

分类号 O159  
U D C \_\_\_\_\_

密 级 \_\_\_\_\_  
编 号 10486

# 武汉大学

## 硕士专业学位论文

# 基于三维视觉的自动工业焊接机器人系 统

研 究 生 姓 名: 彭锐

学 号: 2018282120133

指导教师姓名、职称: 杨文教授, David Navarro-Alarcon 助理教授

专业类别(领域): 电子与通信工程

二〇二〇年五月

# 3D Vision-based Automatic Industrial Welding Robot System

Candidate: PENG RUI

Student Number: 2018282120133

Supervisor: PROF. YANG WEN, DR. DAVID NAVARRO-ALARCON

Major: Electronics and Communication Engineering



School of Electronics Information  
WUHAN UNIVERSITY

May, 2020

## 论 文 原 创 性 声 明

本人郑重声明: 所呈交的学位论文, 是本人在导师指导下, 独立进行研究工作所取得的研究成果. 除文中已经标明引用的内容外, 本论文不包含任何其他个人或集体已经发表或撰写过的研究成果. 对本文的研究做出贡献的个人和集体, 均已在文中以明确方式标明. 本声明的法律结果由本人承担.

学位论文作者 (签名):

年       月       日



## 摘 要

近年来，工业机器人在世界各地的许多工厂得到了广泛的应用，以示教再现法为主的机器人焊接领域。为了追求高效率，本文提出了一种基于点的机器人电弧焊作业方法，由 RGB-D 相机捕捉到的云。该方法包括两部分：焊接坡口检测和三维焊接轨迹一代人。实际焊接场景可以显示在三维点云格式。聚焦于几何特征对于焊接坡口，该检测算法能够适应不同焊接工件的 V 型焊接坡口。同时，三维焊接轨迹包括机器人焊接坡口的六自由度姿态产生运动。在弹道上有可接受的误差一代，机械手可以驱动焊接跟踪焊枪轨迹并执行焊接任务。本文详细介绍了提出的集成机器人系统。实验结果表明了提出的方法的应用可行性。

**关键词：**工业机器人，三维视觉，运动规划，虚拟现实

## ABSTRACT

Nowadays, industrial robotic manipulators are used extensively in many factories around the world, with the teach-playback method dominating the robotic welding field. In order to pursue high-efficiency robotic arc welding tasks, we propose a method based on point cloud captured by an RGB-D camera. The method consists of two parts: welding groove detection and 3D welding trajectory generation. The actual welding scene could be displayed in 3D point cloud format. Focusing on the geometric feature of the welding groove, the detection algorithm is capable of adapting well to different welding workpieces with a V-type welding groove. Meanwhile, a 3D welding trajectory involving 6-DOF poses of the welding groove for robotic manipulator motion is generated. With an acceptable error in trajectory generation, the robotic manipulator could drive the welding torch to follow the trajectory and execute welding tasks. Details of the integrated robotic system are presented in this paper. Experimental results show application feasibility of the proposed method.

**Key words:** Industrial Robot, 3D Vision, Motion Planning, VR

# 目 录

<b>摘要</b>	<b>I</b>
<b>ABSTRACT</b>	<b>II</b>
<b>1 引言</b>	<b>1</b>
1.1 焊接技术 . . . . .	1
1.2 焊接工业机器人 . . . . .	2
1.3 计算机视觉技术赋能焊接机器人 . . . . .	3
1.4 本工作难点和贡献 . . . . .	7
<b>2 系统软硬件设计</b>	<b>9</b>
2.1 三维相机 Kinect-V2 . . . . .	10
2.2 RGB-D 相机 Realsense-D415 . . . . .	11
2.3 工业机械臂 UR-5 . . . . .	12
2.4 计算设备 . . . . .	12
2.5 软件系统框架 . . . . .	13
<b>3 UR5 机械臂动力学</b>	<b>15</b>
3.1 UR5 正运动学原理 . . . . .	15
3.2 UR5 机器人反运动学原理 . . . . .	17
<b>4 相机-机械臂标定</b>	<b>19</b>
4.1 相机-机械臂标定通用方法 . . . . .	19
4.2 eye-in-hand 手眼标定法 . . . . .	20
4.3 eye-to-hand 手眼标定法 . . . . .	21
<b>5 焊接缝定位及机械臂轨迹规划</b>	<b>23</b>
5.1 焊接缝定位 . . . . .	23
5.1.1 焊接工件表面模型获取 . . . . .	24
5.1.2 点云表面重构和滤波 . . . . .	25
5.1.3 点云表面法向量计算 . . . . .	26

5.1.4 法向量方向统一 . . . . .	28
5.1.5 点云梯度描述子 . . . . .	29
5.1.6 焊接缝区域分割 . . . . .	31
5.2 机械臂轨迹运动规划 . . . . .	31
5.2.1 焊接缝区域边缘提取 . . . . .	31
5.2.2 焊接轨迹点生成 . . . . .	33
5.2.3 焊接轨迹后处理 . . . . .	35
5.2.4 焊接轨迹点的旋转部分 . . . . .	38
5.2.5 六自由度轨迹插值 . . . . .	42
<b>6 机器人焊接实验部分</b>	<b>45</b>
6.1 机器人系统运行流程 . . . . .	45
6.2 传感器精度测试 . . . . .	46
6.2.1 Kinect-V2 测量精度测试 . . . . .	46
6.2.2 RealsenseD415 深度测量精度测试 . . . . .	47
6.3 实际焊接工件实验 . . . . .	48
6.3.1 圆柱体型焊接工件测试 . . . . .	48
6.3.2 长方体型焊接工件测试 . . . . .	49
6.3.3 Y-型焊接工件测试 . . . . .	50
<b>7 总结与展望</b>	<b>52</b>
7.1 总结 . . . . .	52
7.2 展望 . . . . .	52
<b>参考文献</b>	<b>53</b>
<b>致谢</b>	<b>57</b>

# 1 引言

## 1.1 焊接技术

焊接在生产、制造、建筑等现代工业里是不可或缺的一项技术，它是将两种或两种以上同种或异种材料通过原子或分子之间的结合和扩散连接成一体的工业过程。焊接又是一种制造或雕塑工艺，通过使用高热将材料（通常是金属或热塑性塑料）熔化在一起，并让它们冷却，从而导致熔合。焊接与低温金属连接技术（如钎焊和钎焊）不同，后者不会熔化母材。除了熔化母材外，通常向接头中添加填充材料，以形成熔融材料池（熔池），该熔池冷却后形成接头，该接头基于焊接配置（对接、全熔透、圆角等）可比母材（母材）更强。压力也可与热量结合使用，或单独使用，以产生焊缝。焊接还需要一种保护层，以防止填充金属或熔化金属受到污染或氧化。

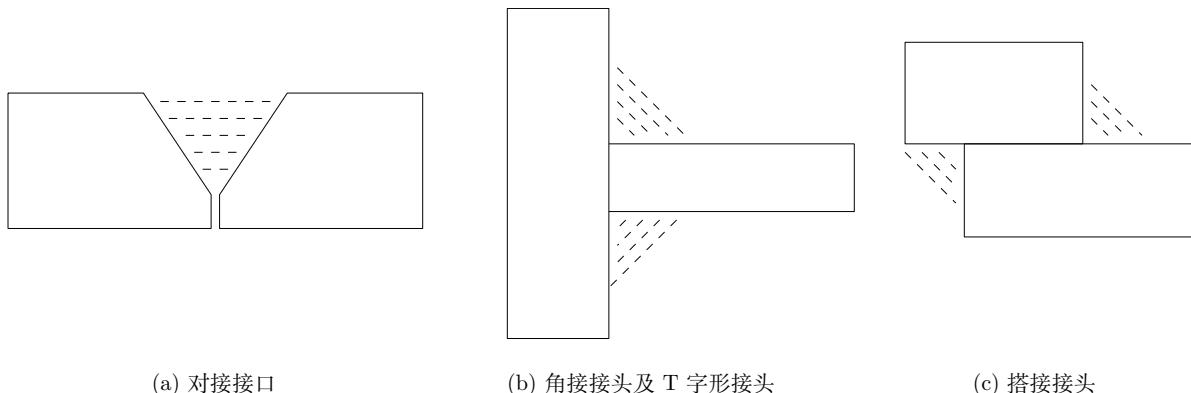


图 1.1: 常见的焊接接口种类

本工作所使用的焊接方法为电弧焊，该方法是指以电弧作为热源，利用空气放电的物理现象，将电能转换为焊接所需的热能和机械能，从而达到连接金属的目的。主要方法有焊条电弧焊、埋弧焊、气体保护焊等，它是应用最广泛、最重要的熔焊方法，占焊接生产总量的 60% 以上。焊条电弧焊是工业生产中应用最广泛的焊接方法，它的原理是利用电弧放电（俗称电弧燃烧）所产生的热量将焊条与工件互相熔化并在冷凝后形成焊缝，从而获得牢固接头的焊接过程。焊接时形成的连接两个被连接体的接缝称为焊缝。焊缝的两侧在焊接时会受到焊接热作用，而发生组织和性能变化，这一区域被称为热影响区。焊接时因工件材料焊接材料、焊接电流等不同，焊后在焊缝和热影响区可能产生过热、脆化、淬硬或软化现象，也使焊件性能下降，恶化焊接性。这就需要调整焊接条件，焊前对焊件接口处预热、焊时保温和焊后热处理可以改善焊件的焊接质量。常见的焊接接口又可以分为对接接头、角接接头及 T 字形接头、搭接接头，这三类接口都将在本工作的实验

部分出现，如图1.1所示。

大部分实际生产工厂中仍然是以人工为主的焊接形式，而且焊接的能量来源有很多种，包括气体焰、电弧、激光、电子束、摩擦和超声波等，除了在工厂中使用外，焊接还可以在多种环境下进行，如野外、水下和太空。无论在何处，焊接都可能给操作者带来危险，所以在进行焊接时必须采取适当的防护措施。焊接给人体可能造成的伤害包括烧伤、触电、视力损害、吸入有毒气体、紫外线照射过度等。

## 1.2 焊接工业机器人

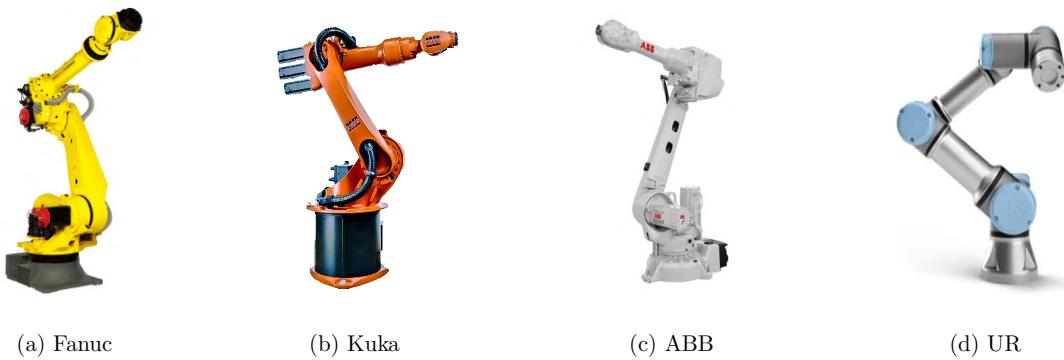


图 1.2: 工业机器人

焊接机器人是从事焊接（包括切割与喷涂）的工业机器人。根据国际标准化组织（ISO）工业机器人属于标准焊接机器人的定义，工业机器人是一种多用途的、可重复编程的自动控制操作机（Manipulator），具有三个或更多可编程的轴，用于工业自动化领域。为了适应不同的用途，机器人最后一个轴的机械接口，通常是一个连接法兰，可接装不同工具或称末端执行器。焊接机器人就是在工业机器人的末轴法兰装接焊钳或焊（割）枪的，使之能进行焊接，切割或热喷涂。市面上常见的工业机器人如图1.2所示。

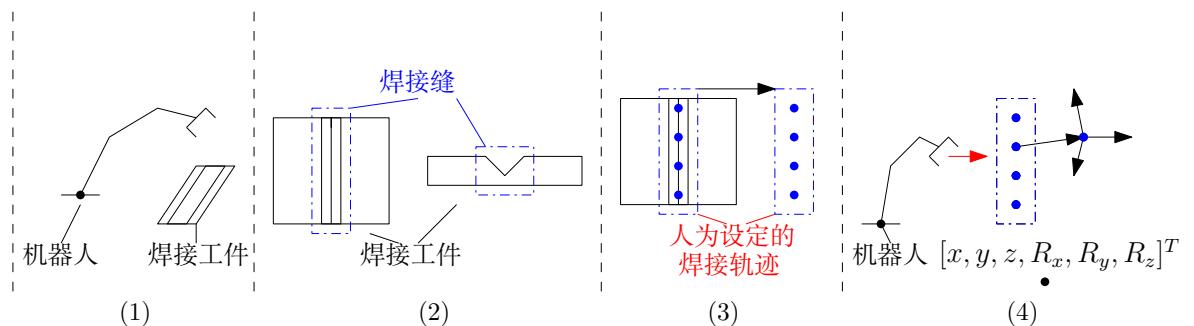


图 1.3: 示教再现法

对于人工焊接的局限性，工业机器人在焊接领域正在被大量投入使用，机器人技术的自动化工作特性既能提高焊接效率，同时又能替代大量焊接工人，从而大大降低工业生产中人为事故的

发生率。目前，工业机器人在世界各地的许多工厂中得到了广泛的应用，其中示教再现方法占据了机器人焊接领域的主导地位。更具体地说，为了操纵机器人手臂执行焊接任务，操作员需要提前设置每个路径点精确的三维位置和三维旋转，如图1.3所示。这种传统机器人焊接的实际情况大致如图1.4所示。



图 1.4: 机器人焊接情况

然而，尽管基于 CAD 的方法可以生成相当精确的焊接轨迹，但是它们仍然需要大量的人工干预，就像示教再现方法一样，而且焊接工件必须放在固定的位置。这些传统的机器人焊接应用已经无法应付工业界日益增长的焊接需求。为了解决这些问题，同时为了提高机器人的焊接效率，传感器由此被研究者用来辅助机器人自动定位焊接缝。当前相关的研究工作中用到的传感器大致包括图像传感器、深度传感器、超声波传感器、红外传感器和激光传感器。在实际的机器人焊接任务中，为了在复杂和不可预测的情况下提高焊接效率和质量，必须在两个工件之间生成精确的焊接轨迹，这就需要自动轨迹规划算法的应用。

### 1.3 计算机视觉技术赋能焊接机器人

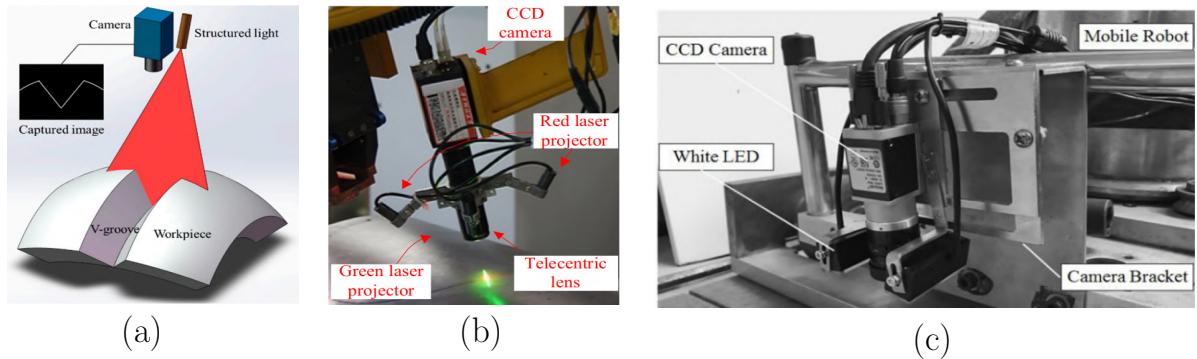


图 1.5: CCD 相机与机器人

生产车间对焊接工艺的要求越来越高，目前还是在很大程度上取决于经验丰富的焊接工人，使用各种算法和方法来获得准确有效的结果。对于工人来说，焊接这项技能需要多年的实际经验和培训才能成为合格的焊接工人，这也是一个非常危险的职业，但有很高的要求。目前，大多数焊接机器人的应用程序都是由工人通过示教方式编写的。在每一个新零件上编程路径和优化焊接参数需要一定的时间和费用。为了给焊接机器人带来效益，从低批量到中批量生产或维修工作都必须

加快手工焊接零件的速度。由于机器人焊接系统是刚性的，不能适应焊缝位置的变化，因此焊工可能不能满足焊接工艺的高要求。

计算机视觉-传感器系统在工业生产中有着广阔的应用前景，不仅在质量控制领域，而且在制造领域，特别是在提高机器人自动化的应用方面，通过视觉传感器提供的感知反馈，实现了计算机视觉与机器人控制运动的紧密联合。为实现机器人自主焊接，[1] 实现了焊接起始位置的识别与定位，该系统有粗到细两步方法，即求解曲线通过拟合建立以 SWP 初始值为中心的小窗口，实现焊缝与工件边界的功能。在这种方法中，使用相对大范围的电荷耦合器件（CCD）相机，而不需要额外的光，如图1.5所示。

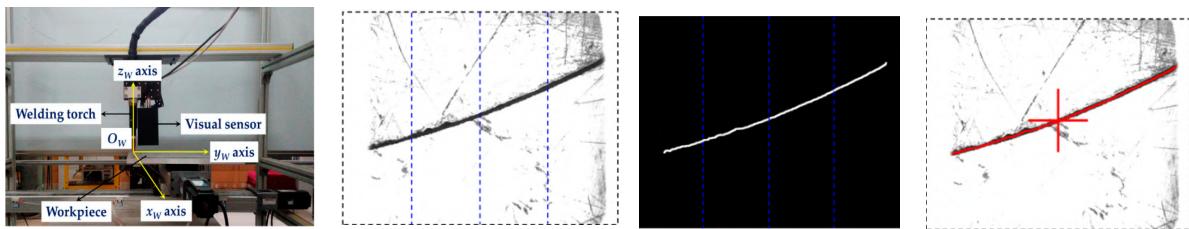


图 1.6: 彩色图像中提取焊缝的方法

基于先验知识的焊缝提取鲁棒算法显示了焊缝的特征以识别焊缝边缘并计算焊缝的焊接轨迹 [2]，该算法是一种从彩色图像中提取焊缝的方法1.6。[4] 提出了一种用于机器人离线编程焊接路径修正的立体视觉方法1.7，该系统将一个摄像头安装在机器人末端执行器上，从不同位置和方向获取焊接接头的图像，然后利用立体视觉算法和一种新的相机模型确定三维焊接接头的几何尺寸。

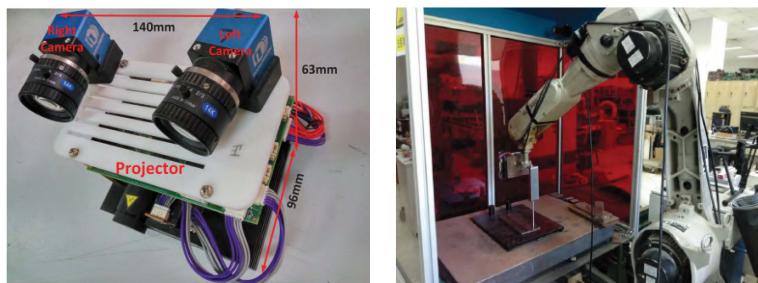


图 1.7: 立体视觉辅助工业机械臂

[5] 引入了基于非线性降维算法 isomap 的焊缝识别图像匹配，图像匹配方法分为模板构建、初始层选择、isomap 定位和下一层选择四个步骤，该研究的目的是修正离线程式化的标准机器人焊接轨迹。[6] 介绍了智能弧焊机器人的焊缝跟踪，电弧焊系统的自动化、智能化控制对保证焊接质量、提高生产率、改善劳动条件具有重要意义。在该系统中，提出了一种焊缝坡口的识别方法。图像处理包括抓取图像、平滑图像去除抓取过程中产生的噪声、生成直方图、从直方图中确定阈值、二值化图像、标记每个粒子、计算和读取图像的极值以及确定凹槽边缘位置等步骤。[7] 将基于轨迹的控制用于机器人激光焊接的实时焊缝跟踪1.8，基于轨迹控制结构的焊缝跟踪算法能够满足机器人激光焊接的精度要求，该控制方法可用于许多不同的操作；（1）利用事先掌握的几何知

识（例如，从 CAD 数据中获得的几何知识）设定焊缝轨迹；（2）感知焊缝轨迹，（3）实时跟踪焊缝轨迹，事先了解其几何结构；

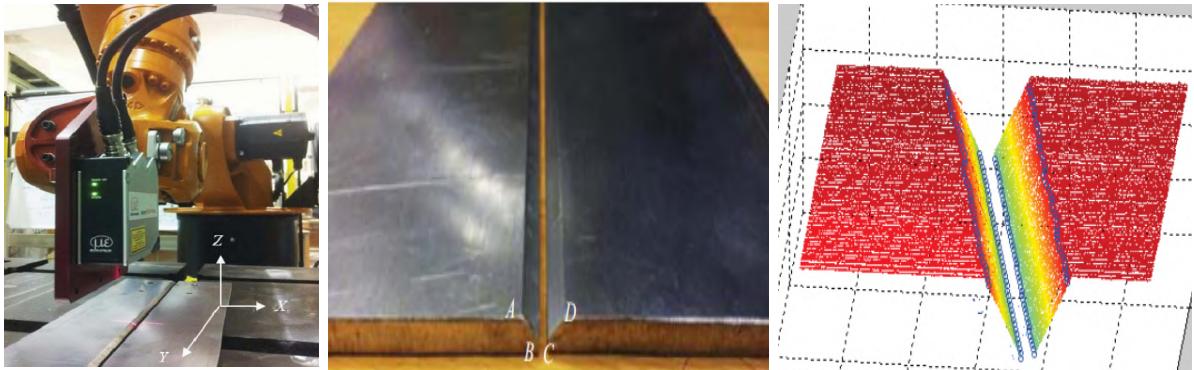


图 1.8: 基于激光扫描的机器人焊接

[8] 开发了基于焊接机器人图像处理技术的搭接接头起始点和终点识别方法，即用 canny 算子对图像进行平滑和边缘提取，T 连接检测，进而对焊缝的起始点和终点进行分析。通过机器人末端上的相机获取的焊缝和中心被提取以检测焊缝位置以及该点与中心轴之间的偏移量，采用自适应多步分割技术增加焊缝实际边缘的可能性，提高了直线拟合精度 [9]，其目的是减少环境条件的制约，提高焊缝跟踪精度。[10] 开发了一种利用自适应线增长算法来自动识别角焊缝的方法，该系统中引入了不考虑焊缝形状的焊缝鲁棒识别方法，大大降低了母材、表面光洁度和表面缺陷，如划痕、轧屑和铁锈等等因素的影响，获取图像时使用了手持式立体相机，安装在焊枪周围。[11] 是基于视觉的机器人弧焊焊缝实时跟踪研究，该工作使用了由 CCD（电荷耦合器件）相机构成的视觉传感器，固定在机器人的端部关节上，用来采集焊接工件的表面图像，焊缝识别算法有三种步骤是选择一个称为窗口的区域，计算焊枪的偏移量和机器人校正的时间精确计算。[12] 开发了一种有效的对接焊缝检测算法，该方法的基本思想是先在局部区域找到一对焊缝边，然后从两条边开始每个边的端点，从每个边的两个端点开始搜索轨迹点 1.9。

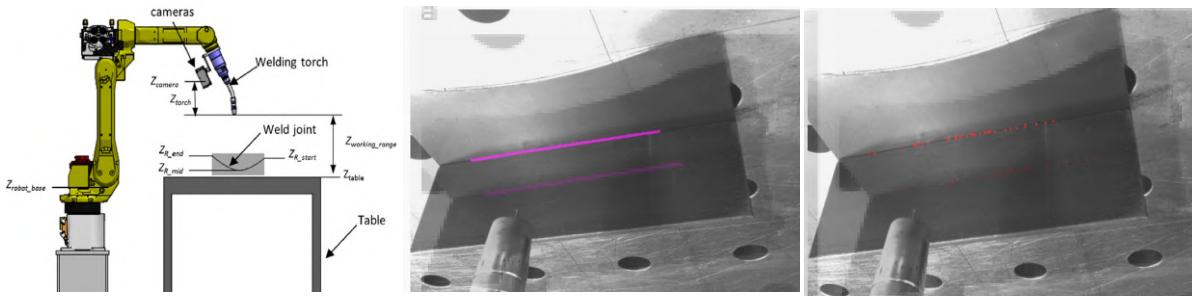


图 1.9: 视觉伺服结合工业机械臂

[13] 提出了一种基于智能空间概念，该系统使用基于两个摄像头的边缘检测算法，与其他图像处理方法包括极线立体匹配和三维预图像重建算法融合，来定位图像中的焊接区域并计算焊接

路径。由机器视觉系统辅助的自动焊接系统，用于计算工件尺寸、位置和的定位 [15]，该工作开发的图像处理算法有四种转换为二值图像的步骤、形态学操作、图像骨架和像素信息用于检测工件的尺寸、位置和方向。[14] 通过使用计算机视觉实现了机器人焊缝的自动识别和定位，将二维图像作为输入 [1.10]，该系统能够对黑色金属的窄焊缝进行稳健识别材料结合可靠的图像匹配和三角剖分。

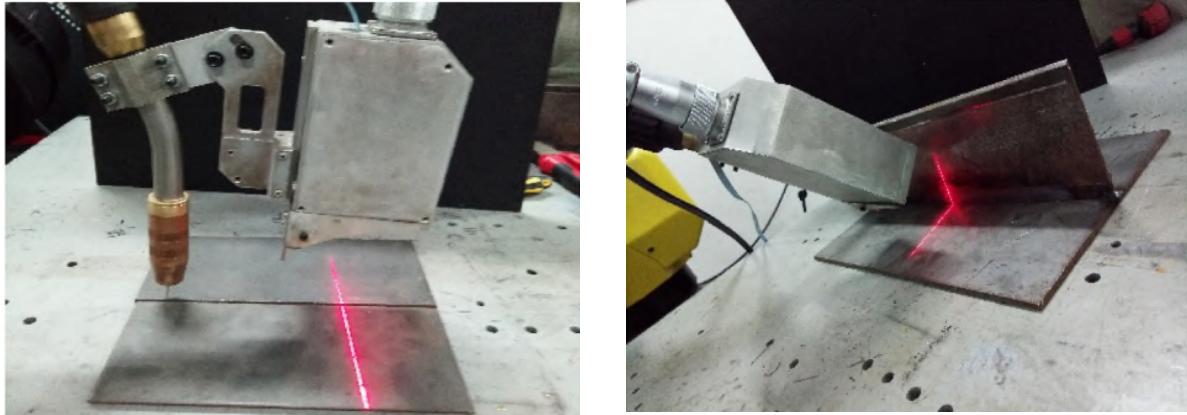


图 1.10: 基于 RGB 图像的焊接缝定位识别

[19] 开发了一个基于机器视觉的低成本焊缝跟踪系统，其中激光装置垂直于工件表面，机器人上的摄像头朝向以  $45^\circ$  的角度投射激光条纹。[16] 引入了一种在对接结构中检测焊缝的算法，实现低碳钢材料的机器人电弧焊，该算法使用 Hough 变换检测焊接件的外部边界，以便移除背景和提取图像中焊缝的位置。[17] 开发了一种新的基于特征点检测的焊缝跟踪算法便携式焊接机器人，该系统的目的是利用激光跟踪焊缝，实现焊接路径规划位移传感器，该算法包括四个步骤：(1) 扫描，(2) 滤波，(3) 生成参考点 (4) 路径规划。[20] 利用角点检测技术在全球范围内识别焊缝的初始位置环境，工件图像由安装在焊接机器人末端的 CCD 摄像机拍摄，然后使用图像处理算法从背景中分割图像，包括平滑、锐化和区域分割，再应用 Haris 算子检测焊缝初始位置。



图 1.11: 爬行机器人搭载摄像头辅助定位焊接缝

[18] 提出了一种用于电弧焊接过程焊缝跟踪的熔池图像质心算法，焊接熔池的图像是由安装在焊枪周围的摄像头采集的，图像处理采用技术手段分析了熔池及其周围环境的特征和焊缝的质心提取池图像作为焊缝跟踪的参数。[21] 采用了一种称为缺陷修正的新修正技术以 2 个视觉传感器为核心的缺陷自动检测，方法强调形状匹配特性，在该系统中，使用高斯平滑特征来确定比中

值滤波更好的图像结果过滤器，检测的缺陷提供了高度（z坐标）、长度（y坐标）和宽度信息（x坐标）。[3] 描述了一种基于双目视觉的焊缝检测算法，并将该算法搭载在爬行机器人上，该研究的目的是利用爬行机器人对大范围墙面上出现的焊缝进行图像采集并提取焊缝位置<sup>1.11</sup>，算法采用中值滤波和同态滤波相结合的方法对焊接图像进行去噪，降低了类平方误差，证明了该方法的有效性。

然而，二维成像处理算法依赖于在颜色信息上，不能处理环境亮度的剧烈变化，特别是在焊接领域。研究人员更愿意使用能够获取焊接环境空间立体信息的三维视觉传感器。此外，随着点云库（PCL）的发展 [22]，它是专门为三维点云处理而设计的，而且基于点云处理可以提取并定位焊接工件表面上的目标区域。一种 3D 传感器是实现焊缝重建的立体摄像机以及跟踪 [23]，从而实现良好的规划准确性。基于 RGB-D 语义分割和三维模型的最新算法比较 ShapeNet 数据集，一个无组织的基于点云的边证明了角点检测算法 [24] 的适用性机器人焊接。



图 1.12: *pointcloud*

随着一些 RGB-D 相机的出现微软 Kinect、英特尔 RealSense 和苹果等传感器 PrimeSense [25]，一些融合 RGB 图像和已经开发出深度信息 [26]。通过整合机器人系统中的 RGB-D 传感器 [27]，控制器能够适应动态焊接环境。锂等。[28] 提出了一种焊接坡口识别方法使用 RGB-D 相机，其中 RGB 图像用于检测焊接坡口并使用点云获得焊接坡口的三维位置。上述基于三维传感器的研究工作的主要问题之一是，实验结果缺乏对不同类型焊接的充分测试工件；而且，目标焊接缝大多位于平面内，相对简单。受 [29] 的启发，本文提出的方法主要针对点云焊接缝检测及三维运动轨迹规划，实验结果包括三个方面：运行时算法处理速度、焊缝检测精度和轨迹跟踪精度。系统在四种类型的 V 型槽工件的实验测试中证明了其具有一定的应用价值。

## 1.4 本工作难点和贡献

总体而言，自动焊接机器人研究现状：

- 目前焊接需要越来越大，传统的机器人焊接方法（操作员手把手设置轨迹）虽然也是实现了自动化，但还是有一定的局限性。
- 非常有必要在焊接机器人上引入新技术来提高焊接效率，很多这方面工作都是基于视觉，也有激光等等。

• 基于视觉的工作绝大部分都是简单的 rgb 图像处理，焊接工件（焊接物体）往往就是一块平板，这类方法只能处理平面或近似平面的情况，对于复杂的较大范围立体轨迹，完全失效。所以出现了用三维点云的方法来定位焊接缝，这类方法是最近才出现的研究方向，相关工作非常少，而且都没有完整提出从感知到运动规划再到实际机器人控制的一系列过程，其中大部分只是用来做三维焊接缝检测定位。实际应用和研究不足：

- 目前焊接领域需求很大，但是目前学术界的方法还比较初级，不够鲁棒，原因之一是在此领域进行研究的学者不多，还需要实地工厂以及专业焊接设备，而且此领域做计算机视觉的研究者也不多。
- 引入三维点云是一个比较新颖，而且事实上很合理的方法，因为实际的焊接工件不可能全部都是平面。
- 基于视觉焊接机器人方向的学术文献里，没有一篇有相关的系统演示视频，没有一篇开源了项目代码，极少的工作真正做了实际焊接（顶多仿真运动）。

本工作的主要贡献：

- 三维点云是一个合理而且有效的研究方向，以此为基石。提出了用表面点云描述焊接缝特征的方法（定位）。进一步用焊接缝点云生成符合实际的焊接三维轨迹（离散三维点，只有三维位置），还有后续一系列空间排序等操作。
- 利用焊接点云计算每个轨迹点的三自由度旋转部分，最后将完整的六自由度轨迹输出给机械臂，并在六自由度机械臂上实现焊接轨迹的跟踪运动。
- 加入其他传感器辅助感知，完善了整个传感器-机器人系统。
- 项目代码已经开源在<https://github.com/professor1996/welding-robot.git>，加入了详细的使用教程和注释，还有系统演示视频。

本工作在智能焊接机器人这个领域，以三维点云为基础，完整地做出了从感知到运动规划方法，具有非常高的原创性，而且已开源，每个焊接工程师都可以参考和使用本系统和教程，实现工业机械臂的自动焊接任务。

## 2 系统软硬件设计

本工作贴合中小型焊接工业生产模式，紧跟当前智能焊接机器人研究现状，设计了一套完整的新型视觉辅助的焊接机器人系统，该传感器-机械臂系统的实际开发环境如图2.1所示。传感器为两个三维相机，一个用来视觉引导机器人运动，另一个用来扫描焊接工件轮廓并采集表面点云数据，后续的焊接缝定位以及焊枪轨迹规划算法都是基于工件的三维点云模型来研发的。系统所用机械臂为优傲公司的UR-5，自带开发接口，易于开发人员编程控制。

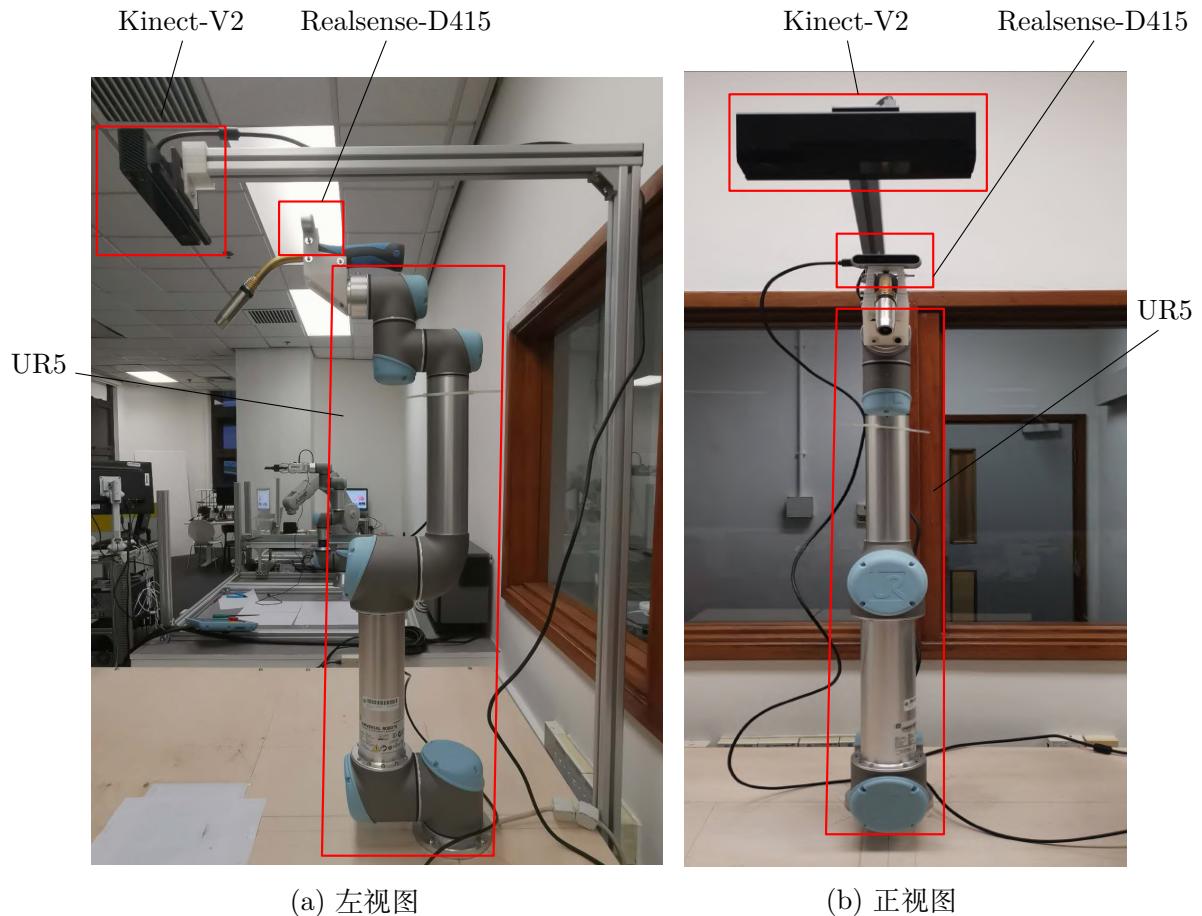


图 2.1: 机器人系统工作空间及布局

硬件部分包括两类三维相机(小节2.1, 2.2),一台协作式机械臂(小节2.3),一个焊接枪,以及两个自行设计并3D打印的传感器安装结构,计算设备为台式计算机(CPU: Intel-i9, GPU: Nvidia-RTX2060)。Kinect-V2负责粗略地定位工件位置,被固定在工作平台的正上方。Realsense-D415扫描工件表面轮廓并得到点云模型,其被安装在机械臂的末端执行器上,可随机械臂移动。

## 2.1 三维相机 Kinect-V2

Kinect-V2 三维相机 [30] 是美国公司 Microsoft 于 2014 年发行的一款体态感知设备如图2.2，该传感器不需要用通过任何第三方设备（如手柄）之类的介入便可实现感知目标姿态的功能，它一经面世便成为了立体视觉研究的热点。PrimeSense 公司开发设计了该传感器的主要数据处理芯片，该芯片是专门为高速计算与视觉算法设计而量身打造的。Kinect 相机内部装有四个扬声器，两个麦克风阵列，一个高速 Flash，一个 USB2.0 总线接口，三个摄像头 [31]。最左侧为红外发射器，在固定视角范围内向外发射红外激光，最右边则为一个红外接收器，用以接收周围环境中的红外激光散斑，进而采集场景的表面深度数据，再通过其内部转换处理算法可生成  $1920 \times 1024$  分辨率的深度图像。同时，Kinect 可发布时间流接近同步的彩色图像和深度图像，且两者分辨率可达到相同，利用简单的图像配准算法处理，可将彩色图像与深度图像对齐。



图 2.2: Kinect-V2 外观图和其感知信息处理流程

彩色摄像头用来获取 RGB 彩色图像，利用增量直方图算法对其计算增量直方图，再通过粒子滤波算法计算权值。另外两个摄像头构成了 3D 深度传感器 [32]，其中红外发射器发射激光，CMOS 摄像头通过对反射光的捕捉来获得一个深度场，从而采集深度数据，并将获取的深度信息转换为深度图像，最终得到被测目标的空间位置和轮廓形状，输出的深度图像以  $1920 \times 1024$  的分辨率发布。Kinect 深度探测的最佳范围为 1.2m~3.5m 之间。Kinect 相比普通摄像头的独特之处在于其使用 TOF(Time of Flying) 技术采集景深数据从而生成深度图像，深度图像的每一个像素点的像素数值代表此像素点在实际三维空间中的点与 Kinect 相机坐标系原点的欧式距离。

使用 Kinect 传感器时，对被测物体周围的环境要求很低，一般不会受到光照、阴影、物体遮挡等条件的影响。而之所以 Kinect 传感器会具有如此优秀鲁棒性的优势，与 Kinect 传感器的深度图成像原理是紧密相关的，主要是因为该传感器采集的是深度图像的像素灰度值仅与距离有关，因此，通过深度图像可以得到目标相对于传感器的三维坐标，而灰度值仅与视场 Z 方向有关。

## 2.2 RGB-D 相机 Realsense-D415

英特尔公司生产的 RealSense™ D415 [33] 拥有标准视野，十分适合近距离高精度感知，例如 3D 扫描。D415 在深度传感器上有一个滚动快门，每度提供了最高的深度质量。深度相机 Realsense-D415 将一个英特尔 RealSense 模块 D415 和英特尔 RealSense Vision 处理器 D4 [34] 放入一个小型的集成器件中，如图2.3所示。



图 2.3: Realsense-D415 外观和视觉处理器

该三维传感器重量轻、功能强大、成本低，有一套定制的图像显示软件和标定软件，用途广泛，普遍被应用于移动机器人建图与定位、三维物体测量与目标检测跟踪等研究。D415 拥有以下功能特性：

- 功能使用环境：室内/室外图像传感器技术：卷帘快门， $1.4 \text{ m} \times 1.4 \text{ m}$  像素大小最大范围：大约 10 米。精度取决于校准、场景和照明条件。深度技术：有源红外立体声。
- 景深 (FOV):  $65^\circ \pm 2^\circ \times 40^\circ \pm 1^\circ \times 72^\circ \pm 2^\circ$  最小深度距离 (Min-Z): 0.16 米。
- 深度输出分辨率和帧速率：最高  $1280 \times 720$  有源立体声深度分辨率，高达 90fps。
- RGBRGB 传感器分辨率和帧速率： $1920 \times 1080$  英寸 RGB 帧速率：30 fpsRGB 传感器 FOV (H x V x D):  $69.4^\circ \times 42.5^\circ \times 77^\circ$  ( $\pm 3^\circ$ )

该相机的深度图效果如图2.4所示。

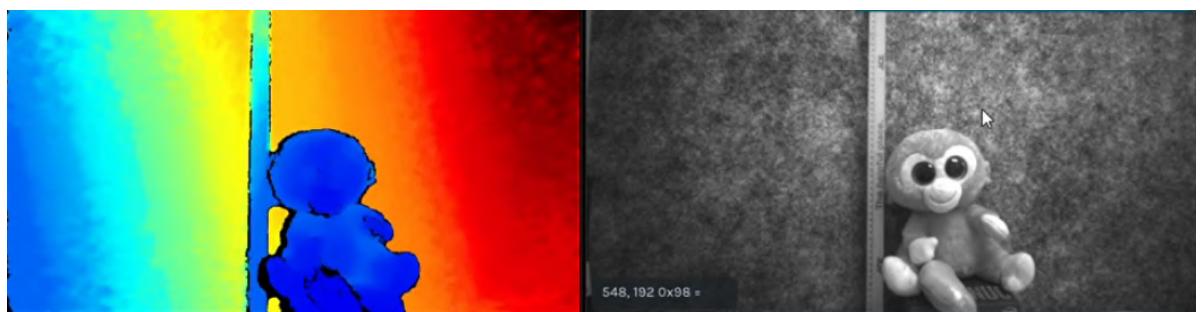


图 2.4: Realsense-D415 深度图效果示意

## 2.3 工业机械臂 UR-5

六自由度协作式 UR5 工业机械臂 [35] 具有轻巧和高度灵活运动特性，能够执行自动化和危险重复工业任务，有效载荷高达 5 公斤，如图2.5。UR5 机器人是优化、低重量协作过程（如拣选、放置和测试）的理想选择，该机械臂工作半径高达 33.5 英寸（850 毫米），有效工作范围大、易于编程、安装速度快、具有协作性和安全性，足以节约工人时间并为其他生产任务提高效率。



图 2.5: UR5 机器人模型和控制箱

UR5 机器人的主要性能：

- 重复精度  $\pm 0.1$  毫米/ $\pm 0.0039$  英寸（4 密耳），温度范围 0–50°，功率消耗最小 90W，通常 150W，最大 325W。
- 1.5 种先进的可调安全功能，TüV NORD 认可安全功能，已经通过测试：EN ISO 13849:2008 PLD。
- 规范：有效载荷 5 公斤/11 磅，最大工作范围 850 毫米/33.5 英寸，自由度 6 个旋转关节。
- 程式设计：Polyscope 图形化使用者介面，12 英寸触控式荧屏。
- 每个关节运动最大范围和最快速度：底座： $\pm 360^\circ$  和  $\pm 180^\circ/\text{秒}$ ，肩膊： $\pm 360^\circ$  和  $\pm 180^\circ/\text{秒}$ ，手肘： $\pm 360^\circ$  和  $\pm 180^\circ/\text{秒}$ ，手腕 1： $\pm 360^\circ$  和  $\pm 180^\circ/\text{秒}$ ，手腕 2： $\pm 360^\circ$  和  $\pm 180^\circ/\text{秒}$ ，手腕 3： $\pm 360^\circ$  和  $\pm 180^\circ/\text{秒}$ ，末端执行器（通常）：1 米/秒/39.4 英寸/秒。
- 特色：IP 等级 IP54, ISO 等級-无尘室-5，噪音低噪音，输入/输出电源工具 12V/24V, 600mA。
- 物理特性：空间占用 Ø149 毫米，材质铝，PP 塑料，工具接头种类 M8，机器人手臂缆线长度 6 米/236 英寸，含缆线重量 18.4 公斤/40.6 磅，该机器人能在 0–50°C 内工作，在关节持续高速运转时，环境温度应降低 [36]。

## 2.4 计算设备

系统所用的 CPU [37] 为英特尔® 酷睿™ i9-9900K 处理器2.6，基本参数如下：

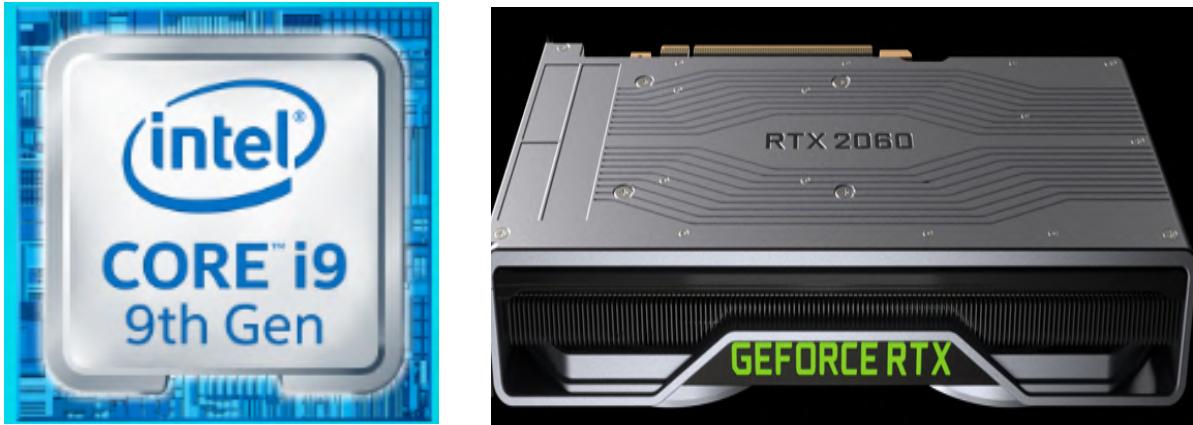


图 2.6: 系统计算设备的 CPU 和 GPU 型号

- 性能方面; 内核数: 8, 线程数: 16, 处理器基本频率: 3.60 GHz, 最大睿频频率: 5.00 GHz, 缓存: 16 MB Intel® Smart Cache, 总线速度: 8 GT/s, TDP: 95 功率消耗最小 90W, 通常 150W, 最大 325W。
- 内存规格; 最大内存大小: 128GB, 内存类型: DDR4-2666, 最大内存通道数: 2, 最大内存带宽: 41.6 GB/s。

图形显示卡 GPU [38] 的型号为 Nvidia-RTX2060, 主要性能如下:

- 引擎参数; 英伟达 CUDA 核数: 1920, RTX-OPS: 37T, 千兆速率: 5 Ray/s, 最高主频: 1680MHz, 基本主频: 1365 MHz。
- 内存参数; 存储速度: 14 Gbps, 标准存储配置: 6 GB GDDR6, 内存接口带宽: 192-bit, 存储带宽: 336 GB/s。

## 2.5 软件系统框架

系统软件部分在 LINUX (ubuntu16.04) 系统上开发, 以平台 ROS (机器人操作系统, 版本为 kinetic) 搭建软件系统框架, 其主要包括了基于点云处理的三维焊接缝定位及焊枪运动规划算法、UR5 机械臂 ROS 驱动、三维相机 ROS 驱动、传感器-机械臂标定算法和机械臂描述文件及运动学规划算法, 框架如图2.7所示。

在机器人系统运行时, 部分模块之间需要建立实时数据通信, ROS 提供了十分便利的 “Topic” 发布和订阅功能, 比如 Realsense-ROS 驱动 [39] 将 D415 相机的实时深度图数据发布出来, 再由模型三维重建算法订阅;Kinect-ROS 驱动将 Kinect-V2 的彩色图像数据发布出来, ARmarker 检测算法包订阅该数据, 再实时反馈输出标记相对于 Kinect 的三维空间姿态; 标定模型算法将目标的空间姿态发布出来, 再由自动标定控制程序订阅从而引导机械臂对其目标位置; 轨迹规划算法模块将计算出来的六自由度轨迹数据发布, 再由轨迹跟踪控制程序订阅, 用以控制机械臂末端执行连续不间断的轨迹运动。为了实现算法处理和机器人系统交互的可视化环境, ROS 中的 Rviz 三维可

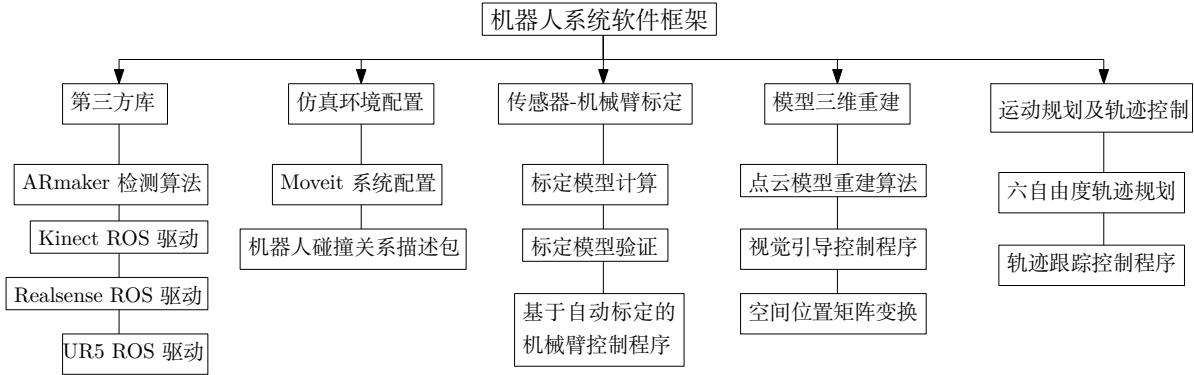


图 2.7: 软件框架

可视化功能恰恰满足了这点，而 UR5-ROS 驱动建立了实际机械臂运动姿态与仿真环境 (Rviz) [40] 里机械臂模型姿态的同步机制，即该驱动将实际机械臂每个关节的实时角度发布，由 Moveit 系统订阅并在 Rviz 中显示机械臂模型的姿态（与实际机械臂一致），如图2.8所示。

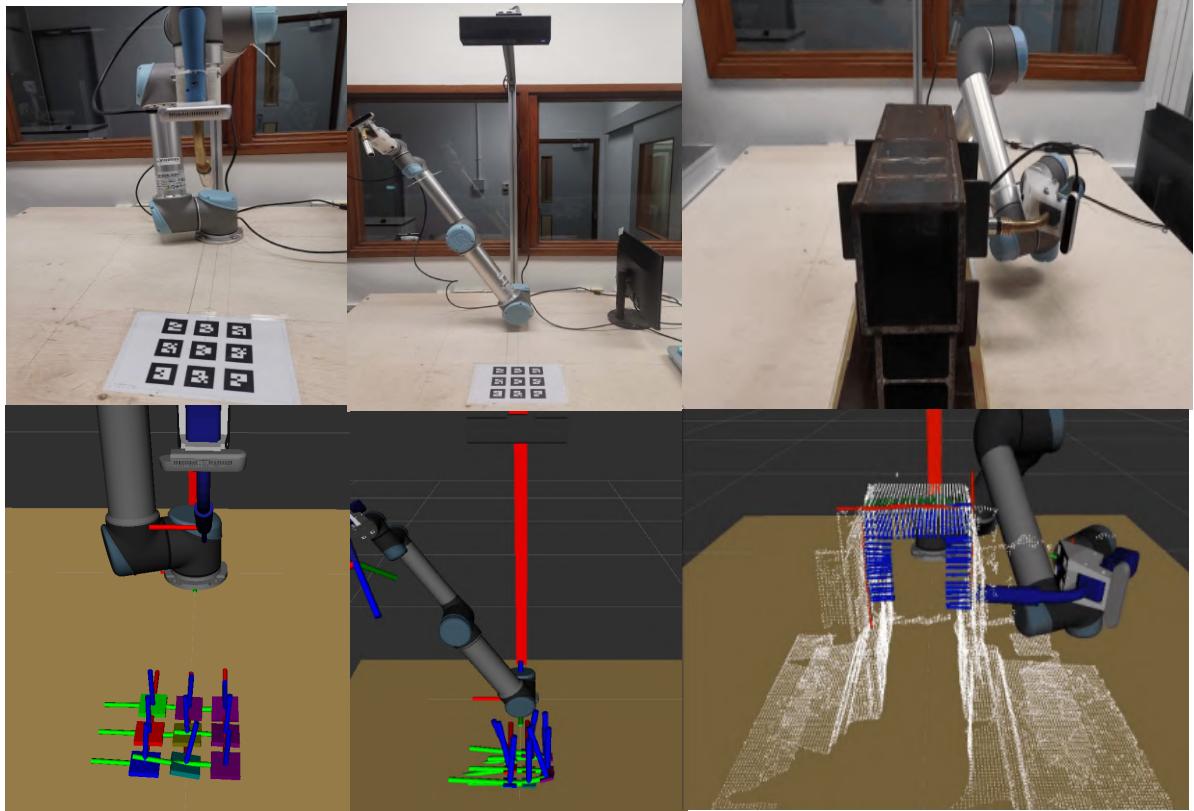


图 2.8: 实际机械臂与仿真环境的机械臂模型姿态同步

### 3 UR5 机械臂动力学

本章详细介绍了 UR5 机器人的正向和反向运动学原理，参考了文献 [41–43]。正运动学是已知机械臂各个关节的长度和旋转角度，来求解末端执行器的位姿。而反运动学是输入末端执行器的位姿，来求解机械臂每个关节的当前角度位置，这个过程还需要考虑与周围环境的碰撞关系等因素。在数学上根据 Denavit-Hartenberg (D-H) 符号首先定义符号表示法 [44]，下面所用的数学符号表示法为：

- ${}^0P_6 = \begin{bmatrix} {}^0P_{6x} \\ {}^0P_{6y} \\ {}^0P_{6z} \end{bmatrix}$  表示坐标系 6 的在坐标系 0 下的原点。
- ${}^0\hat{Y}_6 = \begin{bmatrix} {}^0\hat{Y}_{6x} \\ {}^0\hat{Y}_{6y} \\ {}^0\hat{Y}_{6z} \end{bmatrix}$  表示坐标系 6 的 Y 轴在坐标系 0 下的单位方向向量。
- ${}_6T$  表示从坐标系 6 到坐标系 0 的转换矩阵，意为:  ${}^0P = {}_6T \cdot {}^6P$ 。

#### 3.1 UR5 正运动学原理

UR5 正运动学公式是基于已知的关节角度  $\theta_{1-6}$  计算转换矩阵  ${}_6T$ 。该矩阵被定义为：

$$\begin{aligned} {}_6T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) &= \begin{bmatrix} {}^0R & {}^0P_6 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^0\hat{X}_6 & {}^0\hat{Y}_6 & {}^0\hat{Z}_6 & {}^0P_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^0\hat{X}_{6x} & {}^0\hat{Y}_{6x} & {}^0\hat{Z}_{6x} & {}^0P_{6x} \\ {}^0\hat{X}_{6y} & {}^0\hat{Y}_{6y} & {}^0\hat{Z}_{6y} & {}^0P_{6y} \\ {}^0\hat{X}_{6z} & {}^0\hat{Y}_{6z} & {}^0\hat{Z}_{6z} & {}^0P_{6z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3.1)$$

其中  ${}^0\hat{X}_6$ ,  ${}^0\hat{Y}_6$ ,  ${}^0\hat{Z}_6$  是相对于坐标系 0 的坐标系 6 的三轴单位向量，若将该变换矩阵拆成一条转换链：

$${}^0T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = {}^0T(\theta_1){}^1T(\theta_2){}^2T(\theta_3){}^3T(\theta_4){}^4T(\theta_5){}^5T(\theta_6) \quad (3.2)$$

UR5 机器人在初始状态下的动力学结构如图3.1所示。

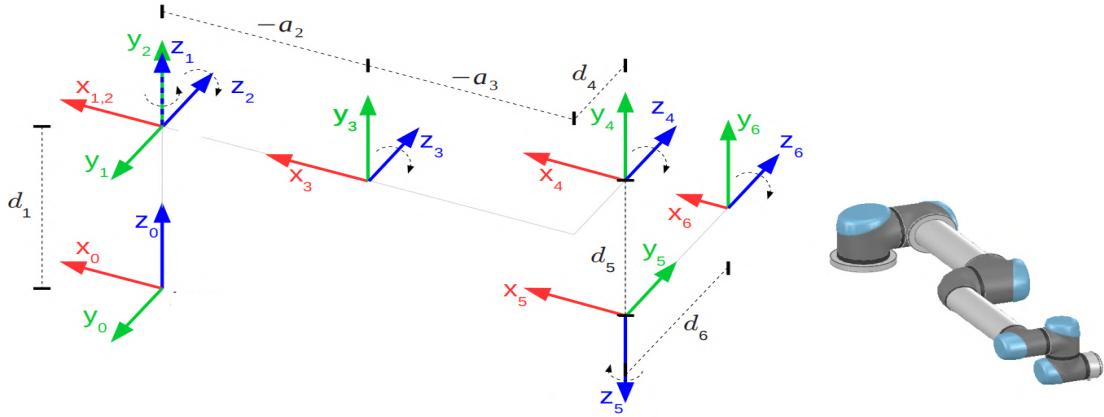


图 3.1: UR5 机器人在初始状态

根据 [34], DH 参数被定义为:

- $a_i$  = 沿着  $X_i$  从  $Z_i$  到  $Z_{i+1}$  的距离
- $\alpha_i$  = 关于  $X_i$  从  $Z_i$  到  $Z_{i+1}$  的角度
- $d_i$  = 沿着  $Z_i$  从  $X_{i-1}$  到  $X_i$  的距离
- $\theta_i$  = 关于  $Z_i$  从  $X_{i-1}$  到  $X_i$  的角度

对于此时初始状态的 UR5 机器人来说, DH 参数为:

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	$d_1$	$\theta_1$
2	$\alpha_1 = 90^\circ$	0	0	$\theta_2$
3	0	$a_2$	0	$\theta_3$
4	0	$a_3$	$d_4$	$\theta_4$
5	$\alpha_4 = 90^\circ$	0	$d_5$	$\theta_5$
6	$\alpha_5 = -90^\circ$	0	$d_6$	$\theta_6$

表 3.1: UR5 机器人的改进 Denavit-Hartenberg 参数 (DH 参数), 与图中 3.1 的坐标系相对应, 只有参数  $\theta_i$  是变量, 而其他的参数是恒量。

DH 参数可用于为任意两个关节定义变换矩阵, 则关节  $i-1$  和关节  $i$  之间的变换矩阵  ${}^{i-1}T_i$  被定义为:

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos(\alpha_{i-1}) & \cos \theta_i \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin \theta_i \sin(\alpha_{i-1}) & \cos \theta_i \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

由此通过在等式3.3中插入表3.1中的 DH 参数，很方便为 UR5 机器人的每两个关节写出变换矩阵。然后，通过将所有六个变换矩阵相乘，可求导出从基座坐标系到末端执行器坐标系的旋转位移矩阵变换，如等式3.2所示。其求解结果是变换矩阵  ${}^0T_6$  中所有十二个参数的解析表达式 [35]。

## 3.2 UR5 机器人反运动学原理

机械臂逆运动学 (Inverse Kinematics) [45] 方程基于最后一个坐标系的所在位置和方向计算所有关节角度  $\theta_{1-6}$ ，已知该坐标系到基座坐标系的变换矩阵为  ${}^0T_6$ 。在下面求解中，所有的关节角度都限制在  $(\theta_1 \dots \theta_6) \in [0; 2\pi]$ 。

首先求解  $\theta_1$ ，有必要先确定最近的坐标系（即手腕坐标系）相对于基坐标系的位置。如图3.2所说明的， ${}^0P_5$  可以向后由沿着  $z_6$  平移坐标系 6 到坐标系 5 得到，而且  ${}^0T_6$  和  $d_6$  是已知的。

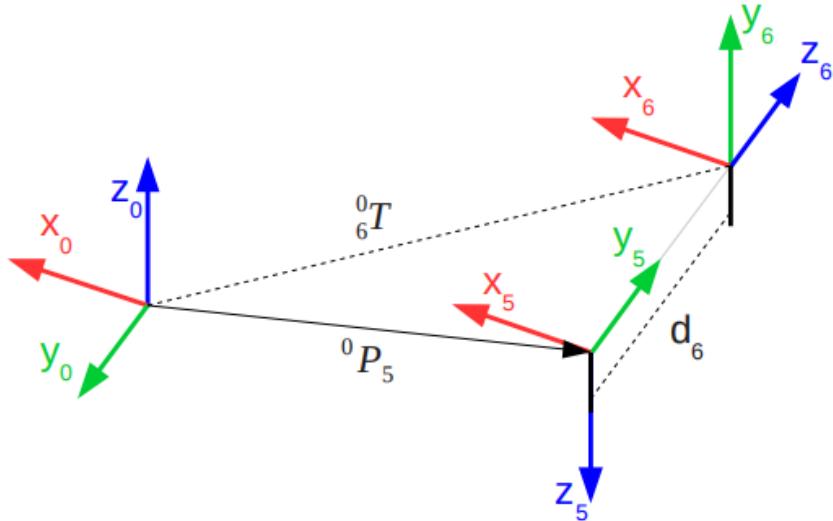


图 3.2: 找到坐标系 5 的原点

平移  ${}^0P_5$  可以表示为：

$${}^0P_5 = {}^0P_6 - d_6 \cdot {}^0\hat{Z}_6 \Leftrightarrow \\ {}^0P_5 = {}^0T_6 \begin{bmatrix} 0 \\ 0 \\ -d_6 \\ 1 \end{bmatrix} \quad (3.4)$$

从上往下观察 UR5 机器人，如图3.3所示。

求出  $\theta_1$  的方法是通过考虑该关节分别相对于坐标系 0 和坐标系 1 的  $P_5$ 。直观来说，从坐标系 0 到坐标系 1 的旋转  $\theta_1$  应该等于从坐标系 0 到坐标系 5 的旋转角度与从坐标系 1 到坐标系 5

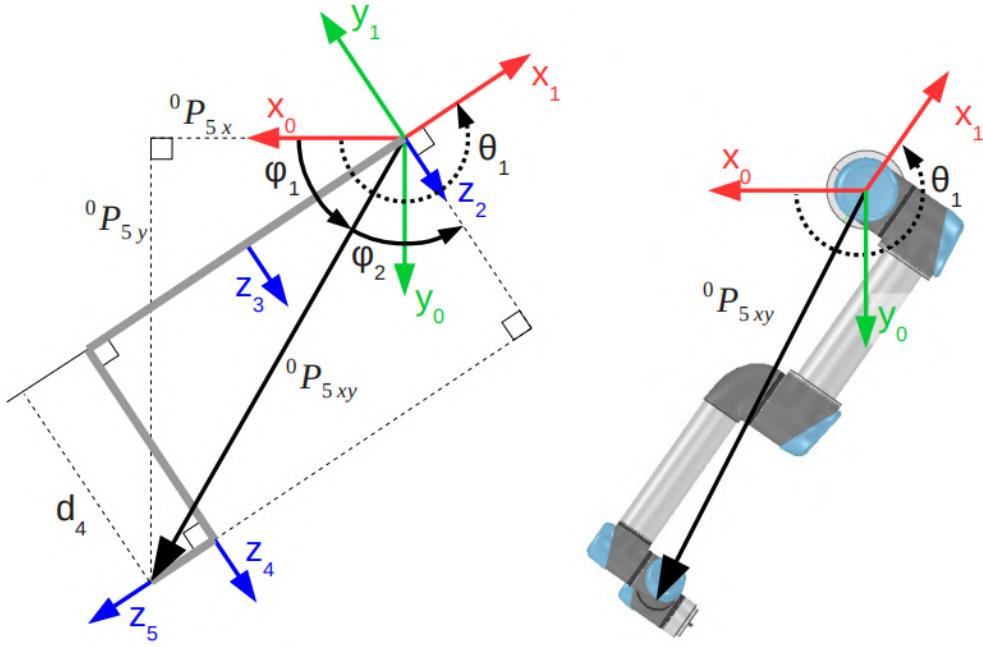


图 3.3: 从上往下观察的 UR5 机器人。该机器人用灰色线段表示, 这里  $\theta_4 \neq 0$  更好地说明了  $\theta_1$  是如何分离的。

的旋转角度的差值, 从图3.3的符号可知:

$$\begin{aligned} v_{0 \rightarrow 1} &= v_{0 \rightarrow 5} - v_{1 \rightarrow 5} \\ v_{0 \rightarrow 1} &= v_{0 \rightarrow 5} + v_{5 \rightarrow 1} \\ \theta_1 &= \phi_1 + (\phi_2 + \frac{\pi}{2}) \end{aligned} \tag{3.5}$$

角度  $\phi_1$  可以由三角边  ${}^0P_{5x}$  和  ${}^0P_{5y}$  关系得到:

$$\phi_1 = \arctan({}^0P_{5y}, {}^0P_{5x}) \tag{3.6}$$

通过观察最右边的三角形,  $\phi_2$  是其中一个角, 而且其中两条边的长度为  $|{}^0P_{5xy}|$  和  $d_4$ , 从而  $\phi_2$  可由此得到:

$$\begin{aligned} \cos(\phi_2) &= \frac{d_4}{|{}^0P_{5xy}|} \Rightarrow \\ \phi_2 &= \pm \arccos\left(\frac{d_4}{|{}^0P_{5xy}|}\right) \Leftrightarrow \\ \phi_2 &= \pm \arccos\left(\frac{d_4}{\sqrt{{}^0P_{5x}}^2 + {}^0P_{5y}}\right) \end{aligned} \tag{3.7}$$

于是角度  $\theta_1$  就由此解出:

$$\begin{aligned} \theta_1 &= \phi_1 + \phi_2 + \frac{\pi}{2} \Leftrightarrow \\ \theta_1 &= \arctan({}^0P_{5y}, {}^0P_{5x}) \pm \arccos\left(\frac{d_4}{\sqrt{{}^0P_{5x}}^2 + {}^0P_{5y}}\right) + \frac{\pi}{2} \end{aligned} \tag{3.8}$$

本章只举例如何求解一个关节的角度, 其他关节角度的解算可同理求出。

## 4 相机-机械臂标定

相机-机械臂的位置关系标定 [46] 是本系统开发中最基础又是非常重要的环节，得到该位置变换关系后能够将相机识别到的视觉信息转换到机械臂的工作空间下，从而机械臂能够感知到相机获取的图像信息，后续的运动控制工作也是基于变换后的传感器数据来完成，例如视觉抓取、物体轨迹跟踪等等。标定工作主要由机械臂自动化完成，标定精度取决于相机内参精度、ARmarker 目标检测算法计算精度，机械臂正运动学解算精度等等。因为本系统用到了两个三维相机，而且各自的安装方式不同，标定方法分别为 eye-in-hand 和 eye-to-hand 手眼标定两种。

### 4.1 相机-机械臂标定通用方法

机器人手眼标定其实就是两个坐标系之间变换关系的标定。假设现在有两个坐标系：机器人基座坐标系和相机坐标系，而且已知某个空间点  $P$  在这两个坐标系的坐标分别为  ${}^{robot}P$  和为  ${}^{camera}P$ 。那么根据坐标系转换公式，可以得到：

$${}^{robot}P = {}^{robot}T_{camera} \cdot {}^{camera}P \quad (4.1)$$

上式中  ${}^{robot}T_{camera}$  表示求出来从相机到机器人的旋转平移矩阵。 ${}^{robot}P$  和  ${}^{camera}P$  为补“1”后的齐次坐标： $[x, y, z, 1]^T$ 。由此求出的齐次坐标变换矩阵  ${}^{robot}T_{camera}$  就同时包含旋转以及平移。于是便可以通过伪逆矩阵计算出来  ${}^{robot}T_{camera}$ ：

$${}^{robot}T_{camera} = {}^{robot}P({}^{camera}P)^{-1} \quad (4.2)$$

上式4.2得到的坐标变换矩阵便可以直接应用于后续的坐标变换。现在计算出来的矩阵允许坐标进行旋转，平移，以及任意方向的缩放。如果想要把这个坐标矩阵变成刚性的变换矩阵，只需要将其中的旋转矩阵  $R$  正交化即可，但是加了这个约束后有可能降低坐标变换精度，即刚性变换矩阵约束太多，造成欠拟合。当然一般情况下，式4.2中的线性矩阵便足以描述机器人和相机之间的坐标变换关系。

通过以上的计算可以得到一个结论：只要能够同时测量出来固定点  $P$  在两个坐标系的坐标  ${}^{robot}P$  和为  ${}^{camera}P$  并且测量出多组数据，就可以很方便地计算坐标变换矩阵。当然在实际过程中有可能很难同时测量出来固定点  $P$  在两个坐标系的坐标，在后面会分相机和机器人的情况讨论如何尽量实现这个测量目标，如何进行标定，以及如何进行标定精度的验证。

## 4.2 eye-in-hand 手眼标定法

一般而言，相机获取到的所有目标的坐标系都是相对于自身相机坐标系的，而相机则需要将检测到的目标位置和旋转信息传递给机械臂，从而让机械臂知道该目标在自身基座坐标系下的三维位置。本系统中，eye-in-hand 手眼标定法的作用是得到从机器人末端关节坐标系到相机坐标系的旋转平移矩阵，从而能够将相机坐标系下的物体位置信息变换到机器人基座坐标系下。本系统将 Realsense-D415 安装在机械臂末端，具体布局如图4.1。

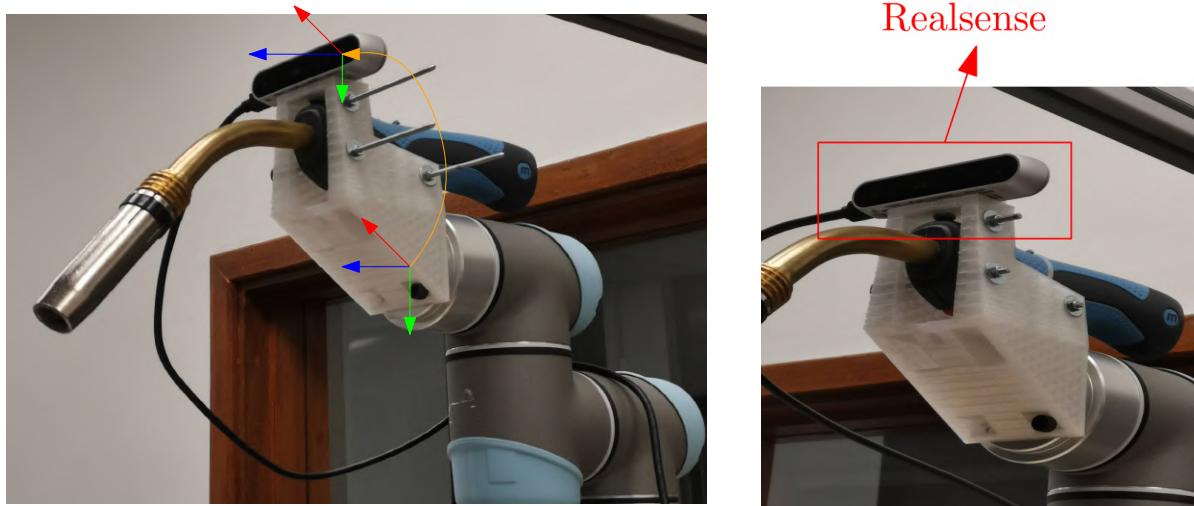


图 4.1: 机械臂末端执行器布局

蓝色、红色、绿色箭头分别代表相机坐标系和机械臂末端坐标系的 Z 轴、X 轴和 Y 轴，橙色箭头代表了需要标定的变换矩阵。为了更方便阐述 hand-in-eye 手眼标定原理，图4.2详细构建了物体在机器人系统坐标系下的位置变换回环。

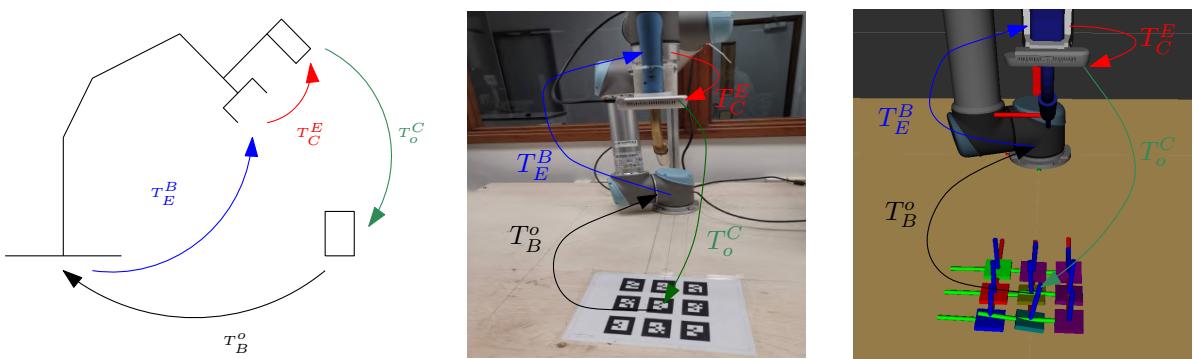


图 4.2: eye-in-hand 手眼标定法示意图

在图4.2中我们需求解  $T_C^E$ :

$$\begin{aligned} T_E^B T_C^E T_o^C &= T_B^o \\ T_C^E &= (T_E^B)^{-1} T_B^o (T_o^C)^{-1} \end{aligned} \quad (4.3)$$

$T_B^o$  是物体到机器人基座坐标系的变换矩阵，该矩阵可以人为地精确测量出， $T_E^B$  是机器人基座坐标系到其末端坐标系的变换矩阵，可以由小节3.1UR5 正运动学原理可知，而且非常精确， $T_o^C$  是相机坐标系到物体坐标系的变换矩阵，在这里我们用 AR 标志4.3来代替物体，同时用开源算法在 RGB 图像上对该标志进行三维定位。



图 4.3: *eye-in-hand* 手眼标定 ARmarker 及 Rviz 中的检测结果

### 4.3 eye-to-hand 手眼标定法

eye-to-hand 手眼标定法是用来标定固定在系统工作空间正上方的相机 Kinect 坐标系到机械臂基座坐标系的变换矩阵的。因为该相机是固定的，而机械臂的基座也是固定的，那么两者之前的变换矩阵也是固定不变的。Kinect 相机与机械臂的位置关系如图4.4所示。

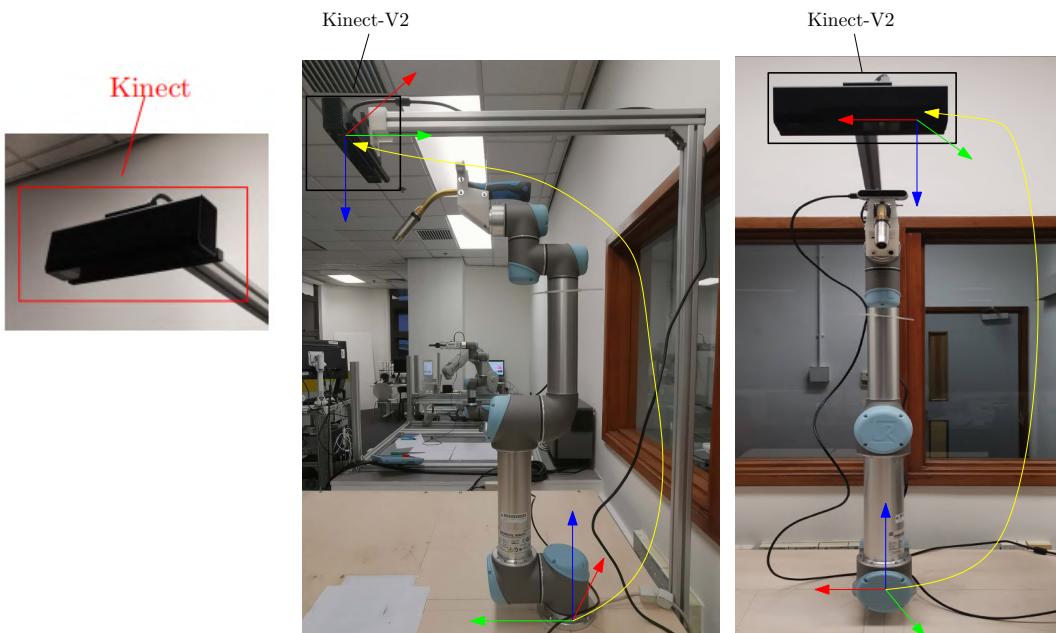
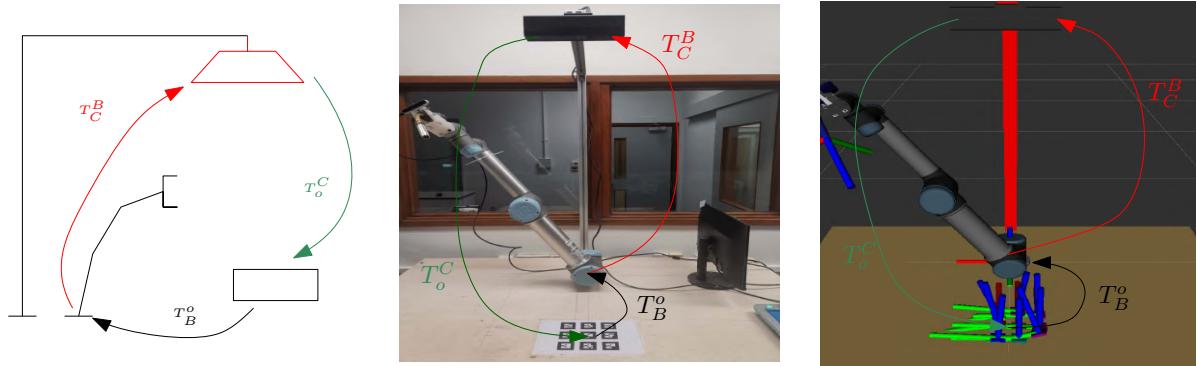


图 4.4: Kinect 相机与机械臂的位置关系

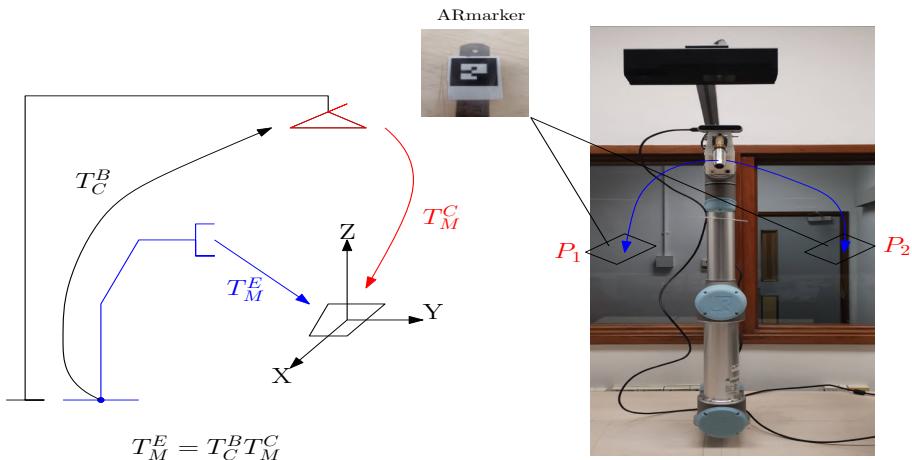
其中 Kinect 相机坐标系和机械臂基座坐标系都已经标注在图中，蓝色、红色、绿色箭头分别代表各自坐标系的 Z 轴、X 轴和 Y 轴，橙色箭头则标明了需要标定的从机械臂基座坐标系到 Kinect 相机坐标系变换矩阵。同样地，与小节4.2标定原理相似，为了标定该矩阵，需要建立从基座坐标系、相机坐标系再到目标坐标系的标定回环，如图4.5所示。

图 4.5: *eye-to-hand* 手眼标定法示意图

于是有：

$$\begin{aligned} T_C^B T_o^C &= T_B^o \\ T_C^B &= T_B^o (T_o^C)^{-1} \end{aligned} \quad (4.4)$$

当得到机械臂基座坐标系到 Kinect 相机坐标系变换矩阵  $T_C^B$  后，只要在系统工作空间范围内用一个 ARmarker 来引导机械臂的目标位姿，那么就可以很轻松地得到机械臂坐标系下的三维目标位置，如图4.6所示。

图 4.6: *eye-to-hand* 手眼标定法示意图

## 5 焊接缝定位及机械臂轨迹规划

### 5.1 焊接缝定位

本工作的焊接缝定位算法完全依赖于焊接工件的三维点云模型，具有不受光照、遮挡影响的优点。本章将通过对曲线型焊接缝的三维定位来详细阐述该定位算法。PCL (Point Cloud Library) 包含了丰富的点云操作算法，如欧式聚类，特征点提取等等。焊接缝定位算法以 C++ 编程语言，PCL 库为基础算法库，对点云处理流程如图5.1所示。

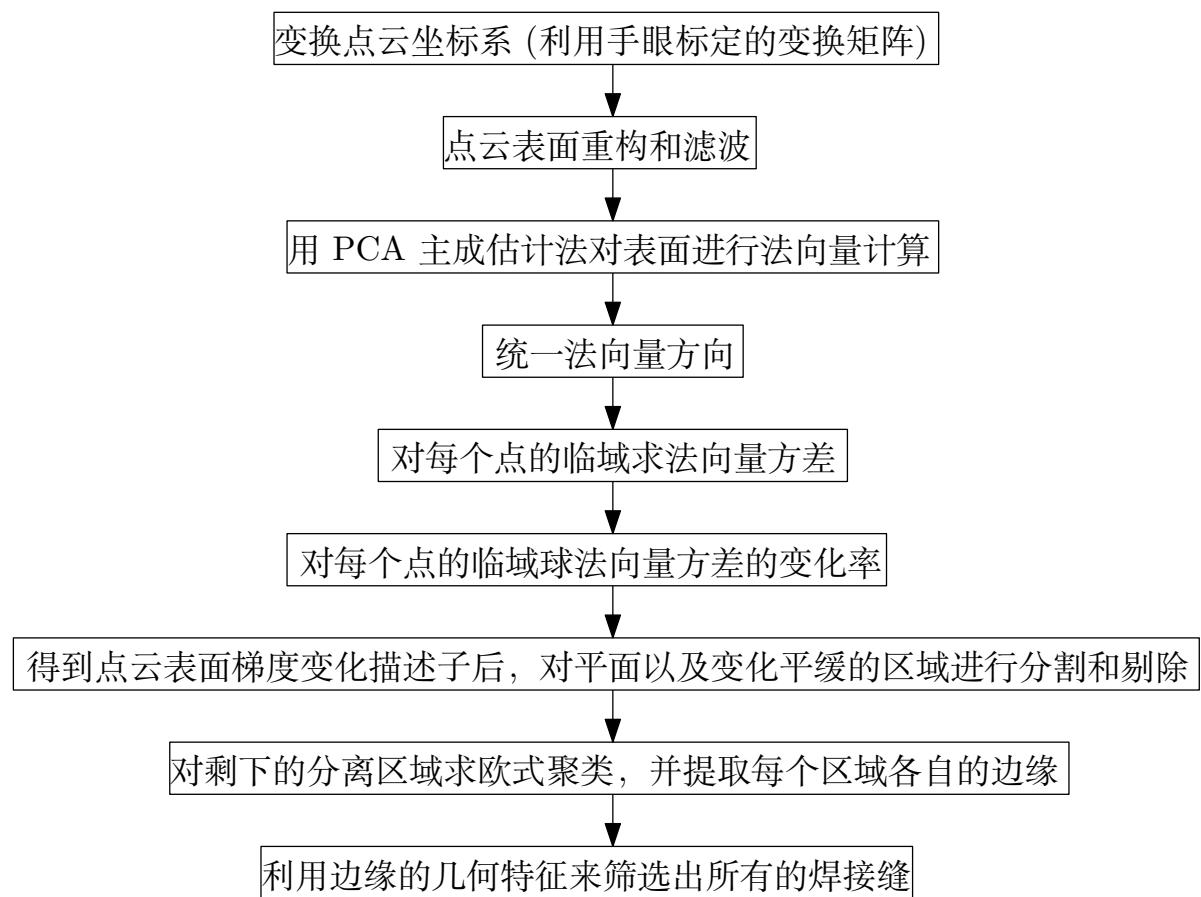


图 5.1: 焊接缝定位算法流程

### 5.1.1 焊接工件表面模型获取

图5.2展示了本系统采集原始点云数据及实验焊接工件。机械臂移动到合适的位姿，该位姿能够使相机的视角覆盖整个焊接工件表面，而且还要保持一定的距离，太远容易导致点云里焊接缝区域的分辨率下降，不利于算法的提取，太近会使相机获取的点云花糊，无法处理。

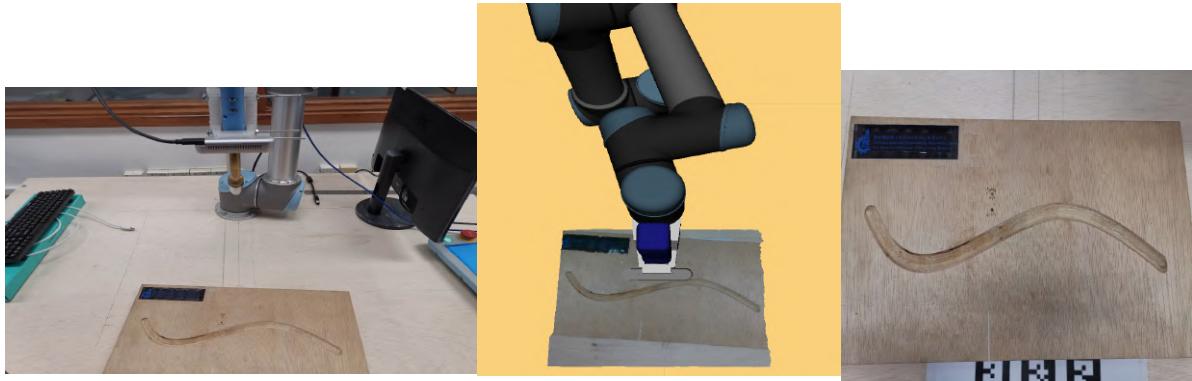


图 5.2: 从左至右依次为实际机器人姿态，仿真环境 *Rviz* 中机器人模型的姿态和实验的焊接工件

点云的原始格式在 RealsenseD415 中是以深度图的形式输出的，而深度图和彩色图是已经配准好的，因此可以通过彩色图的相机内参矩阵将深度图转换成三维点云。假设  $P_m$  和  $P_n$  代表了其中某个点在深度图中的行和列，而且已知该点的深度  $P_d$ ，于是该点在相机坐标系下的三维坐标  $P_i$  可由公式5.1得到：

$$P_i = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} (P_n - c_x) \cdot P_d / f_x \\ (P_m - c_y) \cdot P_d / f_y \\ P_d / \alpha \end{bmatrix} \quad (5.1)$$

上式中  $\alpha$  为深度值的尺度系数，默认为 1000， $c_x$ ,  $c_y$ ,  $f_x$ ,  $f_y$  构成了相机内参矩阵  $K_c$ ，5.2式中也说明了实验中所标定的内参参数：

$$K_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 619.9661254882812 & 0 & 311.2325744628906 \\ 0 & 619.856201171875 & 226.9261474609375 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

通过对  $640 \times 480$  中所有的点进行上述从相机平面到三维空间的变换，就得到了一帧焊接工件的表面点云。然后还需将该相机坐标系下的点云转换到机械臂基座坐标系下，根据公式4.3得到

的从机械臂末端坐标系到相机坐标系的变换矩阵  $T_C^E$ , 则点云变换为:

$$\begin{aligned} {}^{Base}P_i &= T_E^B \cdot T_C^E \cdot \begin{bmatrix} R_o^C & t_o^C \\ 0 & 1 \end{bmatrix} \\ &= T_E^B \cdot T_C^E \cdot \begin{bmatrix} 1 & 0 & 0 & {}^{Cam}P_{ix} \\ 0 & 1 & 0 & {}^{Cam}P_{iy} \\ 0 & 0 & 1 & {}^{Cam}P_{iz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (5.3)$$

在上式5.3中从相机坐标系到单个点坐标系的旋转矩阵  $R_o^C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ , 表示在相机和每个单点之间没有发生任何旋转变换, 这很容易理解, 每个点可以看成一个半径  $< 0.5mm$  的球, 无论任何旋转都不会对坐标系变换结果产生影响, 因为我们只需要得到点云变换后的三维位置信息。

### 5.1.2 点云表面重构和滤波

对于输入的原始点云需要对其进行预处理, 一般来说, 由传感器采集到的原始数据具有大量的噪声, 本系统在实验中采集到的点云表面具有许许多多的断层和缺口, 因此为了保证后续算法的稳定计算, 在此步骤采用了 PCL 中的滑动最小二乘算法 (MLS) 来进行点云表面的重建与平滑, 图5.3阐述了该算法原理。

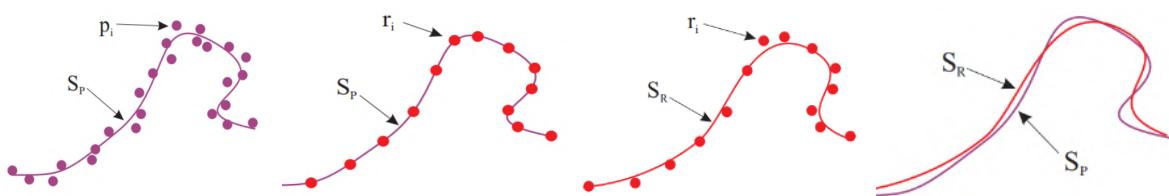


图 5.3: 点云表面重建原理。可能有噪音或冗余的点集 (紫色点) 定义了一个流形 (紫色曲线), 该流形用 (红色) 表示点采样, 表示点定义不同的流形 (红色曲线), 表示点的间距取决于所需的近似精度。

一些数据不规则 (由小距离测量误差引起) 很难用统计分析来消除。要创建完整的模型, 必须考虑数据中的光泽曲面和遮挡。在无法获取额外扫描的情况下, 解决方案是使用重采样算法, 该算法尝试通过在周围数据点之间进行高阶多项式插值来重新创建曲面的缺失部分。通过执行重采样, 可以纠正这些小错误, 并且可以平滑由于将多个扫描注册在一起而产生的“双墙”伪影。

图5.4显示了四个不同的点云区域前后表面重建的结果, 可以看出相对于原始点云, 处理后的点云表面更加平滑, 没有大量的缺陷区域, 从而重建后的点云将输入到下一阶段。

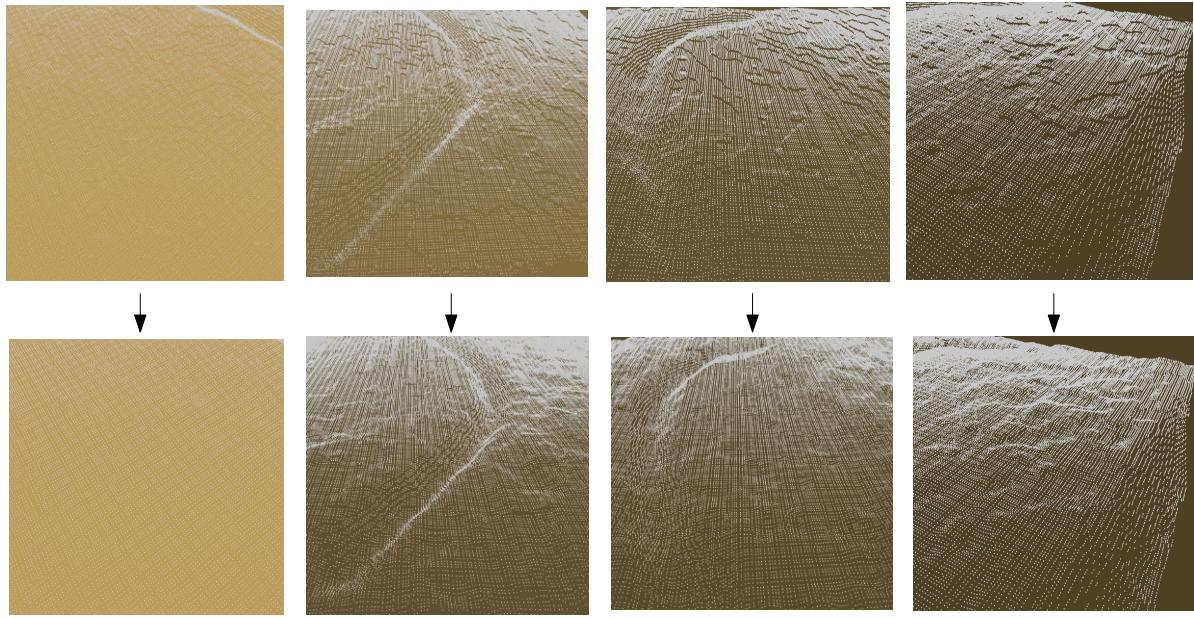


图 5.4: 实验点云表面重建前后结果。

### 5.1.3 点云表面法向量计算

点云表面法向量的计算是提取点云特征的一个重要步骤。法向量估计：表面法线是空间几何体表面的基本属性，在很多领域都有大量应用，比如 ICP 配准，以及曲面重建。对于一个已知的几何体表面，根据垂直于点表面的矢量，从而求出表面某一点的法线方向。然而，由于本系统的相机采集的点云数据集在真实物体的表面表现为一组定点样本，这样就会有两种解决方法：

- 使用曲面重建技术，从获取的点云数据集中得到采样点对应的曲面，然后从曲面模型中计算表面法线；
- 直接从点云数据集中近似推断表面法线。

在本工作的点云算法中所用到的法向量估计方法是，求表面某点法线的问题近似于估计表面的一个相切面法线的问题，进而在数学上表达成一个最小二乘法平面拟合估计问题。解决表面法线估计的问题可以最终化简为对一个协方差矩阵的特征向量和特征值的分析（或者也叫 PCA-Principal Component Analysis 主成分分析），这个协方差矩阵是由单个查询点的最近邻域产生的。

基于 PCA 的点云法向量估计，其实是从最小二乘法推导出来的。假设要估计某一点的法向量，需要通过利用该点的所有近邻点估计出一个经过该点的切平面，然后通过该平面方程 ( $ax + by + cz + d = 0$ ) 求出该点的法向量。或者可以说，通过最小化一个目标函数（要求的参数为法向量），使得该点与其每个近邻点所构成的向量尽可能与法向量的点乘为 0，也就是垂直：

$$\min_{c, \mathbf{n}, \|\mathbf{n}\|=1} \sum_{i=1}^n ((x_i - c)^T \mathbf{n})^2 \quad (5.4)$$

在建立目标函数前，需要得到某查询点的邻域中所有点的中心点，也就是质点：

$$m = \frac{1}{n} \sum_{i=1}^n X_i \quad (5.5)$$

$$y_i = X_i - m$$

于是优化函数就变成：

$$\min_{\|\mathbf{n}\|=1} \sum_{i=1}^n (y_i^T \mathbf{n})^2 \quad (5.6)$$

进一步对该优化函数推导得到：

$$\min_{\mathbf{n}^T \mathbf{n} = 1} \sum_{i=1}^n (y_i^T \mathbf{n})^2 = \min_{\mathbf{n}^T \mathbf{n} = 1} \sum_{i=1}^n \mathbf{n}^T y_i^T y_i \mathbf{n} = \min_{\mathbf{n}^T \mathbf{n} = 1} \mathbf{n}^T \left( \sum_{i=1}^n y_i^T y_i \right) \mathbf{n} = \min_{\mathbf{n}^T \mathbf{n} = 1} \mathbf{n}^T Y^T Y \mathbf{n} \quad (5.7)$$

上面的优化目标函数就简化为：

$$f(\mathbf{n}) = \mathbf{n}^T S \mathbf{n} \quad (5.8)$$

其中  $S = (Y^T Y)$ ,  $\min(f(\mathbf{n}))$ ,  $\mathbf{n}^T \mathbf{n} = 1$ ,  $Y^T Y$  是一个  $3 \times 3$  的协方差矩阵，即 x,y,z 坐标的协方差矩阵。对于这种有约束的优化目标函数，这里用拉格朗日算法进行求解：

$$\begin{aligned} \sigma(\mathbf{n}, \lambda) &= f(\mathbf{n}) - \lambda(\mathbf{n}^T \mathbf{n} - 1) \\ \nabla \sigma &= 0 \end{aligned} \quad (5.9)$$

$$\begin{aligned} \frac{\partial \sigma}{\partial \mathbf{n}} &= \frac{\partial}{\partial \mathbf{n}} f(\mathbf{n}) - \lambda \frac{\partial}{\partial \mathbf{n}} (\mathbf{n}^T \mathbf{n} - 1) \\ \frac{\partial \sigma}{\partial \lambda} &= \mathbf{n}^T \mathbf{n} - 1 \end{aligned} \quad (5.10)$$

$$\frac{\partial}{\partial \mathbf{n}} f(\mathbf{n}) - \lambda \frac{\partial}{\partial \mathbf{n}} (\mathbf{n}^T \mathbf{n} - 1) = (S + S^T) \mathbf{n} - \lambda(I + I^T) \mathbf{n} = 2S\mathbf{n} - 2\lambda\mathbf{n} \quad (5.11)$$

$$\begin{aligned} \frac{\partial \sigma}{\partial \mathbf{n}} &= 0 \iff S\mathbf{n} = \lambda\mathbf{n} \\ \frac{\partial \sigma}{\partial \lambda} &= 0 \iff \mathbf{n}^T \mathbf{n} = 1 \end{aligned} \quad (5.12)$$

$S\mathbf{n} = \lambda\mathbf{n}$  就是线性代数里求特征值和特征向量的相关公式。也就是说， $\lambda$  是矩阵  $S$  的特征值，法向量是对应的特征向量，所以法向量  $\mathbf{n}$  的求解就是要对  $S$  进行向量分解。然后取特征值最小的特征向量作为求解的法向量。这个过程是 PCA 的一个标准求解过程，将每个邻域点与该邻域的中心点相减，得到一个  $n \times 3$  的矩阵。接着，用 SVD 对该矩阵进行分解，得到：

$$Y = U \sum V^* \quad (5.13)$$

$U$  中最后一列就是要求解的法向量  $\mathbf{n}$ , 也就是特征值最小的特征向量。PCA 的协方差矩阵的特征值越大, 其特征向量就越能够描述数据的特征。越小就越不能区分样本之间的不同, 也就是表征了数据中的共性。在上述的优化目标函数中, 就是要为所有的邻域点寻找一个平面, 尽可能使得所有的邻域点都在这个平面上, 或者说所有点与该平面的距离最小, 而不是最大。而这个平面就是能够描述这些点的共性, 即他们的法向量几乎都相同。

### 5.1.4 法向量方向统一

经过公式5.13对整块点云的计算, 就得到了每个点所在表面的法向量。但是此时所有点的法向量并没有一个统一的方向, 有些点的法向量可能指向点云表面内侧, 有些指向点云表面的外侧, 因此为了计算表面变化梯度, 有必要为所有法向量确定一个标准方位。

由图5.5所示, 将单个点指向相机坐标系原点的方向规定为基准方向。

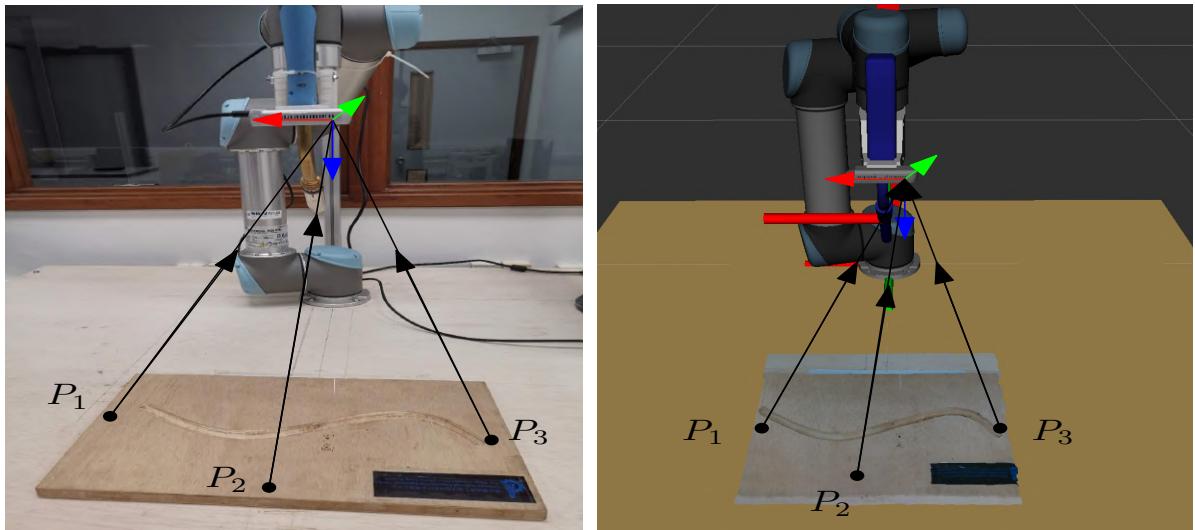


图 5.5: 以任意三个点为例的法向量基准方向

已知相机坐标系相对于机器人基座坐标系的变换为:

$$\begin{aligned} {}^{Base}P_C &= T_E^B \cdot T_C^E \\ {}^{Base}P_C &= \begin{bmatrix} R_C^B & t_C^B \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (5.14)$$

那么得到相机坐标系原点坐标为  $P_C = t_C^B = \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix}$ , 经过5.3式可得到每个点在机器人基座

坐标系下的三维坐标, 从而每个点  $P_i$  的法向量的基准方向可由方向向量定义:

$$\mathbf{P}_c^i = P_C - P_i \quad (5.15)$$

由5.13式已经得到了每个点的法向量  $\mathbf{n}_i$ , 于是点法向量与其基准方向向量的夹角可得:

$$\beta_i = \arccos \frac{\mathbf{n}_i \cdot \mathbf{P}_c^i}{|\mathbf{n}_i| \cdot |\mathbf{P}_c^i|} \quad (5.16)$$

对于  $\beta_i \cdot 180/\pi$  在  $90^\circ-180^\circ$  的法向量反向, 即  $\times(-1)$ , 将所有点的法向量统一基准方向后并在 PCL-viewer 中显示, 以每五个点为间隔, 红色点为点云, 白色箭头表示该点法向量。如图5.6所示。

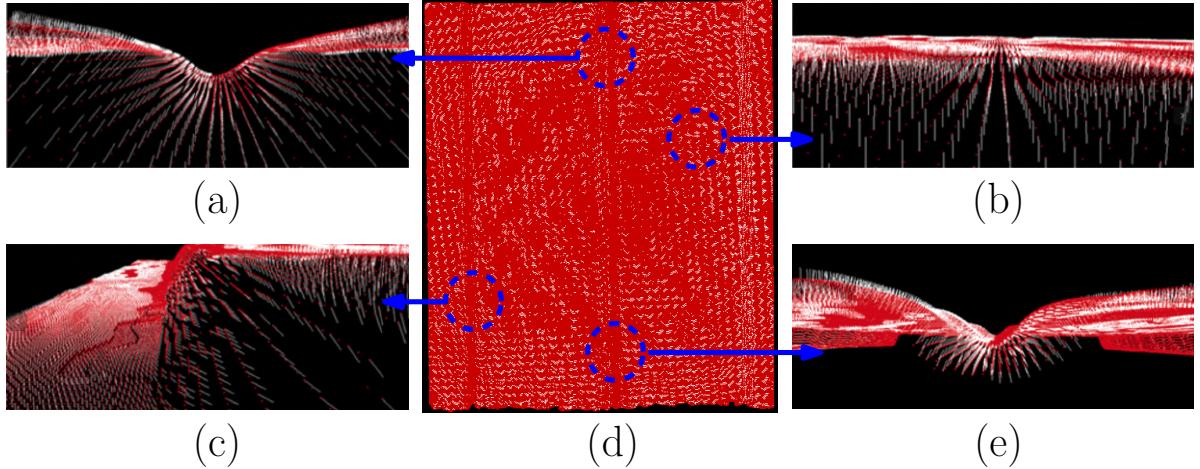


图 5.6: 法向量统一并显示

### 5.1.5 点云梯度描述子

每个点的邻域被定义为一个球体(点是它的邻接中心点)具有恒定半径  $r$ , 在这个球体内, PCL 库里的 kdTree 方法用来搜寻其他邻域内的点(与中心点的欧式距离不超过  $r$ )。邻域内所有点被当成一个单独的点集, 而且每个点都有自己的法向量5.6。将中心点与每一个邻域内的点配对, 每对的两个单位向量都形成一个空间夹角。图5.7展示了一个点的邻域以及其中一对点的法向量关系。

由前面的计算可知, 每个点的三维位置经过刚体变换, 都是相对于机械臂基座坐标系的, 因此所有的点法向量都基于同一个坐标系  $X - Y - Z$  (图5.7右)。则每一个单点邻域内的第  $j$  对法向量的夹角定义为:

$$\theta_j = \arccos \frac{\vec{u}_c \cdot \vec{u}_j}{|\vec{u}_c| \cdot |\vec{u}_j|} \quad (5.17)$$

$\vec{u}_c$  是中心点的法向量,  $\vec{u}_j$  是第  $j$  个与其配对的邻域点的法向量。用公式5.17对邻域内的每一个点进行迭代计算, 将所有的计算结果用一个集合来表示:  $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ ,  $n$  是邻域点的数量。

另外, 单位基准法向量  $\vec{u}_b$  代表了整个点云的主方向, 定义为:

$$\vec{U}_b = \sum_{i=0}^n \vec{u}_i, \quad \vec{u}_b = \frac{\vec{U}_b}{|\vec{U}_b|} \quad (5.18)$$

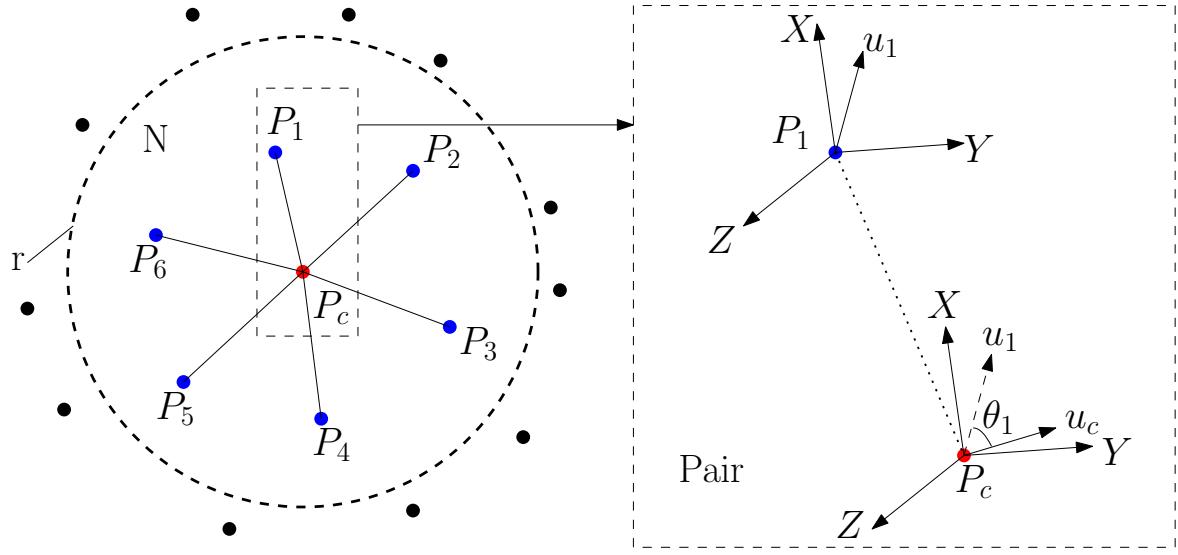


图 5.7: (左)一个单点的邻域  $N$  定义为一个半径为  $r$  的球, 在邻域  $N$  内红色点是球的中心点  $P_c$ , 蓝色点都为邻域点。(右) 中心点  $P_c$  和它的法向量  $\vec{u}_c$  与其中一个邻域点  $P_1$  和它的法向量  $\vec{u}_1$  配对。 $\theta_1$  为  $\vec{u}_c$  和  $\vec{u}_1$  的夹角。

$\vec{u}_i$  是点云中第  $i$  个与其配对的邻域点的法向量。再回到之前的单点 (5.17 中心点  $P_c$ ) , 在此邻域中的每一个点法向量都和基准向量配对  $\vec{u}_b$  , 如图5.8所示。

同理, 在此邻域内的第  $k$  个法向量对夹角  $\beta_k$  定义:

$$\beta_k = \arccos \frac{\vec{u}_b \cdot \vec{u}_k}{|\vec{u}_b| \cdot |\vec{u}_k|} \quad (5.19)$$

从而所有的迭代结果同样地被整合成一个集合  $B = \{\beta_1, \beta_2, \dots, \beta_n\}$ 。根据图5.7和图5.8对点法向量的分析计算, 可以得到两个集合的方差:

$$\begin{aligned} \sigma^l &= \frac{\sum_{j=0}^n (\theta_j - \bar{\theta})^2}{n} \\ \sigma^g &= \frac{\sum_{k=0}^n (\beta_k - \bar{\beta})^2}{n} \end{aligned} \quad (5.20)$$

其中  $\bar{\theta}$  和  $\bar{\beta}$  分别对该点的两个集合求平均得到:

$$\begin{aligned} \bar{\theta} &= \frac{\sum_{j=0}^n \theta_j}{n} \\ \bar{\beta} &= \frac{\sum_{k=0}^n \beta_k}{n} \end{aligned} \quad (5.21)$$

$n$  为邻域内配对的数量, 于是点云的第  $i$  个单点邻域内的点云表面变化程度  $v_i$  定义为:

$$v_i = \sqrt{(\sigma_i^l)^2 + (\sigma_i^g)^2} \quad (5.22)$$

为了描述整体点云表面变化梯度描述子  $\left\{P_i\right\}_{i=1, \dots, n}^T$  , 将所有单点邻域内的表面变化程度  $v_i$  整合成一个集合:  $V = \{v_1, v_2, \dots, v_n\}$  , 同样地对每个单点以半径  $r$  为球的邻域内求集合  $V$  的方差  $\sigma^v$ :

$$\begin{aligned} \bar{V} &= \frac{\sum_{i=0}^n v_i}{n} \\ \sigma^v &= \frac{\sum_{i=0}^n (v_i - \bar{V})^2}{n} \end{aligned} \quad (5.23)$$

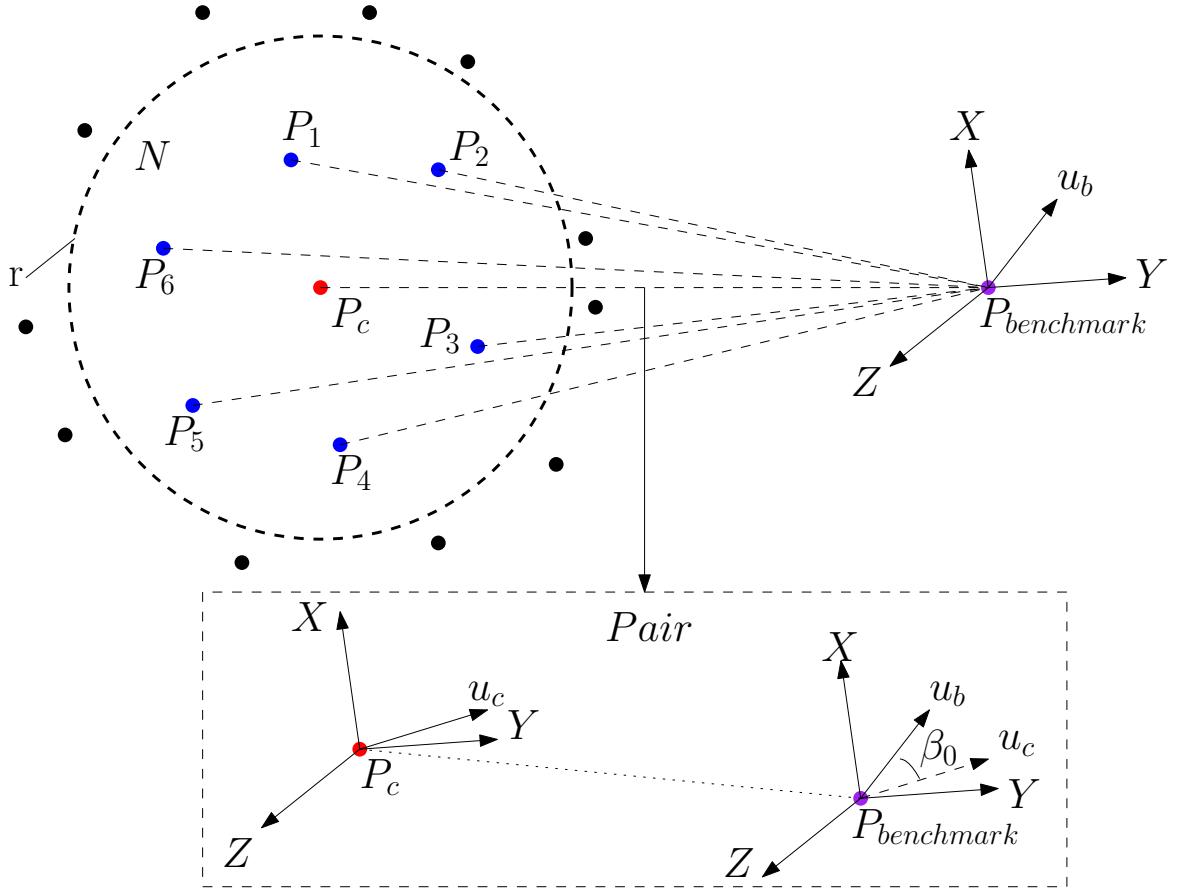


图 5.8: 邻域内的每一个点法向量都与基准法向量配对。以其中一对为例 (中心点法向量  $\vec{u}_c$  和基准法向量  $\vec{u}_b$ ),  $\beta_0$  是  $\vec{u}_c$  和  $\vec{u}_b$  的夹角

于是最后就得到了标量集合  $\left\{\sigma_i^v\right\}_{i=1,\dots,n}$ :

$$\left\{\sigma_i^v\right\}_{i=1,\dots,n} = \{\sigma_1^v, \sigma_2^v, \dots, \sigma_n^v\} \quad (5.24)$$

### 5.1.6 焊接缝区域分割

因此只要设置一个合适的描述子标量阈值, 就能将焊接缝区域分割出来。另外在某些情况下还需要加入人为干预, 例如一帧点云里面有多条焊接缝, 利用上述方法会将所有的焊接缝区域分割出来, 此时需要操作员人为地选择要焊接的区域。图5.9 (a) 展示了分割出来的焊接缝点集。

## 5.2 机械臂轨迹运动规划

### 5.2.1 焊接缝区域边缘提取

当操作员选出了目标焊接区域后5.9 (a), 该区域实际上被定义为焊接缝点云, 后续的算法将以该点云为基础计算六自由度焊接轨迹 (6-DOF)。首先用点云边缘算法提取出焊接缝点云的边缘,

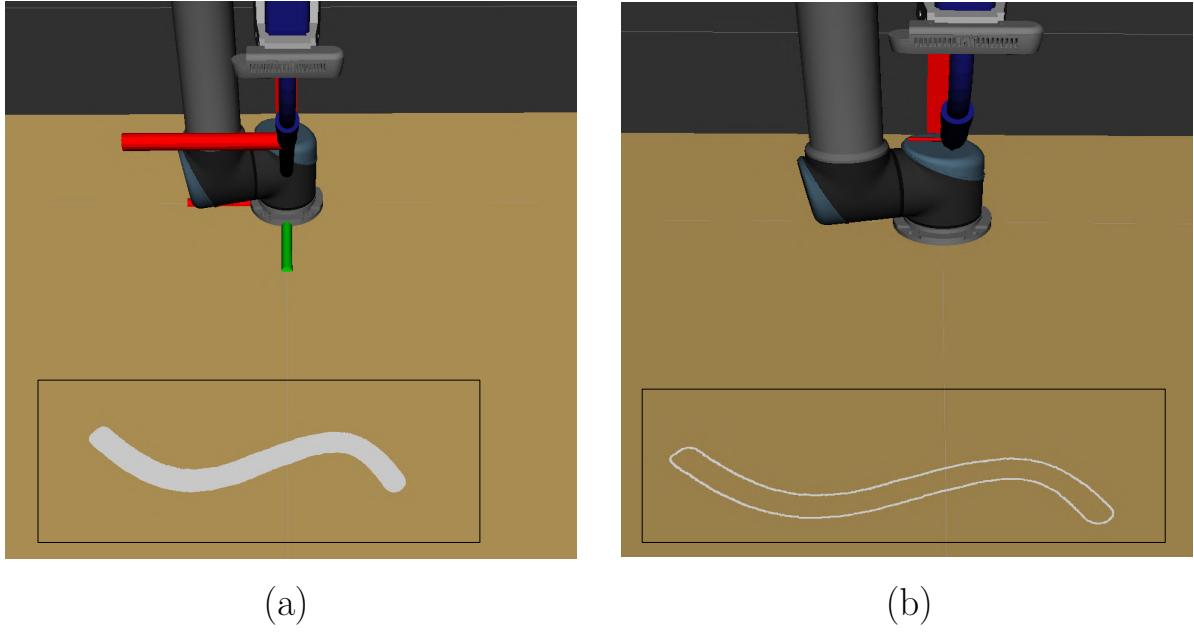


图 5.9: 焊接缝

该方法分析每一个查询点邻域内的散点分布:

$$\begin{aligned} V_X &= \frac{1}{k-1} \sum_i^k (x_i - x_c) \\ V_Y &= \frac{1}{k-1} \sum_i^k (y_i - y_c) \\ V_Z &= \frac{1}{k-1} \sum_i^k (z_i - z_c) \end{aligned} \quad (5.25)$$

其中  $k$  为邻域内所有点的数量,  $x_i, y_i, z_i$  是每个邻域点的三维坐标,  $x_c, y_c, z_c$  查询点(中心点)的三维坐标,  $V_X, V_Y, V_Z$  描述了每个邻域点在各自  $X, Y, Z$  轴上偏离中心点的程度。然后参数  $V_X, V_Y, V_Z$  的范围定义为:

$$\begin{aligned} R_X &= \max(V_X) - \min(V_X) \\ R_Y &= \max(V_Y) - \min(V_Y) \\ R_Z &= \max(V_Z) - \min(V_Z) \end{aligned} \quad (5.26)$$

再通过比较每个邻域点的  $V_X, V_Y, V_Z$  与各自的范围:

$$\begin{aligned} \text{abs}(V_X) &> \text{abs}(R_X)/C \\ \text{abs}(V_Y) &> \text{abs}(R_Y)/C \\ \text{abs}(V_Z) &> \text{abs}(R_Z)/C \end{aligned} \quad (5.27)$$

参数  $C$  是与邻域点数量  $k$  成正比关系, 如果式5.27中任意一个不等式成立的话, 那么这个查询点就被认为是边界点。利用此方法遍历焊接缝点云, 得到所有的边界点, 如图5.9 (b) 所示。

### 5.2.2 焊接轨迹点生成

根据实际的焊接要求，焊枪头的位置应该不能偏离焊接缝的中心线，因此很容易确定焊接轨迹应该尽量沿着其中心线。而且还需要确定轨迹在焊接缝里的深度，一般而言，焊接缝的宽度不超过或者等于焊枪头的直径，所以焊枪最靠近焊接缝的位置在焊接缝的开口处，如图5.10所示。

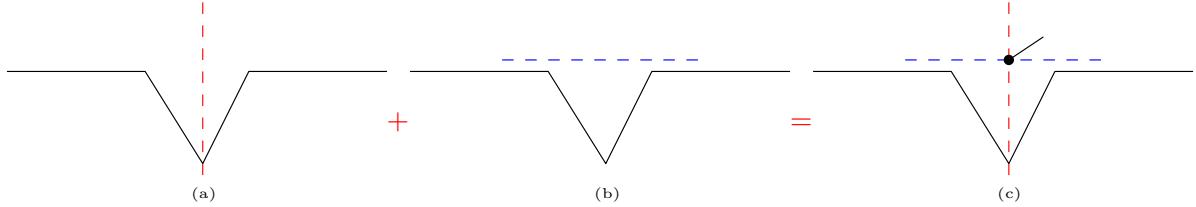


图 5.10: 焊接轨迹位置示意图

由图5.10分析可知，直接找到两个对应的点，再求它们的中心点，即为焊接轨迹上的一点。图5.11展示了三维空间下三种不同的点云边缘分布情况，点云边缘两边是近似平行的，这和实际焊接缝的形状是相符的。可以看出，一对对应点的连线应该大致与生成的焊接轨迹垂直（夹角至少在  $90^\circ$  附近）。

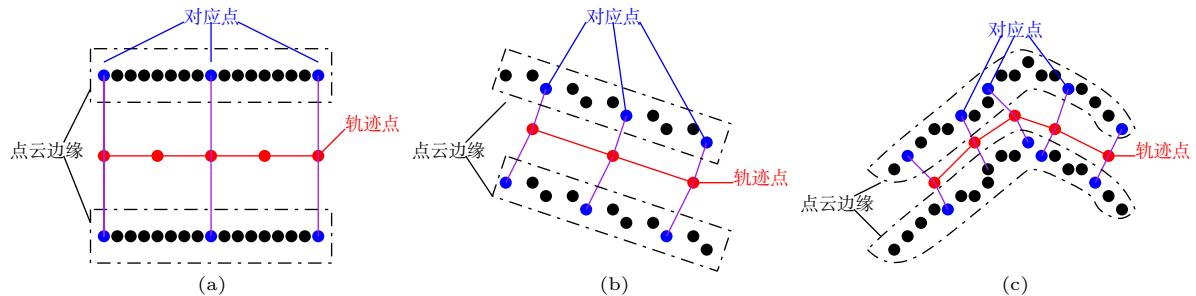


图 5.11: 生成轨迹示意图

但是原始点云是无序点云 (unordered pointcloud)，焊接缝点云5.9 (a) 的边缘也是无序点云。因此无法直接确定某一点的对应点，即无法确定对应点的序号，这里引入三维空间的拓扑关系来寻找每个点的对应点。如果将点云边缘看成一条闭环曲线的话，该方法就是要找到任一查询点在另一条近似平行的曲线上的对应点，而且查询点与其对应点的连线与查询点的切线大致垂直。同样地，以某个查询点为例，利用 kd-Tree 最近邻搜索算法对该点以一定半径范围搜索附近的邻域点。在搜索初期，因为无法直接确定两条平行曲线之间的距离 (焊接缝宽度)，只能以非常小的半径作为初始值 (1mm) 来搜索，如图5.12 (a) 所示。为了方便计算，将穿过查询点切线的方向向量定义为：

$$\vec{v_{\perp}} = P_l^n - P_r^n \quad (5.28)$$

$$|P_l^n - P_r^n| = d_{max}^{nn}$$

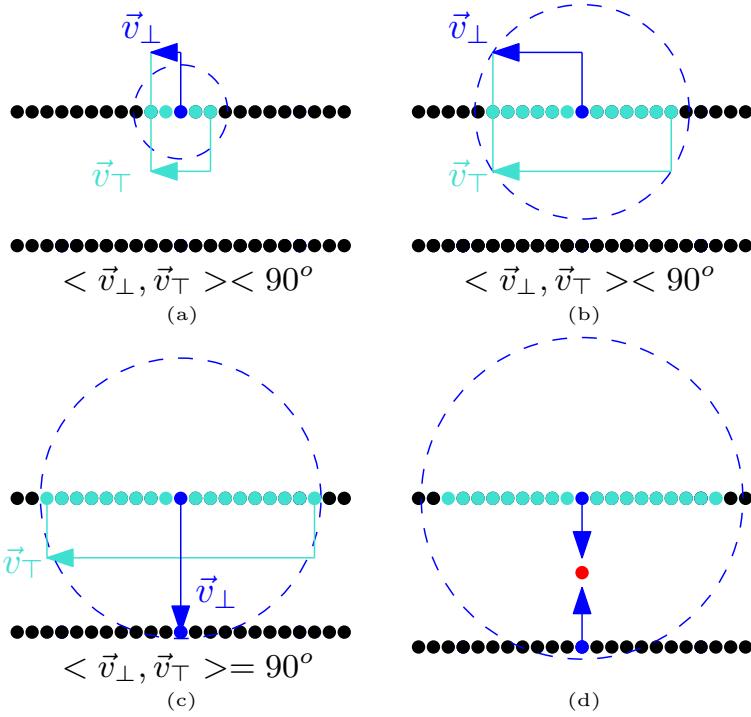


图 5.12: 搜索对应点的迭代过程

其中邻域点  $P_l^n$  和邻域点  $P_r^n$  之间的欧式距离是该查询点邻域内点与点间距离中最大的。同时由查询点指向任一邻域点  $P_i^n$  的方向向量定义为:

$$\vec{v}_{\perp} = P_i^n - P_c^n \quad (5.29)$$

从而计算  $\vec{v}_{\tau}$  和每个  $\vec{v}_{\perp}$  之间的夹角 (规定在  $0^\circ - 180^\circ$  之间) :

$$\alpha_i = \arccos \frac{\vec{v}_{\tau} \cdot \vec{v}_{\perp}}{|\vec{v}_{\tau}| \cdot |\vec{v}_{\perp}|} \quad (5.30)$$

如果遍历所有的邻域点之后都没有满足下列条件:

$$90^\circ - \epsilon < \alpha_i < 90^\circ + \epsilon \quad (5.31)$$

$\epsilon$  为垂直范围容忍值。这说明最近邻搜索范围还不够大, 没有将对应点囊括进来, 如图5.12 (a) (b) 所示。于是继续增大搜索半径直到找到对应点满足条件5.31, 如图5.12 (c) 所示, 那么由该查询点产生的轨迹点就定义为两点的中点:

$$P_t = \frac{P_i + P_j}{2} \quad (5.32)$$

用上述方法遍历焊接缝点云边缘上的每一个查询点, 就得到了所有未处理的轨迹点, 如图5.13所示。

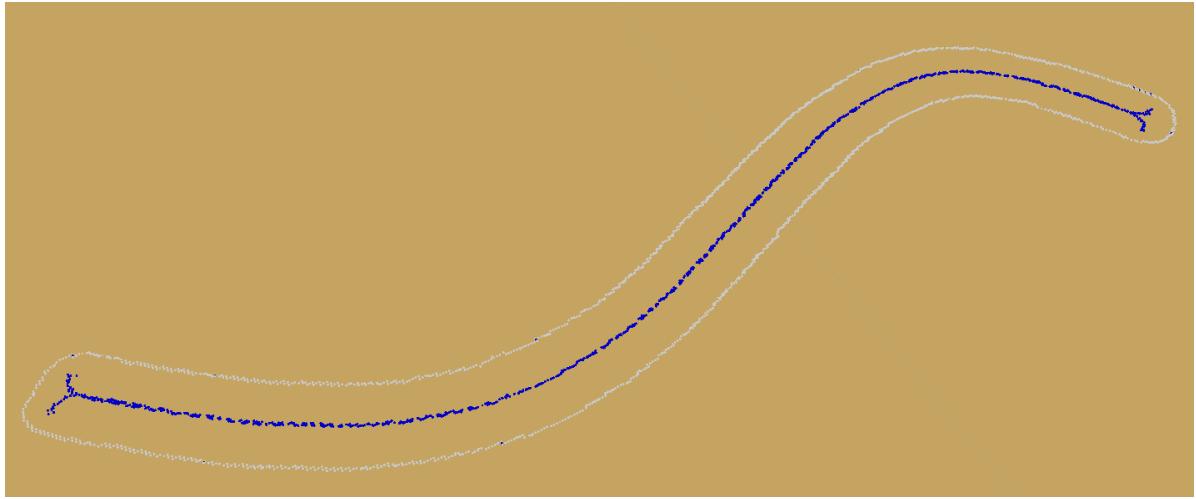


图 5.13: 未处理的原始轨迹点云

### 5.2.3 焊接轨迹后处理

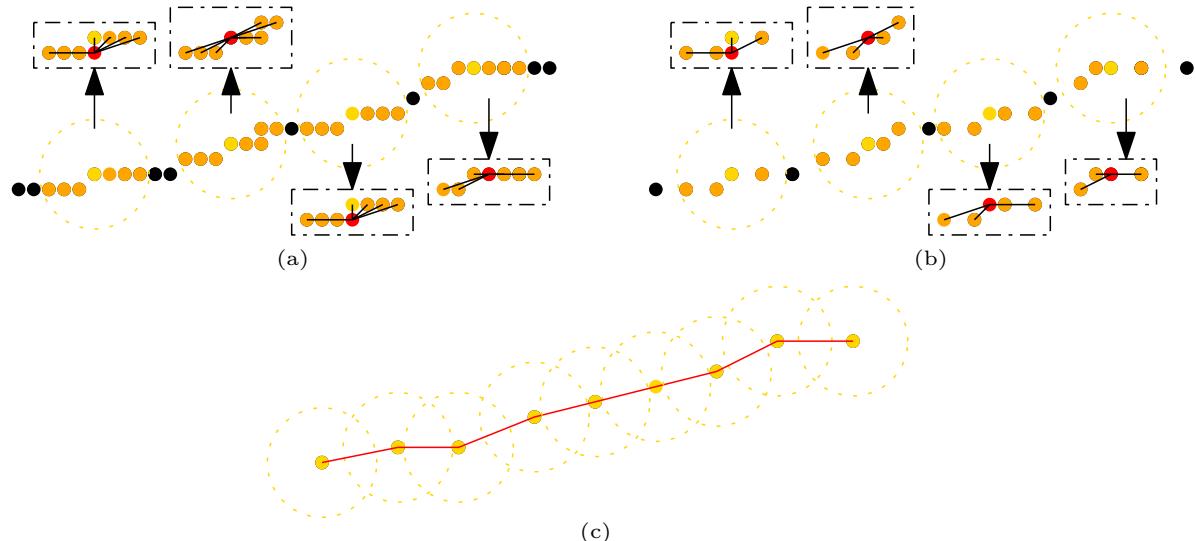


图 5.14: 降稠密度过程示意图

如图5.13所示，可以看到该轨迹有大量噪声点、错位点以及轨迹点分布不均匀等问题，因此需要对其进行曲线平滑、滤波等后处理。利用 kd-Tree 最近邻搜索方法，以恒定的半径对原始轨迹点云上某一段分析，通过求每个轨迹点邻域内所有点的质心，将质心作为新的轨迹点，并用此方法不断迭代，直到每个最新轨迹点的邻域内只有一个点时，终止迭代，得到了降稠密度后的轨迹点云。该迭代方法过程如图5.14所示。每次迭代后单个轨迹点邻域的质心定义为：

$$P_g^n = \frac{\sum_{i=1}^n P_i^n}{n} \quad (5.33)$$

$P_i^n$  是单个轨迹点邻域内的第  $i$  个点（包括该轨迹点），这种以邻域质心来代替新轨迹点的迭代方法，使得零散的噪声点被滤除，并经过一定迭代次数后（实际是 50 次）就获得了相对平滑地

多的焊接轨迹，如图5.15所示。



图 5.15: 降稠密度后的轨迹点云

将原始焊接轨迹下采样处理后，该方法是对每一个点轨迹点迭代，而且每一次迭代都是相同的轨迹点数量，因此轨迹点的稠密度下降了，但是最后的点数量并没有减少，只是有很多点重叠成了一个点。而最终的轨迹应该是一个一个单独的点组成的，所以接下来对每个点进行合并操作，PCL 提供了 VoxelGrid Filter 网格化滤波器来对降稠密度后的轨迹点云进行下采样处理。

对应 pcl::VoxelGrid，它的原理是根据输入的点云，首先计算一个能够刚好包裹住该点云的立方体，然后根据设定的分辨率，将该大立方体分割成不同的小立方体。对于每一个小立方体内的点，计算他们的质心，并用该质心的坐标来近似该立方体内的若干点，从而达到对点云下采样的目的，如图5.15所示。

尽管在可视化上和降稠密度后的轨迹点云并无区别，但轨迹点的数量大大减少，这样就得到了焊接轨迹的主干轨迹点。对于实际焊接而言，轨迹运动方向是一个必不可少的考虑因素，即相邻的两个轨迹点的运动方向一定是连续的，但是图5.15 所示的轨迹点并没有一个统一的序号来规划焊接方向，因此需要对其无序轨迹点云进行排序。

首先找到轨迹点云的几何中心点，定义为：

$$\min(d_i) = \sum_{k=1}^n |P_c - P_k| \quad (5.34)$$

$d_i$  是点云中所有点到点  $P_k$  的距离之和，式5.32说明当  $d_i$  最小时， $P_c$  被确定为轨迹点云几何中心点。以该中心点为分界线，在三维空间将轨迹点云分成两部分，由中心点作为起始点分别对两部分进行轨迹点搜索及排序。接下来用 kd-Tree 最近邻搜索方法对该中心点附近的轨迹点进行搜索，目的是找到距离该点最近的轨迹点，如图5.16 (a) (b) 所示。

确定上半部分的初始点  $P_0^u$  后，可以得到初始搜索方向  $\vec{v}_0^u$ ，然后以点  $P_0^u$  为球心不断扩大搜索半径，直到找到下一个除集合中心点外最近的点，同时还满足条件： $\langle \vec{v}_0^u, \vec{v}_1^u \rangle < 90^\circ$ ，即可确定为上半部分的第二个点  $P_1^u$ ，如图5.16 (c) 所示。

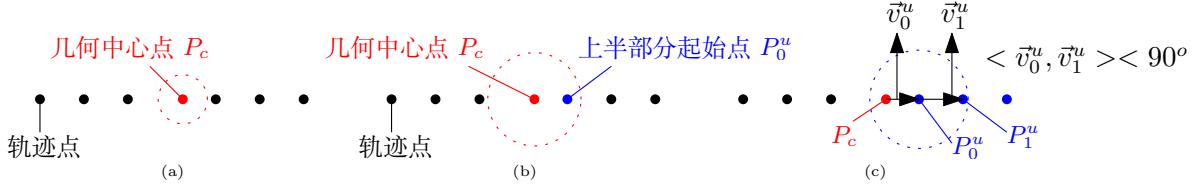


图 5.16: 几何中心点到上半部分方向确定

以此类推，将上一个点  $P_{j-1}^u$  指向当前点  $P_j^u$  的方向向量作为排序的局部方向，在当前点的邻域内搜索下一个点  $P_{j+1}^u$ ，从而继续对上半部分的点排序，如图5.17 (a) 所示。针对上述迭代搜索过程，寻找下一个轨迹点的条件定义为：

$$\begin{aligned}\vec{v}_j &= P_j - P_{j-1} \\ \vec{v}_{j+1} &= P_{j+1} - P_j \\ <\vec{v}_j, \vec{v}_{j+1}> &< 90^\circ\end{aligned}\quad (5.35)$$

直到搜索到轨迹点云上半部分的最后一个点时，如图5.17 (b) 所示。当最后一个轨迹点的邻域搜索半径超过限定值 ( $r_{max} = 0.01m$ )，而且任何一个邻域点都不满足条件5.33，那么终止搜索迭代，轨迹点云的上半部分已排序完毕。

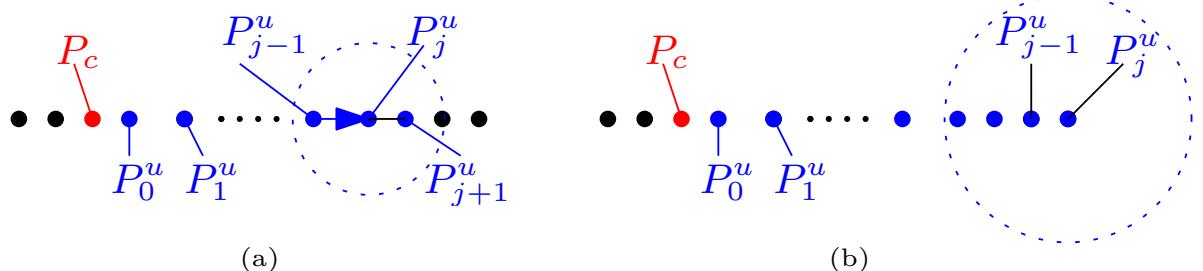


图 5.17: 上半部分继续排序

对于轨迹点云的下半部分也是使用同样的最近邻搜索方法，将局部方向上最近的点作为下一个排序点，既然已知上半部分的起始点  $P_0^u$ ，那么可以确定下半部分的起始点  $P_0^d$  在方向向量： $\vec{v}_0^d = P_c - P_0^u$  的延长线上，而且在此方向上  $P_0^d$  与  $P_c$  的欧式距离是最短的，原理如图5.18 (a) 所示。

对于下半部分的任一轨迹点，仍然使用条件5.35来确定下一个排序点，如图5.18 (b) 所示。当搜索到下半部分的最后一个轨迹点时，和上半部分的最后一个轨迹点的判断情况完全一致，那么说明轨迹点云下半部分排序完毕。

最后将轨迹点云的上半部分和下半部分整合，如图5.19 (a) 所示，以任一端点为焊接轨迹的起始点，另一端点为终点进行顺序统一，就得到了最终的焊接顺序，如图5.19 (a) (b) 所示。

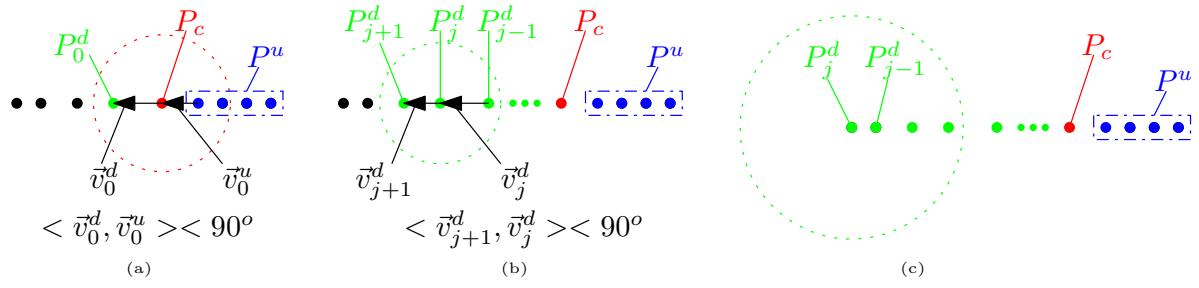


图 5.18: 下半部分排序

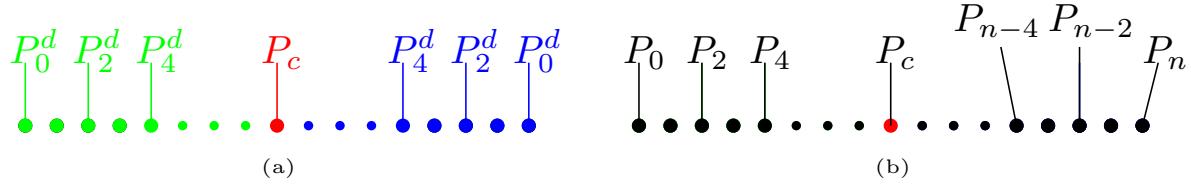


图 5.19: 轨迹点云最终排序

#### 5.2.4 焊接轨迹点的旋转部分

至此，相对于机械臂基座坐标系的焊接轨迹三维位置、焊接移动方向已经确定，还需计算出每个轨迹点的旋转方向才能向机械臂输出最终的六自由度（6-DOF）轨迹。首先确定焊枪的指向方向，也就是焊丝从焊枪头中的伸出方向，根据实际焊接要求，在焊接时该方向须要时刻垂直于焊接缝区域，如图5.20所示。

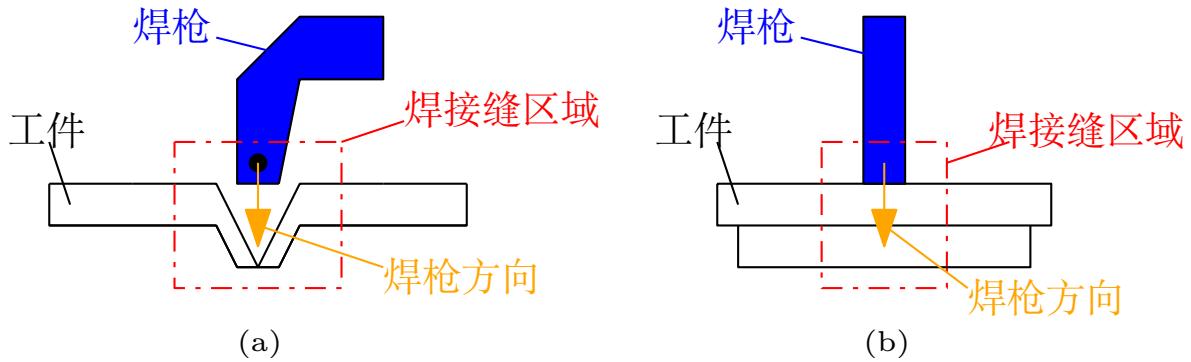


图 5.20: 焊接时的焊枪方向与焊接缝的空间位置关系。(a) 正视图。(b) 侧视图

图5.21展示了焊接缝轨迹与焊接缝点云的可视化关系，要求出单一轨迹点附近焊接缝区域的垂直方向，那么就等价于用该区域点云拟合出一个平面，再用该平面的法向量来代替所求的垂直方向向量，也就是焊枪头的指向方向。

对某一轨迹点分析，仍然用 kd-Tree 最近邻搜索法以半径  $r = 0.012m$  的球来找到该轨迹点邻域内的焊接缝区域点，将此邻域整合为一个局部点云集合  $N_i = \{P_0^t, P_1^t, \dots, P_k^t\}$ （集合内的点数量为  $k$ ）。此时的若干三位空间点拟合一个平面的问题，也就变成了最小二乘法平面拟合估计问题。

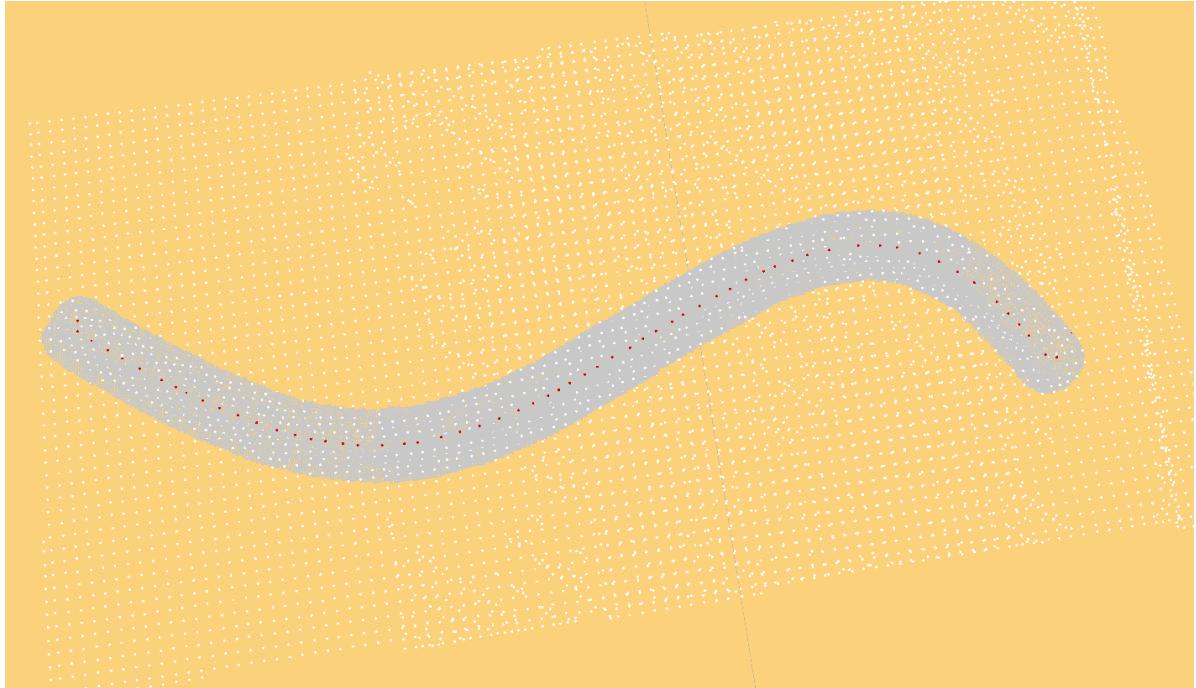


图 5.21: 焊接缝轨迹与焊接工件表面模型点云的可视化关系

从而待拟合的平面方程定义为:

$$\begin{aligned} ax + by + cz = d & \quad (d \geq 0) \\ a^2 + b^2 + c^2 = 1 \end{aligned} \tag{5.36}$$

邻域内任一点到该平面的距离为:

$$d_j = |ax_j + by_j + cz_j - d| \tag{5.37}$$

要获得最佳拟合平面，则需要满足:

$$e = \sum_{j=1}^n d_j^2 \rightarrow \min \tag{5.38}$$

因此转化为求优化函数的极值问题:

$$f = \sum_{j=1}^n d_j^2 - \lambda(a^2 + b^2 + c^2 - 1) \tag{5.39}$$

对  $d$  求偏导:

$$\begin{aligned} \frac{\partial f}{\partial d} &= -2 \sum_{j=1}^n (ax_j + by_j + cz_j - d) = 0 \\ d &= \frac{\sum_{j=1}^n x_j}{n} a + \frac{\sum_{j=1}^n y_j}{n} b + \frac{\sum_{j=1}^n z_j}{n} c \end{aligned} \tag{5.40}$$

将  $d$  代入任意点到平面的距离公式：

$$\begin{aligned} d_j &= \left| a\left(x_j - \frac{\sum_{j=1}^n x_j}{n}\right) + b\left(y_j - \frac{\sum_{j=1}^n y_j}{n}\right) + c\left(z_j - \frac{\sum_{j=1}^n z_j}{n}\right) \right| \\ &= |a(x_j - \bar{x}) + b(y_j - \bar{y}) + c(z_j - \bar{z})| \end{aligned} \quad (5.41)$$

继续对  $a$  求偏导：

$$\frac{\partial f}{\partial a} = 2 \sum_{j=1}^n (a(x_j - \bar{x}) + b(y_j - \bar{y}) + c(z_j - \bar{z})) - 2\lambda a = 0 \quad (5.42)$$

令  $\Delta x_j = x_j - \bar{x}$ ,  $\Delta y_j = y_j - \bar{y}$ ,  $\Delta z_j = z_j - \bar{z}$ , 则：

$$\frac{\partial f}{\partial a} = 2 \sum_{j=1}^n (a\Delta x_j + b\Delta y_j + c\Delta z_j) \Delta x_j - 2\lambda a = 0 \quad (5.43)$$

同理对  $b$ ,  $c$  求偏导：

$$\begin{aligned} \frac{\partial f}{\partial b} &= 2 \sum_{j=1}^n (a\Delta x_j + b\Delta y_j + c\Delta z_j) \Delta y_j - 2\lambda b = 0 \\ \frac{\partial f}{\partial c} &= 2 \sum_{j=1}^n (a\Delta x_j + b\Delta y_j + c\Delta z_j) \Delta z_j - 2\lambda c = 0 \end{aligned} \quad (5.44)$$

将上述三式统一成：

$$\begin{pmatrix} \sum \Delta x_j \Delta x_j & \sum \Delta y_j \Delta y_j & \sum \Delta z_j \Delta z_j \\ \sum \Delta x_j \Delta y_j & \sum \Delta y_j \Delta y_j & \sum \Delta z_j \Delta z_j \\ \sum \Delta x_j \Delta z_j & \sum \Delta y_j \Delta z_j & \sum \Delta z_j \Delta z_j \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \lambda \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (5.45)$$

简化成：

$$Ax = \lambda x \quad (5.46)$$

即转化到了求解矩阵  $A$  的特征值与特征向量的问题，矩阵  $A$  即为  $n$  个点的协方差矩阵。 $(a, b, c)^T$  即为该矩阵的一个特征向量。如上所示，求得的特征向量可能不止一个，应该选择一个特征向量，使得求得法向量为最佳拟合平面的法向量，由：

$$a^2 + b^2 + c^2 = 1 \Rightarrow (x, x) = 1$$

$$\begin{aligned} Ax = \lambda x &\Rightarrow (Ax, x) = (\lambda x, x) \Rightarrow \lambda = (Ax, x) \\ &\Rightarrow \lambda = \sum_{j=1}^n (a\Delta x_j + b\Delta y_j + c\Delta z_j)^2 \\ &\Rightarrow \lambda = \sum_{j=1}^n d_j^2 \end{aligned} \quad (5.47)$$

由  $e = \sum_{j=1}^n d_j^2 \rightarrow \min$ ,  $\lambda \rightarrow \min$ , 因此，最小特征值对应的特征向量即为所求平面的法向量，也就是焊枪的指向方向向量。

然后再确定焊枪在每个点的移动方向。在小节5.2.3中讨论的，已经将所有轨迹点排好序，因此对于某个轨迹点来说，该点的焊枪移动方向向量就定义为：

$$\vec{m}_i = P_i - P_{i-1} \quad (5.48)$$

通过之前的计算，已知该点的焊枪指向方向向量： $\vec{t}_i$ ，而且在本系统中，所有坐标系都统一为右手坐标系，因此，该点的焊枪冗余方向  $\vec{r}_i$  定义为：

$$\vec{r}_i = \vec{m}_i \times \vec{t}_i \quad (5.49)$$

再求这三个两两垂直的方向向量的单位向量：

$$\begin{aligned} \vec{e}_x^g &= \frac{\vec{r}_i}{|\vec{r}_i|} \\ \vec{e}_y^g &= \frac{\vec{m}_i}{|\vec{m}_i|} \\ \vec{e}_z^g &= \frac{\vec{t}_i}{|\vec{t}_i|} \end{aligned} \quad (5.50)$$

于是上式5.50中的三个单位向量就构成了一组正交单位向量，将其作为焊接轨迹点的旋转坐标系  $(\vec{e}_x^g, \vec{e}_y^g, \vec{e}_z^g)^T$ ，并以此来求解焊枪的旋转方向。图5.22显示了机械臂基座坐标系到焊枪坐标系的变换关系。

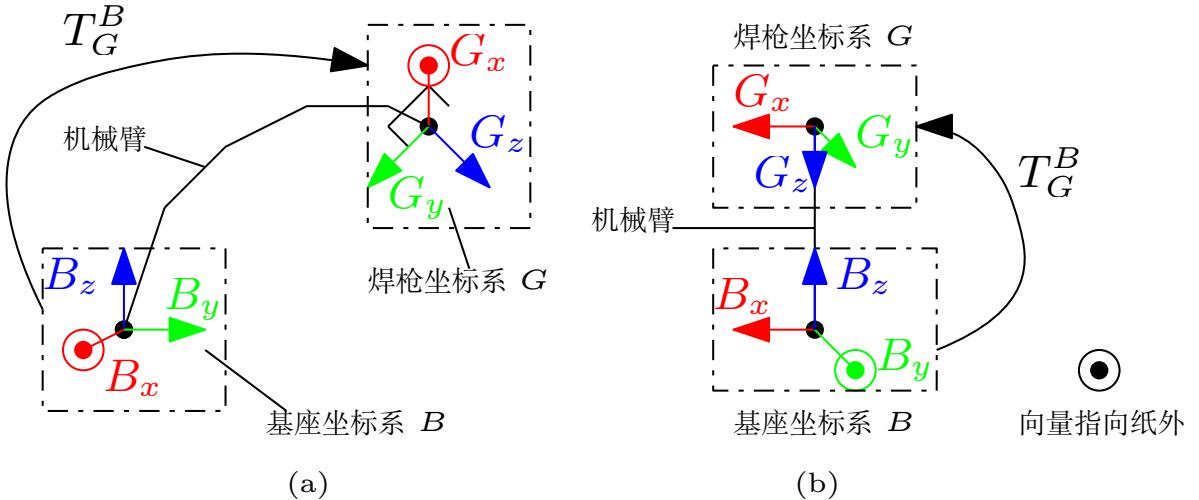


图 5.22: 机械臂基座坐标系到焊枪坐标系的变换关系

要求出焊枪的旋转姿态，实际上就是求机械臂基座坐标系到焊枪坐标系的旋转矩阵，再将该旋转矩阵转化为四元数（在本机器人系统中，机械臂的旋转信息输入格式必须为四元数）。

欧式变换的定义为：同一个向量在不同坐标系下的长度和夹角不会改变。空间中点的坐标系变换是一个欧式变换，一般由一个旋转和一个平移两部分组成。当前只考虑坐标系旋转的情况下，

假设一个点（也叫做一个向量）在上述两个坐标系  $(\vec{e}_x^b, \vec{e}_y^b, \vec{e}_z^b)^T$  和  $(\vec{e}_x^g, \vec{e}_y^g, \vec{e}_z^g)^T$  中的坐标分别为  $[a_x, a_y, a_z]^T$  和  $[a'_x, a'_y, a'_z]^T$ ，则有：

$$\begin{bmatrix} \vec{e}_x^b & \vec{e}_y^b & \vec{e}_z^b \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \vec{e}_x^g & \vec{e}_y^g & \vec{e}_z^g \end{bmatrix} \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix}$$

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \vec{e}_x^b T \vec{e}_x^g & \vec{e}_x^b T \vec{e}_y^g & \vec{e}_x^b T \vec{e}_z^g \\ \vec{e}_y^b T \vec{e}_x^g & \vec{e}_y^b T \vec{e}_y^g & \vec{e}_y^b T \vec{e}_z^g \\ \vec{e}_z^b T \vec{e}_x^g & \vec{e}_z^b T \vec{e}_y^g & \vec{e}_z^b T \vec{e}_z^g \end{bmatrix} \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} = Ra'$$
(5.51)

矩阵 R 由两组基之间的内积组成，刻画了坐标系旋转前后同一个向量的坐标变换关系。只要是同一个旋转，那么矩阵 R 也是一样的，因此矩阵 R 描述了旋转本身，称为旋转矩阵：

$$R = \begin{bmatrix} \vec{e}_x^b T \vec{e}_x^g & \vec{e}_x^b T \vec{e}_y^g & \vec{e}_x^b T \vec{e}_z^g \\ \vec{e}_y^b T \vec{e}_x^g & \vec{e}_y^b T \vec{e}_y^g & \vec{e}_y^b T \vec{e}_z^g \\ \vec{e}_z^b T \vec{e}_x^g & \vec{e}_z^b T \vec{e}_y^g & \vec{e}_z^b T \vec{e}_z^g \end{bmatrix}$$
(5.52)

该矩阵是一个行列式为 1 的正交矩阵： $RR^T = E(I)$ ,  $\det(R) = 1$ 。由旋转矩阵到四元数的转换公式：

$$\begin{aligned} q_w &= \frac{\sqrt{1 + r_{11} + r_{22} + r_{33}}}{2} \\ q_x &= \frac{r_{32} - r_{23}}{4q_w} \\ q_y &= \frac{r_{13} - r_{31}}{4q_w} \\ q_z &= \frac{r_{21} - r_{12}}{4q_w} \end{aligned}$$
(5.53)

其中  $r_{11} = \vec{e}_x^b T \vec{e}_x^g$ ,  $r_{22} = \vec{e}_y^b T \vec{e}_y^g$ ,  $r_{33} = \vec{e}_z^b T \vec{e}_z^g$ ,  $r_{32} = \vec{e}_z^b T \vec{e}_y^g$ ,  $r_{23} = \vec{e}_y^b T \vec{e}_z^g$ ,  $r_{13} = \vec{e}_x^b T \vec{e}_z^g$ ,  $r_{31} = \vec{e}_z^b T \vec{e}_x^g$ ,  $r_{21} = \vec{e}_y^b T \vec{e}_x^g$ ,  $r_{12} = \vec{e}_x^b T \vec{e}_y^g$

### 5.2.5 六自由度轨迹插值

得到焊接轨迹主要点的三维位置和三维旋转信息，即六自由度姿态后，该轨迹只是由若干个离散的点组成，而焊接是一个连续恒定运动过程，因此还需要对每两个轨迹主点之间插值，使整条轨迹近似为一条空间连续曲线。首先对轨迹点之间三维位置线性插值，原理如图5.23所示。

给每两个轨迹主点之间插入欧式距离相等 ( $d = 0.1mm$ ) 的插值点，那么插值点的个数为：

$$P_{num}^t = \frac{|P_{i+1}^t - P_i^t|}{d}$$
(5.54)

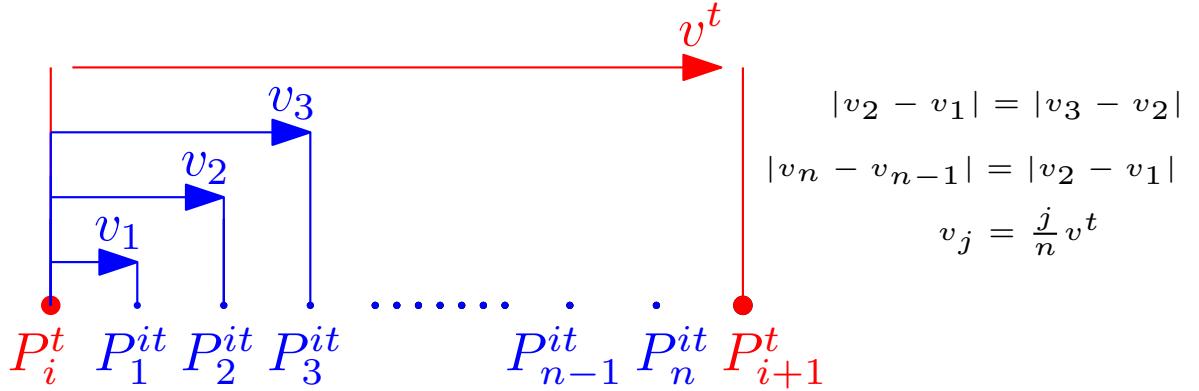


图 5.23: 三维轨迹位置插值

将插值方向定义为两轨迹主点间的方向向量，于是第  $j$  个插值点为：

$$v^t = P_{i+1}^t - P_i^t, \quad P_j^{it} = P_i^t + j \cdot \frac{v^t}{P_{num}} \quad (5.55)$$

用此三维位置线性插值方法遍历所有的轨迹点云，就得到了点距离为  $d = 0.1mm$  的连续轨迹。插值方法处理轨迹点云后，还需要为每个插值点定义旋转方向，这就涉及了四元数插值法。因为已知每个轨迹主点的旋转四元数，而四元数可以看成是四维空间里的一个单位向量，对于一般线性插值来说： $\vec{r} = P_0 + (P_1 - P_0)t$ ,  $t \in [0, 1]$ ，代表了插值矢量  $\vec{r}$  端在弦  $P_0 - P_1$  上的位置。如果  $t$  取值  $\frac{1}{4}$ 、 $\frac{2}{4}$ 、 $\frac{3}{4}$  将弦  $P_0P_1$  弦长均分为 4 等份，可以看出其对应的弧长并不相等。靠近中间位置的弧长较长，而靠近两端处的弧长较短，这就意味着当  $t$  匀速变化时，代表姿态矢量的角速度变化并不均匀，如图5.24 (a) 所示。

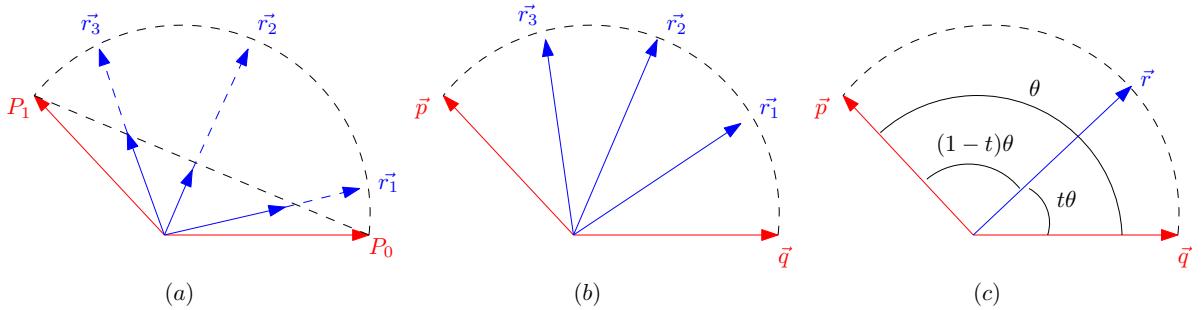


图 5.24: 三维旋转插值

为了更好地解决非恒定转速和归一化问题，我们需要一种称为球面线性插值（通常缩写为 slerp）的插值方法。Slerp 类似于线性插值，只是我们不是沿着直线插值，而是沿着球面上的弧插值。图5.24 (b) 显示了所需的结果。当使用四分之一间距为  $t$  的球面插值时，我们移动弧长的四分之一，从一个方向到另一个方向。我们还可以将 slerp 看作是沿角度插值，或者在这种情况下，将方向之间的角度划分为四分之一的间隔。球面线性插值 (Spherical linear interpolation, 通常简称 Slerp)，是四元数的一种线性插值运算，主要用于在两个表示旋转的四元数之间平滑差值。插

值的一般公式可以写为：

$$\vec{r} = a(t)\vec{p} + b(t)\vec{q} \quad (5.56)$$

现在要找到合适的  $a(t)$  和  $b(t)$  来完成旋转插值。注意单位向量  $\vec{p}$  和  $\vec{q}$  之间的夹角为： $\theta$ ,  $\vec{p}$  和  $\vec{r}$  之间的夹角为： $(1-t)\theta$ ,  $\vec{q}$  和  $\vec{r}$  之间的夹角为： $t\theta$ , 如图5.24 (c) 所示。

将式5.60两边点乘  $\vec{p}$  得到：

$$\vec{p} \cdot \vec{r} = a(t)\vec{p} \cdot \vec{p} + b(t)\vec{p} \cdot \vec{q}, \quad \cos t\theta = a(t) + b(t) \cos \theta \quad (5.57)$$

将式5.60两边点乘  $\vec{q}$  得到：

$$\vec{q} \cdot \vec{r} = a(t)\vec{q} \cdot \vec{p} + b(t)\vec{q} \cdot \vec{q}, \quad \cos[(1-t)\theta] = a(t) \cos \theta + b(t) \quad (5.58)$$

两个方程可以解出两个未知量  $a(t)$  和  $b(t)$ :

$$a(t) = \frac{\cos t\theta - \cos[(1-t)\theta] \cos \theta}{1 - \cos^2 \theta}, \quad b(t) = \frac{\cos[(1-t)\theta] - \cos t\theta \cos \theta}{1 - \cos^2 \theta} \quad (5.59)$$

使用三角函数公式可以将其化简为：

$$a(t) = \frac{\sin[(1-t)\theta]}{\sin \theta}, \quad b(t) = \frac{\sin t\theta}{\sin \theta} \quad (5.60)$$

于是四元数的球面线性插值公式为：

$$Slerp(\vec{p}, \vec{q}, t) = \frac{\sin[(1-t)\theta]\vec{p} + \sin t\theta\vec{q}}{\sin \theta} \quad (5.61)$$

用5.61式对每个插值点进行旋转插值并可视化如图5.25所示。最终就得到了有序、连续且分布均匀的六自由度焊接轨迹。

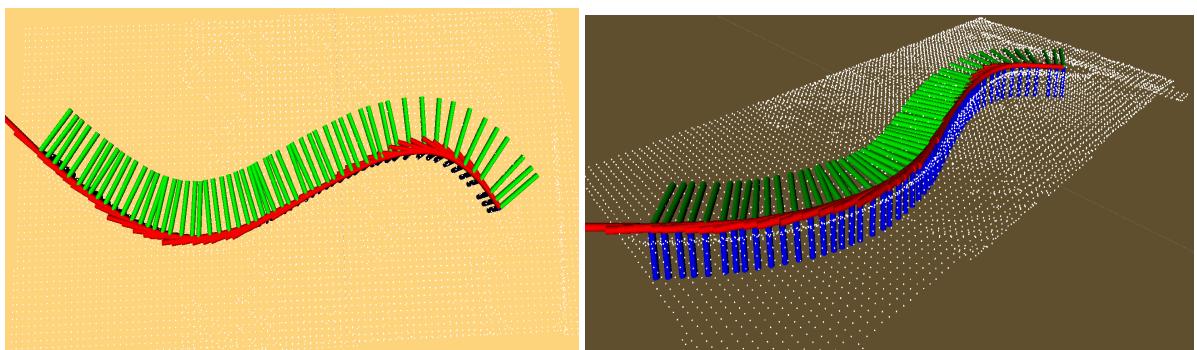


图 5.25: 曲线型焊接工件的六自由度焊接轨迹

## 6 机器人焊接实验部分

本系统对五类不同的焊接工件6.1进行了感知、轨迹规划和运动控制等实验测试，对每个实验目标测试了以下指标：

- 耗时：系统从输入原始点云模型到生成最终六自由度焊接轨迹的运行耗时，用以评价系统执行效率。
- 精度：焊接缝定位算法在点云模型上识别出来焊接缝区域与标准焊接缝区域的空间重叠率，用以衡量定位精度。
- 误差：六自由度轨迹规划算法计算出的焊接轨迹与标准焊接轨迹的误差，作为机器人运动控制的参考。

上述三项指标切实贴合实际工业生产的要求，引言部分1已经讨论过了传统的机器人焊接情况，因而系统运行耗时能够说明本系统在工作效率上的优势。对于焊接来说，最重要的指标是焊枪移动的精度，这直接决定了焊接质量，而且焊接是一项危险的工程作业活动，如果焊枪的轨迹与实际要求的焊接轨迹误差太大，会造成机器设备、焊接工件的损坏甚至是操作人员人身伤害。



图 6.1: 从左至右分别为：直线型、曲线型、圆柱体型、长方体型和 Y 型焊接工件

### 6.1 机器人系统运行流程

本系统功能模块较多，在实验测试及分析前需要详细介绍整个工作流程：传感器-机器人标定、视觉引导控制机械臂、焊接工件三维模型重建、六自由度轨迹规划和机械臂轨迹跟踪控制，这五个模块构成了一套完整又衔接紧密的机器人系统执行逻辑。下文对每个焊接工件的实验测试都是遵循此运行逻辑，因此后面不再赘述。

图6.7展示了以图中长方体型焊接工件为例的实验过程。

## 6.2 传感器精度测试

### 6.2.1 Kinect-V2 测量精度测试

Kinect 相机的主要作用是用来视觉引导控制机器人移动到其工作范围内的某个位姿，这种方式改进了人为手动控制机器人来对焊接工件进行三维扫描，提高了系统自动化集成度。为了衡量该功能模块的精度指标，作者记录了机器人工作范围内可解算的若干组三维位置数据，通过对比测量值与实际真实值，计算所有组的均方根误差，从而分析出该传感器的测量精度。图6.2显示了八组视觉引导机器人的位姿控制结果。

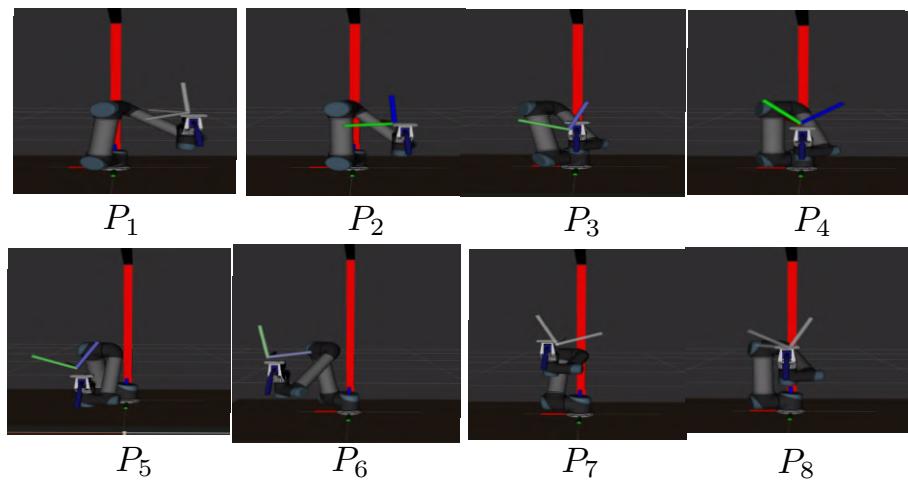


图 6.2: 视觉引导机器人的位姿控制结果

同时记录下了每个位姿的三维位置信息，即测量数据和真实数据，测量数据由 Kinect 相机读出并转换到机器人坐标系下，真实数据通过机器人自身正运动学原理读出（误差  $< 0.1mm$ ），如表6.1所示。

表 6.1: Kinect 视觉引导测量数据和真实数据

测量数据 (m) /m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$
$x_m$	0.285	0.168	-0.013	0.014	-0.202	-0.324	-0.117	0.051
$y_m$	-0.350	-0.239	-0.277	-0.373	-0.422	-0.323	-0.236	-0.368
$z_m$	0.379	0.284	0.313	0.303	0.210	0.358	0.507	0.327
真实数据 (g) /m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$
$x_g$	0.278	0.173	0.006	-0.01	-0.224	-0.320	-0.105	0.045
$y_g$	-0.375	-0.257	-0.267	-0.359	-0.435	-0.344	-0.258	-0.353
$z_g$	0.391	0.321	0.335	0.318	0.192	0.323	0.486	0.338

### 6.2.2 RealsenseD415 深度测量精度测试

本系统的三维模型重建功能是通过 Realsense 相机来实现的，而且重建的精度直接影响到最终轨迹生成的精度，因此 Realsense 相机的深度测量精度要求非常高。为了方便测量该相机深度，在实验之前先控制机械臂让相机镜头垂直向下面对平台（平台默认水平），目的是尽可能保证深度图的每个点代表的深度一致。

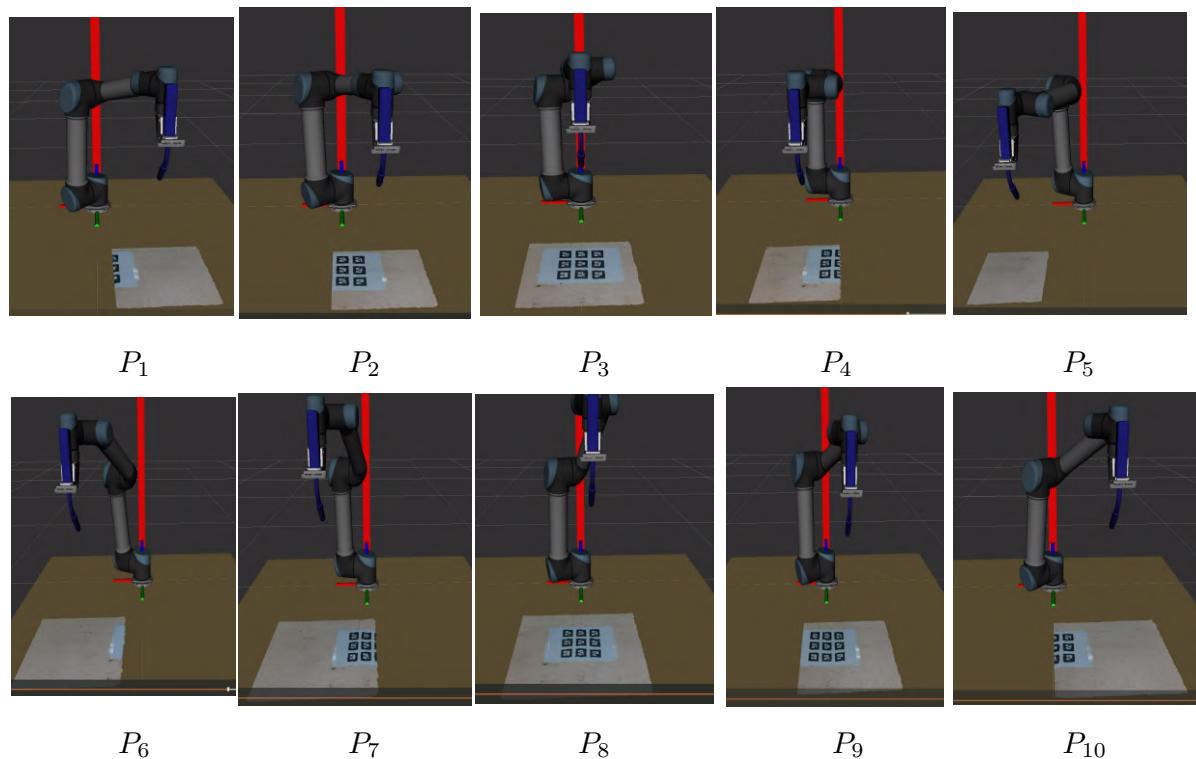


图 6.3: RealsenseD415 深度测量时的机器人位姿

同样地，作者随机选择了十组不同的机器人位姿来获取深度图测量数据，如图6.3所示。对于每个位姿，测量数据用深度平均值代替整个深度图的深度值程度，真实深度数据由机械臂正运动学方程和实验之前手眼标定好的变换矩阵获得，如表6.2所示。

表 6.2: RealsenseD415 深度测量数据和真实数据

测量数据 (m) /m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$
$D_m$	0.430	0.402	0.462	0.394	0.358	0.522	0.559	0.601	0.497	0.531
真实数据 (g) /m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$
$D_g$	0.428	0.408	0.457	0.391	0.360	0.519	0.560	0.605	0.501	0.530

## 6.3 实际焊接工件实验

### 6.3.1 圆柱体型焊接工件测试

圆柱体型焊接工件也是实际生产中常见类型之一，图6.4展示了以该类型工件的机器人轨迹跟踪控制效果。表6.3记录了十组实验的焊接缝定位及轨迹规划算法的处理时间，并计算了平均耗时。

表 6.3: 圆柱体型焊接工件: 算法处理时间

算法处理耗时/s	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$\bar{t}$
$t$	40.1	40.5	41.0	39.4	40.6	39.2	39.0	40.3	40.5	40.9	40.15

其中任意一组算法轨迹测试实验的数据结果如表6.4所示。

表 6.4: 圆柱体型焊接工件: 算法输出的轨迹误差

测量数据 (m)/m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$x_m$	-0.135	-0.111	-0.066	-0.035	-0.008	0.041	0.089	0.119	0.151
$y_m$	-0.508	-0.511	-0.511	-0.509	-0.512	-0.509	-0.509	-0.513	-0.510
$z_m$	0.185	0.201	0.212	0.216	0.226	0.215	0.208	0.196	0.187
真实数据 (g)/m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$x_g$	-0.133	-0.106	-0.069	-0.034	-0.003	0.042	0.093	0.126	0.147
$y_g$	-0.510	-0.511	-0.510	-0.509	-0.510	-0.508	-0.510	-0.512	-0.510
$z_g$	0.190	0.198	0.214	0.220	0.223	0.218	0.207	0.193	0.186

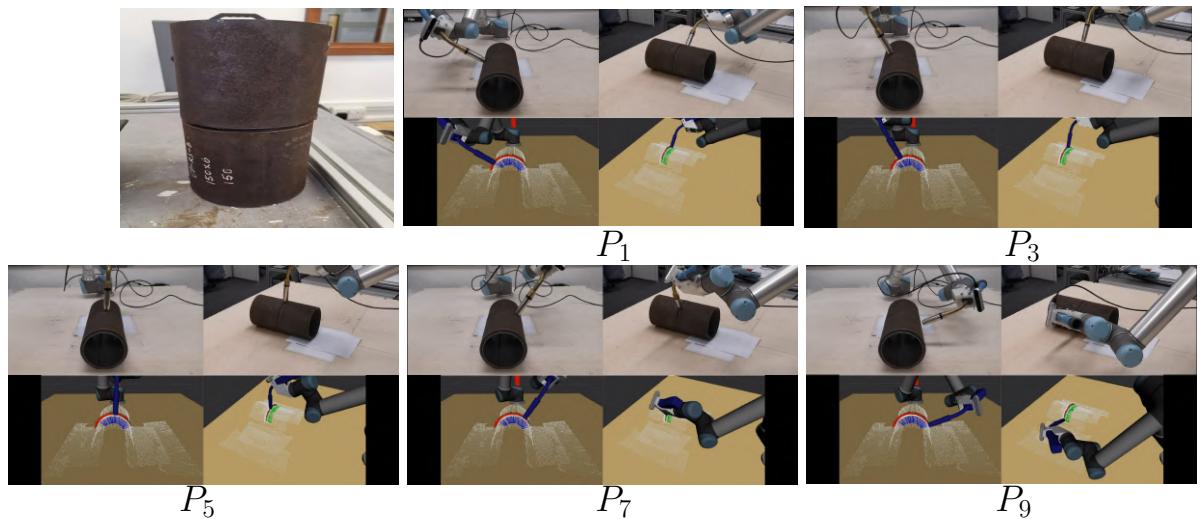


图 6.4: 圆柱体型测试: 每个位姿图的上部分展示系统实际运动情况, 下部分展示仿真环境中机器人与点云交互

### 6.3.2 长方体型焊接工件测试

长方体型焊接工件是建筑业生产作业的常见类型之一，图6.5展示了以该类型工件的机器人轨迹跟踪控制效果。表6.5记录了十组实验的焊接缝定位及轨迹规划算法的处理时间，并计算了平均耗时。

表 6.5: 长方体型焊接工件: 算法处理时间

算法处理耗时/s	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$\bar{t}$
$t$	61.4	62.5	60.7	61.3	62.0	59.6	60.2	60.8	62.1	60.5	61.11

其中任意一组算法轨迹测试实验的数据结果如表6.6所示。

表 6.6: 长方体型焊接工件: 算法输出的轨迹误差

测量数据 (m)/m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$x_m$	0.207	0.208	0.208	0.193	-0.0	-0.189	-0.211	-0.211	-0.209
$y_m$	-0.579	-0.581	-0.582	-0.581	-0.580	-0.583	-0.587	-0.586	-0.586
$z_m$	0.100	0.217	0.310	0.418	0.420	0.422	0.312	0.215	0.101
真实数据 (g)/m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$x_g$	0.211	0.210	0.210	0.183	-0.002	-0.191	-0.215	-0.211	-0.210
$y_g$	-0.585	-0.586	-0.585	-0.582	-0.585	-0.584	-0.585	-0.585	-0.586
$z_g$	0.091	0.219	0.303	0.417	0.421	0.420	0.315	0.211	0.102

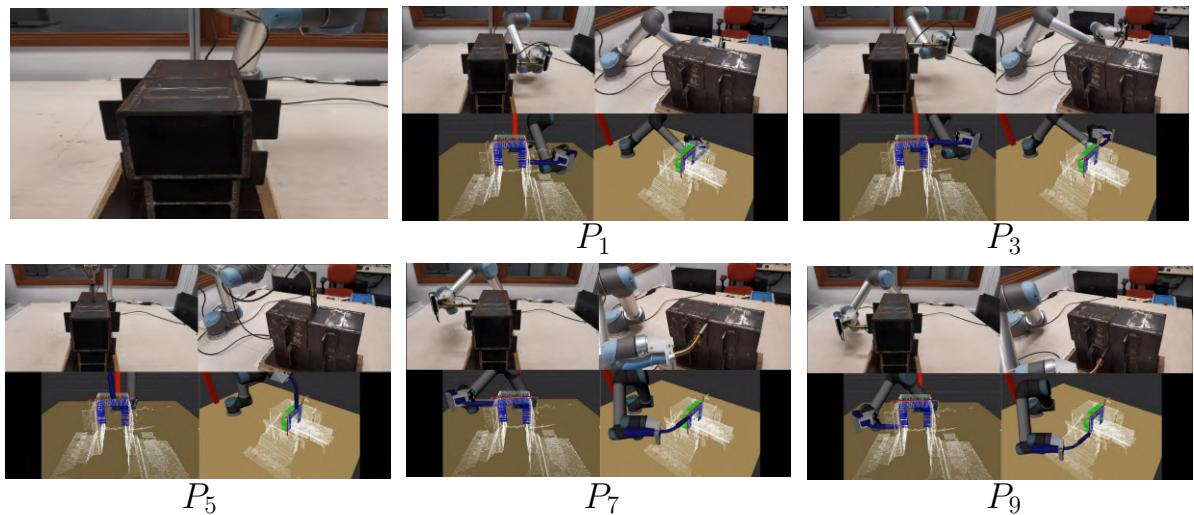


图 6.5: 长方体型测试: 每个位姿图的上部分展示系统实际运动情况, 下部分展示仿真环境中机器人与点云交互

### 6.3.3 Y-型焊接工件测试

Y-型焊接工件是为了测试系统对于复杂形状的焊接工件的鲁棒性，图6.6展示了以该类型工件的机器人轨迹跟踪控制效果。表6.7记录了十组实验的焊接缝定位及轨迹规划算法的处理时间，并计算了平均耗时。

表 6.7: Y-型焊接工件：算法处理时间

算法处理耗时/s	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$\bar{t}$
$t$	25.1	26.5	26.0	25.4	24.6	25.2	25.0	26.3	26.5	25.9	25.65

其中任意一组算法轨迹测试实验的数据结果如表6.8所示。

表 6.8: Y-型焊接工件：算法输出的轨迹误差

测量数据 (m)/m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$x_m$	0.005	-0.003	0	0.001	0.002	-0.003	-0.0	0.003	0.001
$y_m$	-0.630	-0.618	-0.595	-0.581	-0.560	-0.547	-0.531	-0.526	-0.510
$z_m$	0.390	0.383	0.377	0.362	0.349	0.340	0.331	0.320	0.298
真实数据 (g)/m	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$
$x_g$	0.001	0.002	-0.001	0	-0.001	-0.001	-0.0	0.001	-0.001
$y_g$	-0.623	-0.612	-0.598	-0.586	-0.564	-0.553	-0.537	-0.529	-0.512
$z_g$	0.391	0.386	0.380	0.367	0.353	0.341	0.332	0.317	0.303

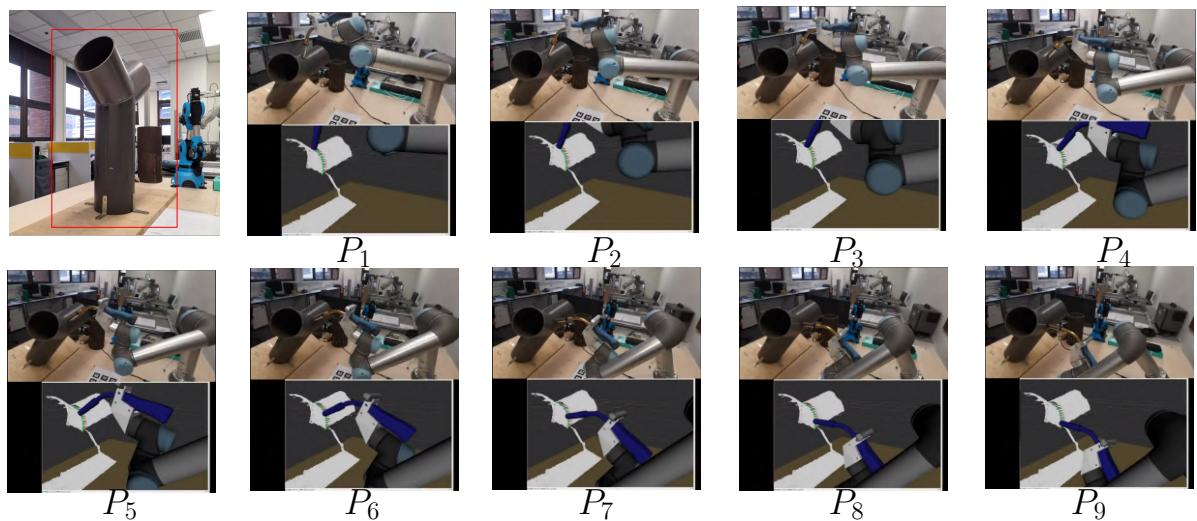


图 6.6: Y-型工件测试：每个位姿图的上部分展示系统实际运动情况，下部分展示仿真环境中机器人与点云交互

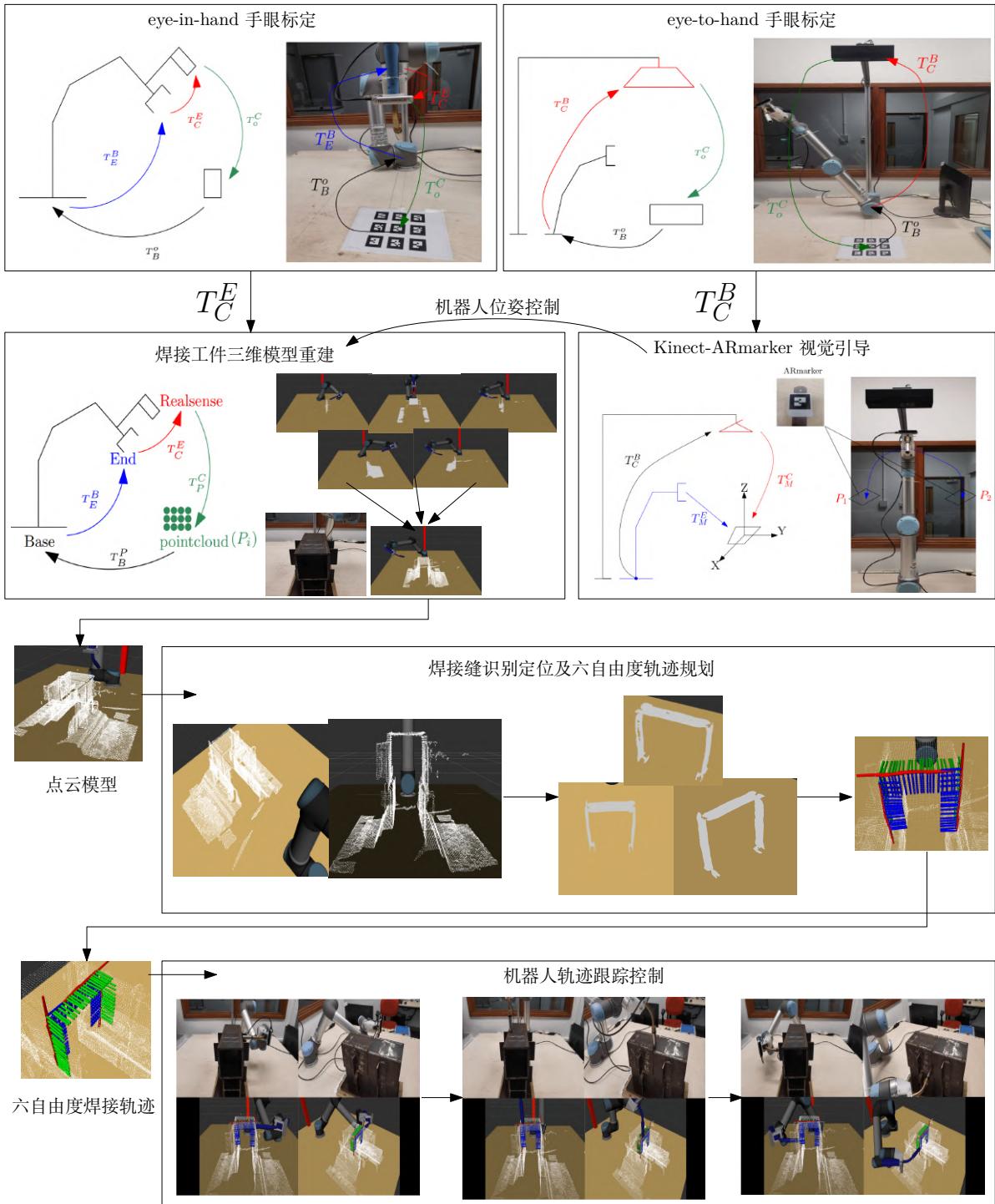


图 6.7: 系统工作流程: 首先对两个三维相机和机械臂进行坐标系变换矩阵标定, 分别得到从各自相机坐标系到机械臂坐标系的变换矩阵; 对于 Kinect 相机而言, 通过 ARmarker 在其视野范围内的移动来传递该 marker 在机械臂坐标系下的位置, 用以引导机械臂末端移动到指示的位置。对于 RealSense 而言, 得到相应的变换矩阵后能够在不同位姿将自身坐标系下的点云变换到机械臂坐标系下; 获取完整的焊接工件表面点云模型后, 将其输入到焊接缝识别定位算法以及后续的六自由度轨迹规划算法中处理, 输出目标焊接轨迹; 机器人的轨迹跟踪控制线程接收到焊接轨迹后则执行相应的轨迹控制运动

## 7 总结与展望

### 7.1 总结

本工作针对当前焊接工业领域的迫切需求，以及基于传感器辅助的焊接工业机器人的研究现状，设计了一套基于三维视觉的智能焊接机器人系统，在此领域具有很高的原创性和系统完整性，其系统有两大理论算法创新点：

- 基于焊接目标物体的三维模型点云的识别定位焊接缝算法。
- 基于焊接缝点云的自动化轨迹规划算法。

算法建立在对三维点云的处理上，空间信息处理能力大大强于当前学术界内基于 RGB 平面图像的工作。实验部分通过对若干个常见的立体焊接工件进行测试，本系统能够在少量的人为辅助操作下，并在短时间内利用上述算法对工件模型计算处理，输出完整的六自由度焊接轨迹，最终由机器人完成跟踪控制任务。

本工作对实际应用的贡献在于大大减少了机械臂焊接轨迹规划的时间，在保证了一定轨迹精度 ( $<2\text{mm}$ ) 的情况下提高了焊接效率。而且系统软硬件结构轻便，均已开源，能够向工业界推广。

但是本系统还存在一些技术问题：

- 每次上电后都需要运行传感器-机器人坐标系变换矩阵标定，否则会影响后续的精度。
- RealsenseD415 相机在扫描焊接工件表面时会因为机械臂抖动而产生不必要的噪声。
- 机械臂反动力学算法无法解算出轨迹规划算法输出的某些位姿。

### 7.2 展望

本工作已经完成了自动工业焊接机器人的所有系统功能模块，在未来会从以下几个方面来改进现在的系统：

- 引入点云深度学习方法来提高焊接缝识别定位算法精度。
- 六自由度轨迹规划算法加入对周围环境中障碍物的规划检测。
- 将本系统用于实际工业焊接测试，进一步迭代，期望将其做成可靠的工业成品。

## 参考文献

- [1] Chang, Doyoung Son, Donghoon Lee, Jungwoo Lee, Donghun Kim, Tae-wan Lee, Kyu-Yeul Kim, Jongwon. (2012). A new seam-tracking algorithm through characteristic-point detection for a portable welding robot. *Robotics and Computer-Integrated Manufacturing*. 28. 1–13. 10.1016/j.rcim.2011.06.001.
- [2] Ahmed S M, Tan Y Z, Chew C M, et al. Edge and corner detection for unorganized 3D point clouds with application to robotic welding[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 7350-7355.
- [3] Patil V, Patil I, Kalaichelvi V, et al. Extraction of Weld Seam in 3D Point Clouds for Real Time Welding Using 5 DOF Robotic Arm[C]//2019 5th International Conference on Control, Automation and Robotics (ICCAR). IEEE, 2019: 727-733.
- [4] Ahmed S M, Tan Y Z, Lee G H, et al. Object detection and motion planning for automated welding of tubular joints[C]//2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 2610-2615.
- [5] Jing L, Fengshui J, En L. RGB-D sensor-based auto path generation method for arc welding robot[C]//2016 Chinese control and decision conference (CCDC). IEEE, 2016: 4390-4395.
- [6] Zeng J, Chang B, Du D, et al. A vision-aided 3D path teaching method before narrow butt joint welding[J]. *Sensors*, 2017, 17(5): 1099.
- [7] Njaastad E B, Egeland O. Automatic Touch-Up of Welding Paths Using 3D Vision[J]. *IFAC-PapersOnLine*, 2016, 49(31): 73-78.
- [8] Liang Z, Gao H, Nie L, et al. 3D reconstruction for telerobotic welding[C]//2007 International Conference on Mechatronics and Automation. IEEE, 2007: 475-479.
- [9] Zhang K, Yan M, Huang T, et al. 3D reconstruction of complex spatial weld seam for autonomous welding by laser structured light scanning[J]. *Journal of Manufacturing Processes*, 2019, 39: 200-207.
- [10] Yang L, Li E, Long T, et al. A novel 3-D path extraction method for arc welding robot based on stereo structured light sensor[J]. *IEEE Sensors Journal*, 2018, 19(2): 763-773.
- [11] Manorathna R P, Phairatt P, Ogun P, et al. Feature extraction and tracking of a weld joint for adaptive robotic welding[C]. 2014 13th International Conference on Control Automation

- Robotics Vision (ICARCV). IEEE, 2014: 1368-1372.
- [12] Liu Z, Bu W, Tan J. Motion navigation for arc welding robots based on feature mapping in a simulation environment[J]. Robotics and Computer-Integrated Manufacturing, 2010, 26(2): 137-144.
- [13] Yang B, Luo W, Urtasun R. Pixor: Real-time 3d object detection from point clouds[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018: 7652-7660.
- [14] Pham N H, La H M, Ha Q P, et al. Visual and 3D mapping for steel bridge inspection using a climbing robot[C]//ISARC 2016-33rd International Symposium on Automation and Robotics in Construction. 2016.
- [15] Pham N H, La H M, Ha Q P, et al. Visual and 3D mapping for steel bridge inspection using a climbing robot[C]//ISARC 2016-33rd International Symposium on Automation and Robotics in Construction. 2016.
- [16] Dinham M, Fang G. Autonomous weld seam identification and localisation using eye-in-hand stereo vision for robotic arc welding[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(5): 288-301.
- [17] Dinham M, Fang G. Detection of fillet weld joints using an adaptive line growing algorithm for robotic arc welding[J]. Robotics and Computer-Integrated Manufacturing, 2014, 30(3): 229-243.
- [18] Yang L, Li E, Long T, et al. A high-speed seam extraction method based on the novel structured-light sensor for arc welding robot: A review[J]. IEEE Sensors Journal, 2018, 18(21): 8631-8641.
- [19] Xiao R, Xu Y, Hou Z, et al. An adaptive feature extraction algorithm for multiple typical seam tracking based on vision sensor in robotic arc welding[J]. Sensors and Actuators A: Physical, 2019, 297: 111533.
- [20] Kiddee P, Fang Z, Tan M. An automated weld seam tracking system for thick plate using cross mark structured light[J]. The International Journal of Advanced Manufacturing Technology, 2016, 87(9-12): 3589-3603.
- [21] Li J, Jing F, Li E. A new teaching system for arc welding robots with auxiliary path point generation module[C]//2016 35th Chinese Control Conference (CCC). IEEE, 2016: 6217-6221.
- [22] Nele L, Sarno E, Keshari A. An image acquisition system for real-time seam tracking[J]. The International Journal of Advanced Manufacturing Technology, 2013, 69(9-12): 2099-2110.

- [23] Rodrigues M, Kormann M, Schuhler C, et al. An intelligent real time 3D vision system for robotic welding tasks[C]//2013 9th International Symposium on Mechatronics and its Applications (ISMA). IEEE, 2013: 1-6.
- [24] Ding Y, Huang W, Kovacevic R. An on-line shape-matching weld seam tracking system[J]. Robotics and Computer-Integrated Manufacturing, 2016, 42: 103-112.
- [25] González-Galván E J, Loredo-Flores A, Cervantes-Sánchez J J, et al. An optimal path-generation algorithm for manufacturing of arbitrarily curved surfaces using uncalibrated vision[J]. Robotics and Computer-Integrated Manufacturing, 2008, 24(1): 77-91.
- [26] Shao W J, Huang Y, Zhang Y. A novel weld seam detection method for space weld seam of narrow butt joint in laser welding[J]. Optics Laser Technology, 2018, 99: 39-51.
- [27] Zeng J, Chang B, Du D, et al. A precise visual method for narrow butt detection in specular reflection workpiece welding[J]. Sensors, 2016, 16(9): 1480.
- [28] Wang Z, Zhang K, Chen Y, et al. A real-time weld line detection for derusting wall-climbing robot using dual cameras[J]. Journal of Manufacturing Processes, 2017, 27: 76-86.
- [29] Weis Á A, Mor J L, Soares L B, et al. Automated seam tracking system based on passive monocular vision for automated linear robotic welding process[C]//2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, 2017: 305-310.
- [30] Rajaraman M, Dawson-Haggerty M, Shimada K, et al. Automated workpiece localization for robotic welding[C]//2013 IEEE international conference on automation science and engineering (CASE). IEEE, 2013: 681-686.
- [31] Fan J, Jing F, Fang Z, et al. Automatic recognition system of welding seam type based on SVM method[J]. The International Journal of Advanced Manufacturing Technology, 2017, 92(1-4): 989-999.
- [32] Li X, Li X, Ge S S, et al. Automatic welding seam tracking and identification[J]. IEEE Transactions on Industrial Electronics, 2017, 64(9): 7261-7271.
- [33] De Maeyer J, Moyaers B, Demeester E. Cartesian path planning for arc welding robots: Evaluation of the descartes algorithm[C]//2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2017: 1-8.
- [34] Gonçalves P, Pinto T C. Dual Mode Optical Measurement Sensor for 3D Surface Reconstruction and Crack Detection[C]//Journal of Physics: Conference Series. IOP Publishing, 2018, 1065(17): 172005.
- [35] Lü X, Gu D, Wang Y, et al. Feature extraction of welding seam image based on laser vision[J]. IEEE Sensors Journal, 2018, 18(11): 4715-4724.

- [36] Dalitz C, Schramke T, Jeltsch M. Iterative Hough transform for line detection in 3D point clouds[J]. *Image Processing On Line*, 2017, 7: 184-196.
- [37] Li Y, Li Y F, Wang Q L, et al. Measurement and defect detection of the weld bead based on online vision inspection[J]. *IEEE Transactions on Instrumentation and Measurement*, 2009, 59(7): 1841-1849.
- [38] Li X, Li X, Khyam M O, et al. Robust welding seam tracking and recognition[J]. *IEEE Sensors Journal*, 2017, 17(17): 5609-5617.
- [39] Soares L B, Weis Á A, Rodrigues R N, et al. Seam tracking and welding bead geometry analysis for autonomous welding robot[C]//2017 Latin American robotics symposium (LARS) and 2017 Brazilian symposium on robotics (SBR). IEEE, 2017: 1-6.
- [40] Lü X, Zhang K, Wu Y. The seam position detection and tracking for the mobile welding robot[J]. *The International Journal of Advanced Manufacturing Technology*, 2017, 88(5-8): 2201-2210.
- [41] Masood A, Siddiqui R, Pinto M, et al. Tool path generation, for complex surface machining, using point cloud data[J]. *Procedia CIRP*, 2015, 26: 397-402.
- [42] Rao S H, Kalaichelvi V, Karthikeyan R. Tracing a Weld Line using Artificial Neural Networks[J]. *International Journal of Networked and Distributed Computing*, 2018, 6(4): 216-223.
- [43] Pachidis T P, Lygouras J N. Vision-based path generation method for a robot-based arc welding system[J]. *Journal of Intelligent and Robotic Systems*, 2007, 48(3): 307-331.
- [44] Peng G, Xue B, Gao Y, et al. Vision sensing and surface fitting for real-time detection of tight butt joints[C]//Journal of Physics: Conference Series. IOP Publishing, 2018, 1074(1): 012001.
- [45] Wang X, Zhang X, Ren X, et al. Point cloud 3D parent surface reconstruction and weld seam feature extraction for robotic grinding path planning[J]. *The International Journal of Advanced Manufacturing Technology*, 2020, 107(1): 827-841.
- [46] Muhammad J, Altun H, Abo-Serie E. Welding seam profiling techniques based on active vision sensing for intelligent robotic welding[J]. *The International Journal of Advanced Manufacturing Technology*, 2017, 88(1-4): 127-145.

## 致 谢

感谢中国共产党;  
感谢人民解放军;  
感谢毛主席;  
感谢父母, 家人;  
感谢武汉大学;  
感谢恩师杨文教授;  
感谢陪我一路成长的朋友, 兄弟姐妹。  
以上。



## 武汉大学学位论文使用授权协议书

本学位论文作者愿意遵守武汉大学关于保存、使用学位论文的管理办法及规定，即：学校有权保存学位论文的印刷本和电子版，并提供文献检索与阅览服务；学校可以采用影印、缩印、数字化或其它复制手段保存论文；在以教学与科研服务为目的前提下，学校可以在校园网内公布部分及全部内容。

- 1、 在本论文提交当年，同意在校园网内以及中国高等教育文献保障系统（CALIS）高校学位论文系统提供查询及前十六页浏览服务。
- 2、 在本论文提交  当年 /  一年 /  两年 /  三年 /  五年以后，同意在校园网内允许读者在线浏览并下载全文，学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。（保密论文解密后遵守此规定）

论文作者（签名）：\_\_\_\_\_

学 号：\_\_\_\_\_

学 院：\_\_\_\_\_

日期： 年 月 日

