

КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ АЛЬ-ФАРАБИ

ЛЕКЦИЯ № 13

Применение OpenCV для распознавания дорожной разметки и автоматического удержания полосы движения электромобиля

Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент · Кафедра «Искусственный интеллект и big data»
«Технологии в интеллектуальных системах управления в IoT» · 4 курс, бакалавриат

1 Цель и задачи лекции

Цель — сформировать системное представление о построении интеллектуальной системы удержания полосы движения (Lane Keeping Assistance System, LKAS) с использованием компьютерного зрения OpenCV, оценивания положения автомобиля и алгоритмов управления.

Задачи: изучить архитектуру системы; освоить обработку видеокадров и извлечение границ полосы; вывести геометрические показатели отклонения; сопоставить ПИД- и интеллектуальные регуляторы; рассмотреть испытания, ограничения и требования безопасности.

2 План лекции

1. Назначение LKAS и место в архитектуре автопилота
2. Камера, калибровка и геометрия изображения
3. Конвейер OpenCV: фильтрация, сегментация и поиск разметки
4. Вычисление отклонения и кривизны полосы
5. Математическая модель бокового движения
6. ПИД-, Stanley- и интеллектуальные методы управления
7. Программная реализация прототипа на Python
8. Оценка качества, отказоустойчивость и безопасность

1 Интеллектуальное удержание полосы как задача управления

Удержание полосы представляет собой замкнутую систему, которая по изображению дороги оценивает геометрию допустимого коридора движения и формирует управляющее воздействие на рулевой механизм. В отличие от предупреждения о сходе с полосы (LDW), система LKAS может выдавать корректирующий рулевой момент или команду рулевому приводу. Наличие камеры и вычислительного модуля само по себе не делает систему автономной: необходимы оценки состояния, диагностика достоверности и безопасное исполнение команд.

Типовой информационный поток: камера → коррекция изображения → выделение дорожной разметки → оценка центра полосы и направления дороги → расчёт ошибки → регулятор → рулевой привод → автомобиль → новая видеокадровая обратная связь. Датчики скорости, инерциальный измерительный модуль и одометрия дополняют видеоданные.

В учебном прототипе алгоритм выполняется на записанном видео или в симуляции. Перенос на реальный автомобиль требует независимых средств безопасности, проверки исполнительных механизмов и соблюдения нормативных требований.

2 Камера, калибровка и геометрические преобразования

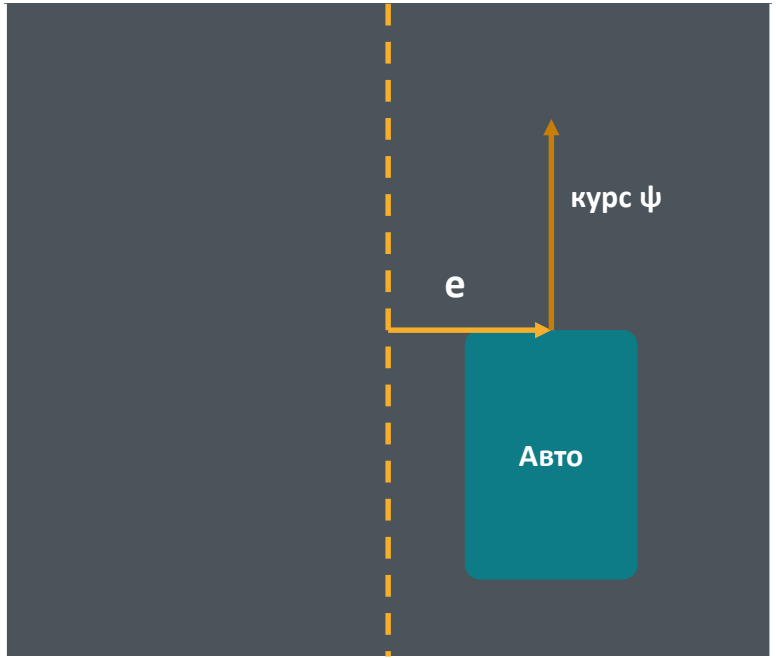
Изображение, полученное камерой, подвержено перспективным искажениям, радиальной и тангенциальной дисторсии. Для измерения положения дорожных линий необходимо откалибровать камеру по изображениям калибровочной мишени и получить матрицу внутренних параметров K, коэффициенты искажений и преобразование к дорожной плоскости.

Для идеализированной модели камеры точка пространства (X,Y,Z) проецируется в изображение по соотношению $s[u,v,1]^T = K[R|t][X,Y,Z,1]^T$, где K — матрица внутренних параметров, R и t — ориентация и положение камеры, s — масштаб.

Преобразование перспективы (bird's-eye view) выполняется гомографией $s[u',v',1]^T = H[u,v,1]^T$. Оно корректно для приближённо плоской дорожной поверхности; при подъёмах, уклонах и неровностях возникают систематические ошибки.

Выбор области интереса (ROI) сокращает ложные срабатывания на небе, зданиях и соседних объектах, но не должен отсекать дальнюю часть дороги, важную для предсказания кривизны.

4 Геометрия отклонения от полосы



e = x(авто) − x(центр)
ψ — угол курса относительно полосы
регулятор формирует рулевое воздействие по e и ψ

Рис. 24. Отклонение от центра полосы и ошибка курса — входы алгоритма удержания.

3 Обработка видеокадров средствами OpenCV

3.1. Подготовка изображения

На первом этапе кадр переводят в подходящее цветовое пространство (HSV или HLS), подавляют шум медианным или гауссовым фильтром и корректируют контрастность. Белую и жёлтую разметку можно выделять порогами цвета, однако эти пороги чувствительны к освещению, дождю и теням. Поэтому целесообразно комбинировать цветовые и градиентные признаки.

Гауссова фильтрация описывается свёрткой $I\sigma(x,y) = (G\sigma * I)(x,y)$, где $G\sigma(x,y) = [1/(2\pi\sigma^2)] \exp[-(x^2+y^2)/(2\sigma^2)]$. Рост σ уменьшает шум, но может сгладить тонкие линии разметки.

3.2. Поиск границ и линий

Детектор Canny оценивает градиент яркости, подавляет немаксимумы и использует два порога с трассировкой связных границ. После Canny применяют вероятностное преобразование Хафа (HoughLinesP), которое возвращает конечные отрезки. Наклон отрезка $m = (y_2-y_1)/(x_2-x_1)$ позволяет грубо разделить левую и правую границы в перспективном изображении.

Обработка видеокадров средствами OpenCV (продолжение)

Для извилистых дорог предпочтительнее аппроксимация полиномом $x(y) = ay^2 + by + c$ или сегментация пикселей разметки нейронной сетью. Прямолинейная модель Хафа эффективна для простых сцен, но не описывает кривизну на поворотах.

3.3. Отслеживание разметки между кадрами

Обнаруженные границы целесообразно сопровождать во времени: ограничивать изменение коэффициентов полинома, использовать фильтр Калмана и проверять правдоподобие ширины полосы. Система должна отдельно обозначать случаи, когда одна или обе границы не обнаружены, вместо подстановки недостоверной траектории.

5 Конвейер обнаружения разметки



Рис. 25. Классический конвейер LKAS на OpenCV.

4 Вычисление геометрических ошибок

Пусть xL(y0) и xR(y0) — координаты левой и правой границы на выбранной строке y0. Тогда центр полосы xC = [xL(y0)+xR(y0)]/2. При xV — координате проекции продольной оси автомобиля боковая ошибка в пикселях epx = xC−xV. При известном масштабе sx [м/пикс] получаем ey = sx·epx.

Угловая ошибка eψ — разность между направлением касательной к центральной линии и направлением автомобиля. Для траектории x(y) касательная определяется производной dx/dy; перевод в физические координаты требует калибровки и выбранной системы осей.

Кривизна плоской траектории y(x) определяется к = y″/(1+y′²)^(3/2), а радиус R = 1/|к|. В полиномиальной модели y(x)=Ax²+Bx+C: R(x) = [1+(2Ax+B)²]^(3/2)/|2A|. Нулевой знаменатель соответствует прямой линии с бесконечным радиусом.

Пример (условные данные): ширина кадра 1280 пикселей; центр автомобиля xV=640; границы на контрольной строке xL=410 и xR=850. Тогда xC=630, epx=−10 пикселей. При sx=0,005 м/пикс боковое отклонение ey=−0,05 м. Знак ошибки зависит от принятого направления оси и должен быть согласован со знаком рулевой команды.

5 Математическая модель движения автомобиля

Для анализа бокового управления при умеренных скоростях и малых углах можно использовать кинематическую велосипедную модель: $\dot{x} = v \cos \psi$, $\dot{y} = v \sin \psi$, $\dot{\psi} = (v/L)\tan \delta$, где (x,y) — положение, ψ — курс, v — скорость, L — колёсная база, δ — угол поворота эквивалентного переднего колеса.

В координатах относительно опорной траектории упрощённая динамика ошибок имеет вид $\dot{e}_y \approx v e_\psi$, $\dot{e}_\psi \approx (v/L)\tan \delta - v\kappa_{ref}$. Это приближение предполагает малые ошибки, малое боковое скольжение и достаточно точную оценку кривизны κ_{ref}.

При проектировании необходимо учитывать насыщение рулевого привода $|\delta| \le \delta_{max}$, ограничение скорости изменения $|\dot{\delta}| \le \omega_{max}$, задержки камеры и вычислителя, а также потерю сцепления с дорогой. В высокоскоростных режимах кинематической модели недостаточно: требуется динамическая модель с боковыми силами шин.

6 Алгоритмы формирования управляющего воздействия

6.1. ПИД-регулятор

Базовый закон: δcmd(t) = Kp ey(t) + Ki∫ey(t)dt + Kd dey(t)/dt. Коэффициенты зависят от знаков осей, скорости, масштаба ошибки и кинематики рулевого привода. Для цифровой реализации интеграл и производную вычисляют дискретно; необходимы фильтрация производной и anti-windup при насыщении.

ПИД-регулятор прост и объясним, но при меняющейся кривизне дороги и скорости требует перенастройки. Ошибка только по положению недостаточна: для устойчивого удержания полезно учитывать также eψ и kref.

6.2. Геометрический регулятор Stanley

Один из известных подходов: δcmd = eψ + arctan[k ey/(v+v0)], где k>0 — коэффициент усиления, v0>0 предотвращает деление на нуль. При реализации знак eψ и ey выбирают согласно единой системе координат; команда ограничивается допустимым диапазоном. Регулятор чувствителен к качеству локализации относительно полосы.

Алгоритмы формирования управляющего воздействия (продолжение)

6.3. Нейросетевые и гибридные подходы

Нейронная сеть может выдавать маску дорожной разметки, оценку центральной линии, угол поворота или прогноз траектории. Прямое отображение изображения в рулевую команду требует значительно более строгой проверки, чем схема «нейросетевая сегментация + интерпретируемый регулятор». Гибридная архитектура позволяет использовать нейросеть для восприятия, а классический контроллер — для управления.

Для обучения сегментации применяют, например, функцию потерь $L = \lambda 1\, L_{CE} + \lambda 2\, L_{Dice}$, где L_{CE} — кросс-энтропия по пикселям, L_{Dice} — мера несоответствия перекрытия масок. Необходимо учитывать дисбаланс классов, когда дорожная разметка занимает малую часть кадра.

6 Алгоритмы формирования управляющего воздействия (продолжение)

Подход	Преимущества	Ограничения
ПИД	Низкая вычислительная сложность; понятная настройка	Чувствителен к скорости и качеству ошибок
Stanley	Учитывает боковую и угловую ошибки	Требует согласованных координат и настройки
Нейросетевая сегментация + регулятор	Устойчивее к сложным визуальным признакам при хорошем обучении	Зависит от датасета и вычислительных ресурсов
Модельно-прогнозирующее управление (MPC)	Учитывает ограничения и горизонт прогноза	Требует модели и оптимизации в реальном времени

7 Учебный программный прототип OpenCV

Следующий код демонстрирует обнаружение прямолинейных сегментов разметки на видеокадрах. Это не готовый автомобильный контроллер: он не формирует команды реальному рулевому приводу и не заменяет калибровку камеры, верификацию и контроль безопасности.

```
import cv2
import numpy as np
cap = cv2.VideoCapture("road_video.mp4")
while cap.isOpened():
    ok, frame = cap.read()
    if not ok:
        break
    h, w = frame.shape[:2]
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    smooth = cv2.GaussianBlur(gray, (5, 5), 1.2)
    edges = cv2.Canny(smooth, 60, 160)
    mask = np.zeros_like(edges)
    polygon = np.array([[(int(.1*w), h),
```

7 Учебный программный прототип OpenCV (продолжение)

```
                        (int(.43*w), int(.58*h)),
                        (int(.57*w), int(.58*h)),
                        (int(.9*w), h)]], dtype=np.int32)
    cv2.fillPoly(mask, polygon, 255)
    roi = cv2.bitwise_and(edges, mask)
    lines = cv2.HoughLinesP(roi, 1, np.pi/180, 40,
                            minLineLength=35, maxLineGap=70)
    if lines is not None:
        for segment in lines[:, 0]:
            x1, y1, x2, y2 = map(int, segment)
            cv2.line(frame, (x1, y1), (x2, y2),
                     (0, 255, 0), 2)
    cv2.imshow("Lane segments (demo)", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

7 Учебный программный прототип OpenCV (продолжение)

Пороговые значения в примере демонстрационные. Для конкретного набора видеозаписей их подбирают на валидационной выборке, не используя тестовые кадры для настройки. Для полноценной оценки положения полосы необходимы группировка отрезков, восстановление левой и правой границы, оценка достоверности и преобразование пикселей в метры.

8 Экспериментальная оценка и безопасность

Исследование проводят по сценариям: дневное освещение, сумерки, дождь, тени, частично стёртая разметка, криволинейные участки, перекрытие разметки транспортными средствами. Видеоданные разделяют по поездкам и дорожным сценам, чтобы почти одинаковые кадры не попадали одновременно в обучение и тест.

Качество восприятия оценивают по IoU и F1 для маски разметки, MAE/RMSE боковой ошибки, доле кадров с корректным обнаружением обеих границ. Качество управления — по среднеквадратичному боковому отклонению, максимальному отклонению, частоте пересечений границ, плавности руля и времени реакции.

RMSE = $\mathrm{sqrt}[(1/N)\,\Sigma_i(\hat{e}y,i - ey,i)^2]$. Задержка контура $T_{total} = T_{capture} + T_{process} + T_{control} + T_{actuator}$. При скорости 15 м/с задержка 0,20 с соответствует прохождению 3 м по траектории, поэтому важно прогнозирование состояния, а не только повышение частоты кадров.

При потере достоверной разметки прототип должен сигнализировать о недоступности функции и переходить к предусмотренному безопасному состоянию. В реальных автомобилях необходимы независимый контроль состояния водителя, резервирование критических функций, диагностика и системная оценка рисков. Испытания с автоматическим рулевым воздействием допустимы только на специализированных закрытых площадках с надлежащими средствами защиты.

9 Методика учебного исследования

1. Подготовить набор видеозаписей минимум из трёх разных дорожных условий и вручную разметить контрольные кадры.
2. Реализовать классический конвейер OpenCV (цветовые пороги, Canny, Hough).
3. Построить полиномиальную модель границ и вычислить боковую ошибку в пикселях и метрах.
4. Сравнить качество детектирования с сегментационной нейросетью на одинаковом тестовом наборе.
5. В симуляции сравнить ПИД- и Stanley-регуляторы при нескольких скоростях и кривизнах.
6. Измерить распределение времени обработки кадров, RMSE боковой ошибки и долю отказов.
7. Оформить результаты в таблицах и графиках; объяснить ошибки, ограничения и воспроизводимость.

12 Заключение

Компьютерное зрение позволяет получать геометрические признаки дорожной сцены для управления автомобилем, однако надёжное удержание полосы требует совместной работы алгоритмов восприятия, оценивания состояния, управления и диагностики. OpenCV удобен для объяснимого прототипирования и создания базового метода сравнения. Нейронные сети расширяют возможности восприятия в сложных условиях, но требуют репрезентативных данных и строгой проверки. Практическая эффективность определяется не только точностью распознавания, но и задержкой, устойчивостью замкнутого контура и безопасностью.

13

Контрольные вопросы

1. Каково функциональное отличие LDW от LKAS?
2. Из каких блоков состоит замкнутый контур удержания полосы?
3. Зачем требуется калибровка камеры?
4. В каких условиях корректно перспективное преобразование?
5. Какие преимущества даёт цветовое пространство HLS/HSV?
6. Как работает детектор Canny?
7. Каковы ограничения преобразования Хафа на криволинейной дороге?
8. Как вычисляют центр полосы и боковую ошибку?
9. Почему необходимо согласовывать знаки координат?
10. Как определяется кривизна опорной траектории?
11. Какие допущения использует кинематическая велосипедная модель?

13 Контрольные вопросы (продолжение)

12. В чём преимущества и недостатки ПИД-регулятора?
13. Как устроен регулятор Stanley?
14. Какие метрики характеризуют качество восприятия и управления?
15. Какие меры необходимы при потере разметки и росте задержки?

Рекомендуемая литература и ресурсы

1. **Bradski G., Kaehler A. Learning OpenCV:** Computer Vision with the OpenCV Library. O’Reilly Media.

2. **Szeliski R. Computer Vision:** Algorithms and Applications. 2nd ed. Springer, 2022.

3. Rajamani R. Vehicle Dynamics and Control. 2nd ed. Springer, 2012.

4. **Thrun S. et al. Stanley:** The Robot that Won the DARPA Grand Challenge. Journal of Field Robotics, 2006.

5. **OpenCV Documentation:** https://docs.opencv.org/

6. **Python Documentation:** https://docs.python.org/3/

7. **ISO 26262 (серия стандартов):** Road vehicles — Functional safety (для углублённого изучения требований к безопасности).