



8.7 关键工具与框架接口



本章目录

关键工具与框架接口

第八章 多模态感知与智能决策

- 8.1 背景意义与图像处理基础
- 8.2 目标识别、跟踪与光流估计
- 8.3 定位建图与预测规划
- 8.4 常用感知传感器
- 8.5 多模态感知坐标转换
- 8.6 多模融合与控制决策
- 8.7 关键工具与框架接口**
- 8.8 实验案例与真机迁移
- 8.9 习题与课堂讨论

本节 5 个教学单元

8.7.1 核心开发工具

教材对应：第8.1.3.1节

8.7.2 MATLAB/Simulink算法库

教材对应：第8.1.3.2节

8.7.3 常用开源算法库

教材对应：第8.1.3.3节

8.7.4 仿真开发框架

教材对应：第8.2.1节

8.7.5 常用接口及二次开发

教材对应：第8.2.3节



8.7.1 核心开发工具：工具分工

同一个任务需要不同层次的工具

层次	负责什么	可检查的产物
运行与通信	Ubuntu/Linux、ROS 提供进程环境和消息传递	运行中的节点、话题 及数据时间戳
算法实现	Python/C++、OpenCV 与深度学习框架	图像特征、目标信息 模型权重及推理输出
计算与复现	数值库、Ceres、Git 支持求解、分析和版本记录	最优参数、结果曲线 可追溯的实验配置

8.7.1 核心开发工具：Ubuntu与ROS

操作系统运行程序，中间件连接模块



Ubuntu/Linux： 提供运行进程、设备驱动、文件和网络；系统版本与算法依赖匹配。

机器人操作系统（ROS）： 以节点、话题和消息连接模块；TF表示坐标变换，消息携带采样时间。

Python接口： rospy对应ROS1，rclpy对应ROS2；跨版本接入需要对应的节点与消息支持。

8.7.1 核心开发工具：语言与OpenCV

用同一套数据约定衔接原型和部署

工具	典型输入与输出	接入方式
Python	图像、数组、模型 → 算法原型与训练结果	平台Python接口 快速试验和集成
C/C++	同一物理数据与参数 → 目标平台在线程序	编译链接库与ROS节点 控制内存和并发
OpenCV	图像→滤波、特征、 目标框或光流	Python/C++调用 检查尺寸和BGR通道

运行时延由算法、后端和数据传递共同决定。

8.7.1 核心开发工具：深度学习框架

训练得到权重，推理得到任务输出



PyTorch / TensorFlow: 支持网络建模、自动求导和设备运算；训练与在线推理采用不同的数据流。

最小例子: 用标注圆环图训练检测器，再把新帧送入固定权重模型，输出框、类别与置信度。

8.7.1 核心开发工具：数值计算

数组、求解与曲线分别回答什么

工具	输入	输出示例
NumPy	像素、状态或轨迹数组	矩阵运算与统计量
SciPy	观测、目标函数或信号	最优参数、滤波结果
Matplotlib	时间和数值序列	误差曲线、轨迹、热力图

教学算例：三次水平误差为3、-4、0像素，均方根保留偏差大小。

$$\text{RMSE} = \sqrt{\frac{3^2 + (-4)^2 + 0^2}{3}} \approx 2.89 \text{ px}$$

8.7.1 核心开发工具：Ceres Solver

用观测残差求最佳参数

Ceres Solver：非线性最小二乘求解库，调节参数以减小残差代价。
残差是预测值减观测值；本例 x 为位置，观测为2米、4米。
用途包括束调整、位姿图优化、传感器标定和轨迹平滑。

$$J(x) = (x - 2)^2 + (x - 4)^2, \quad x \text{ in m}$$

$$\frac{dJ}{dx} = 4x - 12 = 0 \quad \Rightarrow \quad x^* = 3 \text{ m}$$

教学结果：初值1米时平方残差和为10平方米，最优点3米时为2平方米。
优化减小了两次观测的冲突；测量质量仍决定结果可信度。

Ceres残差、参数块与束调整示例

https://ceres-solver.readthedocs.io/latest/nnls_tutorial.html



8.7 关键工具与框架接口



8.7.1 核心开发工具：Git与GitHub

把一次实验还原为可重复的配置

记录对象	具体内容	复现时回答的问题
代码与依赖	Git提交、库版本 Python/ROS与目标架构	能否建立同一运行环境
数据与模型	数据划分、随机种子 权重文件及哈希	使用的训练产物是否相同
任务与结果	Config、任务输入 时间戳、指标与日志	是否在同样工况下比较

Git记录代码版本；GitHub托管仓库并支持协作。

哈希是文件内容的摘要，用于核对权重与数据是否一致。



8.7.1 核心开发工具：时延算例

一帧的总耗时决定能否及时控制

环节	给定耗时	输出
取图与预处理	6 + 4 ms	当前帧与图像张量
推理与规划	18 + 7 ms	目标信息与轨迹
发送控制	2 ms	控制指令

$$T_{\text{total}} = 37 \text{ ms} > T_{30\text{Hz}} = \frac{1000}{30} \approx 33.3 \text{ ms}$$

结果解释：串行链路约能处理27帧/秒；持续接收30帧/秒会产生等待。



8.7 关键工具与框架接口



本章目录

关键工具与框架接口

第八章 多模态感知与智能决策

- 8.1 背景意义与图像处理基础
- 8.2 目标识别、跟踪与光流估计
- 8.3 定位建图与预测规划
- 8.4 常用感知传感器
- 8.5 多模态感知坐标转换
- 8.6 多模融合与控制决策
- 8.7 关键工具与框架接口**
- 8.8 实验案例与真机迁移
- 8.9 习题与课堂讨论

本节 5 个教学单元

8.7.1 核心开发工具

教材对应：第8.1.3.1节

8.7.2 MATLAB/Simulink算法库

教材对应：第8.1.3.2节

8.7.3 常用开源算法库

教材对应：第8.1.3.3节

8.7.4 仿真开发框架

教材对应：第8.2.1节

8.7.5 常用接口及二次开发

教材对应：第8.2.3节





8.7 关键工具与框架接口



8.7.2 MATLAB/Simulink算法库：视觉与机器人接口

前端提取观测，接口把观测传给系统

工具箱	职责	最小输入→输出
Computer Vision	检测、特征、图像几何与对象跟踪	图像→目标框、特征或跟踪结果
Robotics System	机器人模型、运动学及机器人算法	模型和关节状态→位姿与运动结果
ROS Toolbox	消息收发、ROS节点与模型集成	算法结果→ROS消息 订阅消息→模型输入



8.7.2 MATLAB/Simulink算法库：无人机与导航 飞行任务和地图规划如何配合



UAV Toolbox：面向无人机任务、飞行场景与仿真；组织载具、航点和任务参考。

Navigation Toolbox：面向地图表示、路径搜索、轨迹优化及代价评估；输出可行路径或轨迹。

接入例子：以障碍地图、当前位姿和目标环为输入，规划绕障轨迹，再由飞行控制器跟踪。

8.7.2 MATLAB/Simulink算法库：融合与Stateflow

连续状态估计与离散任务切换



融合与跟踪工具箱：把多源观测、关联关系与运动模型用于定位和目标状态估计。

Stateflow：用状态、事件、条件和动作描述时序，管理起飞、探索、避障和降落。

例子：障碍距离小于2米时由前进切换到避障；距离恢复且路径可行时再回到前进。

8.7.2 MATLAB/Simulink算法库：Embedded Coder

从模型生成代码还要适配目标环境



Embedded Coder：支持的模型生成C/C++；需匹配目标编译器、库和接口。

动态链接库（DLL）：保存可由主程序在运行时调用的函数；运动模型DLL可接入CopterSim。

板上结果：取得新数据、输出动作并满足时限，才构成可检验的部署结果。

生成代码缩短实现过程，接口语义仍需保持一致。



8.7 关键工具与框架接口



8.7.2 MATLAB/Simulink算法库：强化学习与MPC

学习策略和在线优化怎样输出动作

方法	输入与动作产生方式	重点检查
强化学习工具箱	观测、动作与奖励 训练策略后在线推理	奖励设计、训练覆盖 与新工况表现
MPC工具箱	状态、参考、预测模型 在线优化约束动作序列	模型误差、可行约束 与每次求解耗时

模型预测控制（MPC）：利用未来状态预测选择当前动作。

强化学习（RL）：通过与环境交互的奖励学习动作策略。



8.7.2 MATLAB/Simulink算法库：模型闭环算例

条件切换后仍由控制器计算速度

教学条件：障碍距离1.6米，切换阈值2米，进入避障状态。
当前位置0米，1秒后参考位置1米；避障速度上限0.5米/秒。

$$J(v) = (v\Delta t - 1)^2 + 0.2v^2, \quad \Delta t = 1 \text{ s}$$

候选速度	代价	约束与结果
0.5米/秒	0.30	满足限速，可执行
1.0米/秒	0.20	超过限速，排除

结果解释：任务逻辑决定控制目标和约束，优化器在可行范围内选择动作。



8.7 关键工具与框架接口



本章目录

关键工具与框架接口

第八章 多模态感知与智能决策

- 8.1 背景意义与图像处理基础
- 8.2 目标识别、跟踪与光流估计
- 8.3 定位建图与预测规划
- 8.4 常用感知传感器
- 8.5 多模态感知坐标转换
- 8.6 多模融合与控制决策
- 8.7 关键工具与框架接口**
- 8.8 实验案例与真机迁移
- 8.9 习题与课堂讨论

本节 5 个教学单元

8.7.1 核心开发工具

教材对应：第8.1.3.1节

8.7.2 MATLAB/Simulink算法库

教材对应：第8.1.3.2节

8.7.3 常用开源算法库

教材对应：第8.1.3.3节

8.7.4 仿真开发框架

教材对应：第8.2.1节

8.7.5 常用接口及二次开发

教材对应：第8.2.3节



8.7.3 常用开源算法库：算法职责

检测、跟踪、定位和规划各产出什么

任务	教材算法	关键输出
检测与身份	YOLO / YOLOE DeepSORT	框、类别、掩膜 跨帧目标身份
自定位与地图	VINS-Fusion、ORB-SLAM FAST-LIO系列	载具位姿、轨迹 及不同形式的地图
运动规划	FASTER、EGO-Planner EGO-Swarm	可行轨迹与时间安排 邻机避碰参考

8.7.3 常用开源算法库：YOLO与YOLOE

类别约定决定检测器如何使用

模型路线	输入约定	输出与用途
YOLOv5 / YOLOv8 YOLO11	图像+固定权重 类别由训练任务规定	类别、置信度和目标框 可选掩膜或关键点任务
YOLOE	图像+文本或视觉提示 也有无提示路线	开放词汇检测与分割 目标范围由模型和提示限定

最小例子：用“圆环”提示寻找目标，检查框是否对应任务所需的通行环。

YOLOE三种提示模式与训练示例

<https://github.com/THU-MIG/yoloe>

8.7.3 常用开源算法库：DeepSORT

给检测结果保持跨帧身份



$$C = \begin{bmatrix} 0.2 & 0.9 \\ 0.8 & 0.1 \end{bmatrix}$$

教学关联代价：
行对应已有轨迹，列对应新检测；数值
越小越匹配。

$$J_{\text{direct}} = 0.2 + 0.1 = 0.3 < 1.7 = 0.9 + 0.8 = J_{\text{cross}}$$

结果解释：保持两对象身份；遮挡时利用预测和外观再次关联。
若两种分配都不合理，应保留丢失状态，而非强行交换身份。

8.7.3 常用开源算法库：VINS-Fusion

相机和IMU约束同一个载具状态



主要输入： 图像与可选IMU；相机内参、相机—IMU外参、时间差和噪声参数。

估计输出： 载具自身的位置、姿态与运动状态；可选回环帮助抑制长期漂移。

全球导航卫星系统（GNSS）： 可通过独立全局融合模块与局部估计结合，提供全局坐标约束。

VINS-Fusion传感器配置与global_fusion

<https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>



8.7 关键工具与框架接口



8.7.3 常用开源算法库：ORB-SLAM系列

特征重投影把轨迹和地图联系起来

系统	输入与能力	使用条件
ORB-SLAM2	单目、双目、RGB-D 稀疏地图与定位	标定、纹理、视差影响关联 纯单目无绝对米制尺度
ORB-SLAM3	视觉或视觉惯性 多地图与回环	IMU时序及相机外参 影响融合估计

最小接入：仿真图像＋相机标定→相机位姿与稀疏地图；纹理和运动影响关联。

ORB-SLAM3支持的传感器与运行配置

https://github.com/UZ-SLAMLab/ORB_SLAM3



8.7.3 常用开源算法库：激光惯性里程计

点云配准与IMU传播怎样互补

算法	主要机制	共同接口需求
FAST-LIO	迭代扩展卡尔曼滤波 融合雷达与惯性	点云、IMU、安装外参 与采样时间
FAST-LIO2	直接原始点云配准 增量点云地图	点内时间与去畸变 影响快速运动结果
Faster-LIO	稀疏增量体素 加速地图查询和更新	输出位姿与地图 后接规划或控制

激光惯性里程计（LIO）：用激光雷达和IMU估计自身运动。
先核传感器约定，再用轨迹连续性、地图一致性与时延评价实现。

8.7.3 常用开源算法库：FASTER与EGO-Planner

未知空间和局部障碍如何进入规划

规划器	处理的关键问题	输入与产物
FASTER	未知空间中的快速前进 与已知自由空间备份	状态、地图、目标 →主轨迹与停止备份
EGO-Planner	局部碰撞约束与 B样条梯度优化	状态、局部地图、目标 →可行局部轨迹

欧氏符号距离场（ESDF）：按位置表示到障碍边界的距离及内外关系。
EGO-Planner采用无需构建ESDF的局部优化路线。

8.7.3 常用开源算法库：EGO-Swarm

共享未来轨迹帮助分布式避碰

EGO-Swarm：各机基于自身观测与邻机轨迹进行局部优化。
输入包括自身位姿、障碍、任务目标和邻机带时间的计划轨迹。
 p_1, p_2 是位置（米）， t 是时间（秒）；最小间距1米。

$$p_1(t) = t, \quad p_2(t) = 4 - t, \quad d(t) = |4 - 2t| \text{ m}$$

预测时刻	一维间距	结果解释
1秒	2米	大于1米，暂满足间距
2秒	0米	将交会，提前改轨迹

邻机轨迹只有带着正确时刻和坐标才有避碰价值。



8.7 关键工具与框架接口



本章目录

关键工具与框架接口

第八章 多模态感知与智能决策

- 8.1 背景意义与图像处理基础
- 8.2 目标识别、跟踪与光流估计
- 8.3 定位建图与预测规划
- 8.4 常用感知传感器
- 8.5 多模态感知坐标转换
- 8.6 多模融合与控制决策
- 8.7 关键工具与框架接口**
- 8.8 实验案例与真机迁移
- 8.9 习题与课堂讨论

本节 5 个教学单元

8.7.1 核心开发工具

教材对应：第8.1.3.1节

8.7.2 MATLAB/Simulink算法库

教材对应：第8.1.3.2节

8.7.3 常用开源算法库

教材对应：第8.1.3.3节

8.7.4 仿真开发框架

教材对应：第8.2.1节

8.7.5 常用接口及二次开发

教材对应：第8.2.3节



8.7.4 仿真开发框架：总体框架

沿图中数据流找到四类核心模块

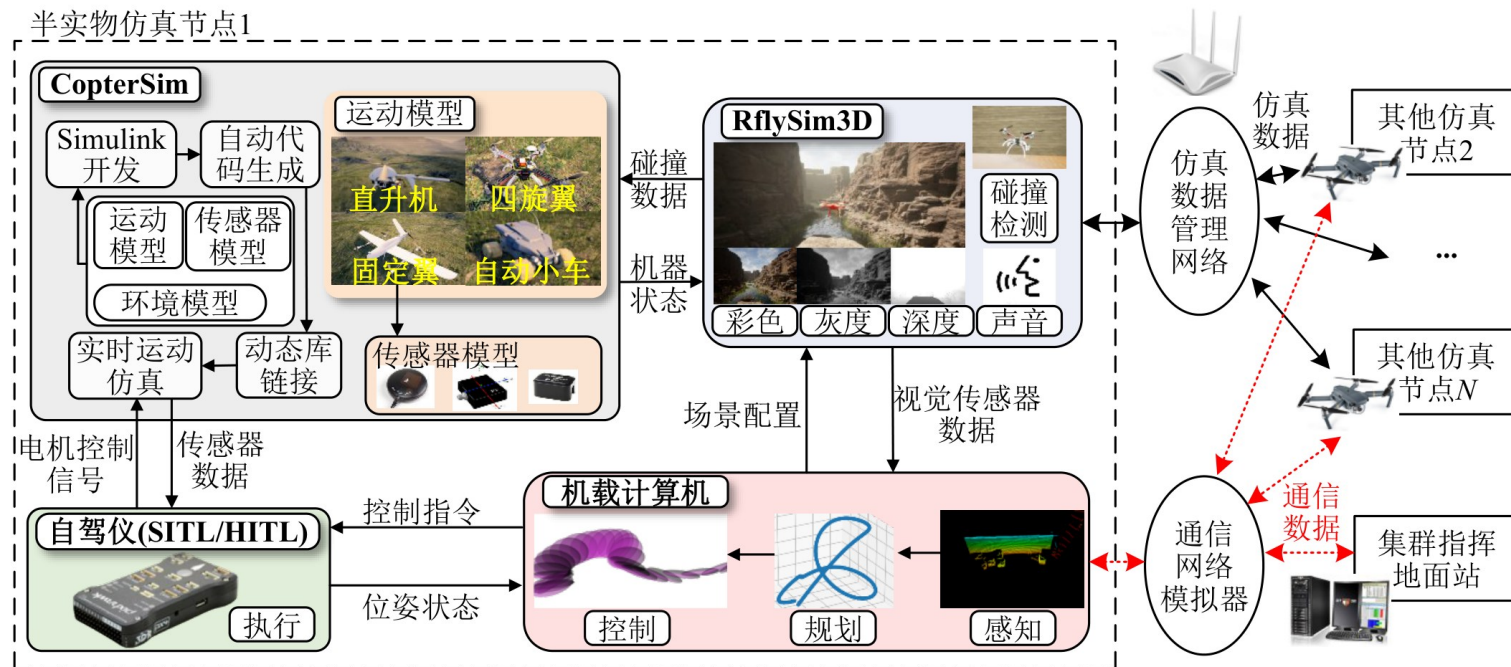


图8.8 感知决策仿真总体框架

读图：电机—运动—传感器构成底层回路；图像—算法—外部指令构成上层回路。

8.7.4 仿真开发框架：图8.8上部放大

运动模型和场景渲染分别输出什么

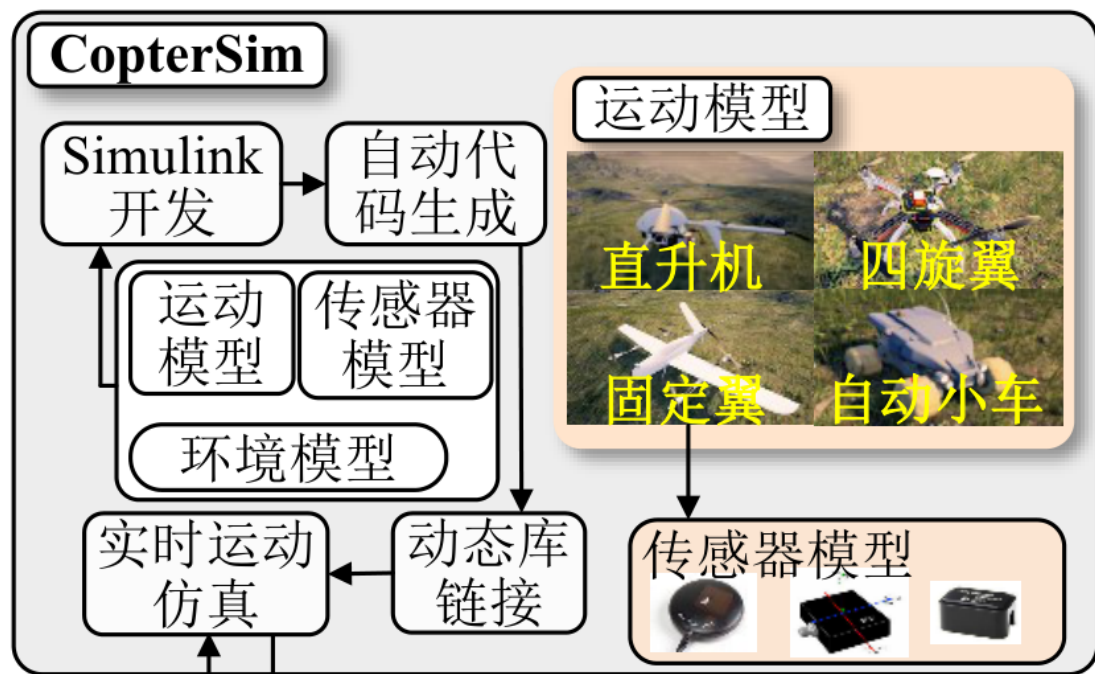


图8.8局部：CopterSim运动与传感器模型



图8.8局部：场景、观测和碰撞检测

读图：运动状态送往RflySim3D；场景碰撞信息反馈给CopterSim。

8.7.4 仿真开发框架：图8.8下部放大

机载算法与自驾仪形成指令和状态回路

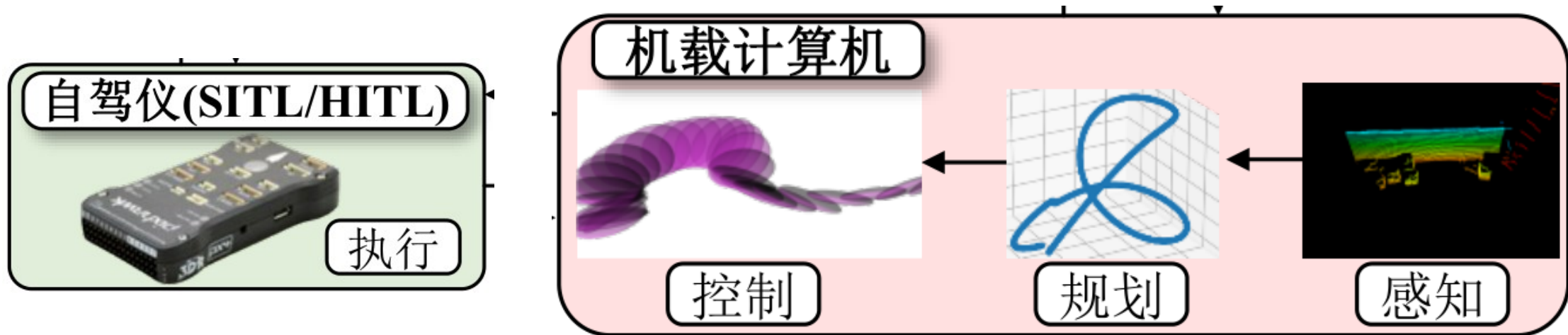


图8.8局部：自驾仪执行

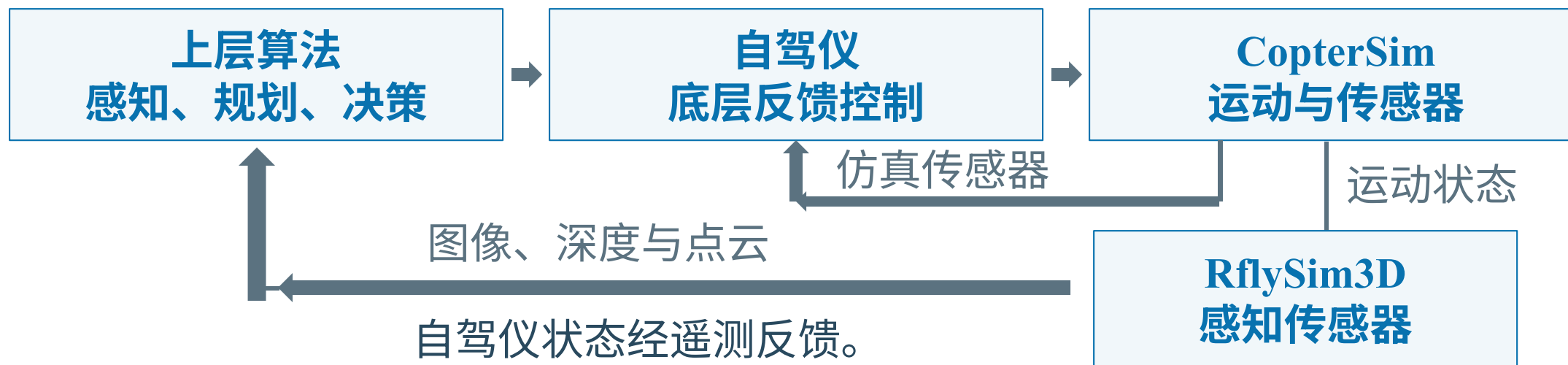
图8.8局部：机载感知、规划、控制

指令链：机载算法→自驾仪→电机控制；CopterSim更新运动。

状态链：传感器→自驾仪→位姿状态；机载算法依据新状态更新决策。

8.7.4 仿真开发框架：模块与开发路径

动力学、感知与控制两个闭环中协作



Python: SDK取数与原型处理；单机可用共享内存。

Simulink / ROS: 模型接入数据与控制接口；ROS消息连接算法和外控。

8.7.4 仿真开发框架：SITL与HITL

真实硬件加入后检验了什么

比较项	软件在环（SITL）	硬件在环（HITL）
自驾仪	PX4软件在环境中运行	真实自驾仪运行固件
上层算法	Windows/Python 或WinWSL/ROS	真实机载计算机 目标架构与设备库
运动与观测	CopterSim和RflySim3D	仍由仿真模型与场景提供
验收重点	逻辑、接口和算法产物	板上时延、资源与通信

8.7.4 仿真开发框架：分布式仿真

多机共享图像先计算带宽需求

教学条件：4路640×480、3通道图像，各30帧/秒；uint8每通道1字节。
每个节点独立配置载具编号、传感器编号、IP、端口和时基。

$$B = 4 \times 640 \times 480 \times 3 \times 30 = 110592000 \text{ byte/s}$$

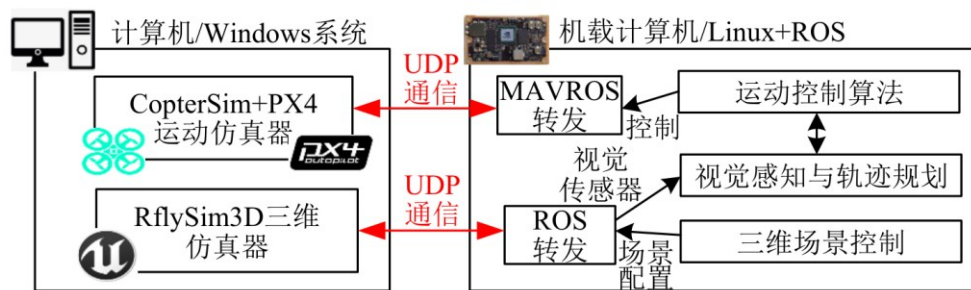
$$R = 8B = 884.736 \text{ Mbit/s}$$

结果解释：这是净图像负载，还没有包头、点云和控制消息；标称链路速率不能全用于图像。

系统选择：按任务发送观测或结果；权衡压缩、解码、丢包与时效。

8.7.4 仿真开发框架：ROS兼容

数据转发与运动控制是两条接口链



(a) ROS通信框架



(b) 真机迁移框架

图8.9 ROS仿真兼容与真机迁移

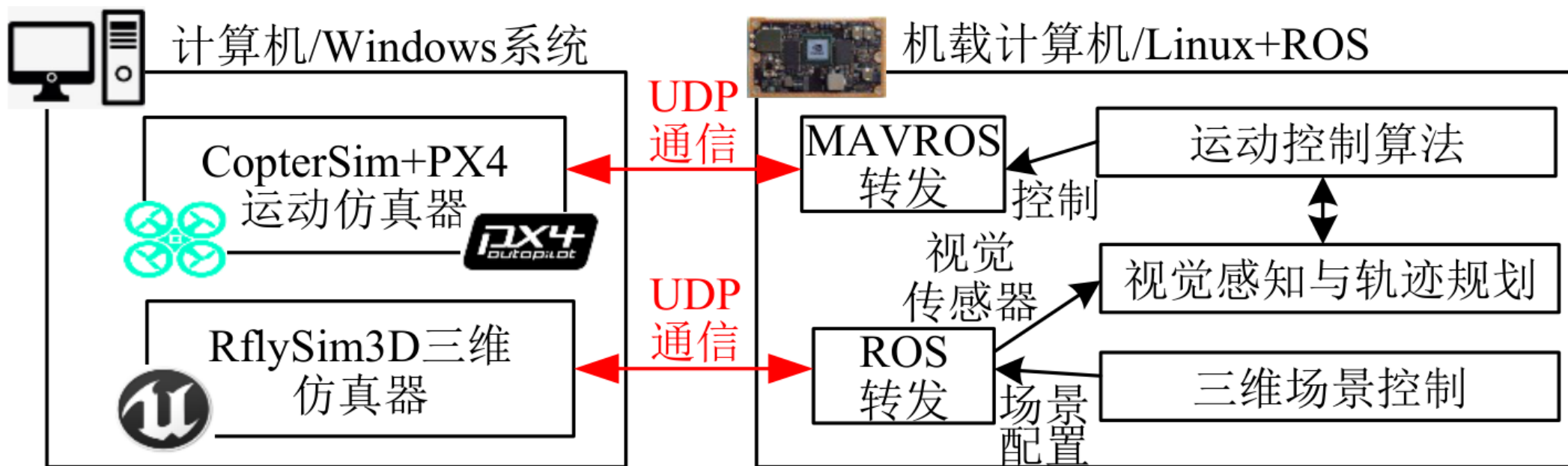
视觉数据链： RflySim3D→网络接收→ROS消息→感知与规划。

运动控制链： 自驾仪状态→MAVROS；规划目标→MAVROS→PX4。

迁移边界： 真实传感器替换观测源；真实载具替换运动仿真。

8.7.4 仿真开发框架：图8.9(a)放大

仿真图像进入ROS，外部目标进入PX4



(a) ROS通信框架

图8.9(a) 原图：仿真运动、观测与上层程序

读图：观测到达不等于控制已生效，必须同时检查数据链与目标执行链。

8.7 关键工具与框架接口

8.7.4 仿真开发框架：图8.9(b)放大

保留算法约定，替换真实观测和执行平台

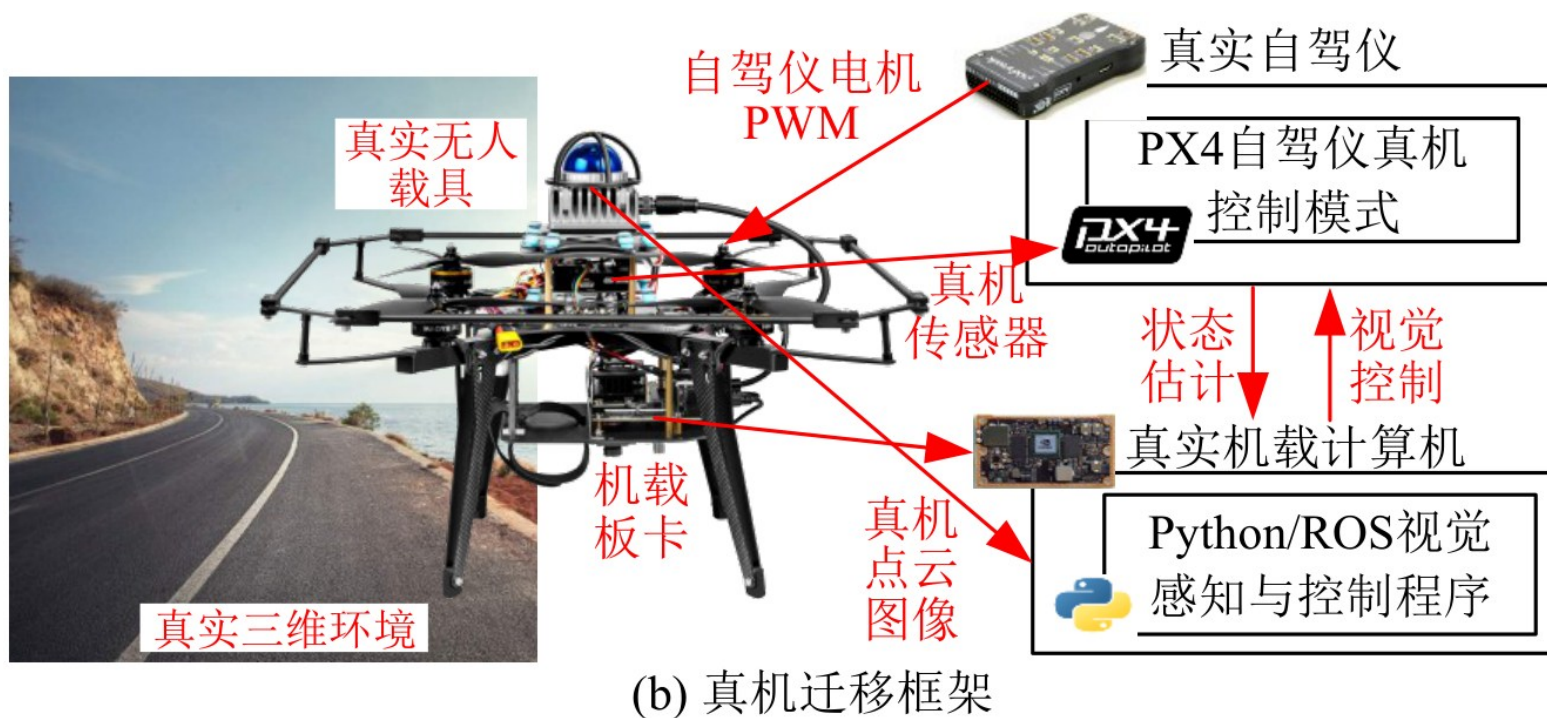


图8.9(b) 原图：真实传感器、载具与机载板

迁移路线：扫描→建模→虚实增强→在环→真机（图8.10见课件8.8.4）。

8.7.4 仿真开发框架：开发流程

用数据、决策和响应判断闭环是否有效

给定现象	已通的环节	继续检查的对象
图像刷新，目标框不变	传感器接收	算法是否读新帧 权重与目标类别
轨迹更新，载具不响应	感知与规划	外控模式、目标坐标 发布频率与控制链
载具飞行但仍穿障碍	运动控制	障碍坐标、规划约束 跟踪误差与采样时刻



8.7 关键工具与框架接口



本章目录

关键工具与框架接口

第八章 多模态感知与智能决策

- 8.1 背景意义与图像处理基础
- 8.2 目标识别、跟踪与光流估计
- 8.3 定位建图与预测规划
- 8.4 常用感知传感器
- 8.5 多模态感知坐标转换
- 8.6 多模融合与控制决策
- 8.7 关键工具与框架接口**
- 8.8 实验案例与真机迁移
- 8.9 习题与课堂讨论

本节 5 个教学单元

8.7.1 核心开发工具

教材对应：第8.1.3.1节

8.7.2 MATLAB/Simulink算法库

教材对应：第8.1.3.2节

8.7.3 常用开源算法库

教材对应：第8.1.3.3节

8.7.4 仿真开发框架

教材对应：第8.2.1节

8.7.5 常用接口及二次开发

教材对应：第8.2.3节





8.7 关键工具与框架接口



8.7.5 常用接口及二次开发：环境与文档

参数协议和Python方法分开查找

环境选择： Windows/Python先验证本机取图；WinWSL、虚拟机、Linux或机载板按ROS及硬件依赖选择。

学生资料根： [安装目录]/RflySimAPIs/8.RflySimVision/API.pdf与VisionSensorAPI.pdf分别用于开发说明和参数协议。

SDK方法文档： [安装目录]/RflySimAPIs/RflySimSDK/html/index.html
下钻4.DevRef/pysdk/vision/viscap.html核对实际方法。



8.7.5 常用接口及二次开发: Config.json

传感器编号与列表下标承担不同角色

字段	意义	入门设置或读法
SeqID / TargetCopter	传感器编号 / 载具编号	0 / 1; 编号不是列表下标
TypeID	决定数据类型与解码	1为彩色图, 2为深度图
DataWidth / DataHeight	输出分辨率, 单位像素	640 / 480
DataCheckFreq / 协议	检查更新频率 / 传输方式	30Hz; SendProtocol[0]=0 (共享内存)

例: 配置列表仅一项SeqID=7时, 读Img[0]; 用VisSensor[0].SeqID核对编号。

8.7.5 常用接口及二次开发：安装外参

固定机体安装先声明单位和参考系

机体参考点
前右下 (FRD)



相机安装点
前移0.3米

TargetMountType=0：固定在载具参考点。

SensorPosXYZ=[0.3,0,0]：单位米，沿机体前向偏移。

SensorAngEular=[0,0,0]：单位度，以当前安装方式解释姿态。

EularOrQuat：选择欧拉角或四元数配置；保留实际键名。

相机光学轴、机体FRD和地面坐标的转换见课件8.5。

8.7.5 常用接口及二次开发：接口生命周期 请求成功之后还要等首帧

```
import VisionCaptureApi
vis = VisionCaptureApi.VisionCaptureApi()
if not vis.jsonLoad():
    raise RuntimeError("Config load failed")
if not vis.sendReqToUE4():
    raise RuntimeError("Request failed")
vis.startImgCap()
```

配置：jsonLoad()按入口脚本位置读取Config.json。

首帧：等待hasData[0]，设超时；再检查数组尺寸与类型。

生命周期：配置→请求→启动接收→首帧超时检查→循环处理→finally清理。
请求返回True仅覆盖请求阶段，首帧有效由接收数据判断。

8.7.5 常用接口及二次开发：读取新帧

处理循环必须辨别时间戳变化

```
import numpy as np
if vis.hasData[0]:
    with vis.Img_lock[0]:
        frame = vis.Img[0].copy()
        stamp = np.asarray(vis.timeStmp[0]).item()
    if stamp != last_stamp:
        last_stamp = stamp
        process(frame)
```

图像锁：保护图像副本；时刻另行更新，融合需成对消息。

timeStmp[0]：仅过滤重复时刻，不保证图像与时刻成对。

last_stamp在循环前初始化为None；process是自定义的感知函数。完整外层循环还要处理超时、结束信号和finally中的资源清理。



8.7 关键工具与框架接口



8.7.5 常用接口及二次开发：彩色图像

RGB类型返回BGR数组

数据层	当前返回或约定	具体例子
传感器类型	TypeID=1，彩色图	类型名RGB不决定数组通道
Python数组	uint8； $H \times W \times 3$ ；BGR	纯红像素=[0,0,255]
算法输入	按模型要求转换通道	cv2.COLOR_BGR2RGB 或cv2.COLOR_BGR2HSV



8.7.5 常用接口及二次开发：深度数据

先屏蔽无效值，再换成米

TypeID=2: uint16单通道；传输方式影响数组是 $H \times W$ 还是 $H \times W \times 1$ 。
深度刻度 u_m 为otherParams[2]，默认每计数0.001米；65535先屏蔽。

$$d_m = d_{\text{raw}} u_m = 2370 \times 0.001 = 2.370 \text{ m}$$

原始整数	处理结果	距离判断
2370	有效深度2.370米	与米制阈值比较
65535	无效观测，掩码排除	保留未知，另取有效观测

结果解释：未知值不提供可通行距离；单位和有效性都在进入算法前确定。



8.7 关键工具与框架接口



8.7.5 常用接口及二次开发：分割与点云

渲染标签和算法估计有不同来源

数据	当前接口含义	接入注意
分割图 Type4	uint8，BGR标签渲染图	标签是真值来源 模型输出另行评价
点云 Type7/20等	$N \times 4$ 浮点数组 XYZ已解码为米	第4列语义或零 不再整体缩放
测距 / 光流	Img[i]内测距对象 vis.OpticalFlowSensor	按结构字段读取 不是通用图像数组

点云Type7、吊舱与红外等功能按安装授权使用；入门先核RGB与深度。



8.7.5 常用接口及二次开发：时间与融合

同时读取不代表同一采样时刻

图像：timeStmp为相对时刻，imgStmp为映射后时刻。

IMU：sendImuReqCopterSim()请求数据；按同一时基缓存并配对。

$$\Delta t = |t_{\text{imu}} - t_{\text{cam}}|, \quad t_{\text{cam}} = 1.000 \text{ s}$$

给定IMU时刻	与相机偏差	教学10ms门限结果
1.005秒	5毫秒	进入配对候选
1.025秒	25毫秒	丢弃或按模型插值处理

AlignTime()更新时刻字段；融合器仍需确定配对、插值和有效性规则。

8.7.5 常用接口及二次开发：ROS与二次开发

消息约定把仿真数据接到算法



图像与点云：核对encoding、shape、单位、frame_id和TF；消息名不代替坐标转换。

ROS路线：视觉数据经消息输入算法；自驾仪状态与位置、速度、姿态目标经MAVROS连接。

Simulink路线：以数据采集和外部控制模块接入算法模型；代码生成与目标支持按具体模块核对。

8.7.5 常用接口及二次开发：控制与退出

感知停更时控制状态也要有去向

```
try:  
    run_perception_loop()  
finally:  
    vis.stopRun()  
    cv2.destroyAllWindows()
```

run_perception_loop是自定义外层循环；
先决定载具结束动作，再释放接收资源。

接口分工： VisionCaptureApi取数；
PX4MavCtrl发布外控；场景接口管理
对象，评估程序计算指标。

退出分支： 共享内存stopRun；
UDP/IMU按线程和套接字处理。载具
状态由控制退出策略决定。

资源结束与载具结束动作都要在控制流程中明确。



8.7 关键工具与框架接口



练习与讨论

诊断一个双相机避障系统

- 1. 编号与新帧：** SeqID为7、9，两项按此顺序配置，应访问哪两个Img下标？怎样避免重复处理？
- 2. 物理量：** 深度刻度0.001米，中心整数2370与无效值65535分别如何进入避障算法？
- 3. 执行与迁移：** 输入30Hz，推理100ms。系统会发生什么？移到ARM机载板要取得哪些验证产物？





8.7 关键工具与框架接口



参考要点

从数据语义追到系统响应

编号与新帧：读Img[0]和Img[1]；用VisSensor[i].SeqID确认编号，比较采样时间戳后处理新帧。

深度： $2370 \times 0.001 = 2.370 \text{ m}$ ；65535进入无效掩码，不能当作远处自由空间。

时延与目标板：100ms推理约10Hz；核对缓存策略并优化整条链。目标板检查构建、依赖、新数据、时延与响应。





8.7 关键工具与框架接口



本节小结

工具选择必须服从任务和接口

工具与模型：核心工具、MATLAB工具箱与开源算法各有职责；按输入输出确定连接方式。

系统与证据：SITL检查逻辑，HITL加入真实机载板与飞控；分布式增加网络和多机时序。

接口与易错点：下标不是SeqID，BGR不是RGB；深度先屏蔽无效值再换米。同轮读取不等于同步；关窗不等于载具结束动作。

下一步：课件8.8按教材实验选择任务，用观察量、结果判据与真机差异检验实现。

