МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ ПРОГРАММ/MATHEMATICAL MODELING, NUMERICAL METHODS AND PROGRAM COMPLEXES

DOI: https://doi.org/10.60797/COMP.2025.7.2

РАЗРАБОТКА ВЫСОКОПРОИЗВОДИТЕЛЬНОГО МОНОКУЛЯРНОГО МЕТОДА ОЦЕНКИ 3Д-ПОЛОЖЕНИЙ ОБЪЕКТОВ НА ПРИМЕРЕ КЛАССА ОБЪЕКТОВ «АВТОМОБИЛЬ» АВТОНОМНОГО НАЗЕМНОГО ТРАНСПОРТА

Научная статья

Ярощук П.О.[1, *]
[1] ORCID : 0009-0006-4311-0666;
[1] Воронежский Государственный Университет, Воронеж, Российская Федерация

* Корреспондирующий автор (yaroschuk[at]amm.vsu.ru)

Аннотация

В данной работе представлен лёгкий и высокопроизводительный метод одновременной 3D-оценки объектов и сегментации дорожной плоскости на основе единой свёрточной сети YOLOv11s. Метод не требует полного восстановления глубины сцены, и имеет модульную структуру и настраиваемые параметры производительности и точности, что позволяет достичь скорости 20–33 FPS на мобильной видеокарте GTX 1650 при сохранении конкурентоспособной точности: mAP BEV (Easy / Moderate / Hard) составляет 17,02% / 19,27% / 19,09%, AOS (Easy / Moderate / Hard) — 82,5% / 66,5% / 59,6%. Предложенный подход демонстрирует компромисс «точность — скорость» и открывает перспективы применения в системах реального времени и встроенных устройствах.

Ключевые слова: монокулярное компьютерное зрение, машинное обучение, обработка изображений, локализация и картирование, автоматическое управление, робототехника, искусственный интеллект.

DEVELOPMENT OF A HIGH-PERFORMANCE MONOCULAR METHOD FOR EVALUATION OF 3D-POSITIONS OF OBJECTS ON THE EXAMPLE OF THE OBJECT CLASS "CAR" OF AUTONOMOUS GROUND TRANSPORT

Research article

Yaroschuk P.O.[1, *]
[1] ORCID : 0009-0006-4311-0666;
[1] Voronezh State University, Voronezh, Russian Federation

* Corresponding author (yaroschuk[at]amm.vsu.ru)

Abstract

This work presents a feasible and high-performance method for simultaneous 3D object evaluation and road plane segmentation based on a unified convolutional network, YOLOv11s. The method does not require full scene depth reconstruction, and has a modular structure and adjustable performance and accuracy parameters, allowing it to achieve 20–33 FPS on a GTX 1650 mobile graphics card while maintaining competitive accuracy: mAP BEV (Easy / Moderate / Hard) is 17.02% / 19.27% / 19.09%, AOS (Easy / Moderate / Hard) is 82.5% / 66.5% / 59.6%. The suggested approach demonstrates the trade-off "accuracy — speed" and opens perspectives of application in real-time systems and integrated devices.

Keywords: monocular computer vision, machine learning, image processing, localisation and mapping, automatic control, robotics, artificial intelligence.

Введение

Существуют различные методы, решающие задачу определения трехмерного положения объектов для автопилотируемого наземного транспорта для монокулярных изображений. К таковым можно отнести, например, Mono3D [7]. Подход к решению заключается в глубоком обучении сверточных нейронных сетей архитектуры YOLO или схожей (обнаруживает сразу все объекты всех классов) с модифицированным выходным слоем, осуществляющим регрессию объемного положения и ориентации обнаруженного класса. В данных подходах наблюдается значительный прогресс и они достигают показателей производительности, позволяющих определять положение в реальном времени, но требовательны к аппаратному обеспечению.

В данной работе рассматривается новый метод, не применяющий регрессию положения через модифицированный выходной слой. Он использует комбинацию классического PnP-решателя и нейросети архитектуры YOLO новейшего поколения (YOLOv11s). Положение выявляется благодаря особому способу обучения нейросети, которая определяет не только сам объект, но и его ключевые визуальные особенности, благодаря чему метод решения PnP определяет конечное положение без дополнительной вычислительной нагрузки на видеоускоритель. Это позволяет достичь значительного (до 10–20 раз) улучшения производительности по сравнению с классическими для этой задачи методами глубокого обучения.

Описание разработанного метода

Система использует калибровочные параметры камеры и априорные знания о геометрии типовых объектов. Предлагаемая архитектура системы включает следующие последовательные этапы обработки:

1. Двумерная детекция объектов: Обнаружение объектов на входном изображении и получение их двумерных ограничивающих рамок и классов с помощью предобученной нейросетевой модели. На этом этапе также могут быть детектированы характерные части объектов, используемые для уточнения.

2. Оценка трехмерного положения опорной точки объекта: Определение начальной трехмерной точки, ассоциированной с объектом, путем сопоставления его двумерной детекции с характерными признаками и последующей проекции на предполагаемую опорную плоскость.

3. Оптимизация ориентации объекта: Итерационное уточнение угла поворота объекта вокруг вертикальной оси путем минимизации ошибки между проекцией его трехмерной модели и наблюдаемыми двумерными признаками.

Входными данными для системы служат: монокулярное RGB-изображение, внутренние и внешние параметры калибровки камеры (калибровочная матрица P), аппроксимирующие 3D боксы для объектов по умолчанию.

Выходными данными являются: трехмерные координаты центра объекта (X_c, Y_c, Z_c) в системе координат камеры. Угол ориентации объекта R_y относительно оси Y камеры.

Использование гибридных моделей также позволяет получать сегментированную поверхность дороги, дорожные знаки и другую полезную информацию без снижения скорости работы. Общая схема представлена на рис. 1.

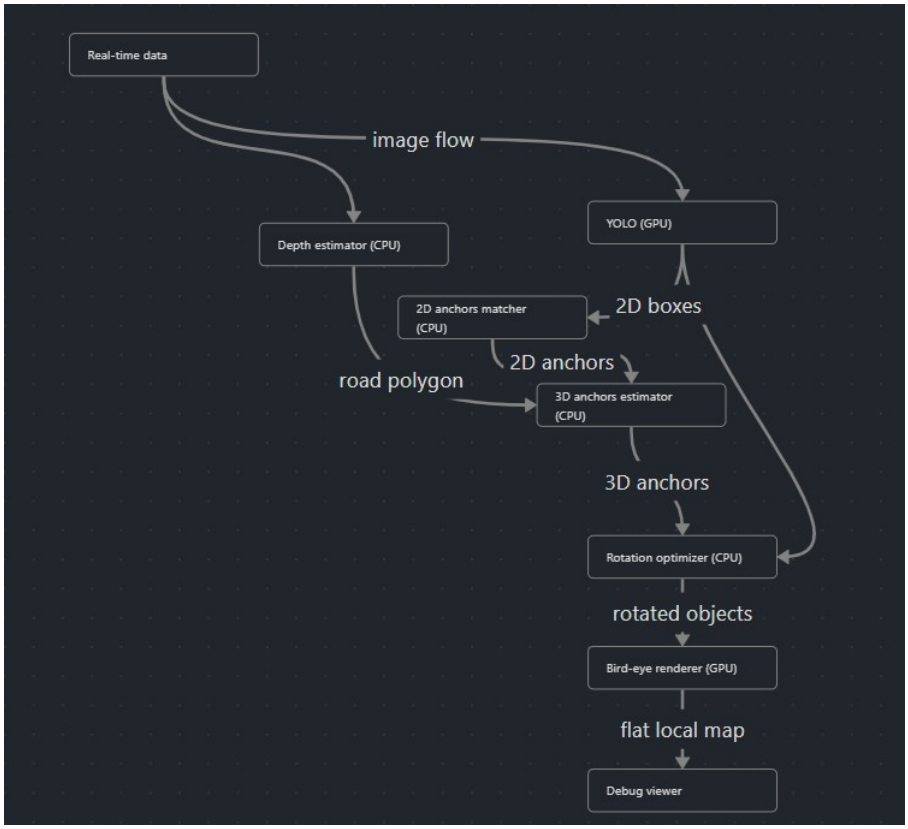

Рисунок 1 - Общая схема разработанного метода
DOI: https://doi.org/10.60797/COMP.2025.7.2.1

Система является логическим развитием представленной в работе [3].

2.1. Основные понятия, термины

Якорь 2D — опорная точка на изображении, которая отвечает положению обнаруженного объекта, привязана к земле. Является отправной для построения якоря 3D.

Якорь 3D — опорная точка в трехмерном пространстве, соответствующая положению объекта на поверхности, по которой он перемещается. Является отправной точкой для построения канонической 3D модели объекта методом решения PnP.

Каноническая 3D модель — габариты объекта выбранного класса.

Тень бокса 3D объекта - проекция канонической 3D модели объекта на плоскость XY мировых координат сцены. Фактически отражает точность положения объекта на плоской карте.

2.2. Модуль двумерной детекции объектов и их характерных признаков

Первичным этапом работы системы является обнаружение объектов на входном изображении. Для этой цели была применена сверточная нейронная сеть, обученная на задачу детекции объектов. Была выбрана YOLOv11s (small), как модель, обеспечивающая достаточную кадровую частоту на GTX1650 (30-35 кадров в секунду) с сохранением приемлемой точности. Более крупные модели (medium, large) уже не способны обеспечить достаточную кадровую частоту. В сравнении с предыдущим поколением демонстрирует прирост точности вместе с улучшением

быстродействия. На данный момент является самой продвинутой среди YOLO-моделей с официальной реализацией от Ultralitycs. Выбранная нейронная сеть была дообучена на наборе классов «car», «car_rear», «car_front» для получения необходимых данных для дальнейшей работы алгоритма. Каждый основной класс требует дообучения для классов-признаков, поэтому в работе приводится один.

Модель принимает на вход изображение и возвращает набор двумерных ограничивающих рамок $(x_{min}, y_{min}, x_{max}, y_{max})$ для каждого обнаруженного объекта, а также соответствующий ему класс («car» — «Автомобиль») и оценку уверенности.

Помимо детекции основного объекта, система может использовать детекции его характерных частей или признаков («car_front», «car_rear» — «передняя часть автомобиля», «задняя часть автомобиля»). Эти вспомогательные детекции получаются либо той же нейросетью (если она обучена на множественные классы), либо отдельной специализированной моделью. Наличие таких признаков позволяет определить якори для последующей 3D-оценки.

После получения двумерных детекций следующим шагом является определение начального трехмерного положения для каждого объекта.

Для каждого основного детектированного объекта (например, «Автомобиль») система ищет наиболее подходящий характерный признак (например, «передняя часть автомобиля» или «задняя часть автомобиля») из числа обнаруженных на предыдущем этапе. Сопоставление производится на основе правила, учитывающего геометрическую близость и взаимное расположение рамок основного объекта и его признаков.

В результате сопоставления для основного объекта выбирается одна из точек на его 2D-рамке (или рамке признака), которая будет служить 2D-якорем (опорной точкой) $p_{2D} = (u, v)$. Эта точка p_{2D} соответствует определенной, заранее известной локальной точке P_{local} на канонической 3D-модели объекта (например, центр нижней грани задней части автомобиля). Выбор конкретной P_{local} зависит от того, какой признак был сопоставлен и какая часть этого признака была выбрана в качестве p_{2D} .

2.3. Модуль определения 3D координат опорной точки

Зная 2D-координаты опорной точки $p_{2D} = (u, v)$ на изображении и калибровочную матрицу камеры P, можно построить луч в трехмерном пространстве, исходящий из оптического центра камеры и проходящий через эту точку. Направление этого луча d_{ray} в системе координат камеры можно получить из уравнения (1)

$$\lambda \begin{bmatrix} u & v & 1 \end{bmatrix} = P \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \quad (1)$$

Разрешая это уравнение относительно $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ при условии, что точка лежит на луче, и нормализуя, получаем вектор направления. Точнее, если K — матрица внутренних параметров камеры ($\mathrm{P} = \mathrm{K}[\mathrm{R}|\mathrm{t}]$, где для простоты можно считать, что камера находится в начале координат и ее оптическая ось совпадает с осью Z , т.е. $\mathrm{R} = \mathrm{I}$, $\mathrm{t} = 0$ для получения луча в системе координат камеры), то направление луча можно найти как:

$$d_{ray} = K^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix} \quad (2)$$

Затем этот луч нормализуется: $d_{norm} = d_{ray}/||d_{ray}||$.

Оптический центр камеры C_{org} (в ее собственной системе координат) обычно принимается за $(0,0,0)^T$.

Далее, предполагается, что выбранная опорная точка объекта P_{local} (например, точка на уровне колес) лежит на известной опорной плоскости (например, дорожном полотне). Эта плоскость задается своей нормалью n_{plane} и точкой P_{plane} на плоскости (например, точка под камерой на высоте дороги). Трехмерные координаты $P_{anchor3D}$ искомой опорной точки объекта находятся как точка пересечения луча (C_{org}, d_{ray_norm}) с этой плоскостью

$$P_{anchor3D} = C_{org} + t \cdot d_{norm} \quad (3)$$

где параметр t находится из уравнения плоскости (3)

$$t = \frac{(P_{plane} - C_{org}) \cdot n_{plane}}{d_{ray\_norm} \cdot n_{plane}} \quad (4)$$

При этом проверяется, что пересечение происходит перед камерой ($\mathrm{t} > 0$) и знаменатель не слишком близок к нулю (луч не параллелен плоскости).

2.4. Модуль оптимизации ориентации объекта

После получения начальной оценки 3D-положения опорной точки $P_{anchor3D}$ и зная соответствующую ей локальную точку P_{local} на 3D-модели, следующим шагом является определение ориентации объекта, в частности, угла поворота R_y вокруг вертикальной оси.

Оптимизация ориентации формулируется как задача минимизации функции ошибки (невязки) $E(R_y)$, которая характеризует степень несоответствия между проекцией 3D-модели объекта (повернутой на угол R_y) на изображение и наблюдаемыми 2D-признаками (например, границами исходной 2D-рамки объекта).

$$R_y^* = \arg\min_{R_y} E(R_y) \quad (5)$$

В качестве функции ошибки может использоваться сумма квадратов разностей между горизонтальными координатами (x -координатами) краев исходной 2D-рамки объекта $(x_{left2D}, x_{right2D})$ и проекциями

соответствующих пар вершин 3D-модели объекта, которые должны соответствовать этим краям при оптимальном R_y :

$$E(R_y) = (x'_{proj\_v_1}(R_y) - x_{target2D_1})^2 + (x'_{proj\_v_2}(R_y) - x_{target2D_2})^2 \quad (6)$$

где $x_{target2D_1}$ и $x_{target2D_2}$ — это целевые x -координаты на изображении (например, левая и правая границы исходной 2D рамки объекта, или x -координаты проекций диагональных точек, которые должны соответствовать определенным частям 2D рамки).

Для вычисления функции ошибки на каждой итерации оптимизации выполняются следующие шаги:

Построение канонической 3D-модели: Используются априорные средние размеры объекта (длина L_0 , ширина W_0 , высота H_0) для определения координат его 8 вершин $V_i{}_{i=1}^{8}$ в локальной системе координат объекта, где якорь лежит на оптимизируемой диагонали.

Локальный поворот модели: Вершины локальной модели V_i центрируются относительно локальной опорной точки P_{local} (которая использовалась на этапе 2.3), поворачиваются вокруг вертикальной оси на текущий угол R_y с помощью матрицы поворота $M_y(R_y)$, а затем смещаются обратно:

$$V'_i = M_y(R_y)(V_i - P_{local}) + P_{local} \quad (7)$$

Глобальное размещение модели: Повернутые локальные вершины V'_i переносятся в глобальную систему координат камеры. Это достигается путем такого смещения, чтобы локальная опорная точка P_{local} (после поворота, если она не центр вращения) совпала с ранее оцененной глобальной 3D-опорной точкой $P_{anchor3D}$. Если P_{local} была центром вращения, то повернутые вершины V'_i смещаются так, чтобы их новый центр (бывший P_{local}) оказался в $P_{anchor3D}$. Более точно, если $V_{centered_i} = V_i - P_{local}$, то $V_{rotated\_centered_i} = M_y(R_y)V_{centered_i}$, и $V_{world_i} = V_{rotated\_centered_i} + P_{anchor3D}$

Проекция на изображение. Мировые координаты вершин V_{world_i} проецируются на плоскость изображения с использованием полной калибровочной матрицы P :

$$p'_{proj_i} = (u'_i, v'_i) = \mathrm{Project}(P, V_{world_i}) \quad (8)$$

Из этих проекций p'_{proj_i} выбираются необходимые координаты (например, u'_i) для вычисления функции ошибки.

Для минимизации функции ошибки $E(R_y)$ используется итерационный численный метод. В данной работе рассматривается применение метода Гаусса-Ньютона (или аналогичного квазиньютоновского метода), который хорошо подходит для задач минимизации суммы квадратов невязок.

Итерационный процесс продолжается до выполнения одного из критериев остановки. Достигнуто максимальное заданное число итераций. Изменение значения функции ошибки $E(R_y)$ или параметра R_y между последовательными итерациями становится меньше заданного малого порога ϵ . После выполнения оптимизации на выходе имеем сориентированные в пространстве канонические 3D модели объектов на виртуальной сцене.

Экспериментальная оценка

Для оценки эффективности были применены две конфигурации тестов. Первая конфигурация включала в себя оценку по неизменной плоскости дороги. Вторая предполагала, что есть точное значение глубины у пикселя якоря. Использовался стандартный оценочный скрипт на C++, поставляемый официально вместе с датасетом. Он также строит графики Precision-Recall для каждой оцениваемой характеристики.

3.1. Набор данных

Все экспериментальные исследования проводятся на широко используемом в задачах автономного вождения наборе данных KITTI, в частности, на его подвыборке для 3D-детекции объектов. Он содержит синхронизированные данные с различных сенсоров, включая монокулярные камеры, лидар, а также точную разметку трехмерных ограничивающих рамок для объектов на изображениях из обучающей выборки (training set). Для работы метода использовались изображения из левой цветной камеры и соответствующие файлы калибровки, содержащие матрицу проекции P_2 . Поскольку нейросеть была дообучена для определения положения объектов класса «автомобиль», именно он и был выбран для проверки эффективности метода. Для оценки использовался полный набор, содержащий 7480 изображений.

3.2. Метрики оценки качества

Для количественной оценки производительности предлагаемой системы используются следующие метрики, стандартные для задачи 3D-детекции объектов на датасете KITTI:

Точность 3D-локализации:

1. Сравнение проекций на плоскость (BEV IoU / тени на землю).

Двумерная оценка перекрытия реализуется посредством проекции 3D-боксов на горизонтальную плоскость — это планарная аппроксимация объекта. Оценивается IoU в Bird’s Eye View.

2. Точность оценки ориентации.

Метрика Average Orientation Similarity (AOS) из протокола KITTI, которая учитывает как точность 2D-детекции, так и ошибку ориентации для R_y (AOS применяется только к тем объектам, для которых выполнено условие перекрытия (IoU) с истинным объектом в 2D-пространстве).

3. Средняя точность.

Average Precision (AP): Средняя точность, вычисленная для различных порогов 3D IoU (стандартная — 0.7 для KITTI) и для разных категорий сложности объектов (Easy, Moderate, Hard), как определено в официальном протоколе оценки KITTI.

Кроме качества будем оценивать скорость обработки данных, а также сравним полученные показатели с другими монокулярными методами.

Первый график — оценка ошибки для трехмерной ориентации бокса по оси y (рис. 2).

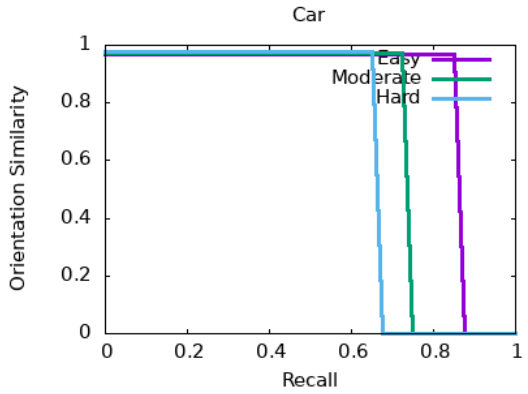

Рисунок 2 - График оценки mAp для 3D-ориентации

DOI: https://doi.org/10.60797/COMP.2025.7.2.2

Как видно из графика, оценка достаточно точно отражает реальный поворот автомобиля по соответствующей оси. Итоговая точность на классе «car» составляет:

Easy: mAP AOS ≈ 82,49.

Moderate: mAP AOS ≈ 71,27.

Hard: mAP AOS ≈ 64,38.

Для сравнения в таблице 1 представлены другие методы и их показатели.

Таблица 1 - Сравнение методов по точности 3D-ориентации

DOI: https://doi.org/10.60797/COMP.2025.7.2.3

Метод	Easy AOS, %	Moderate AOS, %	Hard AOS, %
Deep3DBox [4]	92,90	88,75	76,76
SMOKE [5]	92,94	87,02	77,12
RTM3D [6]	91,75	86,73	77,18
Mono3D [7]	91,01	86,62	76,84
Разработанный метод	82,49	71,27	64,38

Представленные результаты говорят о несильном отставании от передовых методов, при этом скорость обработки выше в разы, что можно видеть в таблице 2.

Таблица 2 - Сравнение производительности методов

DOI: https://doi.org/10.60797/COMP.2025.7.2.4

Метод	Скорость обработки данных, кадры в секунду
Mono3D [7]	1
Deep3DBox [4]	2–3
RTM3D [6]	5
SMOKE [5]	8-12
Разработанный метод	25–33

Несмотря на высокую производительность и точность 3D ориентации, без хорошей аппроксимации глубины в точках установки якоря он не демонстрирует большую точность 3D позиционирования (рис. 3).

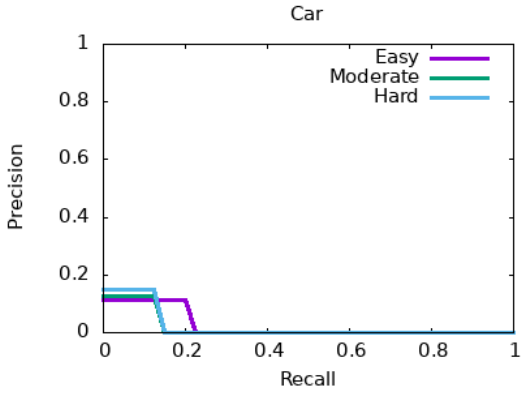

Рисунок 3 - Показатели точности совпадения силуэтов для аппроксимации глубины плоскостью
DOI: https://doi.org/10.60797/COMP.2025.7.2.5

Однако при сопоставлении построении глубины сторонними методами, точность значительно возрастает (рис. 4).

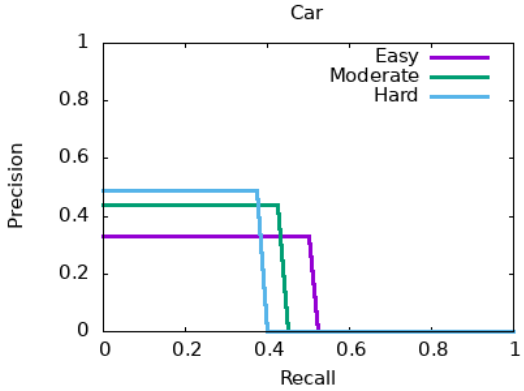

Рисунок 4 - Точность аппроксимации теней 3D-боксов при уточнении глубины
DOI: https://doi.org/10.60797/COMP.2025.7.2.6

Данные показывают, что точность возрастает до значений:
Easy: mAP BEV ≈ 17,02.
Moderate: mAP BEV ≈ 19,27.
Hard: mAP BEV ≈ 19,09.
Сравнивая с другими методами, метод демонстрирует наиболее высокую точность на сложных участках, неплохую на средних и наиболее низкую на легких. Связан такой результат с тем, что методу необходимо видеть объект в кадре полностью, чтобы его аппроксимировать по одному изображению.

Таблица 3 - Сравнение показателей точности для теней 3D-боксов

DOI: https://doi.org/10.60797/COMP.2025.7.2.7

Метод	Easy mAP BEV, %	Moderate mAP BEV, %	Hard mAP BEV, %
SMOKE [5]	21,08	15,13	12,91
OPA-3D [8]	17,05	24,60	14,25
DID-M3D [9]	16,29	24,40	13,75
Разработанный метод	17,02	19,27	19,09

Таким образом, показатели становятся сравнимыми с другими методами, при том, что точность более равномерно распределена между сложными и простыми случаями. Результат работы - рис. 5.

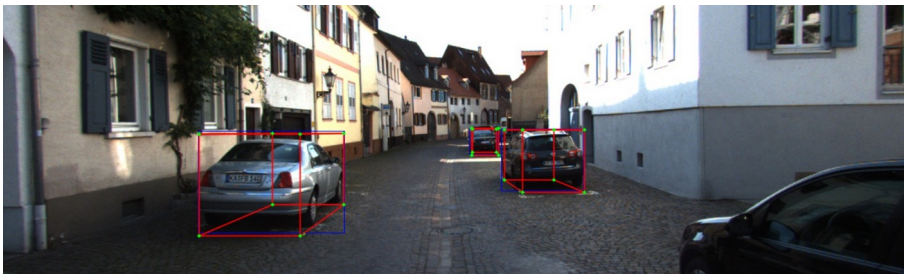

Рисунок 5 - Пример работы алгоритма
DOI: https://doi.org/10.60797/COMP.2025.7.2.8

Обсуждение

В представленном методе впервые показано, что предварительная сегментация дороги и последующая быстрая аппроксимация её плоскости позволяют получить конкурентоспособную точность 3D-оценки и ориентации без явного восстановления полной глубины сцены. Основные выводы и преимущества метода:

4.1. Сбалансированное соотношение «точность — скорость»

Средний BEV mAP на уровнях Easy/Moderate/Hard составляет 17,02% / 19,27% / 19,09%, что близко к результатам передовых монокулярных методов (SMOKE, OPA-3D, DID-M3D), особенно в сложных сценах (Hard).

Ориентационная метрика (orientation AOS) на уровне Easy ≈ 82,5%, Moderate ≈ 66,5%, Hard ≈ 59,6% — лишь слегка уступает SMOKE и другим методам, при этом скорость обработки достигает 25–33 FPS на GTX 1650 Mobile, что в 1,5–2,5× быстрее SMOKE и в 4–6× быстрее RTM3D.

4.2. Встроенная сегментация дороги

Пайплайн на основе нейросетей архитектуры YOLO (ее производных seg-сетей) позволяет выполнять не только детекцию объектов, но и сегментацию дороги, что позволяет избежать отдельного модуля сегментации и снизить вычислительную нагрузку.

Сегментационные данные можно использовать для обнаружения доступной для проезда части поверхности (отделяя пешеходную часть, бордюры, перепады высоты, глубокие ямы), и занося ее в смоделированную сцену.

Наличие сегментной маски может быть использовано для дальнейшего уточнения геометрии, например, RANSAC-аппроксимации плоскости дороги [10] (требуются дальнейшие исследования)

4.3. Ограничения приближённой плоскости

Простая аппроксимация плоскостью может быть полезна в контролируемой среде, например на складе или заводе. В этом случае методы аппроксимации поверхности, по которой едет объект не нужно и это упрощает вычисления. Для применения в реальности необходимо построение разреженной или плотной карты глубины. При этом оно может происходить параллельно, и не в реальном времени, а в режиме «непрерывной дорисовки». Также построение карты глубины можно совместить с локализацией в подобие SLAM-метода [11].

4.4. Фундаментальные ограничения и их обход

Метод имеет фундаментальное ограничение, из-за которого он должен видеть признаки объекта в кадре, чтобы строить модель габаритов. Также, чтобы строить верную модель габаритов, в текущей реализации еще нужно, чтобы сам объект бы полностью видим на изображении. Если алгоритм работает с системой всенаправленных камер, они незначительны. Но даже если существуют слепые пятна, можно улучшить реализацию и добиться хорошей точности даже на не полностью видимых объектах. Первое ограничение нивелируется увеличением количества признаков так, чтобы любая небольшая часть объекта обладала своим. Второе ограничение (даже без наращивания числа признаков) можно обойти, аппроксимируя угол поворота не по двум границам 2D бокса, а по одной, полностью видимой. Видимость можно определять по касанию 2D рамок бокса границ изображения. Также можно, например, помечать неожиданно появляющиеся на виртуальной сцене объекты как «ненадежные», но по мере их существования в зоне видимости увеличивать коэффициент надежности (так как работа ведется с видеопотоком). Это позволит трассирующему маршрут методу уделять особое внимание областям с «ненадежными» объектами.

4.5. Возможные направления улучшения

Учёт углов камеры: интеграция IMU данных или онлайн-оценка кренов/тангажа для компенсации ориентации камеры перед проекцией.

Дополнительные классы габаритов объектов: Для улучшения точности 3D-аппроксимации можно обучить нейросеть таким образом, чтобы разбивать автомобили по классу габаритов, и сопоставлять им соответствующие 3D-боксы. Сейчас всем автомобилям соответствует один стандартный бокс, что не мешает методу верно аппроксимировать поворот, но не габариты теней боксов.

Лёгкая «глубокая коррекция»: добавление мини-глубинной ветви (несколько сверточных слоев), обученной на исправление ошибок плоской аппроксимации, без значительного замедления.

Заключение

Предложенный метод показывает, что простая и быстрая аппроксимация дорожной плоскости на основе единой маски позволяет достигать сопоставимой с современными монокулярными подходами точности 3D-оценки и ориентации при значительном выигрыше по скорости. Его преимущества:

Ключевое преимущество — производительность: 20–33 FPS на мобильной GTX 1650 при сохранении mAP BEV и AOS на конкурентном уровне.

Интеграция сегментации и детекции: отсутствие отдельного модуля сегментации дороги упрощает пайплайн и снижает накладные расходы.

Простота доработок: метод легко дополняется локальной коррекцией плоскости и учётом углов камеры для повышения точности.

В целом, комбинирование эффективного детектора YOLOv11s и упрощённой геометрической модели сцены открывает перспективы применения в системах реального времени и ограниченных устройствах. Дальнейшая работа будет ориентирована на гибридные решения, объединяющие скорость аппроксимации и точность локальных коррекций.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Terzakis G. A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem. / G. Terzakis, M. Lourakis // Lecture Notes in Computer Science. — 2020. — Vol. 12357. — P. 478–494.

2. Zhang Y. Objects are different: Flexible monocular 3d object detection. / Y. Zhang, Y. Lu, J. Zhou // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2021. — № 1. — P. 3289–3298.

3. Ярощук П.О. Метод анализа трехмерного положения объектов на двумерном изображении / П.О. Ярощук // Актуальные проблемы прикладной математики, информатики и механики. — Воронеж: Воронежский государственный университет, 2024. — С. 812–817.

4. Mousavian M. 3D Bounding Box Estimation Using Deep Learning and Geometry. / M. Mousavian, D. Anguelov, J. Flynn et al. // Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). — 2017. — CFP17003-POD. — P. 7074–7082.

5. Liu Z. SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation. / Z. Liu, W. Huang, Y. Yang // CVPR. — 2020. — № 2. — P. 8456–8465.

6. Li P. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving / P. Li // European Conference on Computer Vision. — Cham: Springer International Publishing, 2020. — DOI: 10.1007/978-3-030-58580-8_38

7. Chen X. Monocular 3D Object Detection for Autonomous Driving / X. Chen, K. Kundu, Z. Zhang et al. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — Las-Vegas: IEEE, 2016. — P. 2147–2156.

8. Su Y. Opa-3d: Occlusion-aware pixel-wise aggregation for monocular 3d object detection. / Y. Su // IEEE Robotics and Automation Letters. — 2023. — Vol. 8. — № 3. — P. 1327–1334.

9. Peng L. Did-m3d: Decoupling instance depth for monocular 3d object detection / L. Peng // European Conference on Computer Vision. — Cham: Springer Nature Switzerland, 2022. — P. 71–88.

10. Martínez-Otzeta J.M. Ransac for robotic applications: A survey. / J.M. Martínez-Otzeta, N. Ahuja // Sensors. — 2022. — Vol. 23. — № 1. — P. 327.

11. Mur-Artal R. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. / R. Mur-Artal, J.M.M. Montiel, J.D. Tardós // IEEE Transactions on Robotics. — 2015. — Vol. 31. — № 5. — P. 1147–1163.

Список литературы на английском языке / References in English

1. Terzakis G. A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem. / G. Terzakis, M. Lourakis // Lecture Notes in Computer Science. — 2020. — Vol. 12357. — P. 478–494.

2. Zhang Y. Objects are different: Flexible monocular 3d object detection. / Y. Zhang, Y. Lu, J. Zhou // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2021. — № 1. — P. 3289–3298.

3. Yaroshchuk P.O. Metod analiza trekhmernogo polozheniya obektov na dvumernom izobrazhenii [Method for analyzing the three-dimensional position of objects using a two-dimensional image] / P.O. Yaroshchuk // International Conference "Applied Mathematics, Computational Science and Mechanics: Current Problems". — Voronezh: Voronezh State University, 2024. — P. 812–817. [in Russian]

4. Mousavian M. 3D Bounding Box Estimation Using Deep Learning and Geometry. / M. Mousavian, D. Anguelov, J. Flynn et al. // Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). — 2017. — CFP17003-POD. — P. 7074–7082.

5. Liu Z. SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation. / Z. Liu, W. Huang, Y. Yang // CVPR. — 2020. — № 2. — P. 8456–8465.

6. Li P. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving / P. Li // European Conference on Computer Vision. — Cham: Springer International Publishing, 2020. — DOI: 10.1007/978-3-030-58580-8_38

7. Chen X. Monocular 3D Object Detection for Autonomous Driving / X. Chen, K. Kundu, Z. Zhang et al. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — Las-Vegas: IEEE, 2016. — P. 2147–2156.

8. Su Y. Opa-3d: Occlusion-aware pixel-wise aggregation for monocular 3d object detection. / Y. Su // IEEE Robotics and Automation Letters. — 2023. — Vol. 8. — № 3. — P. 1327–1334.

9. Peng L. Did-m3d: Decoupling instance depth for monocular 3d object detection / L. Peng // European Conference on Computer Vision. — Cham: Springer Nature Switzerland, 2022. — P. 71–88.

10. Martínez-Otzeta J.M. Ransac for robotic applications: A survey. / J.M. Martínez-Otzeta, N. Ahuja // Sensors. — 2022. — Vol. 23. — № 1. — P. 327.

11. Mur-Artal R. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. / R. Mur-Artal, J.M.M. Montiel, J.D. Tardós // IEEE Transactions on Robotics. — 2015. — Vol. 31. — № 5. — P. 1147–1163.