

算法- 视觉 组

VISION

感知 · 识别 · 自瞄



让机器看见，
让目标无处可藏。



坚逾磐石#5

01

视觉组是做什么的

DETECT

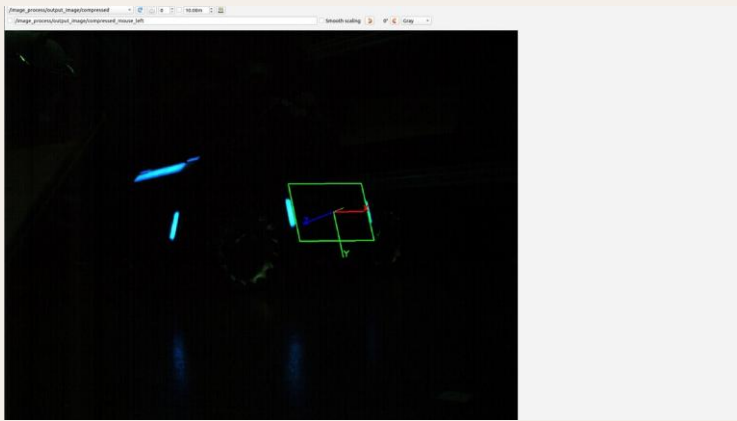
PREDICT

PERCEIVE

AIM

视觉组主要负责让机器人获得稳定的感知能力：从相机与雷达采集信息，完成目标识别、位置解算、运动预测，并将结果交给机器人执行。

工作内容覆盖装甲板识别、自瞄预测、能量机关、飞镖定位、哨兵全向感知与雷达全场识别。



视觉调试界面

装甲板识别

识别目标、类别与装甲板状态。

目标预测

估计目标未来位置，提高命中率。

雷达全图感知

识别并追踪赛场中的机器人目标。

特殊任务

能量机关、飞镖定位与哨兵感知。

视觉组的主要功能

自瞄、能量机关、飞镖与雷达



自瞄外挂

自动识别敌方装甲板，并根据目标运动状态进行预测，让机器人持续、稳定地瞄准目标。

能量机关

识别并解算能量机关的运动规律，辅助操作手完成击打，为全队获得增益。

飞镖定位

定位敌方基地与目标位置，为飞镖系统提供更可靠的瞄准与发射依据。

雷达全场视野

通过高位相机与识别算法掌握全场目标位置，为哨兵、无人机和操作手提供信息支持。

Demo · 自瞄 “外挂” 的实机效果演示



03

雷达与全场感知



雷达硬件

雷达硬件由相机、计算平台与安装结构组成，负责从高处持续采集赛场画面，为识别、定位和追踪提供稳定输入。

全图识别

全图识别结果将目标信息叠加在场地视角中，帮助队伍掌握各机器人位置与运动状态，为战术判断和系统联动提供依据。



RADAR

03

PERCEPTION

TRACKING

COMMAND

04

组内日常与联调

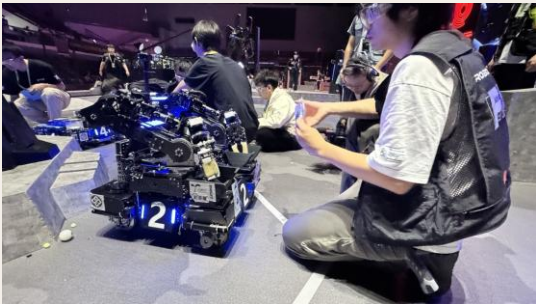
采集、标注、训练、部署，再回到赛场继续调试。



赛场调试



在赛场



05

自学建议

先建立开发基础，再逐步进入计算机视觉与机器人实战。

01 C++ / Python

C++是视觉工程的主要编程语言，Python 常用于脚本、数据处理和深度学习实验。

02 Linux / Ubuntu 24.04

视觉代码主要运行在 Linux 环境；目前队内使用 Ubuntu 24.04 LTS，完成系统安装与环境配置是上手项目的重要一步。

03 Git / ROS2

使用 Git 进行版本管理和团队协作；利用 ROS2 完成模块化开发、通信与调试。

04 OpenCV / 机器 视觉

学习图像处理、相机模型、目标检测与传统机器视觉基础。

05 机器学习 / 深度 学习

建立统计学习和神经网络基础，进一步理解识别、分类与预测模型。

尝试并了解多种算法的优势与特点，
逐步理解每一个算法为什么有效。

06

RESOURCE

INDEX

TOOLS

C++

Python

OpenCV

ROS2

Git

学习资料

基础

01 C++ / cppreference

<https://en.cppreference.com/w/>

02 Python

<https://www.liaoxuefeng.com/wiki/1016959663602400>

03 OpenCV

<https://docs.opencv.org/4.x/>

04 深度学习

<https://zh-v2.d2l.ai/>

工程工具

05 ROS2

<https://docs.ros.org/>

实战项目

https://gitlab.com/rm_vision

https://github.com/CSU-FYT-Vision/FYT2024_vision

建议仔细研究队内代码，至少理解装甲板解算中的卡尔曼滤波、打符号的最小二乘拟合等基本原理

