

# 南京工业大学

—2020 届毕业设计（论文）

题 目：基于深度视觉的室内

移动机器人跟随系统设计

专 业：测控技术与仪器

班 级：测控 1601

姓 名：李云锋

指导老师：冯李航

起讫日期：2020.02-2020.06

2020 年 6 月



# 基于深度视觉的室内移动机器人跟随系统设计

## 摘 要

由于人工智能的广泛应用，社会生活发生着日新月异的变化。根据不同的应用场景诞生了多种多样的机器人，其中尤以目标识别，路径动态规划作为移动机器人的核心技术。但由于实际应用场景极其复杂，各项技术仍有很大的研究空间。本文将针对室内移动机器人的目标跟随问题作如下研究：

(1) 基于 kinect2 获取移动机器人所需要的图像和深度信息。通过比较目前比较流行的深度图像获取方法，采用比较成熟的微软 kinect2 传感器同时获得深度信息和彩色图像。针对在某些情况下，深度相机可能无法提供某些像素的正确值，采用绘制轮廓的方法降低对深度信息获取的影响。

(2) 根据室内跟随机器人的应用场景，采用深度学习中的 YOLOv3 方法对目标进行识别。YOLOv3 有着高精度和实时性强的特点，给了跟随机器人更多的可能性。

(3) 基于人工势场法实现机器人移动的控制策略。为了提高鲁棒性，实验并测试对应参数的阈值，提高移动机器人在动态场景的自主移动能力。

(4) 以 TurtleBot 为实验平台，通过测试场景，评估移动机器人的跟随效果。

**关键词：**跟随机器人 深度视觉 目标识别

# **Design of indoor mobile robot following system based on depth vision**

## **Abstract**

The rapid development of artificial intelligence has made relevant technologies gradually permeate all aspects of our lives. On the basis of different application scenarios, multiple robots have been developed. Among them, target recognition and path dynamic planning are the core technologies of mobile robots. However, due to the extremely complicated practical application scenarios, we still have to do some research on those technologies. This article will do the following research on the target following problem of indoor mobile robots:

(1) Using kinect2 to acquire the image and depth information which we need. By comparing the current popular depth image acquisition methods, the more mature Microsoft kinect2 sensor is used to obtain both depth information and color images. In some cases, the depth camera may not be able to provide the correct values of some pixels, and the threshold segmentation method is used to reduce the impact on the depth information acquisition.

(2) According to the application scenarios of indoor following robots, the YOLOv3 method in deep learning is used to identify the target. The high real-time and high precision of yolov3 gives the follower robot greater possibilities.

(3) Using artificial potential field method to control robot movement. In order to improve the robustness, experiment and test the thresholds of the corresponding parameters to improve the autonomous mobility of mobile robots in dynamic scenarios.

(4) Using TurtleBot as the experimental platform to evaluate the following effects of mobile robots through test scenarios.

**Key Words:** Following robots; Depth vision; Target recognition

# 目 录

摘 要 .....	I
ABSTRACT .....	II
第一章 绪论 .....	1
1.1 论文背景及意义 .....	1
1.2 国内外发展现状 .....	1
1.2.1 跟随机器人发展现状 .....	1
1.2.2 目标跟随发展现状 .....	3
1.2.3 复杂工程与非技术因素分析 .....	4
1.3 论文主要内容和章节安排 .....	4
第二章 基于 KINECT2 的深度图像处理 .....	6
2.1 KINECT2 摄像头简介 .....	7
2.2 KINECT2 的安装和标定 .....	7
2.2.1 kinect2 的安装 .....	7
2.2.2 kinect2 的标定 .....	9
2.3 深度图像在 ROS 中的获取和处理 .....	11
2.4 本章小结 .....	14
第三章 基于 YOLOV3 的目标识别 .....	15
3.1 YOLO 介绍 .....	15
3.1.1 YOLOv1 .....	15
3.1.2 YOLOv2 .....	17
3.1.3 YOLOv3 .....	17
3.1.4 darknet_ros .....	18
3.2 目标检测实验 .....	19
3.2.1 配置参数调试 .....	19
3.3 本章小结 .....	22

第四章 基于 TURTLEBOT 跟随机器人的目标跟随实验 .....	23
4.1 改进人工势场法 .....	23
4.2 机器人跟随人实验 .....	24
4.2.1 实验流程及改进 .....	25
4.2.2 测试结果与分析 .....	28
4.3 本章小结 .....	30
第五章 总结与展望 .....	31
5.1 总结 .....	31
5.2 展望 .....	31
参考文献 .....	32
致谢 .....	34

## 第一章 绪论

### 1.1 论文背景及意义

时过境迁，我国的社会大背景不断发生着变化：社会分工的不断细化，科教产业雄厚，人口红利逐年下降，人力成本不断增加。为了将人类从大量重复或具有危险性的工作中解放出来，重视机器人领域的发展是极为必要的。世界上一些发达国家，将机器人领域作为一个战略高地，力求解放劳动力，发展生产力。

开发一种带自动跟随功能的机器人具有众多现实需求。近年来人工智能的快速发展，为移动机器人的发展创造了良好的条件。文献[1]提到美国在 2011 年就宣布了《先进制造业伙伴关系》计划，德国政府提出的《德国 2020 高技术战略》中的“工业 4.0”，中国提出“工业 4.0”，编撰人工智能白皮书，移动机器人符合智能制造的目标，因而移动机器人正是实现这一现代化目标不可或缺的关键之一。目前机器人的应用范围越来越广，文献[2]介绍了机器人的相关应用，在工业上，移动机器人可以定制化生产零件而不必在一个工位上做一种零件；在物流行业，移动机器人可以分拣和搬运货物，维护仓储系统；在生活娱乐行业，跟随机器人使得机器更具智能化，让生活多一个助手，无缝的融入家庭当中。科技是第一生产力，解放提高生产力也是我国现代化的目标之一。机器人事业可以解放生产力，因此，研究跟随机器人将有助于我国进一步发展制造业，其边际效应将会辐射到医疗，零售，文化等多个行业，具有广阔的市场前景和战略目标。

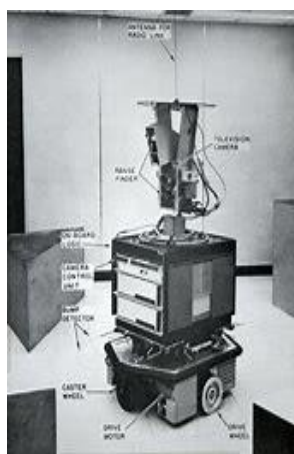
### 1.2 国内外发展现状

#### 1.2.1 跟随机器人发展现状

跟随机器人的发展可以追溯到上世纪 60 年代，国外专家学者们就已经开始了基于视觉跟踪的移动机器人的研究。文献[3]介绍了在 1966-1972 年间机器人的发展，文献[4]中介绍了美国的斯坦福研究院人工智能中心开发了名叫“Shakey”的移动机器人，如图 1-1(a)所示，其搭载了电视摄像机作为视觉传感器，并由场外的两台计算机控制，结合其搭载的测距仪以及碰撞传感器可实现简单的视觉感知及路径规划问题，是当时最成功的移动机器人研究平台，并获得了 IEEE 里程碑奖项。

同样在文献[3]中，1997 年美国国家航空航天局（NASA）发射了火星探测车备受社会

各界广泛关注。在此之后，为了应对火星上复杂的地理环境，NASA 又研制了新一代的机器人命名为“rocky7”如图 1-1(b)所示。



(a) “Shakey”机器人



(b) “rocky7”火星车

图 1-1 移动机器人

文献[5]中，2000 年日本本田公司研制出了一台名为 ASIMO 的仿人机器人，可实现跟随避障功能，并可以通过图像识别算法分辨出手推车上的餐盘，进而完成端起动作，如图 1-2(a)所示。ASIMO 可以实现速度高达 6km/h 的直线奔跑动作。在其研发周期内完成了 7 次迭代改进，但最终因为成本较高，未实现商用。

文献[6]中 2005 年美国的 Boston Dynamics（波士顿动力）机器人公司研发了名为“Big Dog”的军用四足机器人。后经改进推出了“SpotMini”四足机器人，如图 1-2(b)所示，其使用电池替代以往的发动机驱动液压系统，通过前方搭载的 3D 立体摄像头及激光传感器，实现了连贯的避障、跟随、爬楼梯、跳跃、奔跑等自主导航动作。其搭载的机械臂上装有辅助摄像头与机械手共同配合，可实现在人为阻拦情况下对门把手进行定位并完成开门动作。



(a) ASIMO 的仿人机器人



(b) “SpotMini”

图 1-2 跟随机器人



进入 21 世纪，传感器技术有了长足的发展。机器人获得外界信息的渠道逐渐开始丰富了起来。从超声波传感器到激光传感器，检测原理和检测精度都有了巨大提升。配合日益迅速的处理器和相关算法的发展，机器人的研究取得了可观的进展。

我国对机器人技术的研究起步明显的晚于国外。我们国家从 70 年代开始才研究一些定制化的机械手臂，我国“863”计划将机器人技术提升到国家重点发展项目，投入大量的人力物力资源。至今，我国的相关产业链逐渐完善，相关技术迅猛发展，和发达国家的距离进一步显著缩小。

文献[7]介绍了 2014 年由中国科学技术大学自主研发的服务机器人“可佳”，其视觉检测部分采用微软推出的 kinect2 摄像机并结合高分辨率 CCD 摄像机，可以基于人体骨骼模型，实现对人体的识别与跟随。

文献[8]介绍了 2018 年浙江大学 ZJUNlict 团队研发的足球机器人队伍，通过固定于场外的 8 个摄像头对足球目标进行识别，并根据识别出的每个机器人头顶上不同颜色的色块对各个机器人进行定位。

随着国内外对于移动机器人的研究越来越多，一批高等人才也对移动机器人平台进行了深入的研究以及样机研制，做出了若干不菲的成果。但也需要明确，我国的机器人的事业仍然和发达国家如：日本，美国有着相当大的差距。

### 1.2.2 目标跟随发展现状

目标跟随系统根据搭载的传感器不同有以下三种常见的实现方式：基于视觉跟随，基于激光信息跟随，基于多传感器信息融合跟随。

文献[9]中提到，基于视觉信息的跟随有基于人脸跟随，基于轮廓信息跟随和基于模板匹配的跟随等。虽然目前国际上人脸识别已经步入了工业级的水准，但由于人并不是一直面对着机器人的，在实际应用中依然面临着诸多掣肘。基于模板匹配的方法往往在识别速度上表现不佳，且只在水平方向上检测，对旋转的目标缺乏良好的适应性。

基于激光信息跟随主要分为两类，文献[10]使用几何特征识别目标，文献[11]基于运动检测的方法。激光信息不失为一种比较好的跟随方案，但由于激光传感器成本较高，一般在数万人民币以上，目前更多的是在大项目中将其作为多传感器信息融合的一部分，以便弥补各个传感器之间的不足。文献[12]利用多个激光传感器，包括双目摄像机，激光雷达，超声波雷达等实现机器人跟随。

自 2006 年深度学习开始兴起, 2012 年取得重大突破, 以基于卷积神经网络的目标跟随凭借其对深度特征强大的表示能力, 该方法得到了充分发展。文献[13]中 Cirshick 等人在 R-CNN 中提出基于 Region Proposal 的深度学习目标检测框架。文献[14]中 Ren 等人提出 Faster R-CNN 目标检测方法使用 VGG-16 网络模型在 K40GPU 上达到 5fps。文献[15]中还介绍了将分类和定位一步到位的 YOLO, DenseBox 等方法, 在行人, 车辆, 人脸等不同场景取得了前所未有的效果。

### 1.2.3 复杂工程与非技术因素分析

本文的复杂工程特征包括, 在驱动层面, 解决了 freenect2 对显卡驱动包括 libusb, opencv 的相关依赖性问题; 通过修改 image\_method, reg\_method 解决 freenect2 和 iai\_kinect2 在建立连接中的兼容性问题。在应用层面, 验证了虚拟机的 kinect2 外设的不支持; 建立 kinect2 的视场模型确定安装高度和跟随距离; 在下位机上由于配置问题不能满足识别速度需求, 借用配置更高的上位机配置分布式 ROS 进行解决。在问题协调方面, 由于实验室只有一台能用的 kinect2, turtlebot 机器人, 需要和两个人共用 kinect2, 三个人共用 turtlebot。在相关协调下均完成了对课题的实验需求。

机器人的发展需要相关的伦理协同发展, 文献[16]中阐述了相关人工智能与伦理的冲突, 详细说明了发展伦理的必要性。机器人极大的提高了社会生产力的同时, 也不可避免的替代了很多工作, 这涉及到政治经济学的问题。文献[17]中针对人工智能对意识形态的影响进行了详细的分析并提出了若干化解方案。

## 1.3 论文主要内容和章节安排

本文的主要研究的是移动机器人自动识别目标, 在一定范围内对目标的跟随。从功能上主要分为三个模块, 深度图像获取, 目标的识别, 目标的自动跟随。本文的主要脉络如下:

第一章“绪论”主要介绍论文背景及意义, 以及国内外跟随机器人及相关技术发展现状。

第二章“基于 kinect2 的深度图像处理”主要介绍 kinect2 在 ROS 上的使用, 尝试对该传感器在深度像素失效时进行优化处理。

第三章“基于 YOLOv3 的目标识别”叙述从 YOLOv1 到 YOLOv3 逐渐优化的过程, 利用封装成 ROS 节点的 darknet\_ros 实现对人体的识别。与此同时, 选择并调试适合的 YOLO

提供的权重文件供目标跟随程序调用。

第四章“基于 TurtleBot 跟随机器人的目标跟随实验”则利用已识别的障碍物和目标编写控制策略，调整控制参数以期获得鲁棒性更强的控制策略。设立跟随实验场景，验证并优化跟随系统的性能，获得更加稳定的目标跟随效果。

第五章“总结与展望”对上述内容进行反思和归纳，指出实验中存在的问题以及对未来的展望。

## 第二章 基于 kinect2 的深度图像处理

传统摄像头采集 RGB 值,为了更好的实现功能,应该尽可能的获取更多的特征。普通摄像机仅能获取每个像素 RGB 值,但 3D 相机可以获取到 RGB-D 值,即除了图像信息还有深度信息。深度信息可以以不同的形式表现出来,通过颜色和深度的映射可以获得彩色的深度图,如图 2-1 所示。红色越深代表深度值越小,蓝色越深代表深度值越大。深度值结合彩色图还可以进行三维重建,如点云的实现。一般来说,根据需求有许多种 3D 相机可以选择。

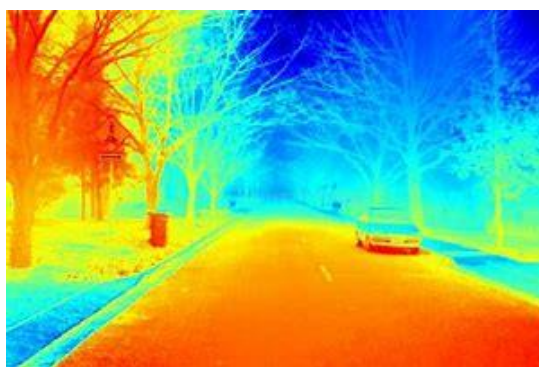


图 2-1 深度图

常见的 3D 相机包括三类,分别是结构光与编码光,立体视觉,TOF 相机,LIDAR 相机。其优缺点总结如表 2-1 所示:

表 2-1 不同 3D 相机的优缺点对比

3D 相机	优点	缺点
结构光与编码光相机	精度高	易受干扰,短距离
立体视觉相机	适应性强,抗干扰强	测量范围小,精度适中,距离较近,受表面纹理影响
TOF 相机	距离误差是线性的	分辨率较低,会产生运动模糊
LIDAR 相机	距离远,有固定误差模型	易受干扰,成本高,耗时

结构光 3D 相机与编码光 3D 相机通过发射特定光源投射到物体,使得物体表面特定的呈现图案。通过比较原始的结构光或编码光与物体表面的图案进行分析就能得到物体的深度信息。立体视觉相机则通过的两个已知传感器之间的距离计算被测对象的深度,原理和小孔成像有异曲同工之处。TOF 相机直译为飞行时间相机,通过计算发射的光和接收的光的相位差或飞行往返的时间差进行测距。其精度能达到像素级别,适合远距离测量。IDAR 工作原理是发射激光到物体表面,然后接收物体的反射光信号,比较与初始激光的不同从

而计算出距离。

根据所需 3D 相机的性能参数如视场角，分辨率，距离精确度和最大最小探测距离和一些非性能参数如尺寸，重量，成本，封装，通信接口等，衡量对比选择 kinect2 作为获取彩色信息和深度信息的设备。

## 2.1 kinect2 摄像头简介

Kinect2 是一种 TOF 相机，提供多种模式、选项和软件开发工具包 (SDK)。kinect2 深度相机的技术特征如表 2-2 所示：

表 2-2 kinect2 的技术特征

技术特征	参数
颜色图像	分辨率 1920x1080, 30fps
深度图像	分辨率 512x424, 30fps
检测范围	0.5~4.5m
水平角度	70°
垂直角度	60°

根据官网手册给出的提示，在超出活动 IR 照明遮罩，滤波异常，多路径干扰等情况会出现无法获取像素深度的情况，此时像素深度值 0。因此，应在安装部署方面尽可能降低出现无效像素的情况，并且在获取深度信息的同时进行一定的预处理。

## 2.2 kinect2 的安装和标定

### 2.2.1 kinect2 的安装

kinect2 能够获取丰富的彩色信息和深度信息，常规的图像识别算法需要待跟随的人大部分身体被摄像头捕捉到，因此对摄像头的安装高度有一定要求。由本文上一小节的列出的相机技术特征可以建立如图 2-2 示的场景：

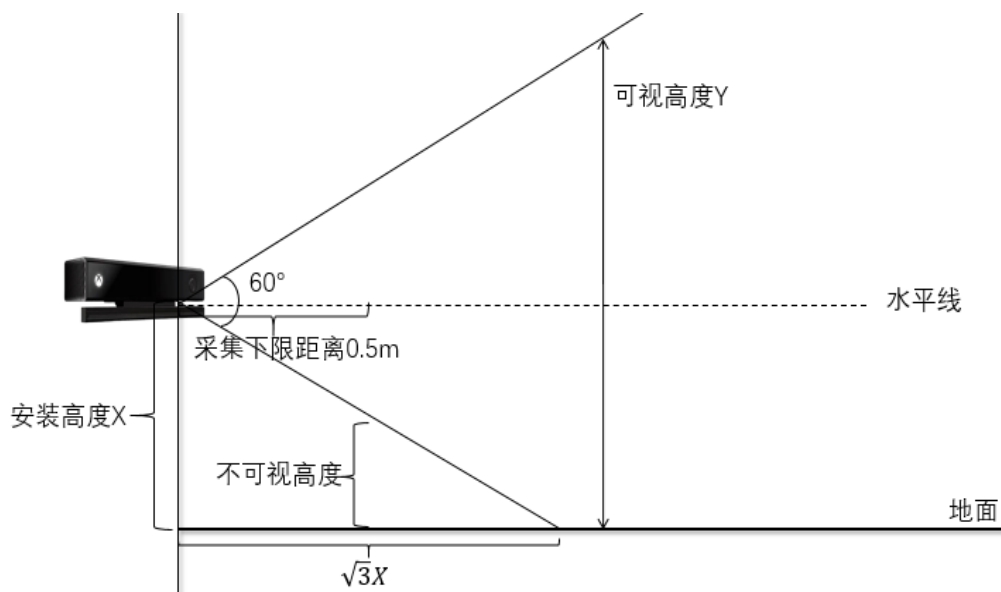


图 2-2 kinect2 的安装场景

已知摄像头距离目标范围为 0.5m~4.5m，垂直方向上视场角为  $60^\circ$ ，设摄像头安装高度为  $X$ ，可视高度为  $Y$ ，目标距离摄像头垂直平面为  $S$ ，据图可得下列式 2-1 表述的关系：

$$Y = \begin{cases} 2\frac{S}{\sqrt{3}} & 0.5 < S < \sqrt{3}X \\ X + \frac{S}{\sqrt{3}} & \sqrt{3}X \leq S < 4.5 \end{cases} \quad (2-1)$$

下面根据上述关系列写不同身高的人需要 kinect2 完整观察到的安装高度如表 2-3：

表 2-3 完整观察到的部分安装高度

安装高度 $X$ (cm)	kinect2 完整 视角的最小 距离 (cm)	kinect2 完整 视角的最小 高度 (cm)	距离 100cm 高度 (cm)	距离 120cm 可见高度 (cm)
50	87	100	108	119
70	121	140	115 可见, 12 不可见	139
90	156	180	115 可见, 32 不可见	139 可见, 21 不可见
110	191	220	115 可见, 52 不可见	139 可见, 41 不可见

根据表 2-3 容易看到，如若选取跟随距离为 100cm，那么摄像头可采集的高度仅为 115cm。过小的人像范围容易造成待跟随目标的识别精度降低，因此跟随距离选择 120cm。又考虑到完整视角最小高度过低过高容易造成只能通过身体的一部分进行识别，同样会降低识别精度，对比表格选择 90cm 安装高度。因此，本文采取 90cm 的安装高度以及 120cm

的跟随距离进行后续的研究和实验。

### 2.2.2 kinect2 的标定

Kinect2 的彩色图像摄像头和深度图像摄像头是相互独立的，并且摄像头的装配等因素也会导致颜色图和深度图不对应。因此，如果想在相应的彩色图像上获取目标的深度信息，就需要彩色图像和深度图像对齐，这就是标定。标定过后像素坐标转换到物理坐标来计算距离，完成深度信息的获取。

在相机成像系统中，存在四个坐标系，分别为世界坐标系、相机坐标系、图像坐标系和像素坐标系。如式 2-2 所示。

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{dX} & -\frac{\cot\theta}{dX} & u_0 \\ 0 & \frac{1}{dY\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} \quad (2-2)$$

其中， $3 \times 3$  的矩阵代表仿射变换， $4 \times 3$  的矩阵代表透视投影，二者合称为内参矩阵。 $2 \times 2$  的矩阵为刚体变换矩阵，也被称为内参矩阵。 $(U, W, V)$  为在世界坐标系下一点的物理坐标， $(u, v)$  为该点对应的在像素坐标系下的坐标， $Z$  为尺度因子。我们将

$$\begin{pmatrix} \frac{1}{dX} & -\frac{\cot\theta}{dX} & u_0 \\ 0 & \frac{1}{dY\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{f}{dX} & -\frac{f\cot\theta}{dX} & u_0 & 0 \\ 0 & \frac{f}{dY\sin\theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2-3)$$

称为相机的内参矩阵。其中  $f$  为像距， $dX, dY$  分别表示  $X, Y$  方向上的一个像素在感光板上的物理长度， $u_0, v_0$  分别表示感光板中心在像素坐标系下的坐标， $\theta$  表示感光板的横边和纵边之间的角度。我们将矩阵： $\begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$  称为相机的外参矩阵，外参矩阵中  $R$  表示旋转矩阵， $T$  表示平移矢量。成像模型如下式 2-3 所示：

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{dX} & -\frac{f\cot\theta}{dX} & u_0 & 0 \\ 0 & \frac{f}{dY\sin\theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} \quad (2-4)$$

相机的标定大部分采用的是张正友标定法。张正友标定法是一种比传统标定法精度高，但比自标定法更便于操作。张正友标定法利用一张棋盘格，这张棋盘格有多个规格满足对

标定的不同需求，一般有  $5 \times 7 \times 0.03$ ， $7 \times 9 \times 0.025$ ， $9 \times 11 \times 0.02$  三种规格的棋盘格。这样将世界坐标系固定在棋盘格上，由于标定板的世界坐标系是给定的如图 2-4 所示的一张  $5 \times 7 \times 0.03$  规格的标定板，每个黑色方块和黑色方块的距离是已知的，角点与角点之间的距离是已知的，因此能求出世界坐标系中的每个点的物理坐标。之后求解内参矩阵与外参矩阵的积，最后对两个矩阵分别求解即可。

对于这上述步骤，iai\_kinect2 已经封装好了，在 README 中有详细的标定说明。标定的各个参数将存储在 kinect2/bridge/data 目录下，每一次标定都应按照标定颜色图，标定深度图，同步标定深度和颜色相机。最后这些参数都将以 yaml 的格式进行存储。根据作者提供的误差估计，标定的偏差在固定的 24mm 左右，验证后如图 2-3 所示。

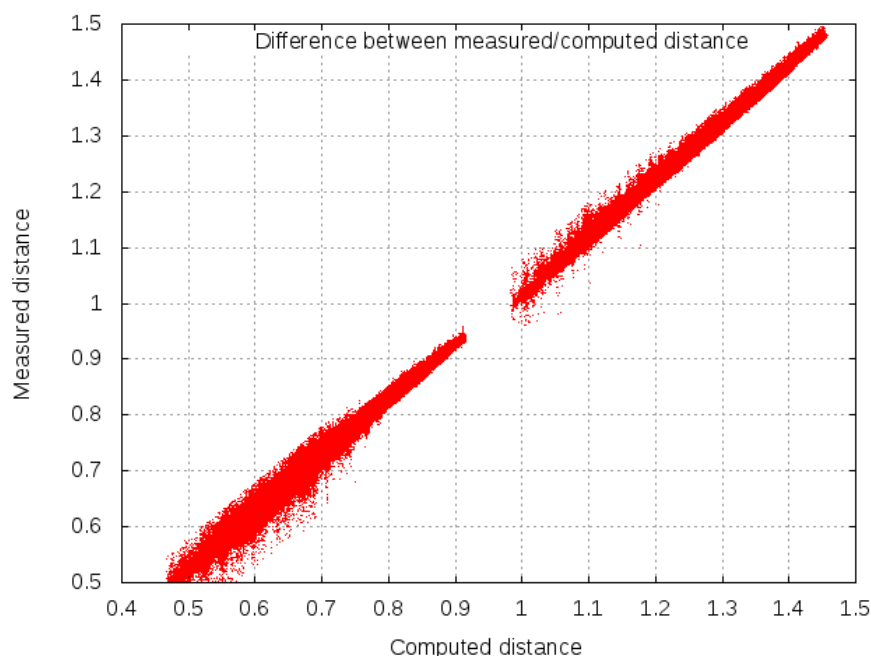


图 2-3 标定偏差



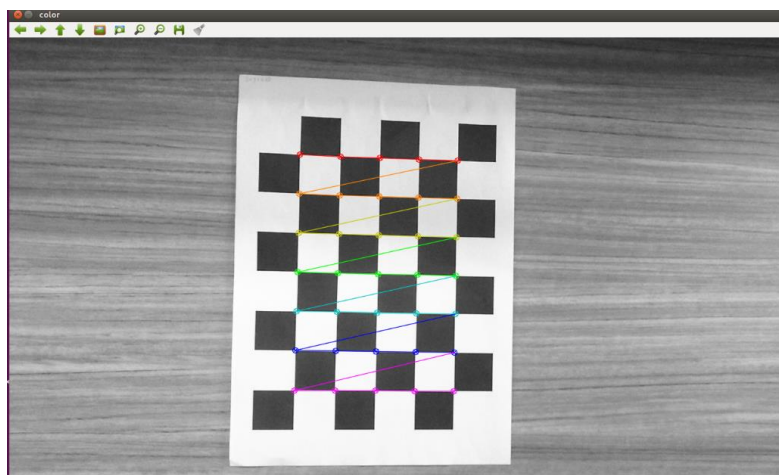


图 2-4 标定过程



图 2-5 未标定的深度图像

图 2-6 标定后的深度图像

图 2-4 显示标定过程中标定程序对黑白框交叉点的定位。图 2-5 为未标定的深度图像，点云和彩色图有较大的偏移。图 2-6 为标定后的深度图像，可以看出点云和彩色图像重合度较高，能更好的反应物体的深度信息。

## 2.3 深度图像在 ROS 中的获取和处理

根据文献[18]，kinect2 通过 libfreenect2 驱动，iai\_kinect2 作为驱动和 ROS 的中间桥梁使得 kinect2 在 ROS 中得以很方便的进行调用 kinect2。其基本流程如图 2-7 所示：

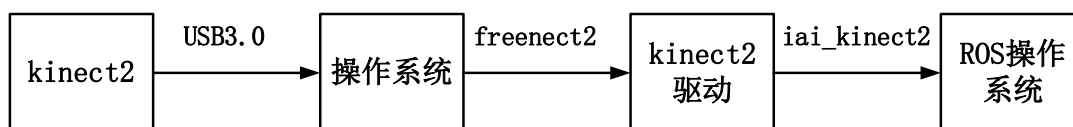


图 2-7 ROS 调用 kinect2 获取深度图流程

iai\_kinect2 提供三种清晰度，分别为高清(1920×1080)，半高清(960×540)以及原始深

度图(512×424)。不同清晰度又提供颜色图，深度图，压缩的颜色图和深度图以及点云等。同时在参数控制中还可以调整 fps，最大深度，最小深度，使用 CPU，GPU，坐标变换等，通过 iai\_kinect2 提供的话题可以方便的调用不同类型的图像。

在本文第一节中提到过由于环境光，距离，设备本身的滤波等因素，导致不可避免地产生噪声和误差，具体表现为深度图像会产生一定的空洞。文献[19]中 kourosh 对 kinect2 地参数性能和成像质量做了比较详尽地分析。理论误差模型说明深度图像地标准差为：

$$\sigma_z = \left(\frac{m}{fb}\right) Z^2 \sigma_{d''}, \quad \sigma_x = \left(\frac{mx}{f^2b}\right) Z^2 \sigma_{d''}, \quad \sigma_y = \left(\frac{my}{f^2b}\right) Z^2 \sigma_{d''} \quad (2-5)$$

其中  $\sigma_{d''}$  是一个满足高斯分布地随机变量地标准差。在 XYZ 轴的误差均与相机到物体地距离平方成正比关系。

文献[20]中 Park 同样使用了非局部均值滤波来处理深度图像，同时也结合了彩色图像中的一些特征信息，例如边缘信息、梯度信息等来增强深度图像的修复，在离线状态下，这些滤波在平滑场景中可以取得不错地恢复效果。但在实时处理方面却遇到了一些瓶颈，越精细地修复所需处理的数据量就会越大。在实际实验中，iai\_kinect2 能提供每秒 30 帧左右的图像，选取半高清的清晰度 960×540，那么一秒内将要处理 30 帧的图像。在不考虑后续目标识别，上位机和下位机之间的通信，跟踪算法对计算机资源的占用，延迟已经十分严重了。因此，参考文献[21][22]提出了针对深度图像的修复，本文采用的方式尽可能降低图像处理对计算机资源的占用，仅对局部的异常值进行平滑处理。过程如下：

首先将彩色图传给目标识别模块返回目标识别框，其次根据目标识别框的信息定位深度图像的位置，利用 opencv 进一步绘制目标轮廓，最后对轮廓中的深度数值取平均值作为该目标的深度信息。过程如图 2-8 所示。

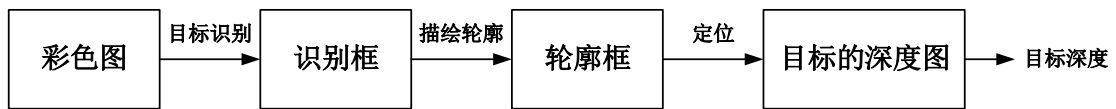


图 2-8 获取目标深度



图 2-9 原始图像



图 2-10 调整轮廓阈值

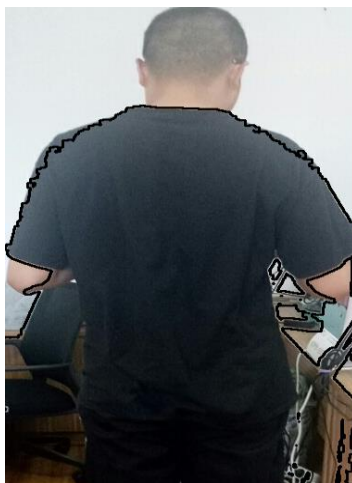


图 2-11 绘制轮廓

粗略的描绘目标轮廓的意义是：尽可能地减小背景对目标深度地影响，利用 `opencv` 内置地函数能很快地排除掉一些非相关区域。具体步骤如下：

- (1)获取原始图像，如图 2-9；
- (2)用  $3 \times 3$  地卷积核进行腐蚀处理；
- (3)将处理后地图片转为灰度图；
- (4)根据灰度图调整轮廓阈值,测试并设置到合适参数，如图 2-10；
- (5)绘制轮廓，如图 2-11。

对比图 2-9 和图 2-11 可以看到，黑线框选区域缩小了待跟随目标的范围，从而使得深度信息尽可能地避免了环境信息对待跟随目标地干扰。不足之处在于未能完全排除环境因素

对待跟随目标深度的影响，对轮廓内的深度取平均值可能提高对距离信息获取的准确度。

## 2.4 本章小结

本章首先对 kinect2 的相关参数进行了介绍，接着探究了 kinect2 的相关部署问题，为了保证获取到的颜色图和深度图的质量对其进行了标定。针对深度图在某些情况容易出现空洞的问题，在尽量不增加计算机计算负荷的情况下，采用勾勒待跟随目标的轮廓之后平均化处理。通过上述相关处理，基本上满足本实验的实时性的需求，但是彩色图转灰度图大概率将部分背景包含了进去，对深度信息的准确性产生影响，采用均值化处理尽可能降低其造成的影响。

## 第三章 基于 YOLOv3 的目标识别

### 3.1 YOLO 介绍

目标识别需要用矩形框框选识别到的目标，即两个任务：目标定位和分类。在深度学习中，主流的算法框架大致分为一步到位（one-stage）与两步到位（two-stage），一步到位的典型算法有 YOLO 系列，即目标定位和分类同时共用一个网络。两步到位的典型算法有 R-CNN 系列，输入图像经过候选框生成网络，再经过分类网络。一步到位和两步到位相比，前者速度更快，精度偏低。在可接受的范围内，YOLO 系列相当于对速度和精度做出一个权衡。随着不断优化直至 YOLOv3，YOLOv4 的出现，在保持较高精度的同时也保持了较高的运行速度。正是由于具备边缘计算和实时性较高的特点，YOLO 系列备受青睐。

#### 3.1.1 YOLOv1

根据文献[23]，在网络架构方面，YOLOv1 借鉴 GoogleLeNet，输入图像大小为为  $448 \times 448$ ，经过 24 个卷积层后，4 个池化层（ $2 \times 2$  步进为 4）以及 2 个全连接层，输出为  $7 \times 7 \times 30$  的张量，网络架构如图 3-1 所示：

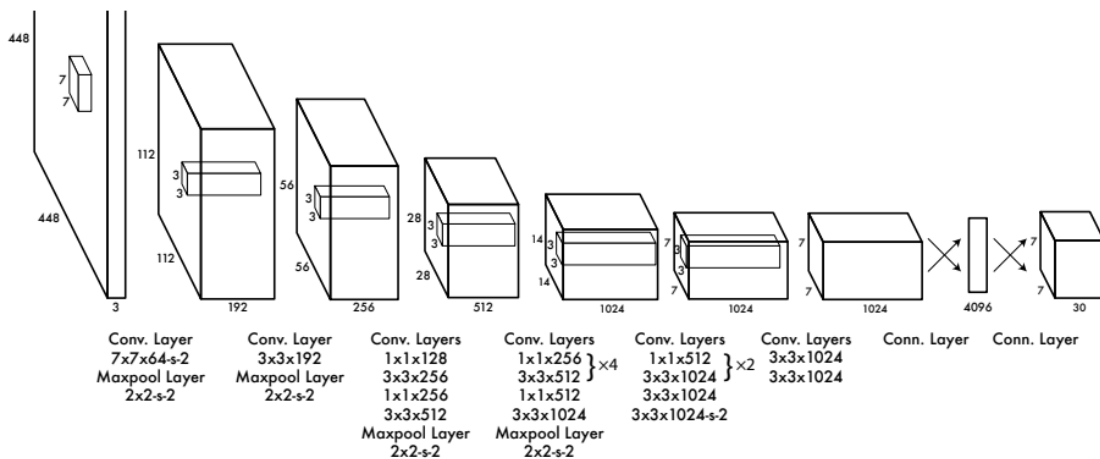


图 3-1 YOLOv1 网络架构

其中  $7 \times 7$  代表将图像分为  $7 \times 7$  的格子，每个格子独立的预测格子内的物体。30 是  $(4+1) \times 2 + 20$  的值，其中  $4+1$  分别代表矩形框的终点坐标  $x, y$ ，长宽  $w, h$  和置信度  $c$ ，2 代表预测两个矩形框，20 代表预测的 20 个类别。由于将图像分成  $7 \times 7$  的网格，每个网格至多检测一个物体，故一张图像至多检测出 49 个物体。即该网络结构最大的不足便是对邻近的小物体识别不太好。

在训练方面，YOLOv1 所有的损失函数都是用平方和误差公式来衡量。主要衡量三个

方面，坐标，预测框和类别。构造的损失函数具体分析如图 3-2 所示：

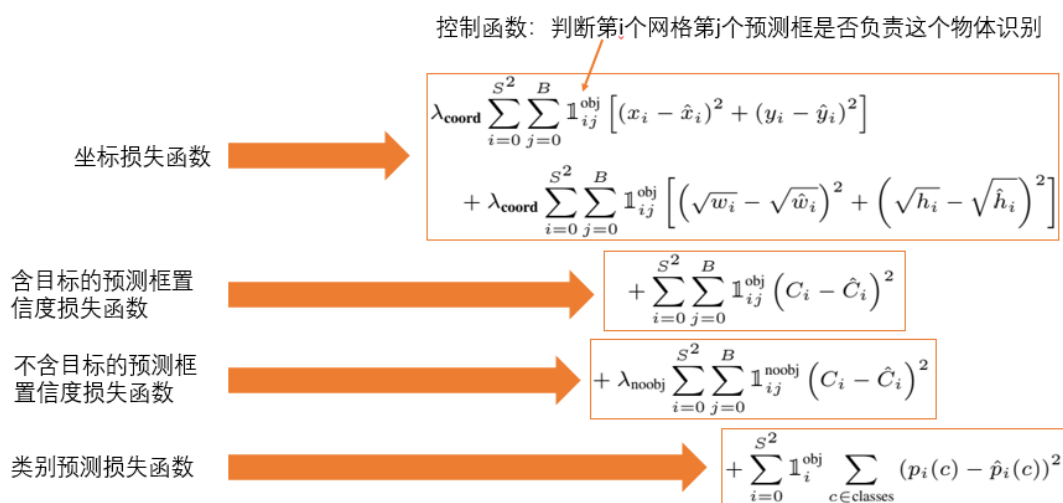


图 3-2 YOLOv1 的损失函数

其中控制函数表示该格点处是否含有物体，如果含有则为 1，不含有则为 0。同理，在第三部分置信度损失函数中，当该格点不含有物体时，将置信度标签设为 0；若含有物体时，置信度标签为 IOU 数值（其中 IOU 数值定义为预测框和真实框交集的面积除以并集的面积）。第五行中，对应的类别位置，该标签数值为 1，其余位置为 0，与分类网络相同。 $\lambda_{coord}$ 代表相应的权重，在整个 49 个格点当中，往往只有几个格点含有物体。因此，有必要采取点措施才能使得 mAP 不要太低，而加个权重进行优化能有效的提升 mAP 指标。论文中和其他算法的比较如表 3-1 所示（硬件均为 GPU Titan X）：

表 3-1 YOLOv1 和其他算法的性能比较

实时检测器	训练	mAP	FPS
100Hz DPM	2007	16.0	100
30Hz DPM	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
Fastest DPM	2007	30.4	15
R-CNN Minus R	2007	53.5	6
Faster R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
YOLO VGG-16	2007+2012	66.4	21

其中，YOLO 更倾向于从整体上来预测物体，虽然在准确率上不如滑动窗口的检测方式，但是在误检率上比它更低。更轻量级的 Fast YOLO 甚至能达到 155FPS 的情况下达到 mAP 比较高的水准。

### 3.1.2 YOLOv2

根据文献[24]，YOLOv2 在 YOLOv1 的基础上进行了下列改进：表 3-2 显示了改进措施对 YOLOv2 性能的提升：

表 3-2 不同优化对 YOLOv2 的提升

对 YOLOv1 的改进措施	VOC2007 mAP
YOLOv1	63.4
批归一化（Batch Normalization）	65.8
分类网络高分辨率预训练（High Resolution Classifier）	69.5
锚框卷积（Convolutional With Anchor Boxes）	69.2
锚框聚类（Dimension Clusters）	69.6
绝对位置预测（Direct location prediction）	74.4
细粒度特征（Fine-Grained Features）	75.4
多尺寸训练（Multi-Scale Training）	76.8

此时 YOLOv2 的精度已经达到当时的业界前沿。

基于以上改进，分类与检测联合训练(jointly training on classification and detection data)打造了 YOLO9000，顾名思义，能检测 9000 种不同的物体。同时，YOLO9000 不需要对所有训练集进行标注，减轻了标注成本和时间。

### 3.1.3 YOLOv3

根据文献[25]，YOLOv3 的基础网络结构(backbone)继承于 YOLOv2 的 darknet-19，升级到了 darknet-53，加深了网络层数，引入了 Resnet 中的跨层加和操作也叫残差网络。残差网络的加入是为了解决深层次的网络梯度消失，否则这样的网络至今找不到比较好的方式进行训练。残差网络的加入使得网络层数更深的 YOLOv2 在性能上有了量变到质变的提升,其他目标识别算法的对比如表 3-3 所示：

表格 3-3 YOLOv3 和其他算法的对比

算法	mAP	时间
SSD321	28.0	61
DSSD321	28.0	85
R-FCN	29.9	85
SSD513	31.2	125
DSSD513	33.2	156
FPN FRCN	36.2	172
RetinaNet-101-800	37.8	198
YOLOv3-320	28.2	22
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

从表 3-3 可以看到，YOLOv3 牺牲比较小的精度而获得了比其他算法快两倍左右的速度。单张处理速度仅需 22ms，简化后的 YOLOv3-tiny 能将速度进一步提升。

#### 3.1.4 darknet\_ros

文献[26]中，darknet\_ros 是 Marko Bjelonic 创建的用于实时目标检测的 ROS 节点，是对 YOLOv3 的一种 ROS 封装。它提供几种预训练模型，支持 CPU 和 GPU，可以自己训练网络用于实施目标检测。

整个工程分为三个部分，第一部分 darknet 是用 C 和 CUDA 写的 YOLOv3 主体。第二部分是 darknet 和 ROS 的桥梁，config 文件夹用于修改订阅和发布的 ROS 主题，修改识别的阈值等，launch 文件夹用于修改相关配置(预训练模型，使用 CPU 还是 GPU 等)，yolo\_network\_config 文件夹用于存储相关的配置文件，如权重文件，src 放置主程序。第三部分是 darknet\_msgs，自定义消息话题。由此可知，预测框的消息发布格式如下：

```
string Class
float64 probability
int64 xmin
int64 ymin
int64 xmax
int64 ymax
```

此外，darknet\_ros 提供的话题还有：发现物体的数量，已绘制好预测框的图像。



## 3.2 目标检测实验

参考文献[27][28]了解到需要对 YOLO 模型参数进行进一步设置，以此来提高识别速率。darknet\_ros 所有参数采用默认设置，利用 kinect2 采集实时图像，经过预处理之后图像传输给 darknet\_ros 进行处理，返回已完成目标识别的图像，评估结果。

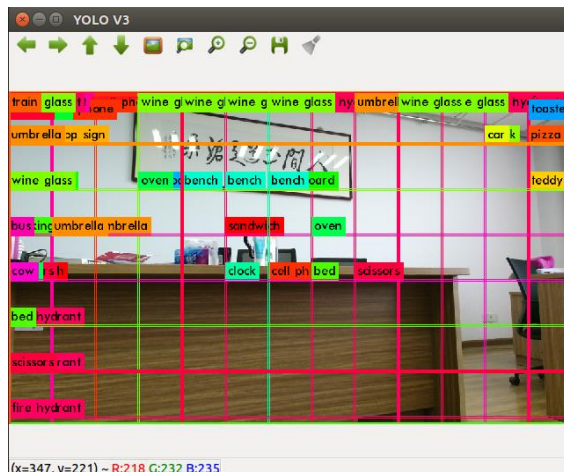


图 3-3 物体框过多

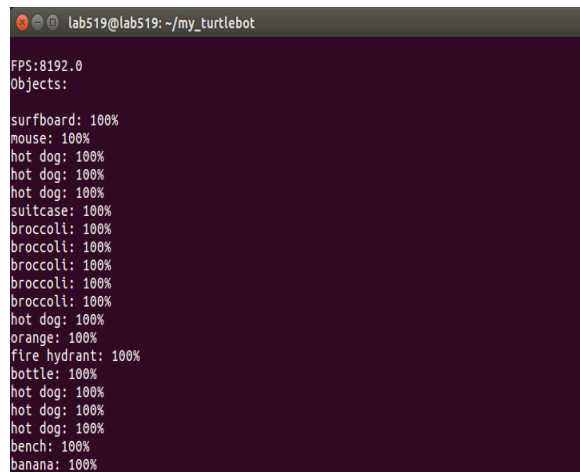


图 3-4 实时帧数

由图 3-3，图 3-4 知，目标识别的精度已经丢失，产生了大量的其他类别的识别框，识别速度极快且准确率显示为 100%。考虑到出现大量的无效框，基本可以定位到是模型本身的问题。默认设置采用 YOLOv2.cfg 模型文件，该配置文件使用的是 darknet-19 的基础架构，而 darknet\_ros 对数据的处理基于 darknet-53。因此，cfg 文件应该选择以 darknet-53 为 backbone 的配置文件。网上提供有 YOLOv3，YOLOv3-tiny，YOLOv3-voc，YOLOv3-coco 四种符合上述需求的配置文件。

### 3.2.1 配置参数调试

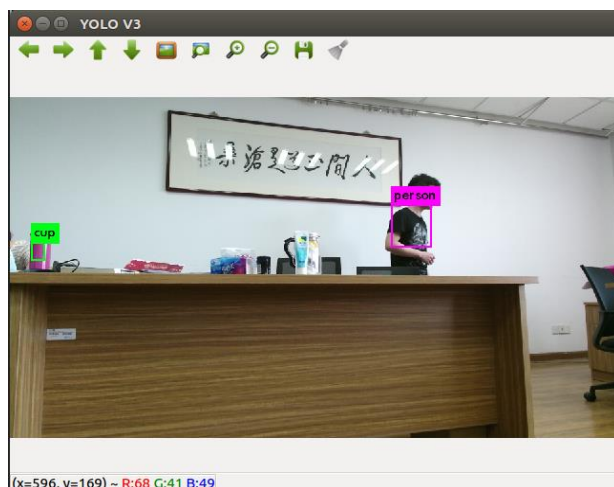


图 3-5 目标检测框

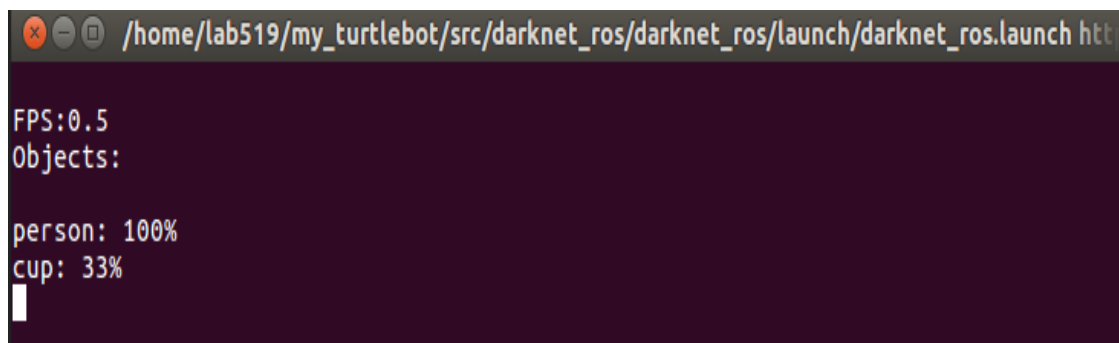


图 3-6 目标检测实时帧数

如图 3-5 所示，在换用 YOLOv3 系列的配置后，检测效果恢复了正常，识别人的精度很高，能满足机器人跟随的需要。但同时可以看到 FPS 只有 0.5 左右，如图 3-6 所示，检测速度满足不了需求。尝试采用不同的配置文件，与此同时换用 GPU 进行识别，得到如下表 3-4 所示：

表 3-4 不同配置的识别速度

	YOLOv3	YOLOv3-tiny	Yoliv3-voc
单张图片+CPU	25.04(s)	2.43(s)	10.31(s)
单张图片+GPU	0.42(s)	0.37(s)	15.12(s)
视频流+CPU	0.1fps	0.5fps	0.1fps
视频流+GPU	15fps	48fps	21fps

其中 CPU 主频为 2.5GHz，GPU 型号为：GTX 1060，测试环境为：ubuntu18.04, ROS melodic

为了进一步提高目标识别的精度和准确度，对各个参数进行多次实验。实验改变模型参数和 ROS 相关参数识别一张图片，通过不同参数识别所用时间来评价实验效果。结果如表 3-2 所示：

表 3-5 调整参数

识别阈值 (threshold)	减少识别类别 (detection_classes)	关闭其它无关 话题 (enable topic)	实 时 显 示	10 次实验 平均耗时 (s)	识别错 误次数/ 总次数
0.1	否	否	否	0.32	0/10
0.1	是	否	否	0.32	0/10
0.1	否	是	否	0.32	0/10
0.1	否	否	是	0.28	0/10
0.2	否	否	否	0.34	0/10
0.2	是	否	否	0.34	0/10
0.2	否	是	否	0.34	0/10
0.2	否	否	是	0.30	0/10
0.3	否	否	否	0.37	1/10
0.4	否	否	否	0.37	1/10
0.5	否	否	否	0.37	4/10
0.6	否	否	否	0.37	1/10
0.7	否	否	否	0.37	4/10
0.8	否	否	否	0.37	7/10

根据表 3-5 我们可以推断出，修改 YOLOv3-tiny.yaml 文件的识别类别，关闭 ros.yaml 文件中的动作、话题的发布没有明显提高检测速度的作用。但减少识别类别会将物体错误地分类为其他物体，只保留 person 类别时所有物体都被分类为 person。将实时检测关闭能略微提高检测速度，查看显卡被检测任务所占用的显存为 477MB，进一步佐证了该想法。识别阈值的提高会导致识别为错误地物体，由于本文设计的是跟随人地机器人系统，除了人以外的物体都是障碍物，其他物体识别错误率对本系统影响不大。

综上所述，本系统所用的配置参数为：使用权重 YOLOv3-tiny.weight，使用 GPU, nvidia-440, cuda-10.2, cudnn-7.7，使能 opencv，识别阈值 0.3，关闭无关话题，关闭实时显示，不进一步减少识别类别。

### 3.3 本章小结

本章粗略的介绍了 YOLO 系列的相关基础，大致分析了整个 YOLOv3 的 ROS 封装包 `darknet_ros` 的使用方法。之后做了对物体识别的相关实验和测试，改进或优化了相关参数和编译条件，在使用 GPU 后，验证了实时识别的可行性。

## 第四章 基于 TurtleBot 跟随机器人的目标跟随实验

### 4.1 改进人工势场法

跟随机器人的另一大难点是控制策略的问题，学界也有大量的相关研究。根据文献[29]，人工势场法是 1986 年 Khabit 提出的一种虚拟力的控制策略，其基本思想是：以目标点为中心构造引力场，以障碍物为中心构造斥力场，共同作用形成向目标点的梯度场。但这里的力场与通常的力场有些不一样，引力场随着离力场中心越远，力的大小越大；斥力场随着离力场中心越近，力的大小越大。这样就能保证机器人在跟踪目标的同时，又能比较平滑的避开障碍物。

Khabit 提出的人工势场法存在目标不是势场的局部最小值的情形，此时机器人的受到的引力小于受到障碍物的斥力。余振中等[30]提出将机器人与目标的相对距离加入到势场的构建中，以此避免该种情形的发生。

设目标在空间中的位置为  $X = [x, y]^T$ ，则引力函数定义为：

$$U_{at}(X) = -\frac{1}{2}k(X - X_{goal})^2 \quad (4-1)$$

其中  $X - X_{\epsilon}$  为机器人与目标的相对距离， $k$  为目标位置增益系数。

此时引力可写成：

$$F_{at}(X) = -grad[U_{at}(X)] = -k(X - X_{goal}) \quad (4-2)$$

同样可定义斥力函数：

$$U_{re}(X) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)^2 (X - X_{goal})^n & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases} \quad (4-3)$$

其中  $\eta$  为障碍物位置增益系数， $\rho_0$  为障碍物影响距离的常数， $\rho$  为机器人和障碍物的距离， $\eta$  为大于 0 的任意常数。这里目标和机器人相对距离存在的意义是：确保目标处是全局势场的最低点。

当  $X \neq X_{goal}$  时，斥力可写成：

$$F_{re}(X) = -grad[U_{re}(X)] = \begin{cases} F_{re1} + F_{re2} & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases} \quad (4-4)$$

其中

$$F_{re1} = \eta \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2} (X - X_{goal})^n \quad (4-5)$$

$$F_{re2} = \frac{n}{2} \eta \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 (X - X_{goal})^{n-1} \quad (4-6)$$

故机器人总受力为:  $F = F_{at} + F_{re}$ , 只要机器人沿着力的方向行驶, 便可到达目标。代码的实现参考文献[31][32]给出的移动机器人路径规划。

为了验证代码的可行性, 创建场景进行仿真。仿真环境为 python3.7, 在图上随意布置蓝色的障碍物, 利用算法规划从左上角到右下角的路径, 结果如图 4-1 所示。

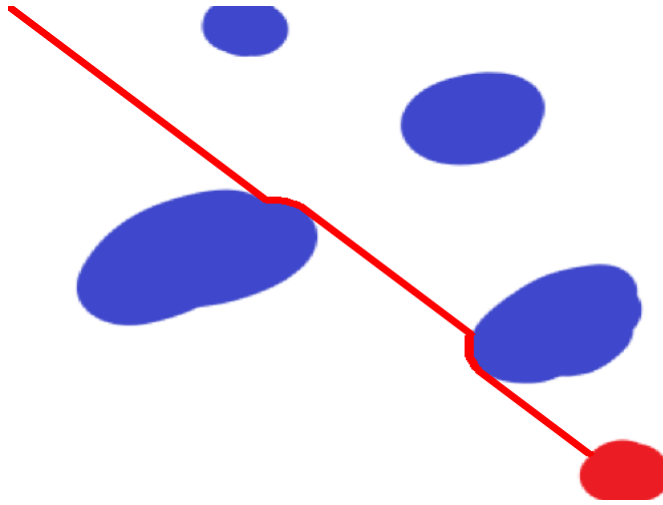


图 4-1 效果图

由图 4-1 可知, 通过上述方法对路径的规划较好的避开了障碍物, 有一定的可行性。该方法优先选择了直线前进。在经过数次仿真实验之后, 均取得了良好的测试效果。

## 4.2 机器人跟随人实验

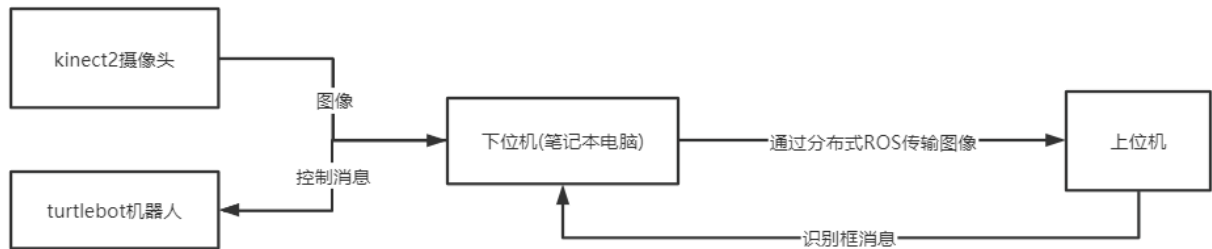


图 4-2 实验硬件连接

如题 4-2 所示, kinect2 通过 libfreenect2 及 iai\_kinect2 在 ROS 中读取摄像头图像, 由于下位机不带 GPU 以及下位机需要随机器人一起移动, 采用分布式 ROS 将图像传输给上

位机进行图像识别后将识别框消息传输给下位机。接着通过下位机将控制消息通过 USB 传输给机器人达到跟随人的目的。其中分布式 ROS 采用 SSH 的方式进行连接，ROS master 节点在上位机启动，下位机启动 `kinect2_bridge.launch`, `minimal.launch`, `image_processing.py` 以及 `turtlebot_follower.py`。

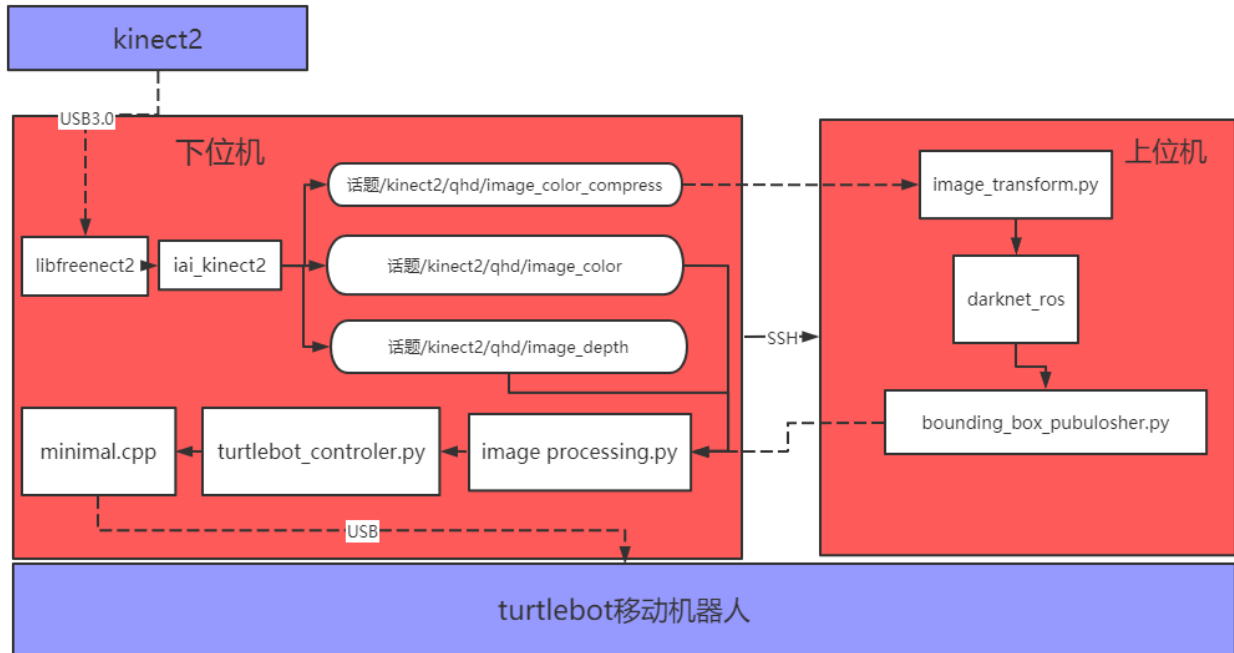


图 4-3 完整控制流程

完整控制流程如图 4-3 所示。首先 kinect2 通过 USB3.0 与下位机连接，libfreenect2 和 iai\_kinect2 读取深度信息和彩色信息并发布三个话题，分别是分辨率为 qhd 彩色图压缩话题，分辨率为 qhd 的彩色图话题以及分辨率为 qhd 的深度信息话题。上位机与下位机通过 SSH 通信，ROS MASTER 进行调控，image\_transform.py 订阅彩色图压缩话题进行解压缩。darknet\_ros 订阅 image\_transform 话题进行目标识别，bounding\_box\_publisher.py 订阅 darknet\_ros 发布的图像框信息，处理后将其通过 ROS MASTER 被 image\_processing 订阅，结合彩色图信息和深度图信息得到目标的深度信息和障碍物的深度信息，将其发布给 turtlebot\_controller.py 运行控制策略，最后调用 minimal.cpp 控制 turtlebot 移动机器人行走，完成机器人跟随人的目的。

进行实验后，针对出现的部分问题做出了以下改进。

#### 4.2.1 实验流程及改进

根据本文第二三章的实验，图像识别的速度能到 50 帧，有理由相信问题并不在上位机

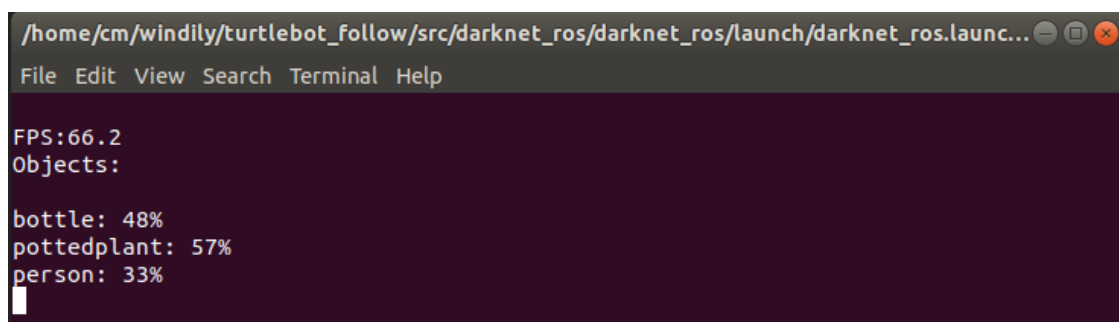
上。查看下位机视频话题的发布频率，基本稳定在 20Hz 以上，也排除下位机本身的问题。考虑到分布式 ROS 基于 SSH 连接，尝试 ping 下位机的 ip，情况如 4-4 所示：

```
cm@cm-X10DA1:~$ ping laptop
PING laptop (10.22.109.85) 56(84) bytes of data.
64 bytes from laptop (10.22.109.85): icmp_seq=1 ttl=64 time=70.6 ms
64 bytes from laptop (10.22.109.85): icmp_seq=2 ttl=64 time=222 ms
64 bytes from laptop (10.22.109.85): icmp_seq=3 ttl=64 time=113 ms
64 bytes from laptop (10.22.109.85): icmp_seq=4 ttl=64 time=21.9 ms
64 bytes from laptop (10.22.109.85): icmp_seq=5 ttl=64 time=205 ms
64 bytes from laptop (10.22.109.85): icmp_seq=6 ttl=64 time=162 ms
64 bytes from laptop (10.22.109.85): icmp_seq=7 ttl=64 time=170 ms
64 bytes from laptop (10.22.109.85): icmp_seq=8 ttl=64 time=515 ms
```

图 4-4 上位机连接下位机的延迟

延迟的平均值在 300ms 左右，比较不稳定。与此同时，每张 qhd 图像以 image\_message 的形式传输需要一定时间，估计在 1s 左右，有理由认为是网络延迟和带宽限制的问题。查看 darknet\_ros 节点，视频的接收频率在 2Hz 左右，网络延迟得问题得到进一步得佐证。

针对上述问题，采用压缩图像的方式进行改进，并且为了进一步减小传输消息的大小，去掉 image\_message 的消息头，直接传输图像像素。



```
/home/cm/windily/turtlebot_follow/src/darknet_ros/darknet_ros/launch/darknet_ros.launch...
File Edit View Search Terminal Help

FPS:66.2
Objects:

bottle: 48%
pottedplant: 57%
person: 33%
```

图 4-5 识别实时帧数

图像识别的 FPS 得到了进一步的提升，达到了 65 帧左右，延迟大概在 1s 左右，如图 4-5 所示，视频较之前有着可观的流畅度。

改进了图像识别的延迟，TurtleBot 依旧出现摇摆不定的情况。考虑到人工势场法是基于局部路径规划，TurtleBot 每行走一步均需要重新规划路径。由于延时的存在，路径变得并不稳定，于是出现摇摆不定的情况。针对上述问题，目前尚未找到比较好的处理方法。一个折中的方法是延长每一次进行移动的判断时间，即一次移动比较远的距离。这样会导致跟随机器人的实时性得到进一步的降低，在视觉感知上达不到“亦步亦趋”的效果。





图 4-6 跟随人室内实验场景

室内实验场景选择实验室如图 4-6 所示，有桌子，椅子，垃圾桶等相对复杂的环境。实验目标是让机器人与目标的直线距离在 120cm 左右。为了测试跟随距离的精度，在 TurtleBot 机器人移动稳定后将其关闭，测量它和被跟随人的距离，多次测量评估跟随的性能。

表 4-1 跟随误差

目标距离 kinect 测量值(cm)	目标距离测量值(cm)	绝对误差百分比(%)
115	120	4.2
115	119	3.4
117	120	2.5
123	125	1.6
116	120	3.3

通过对表 4-1 分析可知跟随距离的绝对误差百分比在 5%以内，跟随距离更倾向和目标保持大于 120cm 的距离，这个误差和距离在可接受的范围内。由于在控制策略中设置了大约 10cm 的阈值，以保证机器人在待跟随目标不动的情况下保持稳定，可以适当调节阈值降低绝对误差。

接下来测试移动机器人开始移动所需时间。由于控制策略优先调整移动机器人和待跟随目标的方向，在进行移动前需要转动姿态至一个合适的角度，这就是开始移动所需时间。



图 4-7 移动机器人调整跟随方向

从图 4-7 可以看出，移动机器人调整方向有一个原地左转和右转的情况，那是因为设定的单位转向角度为 0.3 个单位，有过转向的情况产生。如果调低单位转向角度，会导致移动机器人转向迟缓。通过测试随机角度开始移动所需时间的平均值，确定转向的单位角度。(NAN 表示时间大于 3 分钟)

表 4-2 移动所需平均时间

转向单位角度(rad/s)	开始移动所需平均时间(s)
0.1	NAN
0.2	7
0.3	3
0.4	35
0.5	65

通过表 4-2 可知，随着转向的单位角度增加，开始移动所需的平均时间剧烈增加。在转向角度为 0.1 时，移动机器人几乎不动。因此，在可接受的范围内选择 0.3 作为转向的单位角度。

#### 4.2.2 测试结果与分析

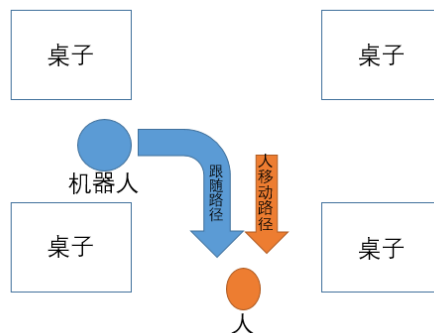


图 4-8 实验场景示意图

在本文中，主要目标是基于深度视觉的室内移动机器人跟随系统设计，因此有必要建立不同深度的实验场景。实验场景如图 4-8 所示，实际上，不同于 2D 平面内的深度，场景中除了桌子椅子之外，还有垃圾桶，电脑等 3D 空间干扰物。人的实际移动路径沿蓝色箭头的起点到终点，跟随机器人的实际移动路径沿红色箭头的起点到终点。为了更清晰直观的反映实验效果，截取关键帧展示如下：



图 4-9 机器人跟随场景

从图 4-9（a）可知，跟随机器人对移动中的人依然迅速的识别了出来，正在尝试寻找一条最佳的移动角度。图 4-9（b）显示跟随机器人已经找到开始移动角度，启动跟随。图 4-9（c）是跟随机器人正在跟随人，在跟随过程中途停下一次判断下一次跟随最佳移动角度。图 4-9（d）显示，人已经停止移动，跟随机器人向前移动，在保持设定的距离后停止移动，等待下一次跟随。

为了更全面的评估跟随效果，进行了 15 次跟随实验，记录完整跟随的成功次数。如图 4-10 所示：

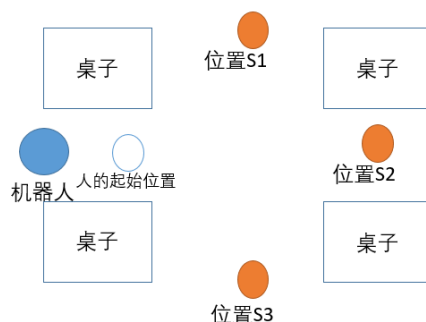


图 4-10 跟随成功率实验

表 4-3 跟随成功率

最终位置	跟随成功次数/跟随 总次数	平均用时	跟随成功率
S1	13/15	39s	86.7%
S2	15/15	25s	100%
S3	13/15	45s	86.7%

从表 4-3 可知，对于直线跟随机器人有着较高的成功率。最终位置 S1，S3 在 15 次跟随跟随中均有 2 次未成功跟随，其主要原因可能是行人行走过快，导致超出了跟随机器人的可视范围。最终位置 S1,S3 的平均用时较长，最终位置 S2 的平均用时较短，主要耗时反映在转弯判断这一步骤当中。

在跟随准确性上，TurtleBot 能比较准确的跟随人，几乎不会出现跟错其他物体的情形，对于半身图像依然保持良好的识别效果。对于不同的障碍物的深度信息，能做到比较准确的判断和待跟随目标的关系。在实时性上，TurtleBot 机器人在可接受范围内能即时反应人体移动的变化趋势，在随后的跟随周期中完成跟随。在整体效果上，能实现基础的跟随人功能和避障功能，实验结果良好。

### 4.3 本章小结

本章首先介绍了改进人工势场法的基本原理，通过仿真绘制机器人在有障碍物的 2D 平面图的移动路径验证了代码的有效性。其次介绍了跟随机器人的整体架构，进行实验后针对识别速度，延迟，跟随策略等做出了相应的改进。最后展示了最终的实验效果，完成了基于深度视觉的室内移动机器人跟随系统设计的基础功能。

## 第五章 总结与展望

### 5.1 总结

本文设计了一种基于深度视觉的室内移动机器人跟随系统。整个系统分为图像处理，目标识别，移动控制三大部分。上位机主要的主要任务是完成目标识别，下位机的主要任务是负责采集深度信息和图像信息并控制机器人的移动，协调 TurtleBot 机器人的动作，从而实现机器人对移动目标的跟随。本文主要研究成果如下：

（1）根据移动机器人，3D 相机，目标检测，跟随策略的国内外最新研究进展，分析了相关技术的重点和难点，提出了基于深度视觉的室内移动机器人跟随系统的整体设计。

（2）搭建了跟随机器人整体的硬件软件框架。针对利用 kinect2 获取深度信息有丢失和易受环境影响的情况，采用绘制轮廓的方式降低对深度信息采集的影响。针对上位机和下位机的网络延迟，提出将图片分辨率降低和压缩图片的方式提升传输效率。针对 TurtleBot 跟随目标时摇摆不定，采取延长移动判断周期的方式增强稳定性。

（3）基于 ROS 实现了从获取深度数据，处理深度图和彩色图，用 YOLOv3 进行目标识别，到机器人跟随目标的功能，设计了相关实验验证了代码的可行性。

### 5.2 展望

移动机器人的发展日新月异，目标识别算法也是常看常新。本文实现了基于深度视觉的室内移动机器人跟随系统，但是由于个人能力和时间有限，未能将该系统进行跟深入的调试和设计。目前来看各个模块仍有较大的提升空间，后续的研究工作有待深入：

（1）本文采用的 freenect2 作为 kinect2 摄像头的驱动，虽然对 linux 平台的兼容性较好，但是在某些细节方面仍和微软对 windows 平台的支持有一定差距。因此，对 freenect2 的研究有助于进一步提升硬件的性能，实现更好的机器人跟随。

（2）移动跟随机器人对实时性的要求较高，但由于网络延迟，硬件条件等限制，移动机器人机动性能受到局限，跟随效果较不理想。鉴于目前单片机性能的极大提升，将下位机替换为单片机有助于在满足实时性要求的同时，提供上位机所要求的运算速度。

（3）在移动跟随机器人的智能化方面，本文采用的控制策略是 2D 平面上的控制。可以参考大疆无人机对 3D 空间的控制策略，实现更鲁棒和智能化的控制方案。

## 参考文献

- [1] 王田苗, 陈殿生, 陶永,等. 改变世界的智能机器——智能机器人发展思考[J]. 科技导报,2015,33(21): 16-22.
- [2] 任福继, 孙晓. 智能机器人的现状及发展[J]. 科技导报, 2015, 33(21): 32-38.
- [3] 孟明辉, 周传德, 陈礼彬,等.工业机器人的研发及应用综述[J]. 上海交通大学学报, 2016, 50: 98-101.
- [4] 李瑞峰, 李汉舟. 地面可移动服务机器人发展现状[J]. 导航与控制, 2017, 16(01): 104-112.
- [5] Chestnutt J. Footstep Planning for the Honda ASIMO humanoid[C]/IEEE International Conference on Robotics and Automation, 2005, 25: 629-634.
- [6] 刘京运. 从Big Dog到Spot Mini波士顿动力四足机器人进化史[J]. 机器人产业, 2018,02(20): 105-119.
- [7] 中国科大“可佳”服务机器人获世界杯冠军[J]. 传感器世界, 2014, 20(08): 37.
- [8] 赵越. 小型足球机器人软件系统设计与轨迹学习[D]. 浙江: 浙江大学, 2015.
- [9] 侯志强, 韩崇昭. 视觉跟踪技术综述[J]. 自动化学报, 2006, 32(4): 603-617.
- [10] Xavier J, Pacheco M, Castro D, et al. Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver[C]/IEEE International Conference on Robotics and Automation, 2006: 3930-3935.
- [11] Kondaxakis P, Baltzakis H, Trahanias P. Learning moving objects in a multi-target tracking scenario for mobile robots that use laser range measurements[J]. 2009: 1667-1672.
- [12] Zhang Z. A Flexible New Technique for Camera Calibration[C]// IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000: 1330-1334. indoor mapping applications [J]. Sensors, 2012, 12(2): 1437.
- [13] Girshick R, Donahue J, Darrell T. et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]/ Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Columbus, USA, 2014: 580-587.
- [14] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]. neural information processing systems, 2015: 91-99.
- [15] 张顺, 龚怡宏, 王进军. 深度卷积神经网络的发展及其在计算机视觉领域的应用[J/OL]. 计算机学报, 2017: 1-29.
- [16] 谢洪明,陈亮,杨英楠. 如何认识人工智能的伦理冲突?——研究回顾与展望[J]. 外国经济与管理,2019,41(10):109-124.
- [17] 赵宝军. 人工智能对意识形态的操控风险及其化解[J]. 江汉论坛,2020(02):11-16.
- [18] Wiedemeyer T. “IAI Kinect2”. URL:[https://github.com/code-iai/iai\\_kinect2](https://github.com/code-iai/iai_kinect2), 2015.
- [19] Khoshelham K. ACCURACY ANALYSIS OF KINECT DEPTH DATA[J]. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2012, 38(1): 133-138.
- [20] Park J, Kim H, Tai Y, et al. High quality depth map upsampling for 3D-TOF cameras[C]. international conference on computer vision, 2011: 1623-1630.
- [21] 刘国华,段建春. 基于Kinect相机的深度图像空洞修复及超像素分割算法[J]. 计算机工程与科学,2020,42(05):851-858.
- [22] 林玲,陈姚节,郭同欢. 基于时空域数据融合的Kinect深度图像修复算法[J]. 科学技术与工程,2019,19(30):215-220.
- [23] Redmon J, Divvala S K, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. computer vision and pattern recognition, 2016: 779-788.
- [24] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]. computer vision and pattern recognition, 2017: 6517-6525.

- [25] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv: Computer Vision and Pattern Recognition, 2018.
- [26] Bjelonic M. "YOLO ROS: Real-Time Object Detection for ROS",URL: [https://github.com/leggedrobotics/darknet\\_ros](https://github.com/leggedrobotics/darknet_ros), 2018.
- [27] 张亚平,吴绩伟,马占刚,曹喜信,郭文海. 基于YOLOv3的神经网络模型压缩与实现[J]. 微纳电子与智能制造,2020,2(01):79-84.
- [28] 郭鸣宇,刘实. YOLOv3图像识别跟踪算法的优化与实现[J]. 电子测试,2019(15):65-66.
- [29] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots[J]. The International Journal of Robotics Research, 1986, 5(1): 90-98.
- [30] 于振中, 闫继宏, 赵杰,等. 改进人工势场法的移动机器人路径规划[J]. 哈尔滨工业大学学报, 2011, 439(01): 50-55.
- [31] 汤迪. 基于改进的人工势场法和水流法的移动机器人三维路径规划[D].中国矿业大学,2019:15-25.
- [32] 陈田田. 基于改进人工势场法的室内移动机器人路径规划研究[D].郑州大学,2019:25-31.

## 致谢

值此论文完成之际，我要衷心感谢我的导师冯李航。冯老师为人和善儒雅，对待学术态度严谨。如果没有他一针见血地指出我的思维误区，如果没有他不厌其烦地为我答疑解惑，我是不可能完成这一篇毕业论文的。感谢冯老师对我学习上的敦促，生活上的关心，让我在学业生活上都有长足的进步

在本课题的研究过程中，我必须要感谢实验室的研究生学长陈修翔对我的指导和关怀。他刻苦钻研不畏艰难的品格深深的激励着我，是我的榜样。除此之外，感谢我的同学李贤杰，邵建坪，张琪杰，李宗昊等同学对我生活上的照顾，并积极和我交流学习心得，有了他们的陪伴才显得学习之路丰富多彩。

最后，我还要感谢我的父母家人，我的学校南京工业大学，是他们坚定的支持和无私的关爱和培养才造就了我。我将永远铭记他们对我的帮助，用自己的努力和成绩回报他们！