

学号： 2015301200070

密级：

# 武汉大学本科毕业论文

## 基于无人机-车的空-地联合目标搜索

院(系)名称： 电子信息学院

专业名称： 电子信息工程

学生姓名： 李皓

指导教师： 杨文 教授

二〇一九年五月

**BACHELOR'S DEGREE THESIS  
OF WUHAN UNIVERSITY**

**Air-Ground Joint Target Search based on  
UAV and UGV**

School (Department) : School of Electronic Information  
Major : Electronic Information Engineering  
Candidate : Li Hao  
Supervisor : Prof. Yang Wen



Wuhan University

May 2019

# 郑重声明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名: \_\_\_\_\_

日期: \_\_\_\_\_

# 摘 要

随着机器人技术的迅速发展，相较于传统的单个机器人的单独作业，多机器人的协作显现出越来越大的优势，因而引起了极大的关注。无人机和无人车是两种使用最多、技术最为成熟的机器人，无人机具有高机动性和全局视觉，但续航能力差、载重量小；无人车续航能力强，载重量大，但只具有局部视觉。当二者联合工作时，无人机广阔的环境感知能力可以为无人车提供导航信息，无人车具有的局部视觉可以使其更好地执行任务，二者的结合可以提高任务的完成度，因此无人机和无人车的联合工作在目标搜索方面具有很大优势。本项目利用无人机进行目标的检测和定位，再利用无人车通过自主导航行使到目标处，实现了一套基于无人机-车的空地联合目标搜索系统。本文的主要工作如下：

- (1) 无人机系统实现。在硬件方面，本文设计了一套基于大疆 M100 无人机、机载电脑 Manifold、云台相机、GPS 模块、蓝牙串口模块和遥控器的无人机开发平台；在软件方面，本文利用基准标记作为搜索目标，实现了目标的快速检测，利用 PnP (Perspective-n-Point) 算法实现了目标定位，利用大疆的 Onboard SDK 实现了无人机的自主飞行；
- (2) 无人车系统实现。在硬件方面，本文设计了一套基于 Robomasters 步兵车底盘、STM32 控制主板、TX2 机载电脑、激光雷达、姿态传感器、GPS 模块、蓝牙串口模块和遥控装置的无人车开发平台；在软件方面，本文利用麦克纳姆四轮编码器和姿态传感器实现了一种低成本的里程计，结合建图和导航算法，实现了无人车的自主定位、建图、避障和导航；
- (3) 空地联合目标搜索实现。本文先利用无人机扫描搜索区域，进行目标检测和定位，然后通过蓝牙串口模块将目标位置信息发送给无人车，最后无人车通过自主导航行驶到目标处，完成目标搜索任务。

实验表明，本设计中的无人机系统和无人车系统具备了预期的功能，并且整个系统具备了空地联合目标搜索的能力。

**关键词：**无人机；无人车；基准标记；目标检测；目标定位；建图；导航

# ABSTRACT

With the rapid development of robotics technology, the cooperation of multi-robots has shown more and more advantages than traditional separate operation of single robot, which has attracted great attention. Unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) are two kinds of most used and most mature robots. UAV has high mobility and global vision, but its endurance is poor and its load is small. UGV has strong endurance and heavy load, but only has local vision. When they work together, UAV's wide environmental awareness can provide navigation information for UAV and UAV's local vision can make it perform tasks better. The combination of UAV and UGV can improve the completion of tasks. Therefore, the joint work of UAV and UGV has great advantages in target search. In this project, using UAV to detect and locate the target, and using UGV to move to the target through autonomous navigation, an air-ground joint target search system based on UAV-UGV is realized. The main work of this thesis is as follows:

- (1) An UAV system has been realized. In terms of hardware, this thesis designs an UAV development platform based on DJI M100, Manifold, gimbal camera, GPS module, Bluetooth serial port module and remote control. In terms of software, this thesis uses fiducial marker as detection target to achieve rapid target detection, uses Perspective-n-Point (PnP) algorithm to achieve target location, and uses DJI On-board SDK to achieve autonomous flight of UAV.
- (2) An UGV system has been implemented. In terms of hardware, this thesis designs an UGV development platform based on Robomaster infantry vehicle chassis, STM32 main control board, TX2 on-board computer, LiDAR, attitude sensor, GPS module, Bluetooth serial port module and remote control device. In terms of software, a low-cost odometry is realized by using Mecanum wheel encoder and attitude sensor. Combining the mapping and navigation algorithm, autonomous location, mapping, obstacle avoidance and navigation of UGV are realized.
- (3) The air-ground joint target search has been realized. In this thesis, the UAV scans the search area, detects and locates the target, then transmits the location information of the target to the UGV through bluetooth serial port module. Finally the UGV moves

to the target through autonomous navigation to complete the target search task.

Experiments show that the UAV system and UGV system in this design possess the expected functions, and the whole system has the ability of space-ground joint target search.

**Key words:** UAV; UGV; fiducial marker; target detection; target location; mapping; navigation

# 目 录

摘要 .....	I
ABSTRACT .....	II
1 绪论 .....	1
1.1 研究背景与意义 .....	1
1.2 国内外研究现状 .....	2
1.3 研究内容与结构安排 .....	3
1.4 本章小结 .....	4
2 无人机系统 .....	5
2.1 系统硬件模块 .....	5
2.1.1 整体硬件框架 .....	5
2.1.2 飞控模块 .....	7
2.1.3 机载电脑 .....	7
2.1.4 云台相机 .....	8
2.1.5 GPS 模块 .....	9
2.1.6 姿态传感器 .....	9
2.1.7 蓝牙串口模块 .....	10
2.1.8 遥控器 .....	10
2.2 目标检测算法 .....	11
2.2.1 基准标记 .....	11
2.2.2 AprilTag .....	12
2.2.3 ChromaTag .....	15
2.2.4 ArUco .....	18
2.3 目标定位算法 .....	20
2.3.1 相机成像模型 .....	20
2.3.2 PnP 算法 .....	22

2.3.3	GPS 坐标计算 .....	23
2.4	无人机自主飞行 .....	24
2.5	系统软件设计 .....	25
2.6	本章小结 .....	25
3	无人机系统 .....	27
3.1	系统硬件模块 .....	27
3.1.1	整体硬件框架 .....	27
3.1.2	控制主板 .....	29
3.1.3	机载电脑 .....	30
3.1.4	激光雷达 .....	31
3.1.5	GPS 模块 .....	32
3.1.6	姿态传感器 .....	32
3.1.7	编码器 .....	33
3.1.8	蓝牙串口模块 .....	33
3.1.9	遥控装置 .....	33
3.2	自主定位 .....	34
3.3	SLAM 建图 .....	35
3.4	自主导航 .....	35
3.5	系统软件设计 .....	37
3.6	本章小结 .....	38
4	系统测试与结果分析 .....	39
4.1	无人机系统测试 .....	39
4.1.1	目标检测测试 .....	39
4.1.2	目标定位测试 .....	41
4.1.3	自主飞行测试 .....	42
4.2	无人车系统测试 .....	42
4.2.1	自主定位测试 .....	42



4.2.2	SLAM 建图测试 .....	43
4.2.3	自主导航测试 .....	43
4.3	系统整体测试 .....	45
4.4	本章小结 .....	46
5	总结与展望 .....	47
5.1	工作总结 .....	47
5.2	存在的问题与展望 .....	47
	参考文献 .....	49
	致谢 .....	54

# 插图

1.1 系统整体框架 .....	3
2.1 无人机系统实物图 .....	6
2.2 无人机系统整体硬件框架 .....	7
2.3 机载电脑 Manifold 妙算 <sup>[1]</sup> .....	8
2.4 DJI ZENMUSE X3 云台相机.....	9
2.5 GPS 模块 .....	9
2.6 HC-05 蓝牙串口模块.....	10
2.7 DJI C1 遥控器 .....	11
2.8 部分基准标记 .....	12
2.9 不同型号的 AprilTag .....	13
2.10 AprilTag 线段检测效果 <sup>[2]</sup> .....	14
2.11 AprilTag 的 LAB 图像.....	16
2.12 LAB 色彩空间 <sup>[3]</sup> .....	16
2.13 AprilTag 生成 ChromaTag 过程 .....	17
2.14 ChromaTag 检测流程 <sup>[3]</sup> .....	18
2.15 不同型号的 ArUco 标记 .....	19
2.16 针孔相机模型 <sup>[4]</sup> .....	20
2.17 局部坐标系与机体坐标系 .....	23
2.18 区域扫描示意图.....	25
2.19 无人机系统软件流程图 .....	26
3.1 无人车系统实物图 .....	28
3.2 无人车系统整体硬件框架 .....	29
3.3 RM 开发板 <sup>[1]</sup> .....	30
3.4 机载电脑 Jetson TX2 .....	31

3.5	Velodyne VLP-16 3D 激光雷达 .....	31
3.6	GT-U7 GPS 模块 .....	32
3.7	MPU9250 姿态传感器.....	32
3.8	DR16 接收机和 DT7 遥控器 .....	33
3.9	局部坐标系与车身坐标系 .....	34
3.10	navigation 导航包整体框架 .....	37
3.11	无人车系统软件流程图 .....	38
4.1	AprilTag 检测结果.....	39
4.2	ChromaTag 检测结果.....	39
4.3	ArUco 标记检测结果.....	40
4.4	ID 为 17 的 4×4 ArUco 标记 .....	40
4.5	目标定位测试图像 .....	41
4.6	无人机自主飞行测试结果 .....	42
4.7	无人车运动轨迹.....	43
4.8	SLAM 建图结果.....	43
4.9	自主导航测试场地 .....	44
4.10	自主导航测试效果 .....	45
4.11	系统整体测试测试场地 .....	45
4.12	系统整体测试测试效果 .....	46

# 表格

4.1	基准标记检测的耗时情况测试结果 .....	40
4.2	目标定位测试结果 .....	42

# 1 绪论

## 1.1 研究背景与意义

如今，机器人技术不断成熟，机器人在许多领域都发挥着越来越重要的作用。随着计算机视觉和深度学习的迅速发展，目标检测、目标跟踪等算法也越来越成熟，使得机器人的应用领域更加广泛。TX2、Manifold 妙算等高性能机载电脑的出现，打破了机器人计算性能的壁垒，使得机器人可以完成更加复杂的任务。同时，随着视觉传感器、激光雷达、惯性测量单元等传感器的不断发展，以及算法的不断提出和改进，让机器人的自主定位、实时建图、自主导航等功能都变得可行、可靠。所有的这些进步，让机器人具有了更加强大的功能和更为广阔的应用，例如未知环境的探索，危险环境下的自主作业等。

同时，相较于传统的单个机器人的单独作业，多机器人的协作也显现出越来越大的优势。单机器人系统实现简单，但所有的机器人都有其优势也有其弊端，因此单机器人系统的应用是受限的；而多机器人系统虽然较为复杂，但是有高性能机载电脑的支持，复杂算法已不是问题，多机器人系统通过机器人之间的协调合作可以进行优势互补，使得整个系统性能最大化，这是单机器人系统不具备的优势。目前，无人机和无人车是两种使用最为广泛、技术最为成熟的机器人，无人机具有高机动性和全局视觉，适用于侦察、检测等任务，但其续航能力弱，一般只能工作十几分钟，同时其载重量小，不适合挂载较多传感器和运输物体；无人车续航能力强，载重量大，可以执行抓取、运输等任务，但只具有局部视觉。当二者联合工作时，利用无人机广阔的环境感知能力可以为无人车提供全局的环境信息和导航信息，无人车具有的局部视觉可以使其更好地执行任务，二者的结合可以提高任务的完成度，因此无人机和无人车的联合工作在目标搜索方面具有很大的优势。

空地联合目标搜索也具有广泛的用途。在民用方面，可用于救援减灾、物流配送、城市垃圾清理等，既可以解放人力，又可以提高工作效率和质量；在军用方面，可以使用空地机器人联合执行侦察等任务，即保障士兵的安全，又保障任务的可靠完成；空地联合目标搜索也很容易扩展为其他应用，例如未知环境的快速探索和 3D 建图等。

## 1.2 国内外研究现状

目前在空地机器人联合目标搜索方面的研究在国际上较为深入, 瑞士的苏黎世联邦理工学院 (ETH Zurich) 的 Autonomous Systems Lab (ASL) 和 Robotic Systems Lab (RSL) 共同实现了一种基于无人机和足式机器人的协同导航系统<sup>[5]</sup>, 该系统先利用无人机扫描未知区域, 通过 SLAM (simultaneous localization and mapping) 建立全局定位地图和半稠密点云地图, 分别用于后续足式机器人的定位和导航。然后足式机器人利用视觉惯性里程计 (VIO) 和全局定位地图进行自身的定位, 同时利用激光雷达的点云数据更新全局点云地图, 并利用路径规划算法生成导航路径, 指引足式机器人前往设置的目标处。该系统最大的两个优点是不需要外部的定位信息, 例如 GPS 信号, 也不需要无人机与地面机器人的相对位置信息, 即无需无人机跟随地面机器人, 因为该系统拥有全局定位地图, 因此该系统的应用范围较广。但该系统对算法的精度要求非常高, 因此算法较为复杂。

在国内, 香港科技大学实现了一种复杂环境下的空地协同目标搜索系统<sup>[6]</sup>, 该系统利用 ArUco 标记作为搜索目标, 以无人车的初始位置为坐标原点, 无人车初始朝向为  $x$  轴, 建立全局坐标系。该系统首先利用无人机扫描周围环境, 进行目标检测和目标在全局坐标系下的定位。扫描结束后无人机将目标位置信息发送给无人车, 然后开始跟随无人车, 同时无人车启动。无人车行使过程中, 无人机和无人车分别利用传感器测量自身的运动速度, 同时无人机利用单目相机检测无人车和目标, 一旦检测到无人机或目标, 就通过 PnP 算法计算无人机与无人车或无人机与目标的相对位置信息, 然后利用速度信息和相对位置信息通过 EKF (Extended Kalman Filter) 估计无人机和无人车的全局坐标, 实现无人机与无人车的定位。同时利用激光雷达的点云数据和路径规划算法进行无人车的动态避障和导航。该算法也不需要 GPS 等外部定位信息, 因此可以在室内、隧道等弱 GPS 或无 GPS 信号的环境下工作。但该系统要求无人机跟随无人车, 同时只使用 EKF 实现无人机和无人车的定位会有累积误差。

此外, 也有一些利用 GPS 信号进行定位的系统<sup>[7, 8]</sup>。同时在空地联合环境探索方面国内外也有深入研究<sup>[9, 10, 11, 12, 13]</sup>, 其方案也可以借鉴和参考。

### 1.3 研究内容与结构安排

参考国内外研究现状，本设计选择使用基准标记作为搜索目标，以实现目标的简单、快速检测和定位，同时利用 GPS 实现无人机与目标的定位，本项目的系统整体框架如图 1.1所示。

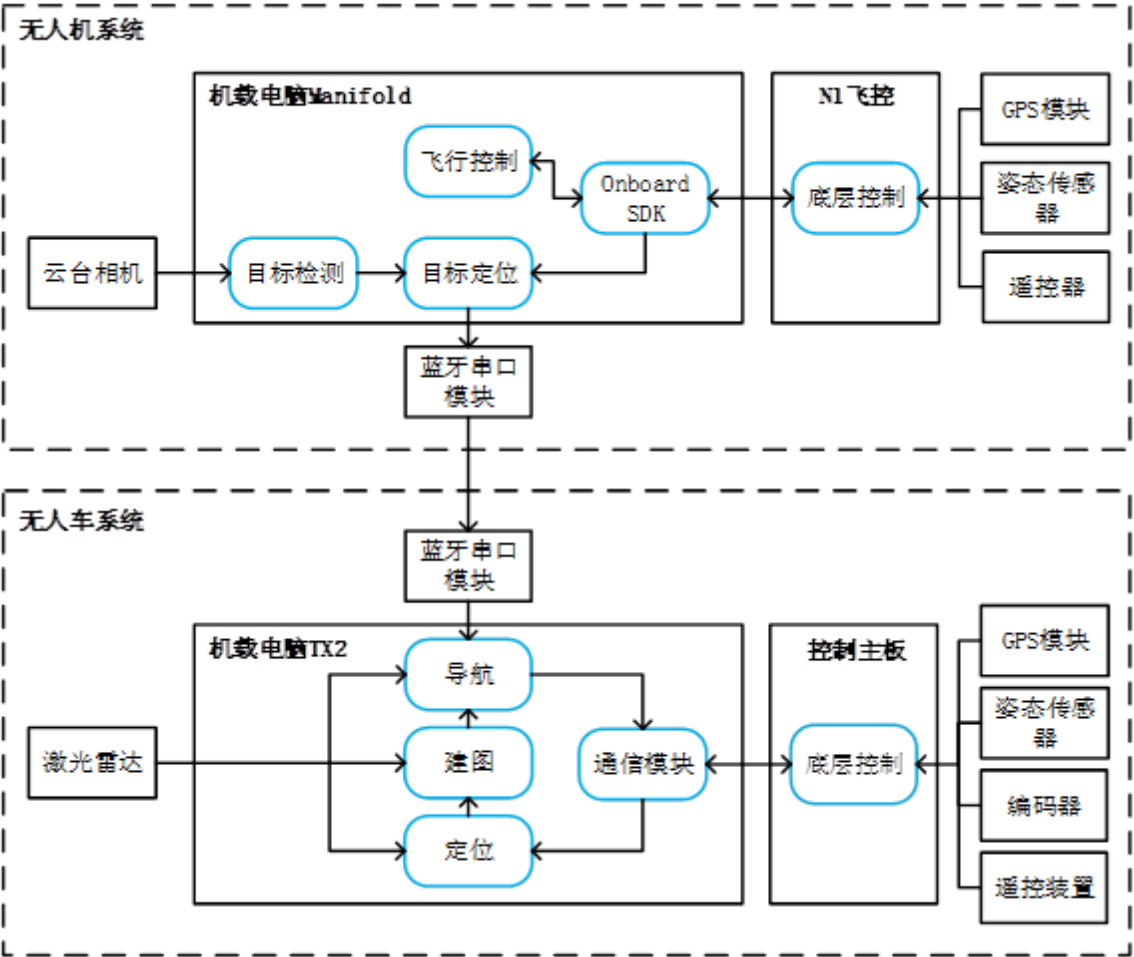


图 1.1 系统整体框架

为了实现图 1.1所示的基于无人机-车的空地联合目标搜索系统，本项目主要研究了以下内容：

- (1) 基准标记的检测与定位算法；
- (2) 无人机的自主飞行；
- (3) 无人车的运动控制，以及自主定位、建图和导航。

本文结构安排如下：

第一章为绪论。主要介绍了基于无人机-车的空地联合目标搜索的研究背景与

意义，国内外研究现状，以及本文的研究内容与结构安排。

第二章是无人机系统。首先介绍了无人机系统的硬件模块。接着详细介绍了 AprilTag、ChromaTag 和 ArUco 三种基准标记及其检测算法。然后分析了相机的成像模型，介绍了 PnP 算法和目标 GPS 坐标计算方法。最后介绍了无人机系统的软件设计。

第三章是无人车系统。首先介绍了无人车系统的硬件模块，然后详细介绍了无人车自主定位、SLAM 建图和自主导航的实现方法，最后介绍了无人车系统的软件设计。

第四章是系统测试与结果分析。分别做了无人机系统的目标检测测试、目标定位测试和自主飞行测试，无人车系统的自主定位测试、SLAM 建图测试和自主导航测试，以及系统整体测试，并进行了结果分析。

第五章是总结与展望。对目前完成的工作进行了总结，并分析了系统的不足以及尚未解决的问题，并对后续的改进和完善进行了展望。

## 1.4 本章小结

本章为全文绪论，首先介绍了基于无人机-车的空地联合目标搜索的研究背景与意义，然后分析了近年来国内外研究现状，大体介绍了几种系统设计方案，最后对论文的研究内容与结构安排做了简要描述。



## 2 无人机系统

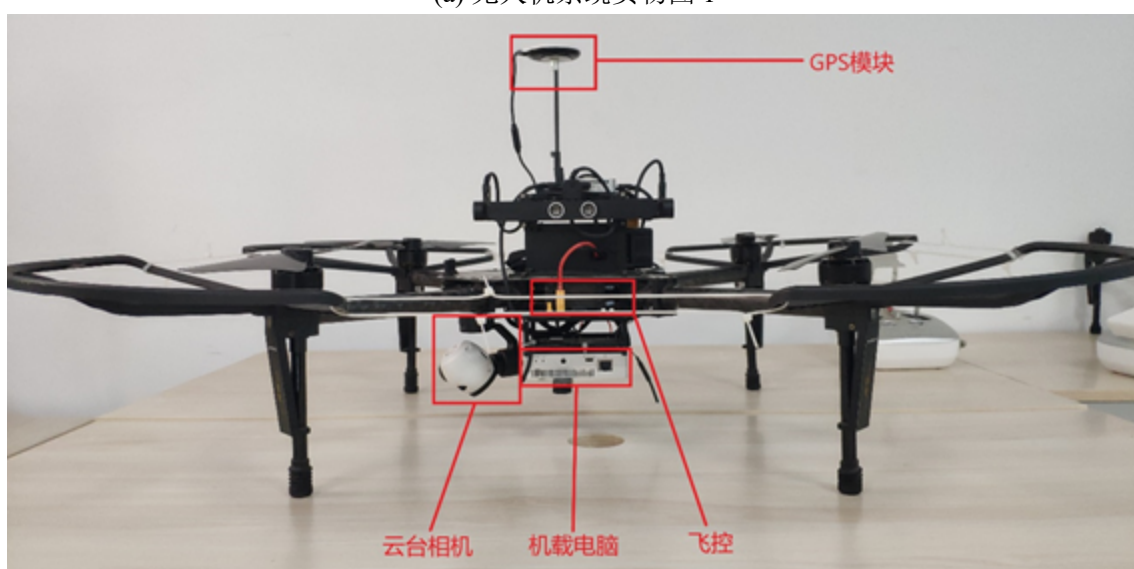
### 2.1 系统硬件模块

#### 2.1.1 整体硬件框架

无人机系统是在 DJI 经纬 Matrice 100 (M100) 无人机基础上搭建的，系统的硬件由飞控、机载电脑、传感器和外设四部分组成。其中飞控为 DJI IN 飞控，主要用来实现对无人机的直接控制，包含核心处理器 (执行无人机的底层飞行控制操作)、串行通讯接口 (与机载电脑通讯)、CAN 通讯接口 (与 GPS 模块、电机集成电调模块等通讯)、RF 通讯接口 (与 DJI Lightbridge 图传模块通讯)、多路 PWM 信号输出 (用于控制旋翼电机转速) 等；机载电脑是 DJI Manifold 妙算，主要用来执行目标检测与定位任务，以及控制无人机的自主飞行，Manifold 妙算包含 Tegra K1 核心处理器 (执行复杂算法和任务)、CAN 通讯接口 (接收云台相机采集的视频数据)、串行通讯接口 (与飞控通讯)、USB 接口 (用于连接蓝牙串口模块和鼠标、键盘等设备) 等；传感器包括云台相机 (用于目标检测)、GPS 模块 (测量无人机的位置)、姿态传感器 (测量无人机的姿态) 等；外设包括蓝牙串口模块 (与无人车通讯)、遥控器 (用于设置无人机控制模式以及手动控制无人机) 等。图 2.1 为无人机系统的实物图，图 2.2 为系统的整体硬件框架。



(a) 无人机系统实物图 1



(b) 无人机系统实物图 2

图 2.1 无人机系统实物图

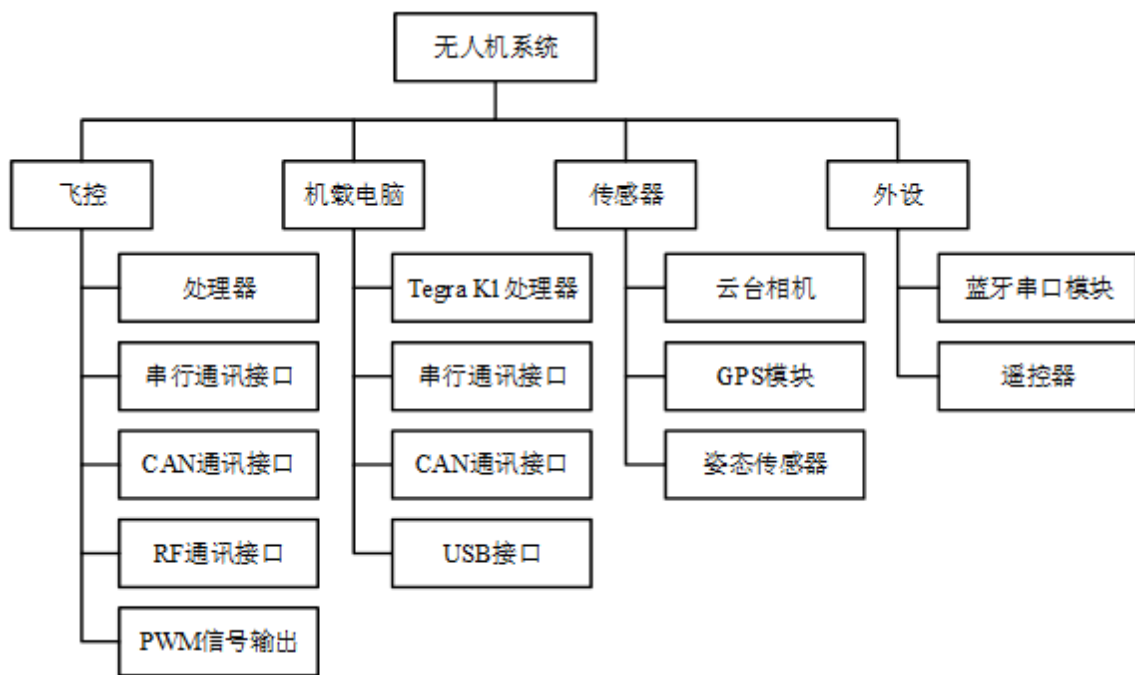


图 2.2 无人机系统整体硬件框架

### 2.1.2 飞控模块

飞控，即飞行控制系统，是无人机的核心部件，其功能是接收 GPS、IMU、气压计、磁力计等传感器数据，利用复杂算法计算得到飞行控制指令，并发送给执行机构，从而实现无人机的飞行、悬停、姿态控制、导航等任务。

本设计采用 DJI M100 自带的 N1 飞控，N1 飞控具有板载的 IMU、气压计、磁力计等传感器，可用于测量无人机的高度和姿态。同时 DJI N1 飞控还搭载丰富的硬件接口，包含 CAN 通讯接口，用于与电调模块通讯，以及接收 GPS 数据；串行通讯接口，用于与机载电脑通讯；RF 通讯接口，用于与 DJI 遥控器自带的 Lightbridge 图传模块通讯，传输云台相机拍摄的图像；Micro USB 接口，用于 PC 端的飞行器仿真与飞控调参；PWM 信号输出接口，用于控制旋翼电机转速。

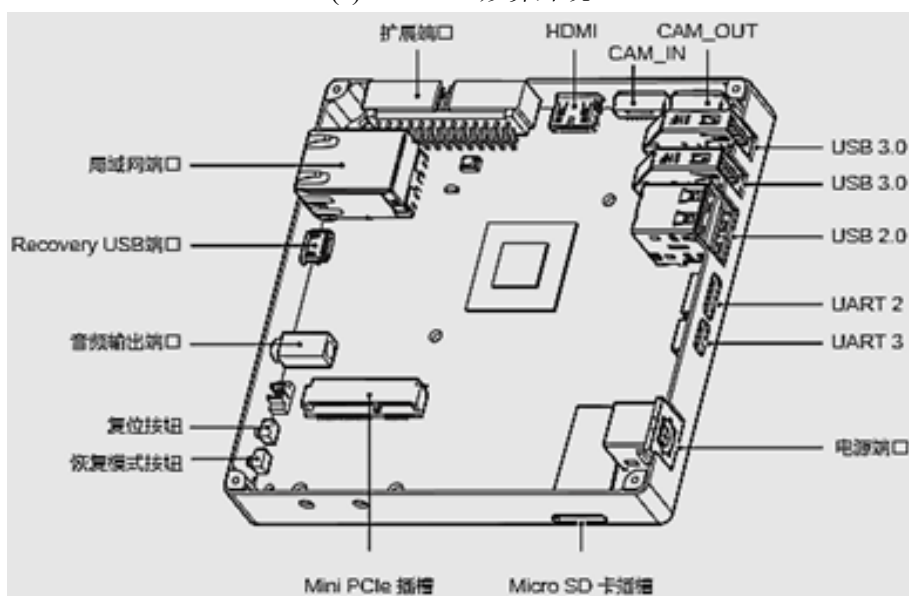
### 2.1.3 机载电脑

机载电脑采用 DJI 推出的 Manifold 妙算，该机载电脑外型小巧，具有 USB、Ethernet、HDMI、UART 等多种接口，可连接鼠标、键盘、显示屏等扩展设备。

本设计中 Manifold 妙算主要用于控制无人机自主飞行，同时利用云台相机拍摄的图像进行目标的检测和定位，并通过蓝牙串口模块把目标的经纬度坐标数据发送给无人车。图 2.3 为机载电脑 Manifold 妙算的外观和主要接口图。



(a) Manifold 妙算外观



(b) Manifold 妙算主要接口

图 2.3 机载电脑 Manifold 妙算<sup>[1]</sup>

#### 2.1.4 云台相机

云台相机采用 DJI ZENMUSE X3 一体化云台相机，该云台相机配置 9 组 9 片镜头，含 2 片非球面透镜，支持最高 4K 视频录制和 1200 万像素静态照片拍摄。三轴云台系统可实现 360 度无死角拍摄，云台可控转动范围，俯仰角为：+30° 至 -90°，翻滚角为：-90° 至 +60°，偏航角为：±320°。

该云台相机利用 CAN 总线传输视频信号，可以接入到 N1 飞控，再利用图传将视频信号传输到遥控器端，并利用手机等移动设备显示。也可以接入到机载电脑 Manifold 妙算，用于进行目标检测等操作。本系统中，首先将云台相机采集的视频信号通过 CAN\_IN 接口传输给 Manifold 妙算，Manifold 妙算再通过 CAN\_OUT 接口传输给 N1 飞控，然后通过图传发送给遥控器，并通过手机显示。

图 2.4为 DJI ZENMUSE X3 云台相机的外观图。



图 2.4 DJI ZENMUSE X3 云台相机

#### 2.1.5 GPS 模块

GPS 模块采用 DJI M100 无人机配套的 GPS 模块，定位精度可以达到米级。GPS 模块通过 CAN 通讯接口连接到 N1 飞控，GPS 信息将用于无人机的定位和导航，以及目标经纬度坐标的计算。图 2.5为 GPS 模块的外观图。



图 2.5 GPS 模块

#### 2.1.6 姿态传感器

姿态传感器采用 N1 飞控搭载的 IMU+ 磁力计组合，通过 IMU 可以获得无人机的 3 轴加速度和 3 轴角速度，通过磁力计可以获得无人机所在位置的 3 轴磁感应强度，将加速度和磁感应强度进行融合可以求解出无人机的 3 轴姿态角 (翻滚角  $roll$ 、俯仰角  $pitch$ 、偏航角  $yaw$ )。同时 IMU 数据也可以与 GPS 数据融合，用于提

高无人机定位精度。本系统需要使用偏航角  $yaw$  进行无人机机体坐标系与局部坐标系的变换。

### 2.1.7 蓝牙串口模块

蓝牙串口模块选用的是 HC-05 主从一体蓝牙串口模块，这是一款低成本的蓝牙串口模块，兼容各种主流的串口波特率，可以直接与微控制器的 UART 接口相连，也可以结合 USB 转 TTL 芯片与 PC 或机载电脑相连。

本设计中无人机和无人车的机载电脑均搭载了蓝牙串口模块，一端作为主机，另一端作为从机，可以非常简便地实现无人机与无人车之间的无线数据传输。本设计中蓝牙串口模块用于传输目标的经纬度坐标数据。图 2.6 为蓝牙串口模块的实物图。



图 2.6 HC-05 蓝牙串口模块

### 2.1.8 遥控器

遥控器使用 DJI M100 配套的 DJI C1 遥控器，该遥控器工作于 2.4GHz 频段，具有长达 5 千米的通信距离，并且集成了 Lightbridge 图传，可以直接输出航拍图像到手机等移动设备。

该遥控器可用于手动控制无人机，包括水平、垂直飞行以及旋转等操作，同时整合了相机操作以及云台操作的功能按键，以便用户在飞行时更轻松自如地航拍。当飞行模式切换到 F 挡时，允许利用机载电脑 Manifold 妙算控制无人机自主飞行。图 2.7 为该遥控器的实物图。



图 2.7 DJI C1 遥控器

## 2.2 目标检测算法

### 2.2.1 基准标记

基准标记是一种用于快速检测的人工视觉特征图案，通常携带独一无二的编码信息用于区分不同的标记，并有相应的检测算法。基准标记类似于二维码，但降低了复杂度以满足实时性要求，其检测具有简单、快速、鲁棒、精确度高等特点。

基准标记最早在增强现实 (augmented reality, 简称 AR) 领域发展并流行起来，现已广泛用于机器人领域。基准标记具有目标检测识别、位姿估计、相机校准等功能。将基准标记贴附于目标物体表面，可以实现目标的快速检测和定位，同时利用基准标记独一无二的编码信息，还可以实现目标的识别和跟踪；利用基准标记作为路标，可用于机器人的定位和导航，以及作为 SLAM 等算法测试的 ground truth。基准标记的检测一般分为 3 个步骤：标记的初检测；标记的精确定位；ID 译码验证标记。

常用的基准标记有圆形基准标记和方形基准标记。圆形基准标记有 CCC (concentric contrasting circle)<sup>[14]</sup>、RuneTag<sup>[15, 16]</sup>、CCTag<sup>[17, 18]</sup> 等，方形基准标记有 ARTag<sup>[19]</sup>、AprilTag<sup>[2, 20]</sup>、ArUco<sup>[21]</sup> 等。多数基准标记都是黑白的，但也有少数彩色基准标记，例如 Y. Cho 等人改进的 CCC 标记<sup>[22]</sup>、ChromaTag<sup>[3]</sup> 等。图 2.8展示了部分基准标记。



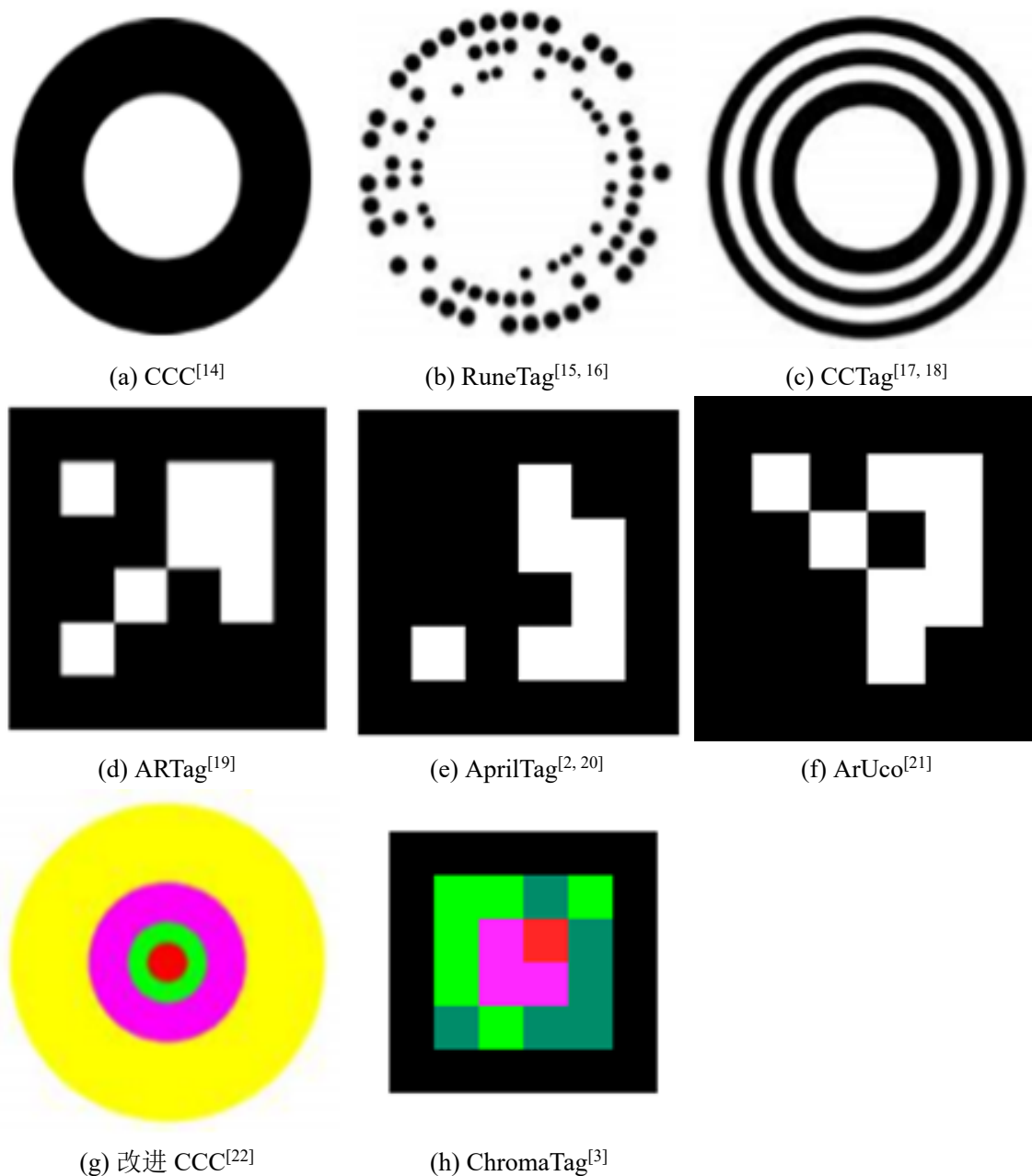


图 2.8 部分基准标记

### 2.2.2 AprilTag

AprilTag 是由 Edwin Olson 于 2011 年在 ARTag 基础上重新设计和实现的一种二进制方形基准标记<sup>[2]</sup>。通过使用快速、鲁棒的线段检测算法，以及性能更好的数字编码算法，相较于之前的 ARTag 等基准标记，AprilTag 具有更好的性能，包括更快的检测速度和更好的鲁棒性。图 2.9 为三种不同型号的 AprilTag。



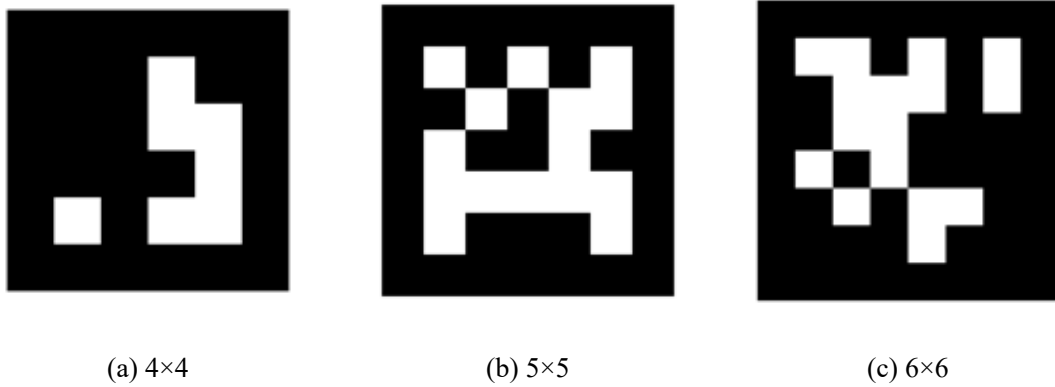


图 2.9 不同型号的 AprilTag

AprilTag 的检测算法分为以下 4 个步骤：

- (1) 线段检测。首先计算每个像素的梯度方向和幅度，并通过聚类算法将梯度方向和幅度相似的像素点聚在一起。聚类算法类似于 Felzenszwalb 提出的基于图的图像分割算法<sup>[23]</sup>，首先创建一个图，每个节点代表一个像素，相邻节点之间添加一条边，权值为 2 个像素点的梯度方向之差，然后对这些边按权值进行升序排序，最后对于每条边，测试相连的 2 个节点所属的 2 个区域是否可以合并。其中，边的权值以定点数的形式存储，这样可以利用线性时间计数排序 (linear-time counting sort) 算法来加速边的排序，同时区域融合操作可以使用高效的并查集 (union-find) 算法来完成。另外基于梯度的聚类方法对噪声敏感，解决方案是对图像进行低通滤波。完成聚类操作后，再利用加权最小二乘算法对直线进行拟合，权重采用像素的梯度幅度。聚类算法是整个检测算法最慢的一个环节，可以采用对图像下采样的方法进行加速；
- (2) 四边形检测。找到线段后，下一步就是四边形检测。作者采用了基于深度优先搜索 (depth-first search, 简称 DFS) 的方法，深度为 4，每一次搜索添加一条边到四边形中。第一层时，遍历所有线段；第二至四层，寻找头部与上一层线段尾部“足够接近”并满足逆时针旋转顺序的线段。“足够接近”通过与阈值比较判断，作者将这个阈值设置得比较大，以得到较小的假阴性 (false negative) 比率，但也导致了较高的假阳性 (false positive) 比率，需要利用后续的译码来拒绝假阳性。另外作者使用了二维查找表进行四边形检测算法的加速。一旦四边形的四条边被找到，4 个顶点可以通过计算线段的交点得到，由于线段是通过大量像素点拟合得到的，因此顶点的定位精度可以达到亚像素级别；

- (3) 单应性和外参估计。通过直接线性变换 (Direct Linear Transform, 简称 DLT) 算法计算出单应性矩阵, 然后利用单应性矩阵、相机内参和标记的物理尺寸计算出相机外参, 详细的算法将在后面“PnP 算法”一节介绍;
- (4) ID 译码。这一步的任务是读取标记中的二进制信息, 首先通过单应性矩阵计算出每一个携带二进制信息的黑白网格对应的像素坐标, 得到对应灰度值, 然后通过灰度阈值判断其携带的二进制信息是 0 还是 1, 最后将二进制信息与码表匹配, 验证是否为 AprilTag 基准标记, 必要时可以采用纠错技术。为了增强对光照的鲁棒性, 作者采用了随空间变化的灰度阈值, 首先利用式 (2.1) 建立黑色像素模型和白色像素模型, 然后利用黑白边界处两侧的灰度值通过最小二乘回归分别训练这两个模型, 最后取两个模型的平均值作为灰度阈值。

$$I(x, y) = Ax + Bxy + Cy + D \quad (2.1)$$

图 2.10 为线段检测效果。

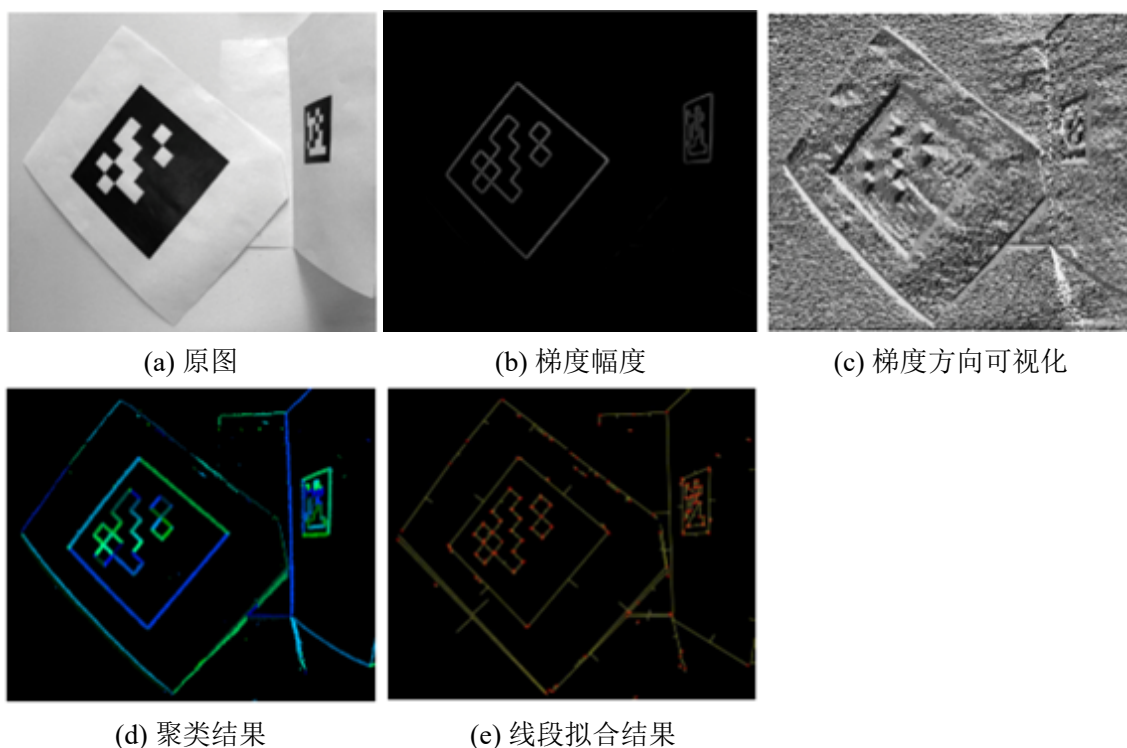


图 2.10 AprilTag 线段检测效果<sup>[2]</sup>

除了上述检测算法, 作者还设计了编码算法, 用于检错和纠错。对于编码算法, 本文不做详细介绍。

2016 年 John Wang 和 Edwin Olson 又对 AprilTag 进行了改进, 提出了 April-

Tag2<sup>[20]</sup>。相较于 AprilTag, AprilTag2 保留了编码算法, 重新设计了检测算法, 拥有了更快的检测速率和更高的准确率。

### 2.2.3 ChromaTag

之前的 AprilTag 等基准标记依赖于 ID 译码来拒绝假阳性, 导致大量时间浪费在了最终会被当作假阳性拒绝的非标记区域的检测上, 针对这一问题, Joseph DeGol 等人在 2017 年基于 AprilTag 提出了一种彩色方形基准标记: ChromaTag<sup>[3]</sup>。ChromaTag 利用彩色信息快速拒绝假阳性, 利用灰度信息实现精确定位, 因此相较于之前的基准标记, 具有显著的提速, 同时达到了相似甚至更好的检测精度。

ChromaTag 的图案设计如图 2.11a 所示, 最内层为红色 (包括 2 种不同红色), 往外是绿色 (包括 2 种不同绿色), 再往外是黑色, 最外层是白色。检测时使用 LAB 色彩空间, 图 2.11a 所示标记图案的 L、A、B 通道分别如图 2.11b、2.11c、2.11d 所示。在 L 通道下标记的黑白边界非常明显, 在 A 通道下标记的红绿边界非常明显, 在 B 通道下标记的 2 种红色和 2 种绿色差异明显。图 2.12 为 LAB 色彩空间示意图。

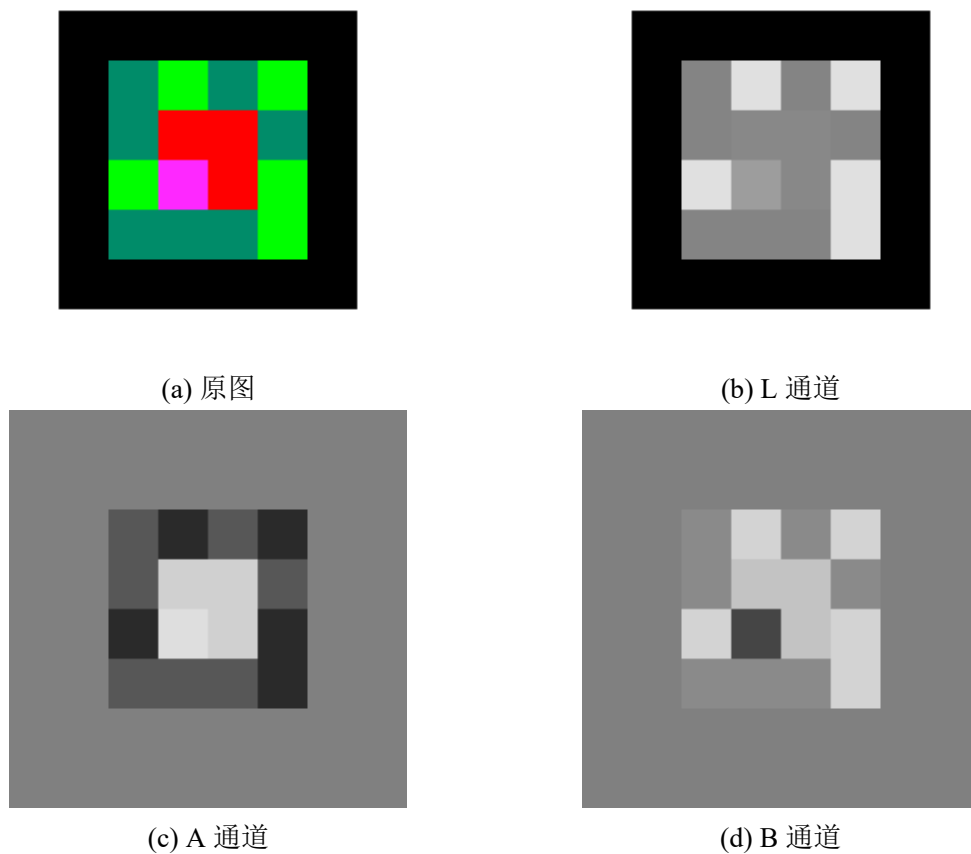


图 2.11 AprilTag 的 LAB 图像

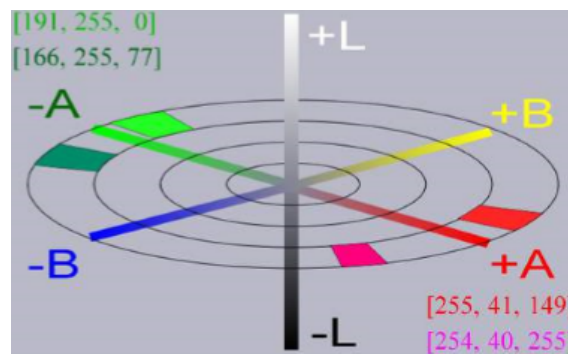


图 2.12 LAB 色彩空间<sup>[3]</sup>

根据 ChromaTag 的上述特性，可以利用 A 通道中红色和绿色的巨大差异进行标记的初步检测，利用 L 通道中黑白边界的高分辨率实现标记的精确定位，利用 B 通道中红色和绿色区域内部的差异进行 ID 编码和译码。ChromaTag 可以直接由 AprilTag 生成，图 2.13 显示了生成过程，最内层的 4 个黑白网格分别用 2 种不同红色代替，外层的 12 个黑白网格分别用 2 种不同绿色代替，即可得到 ChromaTag。ChromaTag 的编码与 AprilTag 相同。

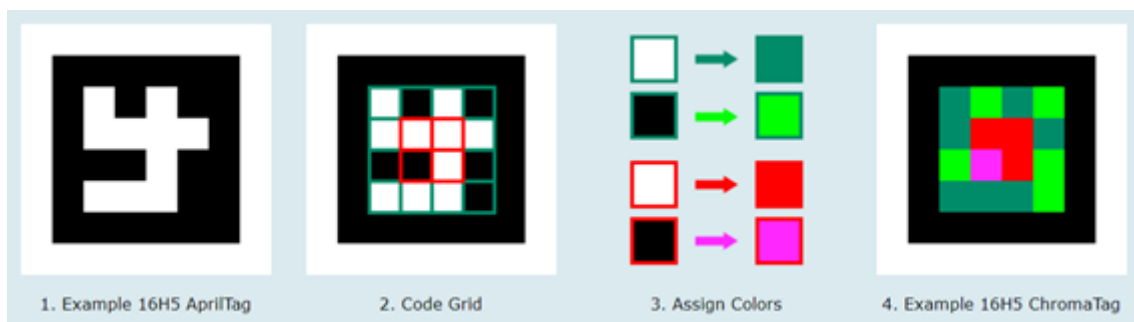


图 2.13 AprilTag 生成 ChromaTag 过程

ChromaTag 的检测算法分为 4 个步骤：

- (1) 扫描。以一定步长遍历整张图像，同时在 A 通道中做横向差分并与阈值比较，确定标记的起始中心点，然后向上下左右 4 个方向扫描，利用 A 通道的红绿差异和边界阈值确定 4 个红绿边界点，同时利用检测到的边界点坐标的均值更新标记中心点坐标，重复这个过程直到标记中心点收敛或者该点检测失败。若中心点收敛，则继续向上下左右 4 个方向扫描，利用 L 通道确定 4 个黑绿边界点和 4 个黑白边界点；
- (2) 构建多边形。利用第一步获得的 12 个边界点，构建 3 个初始多边形，这一步的任务是扩展这 3 个初始多边形，使其尽可能与标记的 3 条边界匹配。在可以使多边形面积增大最多的方向上扫描，确定下一个边界点并加入多边形中作为多边形的顶点，并通过多边形面积和相应阈值判断多边形是否收敛，重复此过程直到多边形收敛或扫描失败；
- (3) 拟合四边形。对多边形的边利用边长作为权值进行加权的 K-means<sup>[24]</sup> (K=4) 聚类，最终得到四边形的 4 条边，4 个顶点通过计算 4 条边的交点得到，再利用 Shi-Tomasi 角点检测算法<sup>[25]</sup> 实现 4 个顶点的精确定位；
- (4) ID 译码。利用黑白边界的 4 个顶点求解单应性矩阵，通过单应性矩阵计算出红色和绿色网格对应的像素坐标，得到对应的 B 通道值，然后分别利用红色和绿色网格的 B 通道阈值判断其携带的二进制信息是 0 还是 1。红色网格的 B 通道阈值为所有红色网格的 B 通道值的最大值和最小值的均值，绿色网格阈值类似。最后将二进制信息与码表匹配，验证是否为 ChromaTag 基准标记。

图 2.14 为 ChromaTag 检测流程。

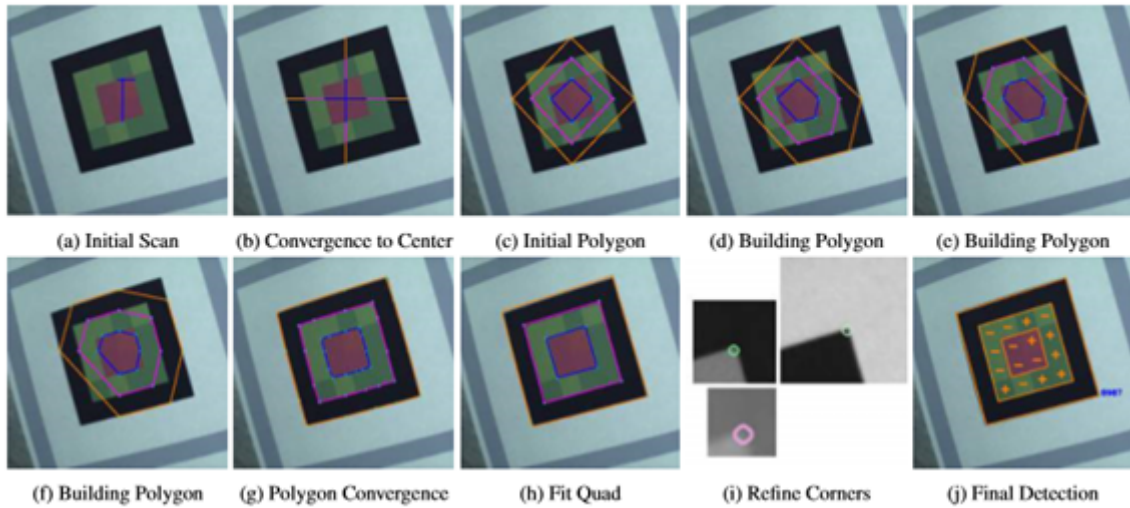


图 2.14 ChromaTag 检测流程<sup>[3]</sup>

#### 2.2.4 ArUco

对于之前的基准标记，其字典 (基准标记的集合) 的大小和标记间距 (标记二进制码之间的汉明距离) 都是固定的，因此存在两个问题，其一是在一些应用场景中所需标记数量可能超过字典大小，其二是如果所需标记数量很少，则使用一个更小的、具有更大标记间距的字典更为合适。同时之前的基准标记也不能很好地解决增强现实应用中的标记遮挡问题。

为了解决上述三个问题，S. Garrido-Jurado 和 R. Muñoz-Salinas 等人在 2014 年提出了一种新的二进制方形基准标记：ArUco<sup>[21]</sup>。首先，ArUco 标记拥有可配置字典 (字典大小和二进制码比特数可配置) 生成算法，该算法遵循最大化标记间距和有效比特数的准则；其次，基于生成的字典，ArUco 标记拥有自动检测和纠错算法；最后，ArUco 标记采用多标记组合和基于颜色的遮挡区域计算方法，解决了标记的遮挡问题。

依靠可配置字典生成算法，ArUco 标记具有很强的灵活性，可以根据需求生成对应大小的字典，同时保证最大化标记间距，既满足应用需求，又最大化检测的准确率。同时，ArUco 标记的检测和纠错算法是一个通用框架，也可用于其他的基准标记，例如 ARTag、AprilTag 等。图 2.15 为三种不同型号的 ArUco 标记。

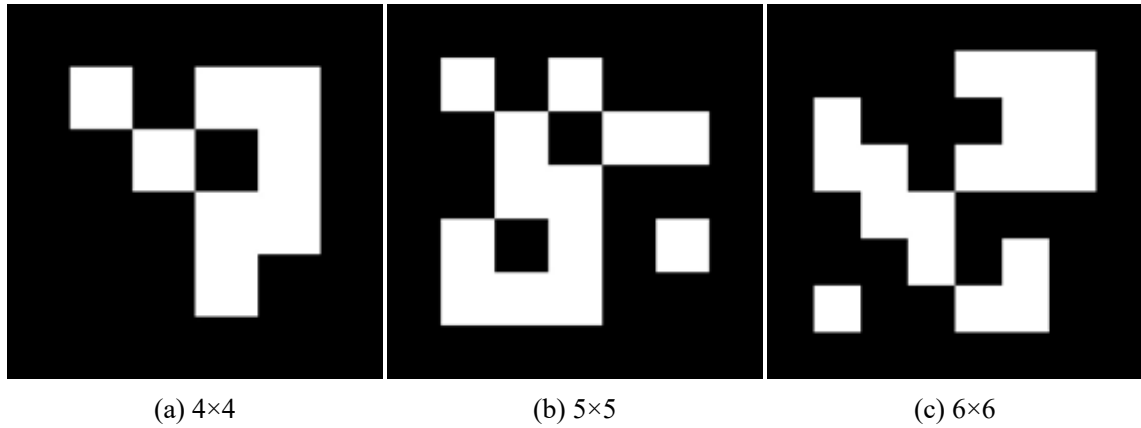


图 2.15 不同型号的 ArUco 标记

ArUco 标记的检测过程分为以下 4 步：

- (1) 图像分割。最初采用的是 Canny 边缘检测算法<sup>[26]</sup>，但不能满足实时检测的需求，后来选用了一种局部自适应阈值二值化的方法，该方法对于光照非常鲁棒；
- (2) 轮廓提取和过滤。首先，在二值图中利用 Suzuki 和 Abe 的算法<sup>[27]</sup> 提取轮廓；接着利用 Douglas 和 Peucker 的算法<sup>[28]</sup> 对轮廓进行多边形近似，由于标记是方形的，因此可以过滤掉不能近似为四边形的轮廓；最后对于具有包含关系的轮廓，只保留最外面的一个；
- (3) 标记二进制码提取。这一步是分析轮廓的内部区域来提取二进制码。首先计算单应性矩阵，并利用单应性矩阵去除相机的透视投影变换；接着利用大津法 (Otsu)<sup>[29]</sup> 对图像做二值化；然后二值图被划分为均匀的网格，网格的每个元素根据内部多数像素的值被赋值为 0 或 1；最后检测最外层网格元素是否全为 0，判断标记的黑色边界是否存在，进行标记的第一步验证；
- (4) 标记识别和纠错。一旦提取出候选标记的二进制码，就可以得到 4 种不同的 ID (对应于 4 种可能的旋转)，如果任意一种 ID 在字典中被找到，则认为候选标记有效，如果都没有找到，再使用纠错技术。

ArUco 标记的代码已经集成在 opencv 库中，同时 AprilTag 和 ChromaTag 的算法也是开源的，因此很方便进行测试，经过实验比较，本设计最终选用了性能最好的 ArUco 标记。

## 2.3 目标定位算法

### 2.3.1 相机成像模型

相机成像实际是将三维世界的点映射到二维像素平面的过程，可以使用简单的针孔模型<sup>[4]</sup>来描述。针孔模型如图 2.16 所示，其中  $O-x-y-z$  为相机坐标系， $z$  轴指向相机前方， $x$  轴向右， $y$  轴向下， $O$  为相机光心，也是针孔模型的针孔， $O'-x'-y'$  为物理成像平面，相机的焦距为  $f$ 。

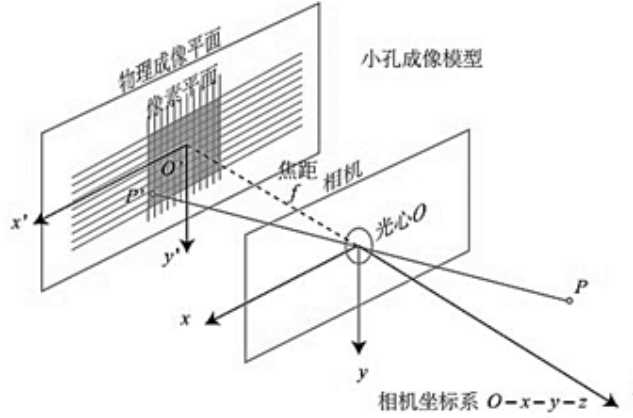


图 2.16 针孔相机模型<sup>[4]</sup>

三维世界的点  $P$  经过小孔  $O$  投影到物理成像平面上的点  $P'$ 。设在  $O-x-y-z$  坐标系下  $P$  的坐标为  $[X, Y, Z]^T$ ，在  $O'-x'-y'$  坐标系下  $P'$  坐标  $[X', Y']^T$ ，根据三角形相似有：

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'} \quad (2.2)$$

整理得到：

$$\begin{cases} X' = f \frac{X}{Z} \\ Y' = f \frac{Y}{Z} \end{cases} \quad (2.3)$$

成像之后还要进行采样和量化，才能得到像素图像。定义像素坐标系  $o-u-v$  在物理成像平面上，原点  $o$  位于图像的左上角， $u$  轴向右与  $x'$  轴平行， $v$  轴向下与  $y'$  轴平行，相对于  $O'-x'-y'$  坐标系，设原点平移了  $[c_x, c_y]^T$ ，在  $u$  轴上缩放了  $\alpha$  倍，在  $v$  轴上缩放了  $\beta$  倍。同时设  $P'$  的像素坐标  $[u, v]^T$ ，则有：



$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \quad (2.4)$$

代入式 (2.4)，并令  $f_x = \alpha f$ ， $f_y = \beta f$ ，得：

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases} \quad (2.5)$$

写成矩阵形式，得：

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \triangleq KP \quad (2.6)$$

式 (2.6) 中，把中间的矩阵称为相机的内参数矩阵 (Camera Intrinsics) $K$ 。一般认为相机的内参是固定的，可以通过标定得到。

式 (2.6) 中使用的是点  $P$  在相机坐标系下的坐标，可以由点  $P$  的世界坐标  $P_w$  通过相机的位姿变换得到，而相机的位姿可以用旋转矩阵  $R$  和平移向量  $t$  来描述，进而有：

$$ZP_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KP = K(RP_w + t) = KTP_w \quad (2.7)$$

其中  $T = [R, t]$ ，注意这里隐含了非齐次坐标到齐次坐标的转换。

式 (2.7) 描述了点  $P$  的世界坐标到像素坐标的投影关系。其中，相机的位姿  $R$  和  $t$  称为相机的外参数 (Camera Extrinsics)，相比于不变的内参，外参会随着相机的运动而变化。由于 (2.7) 使用的是齐次坐标，因此可以把  $Z$  去掉，则有：

$$P_{uv} = KTP_w \quad (2.8)$$

为了获得较好的成像效果，人们在相机的前方加入了透镜，这也导致了相机成像存在畸变，包括径向畸变和切向畸变。径向畸变可以用式 (2.9) 纠正，切向畸变可以用式 (2.10) 纠正。

$$\begin{cases} x_{corrected} = x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} = y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases} \quad (2.9)$$

$$\begin{cases} x_{corrected} = x + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{corrected} = y + 2p_2xy + p_1(r^2 + 2y^2) \end{cases} \quad (2.10)$$

其中  $[x, y]^T$  为归一化坐标,  $k$  和  $p$  称为相机的畸变系数。相机的畸变模型<sup>[4]</sup> 可以用畸变系数来描述。

### 2.3.2 PnP 算法

PnP (Perspective-n-Point) 算法<sup>[30]</sup> 是通过  $n$  对 3D 与 2D 匹配点, 利用最小化重投影误差来求解相机外参的算法, 包括很多种求解方法, 例如 P3P<sup>[31]</sup>、直接线性变换 (DLT)<sup>[32]</sup>、EPnP (Efficient PnP)<sup>[33]</sup>、UPnP<sup>[34]</sup> 等等。这里只介绍共面情况下基于单应性矩阵的求解方法。

根据前面介绍的相机成像模型, 设三维世界坐标的点为  $P_w = [X, Y, Z, 1]^T$ , 对应二维像素坐标为  $P_{uv} = [u, v, 1]^T$ , 则有:

$$P_{uv} = K [R, t] P_w \quad (2.11)$$

将世界坐标系建立在基准标记平面上, 并令基准标记平面为  $Z = 0$  的平面, 则有:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.12)$$

我们把  $K [r_1 \ r_2 \ t]$  记作单应性矩阵  $H$ , 则有:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.13)$$

$$H = [h_1 \ h_2 \ h_3] = \lambda K [r_1 \ r_2 \ t] \quad (2.14)$$

单应性矩阵  $H$  是一个  $3 \times 3$  的齐次矩阵, 所以有 8 个未知数, 而式 (2.13) 包含 2 个方程, 所以至少需要 4 对匹配点就可以计算出单应性矩阵  $H$ 。而方形基准标记正好可以提供 4 个角点, 并且通过基准标记的物理尺寸可以得到 4 个角点的世界坐标, 因此可以很方便地计算出单应性矩阵  $H$ 。

利用单应性矩阵可以计算出相机的内参和外参，这也是张正友相机标定算法<sup>[35]</sup>的原理。而在已知相机内参的情况下，利用单应性矩阵计算相机外参则非常简单，由式 (2.14) 可得：

$$\begin{cases} \lambda = \frac{1}{\|K^{-1}h_1\|} = \frac{1}{\|K^{-1}h_2\|} \\ r_1 = \frac{1}{\lambda}K^{-1}h_1 \\ r_2 = \frac{1}{\lambda}K^{-1}h_2 \\ r_3 = r_1 \times r_2 \\ t = \lambda K^{-1}h_3 \end{cases} \quad (2.15)$$

式 (2.15) 给出了外参的数值解，此外还可以通过迭代或优化的方式求解。

### 2.3.3 GPS 坐标计算

利用 PnP 算法可以求出目标 (基准标记) 在相机坐标系下的坐标，此外还要利用相机坐标系与无人机机体坐标系的几何关系，以及无人机的偏航角和 GPS 坐标计算出目标的 GPS 坐标。

建立局部坐标系和机体坐标系如图 2.17 所示。其中， $O-x-y-z$  为局部坐标系，以无人机起点为原点， $x$  轴指向北， $y$  轴指向东， $z$  轴向上，为左手系，便于与 GPS 坐标系进行坐标变化。 $O'-x'-y'-z'$  为机体坐标系，以无人机中心为原点，机身朝向为  $x'$  轴， $y'$  轴向右， $z'$  轴向上。无人机在飞行过程中可以认为保持水平，即  $O'-x'-y'$  平面与  $O-x-y$  平面平行，因此无人机的偏航角  $yaw$  就是  $x'$  轴与  $x$  轴的夹角。

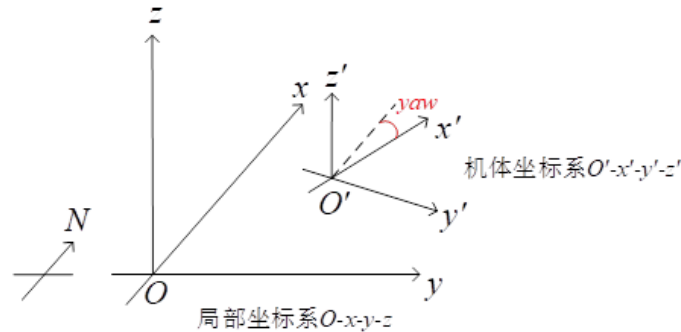


图 2.17 局部坐标系与机体坐标系

相机坐标系与机体坐标系的变换公式可以由云台相机与无人机中心的相对位

置关系和云台的角度得到，本设计中保持云台相机竖直朝下，这样坐标变换会更加简单。设无人机在局部坐标系下坐标为  $P_{lu}$ ，在 GPS 坐标系下坐标为  $P_{gu} = [Lat0, Lon0, Alt0]^T$  (Lat0 为纬度，Lon0 为经度，Alt0 为高度，经纬度采用角度单位)；目标在机体坐标系下坐标为  $P_{bt}$ ，在局部坐标系下坐标为  $P_{lt}$ ，在 GPS 坐标系下坐标为  $P_{gt} = [Lat1, Lon1, Alt1]^T$ ，则有：

$$p_{lt} = \begin{bmatrix} \cos(yaw) & -\sin(yaw) & 0 \\ \sin(yaw) & \cos(yaw) & 0 \\ 0 & 0 & 1 \end{bmatrix} P_{bt} + P_{lu} = R P_{bt} + P_{lu} \quad (2.16)$$

在很小的范围内，将 GPS 坐标系近似为平面直角坐标系，并设地球半径为  $r$ ，则有：

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = R P_{bt} \quad (2.17)$$

$$\begin{cases} \Delta Lat \approx \frac{\Delta X}{r} \cdot \frac{180}{\pi} \\ \Delta Lon \approx \frac{\Delta Y}{r \cos(Lat0 \cdot \frac{\pi}{180})} \cdot \frac{180}{\pi} \\ \Delta Alt \approx \Delta Z \end{cases} \quad (2.18)$$

$$P_{gt} = P_{gu} + \begin{bmatrix} \Delta Lat \\ \Delta Lon \\ \Delta Alt \end{bmatrix} \quad (2.19)$$

利用式 (2.16)~ (2.19) 即可求出目标的 GPS 坐标。

## 2.4 无人机自主飞行

无人机的自主飞行利用 DJI 提供的 Onboard SDK 来实现。飞行任务分为以下 3 个步骤：

- (1) 起飞。无人机垂直起飞到 1.2m 左右；
- (2) 区域扫描。区域扫描采用设置航点任务的方式实现，航点坐标采用 GPS 坐标，可以利用式 (2.16)~ (2.19) 求得，航点高度为 3m。图 2.18 为区域扫描示意图，无人机经过各个航点的顺序为  $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow \cdots \rightarrow P_{N-2} \rightarrow P_{N-1} \rightarrow P_N \rightarrow P_0$ ，起点和终点均为  $P_0$ ，起点的  $P_0$  用于让无人机上升到 3m，终点的  $P_0$  用于让无人机降落到起飞点；

(3) 降落。完成扫描任务后降落到起飞点。

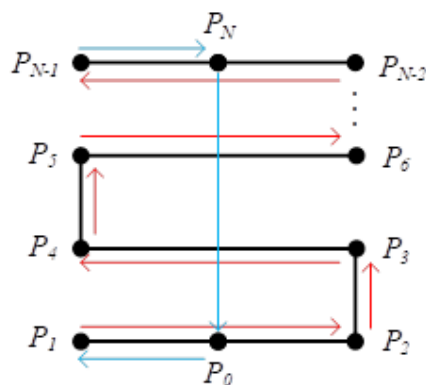


图 2.18 区域扫描示意图

## 2.5 系统软件设计

无人机系统的机载电脑 Manifold 妙算使用 Ubuntu 14.04 操作系统,并装有 ROS (Robot Operating System, 机器人操作系统),本项目利用 opencv 库实现了目标的检测与定位,利用 ROS 和 DJI 的 Onboard SDK 实现了无人机的自主飞行。图 2.19 为无人机系统的软件流程图,系统首先设置云台角度使得云台相机竖直朝下,接着让无人机起飞,然后开始扫描待搜索区域,同时进行目标检测和定位,扫描结束后让无人机降落到初始位置,最后发送目标位置信息给无人车。

## 2.6 本章小结

本章首先介绍了无人机系统的硬件框架和模块,接着利用基准标记作为搜索目标,介绍了 AprilTag、ChromaTag 和 ArUco 三种基准标记的检测算法,以及基于 PnP 算法的基准标记定位算法,然后简要描述了如何实现无人机的自主飞行,最后介绍了系统的软件设计。

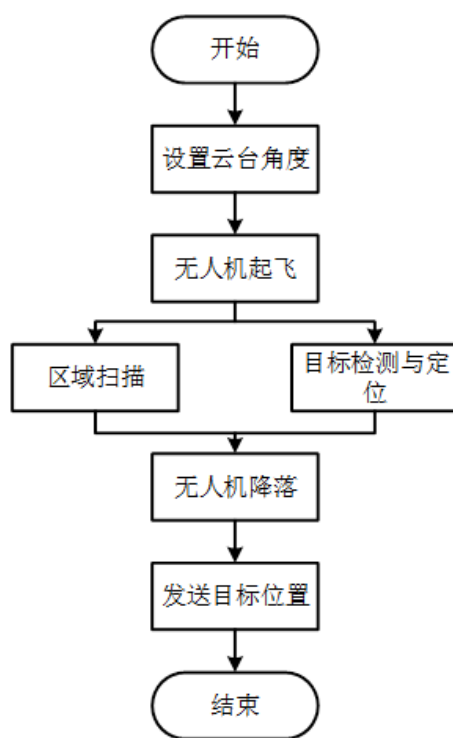


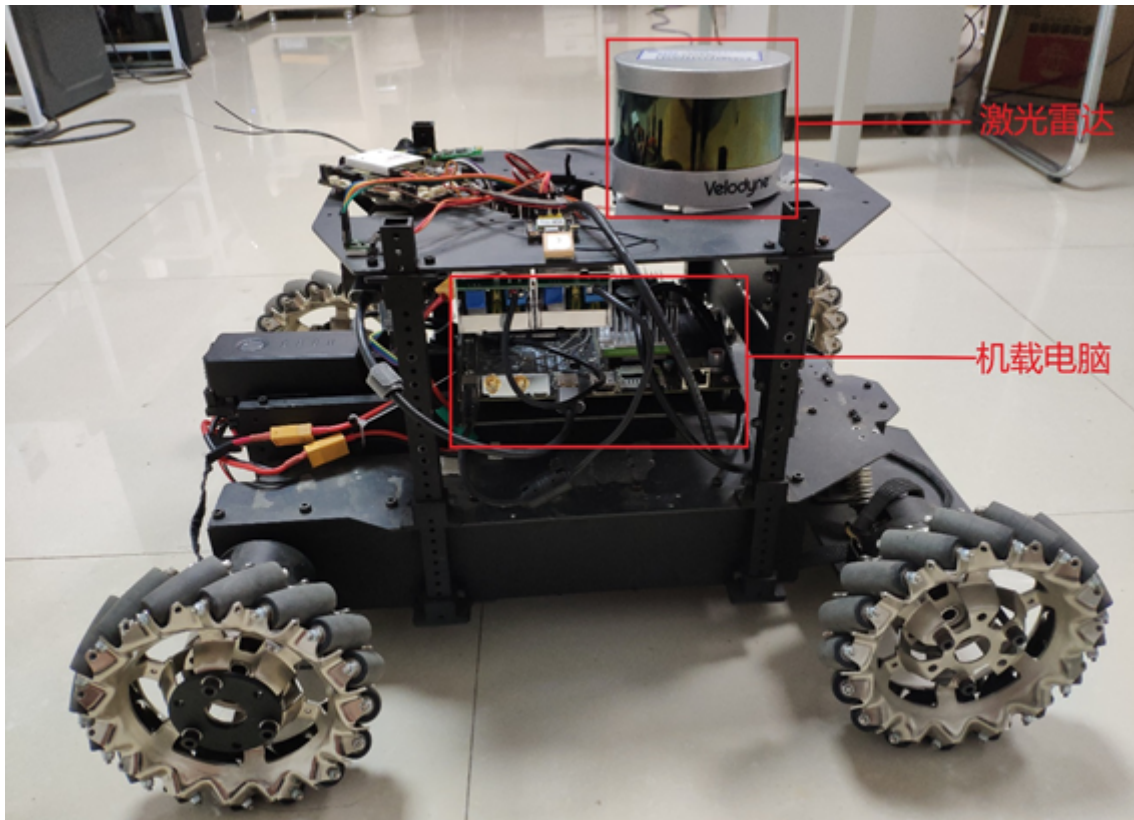
图 2.19 无人机系统软件流程图

## 3 无人机系统

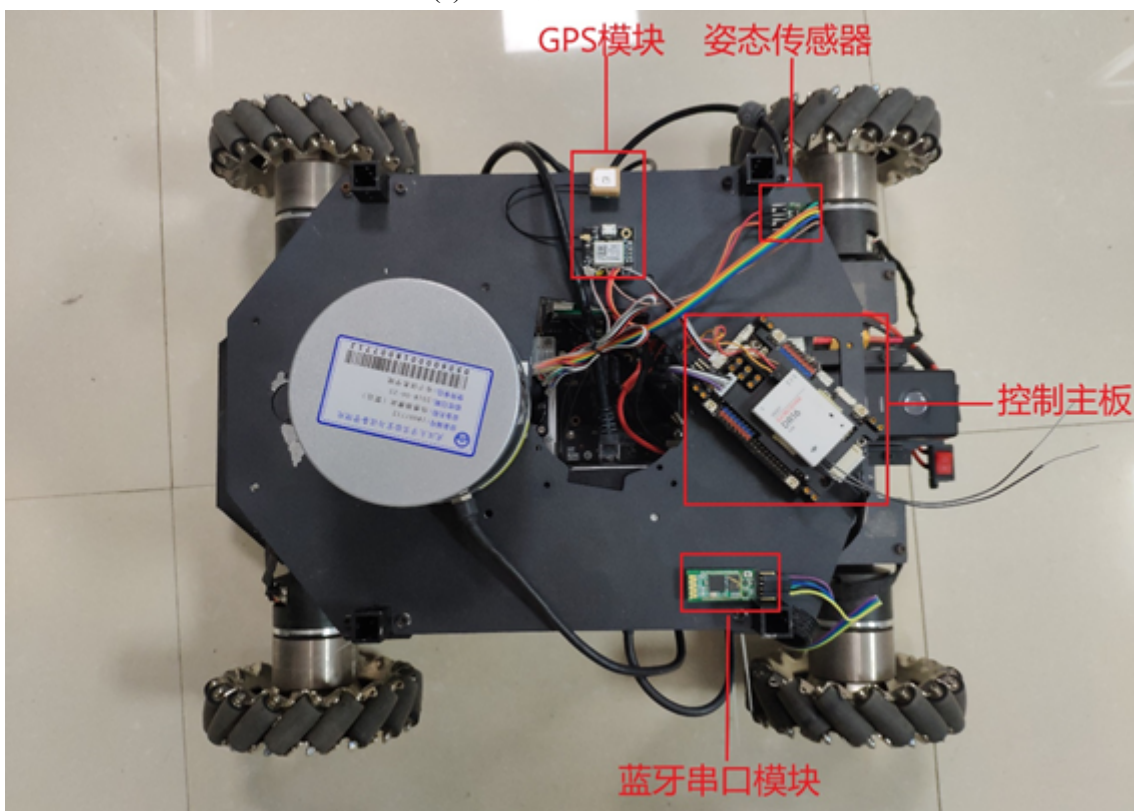
### 3.1 系统硬件模块

#### 3.1.1 整体硬件框架

无人车系统是利用 RoboMasters 步兵车的底盘搭建的，系统的硬件由控制主板、机载电脑、传感器和外设四部分组成。其中控制主板为 RM 开发板，主要用于直接无人车，包含 STM32 微控制器、串行通讯接口 (与机载电脑和遥控器的接收机通讯)、CAN 通讯接口 (与电机集成电调等模块通讯)、多路 PWM 信号输出接口 (用于控制无刷电机转速) 等；机载电脑是 NVIDIA 的 Jetson TX2，主要用来执行建图和导航任务，包含 ARM 架构多核处理器 (运行建图、导航等算法)、串行通讯接口 (与控制主板通讯)、USB 接口 (用于连接蓝牙串口模块和鼠标、键盘等设备) 等；传感器包括激光雷达 (获取点云数据)、GPS 模块 (测量无人车的初始经纬度坐标)、姿态传感器 (测量无人车的航向)、编码器 (测量无人车的位移和速度) 等；外设包括蓝牙串口模块 (与无人机通讯)、遥控装置 (用于设置无人车控制模式以及手动控制无人车) 等。图 3.1 为无人车系统的实物图，图 3.2 为系统的整体硬件框架。



(a) 无人车系统实物图 1



(b) 无人车系统实物图 2

图 3.1 无人车系统实物图



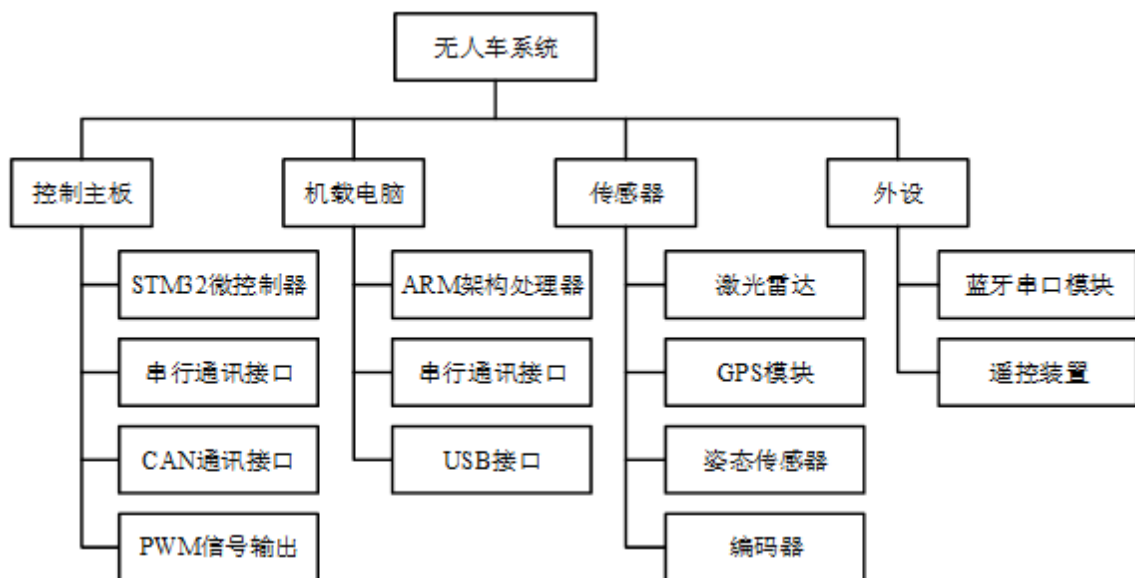


图 3.2 无人车系统整体硬件框架

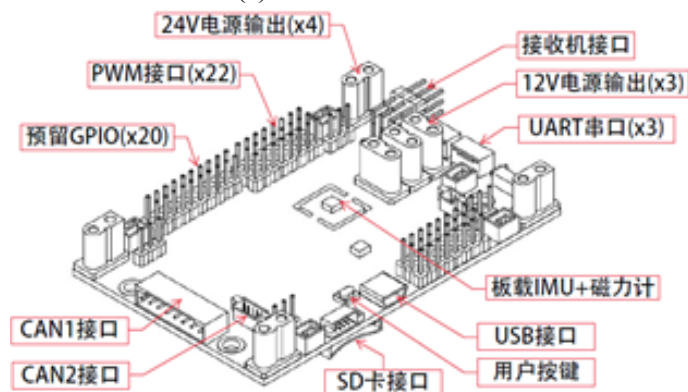
### 3.1.2 控制主板

控制主板采用 DJI 开发的 RM 开发板。该开发板以高性能的 STM32F427IIH6 为主控芯片，同时搭载丰富的外设和硬件接口，包含 2 路 CAN 口、4 路 24V 电源输出、3 路 12V 电源输出、22 路 PWM 输出、3 路串口输出、板载蜂鸣器、板载按键和板载双色 LED 指示灯<sup>[1]</sup>。图 3.3 为该控制主板的外观和主要硬件接口图。

控制主板主要用于接收遥控器或机载电脑发送的运动控制指令，并控制无人车运动，同时将编码器、姿态传感器、GPS 模块等测量的数据发送给机载电脑。



(a) RM 开发板外观



(b) RM 开发板主要硬件接口

图 3.3 RM 开发板<sup>[1]</sup>

### 3.1.3 机载电脑

机载电脑采用 NVIDIA 英伟达公司推出的 Jetson TX2 板卡，它是一台基于 NVIDIA Pascal<sup>TM</sup> 架构的 AI 单模块超级计算机，拥有 256 颗 CUDA 的 GPU 和 2 颗 CPU，内存为 8GB 的 LPDDR4，拥有 USB、PCIE、Ethernet、HDMI 等一系列接口。图 3.4为机载电脑 Jetson TX2 的外观图。

本设计中机载电脑主要用于执行建图和导航算法。首先通过蓝牙串口模块接收无人机发送的目标经纬度坐标数据，然后通过建图和导航算法规划路径，控制无人车安全行驶到目标处。

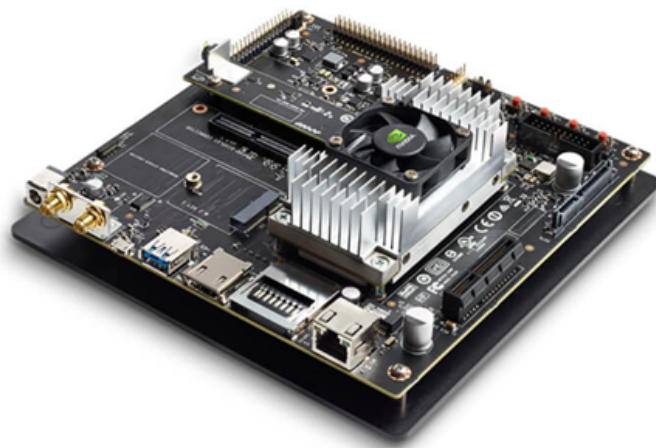


图 3.4 机载电脑 Jetson TX2

#### 3.1.4 激光雷达

激光雷达采用 Velodyne 公司研发的 Velodyne VLP-16 3D 激光雷达。该激光雷达拥有 360° 水平视场角以及 30° 垂直视场角，垂直方向拥有 16 线激光，支持 360 度全覆盖 3D 激光扫描，拥有近 300000 点/秒的高速激光测距采样速率，测量距离长达 100 米，精确度达到  $\pm 3\text{cm}$ 。同时集成有 Web 服务器，可方便地进行监控和配置。图 3.5 为该激光雷达的外观图。

本设计中主要利用激光雷达获取点云数据，再利用点云数据进行建图、避障和导航。同时由于编码器测量的位移和速度具有累积误差，因此可以使用点云数据辅助定位。



图 3.5 Velodyne VLP-16 3D 激光雷达

### 3.1.5 GPS 模块

GPS 模块采用 GT-U7 GPS 模块，该 GPS 模块具有高灵敏度、低功耗、小型化等优点，定位精度达到 2.5m。本设计中，无人车系统中的 GPS 模块用于获取无人车初始位置的 GPS 坐标，从而解算出目标在无人车坐标系下的局部坐标。图 3.6 为 GPS 模块实物图。

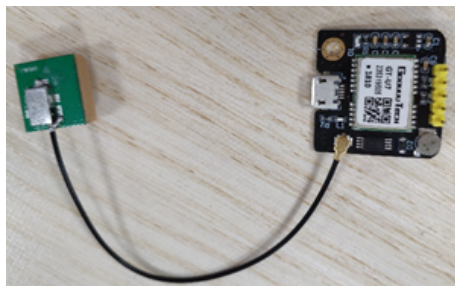


图 3.6 GT-U7 GPS 模块

### 3.1.6 姿态传感器

姿态传感器采用 MPU9250 模块，该模块内部包含加速度计、陀螺仪和磁力计，可输出三轴加速度、角速度和磁感应强度，同时利用卡尔曼滤波算法，可以解算出高精度的三轴姿态角，姿态测量静态精度达到 0.05 度，动态精度达到 0.1 度。该模块支持 UART 和 I<sup>2</sup>C 两种接口，同时可以通过 UART 接口接收符合 NMEA-0183 标准的 GPS 数据。

本设计中，GPS 模块与 MPU9250 模块通过串口相连，GPS 数据先传输给 MPU-9250 模块，然后与加速度、姿态等数据一并传输给控制主板。图 3.7 为 MPU9250 姿态传感器实物图。

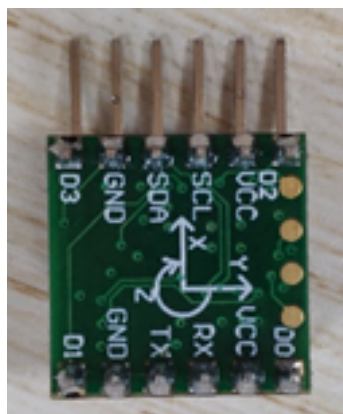


图 3.7 MPU9250 姿态传感器

### 3.1.7 编码器

编码器采用无人车地盘 RM 电机内部集成的编码器，该编码器可以输出电机的角度，通过差分可以求出电机的速度，再利用麦克纳姆四轮全向底盘的运动学传输方程，可以推算出无人车运动的位移和速度。

### 3.1.8 蓝牙串口模块

蓝牙串口模块选用 HC-05 主从一体蓝牙串口模块，与无人机系统一致。本项目中无人车端的蓝牙串口模块用于接收无人机端蓝牙串口模块发送的目标经纬度坐标数据。

### 3.1.9 遥控装置

遥控装置使用 DJI 生产的 DR16 接收机 +DT7 遥控器组成的遥控系统。该遥控器工作于 2.4GHz 频段，采用 DBUS 协议，具有长达 1000 米的通信距离。图 3.8 为该遥控装置实物图。

本设计中遥控器可用于手动控制无人车运动，以及切换无人车控制模式。



图 3.8 DR16 接收机和 DT7 遥控器

### 3.2 自主定位

进行自主定位时，无人车系统的坐标系设定如图 3.9 所示。其中， $O-x-y$  为局部坐标系，以无人车起点为原点， $x$  轴指向北， $y$  轴向左。 $O'-x'-y'$  为车身坐标系，以无人车中心为原点，车身朝向为  $x'$  轴， $y'$  轴向左。 $x'$  轴与  $x$  轴夹角即为偏航角  $yaw$ 。

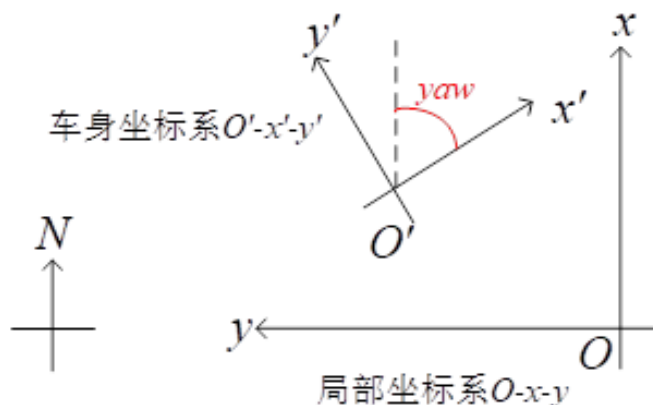


图 3.9 局部坐标系与车身坐标系

本设计中无人车的自主定位基于里程计和激光雷达实现，类似于视觉惯性里程计，只是这里使用激光雷达代替了视觉传感器。

里程计包括无人车在局部坐标系下的位置 ( $x$ 、 $y$  坐标和偏航角  $yaw$ ) 和车身坐标系下的速度 ( $x'$ 、 $y'$  速度和  $yaw$  角速度) 信息，通过编码器和姿态传感器获得，其中编码器可以提供无人车在车身坐标系下的位移和度，姿态传感器可以提供无人车的偏航角及其角速度，利用运动学方程可以计算得到里程计。由于里程计是利用编码器实现的，因此轮子发生打滑将会产生累积误差。另外姿态传感器如果利用磁力计计算偏航角，则容易受到磁场干扰，如果利用角速度积分计算偏航角，则会有累积误差。

利用激光雷达也可以实现无人车的定位，通过激光雷达扫描数据前后 2 帧数据之间的匹配可以估计出无人车的运动信息，但存在噪声引起的累积误差，也会导致里程计漂移。因此可以将里程计数据与激光雷达扫描数据融合，估计出更为精准的无人车位置信息。

本设计利用 ROS 的 `laser_scan_matcher` 包来实现无人车的精确定位。`laser_scan_matcher` 是基于 Andrea Censi 的 CSM (Canonical Scan Matcher) 算法<sup>[36]</sup> 实现的一种增量式激

光匹配器，相较于传统的帧与帧之间的匹配，该匹配器利用关键帧进行匹配，可以有效抑制噪声引起的累积误差。同时该匹配器还可以输入 IMU、里程计等数据，提供对激光雷达传感器当前位置的猜测，可以加速匹配过程并提高定位精度。

### 3.3 SLAM 建图

本设计使用基于激光雷达的 SLAM，即激光 SLAM<sup>[37]</sup>，相较于视觉 SLAM<sup>[38, 39]</sup>，优点是建图精度高，生成的占据栅格地图 (occupancy grid maps) 特别适合进行路径规划。缺点是较为缺乏回环检测能力，累积误差消除较为困难。

在 ROS 中常用的激光 SLAM 算法有 Gmapping<sup>[40, 41]</sup>、HectorSLAM<sup>[42]</sup>、Cartographer<sup>[43]</sup> 等等，这些算法各自具有其优缺点：

- (1) Gmapping 算法。Gmapping 是基于滤波框架的 SLAM，有效利用了里程计信息来提供机器人的位姿先验，因此对激光雷达频率要求低，适用于构建小场景地图。缺点是依赖里程计，无法适用于无人机以及地面小车在不平坦区域建图，并且没有回环检测，在回环闭合时可能造成地图错位；
- (2) HectorSLAM 算法。不需要使用里程计，可用于空中无人机以及地面小车在不平坦区域建图。缺点是要求激光雷达的更新频率较高、测量噪声小，在机器人快速转向时容易发生错误匹配，导致建出的地图发生错位；
- (3) Cartographer 算法。Cartographer 是基于优化框架的 SLAM，具有回环检测，因此建图精度高，可用于构建大场景地图，同时不需要使用里程计，适用于手持激光雷达完成 SLAM 过程。缺点是图优化计算量大。

通过对比上述 3 种常用激光 SLAM 算法，本设计选择了最为合适的 Gmapping 算法。Gmapping 算法基于 Rao-Blackwellized 粒子滤波 (RBPF) 算法<sup>[44]</sup>，在 RBPF 算法基础上做了两点改进：改进建议分布和选择性重采样<sup>[40]</sup>。其定位与建图的过程分为 4 步：采样、计算重要性权值、重采样、地图估计。

### 3.4 自主导航

自主导航采用 ROS 中的 navigation 包，图 3.10 为 navigation 导航包的整体框架。其中，move\_base 模块为整个导航算法的核心，它包含了以下 5 个模块：



- (1) 全局代价地图。用于全局路径规划，由外部提供的地图和激光雷达点云数据计算得到，基本上不会更新，一般是静态 `costmap` 类型；
- (2) 局部代价地图。用于局部路径规划，由激光雷达点云数据计算得到，实时更新；
- (3) 修复机制。包括 `rotate_recovery` 和 `clear_cost_map_recovery`，用于机器人不能找到路径时进行修复；
- (4) 全局路径规划器。依据全局代价地图，利用全局路径规划算法，生成机器人到目标位置的全局路径。导航包提供了 2 种全局路径规划算法，分别是 Dijkstra 算法<sup>[45]</sup> 和 A\* 算法<sup>[46]</sup>，默认使用 Dijkstra 算法；
- (5) 局部路径规划器。依据局部代价地图、全局路径规划器生成的全局路径和机器人的定位信息，利用局部路径规划算法，生成机器人到达目标位置的局部路径，并输出控制机器人运动的速度命令。由于局部代价地图是实时更新的，因此局部路径也是实时更新的，而机器人按照局部路径运动，并非全局路径，因此可以实现动态避障。导航包提供了多种局部路径规划器，包括 `dwa`<sup>[47]</sup>、`trajectory`、`teb`<sup>[48, 49]</sup>、`eban`<sup>[50]</sup> 等等。

当一条全局路径导航失败时，全局路径规划器会尝试生成新的全局路径，如果不能生成可行的全局路径，那么算法会执行修复操作，然后再次尝试生成全局路径，倘若还是无法生成可行的全局路径，则导航失败。

除了 `move_base` 模块，导航包还需要用户提供机器人的定位信息、目标位置信息、激光雷达点云数据和地图。本设计中，定位信息可以由自主定位系统提供，目标位置信息由无人机系统通过检测和定位算法得到，并通过蓝牙发送给无人车，激光雷达点云数据由 `velodyne` 包处理激光雷达输出的原始数据得到。地图是可选项，因此导航算法存在 3 种模式，一是没有地图，进行局部导航，导航效果会差很多；二是事先有地图，直接进行全局导航，但本项目中事先并没有地图，因此该方案不可行；三是利用建图算法生成地图，边建图边导航，虽然效果不及全局导航，但是胜过局部导航，并且方案可行。因此本设计采用边建图边导航的自主导航方式。



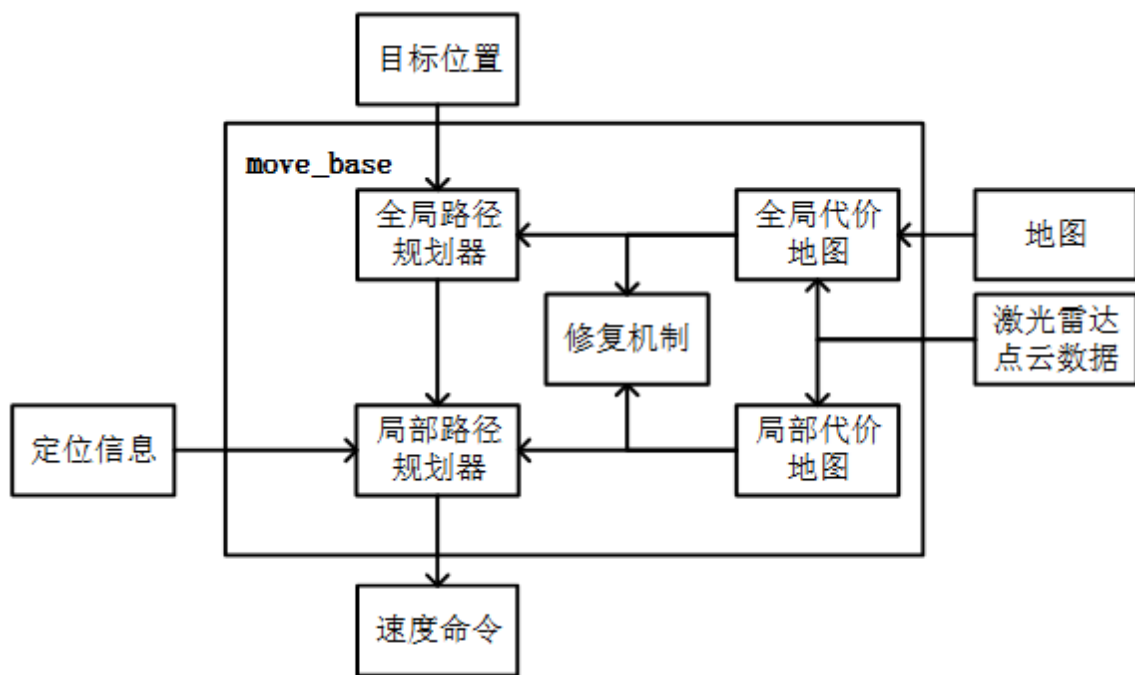


图 3.10 navigation 导航包整体框架

### 3.5 系统软件设计

无人车系统的机载电脑 Jetson TX2 使用 Ubuntu 16.04 操作系统，并装有 ROS，本项目分别利用 ROS 的 `laser_scan_matcher` 包、`gmapping` 包和 `navigation` 包实现了无人车的自主定位、SLAM 建图和自主导航功能。图 3.11 为无人车系统的软件流程图，系统首先等待无人机发送目标位置信息，接收到目标位置信息后，启动无人车，开始同步进行定位、建图和导航，无人车达到所有目标位置后，返回其初始位置，搜索任务完成。

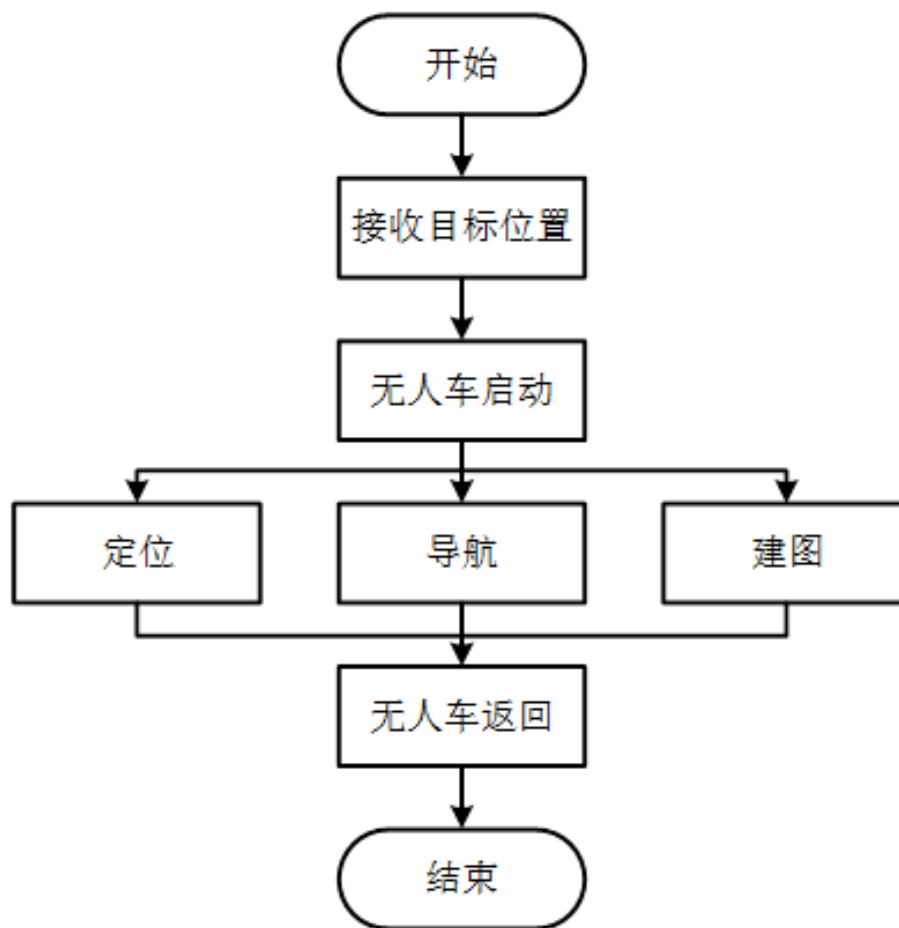


图 3.11 无人车系统软件流程图

### 3.6 本章小结

本章首先介绍了无人车系统的硬件框架和模块，然后依次介绍了如何实现无人车的自主定位、SLAM 建图和自主导航，最后介绍了系统的软件设计。

## 4 系统测试与结果分析

### 4.1 无人机系统测试

#### 4.1.1 目标检测测试

本项目使用基准标记作为搜索目标，图 4.1~ 4.3为摄像头拍摄的三种基准标记图像的检测结果。可以看到，对于 AprilTag，不同型号的标记都可以被正确检测到，但是型号为  $4\times 4$  的标记存在误检现象；对于 ChromaTag (只有  $4\times 4$  一种型号)，同样可以被正确检测到，但也存在误检现象；对于 ArUco 标记，不同型号的标记均可以被正确检测出来，但型号为  $4\times 4$  的标记也存在误检现象。然而通过实验发现，误检给出的 ID 均为 17，该 ID 对应的 ArUco 标记如图 4.4所示，可以看到图案过于简单，因此容易引起误检，而检测算法并非导致误检的主要原因。

通过比较可以发现基准标记的型号越复杂，检测的准确率越高，并且在三种基准标记中，ArUco 标记检测的准确率最高。

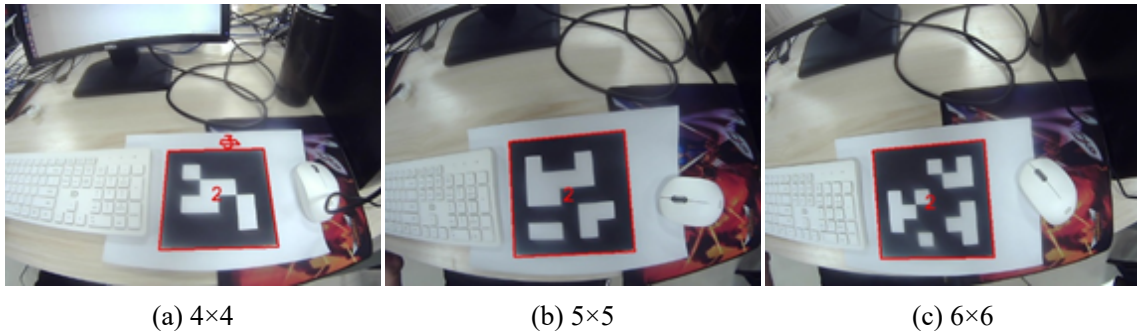


图 4.1 AprilTag 检测结果



图 4.2 ChromaTag 检测结果

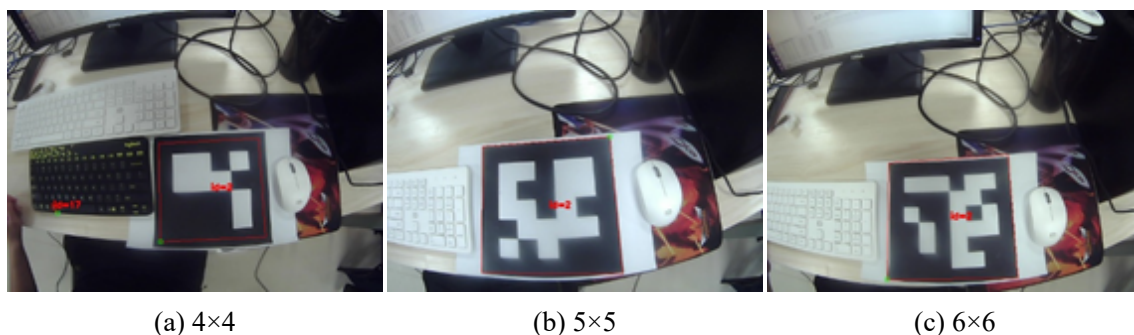


图 4.3 ArUco 标记检测结果

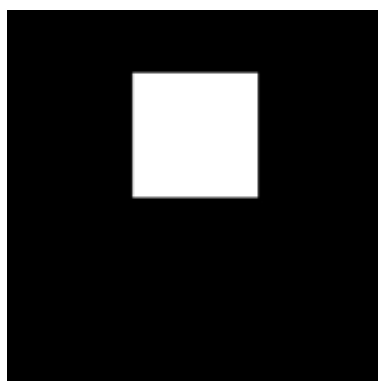


图 4.4 ID 为 17 的 4×4 ArUco 标记

表 4.1 为三种基准标记检测的耗时情况测试结果，测试平台为台式电脑，处理器型号为 Intel i5-4570，主频为 3.2GHz，测试的基准标记数量为 0 或 1。

表 4.1 基准标记检测的耗时情况测试结果

基准标记	型号	平均耗时 (ms)
AprilTag	4×4	17
	5×5	15
	6×6	15
ChromaTag	4×4	7
ArUco	4×4	<b>4</b>
	5×5	<b>4.5</b>
	6×6	<b>4.8</b>

可以看到 ArUco 标记的平均耗时最少，仅为 4~5ms，而 AprilTag 的平均耗时最多。ChromaTag 和型号为 4×4 的 AprilTag 由于误检现象严重，导致检测到的标记数量超过了真实的标记数量，使得检测算法的运算次数增多，最终导致平均耗时增加，这正是 4×4 的 AprilTag 平均耗时高于 5×5 的 AprilTag 的原因。

综合三种基准标记的检测准确率和耗时情况，本设计最终选用了型号为 5×5

的 ArUco 标记。

4.1.2 目标定位测试

图 4.5为无人机拍摄的目标定位测试图像，测试时使用棋盘格作为定位目标 (与基准标记作为定位目标效果类似)。

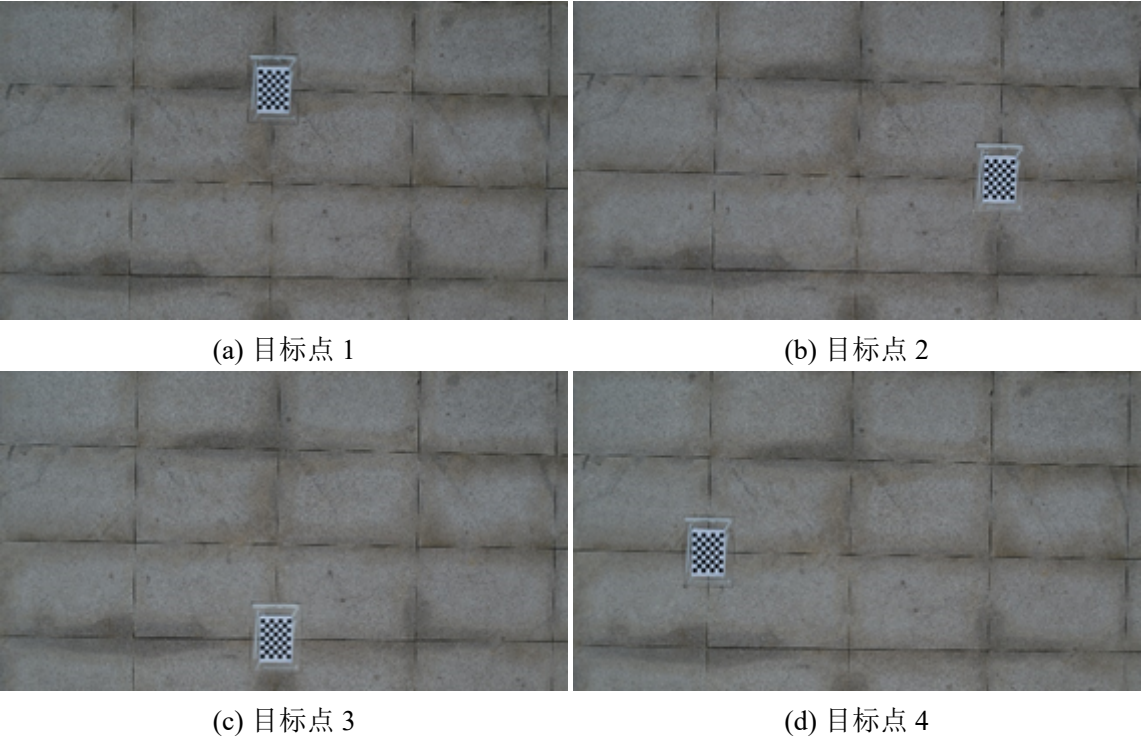


图 4.5 目标定位测试图像

表 4.2为目标定位测试的数据和结果，测试时保持无人机在空中悬停，无人机经纬度为 (30.540224°N, 114.351715°E)，高度为 3m，无人机经纬度和目标经纬度真实值通过高精度差分 GPS 设备测得，地球半径取平均半径 6371.393km。

可以看到在机体坐标系下目标  $(x, y)$  坐标误差在 10cm 以内，经纬度误差在  $10^{-6}$  度以内，定位误差非常小，表明基于 PnP 算法的目标定位方案可行，并且精度可以达到 10cm 以内。

表 4.2 目标定位测试结果

目标序号	偏航角	机体坐标系下目标 $(x, y)$ 坐标 (mm)		
		真实值	计算值	误差
1	80°	(600, 0)	(594, -68)	<b>(-6, -68)</b>
2	170°	(0, 900)	(21, -911)	<b>(21, 11)</b>
3	260°	(-600, 0)	(-547, -50)	<b>(53, -50)</b>
4	350°	(0, -900)	(2, -948)	<b>(2, -48)</b>

### 4.1.3 自主飞行测试

本设计利用 ROS 和 DJI 的 Onboard SDK 实现无人机的自主飞行，通过将无人机的局部坐标数据发布到 ROS，可以利用 Rviz 绘制出无人机的飞行轨迹。图 4.6 显示了无人机的自主飞行轨迹，如绿线所示，可以看到实际的飞行轨迹与预设的非常接近，扫描轨迹形状与图 2.18 所示完全一致，证明本设计可以很好地通过软件控制无人机自主飞行。

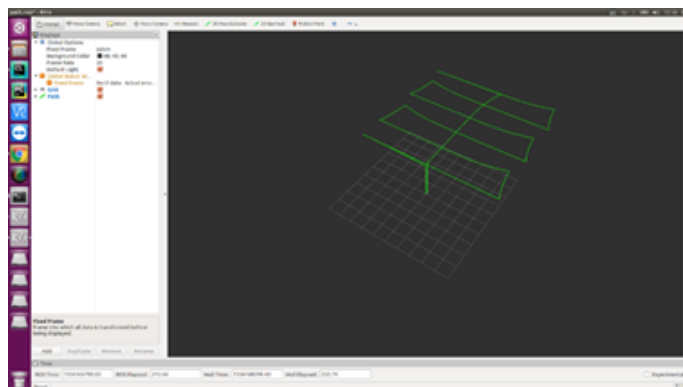


图 4.6 无人机自主飞行测试结果

## 4.2 无人车系统测试

### 4.2.1 自主定位测试

本设计采用基于里程计和激光雷达扫描匹配的自主定位方案，通过将无人车的位姿信息发布到 ROS，可以利用 Rviz 实时显示无人车的当前位置和运动轨迹。图 4.7 为无人车沿着矩形路径运动 2 周的轨迹，其中绿线代表无人车的运动轨迹，红色箭头代表无人车当前的位置和朝向。可以看到 2 次绕矩形运动的轨迹几乎重合，说明累积误差非常小，定位比较精准。

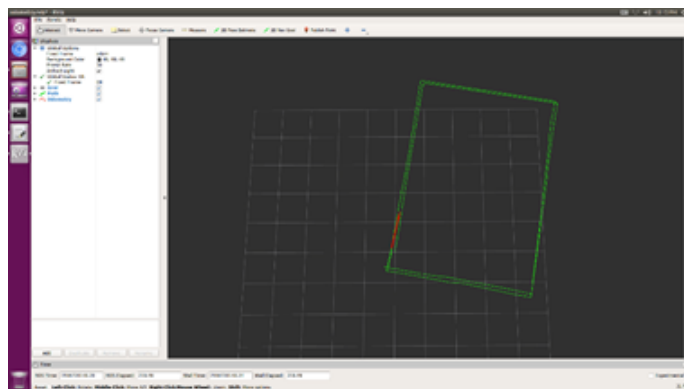


图 4.7 无人车运动轨迹

#### 4.2.2 SLAM 建图测试

SLAM 建图测试的场地为实验楼走廊，图 4.8为 SLAM 建图结果，图中走廊的两侧墙壁非常明显，两处楼梯也正确地显示在地图中，地图没有错位现象，建图效果非常好。

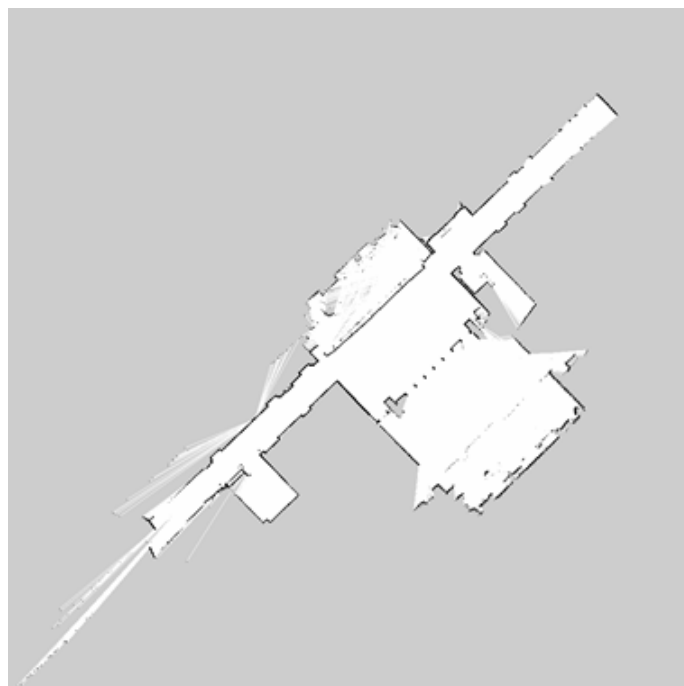


图 4.8 SLAM 建图结果

#### 4.2.3 自主导航测试

本设计采用边建图边导航的自主导航方案，图 4.9展示了自主导航的测试场地，测试时使用了一些箱子和泡沫块作为障碍物，模拟复杂环境。图 4.10展示了自主导航测试效果，图 4.10a为 Rviz 显示的图像，图 4.10b为对应时刻的无人车图像。



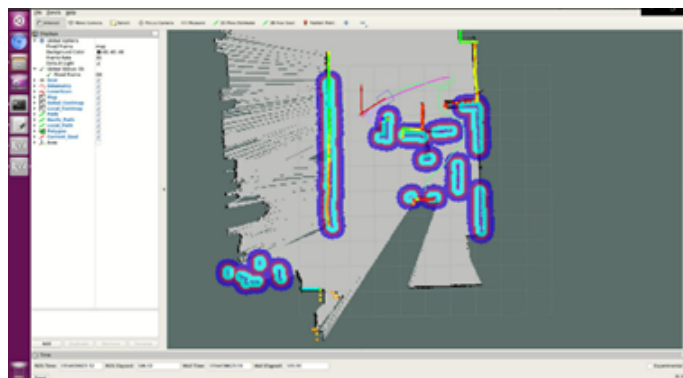
图 4.10a中，较细的曲线代表无人车的路径或轨迹，其中，紫色曲线为导航算法规划的全局路径，绿色曲线为无人车的运动轨迹，绿色曲线端点处的红色箭头代表无人车当前的位置和朝向，另一处红色箭头代表目标的位置。此外，图中较粗的线条代表激光雷达检测到的障碍物，黑色线条代表地图中的障碍物边界，蓝紫色的区域代表对障碍物的膨胀，离障碍物越近颜色越浅、越危险，这是代价地图可视化的效果。

图 4.10展示的是无人车达到第一个目标点后，前往第二个目标点的过程，导航算法已经规划出了正确的全局路径，并且无人车正按着该全局路径向目标处运动。同时，建图算法已经建好了部分地图。最终无人车安全达到了所有目标点，说明无人车系统的导航算法安全、可行。



图 4.9 自主导航测试场地





(a) Rviz 显示图像



(b) 无人车图像

图 4.10 自主导航测试效果

### 4.3 系统整体测试

图 4.11为系统整体测试的实验场地，搜索目标 (基准标记) 周围放置了一些泡沫块作为障碍物。图 4.12展示了测试效果，图 4.12a为无人机正在扫描搜索区域，图 4.12b为无人车正在前往目标处。测试时无人机成功检测到了目标，并比较精确地计算出了目标的 GPS 坐标，最终无人车成功到达了目标处。测试结果表明，本系统已经初步成型，具备了空地联合目标搜索的能力。



图 4.11 系统整体测试测试场地



(a) 无人机正在扫描搜索区域



(b) 无人车正在前往目标处

图 4.12 系统整体测试测试效果

#### 4.4 本章小结

本章分别对无人机系统、无人车系统以及整体系统进行了测试与结果分析，测试结果显示，本设计的主题——基于无人机-车的空地联合目标搜索系统已经初步成型，具备了初期预想的功能。

## 5 总结与展望

### 5.1 工作总结

本文从空地联合在目标搜索方面的优势出发，参考国内外发展现状，利用计算机视觉技术，自主定位、建图与导航技术，自动控制技术，传感器技术等实用技术，通过硬件设计和软件开发，最终实现了一套完整的基于无人机-车的空地联合目标搜索系统。本文的主要工作与贡献如下：

- (1) 实现了无人机系统。在硬件上，搭建了一套基于大疆 M100 无人机、机载电脑 Manifold、云台相机、GPS 模块、蓝牙串口模块和遥控器的无人机开发平台；在软件上，实现了目标的检测与定位，以及无人机的自主飞行；
- (2) 实现了无人车系统。在硬件上，搭建了一套基于 Robomasters 步兵车底盘、STM32 控制主板、TX2 机载电脑、激光雷达、姿态传感器、GPS 模块、蓝牙串口模块和遥控装置的无人车开发平台；在软件上，实现了无人车的自主定位、建图、避障和导航；
- (3) 实现了空地联合目标搜索。首先利用无人机扫描待搜索区域，同时进行目标检测和定位，然后利用蓝牙串口模块将目标位置信息发送给无人车，最后无人车通过自主导航行驶到目标处，完成目标搜索任务。

实验表明，本设计中的无人机系统实现了目标检测与定位的功能，无人车系统实现了自主定位、建图和导航的功能，并且整个系统具备了空地联合目标搜索的能力。

### 5.2 存在的问题与展望

限于研究的时间，本文提出的基于无人机-车的空地联合搜索系统虽然已初步成型，但仍有许多不足之处，有待改进和完善。例如具体目标的检测与定位问题、依赖于 GPS 信号的问题、无人机与无人车的定位精度问题、复杂环境下的工作鲁棒性问题、系统在真实应用场景中的可靠性问题等等。针对存在的问题与不足，作者在此提出以下几点展望：

- (1) 利用深度学习实现具体目标的检测，不过这需要采集大量的训练样本，同时目标定位算法也会更加复杂，可以考虑使用双目相机或深度相机进行目标定

位；

- (2) 利用无人机生成全局地图，用于无人机和无人车的定位，从而摆脱对 GPS 信号的依赖；
- (3) 融合更多高性能传感器，提升无人机与无人车的定位精度，使系统更加可靠。

## 参考文献

- [1] 莫帮杰. 室内自动驾驶物流机器人 [D]. 武汉: 武汉大学, 2017.
- [2] OLSON E. AprilTag: A robust and flexible visual fiducial system[C] // IEEE International Conference on Robotics and Automation. 2011 : 3400–3407.
- [3] DEGOL J, BRETL T, HOIEM D. ChromaTag: A Colored Marker and Fast Detection Algorithm[C] // IEEE International Conference on Computer Vision: Vol 1. 2017: 1481–1490.
- [4] 高翔等. 视觉 SLAM 十四讲: 从理论到实践 [M]. 1st ed. 北京: 电子工业出版社, 2017: 84–90.
- [5] FANKHAUSER P, BLOESCH M, KRÜSI P, et al. Collaborative Navigation for Flying and Walking Robots[C] // IEEE/RSJ International Conference on Intelligent Robots and Systems. 2016 : 2859–2866.
- [6] SHEN C, ZHANG Y, LI Z, et al. Collaborative air-ground target searching in complex environments[C] // IEEE International Symposium on Safety. 2017 : 230–237.
- [7] VIDAL-CALLEJA T A, BERGER C, SOLÀ J, et al. Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain[J]. Robotics and Autonomous Systems, 2011, 59(9): 654–674.
- [8] MINAEIAN S, JIAN L, SON Y J. Vision-Based Target Detection and Localization via a Team of Cooperative UAV and UGVs[J]. IEEE Transactions on Systems Man and Cybernetics Systems, 2015, 46(7): 1–12.
- [9] BUTZKEY J, DORNBUSHY A, LIKHACHEVY M. 3-D exploration with an air-ground robotic system[C] // IEEE/RSJ International Conference on Intelligent Robots and Systems. 2015 : 3241–3248.
- [10] QIN H, MENG Z, MENG W, et al. Autonomous Exploration and Mapping System Using Heterogeneous UAVs and UGVs in GPS-denied Environments[J]. IEEE Transactions on Vehicular Technology, 2019, 68(2): 1339–1350.

- [11] WANG L, CHENG D, FEI G, et al. A Collaborative Aerial-Ground Robotic System for Fast Exploration[C] // IEEE International Conference on Robotics and Biomimetics (ROBIO). 2018 : 2259 – 2266.
- [12] POTENA C, KHANNA R, NIETO J, et al. AgriColMap: Aerial-Ground Collaborative 3D Mapping for Precision Farming[J]. IEEE Robotics and Automation Letters, 2019, 4(2) : 1085 – 1092.
- [13] GAWEL A, LIN Y, KOUTROS T, et al. Aerial-Ground collaborative sensing: Third-Person view for teleoperation[C] // IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). 2018 : 1 – 7.
- [14] GATRELL L B, HOFF W A, SKLAIR C W. Robust image features: concentric contrasting circles and their image extraction[J]. Proceedings of SPIE - The International Society for Optical Engineering, 1992, 1612 : 235 – 244.
- [15] BERGAMASCO F, ALBARELLI A, COSMO L, et al. An Accurate and Robust Artificial Marker based on Cyclic Codes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(12) : 2359.
- [16] BERGAMASCO F, ALBARELLI A, RODOLA E, et al. RENE-Tag: A high accuracy fiducial marker with strong occlusion resilience[C] // Computer Vision and Pattern Recognition. 2011 : 113 – 120.
- [17] CALVET L, GURDJOS P, CHARVILLAT V. Camera tracking using concentric circle markers: Paradigms and algorithms[C] // IEEE International Conference on Image Processing. 2012 : 1361 – 1364.
- [18] CALVET L, GURDJOS P, GRIWODZ C, et al. Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions[C] // Computer Vision and Pattern Recognition : Vol 1. 2016 : 562 – 570.
- [19] FIALA M. ARTag, a fiducial marker system using digital techniques[C] // IEEE Computer Society Conference on Computer Vision and Pattern Recognition : Vol 2. 2005 : 590 – 596.

- [20] WANG J, OLSON E. AprilTag 2: Efficient and robust fiducial detection[C] //IEEE/RSJ International Conference on Intelligent Robots and Systems. 2016: 4193–4198.
- [21] GARRIDO-JURADO S, MUÑOZ-SALINAS R, MADRID-CUEVAS F J, et al. Automatic generation and detection of highly reliable fiducial markers under occlusion[J]. Pattern Recognition, 2014, 47(6): 2280–2292.
- [22] CHO Y, LEE J, NEUMANN U. A Multi-ring Color Fiducial System and A Rule-Based Detection Method for Scalable Fiducial-tracking Augmented Reality[C] //International Workshop on Augmented Reality: Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes. 1999: 147–165.
- [23] FELZENSZWALB P F, HUTTENLOCHER D P. Efficient graph-based image segmentation[J]. International Journal of Computer Vision, 2004, 59(2): 167–181.
- [24] HARTIGAN J A, WONG M A. Algorithm AS 136: A K-Means Clustering Algorithm[J]. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979, 28(1): 100–108.
- [25] SHI J, TOMASI C. Good Features to Track[C] //IEEE Conference on Computer Vision and Pattern Recognition. 2002: 593–600.
- [26] CANNY J. A computational approach to edge detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8(6): 679–698.
- [27] SUZUKI S, BE K. Topological structural analysis of digitized binary images by border following[J]. Computer Vision Graphics and Image Processing, 1985, 30(1): 32–46.
- [28] D. H. DOUGLAS T K P. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature[J]. The International Journal for Geographic Information and Geovisualization, 1973, 10(2): 112–122.
- [29] OSTU N. A threshold selection method from gray-histogram[J]. IEEE Transactions on Systems, Man and Cybernetics, 1978, 9(1): 62–66.

- [30] SHIQI L, CHI X, MING X. A Robust  $O(n)$  Solution to the Perspective-n-Point Problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1444–1450.
- [31] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, et al. Complete solution classification for the perspective-three-point problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(8): 930–943.
- [32] HARTLEY R, ZISSERMAN A. Multiple View Geometry in Computer Vision[M]. 2nd ed. Oxford: Cambridge University Press, 2004: 71–75.
- [33] LEPETIT V. Accurate  $O(n)$  solution to the PnP problem[J]. International Journal of Computer Vision, 2008, 81(2): 155–166.
- [34] ADRIAN P S, JUAN A C, FRANCESC M N. Exhaustive linearization for robust camera pose and focal length estimation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(10): 2387–2400.
- [35] ZHANG Z Y. A Flexible New Technique for Camera Calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330–1334.
- [36] CENSI A. An ICP variant using a point-to-line metric[C] // IEEE International Conference on Robotics and Automation. 2008: 19–25.
- [37] 张霖, 赵祚喜, 张智刚, et al. 基于激光测距传感器的 SLAM 方法研究 [C] // 中国农业工程学会 2011 年学术年会论文集. 2011.
- [38] 权美香, 朴松昊, 李国. 视觉 SLAM 综述 [J]. 智能系统学报, 2016, 11(6): 768–776.
- [39] 何俊学, 李战明. 基于视觉的同时定位与地图构建方法综述 [J]. 计算机应用研究, 2010, 27(8): 2839–2844.
- [40] GRISETTI G, STACHNISS C, BURGARD W. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling[C] // IEEE International Conference on Robotics and Automation. 2005: 2432–2437.



- [41] GRISETTI G, STACHNISS C, BURGARD W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23(1): 34–46.
- [42] KOHLBRECHER S, STRYK O V, DARMSTADT T U, et al. A flexible and scalable SLAM system with full 3D motion estimation[C] // IEEE International Symposium on Safety. 2011 : 155–160.
- [43] HESS W, KOHLER D, RAPP H, et al. Real-time loop closure in 2D LIDAR SLAM[C] // IEEE International Conference on Robotics and Automation. 2016 : 1271–1278.
- [44] MURPHY K. Bayesian Map Learning in Dynamic Environments[C] // In Neural Info. Proc. Systems. 1999 : 1015–1021.
- [45] DIJKSTRA E W. A Note on Two Problems in Connexion with Graphs[J]. NUMERISCHE MATHEMATIK, 1959, 1(1): 269–271.
- [46] E. HART P, J. NILSSON N, RAPHAEL B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100–107.
- [47] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance[J]. IEEE Robotics Automation Magazine, 1997, 4(1): 23–33.
- [48] Roesmann C, Feiten W, Woesch T, et al. Trajectory modification considering dynamic constraints of autonomous robots[C] // 7th German Conference on Robotics. 2012 : 74–79.
- [49] ROSMANN C, FEITEN W, WOSCH T, et al. Efficient trajectory optimization using a sparse model[C] // European Conference on Mobile Robots. 2014 : 138–143.
- [50] Quinlan S, Khatib O. Elastic bands: connecting path planning and control[C] // IEEE International Conference on Robotics and Automation : Vol 2. 1993 : 802–807.

## 致谢

时光飞逝，转眼间大学本科生涯即将结束，回想 4 年的时光，很多事情依旧历历在目。在这宝贵的 4 年青春里，我庆幸自己加入了卓工班，参加了电设培训，进入了电科院实习，所有的这些难忘的历程，都让我收获颇。大学 4 年，学到了很多，看到了很多，算是不留遗憾了。

在此非常感谢我的指导老师杨文教授，在毕业设计中给予了我很多的指导和帮助，也用他的科研热情感染着我；也感谢卓工班和实验中心的老师们的关怀和悉心培养；

同时感谢实验室的师兄师姐，在毕业设计中提供了很多建议和实验上的帮助；也感谢我的室友和同学们，正是在你们的帮助和陪伴中，我度过了美好的大学时光。

还要感谢美丽的武汉大学，给我带来了 4 年美好而难忘的经历，也让我找到了自己的兴趣，明确了今后的追求的方向。最后，感谢我的父母，感谢你们一直以来对我的支持和关爱。

李皓

2019 年 5 月 3 号

于武汉大学