

Санкт-Петербургский государственный университет

СЕМЕНОВ Александр Сергеевич

Выпускная квалификационная работа

Построение и отслеживание модели границ дороги по видеопотоку

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование
информационных систем»*

Основная образовательная программа *СВ.5006.2018 «Математическое обеспечение и
администрирование информационных систем»*

Научный руководитель:
доц. каф. СП, к.т.н. Ю.В. Литвинов

Консультант:
старший инженер-программист ООО «Харман» М.С. Осечкина

Рецензент:
ведущий инженер-программист ООО «Харман» М.В. Ермакова

Санкт-Петербург
2022

Saint Petersburg State University

Semenov Aleksandr Sergeevich

Bachelor's Thesis

Building and tracking a model of road boundaries by video stream

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5006.2018 "Software and Administration of Information Systems"*

Scientific supervisor:

Docent of Software Engineering, C.Sc Y.V. Litvinov

Consultant:

Senior Software Engineer at "Harman" M. Osechkina

Reviewer:

Lead Software Engineer at "Harman" M. Ermakova

Saint Petersburg
2022

Оглавление

1. Введение	4
2. Постановка задачи	6
3. Обзор	7
3.1. Обзор моделей дороги	7
3.2. Обзор методов отслеживания модели	12
4. Выбор модели дороги	14
4.1. Критерии отбора примитивов	14
4.2. Выбор примитивов	14
5. Программная реализация	17
6. Тестирование	20
6.1. Способ тестирования	20
6.2. Тестирование на “идеальных” контурах	20
6.3. Тестирование на реальных контурах	21
7. Возможные улучшения алгоритма	23
Заключение	24
Список литературы	25

1. Введение

С развитием информационных технологий, они все чаще применяются в тех сферах человеческой жизни, которые сопряжены с риском и в которых большую роль играет человеческий фактор. Такие системы призваны уменьшить влияние этого фактора, в следствии чего предполагается снижение несчастных случаев и других нежелательных исходов.

Одной из областей, в которых нужна и уже активно применяется автоматизация, является передвижение на автомобиле, для которой существует множество интеллектуальных систем. Примерами таких систем могут быть системы ADAS¹ – системы помощи водителю при движении. Такие системы работают благодаря различным датчикам, установленным на автомобиле, например, камеры, лидара, гироскопа, акселерометра и прочих.

Важной особенностью такой системы является способность работать в режиме реального времени и без предварительных подготовок. Помимо этого, особенность оборудования на автомобиле накладывает некоторые ограничения на производительность алгоритмов, поэтому это стоит учитывать, ведь от быстроты работы алгоритма в конечном счете может зависеть и человеческая жизнь.

В составляющие системы ADAS входят такие функции, как отслеживание дорожных знаков, измерение расстояния до других объектов, предупреждение водителя при приближении аварийной ситуации и др. Также, одна из основных возможностей системы – предупреждение водителя при пересечении границ полосы движения. Эта функция важна не только непосредственно для водителя, чтобы понимать, что он все еще находится на своей полосе движения, но и для корректной работы других составляющих системы. Помимо этого, можно отслеживать не только свою полосу движения, но и остальные. Это уже позволит отслеживать границы всей дороги, а это в свою очередь позволит системе определять, например, как автомобиль может перестраиваться. Такая

¹Advanced driver-assistance systems

функция может быть основой для более продвинутой системы умного автомобиля – беспилотного вождения.

Существуют подходы, которые частично решают проблему отслеживания модели дороги. Например, в работе [1] предложен алгоритм, который распознает полосы движения на основе статистической модели, которая нуждается в предварительном обучении. А работы [4, 6] используют в своих решениях нейронные сети. Также есть методы, которые не нуждаются в предварительных подготовках, например, [7, 8, 9, 10]. Однако эти подходы не строят модель дороги, а просто находят границы полос на каждом кадре видео. Из-за этого может ухудшаться точность распознавания и становится невозможным отслеживание этих полос.

Поэтому разработка алгоритма отслеживания модели границ дороги по видеопотоку в режиме онлайн является актуальной задачей как для существующих систем, так и для будущего их развития. Планируется разработать прототип с открытым исходным кодом, который в дальнейшем может быть применен для разработки системы ADAS компанией “Харман”.

2. Постановка задачи

Целью работы является написание алгоритма построения и отслеживания модели границ дороги по входному потоку контуров дороги.

Для достижения цели были поставлены следующие задачи:

1. Изучить существующие подходы к решению этой задачи.
2. Выбрать наиболее подходящие примитивы для модели.
3. Реализовать наиболее подходящий алгоритм решения.
4. Провести тестирование полученного решения на реальных данных.

3. Обзор

Одна из основных и самых частых сфер применения компьютерного зрения – системы помощи водителю. Поэтому далее рассматриваются методы, которые либо частично решают поставленную задачу, либо не решают ее, но некоторые идеи из них были бы полезны в решении. Все ниже рассмотренные статьи были найдены в поисковой системе по научным публикациям Google Scholar. В рассмотрение брались только статьи, описывающие подходы, которые работают только с камерой, без вспомогательных устройств, например, дальномера или акселерометра. Всего по запросам “road model”², “road recognition”³, “marking line”⁴ и “road model tracking”⁵ было найдено около четырех миллионов статей. Найденные статьи ранжировались по релевантности поисковой системой, которая учитывает полный текст статьи, рейтинг автора, рейтинг издания, в котором статья опубликована, количество цитат. Поэтому первые ссылки в результатах поиска ведут на более цитируемые статьи. Из этого количества статей было выбрано пятнадцать самых цитируемых. Далее девять статей были отброшены как нерелевантные. В итоге осталось шесть публикаций, которые были изучены в рамках обзора.

3.1. Обзор моделей дороги

3.1.1. Примитивные подходы

В работе “Распознавание линий разметки полос движения на основе видеоряда с использованием весовой классификации сегментов” авторы предлагают распознавать линии дорожной разметки при помощи классификации отдельных сегментов по углу наклона и подобию однородных сегментов [7]. Авторы отмечают, что линии дорожной разметки выделяются контрастностью по сравнению с остальной дорогой. Поэтому в данном подходе для выделения разметки используется информа-

²модель дороги

³распознавание дороги

⁴линия разметки

⁵отслеживание модели дороги

ция о контрастности. Далее для каждой полученной линии определяется длина, угол наклона и точка пересечения линии с горизонтальной осью. Далее с помощью этой информации строится диаграмма, которая показывает распределение сегментов по углу наклона как показано на Рис. 1, где по горизонтали – значение угла, а по вертикали – количество найденных линий.

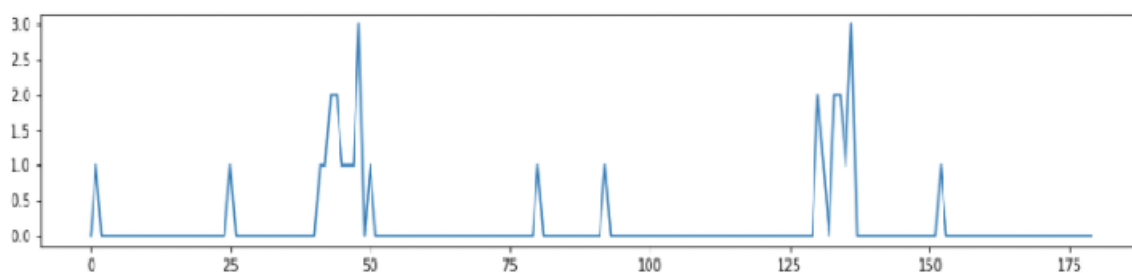


Рис. 1: Распределение сегментов по углу наклона

Далее эти линии домножаются на веса “соседних” элементов, которые зависят от длины сегмента и угла наклона. Благодаря этому отсекаются ненужные линии, оставляя только те, которые соответствуют дорожной разметке, как показано на Рис. 2.

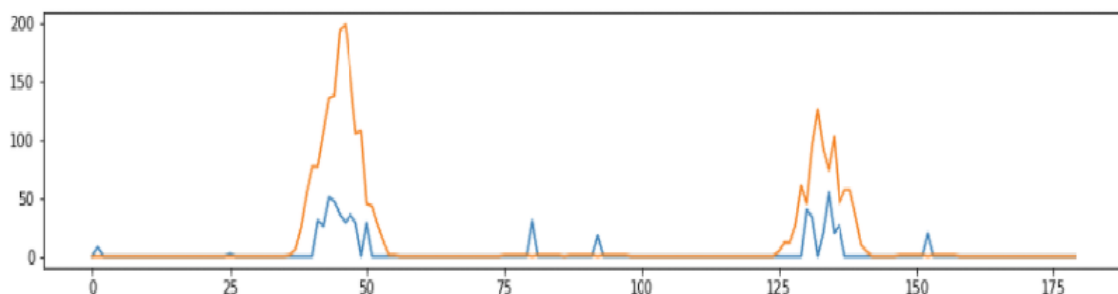


Рис. 2: Распределение сегментов с собственными весами (синий) и весами подобных сегментов (оранжевый)

Таким образом получается выделить прямые, то есть модель дороги будет состоять из прямых, которые в программе можно представлять как две точки. Несмотря на то, что подход достаточно прост в реализации и эффективен с точки зрения вычислений, а также способен справляться с некоторыми ложными срабатываниями поиска линий разметки, представлять модель дороги в виде прямых – не самое точное

решение, потому что закругляющиеся части дороги будут описываться довольно грубо.

В еще одной статье “Распознавание линий дорожной разметки и определение положения автономного автомобиля относительно полосы движения” авторы решают задачу локализации автомобиля, для чего тоже распознают линии разметки [9]. Первоначальный поиск контуров совпадает с подходом, предложенным в предыдущей статье, однако здесь авторы еще используют информацию об угле градиента по разным направлениям. Также для получения “вида сверху” ищется матрица трансформации, которая в дальнейшем может быть использована для извлечения сведений о радиусе поворота на основе данных о кривизне разметки.

В результате получается облако точек, которое потом аппроксимируется одной кривой. Такой подход более точный, чем подход из предыдущей статьи, однако и здесь могут возникнуть неточности описания дороги. Например, наша аппроксимирующая кривая может точно описывать дорогу первые несколько метров от автомобиля, а дальше точность будет сильно падать и кривая будет как бы “вилять хвостом”. Поэтому такой подход к построению модели нам не подходит.

3.1.2. Подход на основе статистической модели

В статье “A model-driven approach for real-time road recognition” описывается метод обнаружения и отслеживания во времени границы дороги по видеопотоку [1]. Отличительной особенностью данного подхода является то, что он строит модель произвольной формы, благодаря чему может применяться на любых автомобильных дорогах. Также положительными сторонами данного метода являются его устойчивость к погодным условиям и возможность отслеживать не только свою полосу движения, но и соседние. Алгоритм состоит из четырех этапов: распознавания дороги, локализации на ней транспортного средства, отслеживания полос движения и управления их номерами. Распознавание дороги основано на вероятностной модели границ дороги на изображении, которая представлена в виде вектора X_d и его ковариационной

матрицей C_{X_d} :

$$X_d = \begin{pmatrix} u_{1L} \\ \cdot \\ u_{nL} \\ u_{1R} \\ \cdot \\ u_{nR} \end{pmatrix}, C_{X_d} = \begin{pmatrix} \sigma_{u_{1L}}^2 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \sigma_{u_{1L}}^2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \sigma_{u_{1R}}^2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_{u_{nR}}^2 \end{pmatrix},$$

где (u_{1L}, \dots, u_{nL}) и (u_{1R}, \dots, u_{nR}) – горизонтальные координаты изображения левого и правого краёв дороги. Матрица C_{X_d} определяет возможные вариации u_{iL} и u_{iR} на изображении. Начальное значение модели получается во время автономного обучения. Затем модель обновляется от кадра к кадру, чтобы найти оптимальное значение X_d , начиная от глубины анализа $p = 0$. Эта процедура является итерационной: модель обновляется, затем на изображении находится зона, представляющая наибольший интерес, затем алгоритм рассчитывает края на глубине $p + 1$. Так продолжается до тех пор, пока не будет достигнут критерий Q :

$$Q = \left\{ P_L = \frac{N_L}{\Delta_v} > 10\% \text{ AND } P_R = \frac{N_R}{\Delta_v} > 10\% \text{ AND } P_t = \frac{N_L + N_R}{2\Delta_v} > 25\% \right\},$$

где N_L и N_R – количество обнаруженных точек на левой и правой стороне соответственно, а Δ_v представляет собой высоту зоны анализа. Это критерий означает, что необходимо обнаружить не менее 10% от каждого края дороги и 25% сторон полосы движения.

Однако недостатком данного подхода можно назвать то, что модель дороги по сути состоит из множества точек левой и правой полосы движения, что неэффективно с точки зрения вычислительной сложности обработки модели, особенно в условиях ограниченной вычислительной мощности в компьютере, установленном в автомобиле. Также, точки модели выясняются благодаря статистической модели, которая нуждается в предварительном обучении, от которого в дальнейшем будет зависеть точность последующих шагов распознавания.

3.1.3. Модель на основе кусочно-линейной функции

Поскольку стало понятно, что прямыми или какой-то одной кривой дорогу описывать получается неточно, то возникает мысль описывать её при помощи комбинации каких-нибудь примитивов. Например, авторы статьи “On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter” предлагают использовать кусочно-линейную функцию, где прямые участки дороги будут описываться отрезками, а закругляющиеся – сегментами окружностей [5].

Авторы предлагают воспользоваться информацией с двух датчиков, установленных на автомобиле – камеры и лазерного дальномера. Данные с них интегрируются при помощи фильтра частиц, где каждая частица сохраняет параметры дороги и положение автомобиля на ней.

Предлагается итеративный алгоритм, состоящий из четырех этапов. На первом этапе генерируются новые частицы на основе модуля предсказания и новых наблюдений. Далее предсказывается следующее наблюдение на основе положения автомобиля и параметров дороги. Затем рассчитываются веса на основе извлеченных характеристик, а затем рассчитываются веса каждой частицы. Потом, в случае достижения определенного условия, происходит процесс передискретизации, чтобы повысить точность наблюдений и предсказаний.

Модель дороги состоит из сегментов, каждый из которых представлен в виде вектора

$$S_i = [x_i^l, y_i^l, x_i^r, y_i^r, h_i, v_i]^T,$$

где (x_i^l, y_i^l) и (x_i^r, y_i^r) – точки левой и правой границы, h_i – длина сегмента, а $v_i = 1/r_i$ – кривизна. Помимо этого, набор сегментов имеет параметр ширины w .

На Рис. 3 показана модель дороги, которая предлагается авторами статьи.

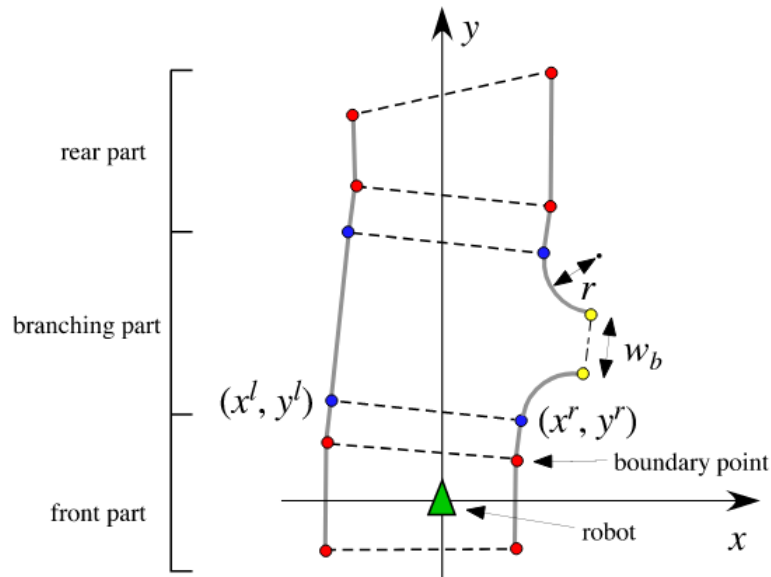


Рис. 3: Модель дороги на основе комбинации примитивов

3.2. Обзор методов отслеживания модели

В статье “An extended hyperbola model for road tracking for video-based personal navigation” авторы предлагают подход к обнаружению и отслеживанию модели на основе множественной точки схода и конденсационного фильтра [2]. Этот фильтр генерирует частицы $\{s, \pi\}$, где s представляет состояние объекта в пространстве состояний, а π – это вероятность того, что состояние s – правильное состояние в каждый промежуток времени. Алгоритм итерационный, при этом изначально всем возможным состояниям приписываются одинаковые веса. Далее на каждой итерации отбираются новые частицы с предыдущей итерации, затем их вероятности обновляются на основе вычислений новых кадров. Состояние системы с наибольшей вероятностью выбирается, как правильное.

Несмотря на то, что отслеживающий модуль хорошо справляется со сложными дорожными условиями, а также оказался более надёжен в сравнении с фильтром Калмана, данный алгоритм работает на основе статистической модели, которая нуждается в предварительном обучении. Также статистические модели имеют большую вычислительную сложность, и у них имеются сложности с поиском оптимального реше-

ния [11].

В статье “Efficient methods and interfaces for road tracking” предлагается алгоритм для отслеживания модели на основе совпадения профилей [3]. Согласно этому методу, производится сопоставление профилей значений серого цвета на изображении и объединение результатов с прогнозом следующего положения дороги, чтобы сделать модель отслеживания надежной.

Однако, у подхода, предложенного в этой публикации, есть один значительный недостаток: алгоритм отслеживания инициализируется пользователем, который должен вручную выделить сегмент дороги. Как сами отмечают авторы, автоматизация этого процесса сильно сказывается на точности отслеживания, а также алгоритм плохо работает на сложных сценах.

4. Выбор модели дороги

Чтобы точность отслеживания дороги была высокой, а также стало возможным получение и хранение дополнительной информации, например, положение автомобиля относительно полос движения или позиционирование на карте, необходимо строить модель дороги по контурам, которая будет обновляться по мере поступления новых данных.

Сперва нужно определиться с примитивами, из которых будет состоять модель. Под примитивами будем понимать графики простых математических функций, таких как прямые, параболы, сплайны и т.д.

4.1. Критерии отбора примитивов

Выделим следующие критерии:

1. Погрешность описания дороги примитивами должна быть наименьшей. В частности, это означает, что наша модель должна точно описывать любые типы дорог, будь то прямые, закругляющиеся или другие.
2. Вычислительная сложность обработки и затраты по памяти на хранение должны позволять работать в режиме онлайн с минимальными задержками.

4.2. Выбор примитивов

4.2.1. Прямые

Использовать прямые линии в качестве примитивов для модели дороги – не самая лучшая идея, несмотря на то, что вычислительная сложность и затраты по памяти будут минимальны. Однако первый критерий является более важным, а прямые ему плохо удовлетворяют: если дорога заворачивается, то прямые будут очень грубо это описывать.

4.2.2. Параболы и гиперболы

Параболы и гиперболы уже гораздо лучше подходят для поставленной цели, так как позволяют описывать закругляющиеся части дороги. К тому же, если представить параболу как $y = Ax^2 + Bx + C$, то прямая – это вырожденный случай параболы, при котором коэффициент $A = 0$. Пользуясь этой формулой можно представить модель дороги с прямыми полосами через вырожденный случай параболы.

Однако это не лучшее решение, так как входные контуры – не идеальные, поэтому может возникать ситуация, при которых коэффициент $A \neq 0$, но очень к нему близок. В результате получится парабола на первых нескольких метрах близкая к прямой, а вдали “мащущая хвостом”. Поэтому параболы в чистом виде не очень хороши для представления модели дороги.

4.2.3. Сплайны

Входные контуры, выделенные базовыми средствами OpenCV, зачастую сильно зашумлены и имеют выбросы, а также представляют собой довольно плотное облако точек. А аппроксимация сплайн-функцией предполагает разбиение кривой на отрезки, на каждом из которых кривая аппроксимируется полиномом определенной степени. И так как контур состоит из множества близких точек, то сплайн-функция будет иметь очень большое разбиение на отрезки, что существенно повышает вычислительную сложность на обработку такой модели, а также повышает сложность её отслеживания.

4.2.4. Комбинация дуг и отрезков

Использование комбинации различных примитивов поможет покрыть большее количество вариаций изменения дороги. Так, например, можно использовать комбинацию дуг и отрезков, чтобы покрывать ими как прямолинейные участки дороги, так и закруглённые. Для этого нужно вычислять кривизну дороги и на основе этой информации выбирать, каким примитивом описывается каждый отдельный сегмент дороги.

Однако в отличии от модели из статьи [5], наша модель предполагает, что прямые участки дороги не обязательно будут описываться прямыми отрезками, что повысит точность описания дороги. Также в данной статье использовалась информация с двух датчиков, в то время как наш подход будет основываться на информации с единственного датчика – камеры.

В дальнейшем для алгоритма решения выбрана в качестве основной модель, состоящая из комбинации отрезков и дуг, которые представляют из себя сегменты окружностей.

5. Программная реализация

В компании ООО “Харман” ADAS-система реализована на языке программирования C++. Поэтому, так как планируется разработать прототип, который в дальнейшем может быть использован для разработки компанией, для реализации алгоритма построения и отслеживания модели дороги был выбран язык C++. В качестве библиотеки алгоритмов компьютерного зрения использовалась библиотека OpenCV, так как она тоже используется в разработке в компании.

Модель дороги в программной реализации является связным списком, состоящим либо из прямых отрезков, либо из сегментов окружностей. Для этого был написан базовый класс `ModelElement`, а так же два класса `CircularArc` и `LineSegment`, которые наследовались от него. Класс `RoadModel` хранит указатель на объект класса `ModelElement`, благодаря чему список может состоять из объектов разных производных этого класса.

Класс `CircularArc` имеет следующие приватные поля для представления сегмента окружности: точку центра, радиус окружности, начальный и конечный угол сегмента. Класс `LineSegment` имеет два приватных поля – точку начала и конца отрезка. Также оба этих класса имеют переопределенные методы `drawModelElement` для отрисовки элемента.

Для построения модели дороги используется информация о кривизне входных контуров, поэтому сначала вычисляется вектор, состоящий из значений кривизны в каждой точке контура.

Кривизна вычисляется геометрическим способом. Для этого в контуре берется точка и две её соседние точки в зависимости от заданного шага, эти точки образуют треугольник. Далее ищется окружность, которая описывает этот треугольник. Тогда величина, обратная к радиусу этой окружности, будет являться значением кривизны в выбранной точке. Иллюстрация данного подхода приведена на Рис. 4.

Построение модели по входному контуру выполняется статическим методом класса `RoadModelBuilder`, который принимает на вход объект класса `RoadModelTracker`, точки входного контура и вектор со значени-

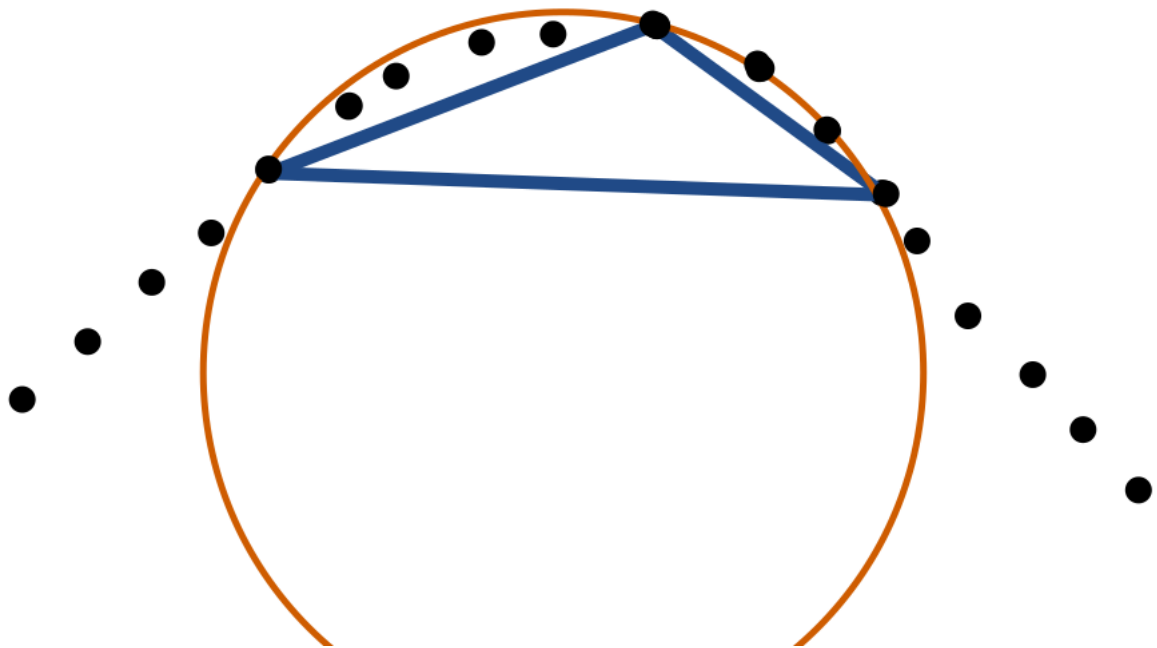


Рис. 4: Геометрический способ вычисления кривизны

ями кривизны в каждой точке. Этот метод разбивает входной контур на участки постоянной кривизны, далее определяет, каким именно элементом описывается каждый участок, и добавляет его в модель через объект класса `RoadModelTracker`.

Отслеживающий модуль представлен классом `RoadModelTracker`, который хранит приватный указатель на объект класса `RoadModel`. Помимо этого, этот класс содержит специальное приватное логическое поле `isModelConstructed`, которое изначально принимает значение `false`, а после построения модели по первому кадру, изменяет свое значение на `true`. Это нужно для того, чтобы модель не начала отслеживаться до того, как была сконструирована.

Далее всякий раз, когда в модель поступают новые контуры, эти контуры анализируются и в них выделяются элементы модели: прямые отрезки и дуги. Как только определяется новый элемент модели, он отправляется в метод класса `RoadModelTracker`, который проверяет, имеется ли в модели “схожий” элемент. “Схожесть” элемента вычисля-

ется как расстояние между новым элементом и имеющимся элементом в модели.

Если окажется, что метрика с каким-то существующим элементом ниже определенного порога, который зафиксирован в частных константных полях класса `RoadModelTracker`, то элемент модели заменяется на новый.

Диаграмма классов получившегося решения представлена на Рис. 5.

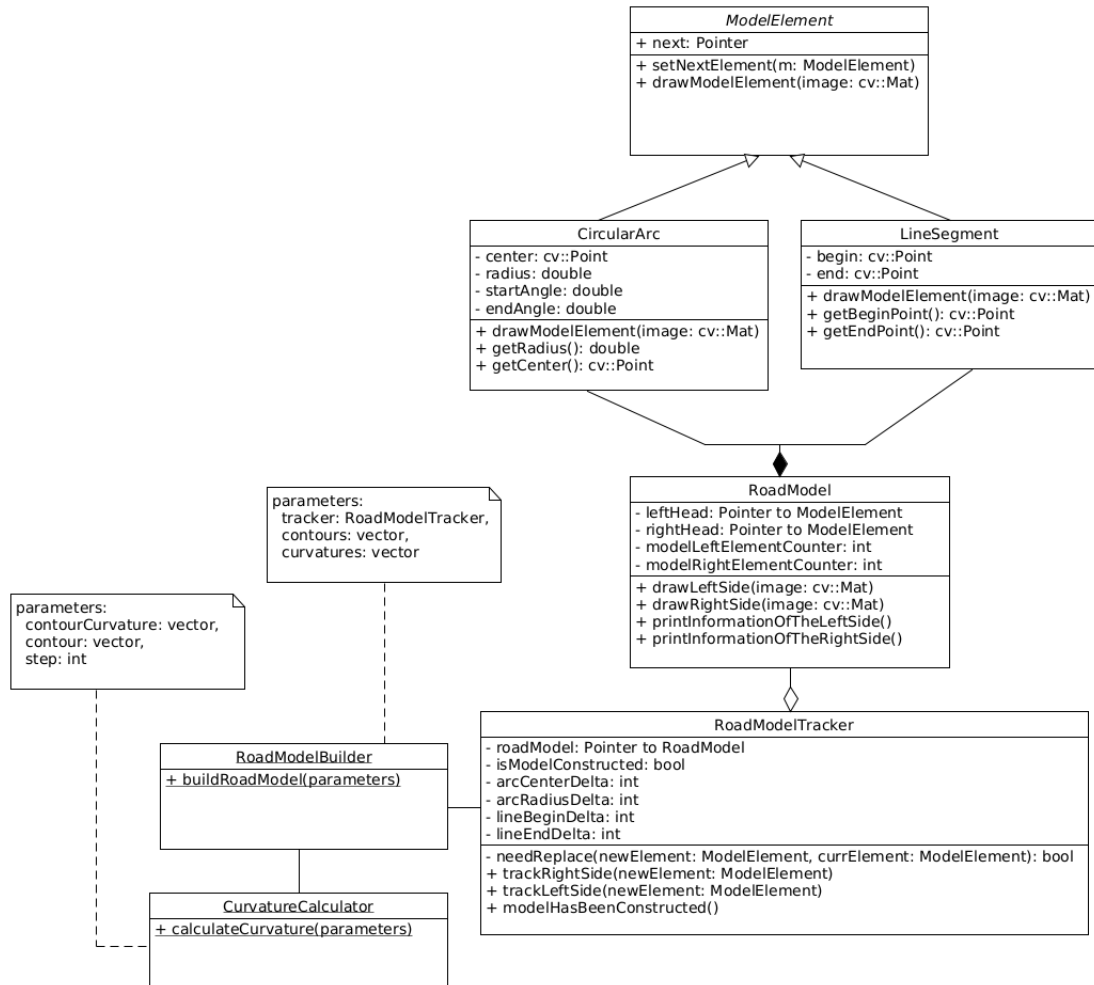


Рис. 5: Диаграмма классов UML

6. Тестирование

6.1. Способ тестирования

Эксперименты, позволяющие определить, насколько хорошо выбранная модель и алгоритм позволяют описывать дорогу, помимо визуальной оценки, проводились следующим образом:

1. На вход алгоритма подаются контуры, выделенные при помощи базовых средств OpenCV.
2. По этим контурам строится модель дороги.
3. Далее оценивается, насколько входные контуры отличаются от построенной модели:
 - Рассчитывается дистанция в пикселях между контурами и моделью на разных расстояниях от машины.

Благодаря такой оценке можно оценить, насколько хорошо была выбрана модель, а также насколько близко она описывает тот или иной вид дороги.

6.2. Тестирование на “идеальных” контурах

Было проведено тестирование реализованного алгоритма на “идеальном контуре”, то есть на контуре, точки которого задаются вручную в программе. Тесты показали, что модель, построенная по таким контурам, точно описывает эти контуры, то есть максимальная погрешность в пикселях равна нулю.

Иллюстрация входного контура и готовой модели представлены на Рис. 6, где красным цветом обозначен сегмент окружности, а синим – прямые отрезки. Также был протестирован механизм отслеживания модели: в построение контура был добавлен параметр времени, в зависимости от которого контур сдвигался вниз. Тесты показали, что отслеживающий модуль правильно отслеживал изменение положения модели.

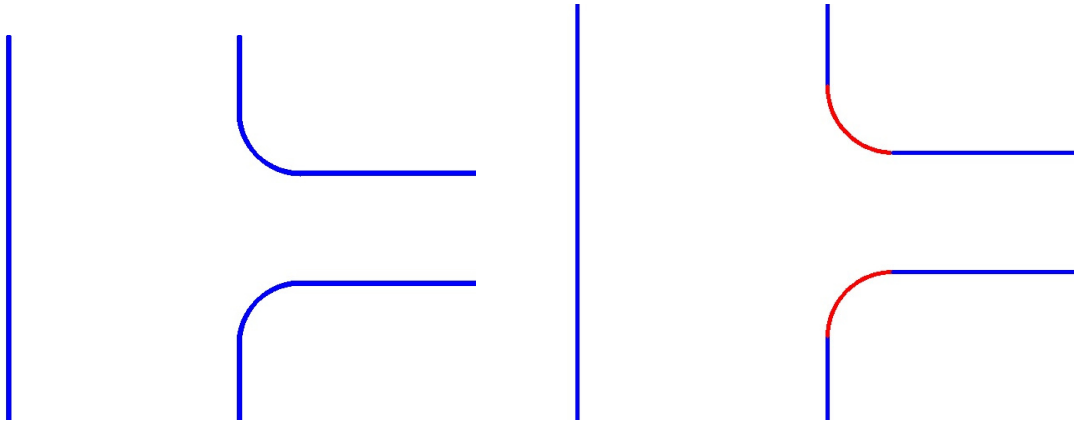


Рис. 6: Входной контур и модель по этому контуру

6.3. Тестирование на реальных контурах

Почти всегда системы компьютерного зрения работают с данными, в которых есть некоторый шум. Поэтому входные данные нужно как-то предобрабатывать, уменьшая количество выбросов.

Входными данными для поставленной задачи является набор точек, выделенных базовыми средствами OpenCV. В этом наборе часто возникает ситуация, при которых точки склеиваются. Также нередко случается, что одинаковыми оказываются точки, находящиеся через одну друг от друга, возникает ситуация, при которой контур как бы скакнул назад. Поэтому входные контуры были предобработаны, чтобы исключить эти ситуации.



Рис. 7: Входной контур и модель по этому контуру

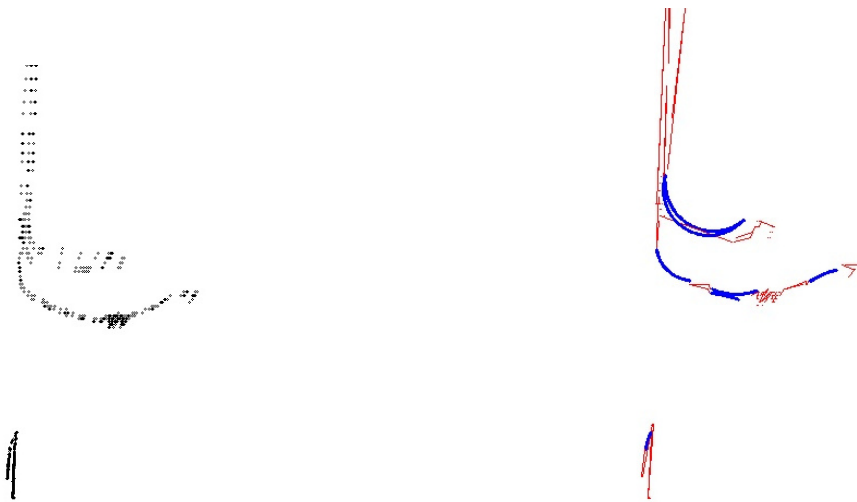


Рис. 8: Входной контур и модель по этому контуру

Примеры построения модели дороги по реальным контурам представлены на Рис. 7 и Рис. 8.

Таблица 1: Результаты тестирования

Мат. ожидание	Дисперсия	Среднеквадратичное отклонение
4.05	5.25	2.29

Результаты тестирования, а именно высчитывания погрешности в пикселях, приведены в Таблице 1.

Также было проведено вычисление зависимости погрешности и расстояния от машины. Получившиеся результаты представлены на Рис. 9. Из гисторгаммы видно, что получившиеся модели в среднем выдают погрешность меньше, чем десять пикселей. Также гистограмма показывает, что чем ближе к машине находятся точки контура, тем больше погрешность. Однако это обусловлено тем, что прямые участки дороги легче всего описывать моделью, а закругляющиеся наоборот. А в тестовых контурах поворот находился ближе к машине.

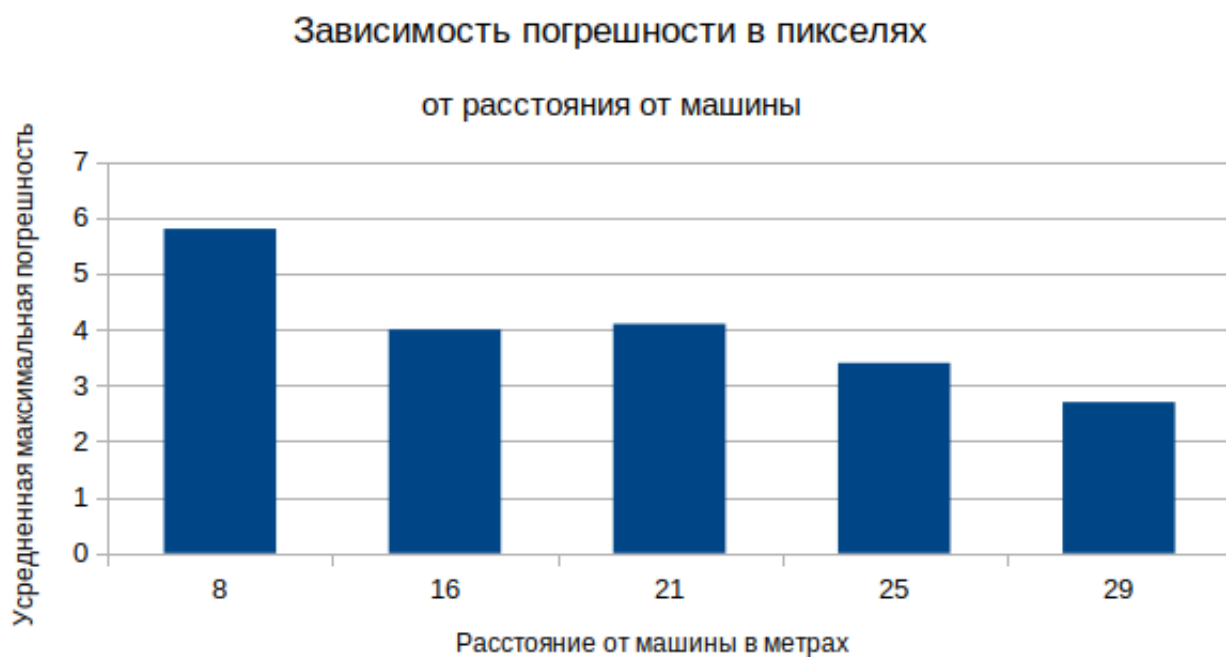


Рис. 9: Гистограмма зависимости между погрешностью и дистанцией

7. Возможные улучшения алгоритма

Как видно на Рис. 8, в модели есть часть, где сегменты дуги идут вперемешку с прямыми отрезками, это возникает из-за зашумленного контура. Такие контуры можно дополнительно предобрабатывать, чтобы исключить подобные ситуации. Например, можно проходить по контуру, высчитывая направляющие векторы по соседним точкам. Анализируя углы этих векторов, можно определять, что контур “резко повернул” в другую сторону. Далее находить еще один такой поворот и исключать получившейся сегмент из контура.

Так как модель строится на основе кривизны контура, то еще одним возможным улучшением будет динамическое вычисление шага, с которым рассчитывается кривизна. Для этого можно предварительно анализировать плотность точек в контуре.

Заключение

В результате работы над ВКР были решены следующие задачи:

1. Изучены существующие подходы к решению задачи.
2. Выбраны наиболее подходящие примитивы для модели.
3. Реализован наиболее подходящий алгоритм решения.
4. Проведено тестирование полученного решения на реальных данных.

Исходный код:

<https://github.com/alechh/building-and-tracking-a-road-model>

Список литературы

- [1] Aufrère Romuald, Chapuis Roland, Chausse Frederic. A model-driven approach for real-time road recognition // ResearchGate. A European commercial social networking site for scientists and researchers to share papers, ask and answer questions, and find collaborators. — 2001. — URL: https://www.researchgate.net/publication/220465009_A_model-driven_approach_for_real-time_road_recognition (online; accessed: 21.10.2021).
- [2] Bai Li, Wang Yan, Fairhurst Michael. An extended hyperbola model for road tracking for video-based personal navigation // Elsevier's premier platform of peer-reviewed literature. — 2007. — URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.193.32&rep=rep1&type=pdf> (online; accessed: 21.04.2022).
- [3] Baumgartner A., Hinz S., Wiedemann C. Efficient methods and interfaces for road tracking // CiteSeer, электронная библиотека и поисковая машина по научным публикациям и препринтам. — 2002. — URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.1786&rep=rep1&type=pdf> (online; accessed: 21.04.2022).
- [4] Yu Zhuoping, Ren Xiaozhou, Huang Yuyao et al. Detecting Lane and Road Markings at A Distance with Perspective Transformer Layers // arXiv. A curated research-sharing platform open to anyone. — 2020. — URL: <https://arxiv.org/pdf/2003.08550.pdf> (online; accessed: 21.10.2021).
- [5] Matsushita Y., Miura J. On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter // Elsevier's premier platform of peer-reviewed literature. — 2011. — URL: <http://www.aisl.cs.tut.ac.jp/old/pdf/files/ras2011.pdf> (online; accessed: 22.04.2022).

- [6] Lee Seokju, Kim Junsik, Yoon Jae Shin et al. VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition // CVF. The computer Vision Foundation. — 2017. — URL: https://openaccess.thecvf.com/content_ICCV_2017/papers/Lee_VPGNet_Vanishing_Point_ICCV_2017_paper.pdf (online; accessed: 21.10.2021).
- [7] Закревский Н.П., Хмелева А.В. Распознавание дорожной разметки с использованием алгоритмов компьютерного зрения // eLibrary. Научная электронная библиотека. — 2020. — URL: <https://www.elibrary.ru/item.asp?id=43309961> (дата обращения: 21.10.2021).
- [8] Коваль А.В., Гришечко В.А., Чайчиц Н.Н. Распознавание линий разметки полос движения на основе видеоряда с использованием весовой классификации сегментов // eLibrary. Научная электронная библиотека. — 2018. — URL: <https://www.elibrary.ru/item.asp?id=32704270> (дата обращения: 21.10.2021).
- [9] Коваль А.В., Прохоренко А.С., Галузо В.Е. Распознавание линий дорожной разметки и определение положения автономного автомобиля относительно полосы движения // Научный горизонты. Международный научный журнал. — 2020. — URL: <https://tinyurl.com/rormladotpotavrttl> (дата обращения: 21.10.2021).
- [10] Комяк А.П. Алгоритм распознавания полосы движения // Научный горизонты. Международный научный журнал. — 2019. — URL: <https://tinyurl.com/lanerecogal> (дата обращения: 21.10.2021).
- [11] Тянь К.Л. Аналитические, статистические и имитационные модели: преимущества и недостатки // Киберленинка, научная электронная библиотека открытого доступа. — 2020. — URL: <https://tinyurl.com/stat-models> (дата обращения: 21.04.2022).