

КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ АЛЬ-ФАРАБИ

ЛЕКЦИЯ № 12

Беспилотные летательные аппараты с машинным зрением: распознавание объектов, местности и траекторий движения

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 лекции

Сформировать системное представление об архитектуре интеллектуального БПЛА, методах компьютерного зрения и машинного обучения для обнаружения объектов и анализа местности, способах сопряжения бортового мини-компьютера с полётным контроллером и критериях экспериментальной оценки алгоритмов автономного движения.

2 Задачи лекции

- Рассмотреть состав и функции бортовой интеллектуальной системы БПЛА.
- Изучить принципы обработки изображений, обнаружения и сопровождения объектов.
- Описать методы визуальной локализации, оценки движения и планирования траектории.
- Показать интеграцию Raspberry Pi или NVIDIA Jetson с автопилотом и IoT-каналом связи.
- Разработать подход к испытаниям точности, быстродействия и надёжности системы.

3 План лекции

1. Роль БПЛА и компьютерного зрения в интеллектуальном управлении
2. Аппаратно-программная архитектура и обмен данными
3. Получение изображений и предварительная обработка
4. Обнаружение и классификация объектов с помощью нейросетей
5. Визуальная одометрия, локализация и анализ местности
6. Планирование траектории и контуры управления
7. IoT-коммуникации и требования безопасности
8. Практический пример и экспериментальная верификация

1 Роль машинного зрения в интеллектуальных БПЛА

Беспилотный летательный аппарат представляет собой мехатронную систему, включающую летательную платформу, двигатели и регуляторы скорости, полётный контроллер, навигационные датчики, каналы связи и полезную нагрузку. В традиционной архитектуре автопилот стабилизирует аппарат по сигналам инерциальных и навигационных датчиков. Машинное зрение расширяет возможности системы: позволяет получать сведения о пространственной сцене, распознавать целевые объекты и оценивать изменения окружающей среды.

В задачах сельского хозяйства камера используется для мониторинга состояния посевов, оценки неоднородности растительного покрова и обнаружения проблемных участков. В промышленности визуальные алгоритмы применяются для обследования сооружений и контроля состояния оборудования. В задачах автономной навигации они помогают распознавать препятствия и ориентиры.

Важно различать распознавание объекта (определение класса), детектирование (определение класса и положения на изображении), сегментацию (выделение пикселей объекта), трекинг (сопровождение во времени) и визуальную локализацию (оценку положения камеры). Это самостоятельные задачи с различными требованиями к данным и вычислительным ресурсам.

2 Аппаратно-программная архитектура

Базовая архитектура включает камеру RGB или стереокамеру, одноплатный компьютер, полётный контроллер, IMU, GNSS-приёмник, при необходимости дальномер или LiDAR, а также телеметрический модуль. Полётный контроллер выполняет критически важные задачи стабилизации и безопасности; нейрокомпьютер осуществляет обработку изображений и формирует высокоуровневые рекомендации.

Передача команд между вычислительным модулем и автопилотом может выполняться через MAVLink по UART, USB или UDP. При проектировании необходимо разграничить высокоуровневое задание маршрута и низкоуровневые контуры управления. Потеря связи с нейрокомпьютером не должна нарушать базовую устойчивость полёта.

На Raspberry Pi удобно исследовать алгоритмы OpenCV и лёгкие нейросетевые модели. NVIDIA Jetson предназначен для более ресурсоёмкого инференса, но выбор платформы должен основываться на фактическом профилировании конкретной модели, её энергопотребления и массы аппаратуры.

5 Бортовая система БПЛА

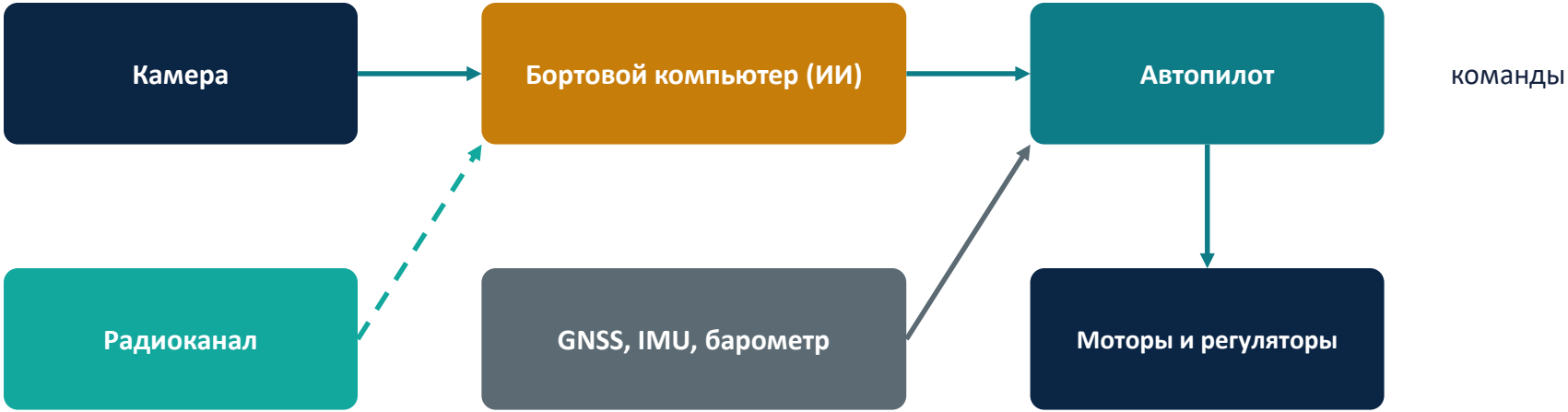


Рис. 22. Бортовой компьютер выдаёт целеуказания; стабилизацию выполняет автопилот.

3 Получение и предварительная обработка изображений

Качество результата зависит от разрешения камеры, частоты кадров, освещённости, экспозиции, вибраций и высоты полёта. Типовой конвейер включает захват кадра, калибровку камеры, коррекцию дисторсии, масштабирование, нормализацию и передачу тензора в модель. При низкой освещённости и смазывании изображений вероятность ошибки возрастает.

Камера описывается приближённой проективной моделью: $s\,[u, v, 1]^T = K\,[R \mid t]\,[X, Y, Z, 1]^T$, где K — матрица внутренних параметров, R и t — ориентация и положение камеры, (X,Y,Z) — координаты точки в пространстве, (u,v) — координаты пикселя, s — масштабный множитель.

Калибровка необходима для корректной геометрической интерпретации изображения. Если камера наклонена, прямое преобразование пиксельных координат в географические без учёта ориентации, высоты и рельефа приводит к систематическим ошибкам.

4 Обнаружение и классификация объектов

Свёрточные нейронные сети извлекают иерархические признаки из изображений. Детекторы семейства YOLO оценивают классы и ограничивающие прямоугольники объектов за один проход сети. Для бортового использования важны не только средняя точность детекции, но и время инференса, размер модели, устойчивость к изменениям освещения и энергозатраты.

Для оценки совпадения предсказанной области Bp и эталонной Bt применяется IoU = |Bp ∩ Bt| / |Bp ∪ Bt|. Предсказание считается истинно положительным при корректном классе и выполнении заданного порога IoU. Precision = TP/(TP+FP), Recall = TP/(TP+FN), F1 = 2PR/(P+R). Метрика mAP усредняет точность по классам и выбранным порогам IoU.

Обучение начинается с определения классов и правил разметки. Данные следует разделять на обучающую, валидационную и тестовую выборки по полётам, участкам или датам, а не случайно перемешивать почти одинаковые соседние кадры. Иначе возникает утечка данных и завышение оценок качества.

5 Визуальная одометрия и анализ местности

Визуальная одометрия оценивает относительное движение камеры по последовательности кадров. Классический подход использует выделение ключевых точек, сопоставление дескрипторов, исключение выбросов с помощью RANSAC и оценку геометрического преобразования. Для монокулярной камеры абсолютный масштаб движения без дополнительных данных в общем случае неоднозначен.

Визуально-инерциальная одометрия объединяет наблюдения камеры и IMU, что повышает устойчивость оценки при кратковременной потере визуальных признаков. SLAM дополнительно строит карту и использует повторное распознавание мест для уменьшения накопленной ошибки. В открытой местности возможны проблемы из-за повторяющейся текстуры посевов, движения растительности и недостатка ориентиров.

Положение БПЛА можно описать вектором состояния $x = [p_x, p_y, p_z, v_x, v_y, v_z, \phi, \theta, \psi]^T$. Оценка состояния объединяет данные GNSS, IMU, барометра и камеры. Для проверки локализации используют абсолютную ошибку траектории ATE и относительную ошибку движения RPE, сравнивая результат с независимым эталоном.

6 Планирование траектории и контуры управления

Автономное движение требует последовательного решения задач восприятия, локализации, планирования и исполнения. Глобальный планировщик определяет маршрут с учётом карты и ограничений, локальный планировщик реагирует на новые препятствия. На уровне автопилота выполняется стабилизация ориентации и отслеживание заданной траектории.

Для упрощённого описания поступательного движения используют $\dot{p} = v$, $\dot{v} = a$. В дискретном виде $p(k+1) = p(k) + v(k)\Delta t + 0{,}5a(k)\Delta t^2$. Реальный мультикоптер является нелинейной динамической системой, поэтому данная модель пригодна только для верхнего уровня планирования и не заменяет уравнения ориентации и тяги.

Типовой критерий оптимизации маршрута может иметь вид $J = w_1L + w_2T + w_3E + w_4R$, где L — длина маршрута, T — время, E — оценка энергозатрат, R — штраф за риск, wi — веса. Для безопасного планирования необходимо учитывать запретные зоны, высотные ограничения, запас энергии, максимальные ускорения и возможность возврата.

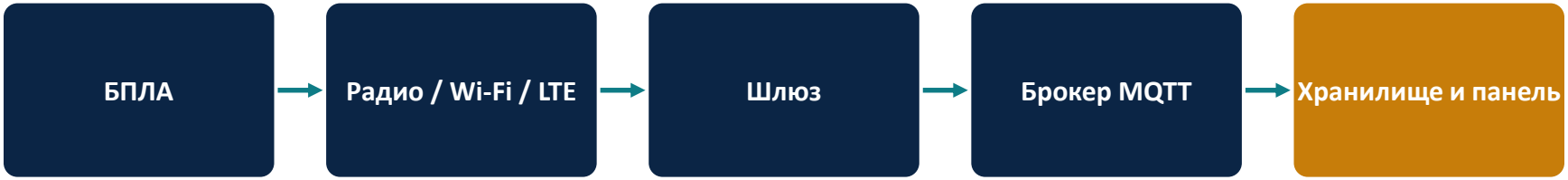
7 IoT-связь, надёжность и безопасность

IoT-интеграция позволяет передавать телеметрию, состояние аккумулятора, координаты, диагностические события и результаты распознавания в наземную информационную систему. MQTT может применяться для обмена некритичными сообщениями мониторинга, тогда как контуры стабилизации и аварийные реакции должны оставаться локальными.

Полная задержка обработки включает Тобщ = Тзахват + Тпредобработка + Тинференс + Тпередача + Трешение. Если система должна обрабатывать поток с частотой f кадров/с без накопления очереди, средняя вычислительная производительность должна соответствовать периоду 1/f, а критические решения дополнительно проверяются по максимальной задержке.

При отказе камеры, перегреве вычислительного модуля, потере GNSS или связи необходимо предусматривать безопасный режим. Кибербезопасность включает аутентификацию, защищённые каналы, минимизацию доступа к управлению, журналирование и обновление программного обеспечения. Полёты выполняются только с соблюдением действующих местных правил и эксплуатационных ограничений.

10 Канал телеметрии и мониторинга



телеметрия, события	линия связи	Raspberry Pi или ПК	публикация и подписка	ThingSpeak, база данных

Рис. 23. Аварийная защита выполняется на борту независимо от канала связи.

8 Практический пример: распознавание объектов местности

Рассмотрим исследовательский прототип: камера БПЛА передаёт видеопоток на бортовой мини-компьютер, где нейросетевая модель обнаруживает заданные классы объектов. Результаты записываются в журнал с временными метками и координатами, полученными от навигационного модуля. В этой конфигурации распознавание используется для наблюдения и формирования карты обнаружений, а не для прямого управления двигателями.

Алгоритм состоит из этапов: (1) инициализация камеры и модели; (2) захват кадра; (3) предварительная обработка; (4) инференс; (5) фильтрация по порогу уверенности; (6) привязка к временной метке и телеметрии; (7) запись результатов; (8) повторение цикла. При наличии калибровки и оценки позы можно вычислять приблизительное положение объектов на земле.

Для эксперимента необходимо фиксировать высоту и скорость полёта, тип освещения, разрешение, порог уверенности, версию модели и аппаратную платформу. Отдельно измеряют качество распознавания, задержку обработки, пропущенные кадры и энергопотребление. Вывод о пригодности системы делают только по результатам повторяемых испытаний.

Сравнение вычислительных платформ

Платформа	Преимущества	Ограничения
Raspberry Pi	Доступность, GPIO, экосистема Python/OpenCV	Ограниченная скорость тяжёлых нейросетей без ускорителя
NVIDIA Jetson	GPU-инференс, инструменты оптимизации моделей	Более высокие требования к питанию и теплоотводу
Google Coral Edge TPU	Энергоэффективный инференс совместимых моделей	Ограничения по поддерживаемым операциям и форматам
Полётный контроллер	Детерминированное выполнение базовых контуров управления	Не предназначен для ресурсоёмкого компьютерного зрения

Пример Python: обнаружение объектов в видеопотоке

Ниже приведён демонстрационный фрагмент для локального стенда с установленными пакетами ultralytics и opencv-python. Путь к модели и номер камеры необходимо адаптировать к оборудованию. Код не управляет полётом.

```
import cv2
from ultralytics import YOLO

model = YOLO("yolov8n.pt")  # демонстрацион
ная модель
camera = cv2.VideoCapture(0)
if not camera.isOpened():
    raise RuntimeE
    rror("Камера недоступна")
    try:

    while True:

    ok, frame = camera.read()
        if not ok:
```

14 Расчётный пример производительности

Пусть экспериментально измерены: захват кадра — 12 мс, предобработка — 8 мс, инференс — 42 мс, постобработка — 6 мс, запись результата — 4 мс. Тогда последовательная задержка одного цикла равна 72 мс, а теоретическая пропускная способность без параллелизма составляет 1000/72 ≈ 13,9 кадра/с. Это расчётная иллюстрация, а не результат измерения конкретного устройства. При задании 20 кадр/с следует уменьшить задержку или применять конвейерную обработку с контролем очередей.

15 Методика экспериментальной проверки

- Сформировать независимый тестовый набор кадров из разных полётов и погодных условий.
- Оценить Precision, Recall, F1 и mAP отдельно по классам объектов.
- Измерить среднюю, 95-й процентиль и максимальную задержку обработки.
- Проверить устойчивость к смазыванию, изменению освещённости и частичной окклюзии.
- Сравнить Raspberry Pi и нейроускоритель при одинаковой модели и входном разрешении.
- Зафиксировать массу, энергопотребление, температуру устройства и длительность автономной работы.
- Провести сначала стендовые, затем разрешённые полевые испытания с независимыми средствами безопасности.

16 Заключение

Машинное зрение превращает БПЛА из платформы сбора изображений в элемент интеллектуальной системы наблюдения и принятия решений. Эффективность определяется согласованной работой камеры, вычислительного модуля, алгоритмов обнаружения, средств локализации и автопилота. Для научно обоснованного выбора архитектуры необходимо сопоставлять точность, задержку, энергопотребление и устойчивость алгоритмов в репрезентативных условиях эксплуатации.

17 Контрольные вопросы

1. Какие функции выполняет машинное зрение в интеллектуальном БПЛА?
2. Чем различаются детектирование, классификация, сегментация и трекинг?
3. Какие компоненты входят в архитектуру бортовой системы компьютерного зрения?
4. Почему низкоуровневое управление следует отделять от нейросетевого инференса?
5. Как описывается проективная модель камеры?
6. Как влияют калибровка и дисторсия на оценку положения объектов?
7. Что представляет собой IoU и для чего используется эта метрика?
8. Как рассчитываются Precision, Recall и F1-score?
9. Почему нельзя допускать утечки похожих кадров между выборками?
10. Как работает визуальная одометрия и в чём ограничение монокулярной камеры?
11. Какие задачи решает визуально-инерциальная локализация?

17 Контрольные вопросы (продолжение)

12. Какие ограничения следует учитывать при планировании траектории БПЛА?
13. Из каких компонентов складывается задержка обработки видеокадра?
14. Какие параметры следует регистрировать при экспериментальном сравнении платформ?
15. Какие отказоустойчивые режимы нужны при потере камеры или связи?

Рекомендуемая литература и ресурсы

- **Szeliski R. Computer Vision:** Algorithms and Applications. 2nd ed. Springer, 2022.
- Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge University Press, 2004.
- **OpenCV Documentation:** https://docs.opencv.org/
- **PX4 Autopilot User Guide:** https://docs.px4.io/
- **ArduPilot Documentation:** https://ardupilot.org/
- **Ultralytics Documentation:** https://docs.ultralytics.com/
- **NVIDIA Jetson Documentation:** https://docs.nvidia.com/jetson/