684186",
                    "x": [450, 399, 400, 449],
                    "y": [39, 30, 42, 27]
                }
            ],
            "radar_targets": [
                {
                    "id": 40,
                    "x": 60.608,
                    "y": 4.5439,
                    "xspeed": -21.4,
                    "yspeed": -0.3,
                    "carlen": 4.0
                },
                ...
            ]
        },
        {
            "matrix_type": 0.0055,
            "focal_length": 35.0,
            "matrix3x4": [ 865.751108, -6388.572118, -544.713764,
                -301.569121, -841.281125, 344.588298, -6313.435574,
                54640.980513, 0.978972, -0.032996, -0.20131, -1.268419 ]
        }
    ]
}
```

цель какому распознанному номеру соответствует, эта проблема будет подобна разобрана в секции 4.2. Поля `matrix_type` и `focal_length` описывают тип матрицы камеры и её фокусное расстояние соответственно. Из типа матрицы можно получить её физические размеры: ширину, высоту в пикселях и миллиметрах и разрешение. И, наконец, поле `matrix3x4` хранит в себе проекционную матрицу камеры 3x4, она понадобится нам в дальнейшем для восстановления координат.

3.2. Объект мониторинга

В главе 2 мы определили, что разбиение автомобиля на части в конечном итоге является неточным способом для определения скорости, так как могут появляться блики, части автомобиля могут быть закрыты, деформированы, сливаться с окружающей средой. Поэтому возникает вопрос: к какой точке

автомобиля необходимо привязываться, чтобы следить за её перемещением с максимальной точностью. В качестве такой точки был выбран номерной знак. Основные преимущества данного выбора заключаются в следующем:

- a) для любого кадра, где распознан номер, известны координаты 4 углов этого номера с точностью до нескольких сантиметров, а в случаях, когда номер не распознан, формирование нарушения невозможно;
- b) для каждого номера известен его точный физический размер согласно ГОСТу (рисунок 14).



Рисунок А.1 — Регистрационный знак типа 1 с двухзначным кодом региона регистрации

Рисунок 14 – Габариты Государственного регистрационного знака согласно актуальному на момент написания работы ГОСТу.

3.2.0.1. Сбор датасета Государственных регистрационных знаков

Так как в работе мы будем использовать габаритные размеры номерной пластины, а их существует большое множество (рисунок 15), был собран датасет, который для каждого типа номерной пластины описывает её габаритные размеры согласно ГОСТу.

3.3. Алгоритм определения скорости

Для начала дадим определение скорости. *Скорость* — векторная физическая величина, характеризующая быстроту перемещения и направление движения материальной точки относительно выбранной системы отсчёта. По определению, равна производной радиус-вектора точки по времени. В СИ измеряется в метрах в секунду.

$$v = \frac{dr}{dt} \quad (5)$$

Так как точное время для каждого фрейма известно, остаётся найти перемещение объекта, чью скорость мы собираемся измерить и так как в секции

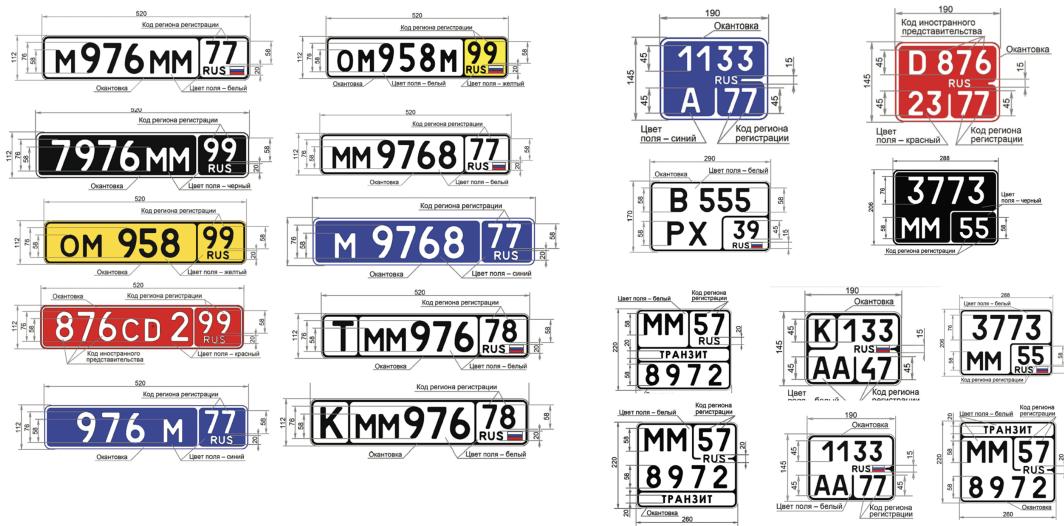


Рисунок 15 – Типы Государственных регистрационных знаков.

3.2 было принято решение наблюдать за номером, задача сводится к нахождению перемещения номерного знака между кадрами.

Согласно законодательству, для формирования штрафа, необходимо предоставить доказательную базу в виде коллажа с изображением транспортного средства и номерного знака с измеренной на момент кадра *моментальной* скоростью. Данное ограничение приводит нас к выводу, что перемещение необходимо измерять между 2 ближайшими фреймами. Теперь перейдём непосредственно к решению.

Рассмотрим положение номера в двух моментах времени — t_1 и t_2 . Схема представлена на рисунке 16.

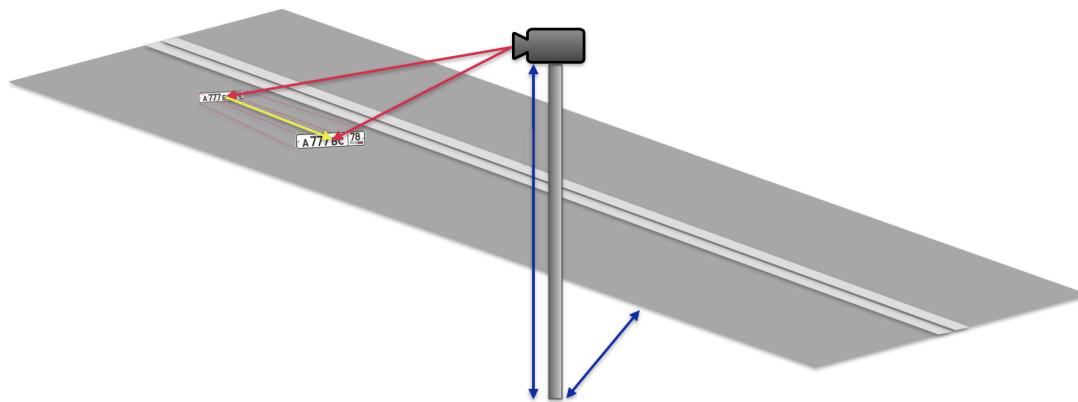


Рисунок 16 – Схема перемещения номерного знака.

Из оптических отношений, рассмотренных в части 2.1, зная физические размеры матрицы камеры, фактический размер объекта и фокусное расстоя-

ние, можно определить расстояние до объекта, то есть длины красных векторов на рисунке 16.

Из векторной геометрии очевидно, что чтобы найти перемещение r достаточно найти разность векторов a и b , однако для нахождения разности векторов, необходимо дополнительно знать либо угол между ними, либо координаты обоих векторов, а ни того, ни другого у нас нет, поэтому возникает промежуточная задача нахождения координат этих векторов в пространстве.

Так как мы рассчитали расстояние до центра номера, чтобы восстановить координаты проведённого вектора достаточно будет найти координаты номерной пластины в пространстве. Для этого решим задачу триангуляции.

Приведение задачи триангуляции Классическая задача триангуляции [3] восстанавливает 3D координаты [8] объекта по изображениям с 2 камер. В общем случае уравнение триангуляции 6 представляет собой простое отношение, где x — координаты объекта на изображении, P — проекционная матрица камеры 3×4 , X — координаты объекта в пространстве.

$$x = PX \quad (6)$$

Так как векторы x и PX имеют одно и то же направления, их векторное произведение будет равняться нулю (уравнение 7).

$$x \times PX = 0 \quad (7)$$

Раскрывая предыдущие уравнения, получаем следующее 8:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (8)$$

Путём некоторых преобразований получаем уравнения 9 для двух камер.

$$\begin{bmatrix} y p_3^T - p_2^T \\ p_1^T - x p_3^T \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} y' p_3'^T - p_2'^T \\ p_1'^T - x' p_3'^T \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (9)$$

Однако в нашем случае всего лишь одна камера и решить второе уравнение представляется невозможным. Зато мы знаем целых 4 координаты но-

мерной пластины и её точные физические размеры согласно ГОСТу. Эта информация позволяет компенсировать отсутствие второй камеры путём вывода одних координат через другие посредством формул для Декартова расстояния 10.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (10)$$

Введём некоторое допущение, что номер движется по направлению движения, это позволит сократить координаты y и z в уравнении 10, что тем самым позволит выразить координаты одной точки номера через другую. Тогда система наших уравнений превращается в однородную.

$$\begin{bmatrix} y p_3^T - p_2^T \\ p_1^T - x p_3^T \\ (d - y) p_3^T - p_2^T \\ p_1^T - (d - x) p_3^T \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (11)$$

Путём решения системы уравнений посредством сингулярного разложения (SVD), восстановим координаты номера в пространстве. Теперь для левой верхней точки номера мы знаем её координаты в пространстве и можем за ней следить. Остаётся только посчитать координаты для каждого фрейма и определить перемещение путём вычисления декартового расстояния в трёхмерном пространстве.

3.4. Генерация синтетических данных

Для качественной оценки работы алгоритма оценки скорости по видео, необходимо иметь достаточно большой набор данных. Кроме этого, распределение скоростей в этих данных должно располагаться в заявленном диапазоне, для наших приборов этот диапазон составляет промежуток от 0 до 300 км/ч. Так как все реальные данные собраны с тестовых приборов в черте города, где транспортные средства в большинстве своём движутся со скоростями 40-80 км/ч, то среди реальных данных почти невозможно найти примеры с очень большими (более 100 км/ч) или очень маленькими (менее 20 км/ч) скоростями. Поэтому возникает задача генерации синтетических данных с недостающими примерами для полноценной оценки работы алгоритма.

3.4.1. Объект разработки

Прежде, чем приступать к процессу разработки, необходимо определить объект этой разработки. Цель данной секции – разработать программу (или связку программ/сервисов), на выходе которой мы получим RC-файл, идентичный по характеристикам с реальными записями. Теперь перейдём непосредственно к процессу разработки.

Основная часть RC-файла — битовые представления изображений с камеры прибора, из которых строится видео, другими словами, видео — это последовательный набор изображений. Изображение, в свою очередь, является проекцией объектов пространства на матрицу камеры. Это значит, что для создания идентичного реальному RC-файлу набора данных, необходимо воссоздать 3D-сцену и её проекцию на виртуальную камеру.

После тщательного исследования существующих инструментов и форматов для работы с 3D-сценами, было принято решение воспользоваться программой Blender для создания 3D-сцен и форматом gltf для их последующей обработки. На рисунке 17 представлен пример созданной 3D-сцены в среде разработки Blender. Теперь подробнее остановимся на формате gltf.

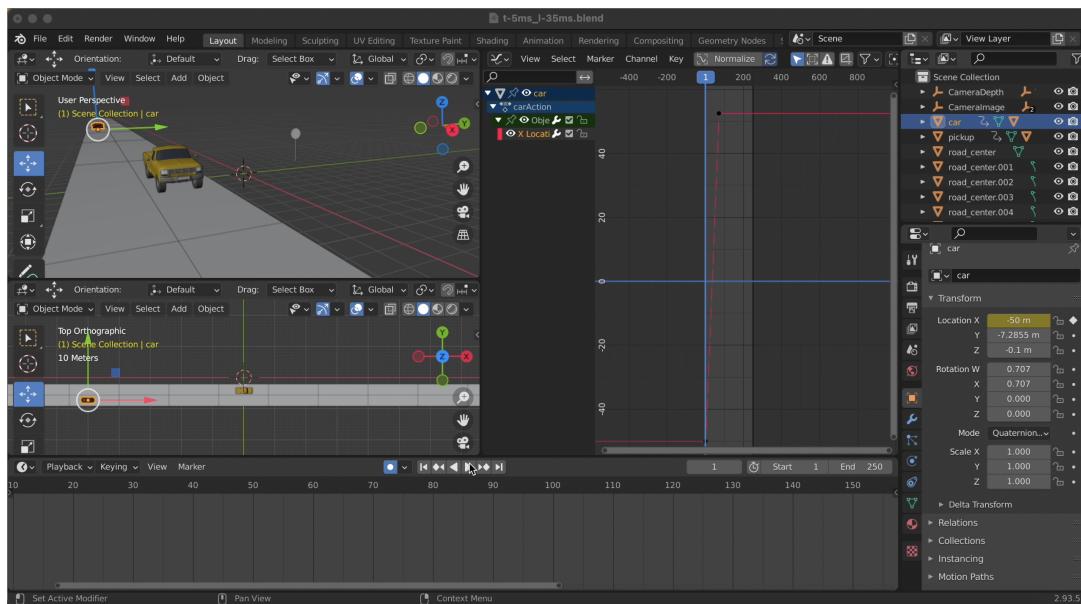


Рисунок 17 – Пример созданной 3D-сцены в среде разработки Blender.

3.4.2. GLTF-формат

Формат gltf представляет собой текстовый файл, напоминающий json, только хранит в себе не примитивные типы, а полную информацию о 3D-сцене, такую как формы объектов, их координаты, информация о материалах,

текстурные карты, описание анимации, информация о камере (её положение, настройки) и другое. В нашем случае, основными преимуществами данного формата являются простая читаемость, полнота информации, возможность задания любых параметров. А преимуществом работы в программе Blender является то, что можно легко экспортить созданную сцену в необходимый gltf формат.

Таким образом мы можем получить готовый gltf-файл с необходимыми нам параметрами, как, например, скорость и госномер. Однако, создавать новую 3D-сцену для каждого набора параметров – весьма ёмкая задача, поэтому возникает вопрос автоматизации данного процесса. Посредством модуля bpy для языка Python, можно работать с теми же 3D-сценами, задавая или изменения параметры, тем самым автоматизируя процесс генерации данных. На листинге 4 представлен пример кода, подставляющий госномер в виде текстуры из файла. Аналогичный код был написан для подстановки типа и скорости анимации, далее эти отрывки были помещены в цикл и сгенерированы необходимые gltf-файлы.

Листинг 4 – Пример скрипта для подстановки ГРЗ из файла на объект в 3D-сцене.

```
import bpy
from bpy import context, data, ops

mat = bpy.data.materials.new(name="New_Mat")
mat.use_nodes = True
bsdf = mat.node_tree.nodes["Principled BSDF"]
texImage = mat.node_tree.nodes.new('ShaderNodeTexImage')
texImage.image = bpy.data.images.load("/Users/tina/dev/simicon/
plates/Y999YY25.jpg")
mat.node_tree.links.new(bsdf.inputs['Base Color'], texImage.
outputs['Color'])

ob = context.view_layer.objects.active

# Assign it to object
if ob.data.materials:
    ob.data.materials[0] = mat
else:
    ob.data.materials.append(mat)
```

3.4.3. Gplay

Итак, мы получили набор gltf-файлов со всеми интересующими нас параметрами, закономерно возникает вопрос: как преобразовать этот формат в необходимый нам полноценный RC. Для этой задачи был разработан сервис gplay, который берёт на вход gltf-файл и по описанной в нём анимации и информации и сцене генерирует RC-файл исключительно с информацией о картинках, то есть видео. Между тем, некоторые из необходимых нам типов данных никак не описаны в gltf-файле, а, соответственно, и в выходном RC. Речь идёт про радарные данные и информацию о распознавании номеров. Решается данная проблема достаточно просто: для восстановления радарных данных можем воспользоваться известными нам координатами объектов в 3D-сцене (известны из gltf-файла) путём простого пересчёта координат в расстояние до камеры; для заполнения информации о распознанных номерах просто пропустим запись через эмулятор с сервисом распознавания номеров, который дополнит наш RC-файл. На рисунке 18 можно увидеть итоговый RC-файл с заполненными радарными данными и распознанными номерами.

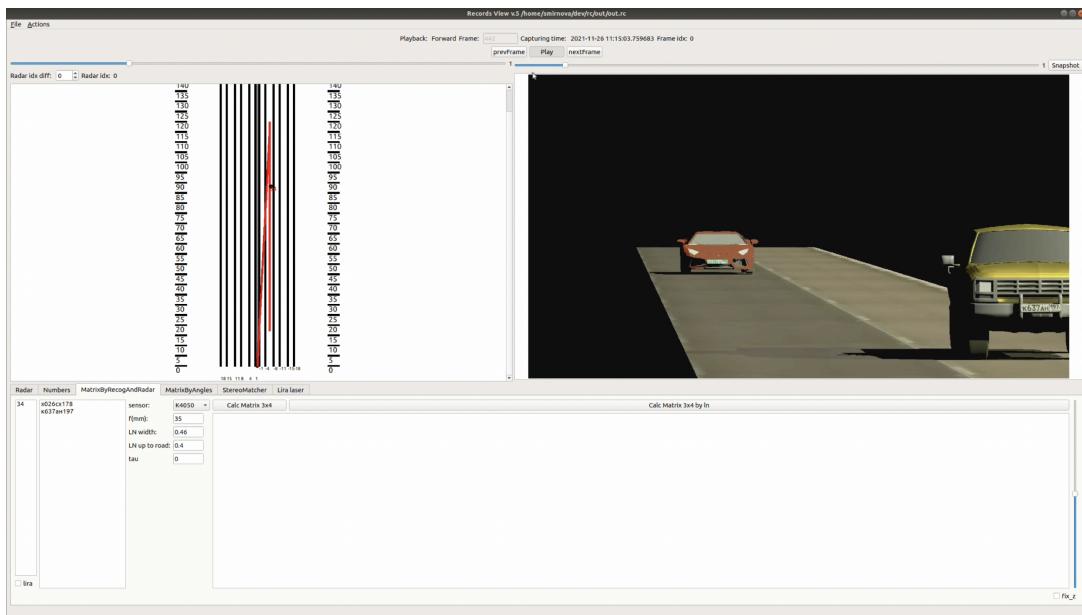


Рисунок 18 – Итоговый RC-файл в программе RecordView.

RecordsView — внутренний инструмент компании «Симикон» для просмотра RC-записей. Справа на изображении находится видео, слева — радарные данные, снизу — распознанные номера и возможность задать параметры камеры и рассчитать матрицу 3x4.

Таким образом, из связки программ, представленных на схеме 19, где в Blender мы создаём основу для 3D-сцены и скриптами генерируем gltf-файлы, далее пропускаем через сервис `gplay`, который конвертирует gltf в rc, далее пропускаем через сервис `hwrecog` для распознавания номеров, он дополняет rc и, наконец, RecordsView для расчёта проекционной матрицы, мы получаем наборы данных с необходимыми нам параметрами, что позволяет нам спокойно переходить к экспериментам.

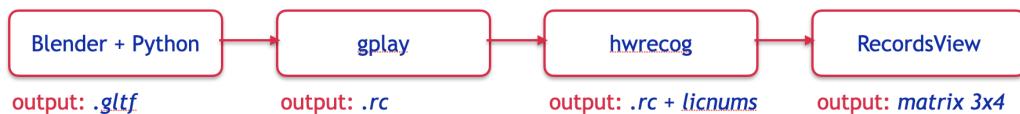


Рисунок 19 – Схема связки сервисов для генерации синтетического датасета.

Выводы по главе 3

В данной части был описан процесс сбора реальных данных для исследования, также был определён объект наблюдений, то есть точка, на чьё перемещение мы будем смотреть. В качестве такого объекта был выбран номер, так как его координаты мы умеем распознавать с точностью до сантиметра. Также был подробно описан и обоснован процесс генерации синтетических данных для оценки качества алгоритма.

Кроме данных для оценки в данной главе был описан непосредственно алгоритм, по которому будет вычисляться скорость. В качестве основы решения была взята задача триангуляции, вторая камера компенсирована дополнительной информацией о номерном знаке.

ГЛАВА 4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

После того, как скорость транспортного средства посчитана, необходимо провести оценку качества. Вспомним, что заявленная скорость прибора от 0 до 300 км/ч, это значит, что предложенное в данной работе решение должно корректно работать на всём диапазоне. В данной главе мы предложим способ оценки полученных результатов в соответствии с техническим заданием и заявленными характеристиками.

4.1. Данные для анализа

В секции 3.1 были собраны реальные записи, каждой из записей соответствует свой набор параметров, таких как количество полос, направление движения (встречное/попутное), высота установки, тип матрицы и т.д. Однако далеко не все параметры было возможно получить, поэтому недостающие данные были дополнены синтетическими.

Среди реальных данных максимальная радарная скорость составляла 52.5 км/ч, а минимальная — 8.5 км/ч, поэтому диапазоны скоростей от 1 до 8 и от 53 до 299 были заполнены синтетическими данными. На этапе разработки именно радарная скорость будет использована для оценки качества.

4.2. Матчинг радарных данных

Так как радар — отдельная составляющая прибора, как, собственно, и камера, хоть и работают они в связке, всё равно независимо друг от друга, поэтому мы не можем с точностью сказать, какая радарная цель какому транспортному средству соответствует. Отсюда возникает задача матчинга радарных целей с распознанными номерами.

По триангулированным координатам номера мы можем получить положение этого номера в координатах дороги. Радарные цели также представлены в координатах дороги. Остаётся лишь определить ближайшие к радарным координатам координаты номера. Для этого пройдёмся по всем радарным целям, посчитаем декартово расстояние (формула 10) между координатами номера и радара и найдём ближайший ответ. В данном случае координату z номера можно опустить, так как нас интересует плоскость дороги. Матчинг происходит на этапе формирования структуры, описывающей трек каждого распознанного номера.

4.3. Оценка погрешности

В данной работе оценивать погрешность стандартными методами будет не совсем корректно, так как, во-первых, в техническом задании требуется получить погрешность *по модулю*, не превышающую 2 км/ч, во-вторых, конечная цель исследования — выписывать скоростные нарушения предложенным методом.

Из правил дорожного движения известно, что нарушение может быть выписано, при зафиксированной скорости более 19 км/ч от установленного на участке ограничения. Поэтому можно положительными результатами будем считать результаты, где посчитанная скорость по модулю *превышает* 19 км/ч. А погрешностью будем считать среднее по модулю расхождение радарной скорости с рассчитанной.

4.4. Результаты

В данной секции приведены результаты экспериментов отдельно для реальных и синтетических данных, рассчитана погрешность в соответствии с подсекцией 4.3 данной главы.

4.4.1. Реальные данные

Результаты экспериментов представлены в таблице Б.1. Количество целей — это количество транспортных средств на записи, чьи номера были распознаны; фреймов на цель — количество фреймов, для которых оценивалась скорость; погрешность представляет из себя среднее по модулю расхождение радарной скорости с рассчитанной, переведённое в километры в час; и, наконец, часть выписанных ложных нарушений от всех рассчитанных скоростей.

Из результатов видно, что погрешность не превышает 2 км/ч, что соответствует ограничению, установленному в техническом задании. При этом среднее количество ложных нарушений составляет 1.5%, что также является допустимым, хотя в дальнейшей работе ложные нарушения планируется исключить полностью.

4.4.2. Синтетические данные

Со сгенерированными данными результат получился намного точнее в силу "идеальности" данных. На рисунке 20 представлен график зависимости

рассчитанной на синтетических данных скорости от реальной и проведена аппроксимирующая прямая. Как несложно заметить, почти все точки лежат на ней.

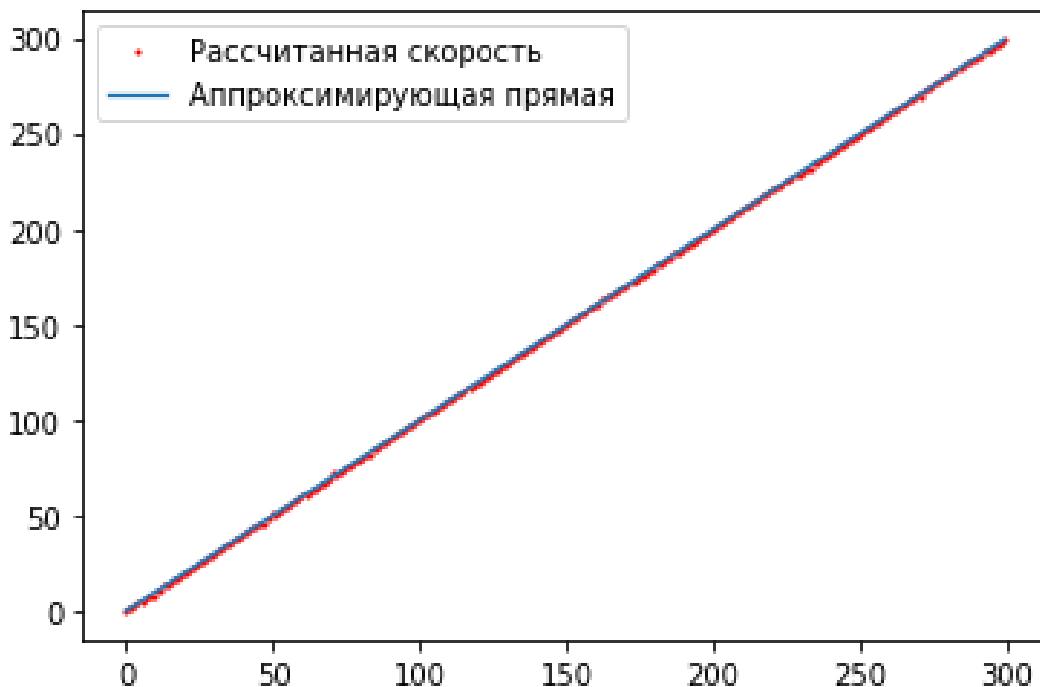


Рисунок 20 – График зависимости рассчитанной скорости от реальной.

Средняя погрешность на синтетических данных составляет 0.1 км/ч, а ложно определённые нарушения отсутствуют. Их этих данных можно сделать вывод, что погрешность, вычисленная на реальных данных вызвана окружающими условиями, а не качеством алгоритма.

Выводы по главе 4

В данной части был описан подход к анализу полученных данных и определены необходимые для этого данные. Был описан алгоритм матчинга радарных целей с рассчитанными скоростями. В качестве "эталона" для вычисленной скорости была взята ближайшая радарная цель.

В результате анализа полученных данных на реальном датасете, была получена средняя погрешность в 0.7 км/ч, при этом ложно зафиксированные нарушения составляют 1.5%. Полученные результаты с запасом вписываются в техническое задание и соответствуют заявленным ограничениям.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены современные приборы фиксации нарушений правил дорожного движения и определены объект и цель работы. Как самый популярный прибор, для исследования был выбран "КОРДОН" вся текущая работа писалась под него и внедрена была также на этот прибор.

В качестве вспомогательных к работе материалов, были написаны скрипты для генерации синтетических данных в среде Blender, парсер для RC-формата прибора. В качестве сервисов совместной разработки выступили `gplay` и `RecordsView` — сервис для генерации RC-файла из gltf-сцены и программа для просмотра RC-файлов соответственно.

За основу алгоритма определения скорости взята задача триангуляции с некоторыми изменениями, такими как компенсация отсутствия второй камеры уравнением, описывающим габариты номерного знака.

После оценки результатов была получена погрешность 0.7 км/ч для реальных данных, что меньше 2 км/ч, заявленных в техническом задании, поэтому можно считать что задание выполнено. Кроме этого мы можем подтвердить корректность работы алгоритма на всём диапазоне скоростей, заявленных в характеристиках прибора. Также, можно вывести, что погрешность на реальных данных вызвана окружающими условиями, так как на сгенерированных данных она практически отсутствует.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Антонов В.* Линейная алгебра и аналитическая геометрия. // Опорный конспект: Учебник. — 2011. — 144 С..
- 2 An algorithm for automatic vehicle speed detection using video camera / J. Wu [et al.] // 2009 4th International Conference on Computer Science Education. — 2009. — P. 193–196. — DOI: 10.1109/ICCSE.2009.5228496.
- 3 *Avidan S., Shashua A.* Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2000. — Vol. 22, no. 4. — P. 348–357. — DOI: 10.1109/34.845377.
- 4 *D. Wilkinson T.* Geometric Optics // Handbook of Visual Display Technology / ed. by J. Chen, W. Cranton, M. Fihn. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. — P. 23–44. — ISBN 978-3-540-79567-4. — DOI: 10.1007/978-3-540-79567-4_2. — URL: https://doi.org/10.1007/978-3-540-79567-4_2.
- 5 Data-driven vehicle speed detection from synthetic driving simulator images / A. H. Martínez [et al.] // CoRR. — 2021. — Vol. abs/2104.09903. — arXiv: 2104.09903. — URL: <https://arxiv.org/abs/2104.09903>.
- 6 *Gawande U., Hajari K., Golhar Y.* Pedestrian Detection and Tracking in Video Surveillance System: Issues, Comprehensive Review, and Challenges // Recent Trends in Computational Intelligence / ed. by A. Sadollah, T. S. Sinha. — Rijeka : IntechOpen, 2020. — Chap. 9. — DOI: 10.5772/intechopen.90810. — URL: <https://doi.org/10.5772/intechopen.90810>.
- 7 *Makris D., Ellis T.* Path detection in video surveillance // Image and Vision Computing. — 2002. — Oct. — Vol. 20. — P. 895–903. — DOI: 10.1016/S0262-8856(02)00098-7.
- 8 *Siswantoro J., Prabuwono A. S., Abdullah A.* Real World Coordinate from Image Coordinate Using Single Calibrated Camera Based on Analytic Geometry // . Vol. 378. — 08/2013. — ISBN 978-3-642-40566-2. — DOI: 10.1007/978-3-642-40567-9_1.

- 9 Vehicle speed detection from camera stream using image processing methods / J. Gerát [et al.] // 2017 International Symposium ELMAR. — 2017. — P. 201–204. — DOI: 10.23919/ELMAR.2017.8124468.
- 10 Vehicle Speed Detection Using Corner Detection / K. K. Kumar [et al.] // 2014 Fifth International Conference on Signal and Image Processing. — 2014. — P. 253–258. — DOI: 10.1109/ICSIP.2014.46.

ПРИЛОЖЕНИЕ А. ОПИСАНИЯ ФОРМАТОВ

Таблица А.1 – Описание содержания RC-файла..

Имя	Тип	Длина	Описание
magic	uint32_t	4	0x11223307
length	uint32_t	4	Общая длина фрейма, включая поля magic и length
frame data	-	length - 8	-

Таблица А.2 – Описание содержания frame-a.

Имя	Тип	Длина	Описание
id	uint16_t	2	Тип описываемого поля
length	uint32_t	4	Длина данных
data	-	length	

Таблица А.3 – Расшифровка поля ID из frame_data.

ID	Данные	Тип/формат данных	Описание
0	frame.capturingtime	struct timeval	Время съемки кадра
3	frame.idx	int32_t	ID кадра
4	frame.frameticks	uint32_t	Время съемки кадра в значениях 90КГц счетчика
5	Результаты распознавания ГРЗ	массив TRectNumber'ов	Распознанные с кадра номерные знаки, достоверности распознавания и координаты
8	Цели с радара	массив umrr_target'ов	Tracked цели с радара UMRR
14	Ширина исходного изображения	int32_t	Ширина исходного изображения
15	Высота исходного изображения	int32_t	Высота исходного изображения
19	Ширина картинки	int32_t	Ширина картинки
20	Высота картинки	int32_t	Высота картинки

21	Размера изображение	int32_t	Размер изображения в байтах (должен быть равен размеру поля 23)
22	Формат картинки	string	JPEG, GRAYSCALE, YUV422P и т.д.
23	Картина	данные	непосредственно сама картинка
24	GPS + данные о рейке	MapString	Данные о GPS (COGDEG=342.190002 DDLAT=59.92576700 DDLON=30.40147000 DDRADDR=559751168 HDOP=0.620000 SOGMPS=0.000000 SRC=ACTIVE SRC_TIME=FPGA) и данные рейки (REIKADISTANCE=60 REIKALEN=1.79 REIKALENINPIXELS=74 REIKAPOINTS=0,0,0,0)
51	Количество карт глубины	uint32_t	Количество карт глубины на один фрейм (Nmaps)
51	Матрица 3x4	array<double, 12>	Матрица 3x4
52	Размер одной карты глубины	uint32_t	Размер одной карты глубины в байтах (Dmap)
53	Карты глубины	uint16_t []	Массив последовательно идущих данных карт глубины
54	Преобразование	uint16_t []	Карта соответствия координат карты глубины координатам изображения фрейма
55	Данные камеры	struct sCameraData	Данные о настройках камеры в момент формирования кадра

65	frame.radar_model	enum class RadarModel: uint32_t	Модель радара
----	-------------------	---------------------------------------	---------------

ПРИЛОЖЕНИЕ Б. РЕЗУЛЬТАТЫ

Таблица Б.1 – Результаты экспериментов на реальных данных.

Имя файла	Коли-чество целей	Фрей-мов на цель	Погреш-ность (м/с)	Погреш-ность (км/ч)	Процент ложных нару-шений (false/all)
1580735812 2020.02.03 16-16-52	124	16	0.254809	0.92	0.007965
1597823270 2020.08.19 12-47-50 outgoint	67	10	0.371781	1.34	0.001629
1593092046 2020.06.25 16-34-06	30	38	0.062644	0.23	0.007929
1649766196 2022.04.12 15-23-16	206	107	0.045629	0.16	0.019732
1581423064 2020.02.11 15-11-04	175	19	0.345780	1.24	0.042837
1597823299 2020.08.19 12-48-19 outgoint	57	11	0.414410	1.49	0.004505
1648122446 2022.03.24 14-47-26	196	94	0.044238	0.16	0.015602
1580735555 2020.02.03 16-12-35	352	15	0.281440	1.01	0.014039

1596722130 2020.08.06 18-55-30	62	16	0.293159	1.06	0.004854
1649767179 2022.04.12 15-39-39	238	126	0.037682	0.14	0.009250
1644564468 2022.02.11 10-27-48	41	92	0.053859	0.19	0.012733
1597767405 2020.08.18 21-16-45	61	9	0.372369	1.34	0.002481
1645438993 2022.02.21 13-23-13	451	75	0.075862	0.27	0.030237
1644995982 2022.02.16 10-19-42	29	45	0.086472	0.31	0.008555
1581422841 2020.02.11 15-07-21	243	21	0.313150	1.13	0.039691