



上海科技大学
ShanghaiTech University

本科毕业论文（设计）

题 目: 基于 YOLOv5 的机器人视觉伺服

学生姓名: 何梓安

学 号: 2018533097

入学年份: 2018

所在学院: 信息科学与技术学院

攻读专业: 计算机科学与技术

指导教师: Sören Schwertfeger

上海科技大学

2022 年 05 月



上海科技大学
ShanghaiTech University

THESIS

Subject: Visual Servoing with YOLOv5 on robots

Student Name: Zian He

Student ID : 2018533097

Year of Entrance: 2018

School: School of Information Science and Technology

Major: Computer Science and Technology

Advisor: Sören Schwertfeger

ShanghaiTech University

Date: 05 / 2022

上海科技大学

毕业论文(设计)学术诚信声明

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名: 何梓安

日期: 2022 年 4 月 27 日

上海科技大学
毕业论文（设计）版权使用授权书

本毕业论文（设计）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海科技大学可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业论文（设计）。

保 密 ，在____年解密后适用本授权书。

本论文属于

不保密 .

（请在以上方框内打“√”）

作者签名：何梓宇

指导教师签名：Sören Schwerger

日期：2022 年 4 月 27 日

日期：2022 年 04 月 28 日

摘要

本次项目的一大目标是 RoboMaster AI 挑战赛。比赛的规则是全自动机器人通过发射弹丸击打敌方机器人进行射击对抗，当比赛结束时，机器人总血量高的一方获得比赛胜利。比赛的核心技术难点在于如何识别追踪并且精准的对敌人造成有效打击。为了实现这一系列目标，我结合了多学科知识，发现了很多有趣的新思路和新方法。在该比赛中，核心致胜的部分就是射击模块，机器人通过识别敌人并射击出小球击中敌方装甲板来对其造成伤害，通过击杀敌方所有成员获得胜利。视觉工作的主要目标是对敌方目标的装甲板进行识别并控制机器人的云台进行瞄准射击，同时，为了进一步增加识别和射击精确度以及系统运行效率，引入了一些优化算法和 PID 控制算法，并且加入弹道分析来提高射击精准度。YOLOv5 作为 YOLOv4 的迭代版本，在各方面都有了显著的提升。首先，YOLOv5 是在 PyTorch 中实现的，相比于基于 DarkNet 框架开发的 YOLOv4 来说，模型的理解、训练和部署变得更加容易。其次，在性能上，得益于多种计算机视觉前沿算法的运用，YOLOv5 在训练速度和运行效率上也大大提升。轻量化的模型加上不错的运行效率使得 YOLOv5 成为了非常不错的机器人视觉解决方案。最终，该项目取得了不错的效果，我方机器人能快速且精准的识别到打击目标并对其做出可观的输出，如果比赛顺利举办相信它能取得不错的成绩。同时，也希望我的工作能为我校未来参赛的学弟学妹作为参考和基石。

关键词： YOLOv5，计算机视觉，机器人，PnP 算法，追踪算法，弹道分析，PID 控制

Abstract

The goal of this project is the RoboMaster AI Challenge. The rule of the game is that the fully automatic robot shoots against the enemy robot by shooting bullets and the team with the highest total HP wins the game when the game is over. The core technical difficulty of the game is how to detect, track and accurately strike the enemy effectively. In order to achieve this series of goals, I combined multi-disciplinary knowledge and discovered many interesting new ideas and methods. In this game, the core part is the shooting module. The robot recognizes the enemy and shoots a small ball to hit the enemy armor board to cause damage, and wins by killing all the enemy members. The main goal of the vision work is to identify the armor board of the enemy target and control the robot's gimbal for aiming and shooting. At the same time, in order to further increase the accuracy of identification and shooting and the operating efficiency of the system, some optimization algorithms and PID control algorithm are introduced. I also add ballistic analysis to improve shooting accuracy. As an iterative version of YOLOv4, YOLOv5 has been significantly improved in all aspects. First, YOLOv5 is implemented in PyTorch, which makes it easier to understand, train, and deploy models than YOLOv4 which is developed based on the DarkNet framework. Secondly, in terms of performance, thanks to the use of a variety of computer vision cutting-edge algorithms, YOLOv5 has also greatly improved its training speed and operating efficiency. The lightweight model and good operating efficiency make YOLOv5 a very good robot vision solution. In the end, the project has achieved good results. Our robot can quickly and accurately identify the target and make considerable output. If the game is successfully held, I believe it can achieve good results. At the same time, I also hope that my work can serve as a reference and a cornerstone for our school's future students to participate in the competition.

Keywords: YOLOv5, computer vision, robotics, PnP algorithm, tracking algorithm, ballistic analysis, PID control algorithm

目 录

第一章 绪论	1
1.1 研究背景	1
1.1.1 比赛规则	1
1.1.2 YOLOv5	3
1.1.3 视觉伺服 (Visual Servoing)	7
1.2 系统架构及软硬件说明	9
1.2.1 系统架构	9
1.2.2 软件	10
1.2.3 硬件	10
第二章 目标识别与追踪	12
2.1 敌方单位识别	12
2.1.1 工业相机设置	12
2.1.2 模型训练	12
2.1.3 实际测试	14
2.2 目标追踪	14
2.2.1 敌方单位追踪	14
第三章 射击控制	15
3.1 敌方单位距离推算	15
3.1.1 获取敌人距离的目的	15
3.1.2 视觉测距	15
3.1.3 LiDAR 测距	16
3.1.4 结合视觉和 LiDAR 测距提高鲁棒性	17
3.2 弹道分析	17
3.2.1 牛顿法	17
3.2.2 拟合法	18
第四章 策略及控制优化	20
4.1 策略优化	20
4.1.1 动态射速及射频控制	20
4.2 控制优化	20
4.2.1 云台 Pitch 与 Yaw 轴控制优化	20

第五章 结论	23
参考文献	24
致谢	26

第一章 绪论

1.1 研究背景

1.1.1 比赛规则

“在 2022 赛季中，采用“全自动机器人射击对抗”的形式，场地内布满功能机关，参赛队伍需利用官方机器人平台，通过感知战场的环境信息，根据场上形势自主决策，进行运动规划与控制。全自动机器人通过发射弹丸击打敌方机器人进行射击对抗。比赛结束时，机器人总血量高的一方获得比赛胜利。”^[1]

1.1.1.1 比赛场地

RMUA 2022 的核心比赛场地被称为“战场”。战场是一个长为 8080 mm、宽为 4480 mm 的区域，主要包含保护围挡区 (A)、障碍块区 (B)、启动区 (C)、哨岗区 (D)、障碍块对应贴纸 (E)、加成/惩罚区 (F)、反射区 (G) 等。具体细节如图1.1所示。

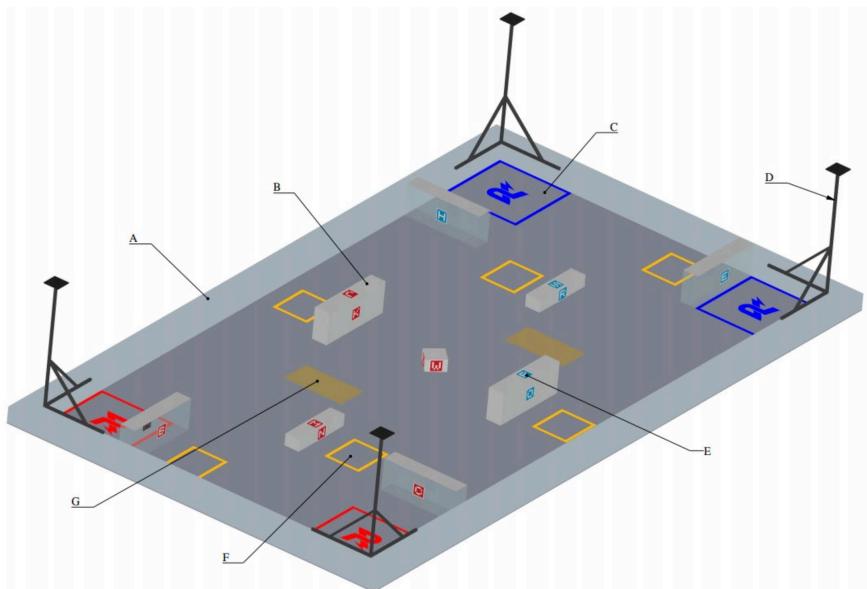


图 1.1 比赛场地图 (引自：RMUA)

1.1.1.2 机器人技术规范

见表1.1。

表 1.1 机器人技术规范

项目	限制
初始血量	2000
运行方式	全自动
最大供电总容量 (Wh)	200
最大供电电压 (V)	30
最大底盘功率 (W)	不限
发射机构	17mm 发射机构
补弹方式	激活弹丸补给加成区
初始弹量	装满弹仓
弹丸射击初速度上限 (m/s)	25
枪口热量上限	240
最大重量 (kg)	25
最大初始尺寸 (mm)	600*600*500
裁判系统	小装甲/测速/场地交互/主控/电源管理/灯条模块

1.1.1.3 比赛机制

若出现以下情况时，机器人会被扣除血量：发射机构枪口热量超过上限、初速度超过上限、装甲模块被弹丸攻击、装甲模块受到撞击、裁判系统重要模块离线、违规判罚等。裁判系统服务器在结算时，会对所扣除的血量进行四舍五入。

出射速度速度上限为 25 m/s，当速度大于 25 时，每当裁判系统检测到一颗子弹，就会扣除一定的血量。具体扣除量见判罚机制表1.2。

表 1.2 超速判罚机制

V	L
$25 < V < 30$	200
$30 \leq V < 35$	1000
$V \geq 35$	2000

初始状态下，机器人枪口热量为 0。每当其进行一次射击，机器人枪口热量就会相应增加。

“比赛过程中，机器人枪口对应的热量上限为 240，每秒冷却值为 120。当机

机器人的血量低于 400 时，每秒冷却值为 240。枪口热量按 10Hz 的频率结算冷却：每周期热量冷却值 = 每秒冷却值 / 冷却结算频率。”^[1] 若 $360 > Q > 240$ ，每周期扣除血量 = $(Q - 240) * 4$ ，扣血后结算冷却。若 $Q \geq 360$ ，立刻扣除血量 = $(Q - 360) * 40$ ，扣血后令 $Q = 360$ 。

装甲模块是通过压力传感器并结合装甲板震动频率检测伤害源。装甲模块的最小检测间隔为 50ms。弹丸需以 12m/s 或以上的速度接触装甲模块受攻击面才能被有效检测。机器人在装甲模块受到撞击时会受到伤害，但是不允许通过撞击（包括与机器人冲撞、抛掷物体等）的方式造成对方机器人血量伤害。在无攻击力增益的情况下血量伤害值数据，具体规则如图1.2。

攻击类型	血量伤害值
弹丸	<ul style="list-style-type: none"> ● 0 号装甲模块: 20 ● 1/3 号装甲模块: 40 ● 2 号装甲模块: 60 装甲模块 ID 相关内容，详见“ 4.4 装甲模块 ID 编号设置 ”
暴击	当且仅当机器人被攻击的顺序满足如下条件： <ul style="list-style-type: none"> ● 在 8 秒内，有任意两块装甲，临时称之为装甲 A 装甲 B。 ● 这两块装甲被攻击的次序为，装甲 A 被连续攻击 3 次——装甲 B 被连续攻击 3 次——装甲 A 被连续攻击 3 次，则触发暴击伤害，该机器人将被额外扣除 150 的血量。  <ul style="list-style-type: none"> ● 撞击无法影响暴击的触发。 ● 当一次暴击产生后，将重新统计装甲模块被攻击的次序。
撞击	10

图 1.2 装甲攻击扣血机制 (引自: [RMUA](#))

以上机制是本项目重点关注的部分，除此之外还有许多别的机制，但因为与本项目无关，故不予说明。

1.1.2 YOLOv5

YOLOv5 的轻量化高效率等特点使其非常契合机器人视觉场景。对于 Robo-Master 智能挑战赛来说，比赛中机器人的识别速度就是硬实力，无论机器人的策略有多完美，如果射击部分不能对敌方目标造成有效打击，那么也无法顺利赢下比赛。在激烈的比赛角逐中，每一秒都尤为关键，如果识别算法能跑到更高的帧

率，那么相对来说机器人就更加灵活，反应速度更为快捷。在更短的时间内识别到敌人并迅速展开攻势对这场比赛来说尤为重要。这也就是我采用 YOLOv5 算法的一大原因。

1.1.2.1 网络结构

YOLOv5 网络结构主要分四大部分：输入端、BackBone、Neck 和 Prediction。本文将基于 YOLOv5s 网络结构进行简要介绍，其网络结构如下图1.3所示。

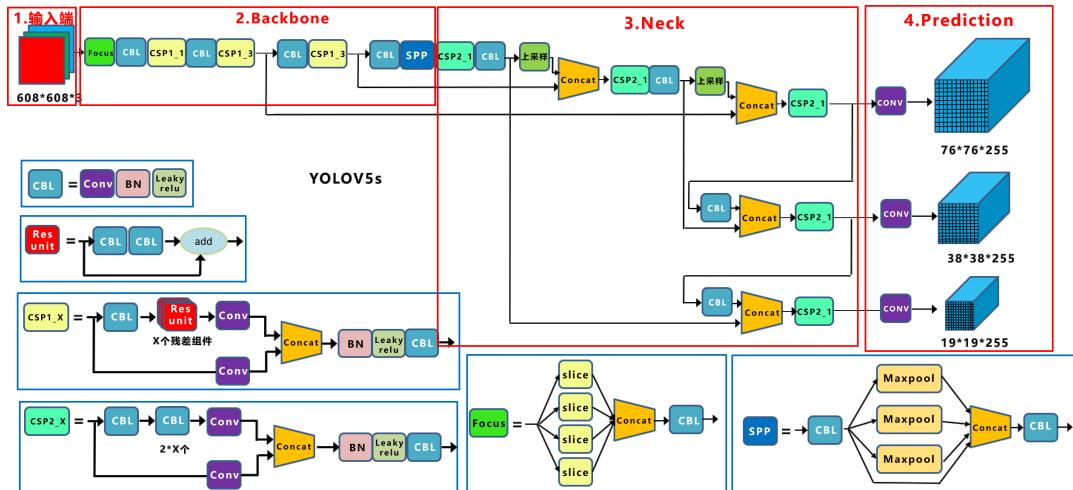


图 1.3 YOLOv5s 网络结构图 (引自：Jiangdabai)

1. 输入端

1) Mosaic 数据增强 Mosaic 是对于 CutMix 数据增强方法的改进与迭代，相比前者将两张图片加以拼接，Mosaic 采用了四张图片，并且运用了随机缩放、裁剪、排布的方式。

采用 Mosaic 数据增强的主要意义在于解决小目标的识别问题。在大多数情况下，小目标相较于中和大目标来说在的 AP 值上会相对低许多。实际上我们常用的数据集中有很多都是以小目标为主的，比如说 COCO 等等。Mosaic 有以下优势，首先，其通过对随机选取的 4 张图片进行随机缩放、分布以及拼接，可以对数据集进行有效的补充，尤其是通过随机缩放，很多小目标在这个阶段得到了补充，网络的鲁棒性得以提升。除此之外，使用其训练时，我们可以一次性对 4 张图片数据进行计算，大为降低了对于 GPU 的要求。

2) 自适应锚框计算 这一部分的核心思想就是让初始锚框先大致在可能的位置框出来目标，之后再在这个基础上对其进行调整。在实际训练中，YOLO 基

于初始锚框得到预测框，然后将其与真实框比对，并算出两者之间的误差，继而反向更新，对参数进行迭代。YOLOv5 在每次训练时，对于不同的训练集，都会自适应的计算其最佳的锚框值。

3) 自适应图片缩放 在大多数目标检测算法中，对于图片各自长宽不同的问题，通常会通过将原始图片统一缩放到一个标准尺寸来解决。对此，YOLOv5 加以了改进，这也是其推理速度更快的一个原因。YOLOv5 的开发者认为，在实际情况中，由于许多图片长宽比不同的因素，在缩放填充后，会使得其两端的黑边大小不尽相同。并且，假如填充过多，大量的冗余就会拖慢推理速度。因此他们通过对 letterbox 的改进，能尽可能的减少给图片添加的自适应黑边，从而使得目标检测速度得到提升。开发者实际测试得出，通过这种简单的改进，推理速度可以得到百分之三十七左右的提升，效果显著。

2.BackBone

1) Focus 结构 这个结构是 YOLOv5 全新加人的，在 YOLOv3 和 YOLOv4 中并不存在，其核心是切片操作。

以 YOLO5s 为例，输入是 608*608*3 的图像，经由切片操作后，转化为 304*304*12 的特征图，经由 32 个卷积核的卷积操作后，最终得到一个 304*304*32 的特征图。对于不同结构的 YOLOv5 网络来说，它们之间在 Focus 结构上的主要区别在于卷积核数量的不同，更复杂的网络结构会使用更多的卷积核。

2) CSP 结构 YOLOv4 网络结构中，借鉴了 CSPNet 的设计思路，在主干网络中设计了 CSP 结构。YOLOv5 与 YOLOv4 不同点在于，YOLOv5 中设计了两种 CSP 结构。

以 YOLOv5s 网络为例，CSP_1X 出现在 Backbone 中，CSP_2X 出现在 Neck 中。CSPNet^[2] 核心目标是从网络结构设计的角度解决推理中较大计算量的问题。通过 CSP 先将基础层的特征映射分为两部分，然后经由跨阶段层次结构进行合并，经过对网络中重复的梯度信息进行优化，使其在保证准确率的同时也大大降低计算量，节省内存开销。采用 CSP 结构的主要优势在于增强网络学习能力，使其轻量与准确兼得。

3.Neck

1) FPN+PAN 结构 这一层结构主要通过在目标检测中融入特征金字塔，提高目标检测准确率，尤其是小目标的检测准确率。这其实是一个经典的取舍问题，对于高层的特征来说，其拥有丰富的语义信息但缺少准确的位置信息，低层恰好相反。如果将低层和高层的特征相互补充，就能做到识别和定位兼得。

FPN 是自顶向下的结构，而 PAN 与 FPN 相反，是自底向上的。FPN 层通过将高层的特征经过上采样进行传递融合，来自顶向下传达强语义特征。而 PAN 则能够自底向上传达强定位特征，两者互相补充。这种设计灵感源于图像分割领域的 PANet^[3]，通过将其应用到 YOLOv5 中，进一步提高了 YOLO 的特征提取能力。

4.Prediction

1) Bounding box 损失函数 YOLOv5 中采用了 CIOU Loss 作为 Bounding box 损失函数。

Bounding Box Regression 的损失函数的发展历程大致为 Smooth L1 Loss、IoU Loss、GIoU Loss、DIoU Loss 和 ClIoU Loss。

IoU Loss 最为简单，就是单纯的交集除以并集，但其存在两个问题：首先，当预测框和目标框没有交集时，此时 IOU=0，损失函数不可导，网络无法训练。其次，在存在相同大小的两个预测框的情况下，IOU 也相同，无法对两者的相交情况进行区分。

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (1-1)$$

因此在 2019 年推出了 GIOU Loss 来改进这一问题。GIOU Loss 通过增加了相交尺度的衡量方式，来缓解单纯运用 IOU Loss 遇到的问题。虽然相比 IoU Loss，GIoU Loss 可以在任意情况下进行训练，但它还是不完美的，当预测框在目标框内部时，如果预测框之前的大小相同，那么不管位置如何，GIoU Loss 都是一个定值，此时 GIOU 遇到了和 IOU 一样的尴尬境地，无法区分相对位置关系。

$$GIoU = IoU - \frac{|A_c - U|}{|A_c|} \quad (1-2)$$

于是 2020 年，DIoU Loss 应运而生。它考虑了重叠面积和中心点距离。其中， b 表示预测框的中心， b^{gt} 代表真实框的中心， ρ 是两个中心点的欧式距离。

c 是预测框和真实框最小外接矩的对角线距离。由于 DIOU 缺少对于长宽比问题的考量，故当预测框被目标框包裹时，如果预测框的中心点位置均相同，那么参照 DIOU 的定义，三者的值均相同。

$$DIOU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} \quad (1-3)$$

为了解决这个问题，CIoU Loss 在 DIoU 的基础上考量了长宽比的影响。其中 α 是权重函数，而 v 用来度量长宽比的相似性，定义为 $v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$

$$CIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (1-4)$$

2) nms 非极大值抑制 在目标检测的后处理过程中，一个相同的识别目标附近往往存在大量的冗余锚框，这时就需要通过 nms 来加以筛选。YOLOv5 采用了加权 nms 的方式来处理冗余。这对于一些遮挡重叠的目标，这会带来不错的提升。

1.1.2.2 YOLO 相关应用

实际上 YOLO 作为优秀的视觉解决方案，已经涌现了大量机器人领域的应用，比如应用在机器人采摘苹果定位^[4]，无人机实时人体检测^[5]，柔索并联机器人移动构件快速定位^[6] 等等。越来越多的人愿意将 YOLO 引入到机器人领域来解决视觉问题，这也坚定了我使用 YOLO 完成这次比赛的信心。

1.1.3 视觉伺服 (Visual Servoing)

1.1.3.1 简介

随着计算机视觉和图像处理技术的发展，视觉传感器已经成为各种机器人的主要感知装置。有了摄像头加持，感知能力增强的机器人可以完成各种复杂的任务。在过去的几十年里，基于视觉伺服的机器人一直是领域最热门的研究课题之一。作为这个领域的开创者，Francois 等人给研究界提供了开创性的基于位置和基于图像的视觉伺服策略的教程^{[7][8]}。

近年来，移动机器人，因其工作空间广、操作方便而得到了广泛应用。结合视觉传感器的移动机器人视觉伺服系统可以说是当今最具吸引力的研究课题之一。与机器人机械手^{[9][10]} 和无人机^{[11][12]} 等其他伺服平台相比，与该项目类似

的轮式移动机器人作为典型的欠驱动系统，伺服控制器设计会相对更加复杂，更加具有挑战性。

在移动机器人的视觉伺服中，视觉传感器通过采集外部环境的信息，进而对图像加以分析，从而得到移动机器人的状态，以进行后续的运动。姿态调节^{[13][14]}和轨迹跟踪^{[15][16]}一般用于移动机器人，它们可以应用于许多领域，如智能驾驶、制造、家居服务等等。姿态调节使得移动机器人可以达到所需的姿态，这常常运用于自动泊车等任务中。轨迹跟踪帮助移动机器人沿着给定的轨迹进行运动。对于视觉移动机器人来说，由于各个模块的特殊性，在制定机器人运动控制策略时要解决许多问题。对于单目视觉传感器，一个主要的问题是由于相机的投影模型而导致的景深不足。因此，移动机器人工作空间和机器人状态的重构是非常困难。在约束视野之后，视觉目标和地标应保持在视野范围内以获得可靠的视觉信息反馈。同时，视觉测量也很容易受到投影参数和外部照明的影响。另一方面，移动机器人平台一般为非完整平台，由于存在非完整约束，难以设计合适的非线性控制器。因此，由于视觉反馈和移动机器人运动约束的不确定性，该控制系统可谓是典型的不确定非完整控制系统，这给控制器的开发带来了很大的挑战。

1.1.3.2 移动机器人视觉伺服方法

现有的移动机器人的视觉伺服方法可分为两部分：视觉感知系统和伺服任务的非线性控制。在视觉系统中，可以根据先验模型、光照强度、多视图几何将图像处理分为三类。此外，视觉不确定性是由多方面因素造成的，包括视觉参数，遮挡，动态场景和 FOV 约束等等。另一方面，针对固有的非完整约束，利用非线性控制理论设计控制律。与传统的基于位置、基于图像以及混合视觉伺服控制器的分类方式不同，我们根据控制目标对控制器进行分类，这对于其他视觉伺服平台也是可行的。伺服任务需要解决输入饱和、目标可视性等诸多挑战。此外，控制器设计中还应考虑系统的不确定性，包括参数未知和状态不可用等问题，这些问题一般可以通过补偿模型和观测模型进行处理。移动机器人视觉伺服技术还可以进一步与智能模块相结合。这样可以使得基于视觉的移动机器人系统对外部环境具有更好的感知能力，控制系统更具有鲁棒性，从而更加高效可靠。

1.1.3.3 总结

总的来说，对于移动机器人来说，视觉伺服主要包括运动控制和视觉感知两个大方面。视觉信息通常是利用计算机视觉中的多视图几何来处理的，也可以通过直接方法和分析多种特征来处理，同时要考虑不确定性因素，如遮挡、动态

对象等问题。在运动控制方面，一般采用非线性控制理论实现姿态调节和轨迹跟踪，解决了非完整运动约束以及其他约束。为了进一步发展，视觉伺服目标需要高性能控制器，并通过多模块集成，来提高精度和系统功能。并且通过学习算法，增强其在不同环境中的适应性。在本研究课题中，视觉伺服的主要作用在于识别追踪敌人并控制机器人云台的运动状态。

1.2 系统架构及软硬件说明

1.2.1 系统架构

该项目的系统架构如图1.4所示。

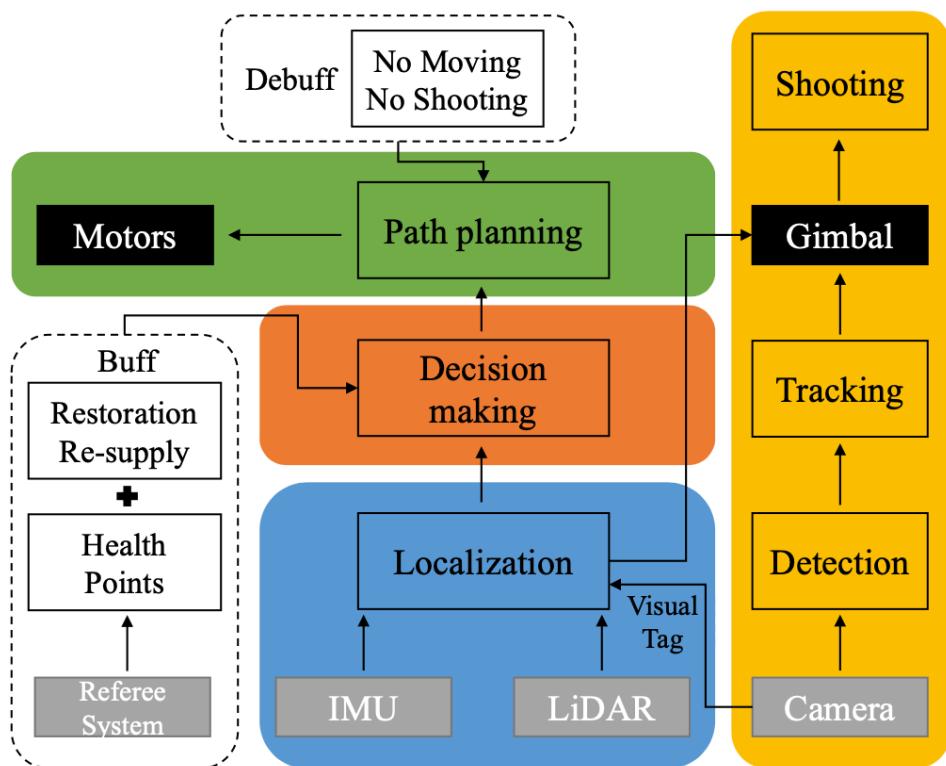


图 1.4 系统架构

本文重点关注的是右侧黄色区块，即视觉相关部分。由图可见，视觉部分在整个系统中相对独立，与其他区块的交互主要是通过定位模块来获取敌人的位置和距离信息。这种较高的独立性也能大大提高视觉模块运行时的鲁棒性。

1.2.1.1 机器人操作系统（ROS）

ROS 简介 ROS (Robot Operating System) 是一种专为机器人设计的操作系统。其通常由大量节点组成，各个节点通过发布和订阅消息来互相通信。比如在我这

个项目中，机器人的雷达单元就是一个节点，雷达单元可以发布其获取的信息，其它节点如导航单元、射击单元，可以通过订阅这个消息来获取雷达信息，加以使用。

ROS 平台的特点及优势 ROS 有许多优势，比如提高代码复用率，同时得益于其分布式的处理框架，每个部分都可以单独进行设计并且在运行时能够松散地耦合。它作为一个操作系统来说十分灵活，它的节点可以位于不同的计算机上，可以位于不同的网络上，每个节点所用的编程语言也可以不同。上述灵活性使得 ROS 可以适应很多不同场合的应用。这就是我选用 ROS 操作系统完成该项目的原因。

1.2.2 软件

表 1.3 软件要求

操作系统	机器人操作系统	YOLOv5 版本与环境
Ubuntu18.04	ROS Melodic	3.0 版本 (Python 3.7)

如表1.3所示，首先在软件层面，我们选用的是 Ubuntu18.04 操作系统，并采用了当前比较流行的 ROS Melodic 机器人平台。其中 Melodic 基于 python2.7 环境，YOLOv5 基于 Anaconda 中的 python3.7 环境。

1.2.3 硬件

机器人计算单元采用的是英特尔 NUC Enthusiast Kit-NUC11PHKi7C，并使用定制的 3D 打印部件将 LiDAR、摄像头、IMU 与炮塔集成在一起。使用该方案的主要原因是确保即使在较快的底盘旋转时，激光雷达和摄像头也能保持稳定，并通过万向节电机反向旋转以校正该角度。该系统与外部 IMU 相结合，有助于防止快速旋转导致的数据漂移和定位失败。由于机器人的身体原本挡住了与炮塔相连的激光雷达的一半范围，因此在机器人的后部放置了一个额外的激光雷达传感器。我们将来自两个传感器的信号合并在一起，以创建我们的 360° 视图。

1.2.3.1 硬件架构

该项目的硬件架构如图1.5所示。

1.2.3.2 传感器

该项目用到的传感器如图1.4所示。

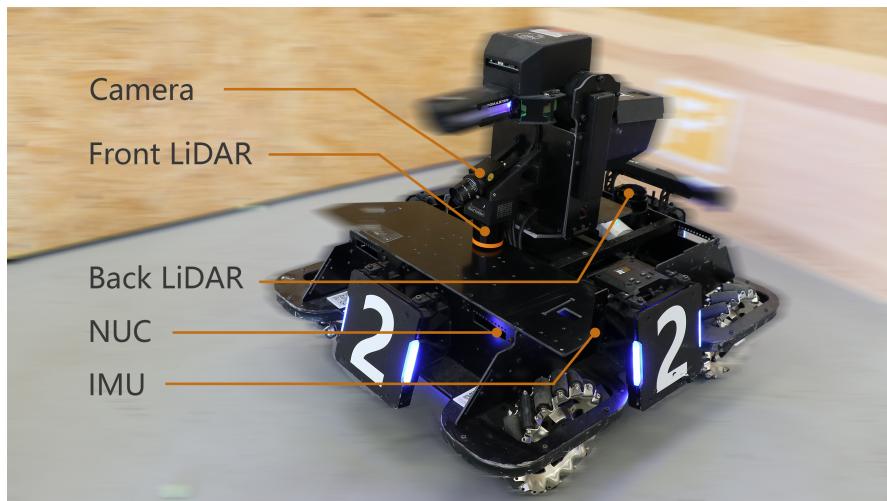


图 1.5 硬件架构图

表 1.4 传感器

传感器	型号	数据频率
前置雷达	Hokuyo UST-10LX	40Hz
后置雷达	RPLiDAR S1	15Hz
前置摄像头	MV-SUA134GC-T	200Hz
OPS 摄像头	MV-SUA134GC-T	200Hz
IMU	Xsens MTi-G-710	400Hz

第二章 目标识别与追踪

YOLOv5 能有效提高目标识别效率和精准度，能让机器人在实战中更快速的识别敌人装甲板，更早实现有效打击。同时，借助一些主流的追踪算法，可以进一步降低计算成本，提升识别帧数。

2.1 敌方单位识别

2.1.1 工业相机设置

相机采用了 MV-SUA134GC-T 工业相机，因为只支持手动对焦，故选用了小光圈来换取更大景深，这样可以尽可能保证远景和近景都不会虚焦。相机驱动方面使用了官方驱动，并配合官方文档的 API 接口进行设置。其中，RGB buffer 大小直接按照相机的最大分辨率来分配。从相机传输到 PC 端的是 RAW 数据，在 PC 端通过软件 ISP 转为 RGB 数据。调整为性能模式后，最终相机可以达到 110FPS 左右的输出效率，并向 YOLO 节点传输消息。

2.1.2 模型训练

模型训练部分一共采用了 350 张图片，分辨率统一调整到了 640*480，这些数据取自不同的距离、角度、曝光、颜色（红色或蓝色或无色）、数字（1 或 2），并且其中涵盖了受到遮挡的情况。具体分布情况见表2.1，训练集示例见图2.1。

表 2.1 数据集分布

	红色 1	红色 2	蓝色 1	蓝色 2	死亡 (无色)
训练集	60	60	60	60	60
验证集	10	10	10	10	10

YOLOv5-3.0 版本的官方代码中，提供了四个版本的目标检测网络，分别是 YOLOv5s、YOLOv5m、YOLOv5l、YOLOv5x。在保持模型尽可能轻量化并且同时获得高识别率和识别速度的前提下，我选取了 YOLOv5s、YOLOv5m 两个网络模型进行实验。希望在保证识别准确度的情况下获得一个更轻量的模型。在经过实际情况验证时发现，YOLOv5s 模型虽然能很好的分辨红蓝色的 1 与 2 标志，并且运行帧率很高，但无法对无色即死亡的情况进行分辨。故最终采用了 YOLOv5m 网络模型。训练结果见图2.2。

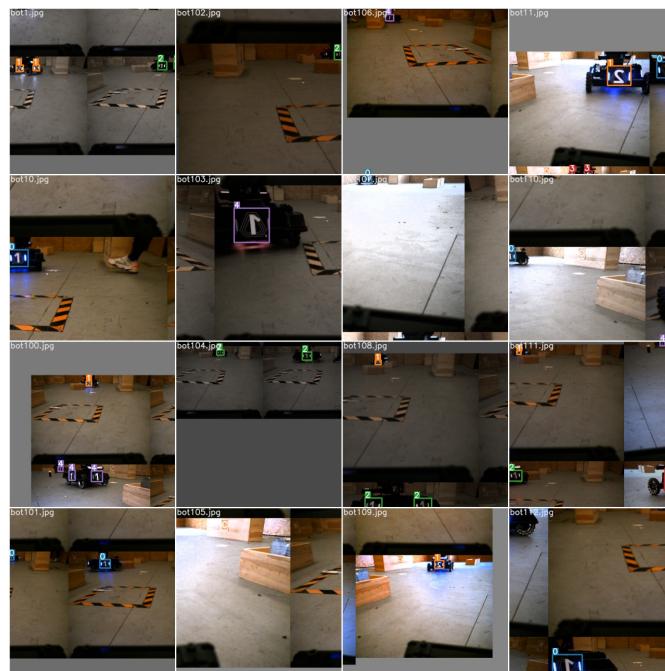


图 2.1 训练集示例

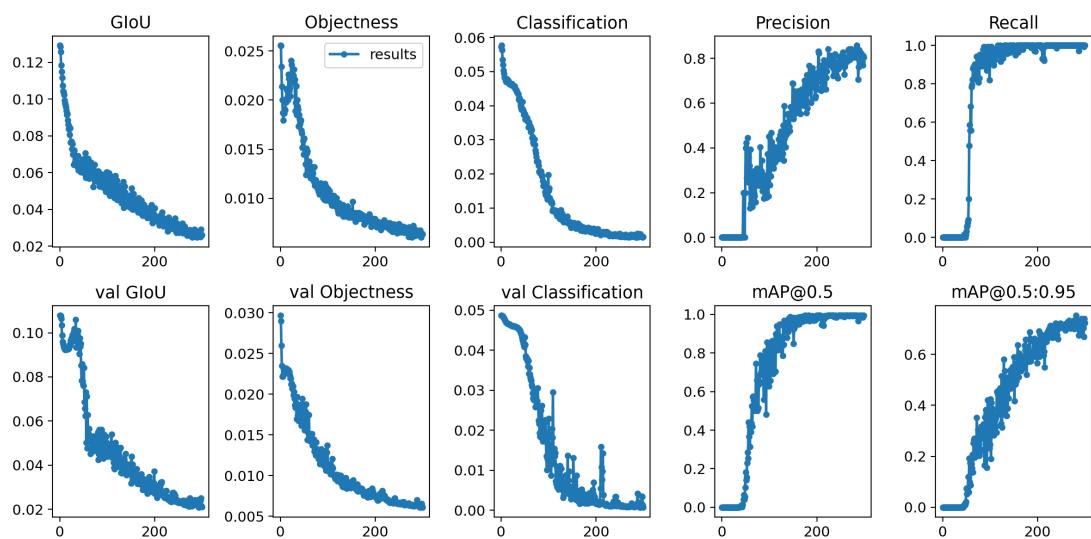


图 2.2 训练结果

YOLOv5 使用 GIoU Loss 作为 bounding box 的损失，该实验中 GIoU 最终达到了 0.02 左右，故最终的检测框位置的精准度可以得到保障。Objectness 值也达到了合理的标准，目标检测的精准度较高。Classification 低于 0.01，分类准确度高。最后，精度、召回率都达到了较为理想的值，mAP 值波动较小。综上，训练结果比较优秀。

2.1.3 实际测试

在实际测试的环节中，不添加任何优化的情况下，YOLOv5 检测的帧率可以达到 60FPS 左右。我将在接下来的环节中继续对其进行优化。

2.2 目标追踪

2.2.1 敌方单位追踪

在追踪方面，为了增加推理速度和模型泛化能力来更好应对敌方机器人装甲受遮挡时的情况。我选用了两种追踪算法，KCF^[17] 和 MedianFlow^[18]。当检测节点通过传感器获取图像队列时，我可以动态调整用于预测的图像和用于追踪的图像之间的比例。KCF 算法基于 Boosting（例如 AdaBoost^[19]）跟踪器和 MIL^[20] 跟踪器，推理速度很快，但缺乏于对物体受到遮挡情况的应对能力。MedianFlow 算法很好地解决了这个问题，但对快速移动的对象不友好。在经过实际测试后发现，MedianFlow 算法在实际使用中虽然能有效预判被遮挡的情况，但因为较慢的推理速度，会导致炮台旋转不及时和抖动现象，故最终采用了较为稳定的 KCF 算法。在采用 KCF 算法后，最终整个识别追踪单元帧数达到了 110FPS。

第三章 射击控制

通过多种方式计算与敌方目标的距离，减少距离测量的误差。

3.1 敌方单位距离推算

3.1.1 获取敌人距离的目的

在比赛中，决定机器人射击角度的主要因素是云台结构。云台一共有两个自由度，一个是水平方向的 yaw 轴，一个是垂直方向的 pitch 轴。云台通过接收视觉模块传递过来的 Box 锚框数据来判断敌人在当前画面内的位置，从而控制云台改变 yaw 轴参数以对准敌人。然而对准敌人并不代表炮台可以精准击中敌人的装甲板，控制 pitch 选择合适的俯仰角才能确保精准度（炮弹出射速度不变的情况下）。当然，想要获取 pitch 值就需要机器人拥有一个距离到 pitch 的映射，很明显，较远的敌人就要选用更大的仰角，近处的敌人则需要较小的仰角。实际上，比赛场地并不大，所以这对距离数据的精确度要求很高，故获取敌人的精确距离十分重要。

3.1.2 视觉测距

3.1.2.1 Perspective-n-Point(PnP) 算法

那么先从视觉方面入手，对相机进行标定，获取其内参矩阵和畸变系数。接下来，就可以用 PnP 算法来解出平移矩阵（tvec）。PnP 算法的原理见图3.1。

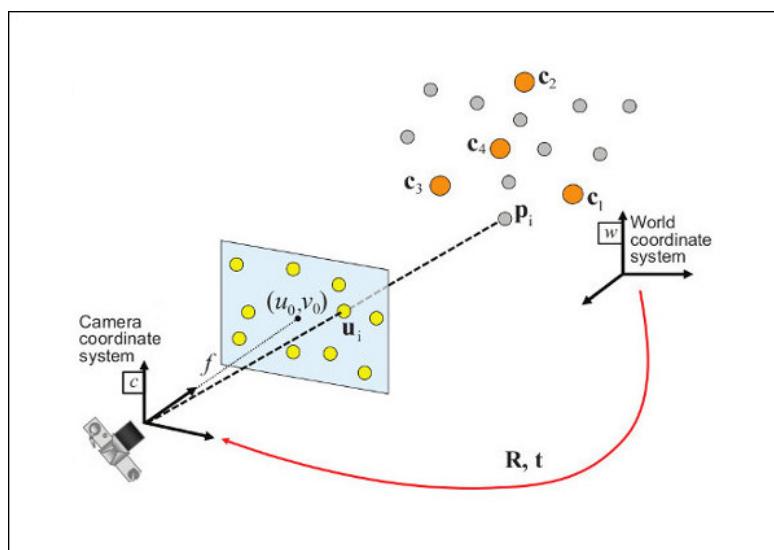


图 3.1 PnP 示意图 (引自：opencv.org)

总的来说，PnP 问题就是在已知世界坐标系下 N 个空间点的真实坐标以及它们在图像上的投影的情况下，如何获取相机的位姿。通过将世界中的坐标点与成像点进行对应，就能通过它们的投影关系，获取相机位姿。在这个问题中，N=4，即识别框四个点的坐标。相机的位姿主要包括两点，一个是旋转矩阵，一个是平移矩阵。在当前问题中，我只想要得到与敌人的距离数据，所以只需要得到平移矩阵。这样就从 3D-2D 匹配对中求得了摄像头位姿。

3.1.2.2 另一种视觉测距思路

这种方案类似 PnP 算法但是更加简便，其核心思路如下。在 YOLO 识别过后，我们可以获得一个 640*480 的 image 和目标装甲板的识别框。因为我们相机的俯仰角在随云台转动时不会变化，所以对于一个物体，在云台转动时在 image 中的 y 坐标不会变化。基于这个基本认识，我们可以将识别框中心位置的 y 坐标映射到距离。y 轴的有效范围为 0-480，比如识别框中心 y 坐标为 10 时，意味着识别框位于 image 靠上的位置，对应敌方目标在距离我方较远的位置。相反的当 y 坐标为 450 时，识别框位于 image 靠下位置，对应较近的敌方目标。

通过拟合 y 轴坐标与距离的关系，我得到了如下关系图3.2。

在实际测试中，这种测距方式在中近距离效果十分理想，远距离精准度一般，但不容易受到目标移动或者旋转的干扰，抗干扰能力强。

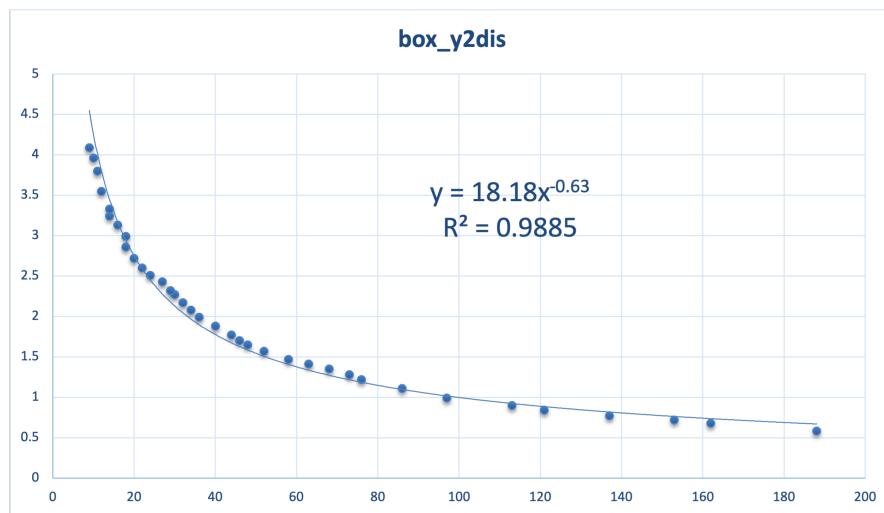


图 3.2 y 轴与距离拟合图 (x: y 轴坐标, y: 距离)

3.1.3 LiDAR 测距

本项目选用的前置雷达是 Hokuyo UST-10LX，雷达探测角度范围是-135° 到 +135°。为了尽可能减少周边障碍物等物体对雷达扫描结果的干扰，我只选取了

靠近中心的五个波束作为参考。且仅当五束波中的每一束有有效读数，即不为无穷大时将其放入 scan_buffer 中。最后将 buffer 中的数据取中位数，即可得到较为准确的距离数据。

3.1.4 结合视觉和 LiDAR 测距提高鲁棒性

视觉方法得到的距离数据与 LiDAR 得到的数据各有优劣。视觉算法虽然精度不高，误差一般较大，但能确保所得到的数据一定是到敌方装甲板的距离。反观 LiDAR，数据虽然精准，但不能确保这一点。事实上，虽然经过了之前的优化处理，雷达还是有可能读取到错误目标的距离数据。所以将两者数据结合分析十分重要。

大部分情况下，雷达数据出错的主要情况是云台没有完全对准敌人的装甲板，导致波束读取到了敌人后方的障碍物。这种情况下，雷达的距离数据会有一个大幅度的跳变，但视觉得到的距离数据不会。在实战中，这会导致机器人的云台 pitch 轴不断上下抽搐，对射击的稳定性造成相当大的影响。为了避免这种情况，我先对雷达数据进行一次滤波，去除大部分抖动的数据。除此之外，机器人每次都会将新读取到的 LiDAR 和视觉距离数据与之前的比较。如果视觉方法得到的距离没有较大抖动但是 LiDAR 数据发生抖动，就会将数据先记录入一个缓冲区，而不是直接交给云台来改变 pitch 轴，只有在连续读取到多次这样的数据直至将缓冲区填满时，机器人才会认可这个新的距离数据，并交给云台处理。这样距离数据的可靠性得到了保证，云台抖动问题也得到了解决。

3.2 弹道分析

获取了敌人的距离数据后，下一步就是建立距离-云台 pitch 的映射了。这里我采用了两种方案，并分别进行了实际测试。

3.2.1 牛顿法

归根结底，机器人的射击运动就是一个斜抛运动，所以其遵守斜抛运动轨迹方程：

$$y = x \tan \theta - \frac{gx^2}{2(V_0 \cos \theta)^2} \quad (3-1)$$

其中， V_0 为抛出速度， θ 为速度与水平面夹角， g 为重力加速度， x 为水平距离， y 为垂直距离。

在本项目中， V_0 为已知的炮台发射初速度， g 为当地重力加速度， x 为已经获取到的距离数据， y 为炮台出射点到装甲板的高度。于是问题便转化为，已知 V_0 、 g 、 x 、 y ，求 θ 。这时，就可以借助牛顿法的思想，尝试通过迭代不断逼近 θ 的值。

然而，在实际测试中，这一方案得到的精准度并不高，因为现实环境和理论的理想物理环境的差别，实际执行上遇到了很多问题。风阻、炮台发射机构的摩擦、炮台的抖动等都会导致牛顿法得出的结果与实际有较大差别，除此之外，牛顿法远程的命中率较低，所以牛顿法最终并未被采用。

3.2.2 拟合法

为了解决之前牛顿法所不能兼顾的问题，最终我采用了拟合法。拟合法一共分为三个步骤（数据表采用 12m/s 的情况演示）：

1. 采集：获取大量的距离-pitch 样本作为原始数据，如图3.3所示。

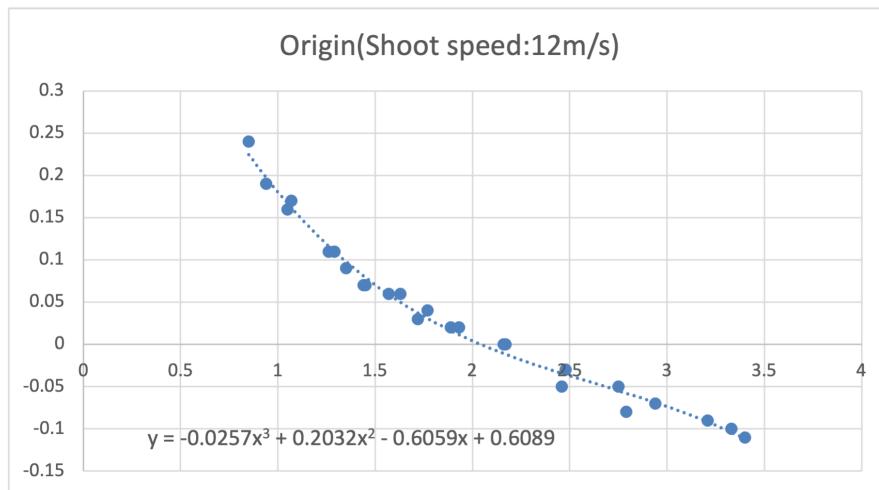


图 3.3 原始数据图 (x: 距离, y: pitch 值)

2. 过滤：根据平均误差（根据多项式拟合结果计算）过滤离散点（明显噪声点），结果如图3.4所示。

3. 插值：采取线性插值的策略来增加数据量，最终得到的拟合结果如图3.5所示。

对于 25m/s 的情况我也进行了拟合，结果如图3.6所示。

经过实际测试发现，对于不同的出射速度，机器人最终都能够达到不错的命中率，可以有效的对敌方装甲板进行打击。

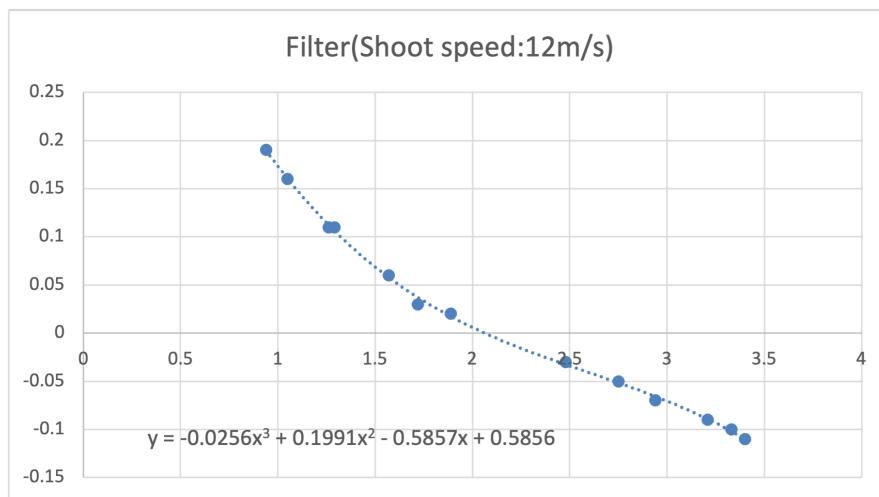


图 3.4 过滤离散点后数据图 (x: 距离, y: pitch 值)

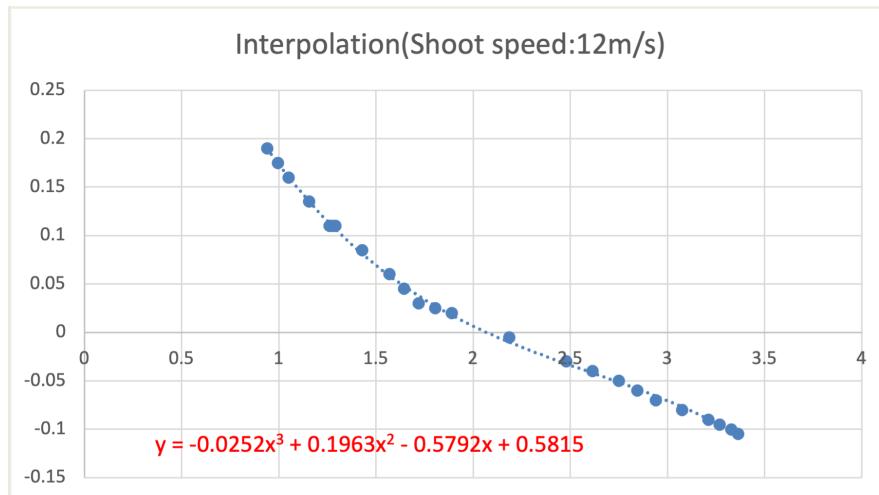


图 3.5 最终拟合结果图 (x: 距离, y: pitch 值)

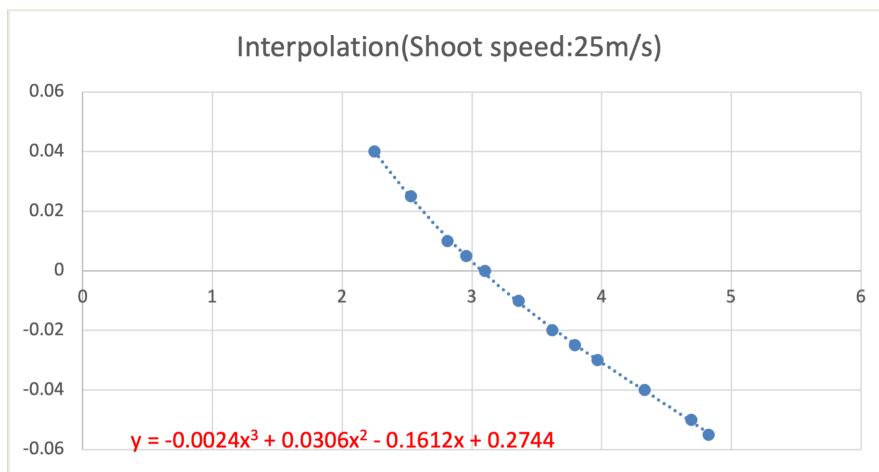


图 3.6 出射速度为 25m/s 时的拟合结果 (x: 距离, y: pitch 值)

第四章 策略及控制优化

借助弹道分析，动态射速和射频控制以及 PID 调整，能使得机器人在交战过程中获得更大的优势。

4.1 策略优化

4.1.1 动态射速及射频控制

参照比赛规则，不同初速度的子弹会消耗不同的能量值，如果机器人单位时间内消耗的能量值超过回复值，那么就会淘汰。所以，赛事方实际上是对射速和射频做出了限制。高速的子弹虽然会减少下落更容易击中远处移动的敌人，但是得使用更低的射频。低速的子弹虽然不容易射中远处的敌人，但射频可以很高。所以为了优势兼得，我选用了动态控制。机器人内部设定了两条拟合曲线，分别对应了高低速情况下的距离—pitch 关系。这样在敌方距离低于 3 米时，机器人就会采用低射速高射频的方案，这样确保近距离精准度的情况下尽可能打出更多的伤害。远距离时，采用高射速低射频，确保了远距离打击的精准度。

对于子弹初速度的设定，可以通过更改机器人摩擦轮转速来调整。具体实现方式可通过调用 /cmd_fric_wheel 服务来达成。经过测试发现，当设定 /cmd_fric_wheel 服务中的摩擦轮转速参数为 1307 时对应 12m/s 的出射速度，1450 时对应 25m/s 的出射速度。

对于射击频率的设定，可以通过调用 /cmd_shoot 来更改。实际测试得出，当出射速度为 12m/s 时，应该采用 2500 的射频，25m/s 时用采用 1000 的射频。

4.2 控制优化

4.2.1 云台 Pitch 与 Yaw 轴控制优化

整个射击模块的主要运作流程是从 Camera 节点发布原始 image 信息到 YOLO 节点，经过 YOLO 节点识别后将识别框 Box 数据发布到 Autofire 节点。Autofire 在整合收到的数据综合处理后，就可以通过 ROS 向 STM 板发送指令来调整 pitch、yaw、摩擦轮转速、射频以完成整个射击流程。但是实际测试中发现，如果 Autofire 直接发送原始 pitch 和 yaw 指令，不经处理的 pitch 和 yaw 数据会导致云台两个自由度的抖动，射击效果很不理想。虽然已经在之前通过结合 PnP 和 LiDAR 进行了初步去噪，但还远远不够。

4.2.1.1 Yaw 轴控制优化

对于 yaw 轴，我们运用了 PID 控制算法进行调整，公式如下：

$$U(t) = kp(err(t) + \frac{1}{T_I} \int err(t)dt + \frac{T_D derr(t)}{dt}) \quad (4-1)$$

简单来说，当得到系统的输出后，算法将输出经过比例，积分，微分 3 种运算方式，叠加到输入中，从而更好的控制系统的行。具体代码实现如下：

Algorithm 1 PID 部分代码

```

1 def convert_error_space(self, x):
2     x = float(x)
3     pixel_error = self.image_width / 2 - x
4     error = (self.image_width / 2 - x) / self.image_width * 3.14 * 0.36
5     pencent = (self.image_width / 2 - x) / (self.image_width / 2) * 100
6     return error
7
8 def box_center(self, box):
9     return (box.xmin+box.xmax)/2, (box.ymin+box.ymax)/2
10
11 def yawController(self, box):
12     centerx, centery = self.box_center(box)
13     errorx = self.convert_error_space(centerx)
14     self.Ierror += errorx
15     if self.Ierror > 1:
16         self.Ierror = 1
17     if self.Ierror < -1:
18         self.Ierror = -1
19     yaw_angle = (errorx) * self.P + self.D * (errorx - self.last_error) + self.Ierror * self.I
20     self.last_error = errorx
21     return yaw_angle

```

在该项目中，errorx 指识别框的中心的 x 轴坐标与当前云台视野即 640*480 视野范围中心的偏移量。经过调参之后，最后确定 $P = 0.45$ 、 $I = 0.0125$ 、 $D = 1.0$ 。在这系列参数下，yaw 轴能很稳定快速的运行。

4.2.1.2 Pitch 轴控制优化

对于 pitch 轴，主要面临的问题还是距离数据的噪声。因为在实际运行中，雷达不能保证一直保持正对敌人，因为 yaw 轴并不能百分百保证对于敌人的锁定。此时，雷达就有可能读到较远距离的墙壁或者别的错误距离数据。虽然我们之

前已经针对这个问题做了一步优化，但是为了进一步降低抖动，我又对 yaw 轴控制指令进行了一个时间尺度上的去噪，通过一个 buffer 对于极小一段时间内的 pitch 解算值进行记录，然后取它们的中位数。这个方法在实际运行过程中取得了很好的效果，机器人 pitch 轴异常抖动的频率大大降低，射击的稳定性有所提高。

第五章 结论

虽然受到疫情影响，最后我们没能如愿参加 RoboMaster 比赛，但是我负责的视觉射击部分已经优化到了很不错的效果。在经过一系列的识别追踪算法的优化后，机器人的识别帧数能稳定在 110FPS 左右，机器人云台也能流畅迅速的定位并且跟踪目标，机器人能正确的识别敌方和友方目标，对于遮挡的目标也能稳定的预判和识别。在射击方面，云台的 pitch 轴能根据敌方目标距离，选取合理的射击角度、出射速度和射击频率，对其造成有效的伤害。因为隔离在寝室内，无法前往实验室，很多资料和数据也难以获取，无法更加直观的呈现，后期也无法进一步加以优化，更无法进行模拟战进行测试。但是在这个项目的过程中，无论是视觉部分还是射击部分，都是很有趣的过程。不断优化识别精准度和速度，提高云台追踪灵敏度，优化射击弹道，在这个过程中，看着原本不灵敏，伴随着奇怪故障的机器人慢慢变得灵巧，有莫大的成就感，于我而言，这就足够了。

参考文献

- [1] DJI. Robomaster 机甲大师高校人工智能挑战赛比赛规则手册[EB/OL]. 2022. <https://www.robomaster.com/zh-CN/resource/pages/announcement/1370?djifrom=rmua1>.
- [2] WANG C Y, LIAO H Y M, WU Y H, et al. CspNet: A new backbone that can enhance learning capability of cnn[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390–391.
- [3] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759–8768.
- [4] 赵德安, 刘晓洋, 赵宇艳. 基于 yolo 深度卷积神经网络的复杂背景下机器人采摘苹果定位[J]. 农业工程学报, 2019, 35(3): 164.
- [5] BOUDJIT K, RAMZAN N. Human detection based on deep learning yolo-v2 for real-time uav applications[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2021: 1–18.
- [6] 訾斌, 尹泽强, 李永昌, 等. 基于 yolo 模型的柔索并联机器人移动构件快速定位方法[J]. 机械工程学报, 2020, 55(3): 64–72.
- [7] CHAUMETTE F, HUTCHINSON S. Visual servo control. i. basic approaches[J]. IEEE Robotics & Automation Magazine, 2006, 13(4): 82–90.
- [8] CHAUMETTE F, HUTCHINSON S. Visual servo control. ii. advanced approaches [tutorial] [J]. IEEE Robotics & Automation Magazine, 2007, 14(1): 109–118.
- [9] WANG H, GUO D, XU H, et al. Eye-in-hand tracking control of a free-floating space manipulator[J]. IEEE Transactions on Aerospace and Electronic Systems, 2017, 53(4): 1855–1865.
- [10] ZHANG Y, LI S. A neural controller for image-based visual servoing of manipulators with physical constraints[J]. IEEE transactions on neural networks and learning systems, 2018, 29(11): 5419–5429.
- [11] ZHANG X, FANG Y, ZHANG X, et al. A novel geometric hierarchical approach for dynamic visual servoing of quadrotors[J]. IEEE Transactions on Industrial Electronics, 2019, 67(5): 3840–3849.
- [12] ZHAO W, LIU H, LEWIS F L, et al. Robust visual servoing control for ground target tracking of quadrotors[J]. IEEE Transactions on Control Systems Technology, 2019, 28(5): 1980–1987.
- [13] FANG Y, DIXON W E, DAWSON D M, et al. Homography-based visual servo regulation of mobile robots[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2005, 35(5): 1041–1050.
- [14] FANG Y, CHAO-LI W. Adaptive stabilization for uncertain nonholonomic dynamic mobile robots based on visual servoing feedback[J]. Acta Automatica Sinica, 2011, 37(7): 857–864.
- [15] DIXON W E, DAWSON D M, ZERGEROGLU E, et al. Adaptive tracking control of a wheeled

- mobile robot via an uncalibrated camera system[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2001, 31(3): 341–352.
- [16] QIU Y, LI B, SHI W, et al. Concurrent-learning-based visual servo tracking and scene identification of mobile robots[J]. Assembly Automation, 2019.
- [17] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 37(3): 583–596.
- [18] KALAL Z, MIKOLAJCZYK K, MATAS J. Forward-backward error: Automatic detection of tracking failures[C]//2010 20th international conference on pattern recognition. : IEEE, 2010: 2756–2759.
- [19] FREUND Y, SCHAPIRE R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of computer and system sciences, 1997, 55(1): 119–139.
- [20] BABENKO B, YANG M H, BELONGIE S. Robust object tracking with online multiple instance learning[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 33(8): 1619–1632.

致 谢

非常感谢 Lima Lab 所有一起备战 RoboMaster 的同学，虽然受疫情影响，指导老师变动，几经波折，大家也都一直在勤勤恳恳地为比赛做准备。特别感谢余琛、辜俊和霄竹学长对于我的指导和帮助，解决了我很多对于机器人软硬件的困惑并且给我提供了很多新思路。此外也感谢指导老师 Sören Schwertfeger 和 Andre Rosendo，他们在我完成这篇论文的过程中给了我很多帮助和建议，特别是 Sören Schwertfeger 教授在指导老师变动的时候向我伸出了援手，给予我了完成这篇论文的动力。