

華中科技大學

# 本科生毕业设计[论文]

题 目：基于机器视觉的轨迹识别及位置标定

院 系 机械科学与工程学院

专业班级 机械 1308 班

姓 名 李含嫣

学 号 U201310795

指导教师 宋 宝

2017 年 6 月 6 日

## 学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者簽名: 年 月 日

# 学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理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。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密□，在 年解密后适用本授权书

2、不保密口。

(请在以上相应方框内打“√”)

作者签名: 年 月 日

导师签名: 年 月 日

## 摘 要

近年来,机器人技术的发展被越来越多的国家所高度重视,机器人的应用也开始从传统的工业领域走向人类生活的方方面面。亚太大学生机器人大赛正是在这样的环境下应运而生。本课题是基于 2017 年大赛赛题“舞盘雅乐”进行的参赛机器人视觉系统的方案设计,这是对新型机器人技术的一项十分有意义的探索。

本课题主要研究机器视觉技术在机器人对外部环境信息获取的应用。根据比赛需求,本课题针对不同任务介绍了不同的方法:针对定位,采用提取场地引导白线的方法对机器人位置进行标定和校正;针对清台,采用区域划分和颜色搜索的方法对敌方飞盘进行识别;针对轨迹,采用帧间差分提取运动目标、采用多项式拟合对轨迹点归类,从而对运动飞盘进行追踪。

通过实验测试,以上研究内容均取得了比较理想的效果。

**关键词:** 深度视觉; 白线提取; 重定位; 背景分离; 目标跟踪

## Abstract

In recent years, great importance has been attached to the development of robot technology by more and more countries. And the application of robots is beginning from the traditional industrial areas to all aspects of human life. In such an environment, the Asia-Pacific Robot Contest comes into being. Positioning and motion control scheme of the energy-saving unpowered robot based on the "lauch frisbee", the title of the contest in 2017, is researched in this paper. It is a new type of robot technology and is very meaningful to explore.

The subject is mainly about the research of the machine vision technology in the application for robot to aquire external environment information. According to the demand, the subject contains the different methods for different task: for relocation, a method of extracting the white line of the field is used for calibration and correction of the robot's position; For platform cleaning, a method of zoning and color search is used for frisbee identification;for tracking frisbee, using interframe difference to extract moving objects, using polynomial fitting to track the points, thus tracking the movement of frisbees.

Experiment is done, and the above researches achieve relatively good results.

**Key Words:** Depth perception; White line extraction; Relocation; Background subtraction; Objects tracking.

# 目 录

摘 要.....	II
Abstract .....	III
1 绪论 .....	1
1.1 课题来源 .....	1
1.2 课题背景与意义 .....	2
1.3 国内外研究现状 .....	3
1.4 主要设计内容与章节安排 .....	4
2 整体方案设计 .....	5
2.1 机械结构 .....	5
2.2 硬件平台 .....	6
2.3 控制系统 .....	7
2.4 视觉系统设计 .....	7
3 白线提取重定位 .....	9
3.1 图像采集和预处理 .....	10
3.1.1 图像采集 .....	10
3.1.2 运动模糊的消除 .....	10
3.1.3 图像预处理 .....	10
3.2 基于差分拟合的白线提取 .....	11
3.2.1 提白线的方案比较 .....	11
3.2.2 差分拟合的原理及算法实现 .....	12
3.2.3 差分拟合的算法实现 .....	13
3.3 坐标转换计算偏移量 .....	14
3.3.1 像素坐标转毫米坐标 .....	15
3.3.2 毫米坐标转机器人坐标 .....	15
3.3.3 机器人坐标转世界坐标 .....	15
3.4 重定位模块测试 .....	16
4 静态飞盘识别清台 .....	18
4.1 图像采集和预处理 .....	20

4.1.1 相机选型 .....	20
4.1.2 图像采集与预处理 .....	21
4.2 基于广度优先的连通域柱子提取 .....	22
4.2.1 种子点的选取 .....	22
4.2.2 广度优先的遍历 .....	24
4.3 基于区域划分和颜色的飞盘识别 .....	26
4.3.1 台面范围的划分 .....	26
4.3.2 飞盘的识别 .....	27
4.3.3 角度计算清台 .....	29
4.4 清台模块测试 .....	30
<b>5 动态飞盘轨迹跟踪 .....</b>	<b>31</b>
5.1 基于三帧帧差的运动目标提取 .....	33
5.1.1 运动目标提取的方案比较 .....	33
5.1.2 三帧帧差法的原理及算法实现 .....	34
5.1.3 基于轮廓提取的飞盘形心计算 .....	34
5.2 基于多项式拟合的轨迹跟踪 .....	35
5.2.1 多项式拟合轨迹方程 .....	36
5.2.2 轨迹点分类及多目标追踪的实现 .....	37
5.3 飞盘轨迹模块测试 .....	38
<b>6 总结与展望 .....</b>	<b>39</b>
<b>致 谢 .....</b>	<b>40</b>
<b>参考文献 .....</b>	<b>41</b>
<b>附录 .....</b>	<b>错误!未定义书签。</b>

# 1 绪论

## 1.1 课题来源

本课题题目是 CCTV 机器人大赛——基于机器视觉的轨迹识别及位置标定，这是 2017 年 ABU ROBOCON 国内选拔赛华中科技大学 HUST ROBOCON 团队参赛机器人视觉控制系统的研究。

ROBOCON 比赛每年有不同的主题,2017 ABU ROBOCON 大赛主办方日本把“舞盘雅乐”确定为今年的比赛主题，比赛要求参赛双方各制作一台机器人，利用比赛提供的 50 个飞盘，击落 7 个不同着陆台上的充气沙滩球，然后将飞盘发射到各着陆台上。如果所有球已被从着陆台上击落且某一参赛队成功地使其飞盘落在所有着陆台上且每个着陆台上至少有一个停止运动 3 秒钟后未掉落的飞盘，该队就获胜。若 3 分钟内没有队伍完成比赛目标，则根据双方的分数判定胜负。比赛赛道详情见图 1-1。

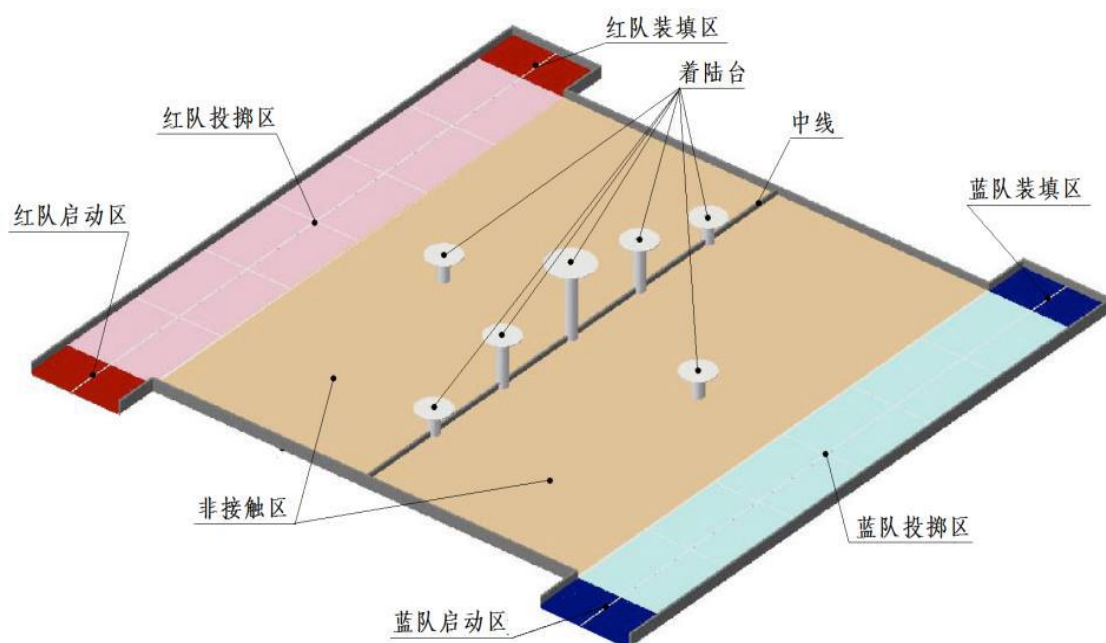


图 1-1 比赛场地的 3D 图

具体比赛规则见表 1-1。

表 1-1 比赛规则

机器人设置	1. 开始比赛前, 进行机器人的设置, 时间不超过 1 分钟
	2. 参赛队伍进行机器人的设置的队员不超过 3 个
	3. 如果没有在 1 分钟内完成设置, 可以在比赛开始后继续进行设置
初始状态	1. 停止设置前必须将机器人完全纳入启动区并保持静止
	2. 赛方提供的 50 个飞盘放在装填区, 码放方式由参赛队自定
	3. 未经裁判员允许, 队员不得进入比赛场地
装载飞盘	1. 机器人必须在完全进入装填区且裁判允许时才能开始装弹
	2. 装填时可以手动也可以使用夹具和弹舱, 但受到尺寸、重量限制
发射飞盘	1. 当机器人与投掷区接触且与其它区域没有接触时才可以投掷飞盘
	2. 比赛中装上机器人而掉落的飞盘不可重复使用
得分	1. 优先实现所有目标则获胜
	2. 比赛结束后对无球着陆台上的飞盘计分, 最近着陆台 1 分(与飞盘数量无关), 中线着陆台每个飞盘 1 分, 最远着陆台每个飞盘 5 分
	3. 得分相同时, 在远台得分高的队伍获胜; 远台分数相同时, 得分着陆台多的队伍获胜; 得分着陆台数也相同, 则在中台上得分多的队伍获胜; 以上均相同, 则裁判确定获胜队伍

## 1.2 课题背景与意义

机器人技术作为 20 世纪人类最伟大的发明之一, 自 60 年代初问世以来, 经历 40 年的发展, 已取得长足的进步。机器人的研究与应用水平, 是一个国家经济实力和科技发展的反映。因此, 世界上许多国家, 包括我国在内, 都对机器人的发展予以高度重视。

ABU ROBOCON 倡导“让思维行动起来, 让智慧沸腾起来”。该赛事旨在激发创造的激情、创新的理念, 致力于培养各国青少年及未来的工程师对于独立思考、开发和研制科技产品的兴趣与爱好, 提高各参赛国的科技水平, 为机器人工业的发展发掘、培养后备人才。



本课题研究参赛机器人的视觉系统——基于机器视觉的轨迹识别及位置标定，除了完成比赛本身，这项课题还有两重意义。其一，增强参赛人员对机器人技术的了解，使其掌握一些相关技术，提高动手能力和解决问题的能力；其二，目标跟踪和位置标定在现代机器人和人工智能领域的应用日趋广泛，比如无人机的发展、交通流量的监测、汽车的自动驾驶和辅助驾驶等，机器视觉的影响和作用越来越大。

### 1.3 国内外研究现状

目前，工业领域是机器视觉应用中比重最大的领域，关于位置标定和目标跟踪的研究在国内外都有一定的研究和发展，某些成熟技术已投入实际应用。

基于机器视觉的位置标定，就是利用摄像头信息得到机器人的位置、偏移量，进而进行定位、校正。该技术在无人驾驶车辆等领域具有广阔的应用前景。国外，斯坦福大学(Stanford University)研制的机器人小车斯坦利(Stanley)，使用激光雷达和摄像头扫描前方道路，实现自主导航，能够在沙漠上行驶。国内，上海交通大学自主研发的 Frontier-II 移动机器人，采用全景视觉、前向视觉摄像头、红外测距传感器等构成导航系统，可以完成机器人的目标提取、跟踪，避障等等。



图 1-2 Frontier-II 移动机器人



图 1-3 羽毛球机器人 Robomintoner

目标跟踪也是机器视觉研究领域的一个热点问题，目前，基于视觉的运动目标检测与跟踪已广泛应用于视频监控、智能交通和人机交互等领域。早在 60 年代，国外就已经开展了对运动目标检测与跟踪的研究。比利时的一些研究人员曾研制出一台可以打羽毛球的机器人，研究人员将这台机器人命名为“杰达”。“杰达”

配备有两个摄像头，以让其获得立体视觉，进而检测出羽毛球的位置。同时，利用视觉算法计算出羽毛球的运动轨迹，进而在预测轨迹中合适的点将羽毛球击回。这是世界上第一台会打羽毛球的机器人。而国内，由成都科学家研发的羽毛球机器人 Robomintoner 已经有了长足进步，该机器人来源于 2015 届 Robocon 大赛——“羽毛球双雄”，现已发展为一款上市羽毛球健身陪练机器人。Robomintoner 运用高精度滑轨式挥拍支架与快速机械运动底盘、高精度快速移动目标视觉跟踪算法、机器人实时位置校正等技术，能达到一般羽毛球爱好者的基本水平。

## 1.4 主要设计与章节安排

本课题针对参赛机器人的视觉系统进行研究和设计，在机器人跑动中利用场地上的引导白线信息，进行位置标定、校正，简称重定位；在比赛中实时监测己方着陆台上是否有对方飞盘，一旦检测到立即进行清台；跟踪运动的飞盘，进行轨迹拟合和判断。

第一章介绍课题的来源、背景与意义以及国内外的研究现状。

第二章介绍机器人的整体方案设计，从机械、电路、控制三个方面介绍机器人的总体设计，着重介绍视觉系统设计，对课题采用的软硬件平台和总体方案进行分析。

第三章介绍重定位的实现，利用对场地引导白线的提取，根据白线、十字的全场坐标，计算机器人的偏移量，从而进行位置偏差校正。

第四章介绍对己方着陆台上的对方飞盘的识别，用于清台。提取距离最近的柱子（己方着陆台），划出台面范围，然后映射到彩图下根据颜色等信息判断是否有对方飞盘，有则立即实行清台。

第五章介绍对运动飞盘的跟踪，主要用到背景分离技术和轨迹拟合技术。

第六章对整个比赛实验结果进行了总结，并提出可以改进的方向。

## 2 整体方案设计

完整的机器人系统包括机械结构、硬件平台和控制系统三个部分。机械结构是控制的实体对象，也是决定控制方案和控制效果的基础；硬件平台是实现软件控制系统的硬件保证；控制系统则是机器人实现各目标的逻辑依据，本课题主要研究控制系统中的视觉部分。本章先介绍整体的三个部分，然后着重介绍视觉系统的方案设计。

### 2.1 机械结构

机器人的机械结构如图 2-1 所示，大体上分为底盘部分、发射部分和装弹部分。

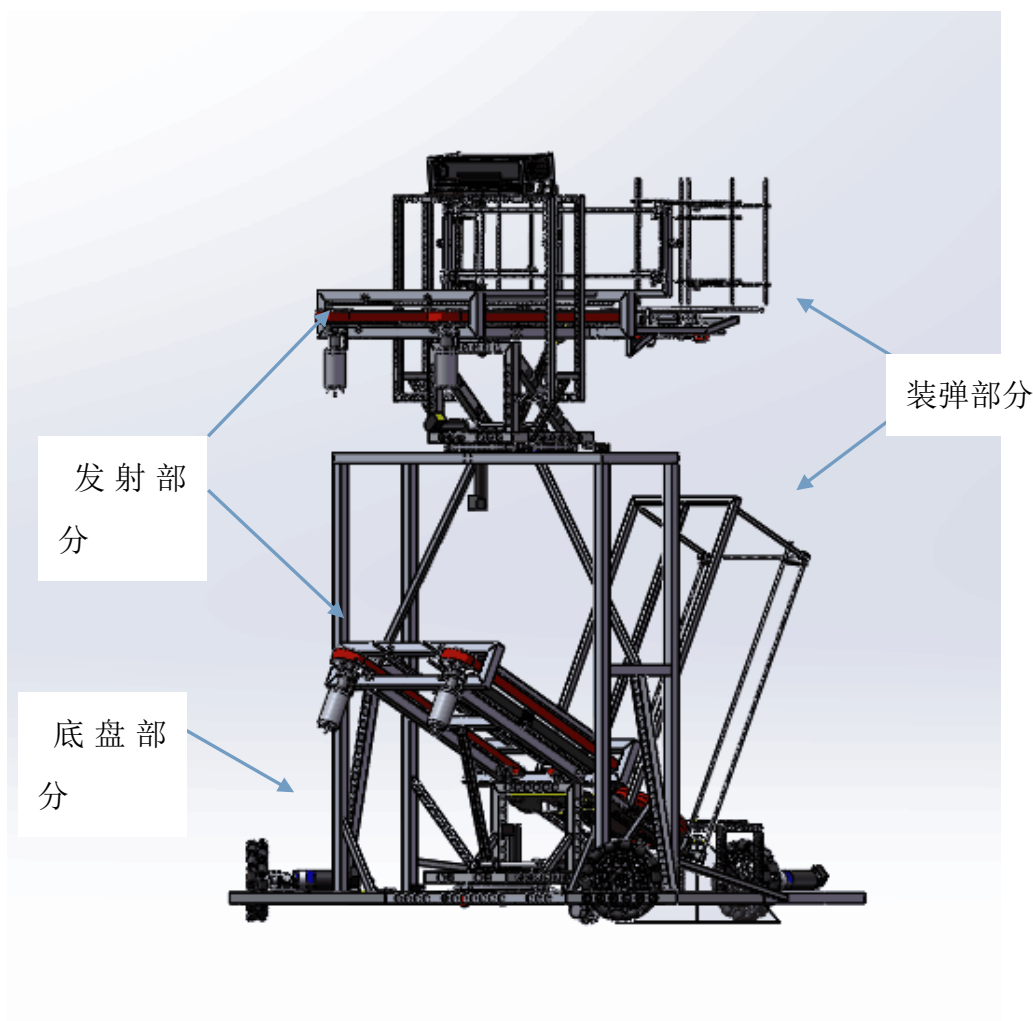


图 2-1 机器人总装图

底盘主要负责跑动，本届比赛赛道简单，机器人只需走直线来回，而要实现飞盘准确落台对位置精度和稳定性要求高，所以采用三轮全向轮底盘。

发射部分分上下两层，由同步带差速原理实现飞盘的发射，双发射机构可以两个机构一起发射飞盘，而且上层发射机构高度高于己方着陆台，可以用于清台。这种双发设计不仅提高了时间利用率，而且为策略多样化提供了实现的基础。

装弹部分也是两个，分别对应上下两层的发射机构，下层装弹机构由同步带实现飞盘的送出，上层装弹机构由推杆实现飞盘的送出。

## 2.2 硬件平台

硬件电路采用了模组化的思想，整个系统分为嵌入式主控，嵌入式从站，供电模块和 PC 四个部分。电路构架如图 2-2 所示。

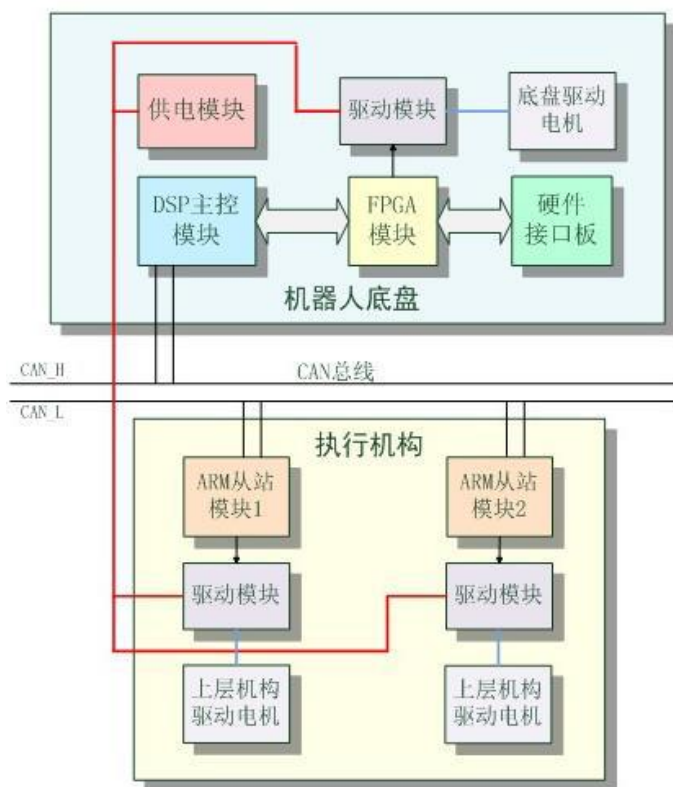


图 2-2 电路构架

嵌入式端，主控为运算和调度的核心，负责接收各传感器的数据，与各嵌入式从站通信，控制各电机的动作规划，实现策略。嵌入式从站包括定位板、各电

机驱动以及舵机驱动等等，定位板负责接收码盘和陀螺仪的数据并实现轨迹规划，用电机和舵机完成机构的动作。从站与主控间通过 CAN 总线进行通信。

PC 端主要完成视觉图像处理，硬件核心为一台 miniPC，主要传感器为用到的两个摄像头。PC 计算能力强，处理速度快，有利于处理视觉的复杂数据；两个摄像头一个装在底盘上获取场地信息，用于对机器人位姿进行重定位，另一个装在机器人顶部，用于获取飞盘信息。PC 端与嵌入式端通过两个串口进行通信。

## 2.3 控制系统

控制系统是实现各机构动作的逻辑依据，可分为底层和上层两部分，底层即嵌入式端程序，上层即 PC 端程序。控制系统框图如图 2-3 所示。

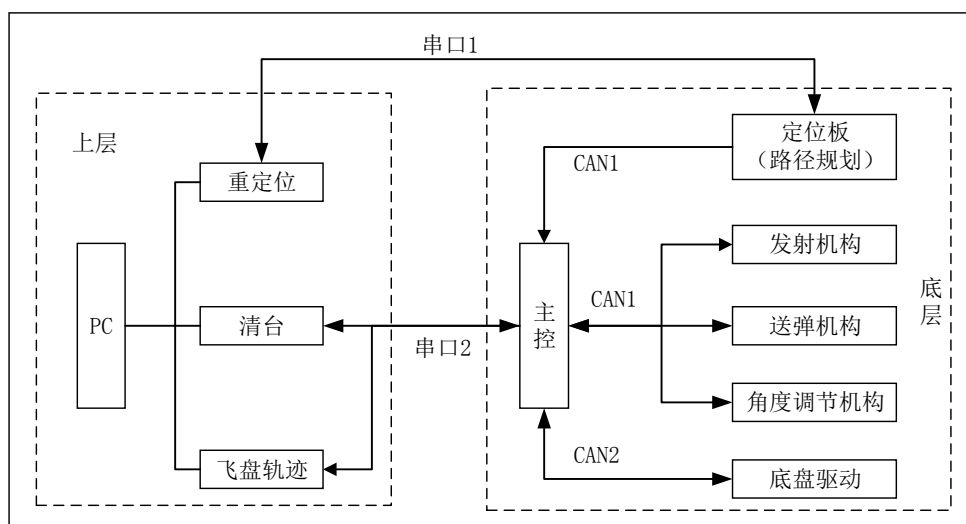


图 2-3 控制系统

底层通过主控控制发射机构发射飞盘、装弹机构送弹、餐盘调节旋转角度以及底盘的跑动，定位板利用陀螺仪及码盘数据进行路径规划；上层利用两个摄像头获取的数据实现重定位、清台以及飞盘轨迹跟踪，其中重定位通过串口 1 与底层的定位板进行点对点的通信，而清台通过串口 2 与主控间进行通信。

## 2.4 视觉系统设计

视觉处理的硬件平台是 PC+两个摄像头，软件平台是 Visual Studio 2013+OpenCV，USB 摄像头用于重定位，Kinect 用于飞盘检测（包括清台和飞盘

跟踪)。重定位用到的底盘位姿信息通过串口通信从定位板获取，也通过串口通信将计算出的偏移量发给定位板；清台和飞盘跟踪的信息通过串口 2 传给主控。

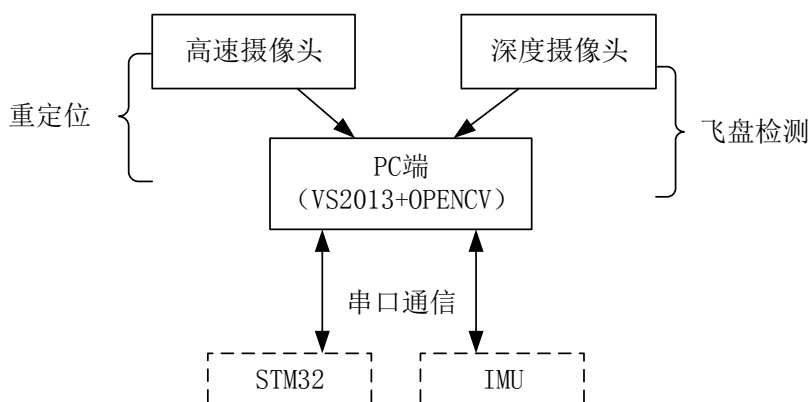


图 2-4 视觉总体设计

重定位时，PC 端通过高速摄像头获取场地白线、十字信息，得到像素坐标；STM32 从码盘和 IMU 中获取机器人当前位置和姿态信息，通过串口通信传到 PC，PC 端就可以进行坐标转换，得到世界坐标下机器人相对白线、十字的偏移量，从而进行位置校正。

PC 端通过深度摄像头对己方着陆台上的飞盘进行监控，一旦发现对方飞盘则立刻实行清台。在调试过程中，进行了飞盘轨迹检测和跟踪的实验，用飞盘的轨迹信息对发射参数进行分析和指导。

### 3 白线提取重定位

白线一般用来做智能机器人的引导线，因为它们具有特征显著，容易识别的优点，用来定位容易实现且精度较高。本章将介绍利用场地的引导白线进行机器人的重定位的实现过程。重定位的框架如图（3-1）所示。

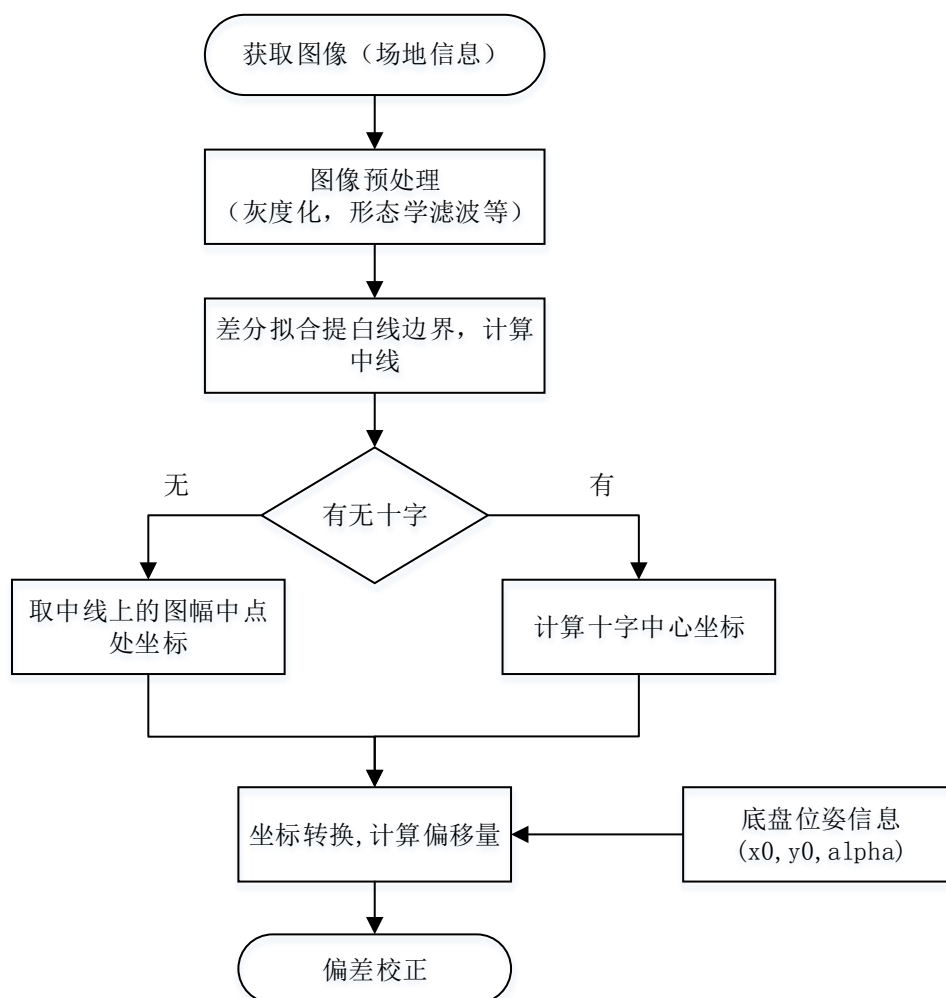


图 3-1 重定位流程图

通过高速摄像头获取场地信息，进行预处理后，采用差分拟合提白线边界的方法得到边界线，计算出中线，在无十字处取中线上一点的坐标进行重定位，在有十字处利用十字中心坐标重定位。利用底盘位姿信息，将像素坐标转换到世界坐标，计算世界坐标系下的两个方向的偏移量，将偏移量传给 STM32 就能进行位置偏差校正。



### 3.1 图像采集和预处理

#### 3.1.1 图像采集

采用的是具有较高帧率的普通 USB 摄像头，采集简单，课题采用 DirectShow 流媒体开发包采集图像。好处是可以人工调节相机曝光等参数，使成像更符合课题需求。

快速运动的底盘会造成运动模糊，场地颜色太浅导致白线不突出，这都将造成边界提取困难。所以先针对这两个问题提出解决办法。

#### 3.1.2 运动模糊的消除

摄像头在正常静止条件下，但车在以较快的速度运动的过程中，摄像头的成像则会变得模糊，这称之为运动模糊。运动模糊的存在使得白线边缘变得模糊不清（如图 3-2），难以进行边沿提取。消除运动模糊的一个最简单的办法就是降低摄像头的曝光时间，但是降低曝光时间之后，摄像头的成像变得很昏暗，必须加上强而均匀的补光才能使成像质量恢复（如图 3-6）。



图 3-2 运动模糊 ( $v=2\text{m/s}$ )

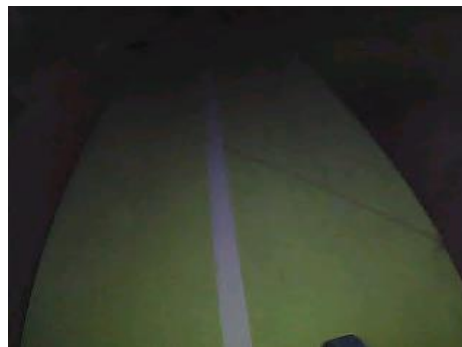


图 3-3 补光后的成像 ( $v=2\text{m/s}$ )

由图 3-2 和图 3-3 对比可以看出，降低曝光时间后补光，能够消除运动模糊对白线边界的影响，从而利于白线的提取。

#### 3.1.3 图像预处理

摄像头所采集回来的是图像是 RGB 320\*240 格式的，3 通道的处理会大大增加算法时间，效率低下，一般转化为灰度图进行处理。由于场地颜色太浅，灰度化后的图像白线很不明显，难于提取，分析比较了各个通道的效果后，最终确定



用 S 通道提红场白线，用 H 通道提蓝场白线。图（3-4）和（3-5）分别为分通道后的效果图。



图 3-4 红场 S 通道

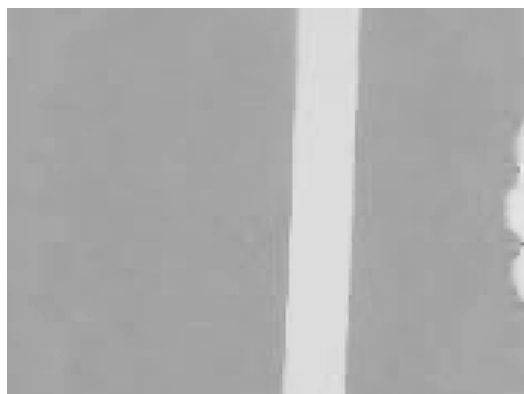


图 3-5 蓝场 H 通道

可以看出，分通道后白线对比度明显变大，有利于边界的提取。

考虑到噪声的干扰，所有的图像必须经过图像预处理才能进行图像信息的提取。结合赛道的具体情况，赛道上主要是一些椒盐噪声和脉冲噪声，而中值滤波对脉冲噪声和椒盐噪声有良好的滤除作用，特别是在滤除噪声的同时，能够保护信号的边缘，使之不被模糊。这样就保证了赛道白线边沿的特征能被很好的保留。虽然中值滤波的算法复杂度比较高，考虑其滤除噪声的效果以及 PC 的计算性能和摄像头的帧率，综合起来并不会对系统的响应周期造成太大的影响。

## 3.2 基于差分拟合的白线提取

白线的提取方案很多，本课题主要对三种方案进行了测试比较：基于霍夫变换的白线提取，基于连通域边界的白线提取，基于差分拟合的边界提取。本节将先对这三种方案进行比较，然后对所确定的方案进行详细分析。

### 3.2.1 提白线的方案比较

#### （1）基于霍夫变换的白线提取

首先，最容易想到的提白线的方法就是霍夫线变换。对图像预处理之后，得到滤波降噪的二值图，然后用概率霍夫线变换提取直线，因为摄像头视野很小，选合适的参数提到的直线就是白线的边界（会找到很多条），无十字处找最左、最右的两条求中点，有十字处除了最左最右的中点，还找最上、最下的两条线的

中点, 这样可以实现两个方向的重定位。此方法存在的问题: 一、概率霍夫线变换提取的直线是斜的, 两边界不平行, 十字处一个正方形提成了梯形, 虽然误差不会很大, 但是定位对精度要求高; 二、霍夫线变换非常耗时, 对于实时性要求很高的底盘快速运动, 满足不了处理速度的要求。

### (2) 基于连通域边界点集拟合直线的白线提取

用 HSV 图仅保留 H 分量提连通域, 可以避免光线以及粉色白色太接近的影响, 白线的 H 值稳定在 60-80 范围内, 拉阈值就能提取出较完整的白线; 将连通域边界存入左、右、上、下四个点集, 对每个点集进行最小二乘法拟合直线。无十字处, 左右边界线求中线, 取图幅中点计算偏差; 有十字处, 增加上下边界线求中线, 两条中线求交点得到十字中心, 计算偏差。此方法存在的问题: 提连通域本质上是依赖颜色的, 会受环境影响, 算法适应性不好; 二、红色粉色过渡段, 摄像头视野会花, 导致第一个十字总检测不到。

### (3) 基于差分边界点集拟合的白线提取

鉴于依赖颜色的白线识别不稳定, 利用了白线的边界特征进行差分提取白线, 不依赖于颜色。直接用灰度图, 按行扫描, 满足差分阈值条件和线宽条件的点分别存入左、右点集, 有十字处再按列扫描, 满足条件的点分别存入上、下点集, 用点集拟合直线求中心。

本课题选择采取差分拟合的办法对白线进行了提取, 利用差分的好处是可以准确快速知道白线在图像中的相对位置, 定位更准更快。

## 3.2.2 差分拟合的原理及算法实现

差分拟合提取白线的原理如图 3-6 所示, 首先利用差分可求取边沿点, 每行扫描得到最大上升沿和最大下降沿两个点, 沿扫描方向依次扫描后可得到两组边沿点, 然后对两组边沿点进行拟合, 得到两条直线, 最后利用这两条直线即可求得白线的中心线。

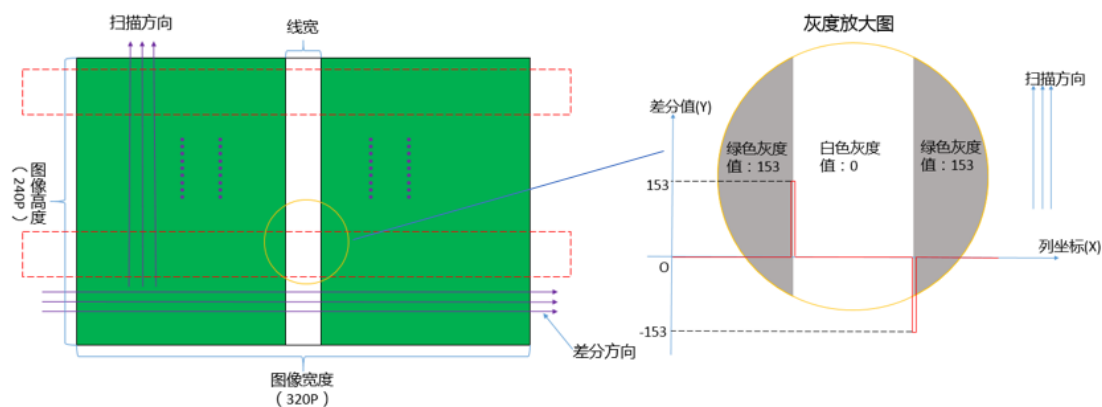


图 3-6 差分拟合提取白线原理

### 3.2.3 差分拟合的算法实现

差分拟合的流程如图 3-7 所示。

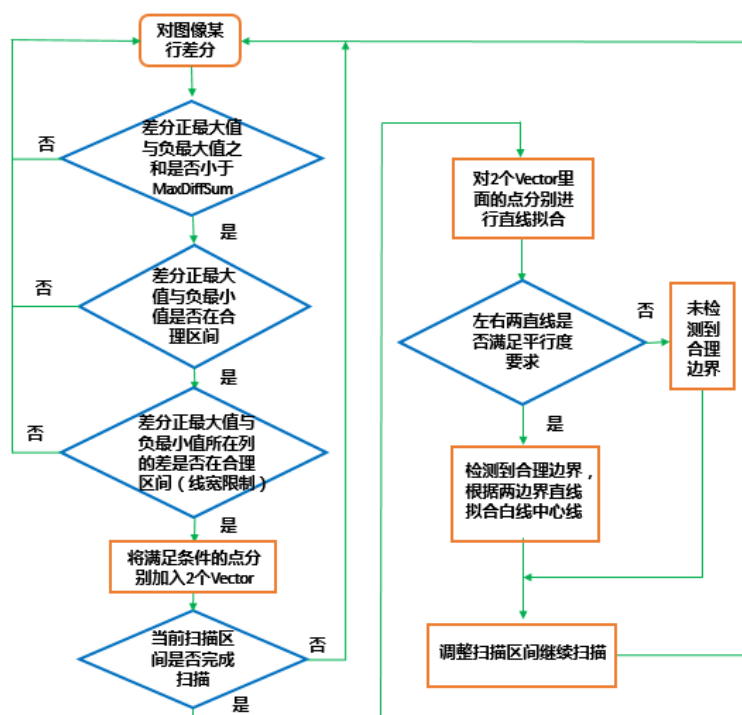


图 3-7 差分拟合流程

为了防止白线误识别,对差分拟合的方法加上一些限制条件。这里本课题提出了 5 个限制条件,其中 1 个是针对单个边沿点的,2 个是针对两个边沿点关系的,2 个是针对拟合的直线。

(1) 差分值绝对值最小限制。白线与周围颜色是有明显区分的,所以灰度值的差分必须大于一定的值,即扫描某一行得到的最大差分值  $diff\_max$  和最小差分值  $diff\_min$  的相反数应该大于一个阈值  $th\_diffv$ 。可用公式 (3-1) 表示。

$$\begin{cases} diff\_max > th\_diffv \\ -diff\_min > th\_diffv \end{cases} \quad (3-1)$$

(2) 差分值之和的限制。由白线的对称性可知,白线上升沿和下降沿的值之和必须接近于 0,即两者之和的绝对值小于一个接近 0 的值  $th\_diffv\_sum$ 。可用公式 (3-2) 表示。

$$diff\_min + diff\_max < th\_diffv\_sum \quad (3-2)$$

(3) 宽度限制。白线都是有一定线宽的,即 2 个边沿点的横坐标之差应当满足一定的宽度条件。即使由于摄像头和白线的夹角造成线宽变大或变小,也不会超出最小阈值  $minwidth$  和最大阈值  $maxwidth$  的范围。可用公式 (3-3) 表示

$$\begin{cases} diff\_min\_x - diff\_max\_x > minwidth \\ diff\_min\_x - diff\_max\_x < maxwidth \end{cases} \quad (3-3)$$

(4) 点数限制。在进行直线拟合前,必须找到足够多的符合条件的点,即找到的边界点数大于一定阈值  $vPointsnum$  时,才进行直线拟合。可用公式 (3-4) 表示。

$$vPoints.size() > vPointsnum \quad (3-4)$$

(5) 角度限制。在成功对差分得到的边沿点进行拟合之后,得到一条竖直中线,因为底盘沿直线跑动,摄像头视野中始终有一条竖直白线,只有当这条直线的倾斜角小于一定的值  $thv\_angle$  时,才会认为白线提取成功,可用公式 (3-5) 表示。

$$abs(vcos\_alpha) < thv\_angle \quad (3-5)$$

### 3.3 坐标转换计算偏移量

提取出白线后,必须进行坐标的计算与转换,才能对机器人位置偏差进行校正。看十字的坐标,与真值相减,就是机器人当前偏移量。

图像处理中有四个坐标系间的关系，像素坐标系，毫米坐标系，机器人坐标系，世界坐标系。下面对它们之间的转换关系进行详细论述。

坐标转换分三步进行，由像素坐标转换到毫米坐标，有毫米坐标转换到机器人坐标，有机器人坐标转换到世界坐标。

### 3.3.1 像素坐标转毫米坐标

以图像左上角为原点，向右为 x 正方向，向下为 y 正方向。从像素坐标转换到毫米坐标用的是投影标定，图像中一段距离对应到实际物理距离是一个比例关系，采用投影标定得到比例系数，就可以进行它们之间的坐标转换，公式（3-6）表示。

$$\begin{cases} dx = \frac{28.0}{35} * xc; \\ dy = \frac{28.5}{36} * yc; \end{cases} \quad (3-6)$$

其中，（xc,yc）为像素坐标，（dx, dy）为毫米坐标。

### 3.3.2 毫米坐标转机器人坐标

像素坐标系和毫米坐标系都是相机坐标系，要转换到机器人坐标系下进行计算，机器人坐标系以机器人中心为原点，正前方为 x 正方向，沿跑道方向为 y 方向，垂直地面向上为 z 正方向。

从相机坐标系到机器人坐标系分平移和旋转两步进行，坐标原点的偏移量在相机安装标定时得到，旋转以机器人 z 轴逆时针为正，旋转角度为陀螺仪数据 ALPHA，所以坐标转换公式为式（3-7）和（3-8）。

$$\text{平移变换: } \begin{cases} xi = dx + 70.5; \\ yi = 112 - dy; \end{cases} \quad (3-7)$$

$$\text{旋转变换: } \begin{cases} xj = -xi * \cos(\alpha) + yi * \sin(\alpha); \\ yj = -xi * \sin(\alpha) - yi * \cos(\alpha); \end{cases} \quad (3-8)$$

其中，（dx,dy）为相机毫米坐标，（xj,yj）为机器人坐标。

### 3.3.3 机器人坐标转世界坐标

世界坐标系建立如图（3-8）：

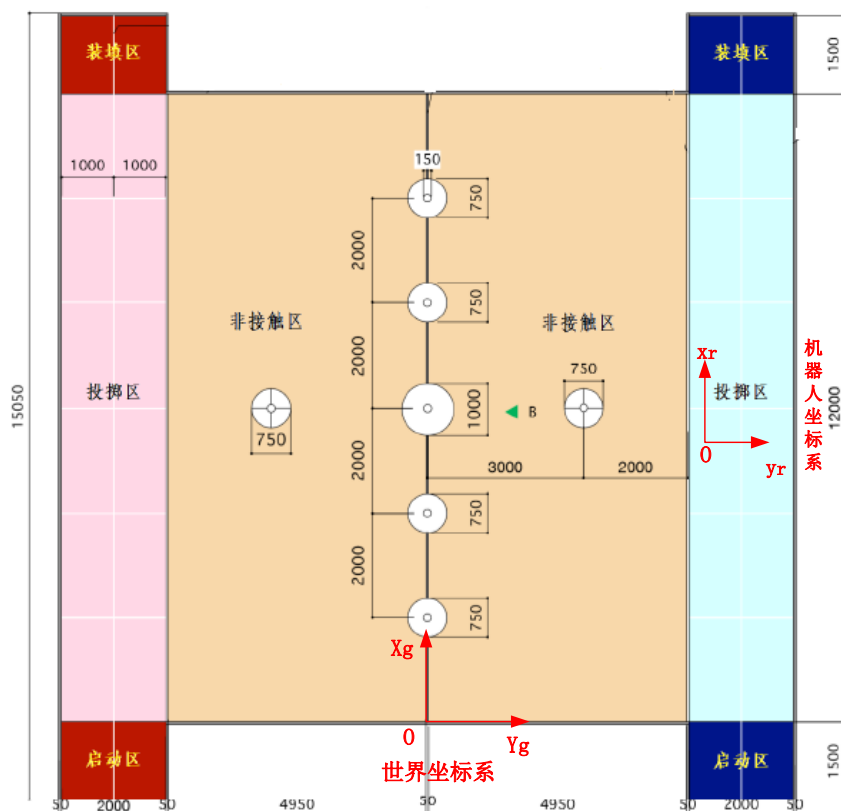


图 3-8 坐标系建立

所以,十字中心的坐标分别为 $(\pm 6025, 3500)$ ,  $(\pm 6025, 5500)$ ,  $(\pm 6025, 7500)$ ,  $(\pm 6025, 9500)$ ,  $(\pm 6025, 11500)$  (蓝场为正, 红场为负)。根据已知的十字全局坐标, 可以计算出机器人相对十字的偏移量, 将偏差发给定位板进行位置校正。

### 3.4 重定位模块测试

#### (1) 测试方案

让底盘分别不带重定位和带重定位从启动区跑到装填区, 测试重定位的帧率有没有达到要求, 以及加上重定位后跑动的精度。

#### (2) 测试结果

根据摄像头现有的安装高度计算出视野范围为 200mm, 底盘速度最快可达 4m/s, 要使摄像头获取到每个十字, 摄像头应具有帧率为 20fps, 处理时间应保证在 50ms 以内。而现有程序运行一遍时间为 20-30ms, 摄像头帧率可以保证在 30fps 以上, 完全满足速度要求。

经过多次跑动测试，不带重定位时，底盘位置有 30-50mm 的误差，加上重定位之后，x 方向误差可控制在 2-5mm，y 方向误差可控制在 5-10mm。跑动精度明显提高。

## 4 静态飞盘识别清台

机器人整体功能可分为攻守两个方面，一方面，要实现尽快地攻占所有着陆台，取得胜利；另一方面，当对方比己方速度快时，要阻止对方占领所有着陆台，而清台就是守住己方着陆台的策略。从实现的可能性和代价来看，守距己方最近的着陆台是最优的，因为这个台子离对方最远，对方落台难而已方清台容易，且从清台所需要的高度、飞盘轨迹直线度来看，都是清己方着陆台的可能性和准确率最高，所以，清台策略定为一旦有对方飞盘落到己方着陆台上，则上层发射机构立即进入清台模式，发射一个飞盘将对方飞盘清掉。清台时上层发射机构实行清台，而下层发射机构继续落台，这也是双发的优势所在。本章介绍视觉控制清台的具体实现，根据清台的三个线程分别讲述，在对方飞盘识别上还用到了三种方案比较。清台总框架如图（4-1）所示。



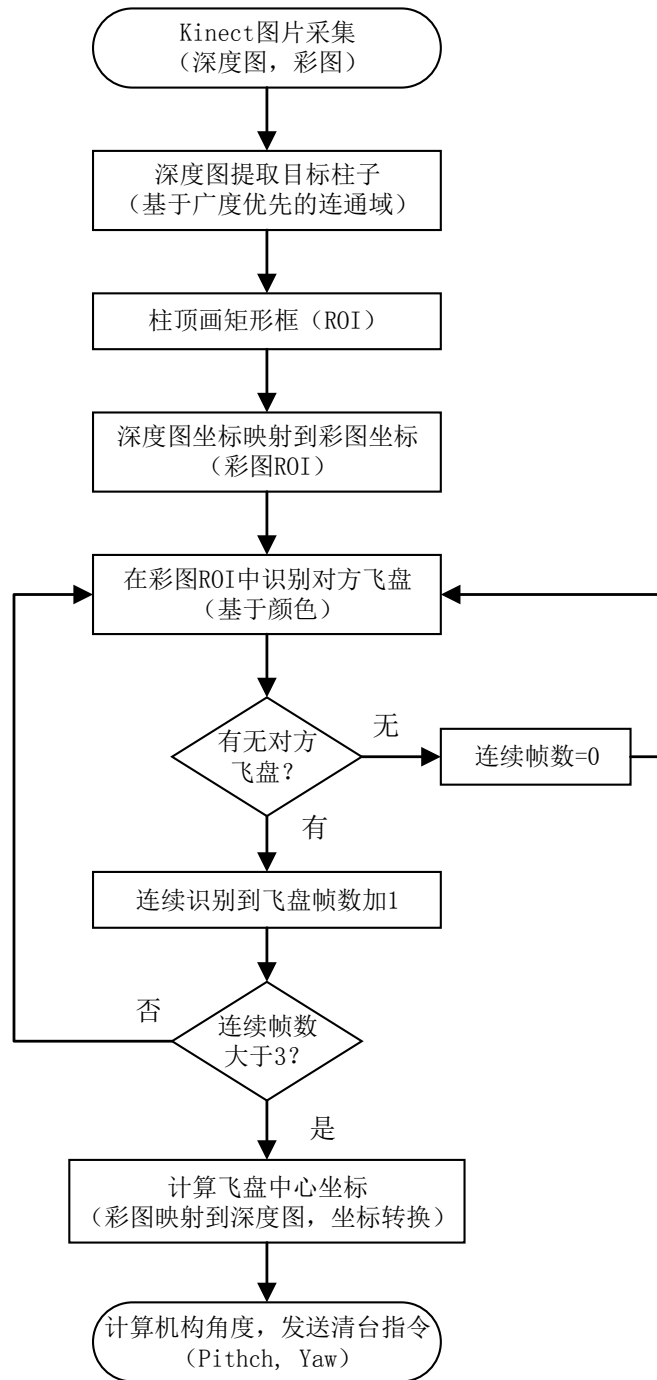


图 4-1 清台流程图

清台程序上分三个线程完成，分别为图像采集线程、目标柱子提取线程以及飞盘识别线程，采集线程用于获取 Kinect 图像，完成一些简单的图像预处理；然后在深度图下找出目标柱子以及己方着陆台，即划分出了找飞盘的区域——感兴趣区域（ROI, Region Of Interest）；再到彩图的相应区域基于颜色信息识别和判断飞盘，一旦发现对方飞盘则立即发出清台指令。

## 4.1 图像采集和预处理

### 4.1.1 相机选型

由于清台用到三维空间坐标，所采用的视觉设备必须具备获取三维坐标的能力。下面对课题研究过的几种设备进行介绍，最终完成相机选型。

#### 4.1.1.1 双目视觉

双目立体视觉（Bincular Stereo Vision）是机器视觉的重要形式之一。与人眼睛观测的原理相似，它是基于视差原理。融合两个摄像头获得的图像并观察它们之间的差别，由此建立特征间的对应关系，将同一空间物理点在不同图像中的映像点对应起来，这个差别称作视差。利用视差可以对图像进行一些数学运算，进而获取物体三维坐标信息。



图 4-2 (a) 双目视觉系统

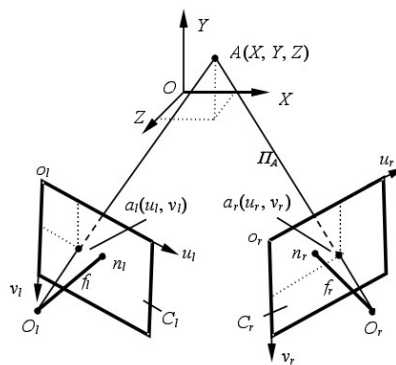


图 4-2 (b) 双目测距原理

双目立体视觉具有效率高、精度高、测距远等优点，在对运动物体的测量中，由于图像获取是在瞬间完成的，因此双目视觉方法是一种很有效的测量方法。然而，双目视觉在对高速物体进行观察时，必须要求两个相机同步触发，要使两个相机完全同步很难做到。且由于帧率较高，同时需要处理两幅图像，并进行匹配，双目视觉计算量是十分巨大的，一般来讲需要使用服务器进行处理。

#### 4.1.1.2 深度相机

深度相机是指可以直接返回物体深度值的相机，目前主要有两种技术——Light Coding 和 Time Of Flight（简称 TOF）。微软 Kinect 一代采用了 Light Coding 技术，而其二代产品 Kinect V2 则采用数据稳定性更好、精度更高的 TOF 技术。

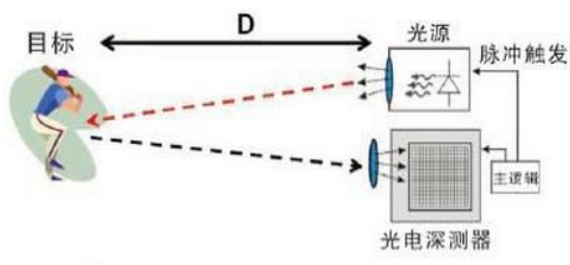


图 4-3 (a) TOF 技术



图 4-3 (b) Kinect V2

Time Of Flight，顾名思义，是通过给目标连续发送脉冲，然后用传感器接收从物体返回的光，通过探测光脉冲的飞行（往返）时间来得到目标的距离。原理如图 4-3 (a) 所示。

与双目立体视觉相比，TOF 相机的体积可以说是非常小巧，适合于一些需要轻便、小体积的场合。双目立体相机由于需要用到较为复杂的匹配算法，处理速度较慢。而 TOF 相机能够实时快速的计算深度信息。最为重要的是，TOF 相机深度的测量精度基本不随距离改变而变化，这对于一些大范围运动的应用场合非常有意义。微软的 Kinect V2（图 4-3 (b)），采用了最新的 TOF 技术，成本低廉，十分适用于本课题。

#### 4.1.2 图像采集与预处理

Kinect V2 有一个彩色镜头、红外发射和一个红外接收，可以获取 1920\*1080 的高清彩色图像和 512\*424 大小的深度图像。Kinect 图像采集利用的是专用的微软 SDK 开发包，专用 SDK 只针对特定设备，且采集图片质量更高。

获取的彩图是 BGRA 四通道，图片内存大，且对彩图的处理不需要 A 通道透明度的信息，因此对彩图进行了 BGRA 四通道到 BGR 三通道的转化。采集到的深度图是 16 位单通道，处理时用的是像素值 0-255 的 8 位单通道图像，所以对深度图进行了 16 位到 8 位的转化。

## 4.2 基于广度优先的连通域柱子提取

在对飞盘进行识别之前，需要先划分出找飞盘的区域（只在台面范围内进行识别），所以开了一个柱子线程，完成对目标柱子的提取以及圆台区域的确定。柱子在空间中是一个独立的连通的区域，可以采用对深度图进行提连通域的方法将其提取出来。图像中像素点的邻接关系有两种，四邻接和八邻接，课题采用四邻接的连通域。

提连通域需要种子点来启动，种子点即为目标连通域中的一个点，以此为初始点向周边逐一遍历，遍历一遍即可提取出一个完整的连通域。课题需要提取柱子连通域，因此有效种子点应为图幅中柱子上的点。而连通域的遍历方式有深度优先搜索和广度优先搜索两种基本方式，深度优先搜索用栈实现，即把每次搜索满足条件的点存入栈中，先进后出，表现出来就是尽可能先对纵深方向进行搜索；广度优先搜索使用队列来实现，即把每次搜索满足条件的点存入队列中，先进先出，表现出来就是尽可能先对广度方向进行搜索。

本课题采用了种子点区域预判的方式选取可疑种子点，采用广度优先搜索的方式对可疑种子点的四个邻接方向进行遍历，通过一系列的判断条件，最终提取出目标柱子连通域。

### 4.2.1 种子点的选取

场地条件固定，场地中目标柱子的位置也是固定的，可以得到柱子的大致坐标，从而划分可疑种子点区域。建立的世界坐标系和机器人坐标系如图 4-4 所示。



图 4-4 世界坐标系与机器人坐标系

已知目标柱子的世界坐标 ( $pillar_x, pillar_y$ )，从底盘定位板可以得到机器人的当前位姿信息——车中心坐标 ( $chassis_x, chassis_y$ ) 以及姿态角  $\alpha$ ，则通过坐标转换可以得到目标柱子在机器人坐标系下的位置，计算公式如式 (4-1)。

$$\begin{cases} aimPillar_x = (pillar_y - chassis_y) * \cos(\alpha) - (chassis_x - pillar_x) * \sin(\alpha); \\ aimPillar_y = (chassis_x - pillar_x) * \cos(\alpha) + (pillar_y - chassis_y) * \sin(\alpha) \end{cases} \quad (4-1)$$

通过底盘位姿和目标柱子世界坐标计算出的坐标 ( $aimPillar_x, aimPillar_y$ )，是预判的柱子位置。由于数据传输、程序运算等的延时和计算的误差，预判的坐标精度不高，要保证选取的种子点一定包括有效种子点（在柱子上的点），需要画一个狭长的矩形区域作为可疑种子点候选区。划取的效果如图 (4-4) 所示。

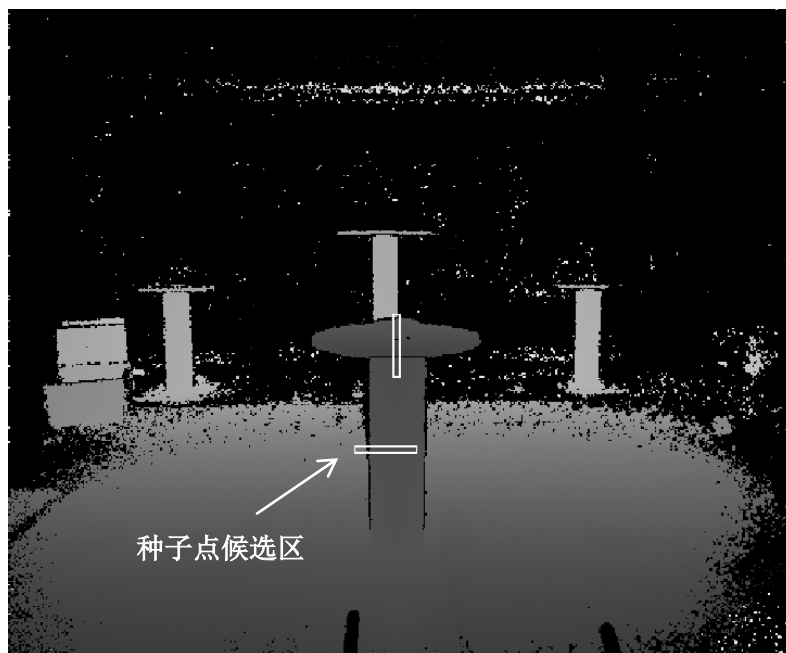


图 4-4 种子点的选取

深度图的像素点坐标代表像素点的位置，而像素值代表深度距离，单位为毫米。取预判的柱子高度  $1/2$  处的点  $(aimPillar_x, aimPillar_y, aimPillar_z/2)$  进行坐标转换，得到图像中的一点  $(image\_x, image\_y)$ ，即可在图像上画矩形框。

#### 4.2.2 广度优先的遍历

连通域的遍历方式有深度优先搜索和广度优先搜索两种。深度优先搜索用栈来实现，即把每次搜索中满足条件的点存入栈中，完成了当前点的四邻接搜索则从栈顶弹出一个新点进行搜索，栈中的点先进后出，表现出来就是尽可能先对纵深方向进行搜索；广度优先搜索使用队列来实现，即把每次搜索满足条件的点存入队列中，完成搜索的点退出队列，从队尾弹出一个新点进行新一轮搜索，队列中的点先进先出，表现出来就是尽可能先对广度方向进行搜索。课题采用广度优先搜索的方式实现连通域遍历，图（4-5）中的数字显示了广度优先搜索时顶点被访问的顺序。

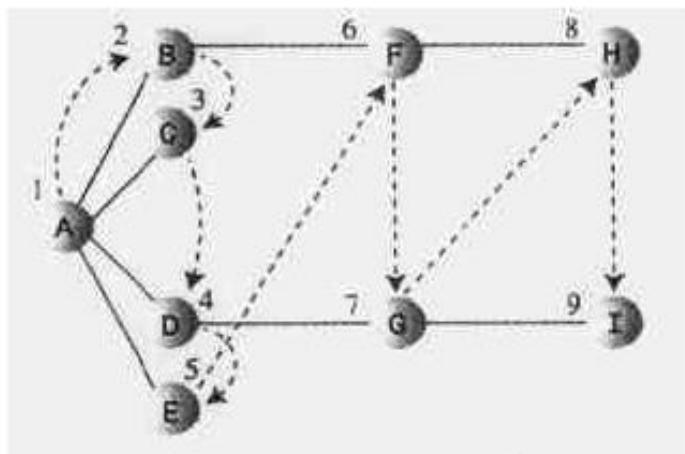


图 4-5 广度优先搜索

对每个可疑种子点进行广度优先搜索，判断左右上下四领域的点是否属于同一连通域，每个被遍历过的点标志位置 1，保证不重复搜索。把满足连通条件的点做上标记，就完成了连通域的标记工作，对标记出的连通域进行面积、高度等限制，就得到目标柱子连通域。只要种子点候选区中有一个柱子上的点，就能一次遍提出整根柱子，提取出的柱子如图（4-6）所示。

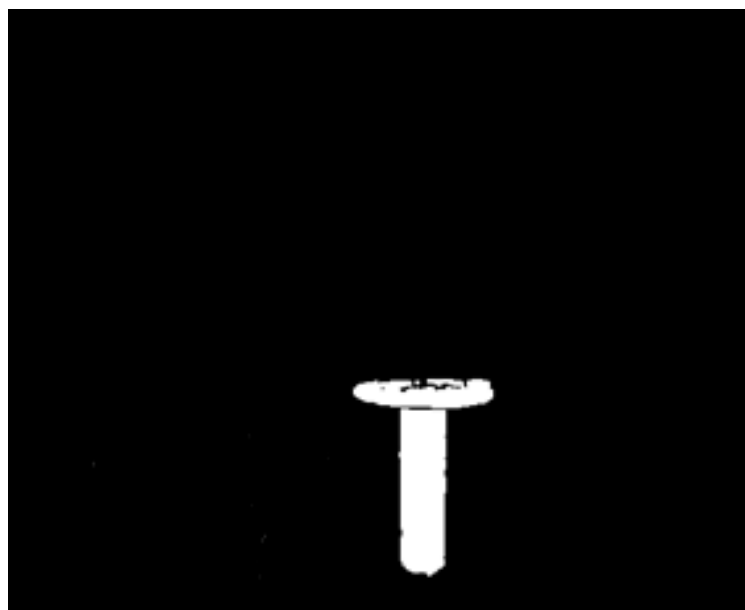


图 4-6 连通域遍历提柱子

由于场地较空旷，柱子在空间中为一个独立的连通域，对满足深度条件、面积和高度条件的连通域进行遍历后，可以得到图幅中一个完整的柱子区域，从而得到柱顶坐标（ $\text{pillar\_peak\_x}, \text{pillar\_peak\_y}$ ）。

### 4.3 基于区域划分和颜色的飞盘识别

对柱子的提取得到了己方着陆台所在的位置,实现清台要先识别出对方飞盘,再计算飞盘坐标以及机构角度。静态飞盘的提取主要用到区域划分和颜色识别的方法,即先把台面范围划分出来,滤掉背景、周围环境的干扰,再进行颜色的识别,比赛中只有红蓝两色飞盘,红场用红飞盘,蓝场用蓝飞盘,为了方便论述,拟定己方在蓝场的情况,进行红飞盘的清台。

#### 4.3.1 台面范围的划分

提连通域是在深度图下进行的操作,在对飞盘进行识别时要转换到彩图下处理。台面范围的划分就是把深度图下的目标柱子位置映射到彩图下,再在柱顶根据着陆台的大小划分出感兴趣区域(ROI, Region Of Interest)。

在图片采集线程,得到的是 512\*424 的深度图和 1920\*1080 的彩图,两者像素明显不对齐。从深度图一点映射到彩图一点,课题采用了查表得方式,查表的前提是得到一张把深度图像素点和彩图像素点一一对应起来的表,这是调用微软 SDK 中的库函数 MapDepthFrameToColorSpace 进行了一遍全图幅遍历生成的。用两张二维数组分别储存 x 坐标和 y 坐标,从彩图一点映射到深度图一点的公式如式(4-2)。

$$\begin{cases} x_{color} = \text{mapdepth2color\_x\_data}[y_{depth}][x_{depth}]; \\ y_{color} = \text{mapdepth2color\_y\_data}[y_{depth}][x_{depth}]; \end{cases} \quad (4-2)$$

式中,  $(x_{color}, y_{color})$  为彩图像素坐标,  $(x_{depth}, y_{depth})$  为深度图像素坐标, mapdepth2color\_x\_data 和 mapdepth2color\_y\_data 分别为 x 坐标映射表和 y 坐标映射表。

将深度坐标 (pillar\_peak\_x, pillar\_peak\_y) 映射到彩图下之后,根据台面大小可以划分出一个范围,即为找飞盘的 ROI。划分的效果如图(4-7)所示。





图 4-7 台面范围

在彩图中划出感兴趣区域进行处理，一方面可以滤除周围环境的干扰，使识别更准确、更稳定；另一方面减少了需要处理的图幅，算法更简单，处理时间更短，提高了程序运行效率。

### 4.3.2 飞盘的识别

比赛提供的飞盘如图（4-8）所示。

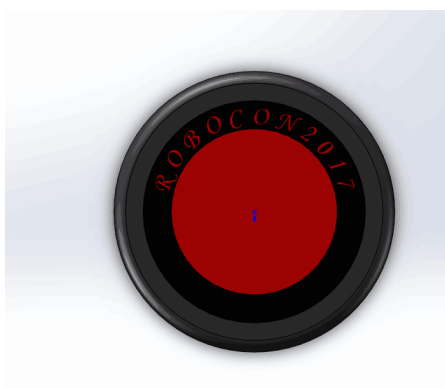


图 4-8 比赛红飞盘

在彩图 ROI 中找飞盘主要用到飞盘的颜色信息，提取出飞盘后计算其形心位置的方法有三种，课题分别进行了比较并确定了使用的方案。

#### 4.3.2.1 基于颜色的飞盘提取

台面上的飞盘非红即蓝，对红色和蓝色部分的提取分两步进行：

- (1) 二值化，满足红蓝阈值条件的像素点高亮显示，阈值条件如式（4-4）。

$$\text{Th\_low} < \text{gray} < \text{Th\_high} \quad (4-4)$$

经过这一步，ROI 中的红蓝两色被提取出来，白色高亮显示的部分非红即蓝。

(2) 分通道, 将彩图分为 B,G,R 三个单通道图, 比较像素点的各通道值, R 通道值最大的为红色, B 通道值最大的为蓝色。判断条件如式 (4-5)。

$$\begin{aligned} \text{红色: } R > B \ \&\& \ R > G \\ \text{蓝色: } B > R \ \&\& \ B > G \end{aligned} \quad (4-5)$$

经过颜色的提取, 可以识别出台面上的红色部分和蓝色部分, 当面积满足飞盘大小阈值时, 认为提取到红色或者蓝色飞盘。提取的效果图如图 (4-9) 所示。



(a) 彩图

(b) B 通道

(c) R 通道

图 4-9 提取红蓝飞盘效果图

从图中可以看到, 台面上共有两蓝一红三个飞盘, 通过 B 通道提取出了两个蓝色飞盘, 通过 R 通道提取出了一个红色飞盘。

#### 4.3.2.2 飞盘形心计算

课题尝试了三种计算飞盘形心的方法, 分别为最小包围矩形法、轮廓中心矩法和连通域中心法。它们的比较如表 (4-1) 所示。

表 4-1 飞盘形心计算方法比较

方案	最小包围矩形法	轮廓中心矩法	连通域中心法
原理	对飞盘进行提轮廓(为一组边界点集), 根据最外层轮廓求最小包围矩形, 然后用矩形的中心代表飞盘的中心。	对飞盘进行提轮廓(为一组边界点集), 计算轮廓中心矩, 求得轮廓的中心即为飞盘的中心。	用基于广度优先的连通域算法遍历出整个飞盘, 返回连通域中心坐标即为飞盘中心
效果图	 <p>(图中蓝色为最小包围矩形, 黄点为提取的中心)</p>	 <p>(图中蓝色为提取的轮廓, 红点为计算的中心)</p>	 <p>(图中白色为遍历出的飞盘, 黑点为飞盘中心)</p>

优缺点	最小包围矩形的中心容易计算，但用作飞盘的中心不太准确，此法耗时 0.2ms	用轮廓中心计算飞盘形心较为准确，但计算耗时，运行一遍需 1ms	连通域算法较复杂，但用数组实现也很快，耗时与连通域的面积大小有关，遍历一个飞盘需 0.4-0.8ms
-----	---------------------------------------	---------------------------------	--

由于清台不需要非常准确的飞盘中心，只要发射出去的飞盘能够击中对方飞盘即可将其清掉，而落台成功有 3 秒的时限要求，所以清台要快。综合考虑，选用了时间复杂度最低的最小包围矩形法来计算飞盘的中心，得到像素坐标 (disk\_x, disk\_y) 以及中心处的深度值 disk\_depth。

### 4.3.3 角度计算清台

指导上层发射机构清台要发给机构执行参数，发射机构进行发射时，有四个调节的参数——主电机转速，辅电机转速，俯仰角 Pitch 以及餐盘转角 Yaw。

清台时，要求发射出去的飞盘速度较快，以保证足够的直线度来命中目标，所以主辅电机给了固定的转速 6500r/min 和 500r/min，差速 6000r/min 使飞盘出射速度可以满足要求。

清台需要的俯仰角 Pitch 和餐盘转角 Yaw 是通过飞盘所在位置计算出来的，分两步进行，飞盘中心坐标转换和角度计算。

(1) 飞盘中心坐标转换，将求得的飞盘中心的像素坐标转换到机器人坐标系下，得到一个实际的三维物理坐标，计算公式如式 (4-5)。

$$\begin{cases} center\_xg = disk\_depth * TANX / (width/2) * (width/2 - x); \\ center\_zg = disk\_depth * TANY / (height/2) * (\frac{height}{2} - y); \\ center\_yg = disk\_depth; \end{cases} \quad (4-5)$$

式中，(center\_xg, center\_yg, center\_zg) 为机器人坐标系下的飞盘中心坐标，width 和 height 分别为图像宽度和高度方向上的分辨率，TANX 和 TANY 为 Kinect 的水平半角和竖直半角的正切值。

(2) 角度计算，根据飞盘中心的三维空间坐标计算出机构实现清台需要的 Pitch 角和 Yaw 角。计算公式如式 (4-6)。

$$\begin{cases} \text{Pitch} = \text{atan2}((\text{LAUNCH\_Z} - \text{center\_zg}) / \text{center\_yg}); \\ \text{Yaw} = \text{atan2}(\text{center\_xg} / \text{center\_yg}); \end{cases} \quad (4-6)$$

式中, LAUNCH\_Z 为发射机构距地面的高度。

通过角度计算得到机构的另外两个参数 Pitch 和 Yaw, 就能向主控发送清台指令, 指导上层机构实现清台。

## 4.4 清台模块测试

### (1) 测试方案

在己方着陆台上的前后左右各个方位放上对方飞盘, 测试发送的清台指令能不能把飞盘清掉, 以及清掉飞盘所需要的时间。

### (2) 测试结果

在着陆台上各个方位的飞盘都能被识别出来, 并且可以被己方发射机构清理掉, 但有时发射一个飞盘就能清掉对方飞盘, 有时要两发才能清掉。

赛规规定, 每个着陆台上有超过 3 秒未掉落的飞盘为落台成功, 所以有效的清台应在对方飞盘落上后的 3 秒内。经过反复测试, 发射机构发射一个飞盘需要 1 秒的时间, 用一发或两发飞盘清台均能保证在 3 秒的有效时间内。故清台可行。

## 5 动态飞盘轨迹跟踪

机器人发射出飞盘后飞盘就不再可控,要使发射出的飞盘准确落到着陆台上,需要对飞盘轨迹特点进行分析,找到落点关于出射速度、出射角度等的关系,进而指导发射机构需要的参数。对运动的飞盘进行跟踪,需要机器视觉来实现。同样,飞盘轨迹信息要用到三维空间坐标,需要一款能获得深度数据的摄像头,Kinect 相机选型已在 4.1.1 中做了详细论述,这里还是使用 Kinect 相机。本章将详细介绍飞盘轨迹的识别方法,图像采集和预处理和清台使用的同一线程,因此直接介绍处理线程的实现,即帧差法提取运动飞盘,多项式拟合飞盘轨迹。轨迹跟踪的框架图如图(5-1)所示。

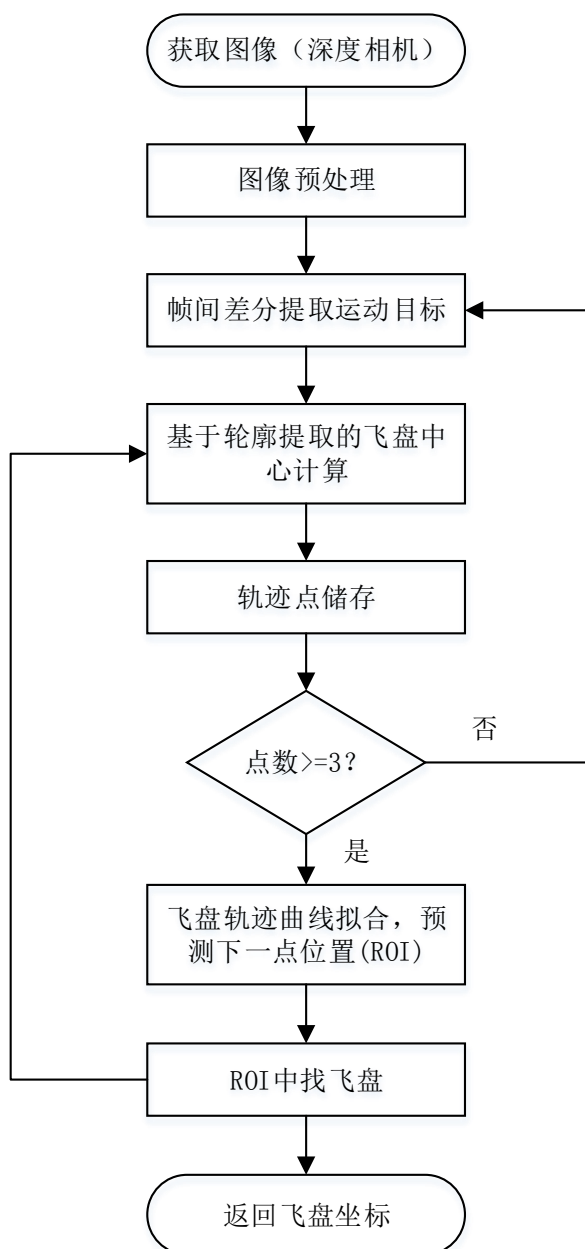


图 5-1 飞盘轨迹跟踪流程图

通过深度相机获取有深度信息的图像，经过预处理后，应用帧间差分的算法提取运动的飞盘，用基于轮廓提取的方法获取飞盘位置、面积、周长等信息，将飞盘的轨迹点存为点集，点数大于 3 时即可进行拟合预测，利用多项式拟合算法预测下一点位置，得到感兴趣区域(ROI)，下次就可以在 ROI 中找飞盘，直到一条轨迹的结束。

## 5.1 基于三帧帧差的运动目标提取

在进行飞盘轨迹拟合前，要先识别出运动的飞盘，对运动飞盘的检测方法有很多，下面就课题尝试过的三种方法进行比较，并详细论述最后所采用的方法。

### 5.1.1 运动目标提取的方案比较

#### (1) 基于颜色和轮廓的飞盘识别

比赛中使用的飞盘是一个固定道具，具有确定的颜色、形状和大小，飞盘的特征还是比较明显的。图像处理时，先做阈值划分，提取出红蓝色的区域，再对提取的部分进行轮廓分析，满足面积、周长、宽高比、圆形度等限制条件的认为是飞盘。这种方法可以识别不论静态还是动态的飞盘，但是它存在的问题是：一方面，颜色信息对光照敏感，做颜色划分时的阈值不容易确定；另一方面，图像中背景元素较多，很容易误识别，满足颜色和轮廓要求的往往不只有飞盘。

#### (2) 基于模板匹配的飞盘识别

模板匹配是指用一事先准备好的模板，在全图幅中滑动，寻找与模板最相似的部分。用于飞盘识别时，先取飞盘中的一部分作为模板，模板要尽可能具有适应性；进行模板匹配时，调用 OpenCV 的库函数 `matchTemplate()`，返回的是图像中最可能是飞盘的部分。此方法既可用于静态飞盘识别也可用于动态飞盘识别，且直接调用库函数实现，简单、速度快。但它存在的问题是：很难找到在任何条件下都适用的模板，而且取的飞盘上一部分会存在遮挡问题，一旦丢失目标，就会返回一个错误结果。

#### (3) 基于帧间差分的动态飞盘识别

帧间差分是最常用的运动目标检测的方法之一，对于一个连续运动的物体，取图像的相邻两帧作差，就能得到图像中运动的像素。由于相邻两帧做差后，两帧图像上的运动物体都会被提取，可以对相邻三帧作差以避免这个问题。帧间差分运用了飞盘的运动特征，很容易滤除背景，但实现的前提是相机不能抖动，要保证背景的绝对静止。这也是最终轨迹识别部分没有加到机器人上的原因之一。

在进行轨迹实验时，摄像头是固定在三脚架上的，可以保证图像的背景静止，综合上述，帧间差分的方法最适合于使用的场景。

### 5.1.2 三帧帧差法的原理及算法实现

三帧帧差法的实现过程如图(5-2)所示。

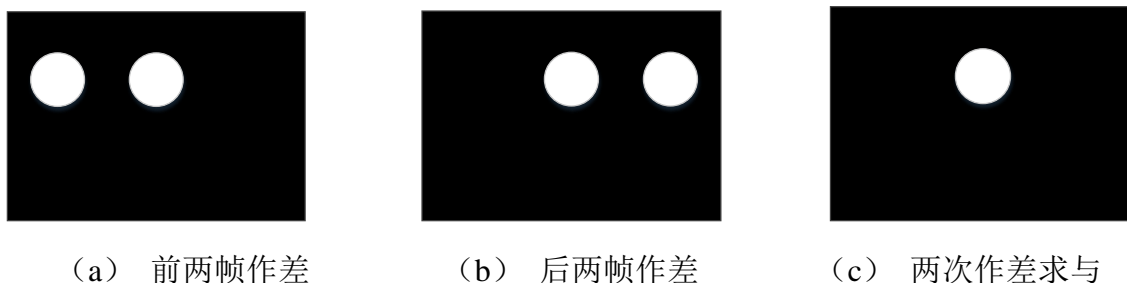


图 5-2 三帧帧差法

过程为：图片 1 与图片 2 进行差分，得到中间图片 (a)，图片 2 与图片 3 进行差分，得到中间图片 (b)，最后使用图片 (a) 与图片 (b) 进行和运算，求出交集图片图 (c)。得到的结果就是在采集第二幅图片时所得到的运动物体，或者说是运动物体运动部分与背景有差异的地方。经过帧差法处理后，得到的和运算图片再经过后续的图片处理就能够分离出飞盘。

### 5.1.3 基于轮廓提取的飞盘形心计算

将飞盘与背景分离后，要进行飞盘形心的计算，与清台不同的是，进行轨迹拟合需要准确的形心坐标，经过 4.3.2.2 的形心计算方案比较，选择轮廓的矩计算方法比较合适。形心计算分两步进行，第一步提轮廓，选择面积、周长符合条件的轮廓；第二布，计算轮廓中心矩，得到飞盘形心坐标。

(1) 提轮廓，利用 OpenCV 的库函数 findContours() 可实现轮廓的提取，计算轮廓面积周长分别使用 contourArea() 和 arcLength() 来实现，限制条件如式 (5-1) 所示。

$$\begin{cases} \minArea < Area < \maxArea; \\ \minLen < Len < \maxLen; \end{cases} \quad (5-1)$$

式中，minArea 和 maxArea 分别为飞盘面积的下限和上限，minLen 和 maxLen 分别为飞盘周长的下限和上限，据此选择符合条件的轮廓，计算其形心。

(2) 轮廓中心矩的计算。

对于灰度分布为  $f(x, y)$  的离散化图像，其  $(i+j)$  阶矩定义为：



$$m_{ji} = \sum_{x,y} (f(x,y) * x^j * y^i) \quad (5-3)$$

可以知道，对于 01 二值化的图像，m00 即为轮廓的面积。中心矩的公式为：

$$m_{ji} = \sum_{x,y} (f(x,y) * (x - \bar{x})^j * (y - \bar{y})^i) \quad (5-4)$$

其中， $(\bar{x}, \bar{y})$  为矩的中心。所以，轮廓中心的计算公式为：

$$\bar{x} = m10/m00; \quad (5-5)$$

$$\bar{y} = m01/m00;$$

在 OpenCV 中，轮廓中心矩的计算都可以调用库函数来实现。如式（5-6）所示。

$$\begin{aligned} \text{mu}[i] &= \text{moments}(\text{contours}[i], \text{false}); \\ \text{mc}[i] &= \text{Point2f}((\text{mu}[i].m10/\text{mu}[i].m00), (\text{mu}[i].m01/\text{mu}[i].m00)); \end{aligned} \quad (5-6)$$

式中，i 表示第 i 条轮廓，mu 为计算出的轮廓中心矩，而矩的中心  $(\bar{x}, \bar{y})$  即为 mc[i].

## 5.2 基于多项式拟合的轨迹跟踪

在目标跟踪上，根据算法的具体实现，跟踪算法可分为基于运动分析的跟踪、基于目标表示的跟踪、基于变形模板的跟踪以及基于分类的跟踪等。由于飞盘的运动轨迹特征较明显且固定，课题适用基于运动分析的跟踪，主要用到曲线拟合技术，对飞盘的运动轨迹进行多项式拟合，预测下一个飞盘的位置。

### 5.2.1 多项式拟合轨迹方程

飞盘轨迹是一条特征明显的曲线，可用多项式进行拟合，拟合时将飞盘三维坐标分别表示为时间  $t$  的参数，求解其参数方程。

多项式拟合分三种情况进行：找到点数  $n$  为 1 时，进行一次拟合；找到轨迹点数  $n$  为 2 时，进行二次拟合；找到轨迹点数  $n$  大于等于 3 时，均进行三次拟合。

轨迹方程的表示如式 (5-1) ~ (5-3) 所示。

$$n=1: \quad \begin{cases} x = a_1 * t + b_1; \\ y = a_2 * t + b_2; \\ z = a_3 * t + b_3; \end{cases} \quad (5-1)$$

$$n=2: \quad \begin{cases} x = a_1 * t^2 + b_1 * t + c_1; \\ y = a_2 * t^2 + b_2 * t + c_2; \\ z = a_3 * t^2 + b_3 * t + c_3; \end{cases} \quad (5-2)$$

$$n \geq 3: \quad \begin{cases} x = a_1 * t^3 + b_1 * t^2 + c_1 * t + d_1; \\ y = a_2 * t^3 + b_2 * t^2 + c_2 * t + d_2; \\ z = a_3 * t^3 + b_3 * t^2 + c_3 * t + d_3; \end{cases} \quad (5-3)$$

轨迹拟合就是根据已有轨迹点的三维坐标，求出各自的系数，从而得到  $x, y, z$  分别关于时间  $t$  的参数方程。

常见的曲线拟合方法有三种：使偏差绝对值之和最小，是偏差绝对值最大的最小，使偏差平方和最小。课题采用使偏差平方和最小的最小二乘法拟合。最小二乘法拟合曲线可用矩阵来实现，表示成矩阵运算关系如式 (5-4) 所示。

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \cdots & \sum_{i=1}^n x_i^k \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \cdots & \sum_{i=1}^n x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_i^k & \sum_{i=1}^n x_i^{k+1} & \cdots & \sum_{i=1}^n x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \vdots \\ \sum_{i=1}^n x_i^k y_i \end{bmatrix} \quad (5-4)$$

也就是说  $X * A = Y$ ，那么  $A = (X' * X)^{-1} * X' * Y$ ，便得到了系数矩阵，同时也得到了曲线方程。以上计算过程均可用代码实现。

轨迹拟合的效果图如图 (5-3) 所示。

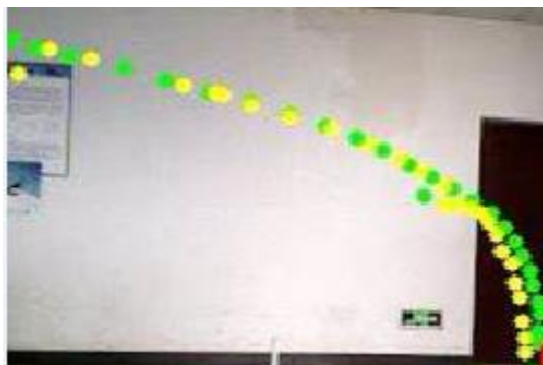


图 5-3 轨迹拟合

图中绿点为实际轨迹点，黄点为拟合轨迹点。可以看出拟合结果与实际较接近。

### 5.2.2 轨迹点分类及多目标追踪的实现

求出轨迹方程后，给出下一个点的时间步长 $\Delta t$ ，即可预测下一个轨迹点的位置，在预测点周围画一个矩形框作为下一次搜索飞盘的感兴趣区域 ROI，可以对不同轨迹点进行分类。划分 ROI 的效果图如图（5-4）所示。

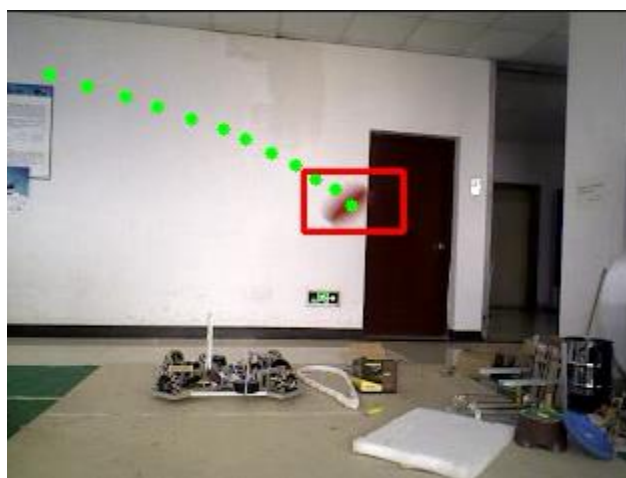


图 5-4 预测下一点 ROI

划分出不同轨迹点所属的曲线后，就可以进行多目标的跟踪，每新出现一条轨迹则新增一个数组，不同轨迹点储存在不同的数组中，进行多条轨迹曲线的拟合。多目标跟踪的结果如图（5-5）所示。

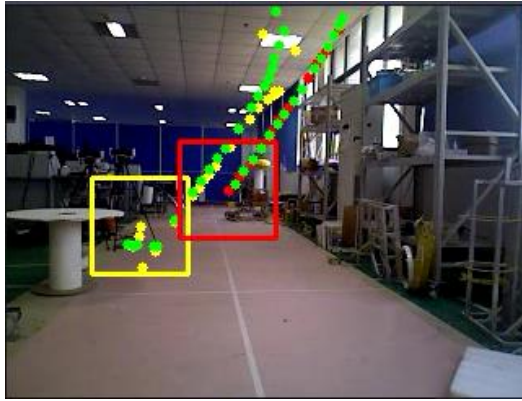


图 5-5 多轨迹拟合效果图

图中绿点为实际轨迹点，有两条；黄点和红点分别对应各自的拟合点，末端的矩形框分别对应预测的下一个 ROI。

### 5.3 飞盘轨迹模块测试

#### (1) 测试方案

一方面，发射飞盘，进行轨迹拟合，看拟合的结果与实际是否相符；另一方面，保持空中有多个飞盘，看飞盘跟踪最多能跟踪几个飞盘。

#### (2) 测试结果

从图（5-3）两条轨迹曲线可以看出，拟合结果基本与实际相符。

由于同时保证空中有多个飞盘比较困难，实验只测试了最多有 3 个飞盘的情况，3 条飞盘轨迹都能识别并区分开来，但是没有 2 条轨迹跟踪的稳定。

## 6 总结与展望

经过近十个月的努力，我们的机器人终于能够完成所有比赛任务，最快落台参数为 39 秒，且能自动清台不让对方先取得胜利。课题所研究的机器人定位和飞盘识别的视觉系统，针对比赛给出了可行方案，并在多次实验中能够满足设计需求。

在本届赛题对于机器人环境信息采集的能力提出了更高要求的大背景下，不仅在原有的视觉技术上加以革新使之更具有适应性，还对技术做了进一步拓展，引入了深度视觉技术。经过不断的测试与改进中，完成了利用 USB 摄像头进行的重定位，具有良好的重复性与抗干扰能力；完成了利用深度相机进行的飞盘静态和动态识别，具有良好的精度与抗干扰能力，取得了不错的检测效果。

优化是没有止境的，我们的工作也存在着许多不足之处，有如下几个方面亟待我们去改进：

- 1.重定位始终不可能与底盘跑动保持同步，图像处理存在一个延时；
  - 2.清台应该具有更高的效率，用己方两个飞盘清对方一个飞盘是对己方弹药的浪费；
  - 3.大量运动物体的跟踪还不能做到稳定；
- 这些问题仍需要进一步的研究来完善。

## 致 谢

首先要感谢九个月来与我一起奋斗拼搏的队友们，他们分别是：李新新、张栋豪、贾重洋、帅治民、娄文谦、杨标强，没有他们，就无法完成这个比赛，也就没有现在的这些成果。

感谢那些一直在背后默默支持我们的师兄们：刘良、徐意、李嘉伦、胡诗尧、但威、秦国华、谭欢、田开旺等，是他们在我们迷惘时，拨云见日，助我们走得更稳更远。

感谢我的毕设指导老师宋宝教授以及唐小琦教授，没有他们的帮助这个课题就无法完成。

感谢我们的团队指导老师：丁雨葵、王建庄等，他们在百忙中时刻牵挂着我们，更是为我们提供了强大的技术后盾和后勤保证。

同时还要感谢那些我们可敬的竞争对手们，他们是电子科技大学、西安交通大学、太原工业大学、哈尔滨工业大学、武汉大学、东北大学、河北工程大学等。没有压力就没有动力，是他们的强大时时督促着我们前进，和他们的交流使我们获得了技术上的提升。

大学四年一晃而过，ABU Robocon 的经历或许是这四年以来最难忘的，忘不了和队友一起彻夜奋斗的经历，忘不了我们的机器人终于通车时的兴奋。人生的道路有无数选择，而这个路口我的选择是跟你们一起做 ABU Robocon。

## 参考文献

- [1] I. Culjak,D.Abram,T.Pribanic,H.Dzapo and M.Cifrek.A brief introduction to OpenCV.MIPRO 2010:1725-1730
- [2] Vladimir Katkovnik,Alessandro Foi,Karen Egiazarian,Jaakko Astola.From Local Kernel to Nonlocal Multiple-Model Image Denoising[J].Int.J.Comput,2010,86:1-32
- [3] M.Pakdaman,M.Sanaatiyan and M.R.Ghahroudi.A line follower robot from design to implementation:Technical issues and problems.ICCAE,2010,The 1st:5-9
- [4] Jiraphan Inthiam and Chirdpong Deelertpaiboon.Self-Localization and Navigation of Holonomic Mobile Robot using Omni-Directional Wheel Odometry[J].TENCON 2014-2014 IEEE Region 10 Conference,2014:1-5
- [5] Nirmal Baran Hui,Dilip Kumar Pratihari.Camera calibration using a genetic Algorithm.Engineering Optimization,2008,40(12):1151-1169
- [6] Nicholas Fung and Philip David.Implementation of efficient Pan-Tilt-Zoom camera calibration.ARL-TR-4799,Tech report,2009
- [7] Andreas KriiuBling.Tracking Multiple Objects Using the Viterbi Algorithm.Lecture Notes Electrical Engineering,2008,15:233-247
- [8] Tao Zhao,Ram Nevatia,Bo Wu.Segmentation and tracking of multiple humans in crowded environments.IEEE Transactions on Pattern Analysis and Machine Intelligence,2008,30(7) :1198-1211
- [9] W.J.Crowther and J.R.Potts.Simulation of a spin-stabilised sports disc.isea Sports Engineering ,2007,10: 3-21
- [10] Yin J, Liu L, Li H, Liu Q. The infrared moving object detection and security detection related algorithms based on W4 and frame difference. INFRARED PHYSICS & TECHNOLOGY,2016,77:302-15
- [11] 张五一,赵强松,王东云等.机器视觉的现状与发展趋势[J].中原工学院报,2008,19(1):9-12

- [12] 李树杰.中国机器视觉的发展趋势[J].赤峰学院学报(自然科学版),2010,26(1):161-162
- [13] 杨家军.机械原理[M].武汉:华中科技大学出版社,2009
- [14] 吴昌林,张卫国.机械设计[M].武汉:华中科技大学出版社,2011
- [15] 熊有伦.机器人技术基础[M].武汉:华中理工大学出版社,1996
- [16] Gary Bradski and Adrian Kaehler.学习 opencv(中文版)[M].北京:清华大学出版社,2009
- [17] 徐德,谭民,李原.机器人视觉测量与控制[M].北京:国防工业出版社,2011
- [18] Rafael C,Gonzalez,Richard E,Woods.数字图像处理(第三版)[M].北京:电子工业出版社,2011
- [19] M. Sonka,V.Hlavac,R.Boyle.图像处理、机器视觉分析(第3版)[M].北京:清华大学出版社,2011
- [20] 杨杰,张翔.视频目标检测和跟踪及其应用[M].上海:交通大学出版社,2012
- [21] 郭斌.图像去噪处理技术[D].西安:西安电子科技大学,2012
- [22] 范容谦.基于机器视觉的位姿测量关键技术研究[D].重庆大学,2014
- [23] 刘伟,赵剑波,高峰等.基于相对坐标的机器人末端位姿测量方法[J].机器人,2009,(1):15-19
- [24] 李谷全,陈忠泽.视觉跟踪技术研究现状及其展望[J].计算机应用研究,2010,27(8):2814-2821
- [25] 张娟,毛晓波,陈铁军.运动目标跟踪算法研究综述[J].计算机应用研究,2009(12):4407-4410
- [26] 邓红德,王丽君,金波,Deng Hongde,Wang Liun,Jin Bo.一种无人机自主着陆视觉跟踪方法[J].计算机测量与控制,2009
- [27] 刘钟远.复杂环境下运动目标检测和跟踪技术研究[D].电子科技大学,2016.
- [28] 刘涛.基于光流场的视频车辆检测与跟踪算法研究与应用[D].武汉科技大学,2011
- [29] 李惠松,王小铭,张玉霞.一种基于帧间差分与时空相关性分析的运动目标检测算法[J].计算机与数字工程,2007,35(12):32-34