

САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математическое обеспечение и администрирование
информационных систем
Кафедра системного программирования

Назаренко Владимир Владимирович

**Выделение объектов
на видеопоследовательности**

Выпускная квалификационная работа

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

Рецензент:
ПЕНКРАТ Н. А.
менеджер проектов, ООО “Ланит-Терком”

Санкт-Петербург
2018

SAINT PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Software Engineering Department

Vladimir Nazarenko

Object detection in a video sequence

Master thesis

Scientific advisor:
sr. Lecturer MIKHAIL SMIRNOV

Reviewer:
NICKOLAY PENKRAT
Project Manager, Lanit-Tercom LLC

St Petersburg
2018

Содержание

1 Постановка задачи	6
2 Обзор	7
2.1 Основные определения	7
2.2 Карта глубины	8
2.2.1 Алгоритм Semi-global Matching	8
2.2.2 Алгоритм AD-Census	9
2.2.3 Алгоритм CVS	9
2.2.4 Сравнение алгоритмов построения карты глубины	10
2.3 Существующие подходы к детектированию препятствий с помощью стереокамеры	10
2.4 Динамическое определение положения плоскости дорожного полотна	12
2.5 Подход Stixel World	13
2.6 Подходы к детектированию маркеров дорожной разметки	16
3 Алгоритм поиска препятствий движению автомобиля	17
3.1 Поиск препятствий с помощью плотной карты диспаритета	17
3.2 Поиск препятствий с помощью разреженной карты диспаритета	21
4 Алгоритм детектирования маркеров дорожной разметки	24
5 Апробация	26
5.1 Данные	26
5.2 Оценка качества работы алгоритмов	27
5.2.1 Методика оценки алгоритма детектирования препятствий	27
5.2.2 Методика оценки алгоритма детектирования маркеров дорожной разметки	29
5.2.3 Результаты оценки алгоритма детектирования препятствий	30
5.2.4 Результаты оценки алгоритма детектирования маркеров дорожной разметки	31
6 Заключение	34

Введение

На автомобильных дорогах происходит большое количество дорожно - транспортных происшествий (ДТП). Согласно исследованию [24] причиной большого количества ДТП является водитель, внимательность которого ослаблена в следствие усталости, приёма медицинских препаратов, имеющих седативный эффект, алкогольного опьянения, выполнения задач, не связанных с управлением транспортным средством и др.

В настоящее время широкое распространение получили системы помощи водителю[23]. Такие системы, например, предупреждают водителя об ограничении скорости на участке дороге (с помощью распознавания соответствующих дорожных знаков), о пересечении маркеров дорожной разметки, об опасности столкновения с различными объектами[17]. Существуют исследования, экспериментально доказывающие практическую полезность таких систем [25].

Важным элементом таких систем являются различные сенсоры. Наиболее распространёнными сенсорами для решения перечисленных выше задач являются лидары, радары, ультразвуковые датчики и оптические системы видимого спектра (стерео- и монокамеры). Область применения каждого из типов сенсоров ограничена[17, 28]. Так, радары обладают низкой точностью определения формы и расстояния до объекта. Лидары обладают низкой точностью в плохих погодных условиях. Кроме того, высокая стоимость лидаров делает решения на их основе недоступными для массового сегмента автомобильной промышленности. Ультразвуковые датчики способны обнаруживать препятствия только на небольших расстояниях. Использование оптических систем требует использования ресурсоёмких алгоритмов для определения расстояния до объектов.

Стереокамера сочетает в себе невысокую стоимость, возможность с высокой точностью определять геометрию и расстояние до объектов, простоту монтажа.

На кафедре системного программирования Санкт-Петербургского государственного университета в совместных исследовательских проектах с компанией Prosense[20] (Южная Корея), разрабатываются алгоритмы для системы помощи водителю. Одним из требований к этой системе является низкая стоимость и возможность установки системы без существенных модификаций конструкции автомобиля. Одной из частей этой системы является так называемый **сенсор безопасного движения** – подсистема, предупреждающая водителя о потенциальных столкновениях и пересечении маркеров дорожной разметки.

В данной работе мы сфокусировались на разработке и апробации алгоритмов для сенсора безопасного движения. В связи с требованиями к разрабаты-

ваемой системе, а именно, ограничением на стоимость и простоту монтажа, в качестве сенсора мы выбрали стереокамеру, состоящую из двух откалиброванных камер видимого диапазона.

Существует, как минимум, два класса алгоритмов для решения задач помощи водителю, использующих оптические сенсоры: нейросетевые алгоритмы и алгоритмы на основе методов классического компьютерного зрения. В данной работе было решено использовать алгоритмы на основе классического компьютерного зрения. Связано это со следующими проблемами нейросетевых алгоритмов[16].

- Сложность модификации нейросетевых алгоритмов.
- Неуниверсальность нейросетевых алгоритмов.
- Сложность интерпретации решений нейросетевых алгоритмов.

Строго говоря, под предупреждением водителя о потенциальном столкновении мы понимаем детектирование на изображении **препятствий** – любых объектов, которые делают невозможным или опасным проезд через занимаемую ими область пространства. Типовыми примерами препятствий являются люди, автомобили, столбы, здания. Также мы считаем препятствиями особенности рельефа (холмы) и различные мелкие объекты, такие как бордюры. Слова “объект” и “препятствие” для нас являются синонимами. Под **детектированием препятствий** мы понимаем выделение препятствий на изображении в том или ином виде. Например, в виде описывающего прямоугольника или в виде области на изображении, движение в которой безопасно – **безопасной области**.

Также отметим, что термины “выделение объектов”, “детектирование объектов” и “сегментация изображения на объекты” мы считаем эквивалентными.

Под **детектированием маркеров дорожной разметки** мы понимаем задачу выделения на изображении таких маркеров, как одиночная сплошная линия, двойная сплошная линия, прерывистая линия, бордюры.

1 Постановка задачи

Целью данной работы является разработка и реализация, на основе подходов классического стерео-зрения и классической обработки изображений, алгоритмов для сенсора безопасного движения. Для достижения этой цели в рамках работы были сформулированы следующие задачи.

- Разработать и реализовать прототип алгоритма поиска препятствий движению автомобиля на изображении, полученном с помощью стереокамеры.
- Разработать и реализовать прототип алгоритма поиска маркеров дорожной разметки на изображении, полученном с помощью стереокамеры.
- Провести апробацию разработанных алгоритмов.

2 Обзор

2.1 Основные определения

Единственными сенсорами, которые мы используем для решения поставленных задач являются стереокамера и монокамера.

Стереокамера это система из двух камер, расположенных на небольшом расстоянии друг от друга, взаимное расположение и калибровка которых известны в любой момент времени. Ключевым свойством такой системы является возможность оценки расстояния до объектов на изображении за счёт параллакса.

Оптический поток – это сопоставление пикселям изображения пары целочисленных значений, соответствующих движению объекта реального мира, спроектированного в пиксель, в плоскости камеры.

С использованием стереокамеры становится возможным вычислить **карту глубины** – отображение, сопоставляющее пикслю исходного изображения расстояние от оптического центра камеры до точки в пространстве, которая была спроектирована в данный пиксль. Карта глубины может быть плотной, в таком случае подразумевается, что большинству пикселей сопоставлено значение глубины, либо неплотной – значение глубины сопоставлено лишь некоторым пикселям. В случае неплотной карты глубины, как правило, значения глубины сопоставлены так называемым ключевым точкам – точкам, в которых на изображении имеется выраженный перепад яркостей.

В настоящее время алгоритмы построения карты глубины используют **ректифицированные изображения** – это пара изображений, к которым применено перспективное преобразование, модифицирующее исходную пару изображений, симулируя расположение плоскостей обеих камер в одной и той же плоскости.

Основным шагом вычисления карты глубины является вычисление диспаритета для каждого пикселя. **Диспаритет** это смещение между координатами пикселя, соответствующего одной и той же точке пространства на первом и втором сенсорах стереокамеры. Как правило, диспаритет вычисляется для пары ректифицированных изображений. В таком случае диспаритет – это одно число, смещение по горизонтальной оси. В тексте данной работы в некоторых местах мы взаимозаменяем употребляем термины “карта диспаритета” и “карта глубины”. Это связано с тем, что переход от одной карты к другой в нашем случае возможно осуществить путём домножения на константу.

2.2 Карта глубины

Построение карты глубины является популярным методом предобработки пары изображений, полученных с помощью стереокамеры, и используется во многих работах, авторы которых решают задачу детектирования препятствий.

Существуют альтернативные методы предобработки для вычисления геометрии и расстояния до объектов на изображении [9, 13], основанные на использовании одной камеры, однако они проигрывают построению карты глубины по снимку, полученному с помощью стереокамеры либо в точности, либо в качестве результатов. Поэтому в данной работе мы придерживаемся подхода на основе построения карты глубины.

Способ построения карты глубины критически важен для алгоритма детектирования препятствий, разработанного в рамках данной работы. Различные алгоритмы отличаются друг от друга как качеством, так и скоростью работы. Разработка собственного алгоритма построения карты глубины находится за пределами данной работы, однако нами были рассмотрены следующие реализации алгоритмов.

- Реализация алгоритма Semi-global Matching из открытой библиотеки компьютерного зрения OpenCV[12].
- Закрытая реализация алгоритма AD-Census, выполненная инженерами компании “Ланит-Терком”.
- Закрытая реализация алгоритма вычисления разреженной карты глубины CVS.

2.2.1 Алгоритм Semi-global Matching

Алгоритм Semi-global Matching [11] (SGM) широко применяется для вычисления карты глубины.

Данный алгоритм минимизирует целевую функцию (1) на паре ректифицированных изображений, полученных с помощью стереокамеры.

$$E(D) = \sum_p [C(p, D_p) + \sum_{q \in N_p \wedge |D_p - D_q|=1} P_1 + \sum_{q \in N_p \wedge |D_p - D_q|>1} P_2] \quad (1)$$

Здесь p – это координаты пикселя на левом изображении, $C(p, D_p)$ – значение функции схожести левого и правого изображения стереокамеры, P_1 и P_2 – константные штрафы, N_p – окрестность точки p на левом изображении, D_p и D_q – значения диспаритетов для пикселей p и q .

Особенностью алгоритма SGM является то, что минимизация целевой функции производится сразу по нескольким направлениям, как это изображено на рисунке 1.

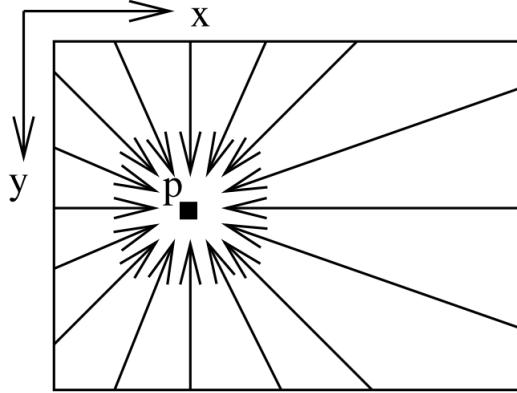


Рис. 1: Схема направлений оптимизации в алгоритме SGM [11]

В целях предобработки пары изображений, полученных с помощью стереокамеры, нами была использована общедоступная реализация алгоритма SGM из открытой библиотеки OpenCV. В данной реализации в качестве функции схожести $C(p, D_p)$ используется сумма абсолютных разниц (SAD) в окрестности пикселя.

2.2.2 Алгоритм AD-Census

Данный алгоритм [18] является модификацией алгоритма SGM для систем массового параллелизма, таких как графические ускорители.

В отличие от алгоритма SGM, в алгоритме AD-Census используется функция схожести, основанная на взвешенном среднем двух компонент: абсолютной разности интенсивностей (компоненты AD) и расстоянии Хэмминга (компоненты Census). Кроме того, авторы предлагают ряд модификаций оригинального алгоритма SGM для практического уменьшения количества требуемых вычислений.

В целях предобработки пары изображений, полученных с помощью стереокамеры, нами была использована закрытая реализация алгоритма AD-Census, выполненная инженерами компании Ланит-Терком[15] и предоставленная нам в рамках сотрудничества компании с кафедрой системного программирования.

2.2.3 Алгоритм CVS

Алгоритм CVS[2] является запатентованной разработкой компании “Системы Компьютерного Зрения”[6] и стал доступен нам благодаря сотрудниче-

ству компании с кафедрой системного программирования. Данный алгоритм позволяет строить разреженную карту глубины, где значения глубины сопоставляются ключевым точкам – пикселям на изображении, соответствующим выраженным перепадам яркости на изображении.

Ключевым преимуществом данного алгоритма является возможность реализовать его крайне эффективно, в том числе на ПЛИС[2].

2.2.4 Сравнение алгоритмов построения карты глубины

На рис. 2 представлены результаты запуска рассмотренных алгоритмов расчёта карты диспаритета на одном и том же изображении. Заметно, что алгоритм CVS вычисляет диспаритет лишь в небольшой доле точек. Более того, эти точки расположены неравномерно. Также заметно, что результаты работы алгоритмов SGM и AD-Census дают приблизительно одинаковый результат. Однако есть и существенные отличия. Так, в выбранной реализации алгоритма SGM нет заполнения заслонённых областей, а выбранная реализация AD-Census хуже справляется со сглаживанием плоскости дорожного полотна. Вторая проблема представляет большую сложность для разработанного нами алгоритма детектирования препятствий, поэтому было решено для экспериментов, в которых необходимо вычисление плотной карты диспаритета, использовать реализацию SGM из библиотеки OpenCV.

2.3 Существующие подходы к детектированию препятствий с помощью стереокамеры

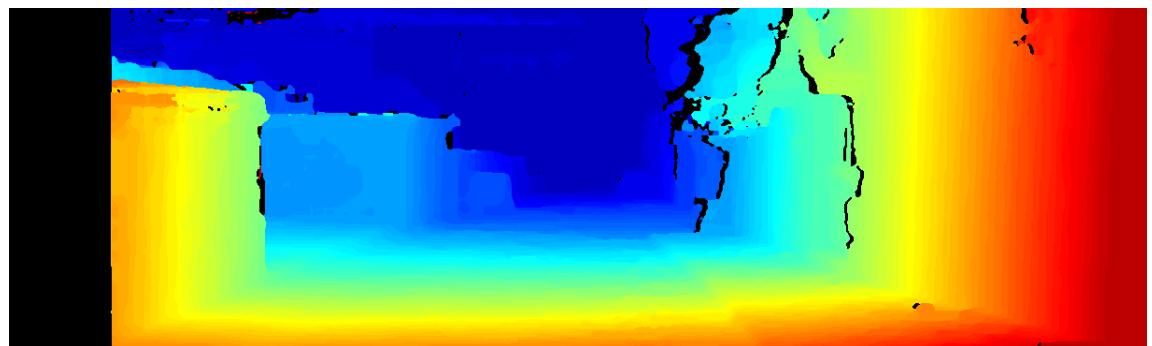
Существует большое количество исследований в области детектирования неклассифицированных объектов с помощью стереокамеры. В рамках работы было проведено исследование работ в данной области. Перечислим далее работы, наиболее релевантные нашей цели.

Авторы [10] предложили использовать геометрическую зависимость между оптическим потоком и расстоянием до объекта, вычисленным с помощью алгоритмов построения карты глубины. Как отмечают авторы, данный подход неустойчив к движению камер в плоскости кадра и к неточностям алгоритмов расчёта оптического потока и карты глубины. К плюсам предложенного решения можно отнести высокую скорость работы, без учёта вычисления оптического потока и расчёта карты глубины.

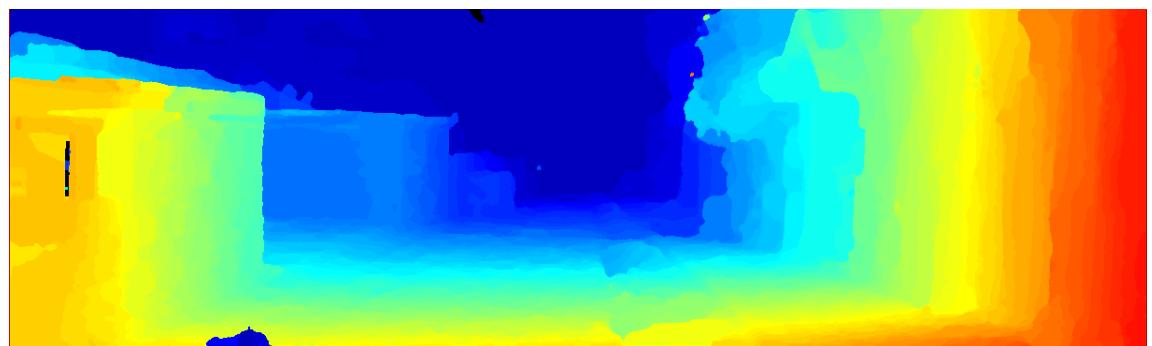
В работе [14] авторы представили способ определения рельефа дорожной поверхности без использования карты глубины. Так же авторы показали принципиальную возможность выделять отдельные препятствия с помощью представленного ими алгоритма. Однако точность алгоритма невысока, особенно



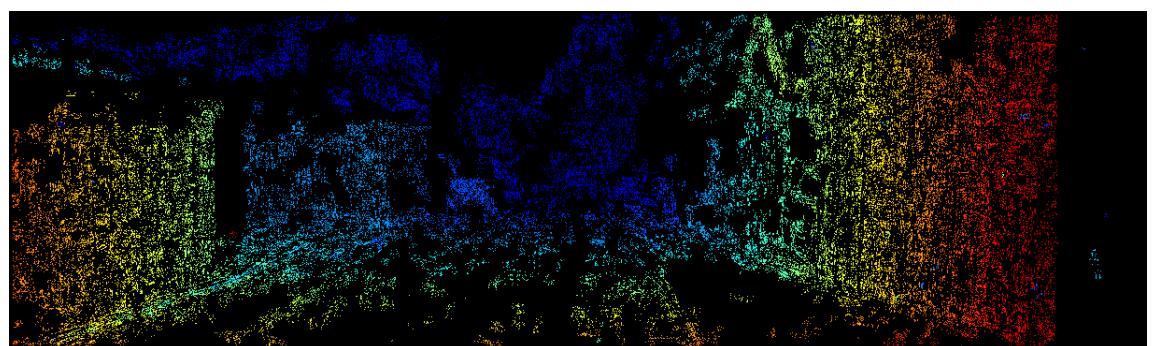
(a) Левое ректифицированное изображение



(b) Semi-global Matching



(c) AD-Census



(d) CVS

Рис. 2: Результаты запуска различных алгоритмов расчёта карты диспаритета (значение диспаритета закодировано цветом)

в случае наличия большого количества препятствий на изображении. Тем не менее, способ динамического определения положения дорожного полотна, предложенный в этой работе, широко используется. В том числе, он был использован в нашем исследовании.

Развивая подход Labayrade [14], авторы [27] предлагают улучшения для алгоритма, позволяющие применять алгоритм в ситуациях бездорожья. Тем не менее авторы приводят только качественную оценку своего алгоритма, из которой следует вывод, что алгоритм применим только в ситуациях с небольшим количеством препятствий на изображении, как и подход Labayrade [14].

В работе [1] предложен подход выделения препятствий на изображении, основанный на расчёте неплотного оптического потока с помощью KLT-трекера и фильтра Калмана и расчёте неплотной карты глубины. Авторы данной работы не приводят способы выделения отдельных объектов на изображении, также в данной работе не рассмотрено выделение препятствий, не имеющих собственного движения.

В статье [19] авторы предлагают вместо вычисления карты глубины сегментировать изображение на вертикальные полосы, для каждой из которых вычисляются две горизонтальных границы, соответствующие основанию ближайшего препятствия и его высоте. Затем, используя эту сегментацию, можно выделить объекты не прибегая к сложным вычислениям. Данный подход показался нам перспективным и многие идеи нашего исследования были заимствованы из данной работы, поэтому опишем этот алгоритм подробно.

2.4 Динамическое определение положения плоскости дорожного полотна

В данной работе мы использовали оценку плоскости дорожного полотна с помощью алгоритма, предложенного в статье [14]. Авторы предлагают для каждой строки карты диспаритетов построить гистограмму, затем получившиеся гистограммы сконкатенировать. Получившееся изображение авторы носит название “vdisparity image”. На рис. 3 приведён пример вычисления плоскости дорожного полотна. Слева направо на этом изображении представлены фрагмент исходного левого изображения, “vdisparity image”, “vdisparity image”, на котором пунктирной линией отмечена область, соответствующая диспаритету дороги. Поиск этой области осуществляется наложением на модель параметрической модели (в данном случае, линии), с помощью преобразования Хафа. Получившаяся линия имеет следующую интерпретацию: если точка с координатами (x, y) принадлежит линии, то ожидаемое значение диспаритета плоскости дороги в строке изображения y равно x .

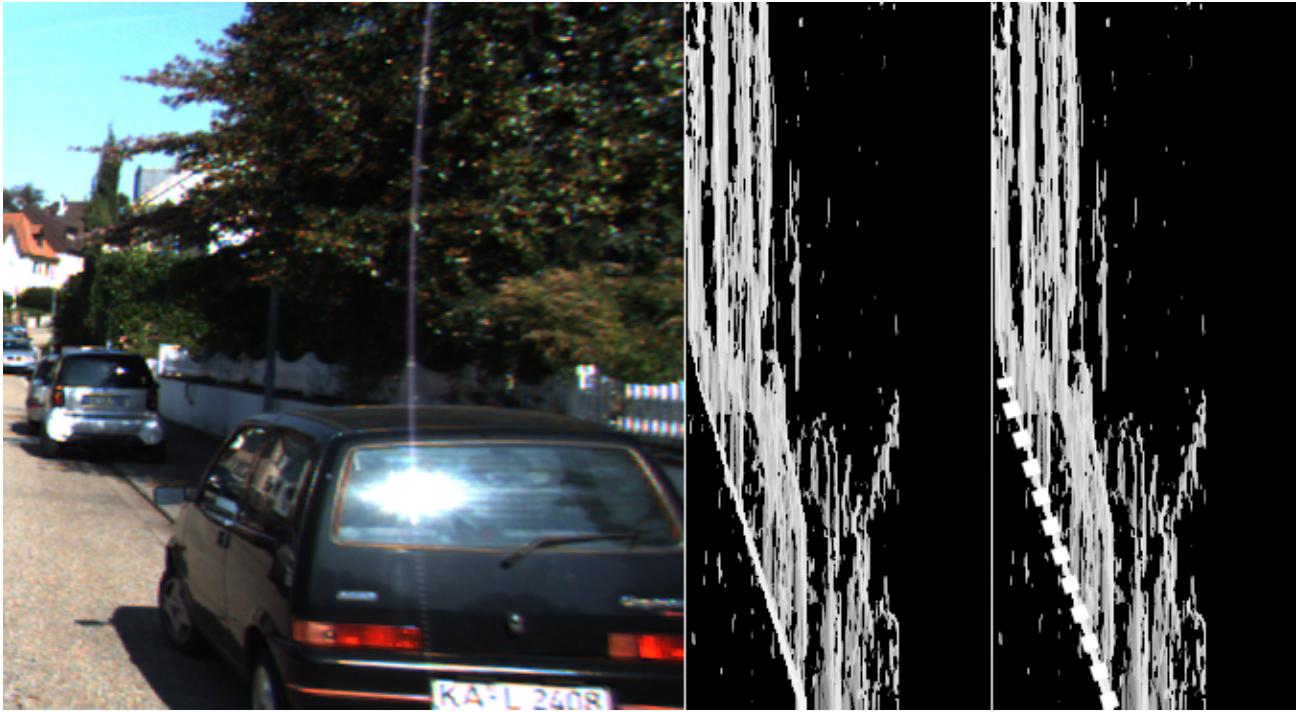


Рис. 3: Пример определения плоскости дорожного полотна

2.5 Подход Stixel World

Опишем данный подход [19], подробнее, так как он оказал существенное влияние на наше исследование.

В модели Stixel World препятствия описываются упрощённой моделью: предполагается, что любое препятствие может быть хорошо приближено параллелепипедом. В таком случае на изображении препятствия могут быть хорошо приближены набором вертикальных прямоугольников с сопоставленными им значениями глубины. На рис. 4 изображена схема упрощённой модели препятствий, принятой в модели Stixel World.

Таким образом, модель Stixel World предполагает сегментацию изображения на стиксели – прямоугольники фиксированной ширины. Каждому стикселию соответствует 4 следующих значения.

1. Индекс стикселя.
2. Координата строки изображения – нижняя граница стикселя.
3. Координата строки изображения – верхняя граница стикселя.
4. Значение расстояния. При этом предполагается, что расстояние одинаково до всех пикселей, принадлежащих стикселию.

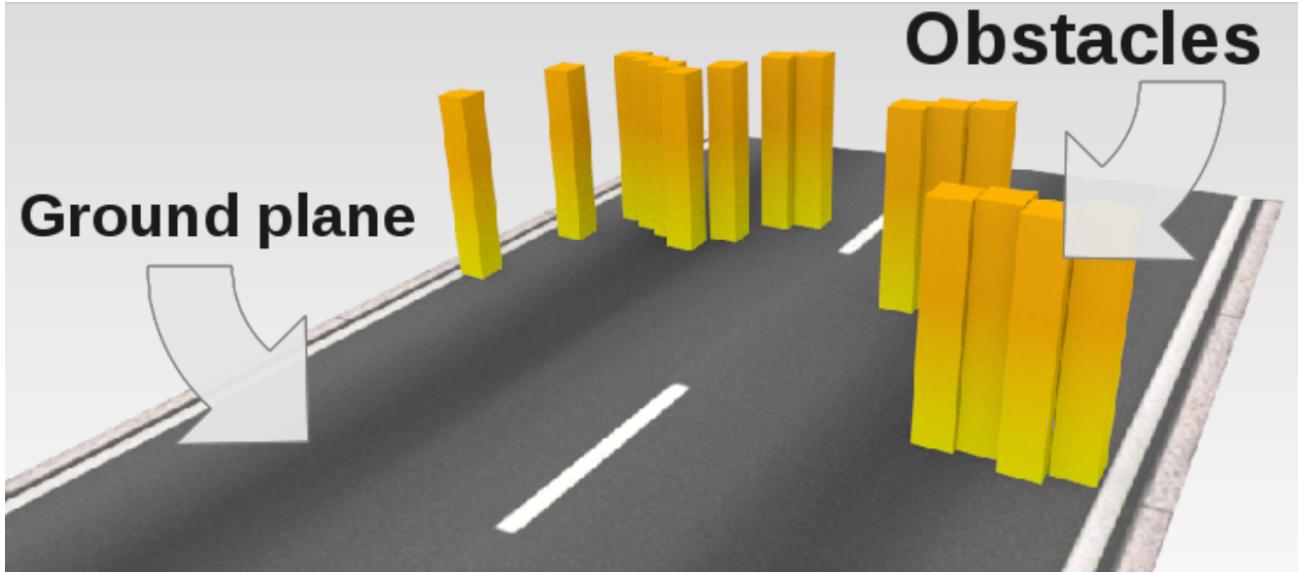


Рис. 4: Упрощённое представление мира в модели Stixel World [4]

Пример сегментации изображения, получаемый с использованием модели Stixel World, представлен на рис. 5.

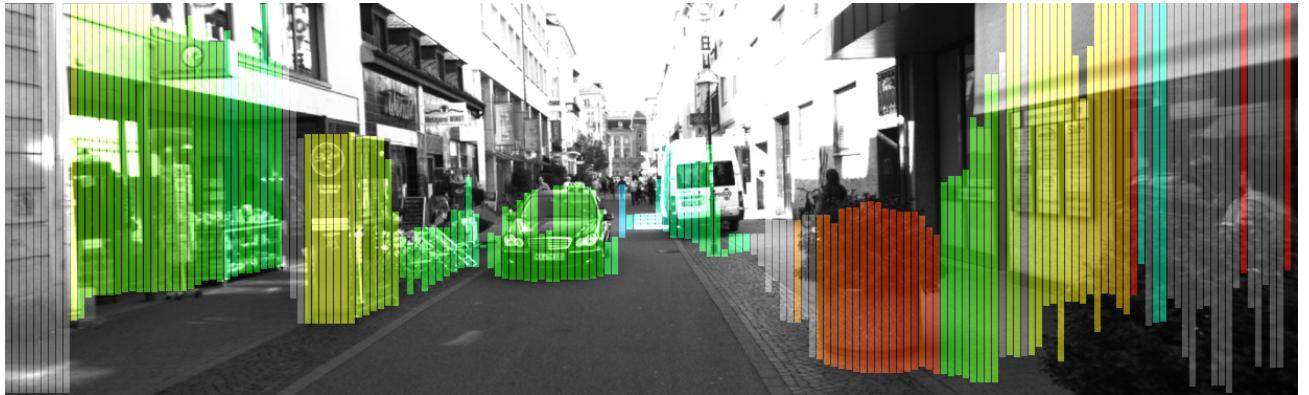


Рис. 5: Пример сегментации препятствий с помощью стикселей. Цветом закодировано расстояние.

Для построения модели Stixel World по двум изображениям, полученным с помощью стереокамеры, авторы предлагают минимизировать сумму двух целевых функций.

$$d_s^*(u) = \operatorname{argmin}_{d(u)} \sum_u c_s(u, d(u)) + \sum_{u_a, u_b} s_s(d(u_a), d(u_b)) \quad (2)$$

$$c_s(u, d) = c_o(u, d) + c_g(u, d) \quad (3)$$

$$c_o(u, d) = \sum_{v=v(h_o, d)}^{v(d)} c_m(u, v, d) \quad (4)$$

$$c_g(u, d) = \sum_{v=v(d)}^{|V|} c_m(u, v, f_{ground}(v)) \quad (5)$$

$$s_s(d_a, d_b) = \begin{cases} \infty, & \text{если } d_a < d_b - 1 \\ c_o(u_a, d_a) & \text{если } d_a = d_b - 1 \\ 0, & \text{если } d_a > d_b - 1 \end{cases} \quad (6)$$

В формулах (2) – (6) приняты следующие обозначения:

- u – порядковый номер стикселя,
- v – индекс строки изображения,
- d_s^* – вычисленное значение диспаритета для стикселя под номером u ,
- h_0 – минимальная высота объекта на сцене, в метрах,
- $c_m(u, v, d)$ – значение функции схожести в точке (u, v) для диспаритета d ,
- $f_{ground}(v)$ – диспаритет плоскости дороги для строки v изображения,
- u_a, u_b – индексы соседних стикселей.

Также авторы предлагают способ оценки высоты объектов с помощью оценки высоты стикселей. Для этого вводится схожая целевая функция, которую мы опустим для краткости.

Такая целевая функция допускает минимизацию на ректифицированной паре изображений с помощью техник динамического программирования, что и было предложено авторами модели Stixel World.

Данный подход требует значительно больше вычислительных ресурсов, в отличие от предыдущих, однако авторы [4] сообщают, что алгоритм может быть оптимизирован для работы в реальном времени.

Авторами работы [4] была выполнена и выложена в открытый доступ реализация [5] вычисления параметров модели Stixel World для пары изображений, полученных с помощью стереокамеры.

2.6 Подходы к детектированию маркеров дорожной разметки

В рамках задачи детектирования маркеров дорожной разметки было проведено большое количество исследований[22]. Множество подходов основано на методах глубокого обучения, однако мы отказались от таких методов в силу специфики задачи.

Альтернативой являются подходы на основе классического компьютерного зрения и параметрических моделей. Простейшей параметрической моделью является прямая. Оказывается, что детектирования прямых достаточно в случае движения по автомагистралям[22].

Работы, связанные с параметрическими моделями, во многом схожи, поэтому здесь приведём только две наиболее релевантные.

Подход, предложенный авторами работы [21] состоит из двух этапов. Сначала к изображению применяется преобразование Bird's Eye View, проецирующее плоскость камеры на плоскость дороги, затем производится выделение выраженных горизонтальных линий в пространстве Хафа, соответствующих набору маркеров дорожной разметки.

Подход, предложенный авторами работы [3] во многом схож с подходом [21]. Однако линии задаются более сложной моделью – в виде кубических сплайнов. Подбор параметров такой модели осуществляется с помощью алгоритма RANSAC[7]. Однако такое усложнение модели ведёт к увеличению количества шума.

3 Алгоритм поиска препятствий движению автомобиля

В данном разделе описан предложенный и реализованный в виде прототипа алгоритм поиска препятствий с помощью стереокамеры. Основные идеи данного подхода взяты из модели Stixel World [19], описанной в разделе 2.5, однако вместо подбора параметров модели с помощью пары изображений, полученных стереокамеры, предложенный нами алгоритм производит подбор параметром модели с помощью карты диспаритета, построенное по паре изображений.

3.1 Поиск препятствий с помощью плотной карты диспаритета

В данном разделе описан алгоритм выделения препятствий с помощью модели Stixel World, вычисленной с помощью плотной карты глубины.

Использование плотной карты глубины вместо исходной пары изображений ведёт к модификации целевой функции. В связи с тем, что расстояние до точек изображения уже вычислено, для построения модели Stixel World достаточно для каждого стикселя вычислить координаты верхней и нижней границ. Модифицированная целевая функция для оценки нижней границы стикселей имеет вид, описанный в формуле 7. Отметим, что размер левого изображения, правого изображения и карты диспаритетов мы считаем одинаковым.

В формулах (7) – (11) принятны следующие обозначения, в целом соответствующие обозначениям раздела 2.5.

- u – индекс стикселя.
- v – индекс строки изображения.
- $v_{bot}(u)$ – нижняя граница стикселя с индексом u .
- $v_{bot}^*(u)$ – оптимальная нижняя граница стикселя с индексом u .
- $|V|$ – количество строк изображения.

$$v_{bot}^*(u) = \operatorname{argmin}_v \left[\sum_u c_s(u, v) + \sum_{u_a, u_b: |u_a - u_b|=1} s_s(v_{bot}(u_a), v_{bot}(u_b)) \right] \quad (7)$$

Формула (7) состоит из двух слагаемых. Первое слагаемое, $c_s(u, v)$, представляет собой оценку правдоподобности наличия нижней границы препятствия в строке v стикселя u и имеет вид, описанный в формулах (8) – (10).

$$c_s(u, v) = c_o(u, v) + c_g(u, v) \quad (8)$$

$$c_o(u, v) = \sum_{v' = v(h_o, d(u, v))}^v |d(u, v) - d(u, v')| \quad (9)$$

$$c_g(u, v) = \sum_{v' = v}^{|V|} |d(u, v) - d_{ground}(u, v')| \quad (10)$$

В формулах (8) – (10) приняты следующие обозначения.

- $c_o(u, v)$ – функция правдоподобности наличия препятствия в стикселе u на строке изображения v .
- $c_g(u, v)$ – функция правдоподобности отсутствия препятствия в стикселе u в во всех строках изображения $v' : v' > v$.
- $d_{ground}(u, v)$ – ожидаемое значение диспаритета в стикселе u , в строке v для плоскости дорожного полотна.
- $d(u, v)$ – медиана значений в строке v карты диспаритетов, соответствующих стикселям u , т.е. медиана значений карты диспаритетов на позициях $(w * u, v) \dots (w * (u + 1) - 1, v)$, где w – предопределённая ширина стикселя.

Второе слагаемое формулы (7) представлено в формуле (11). c_{jump} в этой формуле обозначает наперёд заданный порог.

$$s_s(v_a, v_b) = \begin{cases} \infty, & \text{если } |v_a - v_b| > c_{jump} \\ |v_a - v_b| & \text{иначе} \end{cases} \quad (11)$$

Функция оценки верхней границы стикселей имеет схожий вид и представлена в формулах (12) – (13).

$$v_{top}^*(u) = \operatorname{argmin}_v \sum_u c_s^h(u, v) + \sum_{v_{top}^a, v_{top}^b} s_s(v_{top}^a, v_{top}^b) \quad (12)$$

$$c_s^h(u, v) = \sum_{v' = v}^{v_{bot}} |d(u, v) - d(u, v')| - k * |d(u, v_{bot})| * |v' - v| \quad (13)$$

Для вычисления функции (10) требуется знание диспаритета плоскости дорожного полотна в стикселе u в строке v . Для этого мы используем подход, предложенный в статье [14] и описанный в разделе 2.4.

В листинге 1 представлен псевдокод алгоритма сегментации карты диспаритетов согласно модели Stixel World.

На рис. 6 и 7 приведены примеры работы алгоритма в двух ситуациях. На рис. 6 алгоритм применён для детектирования небольшого объекта (детской игрушки), на рис. 7 верхняя граница стикселей не показана, вместо этого выделена область, ограниченная нижней границей изображения и нижней границей стикселей.

Листинг 1 Сегментация плотной карты диспаритета согласно модели Stixel World

Вход: Карта диспаритета, ширина стикселя

Выход: Множество стикселей, описывающих карту глубины согласно модели Stixel World

- 1: Вычислить значение $d_{ground}(u, v)$, используя алгоритм из раздела 2.4.
 - 2: **Для всех** (u, v)
 - 3: Вычислить значение $c_s(u, v)$, заданное функцией (8).
 - 4: **Конец цикла**
 - 5: С помощью динамического программирования вычислить значения $v_{bot}^*(u_1) \dots v_{bot}^*(u_n)$, заданные функцией (7).
 - 6: **Для всех** (u, v)
 - 7: Вычислить значение $c_s^h(u, v)$, заданное функцией (13).
 - 8: **Конец цикла**
 - 9: С помощью динамического программирования вычислить значения $v_{top}^*(u_1) \dots v_{top}^*(u_n)$, заданные функцией (12).
-

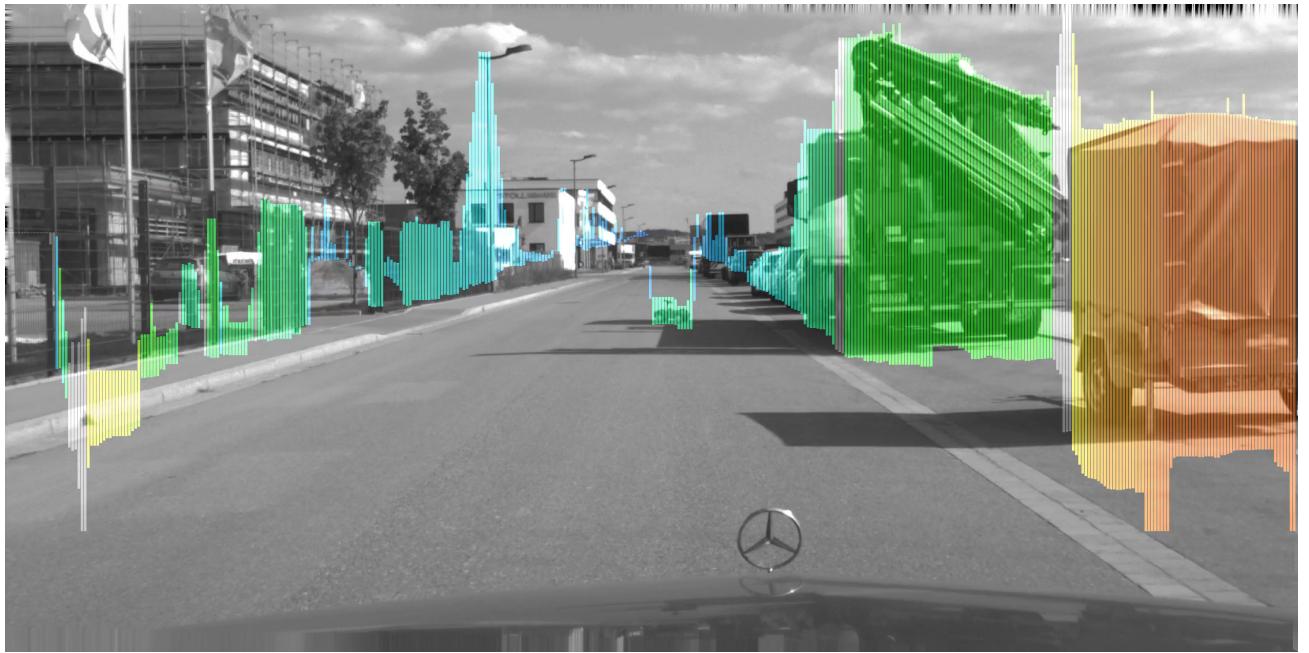


Рис. 6: Сегментация изображения согласно модели Stixel World. Цветом закодировано расстояние.



Рис. 7: Выделение области, безопасной для движения, с помощью сегментации согласно модели Stixel World.

3.2 Поиск препятствий с помощью разреженной карты диспаритета

Реализация алгоритма подбора параметров модели Stixel World для разреженной карты диспаритета представляет интерес, в связи с наличием высокоэффективных алгоритмов вычисления разреженной карты диспаритета, например, CVS[2]. Благодаря сотрудничеству с компанией “Системы компьютерного зрения” нам стала доступна реализация алгоритма CVS, которая и была использована в наших экспериментах.

Прямой перенос алгоритма из раздела 3.1 на карту глубины, вычисленную с помощью алгоритма CVS затруднителен, так как дорожное полотно зачастую низкотекстурировано. Это приводит к тому, что количество точек дорожного полотна на изображении, в которых вычислено значение диспаритета, оказывается небольшим по сравнению с количеством точек, принадлежащим препятствиям. Следствием этого является зачастую неверно вычисленные параметры модели плоскости земной поверхности.

В связи с этим мы модифицировали исходный алгоритм.

Идея модифицированного алгоритма состоит в следующем. Оценка плоскости дорожного полотна с помощью $vdisparity$ заменяется оценкой с помощью алгоритма RANSAC. При этом для подбора параметров модели плоскости дорожного полотна используются только те точки изображения, в которых вычислено значение диспаритета и которые находятся на небольшом удалении в смысле нормы l_1 от середины нижнего края изображения. Связано это с тем, что в случае расположения камеры на лобовом стекле автомобиля и предварительном удалении капота автомобиля с изображения, как это сделано во всех использованных нами наборах данных, середина нижнего края изображения соответствует точке непосредственно перед автомобилем. То есть, в случае, если соблюдена нужная дистанция, эти точки соответствуют только дороге.

Далее происходит вычисление нижней границы стикселей в манере алгоритма из раздела 3.1, однако в связи с низкой плотностью карты диспаритета целевой функционал претерпел следующие изменения. С использованием модели плоскости дорожного полотна, вычисленной ранее, точки изображения, для которых было вычислено значение диспаритета помечаются как “дорога”, если они удовлетворяют модели плоскости дорожного полотна и “препятствие” иначе. Далее целевая функция нижней границы вычисляется аналогично формуле (7), однако первое слагаемое этой формулы заменяется на формулу (14). В формуле (14) $object(u, v')$ обозначает количество точек изображения в строке v' стикселя u , помеченных как “препятствие”, аналогично $ground(u, v')$ обозначает количество точек изображения в строке v' стикселя

u , помеченных как “дорога”. Отметим, что такая функция может быть эффективно вычислена с использованием нарастающих сумм.

$$c_s(u, v) = \sum_{v'}^{|V|} object(u, v') + \sum_0^{v'} ground(u, v') \quad (14)$$

Вычисление верхней границы стикселей для разреженной карты глубины нами не проводилось, так как, с одной стороны, оно не является необходимым для детектирования объектов, с другой стороны, является недостаточно робастным в следствие разреженности стерео.

Псевдокод модифицированного алгоритма приведён в листинге 2.

Листинг 2 Сегментация разреженной карты диспаритета согласно модели Stixel World

Вход: Разреженная карта диспаритета, ширина стикселя

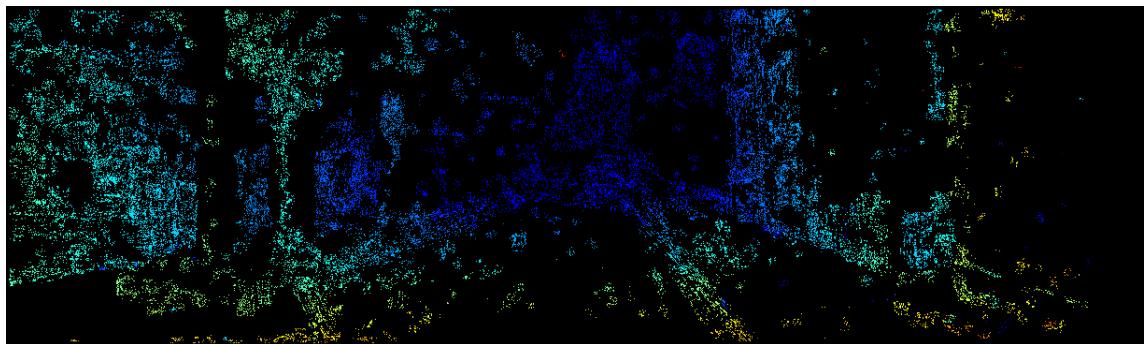
Выход: Множество нижних границ стикселей, описывающих карту глубины согласно модели Stixel World

- 1: Из всех точек карты диспаритета выбрать те, для которых выполнено неравенство $\max(|y_i - height_{img}|, |x_i - (width_{img}/2)|) < C$. Обозначим множество таких точек P_{road} .
 - 2: Вычислить нормаль к плоскости дороги, максимизируя количество точек из множества P_{road} , принадлежащих плоскости дороги при помощи алгоритма RANSAC.
 - 3: Вычислить карту стоимости согласно формуле (14).
 - 4: С помощью динамического программирования вычислить значения $v_{bot}^*(u_1) \dots v_{bot}^*(u_n)$, заданные функцией (7).
-

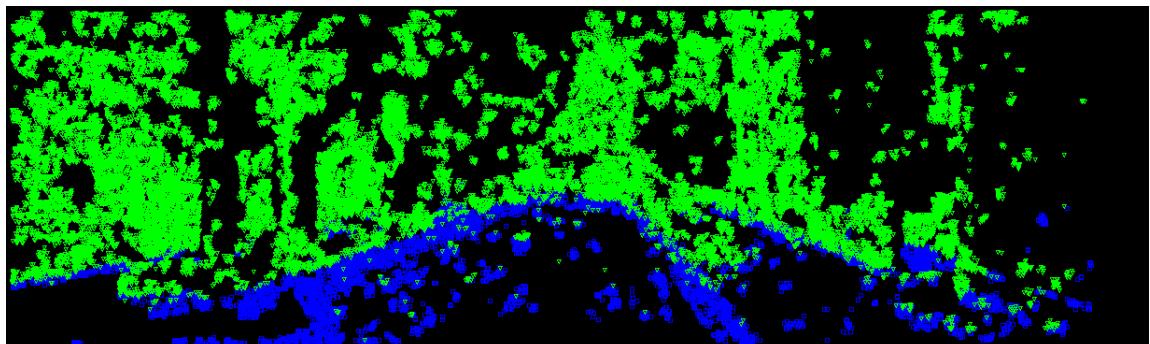
На рис. 8 представлена визуализация шагов алгоритма, описанного в данном разделе. На рис. 8.b видно, что количество точек на дороге невелико. Как упоминалось выше, это может привести к неверному определению плоскости дороги. Для того, чтобы снизить этот эффект, возможно считать положение плоскости дороги относительно камеры постоянным, что, однако, приводит к существенному снижению точности определения нижней границы объекта в типичном случае.



(a) Левое ректифицированное изображение



(b) Карта диспаритета, вычисленная с помощью алгоритма CVS



(c) Результат разделения неплотного облака стерео-точек на классы “дорога” и “препятствия”



(d) Результат оценки нижней границы стикселей

Рис. 8: Этапы вычисления нижней границы стикселей с помощью алгоритма из листинга 2

4 Алгоритм детектирования маркеров дорожной разметки

На основе алгоритме из статьи [3] нами был разработан алгоритм детектирования маркеров дорожной разметки. Данный алгоритм рассчитан на поиск маркеров дорожной разметки в виде прямых линий. Алгоритм включает в себя защиту от ложных срабатываний на некоторых дополнительных элементах разметки: пешеходный переход, остановка общественного транспорта, маркеры направлений движения. Кроме того, с помощью разработанного алгоритма возможно детектировать, например, бордюр, что необходимо в условиях движения в условиях города.

Псевдокод разработанного алгоритма представлен в листинге 3.

Листинг 3 Выделение маркеров дорожной разметки на изображении

Вход: цветное изображение img , калибровочные данные для камеры

Выход: Маска для изображения img , на которой отличные от нуля значения соответствуют маркерам дорожной разметки

- 1: Применить к img преобразование Bird's Eye View.
 - 2: Преобразовать изображение в пространство HSV и извлечь канал Value, обозначим это новое изображение $imgHsvValue$.
 - 3: Применить морфологическое размыкание к изображению $imgHsvValue$ для удаления недостаточно тонких линий.
 - 4: Вычислить в каждой точке $imgHsvValue$ значение оператора Собеля, обозначим получившееся изображение $imgSobel$.
 - 5: На $imgSobel$ приравнять к нулю все точки, направление градиента в которых существенно отличается от горизонтального.
 - 6: Разбить $imgSobel$ на непересекающиеся прямоугольники одинакового размера, обозначим каждый такой прямоугольник $Rect(i, j)$.
 - 7: **Для всех** (i, j)
 - 8: Пометим $Rect(i, j)$ как содержащий разметку, если сумма пикселей, попавших в этот прямоугольник больше порога.
 - 9: **Конец цикла**
 - 10: Интерпретируя центры $Rect(i, j)$ как точки на изображении, с помощью алгоритма RANSAC поочерёдно извлекаем все линии разметки.
-

Отметим, что отсутствие срабатываний на пешеходных переходах и на маркерах направлений движения достигается за счёт строки 3 листинга, а отсутствие срабатываний на остановках общественного транспорта достигается за счёт строки 5.

На рис. 9 изображены промежуточные результаты алгоритма. Слева направо на изображении представлены результаты после шагов 5, 9, 10.

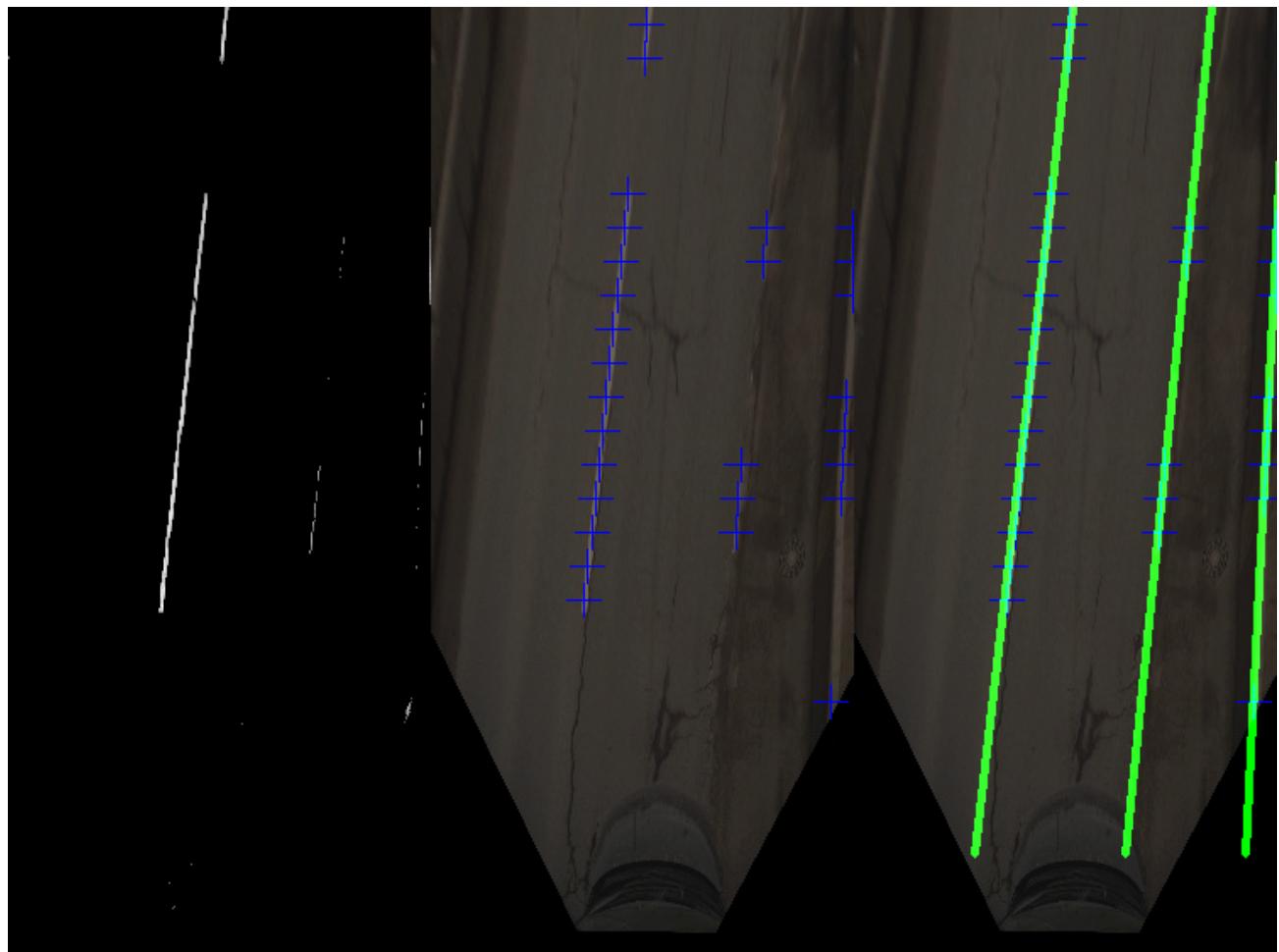


Рис. 9: Пример работы алгоритма выделения маркеров дорожной разметки

5 Апробация

5.1 Данные

Для оценки качества разработанных алгоритмов были выбраны следующие наборы данных.

- Тестовый набор для алгоритмов детектирования объектов KITTI [8].
- Тестовый набор для алгоритмов детектирования маркеров дорожной разметки TuSimple [26].
- Тестовый набор для алгоритмов детектирования маркеров дорожной разметки “Петергоф”, собранный специалистами компании Ланит-Терком и предоставленный нам для оценки алгоритма.

Набор данных KITTI состоит из приблизительно 7000 изображений, полученных с помощью стереокамеры, установленной на крыше автомобиля. Изображения были получены в рамках движения автомобиля по дорогам общего пользования Германии как в городской среде, так и в сельской местности, в различных погодных условиях. Каждое изображение проаннотировано прямоугольниками, описанными вокруг объектов, принадлежащих следующим классам: велосипедист, пешеход и автомобиль.

Пример изображения из набора данных KITTI представлен на рис. 10.

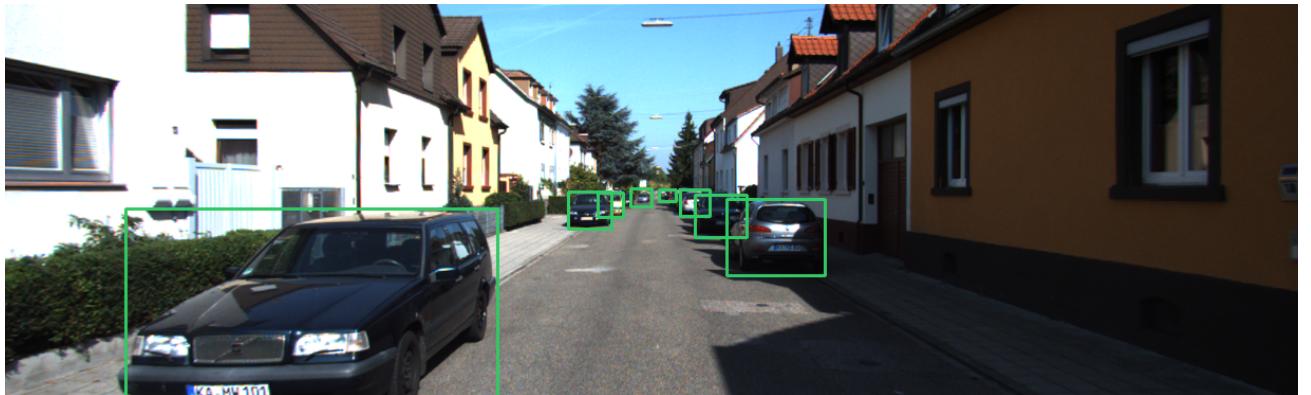


Рис. 10: Пример аннотированного изображения из набора данных KITTI

Набор данных TuSimple представляет собой набор из 358 изображений, снятых камерой, закреплённой на автомобиле в рамках проезда автомобиля по дорогам общего пользования. Для каждого изображения присутствует аннотация, содержащая маркеры дорожной разметки. В аннотации присутствует не более пяти маркеров: два маркеров для текущей полосы движения

автомобиля, два для соседних полос и пятый маркер для случая перестройки из одной полосы в другую, когда невозможно определить, какая полоса является текущей.

Пример изображения из набора TuSimple представлен на рис. 11.

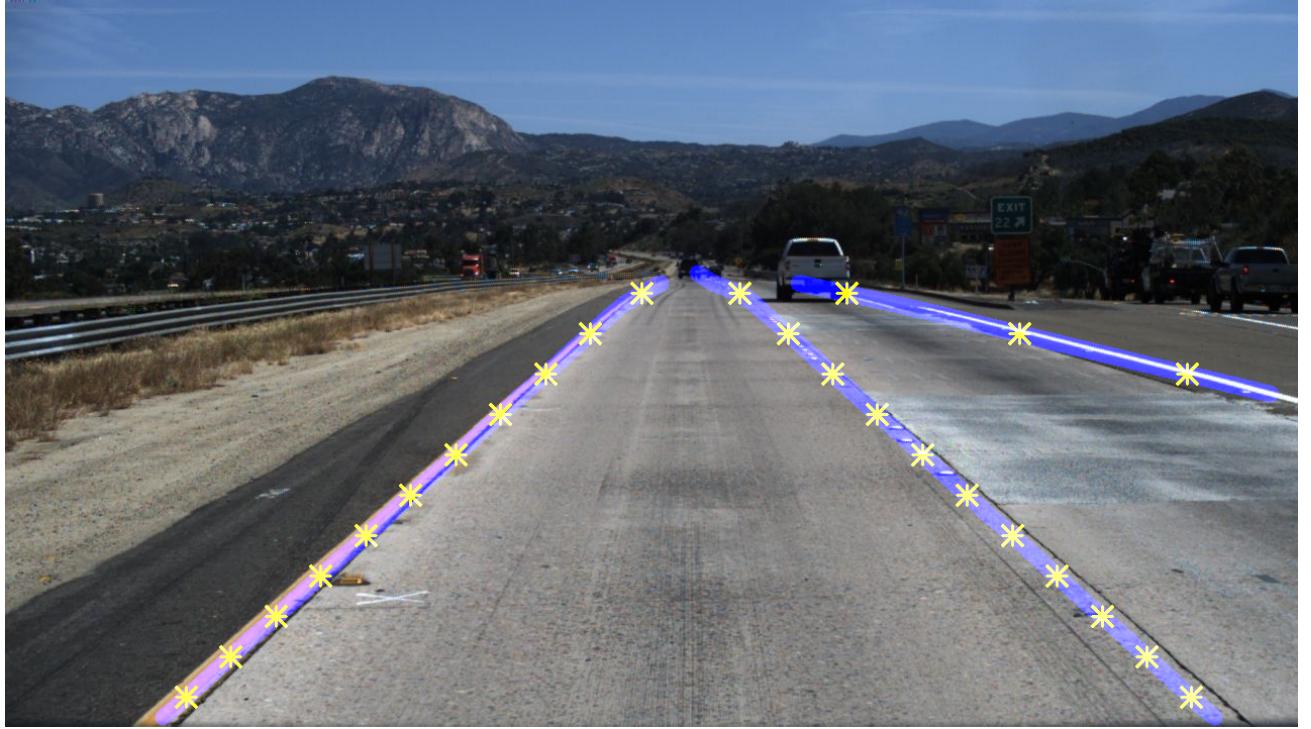


Рис. 11: Пример аннотированного изображения из набора данных TuSimple

Набор данных Петергоф, представляет собой видеопоследовательность длительностью 5 минут из 5000 кадров, записанную в рамках движения автомобиля по дороге г. Петергофа. Отличительной особенностью этой последовательности является наличие бордюров, заменяющих крайние маркеры дорожной разметки. Каждый 50-й кадр этой последовательности был нами размечен: на изображении были выделены в виде маски маркеры дорожной разметки, аналогично последовательности TuSimple. При этом бордюры мы также считали маркерами разметки.

Пример изображения из набора Петергоф представлен на рис. 12.

5.2 Оценка качества работы алгоритмов

5.2.1 Методика оценки алгоритма детектирования препятствий

Так как выбранные наборы данных содержат аннотацию в виде описывающих прямоугольников, было решено применить следующий способ оценки алгоритма.

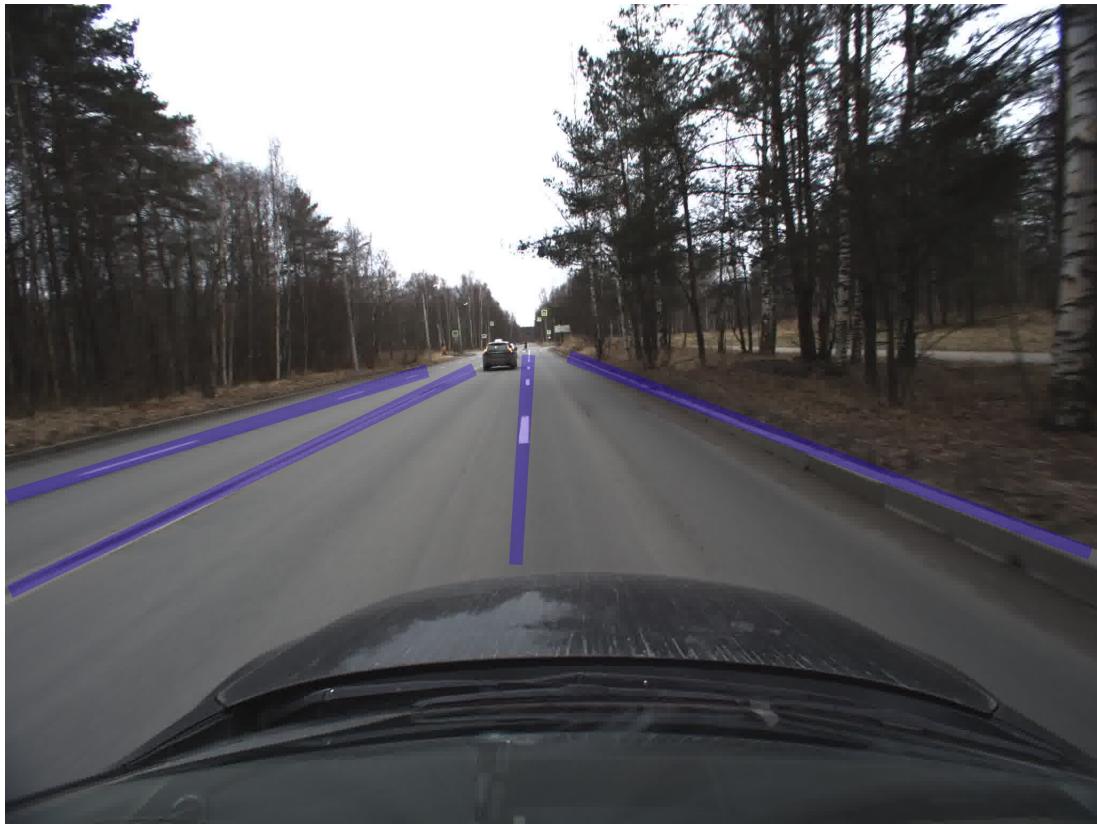


Рис. 12: Пример аннотированного изображения из набора данных Петергоф

- Подобрать параметры модели Stixel World для изображения, т.е. вычислить набор стикселей.
- Для каждого описанного прямоугольника из всех стикселей выбрать те, горизонтальная координата которых заключена между левой и правой границей прямоугольника.
- Для выбранных стикселей вычислить медиану нижней границы. Обозначим эту медиану $v_{predicted}$. Обозначим координату нижней границы прямоугольника $v_{expected}$.
- Далее, в зависимости от значения $v_{diff} = v_{predicted} - v_{expected}$, мы рассматриваем три случая:
 1. $|v_{diff}| < 0.2 * bb_{height}$ – препятствие обнаружено,
 2. $v_{diff} < 0.2 * bb_{height}$ – препятствие не обнаружено,
 3. $v_{diff} > 0.2 * bb_{height}$ – обнаружено препятствие ниже.

В наборе данных KITTI присутствует большое количество вырожденных случаев, когда в следствие перекрытий объекту соответствует слишком малое

количество пикселей. Такие объекты, как правило, расположены на горизонте и не имеют достаточного параллакса для их выделения с помощью алгоритмов стереоизрения. Кроме того, расчёт карты диспаритета затруднителен на краях изображения. В связи с этим было решено исключить из тестовой выборки следующие объекты.

- Объекты, ширина или высота описывающего прямоугольника для которых не превышает 25 пикселей.
- Объекты, центр которых расположен на расстоянии менее 200 пикселей от вертикального края изображения.

5.2.2 Методика оценки алгоритма детектирования маркеров дорожной разметки

На наборе данных TuSimple нами принята следующая методика оценки результата.

- С помощью разработанного нами алгоритма вычислить и сохранить предполагаемые положения маркеров дорожной разметки.
- Для каждого найденного нашим алгоритмом маркера выполнить поиск в аннотации размеченного положения маркера, с которым есть пересечение на 50 и более процентов.
- Если такой маркер из аннотации был найден, пометить его как обнаруженный.
- Если такой маркер из аннотации не был найден, считать найденный нашим алгоритмом маркер ложным срабатыванием.
- Если для какого-то маркера из аннотации не нашлось обнаруженного нашим алгоритмом маркера, считать, что маркер из аннотации не был обнаружен.

Кроме того, два ближайших к центру изображения относительно горизонтальной оси маркера из аннотации мы считаем границами собственной полосы, что позволяет разделить маркеры на два класса: маркеры собственной полосы и маркеры соседних полос.

5.2.3 Результаты оценки алгоритма детектирования препятствий

В таблице 1 представлены результаты сравнения нашей реализации подбора параметров модели Stixel World с реализацией [5]. Столбец “Dense” соответствует результату алгоритма из раздела 3.1, столбцы “SparseR” и “SparseC” соответствуют результату алгоритма из раздела 3.2, при этом в “SparseR” использован подбор параметров плоскости дорожного полотна с помощью алгоритма RANSAC, а в “SparseC” плоскость дороги принята постоянной для всех изображений. Столбец Benenson et al. соответствует открытой реализации [5]. Стока “перекрытие” соответствует случаю “обнаружен объект ниже”. Это означает, что для этих объектов стиксели, соответствующие столбцам прямоугольника, описанного вокруг объекта, имеют нижнюю границу существенно ниже, чем нижняя граница описывающего прямоугольника.

Видно, что как в нашей реализации, так и в реализации Benenson et al. большое количество объектов находится в строке “перекрытие”. Связано это может быть с двумя причинами: либо на изображении действительно находится препятствие, расположенное ниже объекта, заданного описанным прямоугольником, либо имеется ложное срабатывание. Отделить один случай от другого возможно только для объектов, перекрытых объектами, присутствующими в разметке. Для этого необходимо удалить из тестового множества все такие прямоугольники a_i , что существует b_i , отличный от него, пересекающийся с ним и имеющий нижнюю границу ниже, чем нижняя граница a_i . Результаты работы алгоритмов на таком усечённом наборе данных приведены в таблице 2. Обозначения повторяют обозначения таблицы 1.

Заметно, что и на исходном наборе данных, и на усечённом, наша реализация выигрывает в количестве обнаруженных объектов. Тем не менее, результаты запуска на наборе данных KITTI показали, что большинство неверных результатов нашего алгоритма обусловлено двумя причинами.

- Наличие артефактов на карте диспаритета, представляющих собой области на карте диспаритета, имеющие неверное, либо вовсе не имеющие значения диспаритета. Артефакты возникают, как правило, на зеркальных поверхностях и в засвеченных областях изображения и связаны с нарушением условий, необходимых для корректной работы алгоритмов расчёта карты диспаритета.
- Неверная оценка плоскости дороги. Выбранная нами модель плоскости дороги слишком проста и не в состоянии корректно описать дорожное полотно в некоторых ситуациях. Например, при наличии крена. Кроме того, в виду наличия большого количества объектов на сцене, подбор параметров модели дорожного полотна с использованием подхода

	Dense	SparseR	SparseC	Benenson et al.
Обнаружено	15780	11889	12635	10168
Не обнаружено	355	4116	3714	9315
Перекрытие	5692	5822	5478	2344

Таблица 1: Сравнение реализаций модели Stixel World без фильтрации перекрытий

	Dense	SparseR	SparseC	Benenson et al.
Обнаружено	11167	7738	7817	6164
Не обнаружено	271	2891	2983	6505
Перекрытие	1868	2677	2506	637

Таблица 2: Сравнение реализаций модели Stixel World с фильтрацией перекрытий

vdisparity может привести к недостаточно точным значениям параметров. Также для случая неплотного стерео к неверной оценке плоскости дорожного полотна может привести недостаточная текстурированность дороги, необходимая для качественной работы алгоритма CVS.

5.2.4 Результаты оценки алгоритма детектирования маркеров дорожной разметки

В таблице 3 представлены результаты запуска разработанного нами алгоритма детектирования маркеров дорожной разметки на наборе данных TuSimple. Из этой таблицы, например, видно, что точность детектирования маркеров собственной полосы выше, что неудивительно, так как этим маркерам соответствует большее число пикселей изображения.

В таблице 4 представлены результаты запуска алгоритма на последовательности Петергоф. Видно, что процент верно обнаруженных маркеров собственной полосы (включая бордюры), так же высок, как и на последовательности TuSimple. Кроме того, уровень ложных срабатываний достаточно низок. Однако, снизился также и процент обнаруженных маркеров соседних полос движения.

	Собственная полоса	Несобственная полоса
Обнаружено	146	16
Не обнаружено	42	63
Ложное срабатывание		4

Таблица 4: Результаты работы алгоритма детектирования маркеров дорожной разметки на наборе данных Петергоф

	Собственная полоса	Несобственная полоса
Обнаружено	634	196
Не обнаружено	80	208
Ложное срабатывание		98

Таблица 3: Результаты работы алгоритма детектирования маркеров дорожной разметки на наборе данных TuSimple

Улучшение качества детектирования на последовательности Петергоф связано с тем, что на этой последовательности маркеры дорожной разметки чётко видны, в то время как на последовательности TuSimple качество маркеров варьируется. Это позволяет сделать вывод, что в условиях наличия качественной дорожной разметки, разработанный алгоритм справляется с возложенной на него задачей.

На рис. 13 приведены некоторые случаи ошибок разработанного алгоритма. Так, заметно, что в случае недостаточно качественной разметки данный алгоритм не справляется с детектированием полосы. Кроме того, такие элементы как край дороги, либо полосы на объектах определённой ширины могут быть восприняты в качестве маркеров дорожной разметки. С первой проблемой справиться возможно только путём существенных модификаций алгоритма. Например, добавлением новых случаев. Вторая проблема теоретически может быть решена, например, путём интеграции с алгоритмами из раздела 3. А именно в качестве предобработки возможно удалить с изображения область, соответствующую препятствиям и выполнять поиск маркеров в оставшейся области. Нами проводились соответствующие эксперименты, которые показали улучшение качества детектирования маркеров дорожной разметки, однако в нашем распоряжении не оказалось убедительного набора данных, чтобы достоверно подтвердить эффективность такого подхода.



(a)



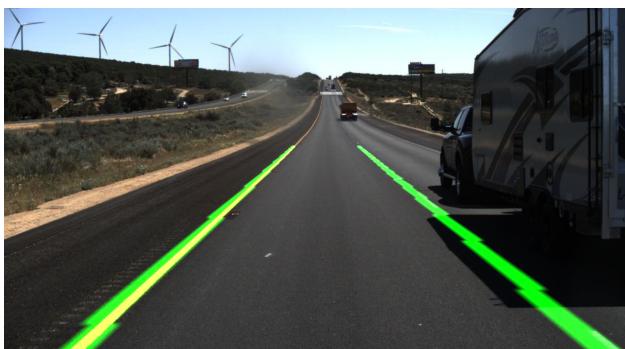
(b)



(c)



(d)



(e)



(f)

Рис. 13: Примеры неверной работы алгоритма детектирования маркеров дорожной разметки на последовательности TuSimple. (а), (д), (ф) – некачественная разметка дорожного полотна. (е) – полоса заслонена. (б), (с) – ложные срабатывания.

6 Заключение

В рамках данной работы были достигнуты следующие результаты.

- На основе подхода Stixel World и классического стереозрения разработан и реализован на языке C++ прототип алгоритма решения задачи поиска препятствий движению автомобиля на изображении, полученном с помощью стереокамеры, закреплённой на лобовом стекле автомобиля с помощью карты диспаритета. Реализована модификация этого алгоритма для работы с неплотной картой диспаритета.
- На основе методов классической обработки изображений разработан и реализован на языке Python прототип алгоритма поиска маркеров дорожной разметки на изображении, полученном с камеры, закреплённой на лобовом стекле автомобиля.
- Выполнена апробация разработанных алгоритмов на наборах данных KITTI и TuSimple, а также на наборе данных Петергоф.

В качестве дальнейшего продолжения работы планируется предпринять следующие шаги.

- Разметить тестовый набор и проверить предположение об увеличении качества детектирования маркеров дорожной разметки при интеграции с алгоритмом детектирования препятствий.
- Улучшить алгоритм детектирования маркеров дорожной разметки для случая нечётких маркеров.
- Добавить синхронизацию предсказаний алгоритмов для последовательных кадров видеопоследовательности.

Список литературы

- [1] 6d-vision: Fusion of stereo and motion for robust environment perception / Uwe Franke, Clemens Rabe, Hernán Badino, Stefan Gehrig // Joint Pattern Recognition Symposium / Springer. — 2005. — P. 216–223.
- [2] A. Pimenov V. Ufnarovsky T. Bryksin A. Terekhov. Stereo-approach distance and speed meter. — US Patent 9930314B2. — 2014.
- [3] Aly Mohamed. Real time detection of lane markers in urban streets // Intelligent Vehicles Symposium, 2008 IEEE / IEEE. — 2008. — P. 7–12.
- [4] Benenson Rodrigo, Timofte Radu, Van Gool Luc. Stixels estimation without depth map computation // Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on / IEEE. — 2011. — P. 2010–2017.
- [5] Benenson Rodrigo et al. Doppia Repository. — <https://bitbucket.org/rodrigob/doppia>. — Accessed in April, 2018.
- [6] CVS Co. Ltd. CVS official web page. — <http://www.compvisionsys.ru/home-en-US/>. — Accessed in April, 2018.
- [7] Fischler Martin A, Bolles Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography // Readings in computer vision. — Elsevier, 1987. — P. 726–740.
- [8] Geiger Andreas, Lenz Philip, Urtasun Raquel. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite // Conference on Computer Vision and Pattern Recognition (CVPR). — 2012.
- [9] Godard Clément, Mac Aodha Oisin, Brostow Gabriel J. Unsupervised Monocular Depth Estimation with Left-Right Consistency // CVPR. — 2017.
- [10] Heinrich Stefan. Fast obstacle detection using flow/depth constraint // Intelligent Vehicle Symposium, 2002. IEEE / IEEE. — Vol. 2. — 2002. — P. 658–665.
- [11] Hirschmuller Heiko. Accurate and efficient stereo processing by semi-global matching and mutual information // Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on / IEEE. — Vol. 2. — 2005. — P. 807–814.

- [12] Itseez. Open Source Computer Vision Library. — <https://github.com/itseez/opencv>. — Accessed in July, 2017.
- [13] Koenderink Jan J, Van Doorn Andrea J. Affine structure from motion // JOSA A. — 1991. — Vol. 8, no. 2. — P. 377–385.
- [14] Labayrade Raphael, Aubert Didier, Tarel J-P. Real time obstacle detection in stereovision on non flat road geometry through “v-disparity” representation // Intelligent Vehicle Symposium, 2002. IEEE / IEEE. — Vol. 2. — 2002. — P. 646–651.
- [15] Lanit-Tercom Co. Ltd. Lanit-Tercom official web page. — <http://www.lanit-tercom.ru/en/homepage>. — Accessed in April, 2018.
- [16] Lipton Zachary C. The mythos of model interpretability // arXiv preprint arXiv:1606.03490. — 2016.
- [17] Lu Meng, Wevers Kees, Van Der Heijden Rob. Technical feasibility of advanced driver assistance systems (ADAS) for road traffic safety // Transportation Planning and Technology. — 2005. — Vol. 28, no. 3. — P. 167–187.
- [18] On building an accurate stereo matching system on graphics hardware / Xing Mei, Xun Sun, Mingcai Zhou et al. // Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on / IEEE. — 2011. — P. 467–474.
- [19] Pfeiffer David, Franke Uwe. Efficient representation of traffic scenes by means of dynamic stixels // Intelligent Vehicles Symposium (IV), 2010 IEEE / IEEE. — 2010. — P. 217–224.
- [20] Prosense Co. Ltd. Prosense official web page. — <http://www.prosensetek.com/theme/f010/eg/index.php>. — Accessed in April, 2018.
- [21] Real-time lane detection and forward collision warning system based on stereo vision / Wenjie Song, Mengyin Fu, Yi Yang et al. // Intelligent Vehicles Symposium (IV), 2017 IEEE / IEEE. — 2017. — P. 493–498.
- [22] Recent progress in road and lane detection: a survey / Aharon Bar Hillel, Ronen Lerner, Dan Levi, Guy Raz // Machine vision and applications. — 2014. — Vol. 25, no. 3. — P. 727–745.

- [23] Shaout Adnan, Colella Dominic M., Awad S Ghada E. A. Advanced Driver Assistance Systems - Past, present and future // 2011 Seventh International Computer Engineering Conference (ICENCO'2011). — 2011. — P. 72–82.
- [24] Staubach Maria. Factors correlated with traffic accidents as a basis for evaluating Advanced Driver Assistance Systems // Accident Analysis & Prevention. — 2009. — Vol. 41, no. 5. — P. 1025–1033.
- [25] Studying effects of advanced driver assistance systems (ADAS) on individual and group level using multi-driver simulation / Christian Maag, Dominik Muhlbacher, Christian Mark, Hans-Peter Kruger // IEEE Intelligent Transportation Systems Magazine. — 2012. — Vol. 4, no. 3. — P. 45–54.
- [26] TuSimple Ltd. TuSimple lane detection benchmark. — <http://benchmark.tusimple.ai>. — Accessed in April, 2018.
- [27] The single frame stereo vision system for reliable obstacle detection used during the 2005 DARPA grand challenge on TerraMax / Alberto Broggi, Claudio Caraffi, Pier Paolo Porta, Paolo Zani // Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE / IEEE. — 2006. — P. 745–752.
- [28] A survey of ADAS technologies for the future perspective of sensor fusion / Adam Ziebinski, Rafal Cupek, Huseyin Erdogan, Sonja Waechter // International Conference on Computational Collective Intelligence / Springer. — 2016. — P. 135–146.