上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS

论文题目：工业机器人路径规划算法研究

学生姓名：王逸潇

学生学号：5140219196

专　　业：机械工程

指导教师：丁烨

学院(系)：机械与动力工程学院

工业机器人路径规划算法研究

摘要

路径规划作为工业机器人应用的重要环节，目前仍以传统示教方式为主，存在规划时间长、人力成本高等缺点。本文通过 Kinect 传感器获得点云数据，设计基于八叉树的层次包络盒法，构建三维地图信息，并进行 UR10 机械臂正向运动学和逆向运动学求解，在仿真环境中进行机械臂路径规划。针对 UR10 多自由度机器人路径规划问题的高维性，实现并测试已有的快速扩展随机树算法及其改进算法，根据算法优劣，开发出 RRT-CS、RRT-CSD、基于随机采样的 A*算法三种新型路径规划算法。在实现并测试的多维路径规划算法中，RRT-CSD 综合性能最佳，在复杂环境中更具适用性，在保证计算效率的情况下，优化效果显著。针对抓取姿态的多解性，研究不同抓取姿态对路径规划结果的影响，并对其进行优化。优化结果对路径规划效果提升显著。

关键词:，路径规划，随机采样，抓取姿态

STUDY ON INDUSTRIAL ROBOT PATH PLANNING ALGORITHM

ABSTRACT

Path planning, as an important part of industrial robot applications, is still dominated by traditional teaching methods. It takes long planning time and has high labor costs. In this paper, three-dimensional map information is rebuilt by hierarchical envelope box method based on octree method, using point cloud data from Kinect sensor. With the complete analysis of UR10's forward kinematics and inverse kinematics, path planning can be performed in the simulation environment. Traditional path planning algorithms cannot solve the high dimensionality of multi-degree-of-freedom robot path planning problem well. The existing Rapidly-exploring Random Tree algorithm suitable to hyperspace and its improved algorithms are implemented and tested. Based on the merits of those algorithms, RRT-CS, RRT-CSD and A* algorithm based random sampling are developed. RRT-CSD has the best comprehensive performance and is more applicable in complex environments. Ensuring calculation efficiency, the optimization effect of RRT-CSD is significant. For the multi-solution of grab gestures, the influence of different grab gestures on the path planning results is studied and optimized. The result of the optimization to grab gesture can significantly improve the outcome of path planning.

Key words: path planning, random sampling, grasping posture

目 录

第一章 绪论

1.1 研究背景及意义

从 1920 年捷克作家卡雷尔·恰佩克在科幻小说《罗萨姆的机器人万能公司》中第一次提出机器人这一词开始，人们对未来机器人的憧憬不断增强，对机器人的研究不断加深。随着第一台电子数字式计算机于 1946 年正式投入运行，计算机与机器人的结合不断地深入。人工智能——让机器像人一样思考，让机器像人一样认识世界、执行任务，在近几年成为研究热门。人工智能之父马文·明斯基提出的智能机器“能够创建周围环境的抽象模型，如果遇到问题，能够从抽象模型中寻找解决方法”概念，至今仍然影响着现代智能机器人的发展。

据国际机器人联合会统计，截至 2011 年年底，115 万台至 140 万台工业机器人服务于全球的制造加工、装配运输等工业领域，市场价值已百亿美元计。但目前工业机器人智能水平仍然较低，主要通过固定编程，执行重复性任务。其存在的问题主要有：

（1）主要通过示教、试错等方式，进行工业机器人任务规划。规划时间成本、人力成本、经济成本等均较高。

（2）无法应对环境变化的情况。若环境是随机的，其任务根据环境的变化而发生变化，工业机器人难以识别任务并且自行规划并执行任务。

（3）工业机器人存在其智能水平与生产成本之间的矛盾。智能水平越高，则其生产成本越高，在工业领域则越难以推广。

随着工业 4.0 的提出，企业对智能制造需求的丰富，目前的工业机器人渐渐无法满足市场需求，尤其对工业机器人任务规划提出了更高的要求。工业机器人任务规划主要分为路径规划以及特殊任务执行规划。路径规划是指为机器人规划一条从起始位置到达终止位置，尽可能达到某一目标（如路径长度最短、所用时间最短、代价最少等）且避开障碍物的运动路径。特殊任务执行规划主要是指利用机器人操控相关工具完成打磨、运输、装配等特定的工业操作。特殊任务执行规划方法根据工业操作的不同而不同，路径规划方法则是普遍适用于工业机器人。目前工业机器人主要通过示教方式规划路径。机器人示教是指操作机器人运动，存储机器人运动过程，并能够复现该运动过程。当机器人运动过程较为复杂、环境较为复杂时，示教耗时较长、人力成本较高；当机器人所要执行的任务发生改变时，则需要重新示教，智能水平较低。而且随着对生产效率的要求不断加深，机器人示教仅能从人直观的判断，进行较短路径的选择，示教得到的路径往往消耗很多的代价（如时间、能源等）。示教最大的缺点在于存在一次操作不成功，与障碍物发生碰撞的可能，需要回溯至先前的操作，删除所保留的部分运动过程，当面对的环境较为复杂时，如汽车装配、焊接等，示教所带来的成本剧增。

路径规划问题是机器人领域一大研究重点。其解决实际问题的主要方法是在仿真环境中，使用路径规划算法，规划出一条满足约束并且尽可能达到优化目标的路径。路径规划问题可以分为仿真环境的构建和规划最优路径两部分。仿真环境的构建是指通过传感器或者测量的方式，搭建与真实环境类似的仿真环境，并且将真实环境中的任务，在仿真环境中正确表达。目前，工业机器人仿真环境的构建主要依靠测量，即通过测量真实物体的尺寸和物体之间的距离，在电脑或者其他终端中画出二维或三维模型。其优势在于准确，其

劣势在于时间、人力、经济成本过高，且不适用于动态环境中。规划最优路径是指在仿真环境中，通过算法生成一条满足约束（如不发生碰撞）且尽可能达到某一优化目标（如路径最短、用时最少等）的路径。对于较为复杂的工业机器人，路径最短、用时最小、资源消耗最少等优化目标并不是使机器人末端执行器的路径长度最短，而是使机器人状态空间的改变行程最短。如六轴机械臂的路径规划目标是末端执行器到达目标位置所用时间最短，而决定机械臂运动时间的是六个转轴所转过的角度、角速度、角加速度等。因此对于 n 自由度工业机器人，其优化目标和路径规划均是在 n 维空间中的。目前大多数工业机器人的自由度较大，相比二维移动机器人的路径规划问题维度更高，难度更大。许多适用于低维度的算法，在解决高维问题时，会陷入“维度灾难”，计算复杂度随维度提升呈指数型增长，还会遇到不收敛等问题。

本文基于苏州博众公司所提出的分拣垫片的需求，主要研究仿真环境的构建和工业机器人多维空间路径规划，具有重要的经济意义和现实意义。

1.2 文献综述

国内外学者对三维地图构建领域的研究已有数十年的历史。随着各类传感器的研发、普及、功能的完善，基于传感器的三维地图构建技术愈发成熟。辛煜采用三维激光雷达获得智能无人车前方的点云信息，通过栅格划分和聚类生成障碍物信息，并与四线激光雷达获得的矩阵形式障碍物信息进行匹配，从而构建动态的三维地图信息，实现避障功能。王荣本等针对环境光照多变、地形复杂的情况，基于双目摄像机，提出一种基于稀疏视差匹配的方法进行障碍物检测，从而保证智能车在未知环境中行驶稳定。霍艳艳等则利用双目立体视觉摄像机，通过立体匹配以及光流计算的方法，识别动态障碍物的位置信息和运动信息，初步构建三维地图信息。崔坤征等利用超声波传感器，通过发出并接受回波确定前方是否存在障碍物。赵亮使用无人机搭载的 Kinect 传感器，持续地获得点云数据，通过配准和点云数据融合，构建较为完整的地图信息，从而完成三维地图的构建。

路径规划是机器人领域的一大研究方向。工业机器人所在环境一般为静态环境，其路径规划主要是全局路径规划。Dijkstra 算法、A*算法是较为传统的全局路径规划算法。目前，各类智能算法应用于全局路径规划，并取得一定的成果。Zhang Q 等提出用遗传算法和模拟退火法解决二维地图中路径规划问题，提高了路径规划的效率。张琦利用局部路径信息增强蚁群选择路径的正确性，改进蚁群算法容易陷入局部最优的缺点，提高了路径规划的完备性以及效率。面对工业机器人多维状态空间的路径规划问题，基于随机采样的路径规划算法具有良好的完备性、效率。Kacraki L 等提出的随机路标法（Probabilistic Road Map, PRM）、LaValle S M 等提出的快速扩展随机树法（Rapidly-exploring Random Tree, RRT）以及相应的改进算法，在解决多自由度机器人路径规划问题上应用很广。杨扬为提高服务机器人抓取任务规划的成功率，提出将抓取姿态与运动规划相结合的基于高斯混合模型的多目标快速随机扩展随机树。

1.3 存在的问题和主要工作

根据文献查阅和分析，工业机器人路径规划主要存在以下几个方面的问题：

（1）仿真环境的构建较为复杂，成本较高，主要通过测量的方式建立仿真环境。

（2）路径规划存在效率与最优之间的矛盾。

基于上述问题，本文设计并实现了一套基于实际工业机器人 UR10 的仿真环境构建及路径规划的算法。该算法可以根据用户设定的目标位置以及优化目标，自动构建仿真环境，并生成一条无碰撞且较优的 UR10 运动路径。在仿真环境构建上，本文通过 Kinect 传感器获取点云数据，从而构建障碍物信息。基于实际情况，对比多种构建障碍物信息的算

法，选择其中较优的算法并予以改进，兼顾计算效率以及障碍物信息的保真性。在多维度路径规划上，本文选择应用较广、优势较为明显的传统路径规划算法予以实现，对比其性能，并根据传统路径规划算法的优劣势，开发并实现三种新型路径规划算法：RRT-CS、RRT-CSD、基于随机采样的 A*算法。相较传统路径规划算法，新型路径规划算法在特定方面均具有明显的优势。其中 RRT-CSD 的综合性能最优。此外，本文基于 UR10 实际尺寸参数，完成了 UR10 机械臂的正向运动学和逆向运动学的求解，并且在抓取姿态的优化上做了一定的工作。

1.4 章节安排

本文主要由以下几个章节构成：

第一章为绪论，主要介绍研究背景及意义，综述国内外学者在工业机器人路径规划领域的主要研究成果，分析和总结工业机器人路径规划主要存在的问题，进而提出课题的研究内容和主要工作，最后对全文的章节安排进行一定介绍。

第二章介绍了基于八叉树的三维地图构建过程以及碰撞检测方法。准确地建立仿真环境是工业机器人进行路径规划的前提，同时由于工业应用，对其实时性提出更高的要求。本文主要通过 Kinect 传感器获得环境中的点云信息，构建三维地图信息，并将其转换到 UR10 工业机器人坐标系。通过对比空间分割法、层次包围盒等在计算效率和保真性上的优劣，提出基于八叉树的层次包围盒法构建三维地图信息，并进行碰撞检测，其性能满足实际工业需求。

第三章介绍了工业机器人正向运动学和逆向运动学的求解。本文所使用的工业机器人是 UR10，其满足 Pieper 准则，存在逆向运动学解析解。本文基于 UR10 实际尺寸参数，通过 DH 坐标法，建立正向运动学；通过调整 DH 坐标系的选择，完成 UR10 逆向运动学的求解，并提出逆向运动学解析解求解简化的策略。

第四章介绍了多维度路径规划算法。本文选择并实现了应用较广且优势相对明显的快速扩展随机树算法及已有的其改进算法 RRT、RRT-P、RRT-CONNECT、RRT-STAR，比较其优劣，并开发和实现三种新型的路径规划算法 RRT-CS、RRT-CSD、基于随机采样的 A*算法。新型路径规划算法相较传统的 RRT 算法，在特定方面具有明显优势。RRT-CSD 具有最佳的综合性能。

第五章介绍了抓取姿态的优化。本文研究了不同抓取姿态对路径规划结果的影响，提出抓取姿态优化策略。结果表明，合适的抓取姿态能够显著提高路径规划的优化性能。

第六章介绍了本文所设计的工业机器人路径规划系统的结构和使用流程，并对有障碍物的路径规划进行实验。系统实现了对障碍物的建模，为 UR10 规划了一条不发生碰撞的路径，最终控制 UR10 完成实际任务。

第五章是对全文的总结，对未来工作的展望

第二章 基于八叉树的三维地图构建

2.1 问题描述

机器人在执行任务时，处于包含多种物体的空间环境中。除了准确、快速完成任务之外，机器人不得影响周围的空间环境，比如打到周围物体，使环境或者机器人本身受到损坏。因此在进行规划机器人如何完成具体任务时，需要充分考虑环境信息，保证环境、机器人安全的基本前提。因此，快速、准确构建三维地图，并且进行机器人与环境的碰撞检测，是机器人规划任务之前所面临的问题。

本文 RGBD 传感器所采集到的点云数据，构建三维地图信息，并且实现机器人与环境的碰撞检测。该项工作面临以下几个挑战：

（1）由 RGBD 传感器采集到的点云数量庞大，直接对点云数据进行处理，将耗费大量时间。本文所采用的 Kinect1.0 采集的点云数据达到上万个。

（2）三维地图信息需要进行 RGBD 传感器指定坐标系到机器人坐标系的转换。RGBD 传感器通过标定，可以获得内参与指定坐标系的外参，从而将三维地图信息从传感器坐标系转换到世界坐标系。由于无法将世界坐标系直接建立在机器人坐标系下，因此需要将三维地图信息再次转换到机器人坐标系下。如何在缩短转换时间的同时保持三维信息的精度，以提高碰撞检测的有效性及任务规划的可解性，是又一挑战。

（3）点云数据无法直接描述环境，需要通过点云数据快速建立高保真、容易碰撞检测环境三维模型。

2.2 基于八叉树的三维地图信息

八叉树是一种每个节点有八个子树的数据结构，是四叉树在三维上的扩展。其每个节点表示一个正方体，每个子节点是对父节点所代表的正方体的八等分。对每个父节点进行八等分，其子节点编号可由图 2-1 表示。

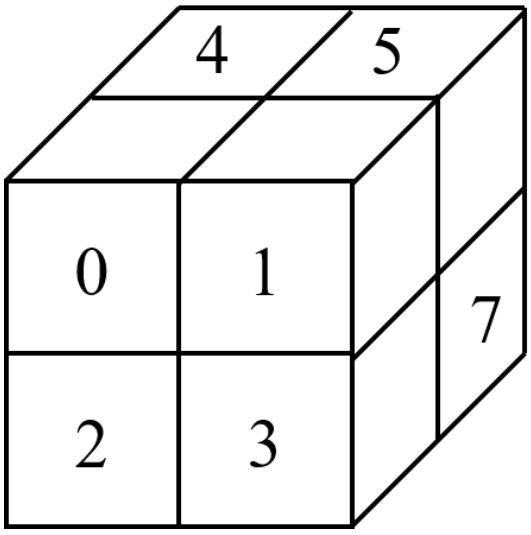


图 2-1 八叉树节点表示

假设环境空间是一个边长为 S 正方体，设置叶节点代表的正方体的边长为 S_0。设八叉树结构的层次 n。

$$n \geq \frac{S}{S_0} \tag{2-1}$$

则 n 是满足式的最小整数。由此每个叶节点对应的小正方体可以由 $P_1P_2\ldots\ldots P_{n-1}P_n$ 来表示。P_{i-1} 代表在 P_i 父节点所代表的正方体所分割的八个子正方体中的哪一个。

对一组点云数据，可以将坐标对应到环境中每一个叶节点所对应的正方体中去。设点云坐标为（x，y，z）。设根节点对应的正方体八个顶点中，三个方向的均处于最低状态，即数值上最小的顶点的坐标为根坐标，为（x_0 ，y_0 ，z_0）。设（i，j，k）是点云坐标在三个方向上距根坐标的叶正方体的个数。

$$\begin{cases} i = [(x - x_0)/S_0] \\ j = [(y - y_0)/S_0] \\ k = [(z - z_0)/S_0] \end{cases} \tag{2-2}$$

将（i，j，k）用二进制进行表示，分别为 $i_1i_2\ldots\ldots i_{n-1}i_n$、$j_1j_2\ldots\ldots j_{n-1}j_n$、$k_1k_2\ldots\ldots k_{n-1}k_n$。

$$P_m = i_m + 2j_m + 4k_m, m = 1,2\ldots\ldots n-1,n \tag{2-3}$$

由此可以从点云三维坐标转换到八叉树编码。

将所有点云数据进行八叉树编码，并将用编码对应的叶正方体的体心来表示同一编码的所有点云数据，可大大减少点云数据，其外表面精度误差在 S_0。其设置的叶正方体边长越小，则外表面精度越高。

2.3 基于八叉树的层次包络盒法

直接通过八叉树所构建的三维地图信息存在以下的缺点：

（1）数据存储占用空间较大，对于实际由点云构建的应用场景，八叉树地图信息所需要存储的数据点达到上千个，同时还需要存储父节点与子节点之间的关系；

（2）点云存在数据丢失的情况，若叶正方体边长过小，则数据丢失会造成碰撞检测失灵，与精度存在一定矛盾；

（3）针对本文所涉及的三维地图信息需要进行坐标转换的特殊情况，直接对八叉树构建的三维地图信息进行三维转换，需要重新进行八叉树编码，同时存在数据量大造成的转换效率低的问题。

因此，本文使用八叉树精简点云数据后，再使用层次包络盒法构建环境信息。

2.3.1 层次包络盒简介

包络盒是指能够包住物体外表面、且形状具有特定优势的几何形体。在碰撞检测中，要求包络盒具有以下两种优势：能够尽量贴合三维物体的外表面，保真性更高，以获得更大的自由空间；能够快速准确地进行碰撞检测。针对本文所涉及的坐标转换，要求包络盒具有旋转不变性。

常见的包络盒有坐标轴包围盒 AABB（axis-aligned bounding boxes）、包围球（spheres）、方向包围盒（oriented bounding boxes）等。

AABB 是传统的碰撞检测领域的包围盒算法。其构造简单，仅需确定空间中 x、y、z 三个方向上的最小和最大值即可表示该包围盒；其储存空间较小，一个包围盒只占用六个变量空间。其最大的优势在于碰撞检测十分快速简单。在判断某点是否在包围盒内时，仅需根据该点的坐标，在三个方向上是否均在最大值与最小值之间最多六次判断即可。在判断两个包围盒是否相交，将两个包围盒在 x、y、z 平面投影，当三次投影面积均产生重叠时，两个包围盒相交。但 AABB 存在一定的缺点，主要是其空间存在较大的空隙，保真性和紧密性较差。AABB 包围盒的边平行与坐标轴，当物体呈倾斜，尤其与坐标轴呈 45°时，该缺点更加

明显。这一缺点会导致空间中的可行空间大幅减小，很多不会产生碰撞的情况会被认为产生了碰撞。在机器人任务规划时，会造成规划时间加长，甚至导致不收敛或无解的情况。

包围球算法是构建能够包围物体的最小球体。包围球的构造只需将该物体的元素如点云的坐标取平均作为球心坐标，选取球心到各元素的最大距离作为半径。其优势在于旋转不变性以及存储空间较小。当坐标系发生转换或者物体产生旋转运动时，仅需更新包围球球心位置即可，物体姿态无须考虑。在存储空间上，其存储一个物体，仅需四个变量，即球心坐标以及半径。包围球的碰撞检测同样直观和简便。判断某一点是否在包围球内部，根据点到球心的距离是否小于球半径即可。判断两个球是否碰撞，根据球心间的距离是否小于两个球半径之和即可。相较 AABB，在构造上，因为包围球法不仅需要遍历数据进行平均计算，还需遍历数据进行求距离的计算，所需的计算时间更长，尤其当数据达到万级时，如处理点云数据时；其碰撞检测更加复杂，求距离的计算所需时间远大于判断；其保真性和紧密性依赖于物体本身的形状，若本身呈现长条形，则将产生巨大的空隙，严重压榨环境中的自由空间。

OBB 是包围物体的最小长方体，其方向没有任何限制。其缺点在于构造难度。如何选取长方体最佳方向，使其包围物体，并使空隙最小，是较大的难点。目前构造 OBB 的主要方法是通过主成分分析确定最佳方向，由此构建包络长方体。OBB 法所占用的存储空间也较大。一个 OBB 包围盒需要 15 个向量，9 个方向信息，3 个中心点坐标，3 个长度信息。其碰撞检测相较于 AABB 与包围球更加复杂，用时更多。判断某点是否在 OBB 包围盒内部，需要将该点在世界坐标系下的坐标转换到 OBB 包围盒坐标系下，再通过六次判断，才能确定；判断两个 OBB 包围盒是否碰撞，需要联立十二个三元一次不等式组（每个长方体所包围的空间可以通过六个不等式来表示），根据是否有解才能判断。OBB 虽然具有空隙较小的优势，但在构造和碰撞检测的效率、存储所用的空间上，均存在一定的缺点和劣势。

本文点云提取所用的传感器为一架 Kinect，其获得的点云数据为物体的部分表面。由于工业场景中的物体多为多面体结构，因此使用包围球方法进行障碍物的检测会造成空隙过大的问题。使用 OBB 方法进行障碍物检测则会降低构建障碍物信息及碰撞检测的效率。本文最终使用 AABB 构建障碍物信息，并通过实际测量障碍物尺寸信息，补充一台 Kinect 获得的物体残缺的点云，从而提高三维地图信息建模的保真性。

2.3.2 障碍物聚类及平面分割

从 Kinect 获得的点云数据具有以下几个问题：

（1）无法区分不同的障碍物。孤立障碍物生成的孤立点云团需要从整个点云集中区分开来。

（2）无法解决相互接触的障碍物的分离。

针对本文所涉及的抓取垫片的具体任务，本文主要使用聚类及平面分割法解决简单环境中的障碍物区分问题。问题所在环境是在桌面上放有待抓取的垫片，同时具有长方体的障碍物。

本文使用聚类算法解决孤立障碍物的分离问题。主要步骤如下：

（1）设总点云数据集为PC，已聚类的点云数据集为$Cluster$，临时聚类集为S，距离阈值为d。

（2）在PC中任取一点加入S。

（3）对S中的任意一点，选择PC中所有满足距该点距离小于d的点，加入S，并从PC中剔除。

（4）遍历S中所有的点，执行步骤（3），直至没有新的点加入S，将S作为一个$Cluster$，清空S。

（5）重复步骤（2）、（3）、（4），直至PC为空集。

本文使用 RANSAC 平面分割法，实现桌面与桌面上的障碍物的分割。其主要步骤如下：

（1）设总点云数据集为PC，平面点云数据集为$Cluster_{face}$，临时聚类集为S，桌面法向量为$\vec{n}$。

（2）在PC中任取三点P_1、P_2、P_3，若三点不共线，则生成平面单位法向量$\vec{m}$，并将三点纳入S，从PC中剔除。

（3）对PC其他点P_i，若$\left|(\mathrm{P_i}-\mathrm{P_1})\cdot\frac{\vec{\mathrm{m}}}{|(\mathrm{P_i}-\mathrm{P_1})|}\right|<\varepsilon$（$\varepsilon$为人为设定的小于 1 的阈值），则将$P_i$纳入$S$，从$PC$中剔除。

（4）将S中的所有点云数据经过最小二乘平面拟合，得到新法向量$\vec{l}$。对S中的所有点云数据，进行$\left|(\mathrm{P_i}-\mathrm{P_1})\cdot\frac{\vec{\mathrm{l}}}{|(\mathrm{P_i}-\mathrm{P_1})|}\right|<\varepsilon$检测，对不通过的点予以剔除。

（5）将S作为一个$Cluster_{face}$，清空S，重复步骤（2）、（3）、（4）、（5），直至PC为空集。

（6）选择$Cluster_{face}$中点云数据最多的点集作为桌面。

2.3.3 三维地图信息的构建

三维地图信息的构建流程如下：

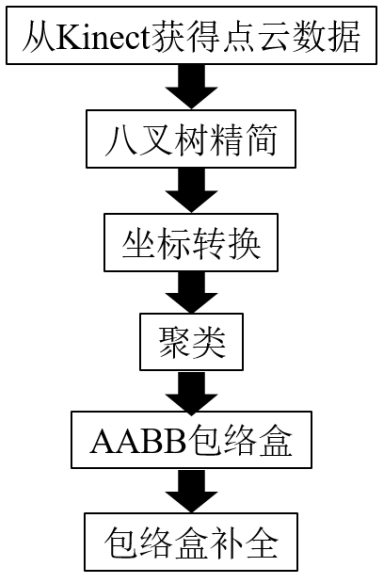


图 2-2 三维地图信息构建流程框图

本文针对工业机器人抓取垫片的实际场景，通过一架 Kinect 获得点云数据，依照图 2-1 所示流程，选取边长为 20mm 的正方体作为八叉树的叶节点，进行三维地图信息构建，依次得到的效果图如下。

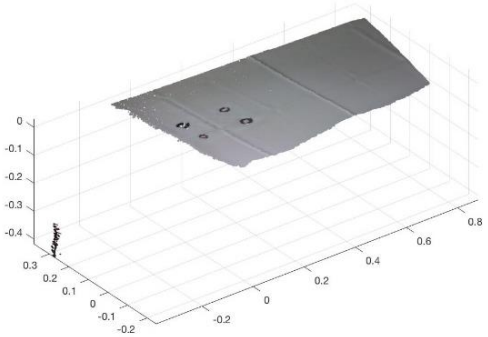


图 2-3 原始点云数据

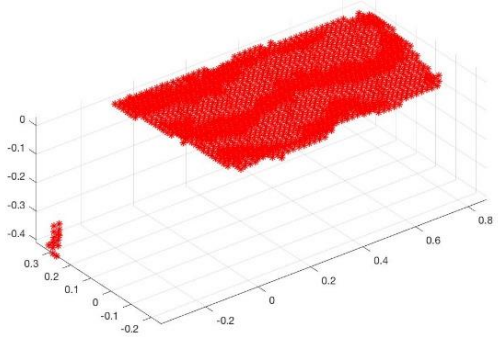


图 2-4 精简后点云数据

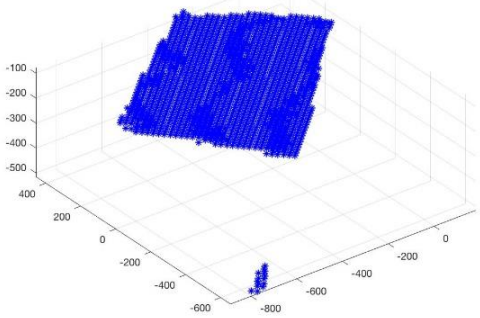


图 2-5 坐标转换后点云数据

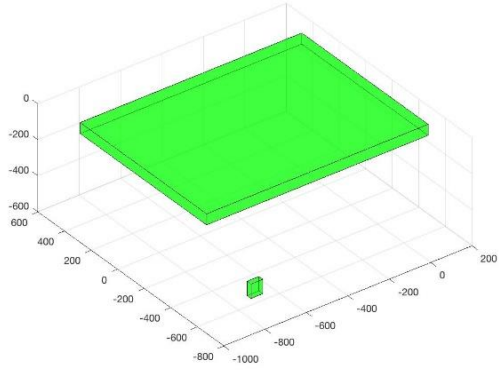


图 2-6 基于尺寸的 AABB 包络盒构建

表 2-1 三维地图构建算法性能对比表

	八叉树转换用时/s	坐标转换用时/s	包络盒用时/s	总用时/s	包络盒平均边长空隙/mm
基于八叉树的层次包络盒法	2.99	0.06	0.05	3.10	D-9.8
层次包络盒法	0.00	8.63	0.08	8.71	D

基于八叉树的层次包络盒法相较于传统的层次包络盒法，在牺牲较小的精度下，能够大幅提高三维地图构建的计算效率，减少约 64.4%的时间。

2.4 包络盒膨胀与碰撞检测

判断机器人与障碍物是否发生碰撞，传统做法是分别对障碍物和机器人进行包络，通过包络盒之间的碰撞检测进行判断。这种碰撞检测方式有以下几个缺点：

（1）机器人包络盒构建较为复杂。机器人的包络盒构建需要通过对机器人外形具有较为精确的三维模型。同时，包络盒需要随着机器人的运动而产生运动，需要对包络盒实时进行动态更新。

（2）包络盒与包络盒之间的碰撞检测计算效率较低。相较判断点是否在包络盒内部，两个包络盒的相交检测计算复杂度显著更高。

由于传统工业机器人主要为多轴机械臂，外形呈沿轴线向径向均匀扩展。因此本文以机器人轴线代替机器人整体，将障碍物包络盒向外膨胀机器人最大轴半径的方式，通过判断轴线与包络盒之间是否相交，模拟实际碰撞检测。判断轴线与障碍物包络盒是否相交，可以将轴线等分，若各等分点均不在包络盒内部，则表示并没有相交。其碰撞检测算法简单，计算效率较好，同时保证了碰撞检测的有效性，具有一定的工业应用价值。

2.4 OBB 包络盒及碰撞检测

由于 AABB 包络盒生成的障碍物信息相比障碍物本身的信息会产生较大的空隙，同时考虑到本文所涉及的障碍物是长方体，并且平放在桌面上，其 OBB 包络盒构造以及碰撞检测均能够大幅简化，因此本文使用 OBB 包络盒法进行障碍物信息构建以及碰撞检测。

OBB 包络盒法主要步骤如下：

（1）通过主成分分析法确定 OBB 包络盒主方向。对已经完成聚类的点云数据构造其协方差矩阵，即

$$\mathrm{A}=\begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(x,y) & cov(y,y) & cov(y,z) \\ cov(x,y) & cov(y,z) & cov(z,z) \end{bmatrix} \tag{2-4}$$

其中$cov(x,y)$表示x、y之间的协方差。

然后计算协方差矩阵的特征向量，该组特征向量即为 OBB 包络盒主方向。

（2）将该类点云数据在主方向进行投影，取各方向上的投影的均值以及长度，作为 OBB 包络盒体心以及边长，完成构造。

OBB 包络盒碰撞检测方法为将点在转换到以 OBB 体心为原点，主方向为坐标轴方向的坐标系中，通过 OBB 包络盒边长，经过六次比较判断，即可完成点是否在包络盒内的判断。

对于本文所涉及的简单环境，即长方体障碍物位置水平桌面上的情况，其 OBB 包络盒其中一个主方向必为重力方向，仅需将障碍物点云数据投影到水平平面，在该平面上进行主成分分析，确定另外两个主方向即可。其碰撞检测只要先判断高度是否在该包络盒高度范围

内，若是则在除重力方向外的主方向进行投影，然后进行四次比较判断即可。在狭窄的环境中，OBB 包络盒法能够显著减少空隙，提高路径规划的成功率。

针对本文所涉及的避开障碍物的抓取垫片的任务，通过 Kinect 获取点云数据，使用基于八叉树的 OBB 层次包络盒法进行三维地图信息构建。

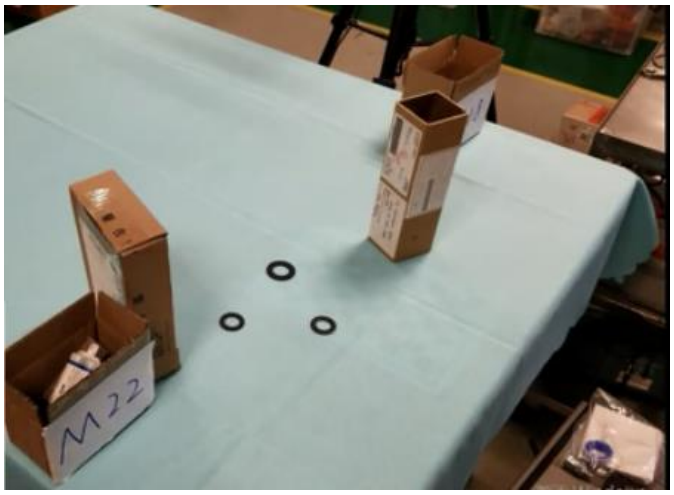


图 2-7 避障的垫片抓取任务场景

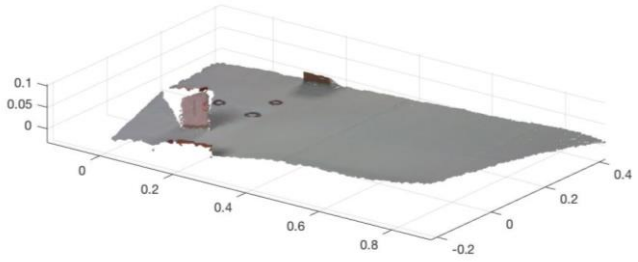


图 2-8 从 Kinect 获得的实际场景点云图

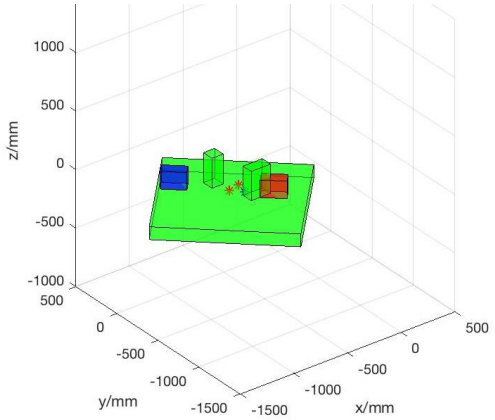


图 2-9 UR10 坐标系下的三维地图构建

图 2-9 中蓝色和红色盒子是不同垫片需要分拣到的地方。因为 Kinect 无法拍到这两个盒子，因此通过测量方式构建。绿色盒子是通过点云数据并进行尺寸补偿后构建的障碍物信息。其中绿色盒子与红色盒子发生了干涉，这是因为对障碍物信息进行了膨胀操作。

第三章 机械臂运动学

3.1 机械臂运动学简介

机械臂通过驱动转轴，改变转角大小，最终改变末端执行器的位置和姿态。

机械臂运动学分为正向运动学和逆向运动学。机械臂正向运动学是指由转角大小，解出末端执行器的位置和姿态；而机械臂逆向运动学是指由末端执行器的位置和姿态，解出各转角的大小。当机械臂执行任务时各时间对应的转角大小已知，则末端执行器的位置和姿态可以通过正向运动学得到。但通常情况下，转角大小是未知的，而机械臂末端执行器的位置和姿态可以通过任务目标确定。因此机械臂逆向运动学在是机械臂路径规划的前提。同时，机械臂在执行任务时，不能与周围环境产生碰撞而发生损坏和危险，因此通过转角确定机械臂各臂以及末端执行器的位置和姿态，即正向运动学，是保证机械臂能否安全、成功地执行任务的前提。

机械臂正向运动学方法主要有 DH 坐标法、指数坐标法等。DH 坐标法具有构造简单的优点。机械臂逆向运动学主要有解析解与数值解法。当三个相邻关节轴交于一点或三轴线平行，符合 Pieper 准则，其逆解具有解析解。解析解具有运算速度快，精度高的优点。解析解主要通过对末端执行器的齐次变换矩阵的求解，确定各转角的大小。本文所采用的 UR10 机械臂满足 Pieper 准则，存在解析解。

3.2 机械臂正向运动学

3.2.1 DH 坐标法

机器人可以简化为关节和杆件的连接。关节是指可以旋转和移动的结构。在实际的生产和应用中，受制造、应用等约束，机器人的关节主要分为旋转关节和移动关节。杆件是刚性的连接件。DH 坐标法主要通过构造各关节的坐标系，通过坐标系之间的转换，建立正向运动学，从而可以从各关节的旋转和移动，解出各杆件和末端执行器的位置和姿态。本文主要参考蔡自兴教授所著《机器人学》，完成 UR10 DH 坐标系的建立。其主要步骤为：

（1）坐标系的构建。设从基座开始的各旋转关节的轴向和移动关节的移动方向为 J_1……J_{n-1} J_n，每个 J_i 对应的方向为各关节坐标系的 z_i 轴。世界坐标系的原点和方向可以根据 z_1，选择容易建立齐次变换矩阵选择原点以及方向。x_{i-1} 轴是 z_{i-1}、z_i 的公垂线，方向为 z_{i-1} 到 z_i。y_{i-1} 则根据右手定则确定。

（2）DH 参数确定。a_{i-1} 表示 z_{i-1} 到 z_i 沿 x_{i-1} 的距离。d_i 表示 x_{i-1} 到 x_i 沿 z_i 的距离。α_{i-1} 表示 z_{i-1} 绕 x_{i-1} 转到 z_i 的角度。θ_i 表示 x_{i-1} 绕 z_i 转到 x_i 的角度。

构建坐标系，确定 DH 坐标系后，可以确定由 z_{i-1} 坐标系到 z_i 坐标系之间的齐次变换矩阵 ${}^{i-1}_{i}T$。具体为几个齐次变换矩阵连乘结果，即先绕 x_{i-1} 轴旋转角 α_{i-1}，使 z_{i-1} 和 z_i 平行，再沿 x_{i-1} 轴平移 a_{i-1}，使 z_{i-1} 坐标系的原点位于 z_i 轴上，然后绕 z_i 轴旋转，使 x_{i-1} 和 x_i 平行，最后沿 z_i 轴平移 d_i，使两个坐标系重合。

$$\begin{cases} {}^{i-1}_{i}T = Rot(x,\alpha_{i-1})Trans(a_{i-1},0,0)Rot(z,\theta_{i-1})Trans(0,0,d_i) \\ Rot(x,\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & 0 \\ 0 & \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Trans(a_{i-1},0,0) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Rot(z,\theta_{i-1}) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Trans(0,0,d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{cases} \quad (3\text{-}1)$$

由此可以解得：

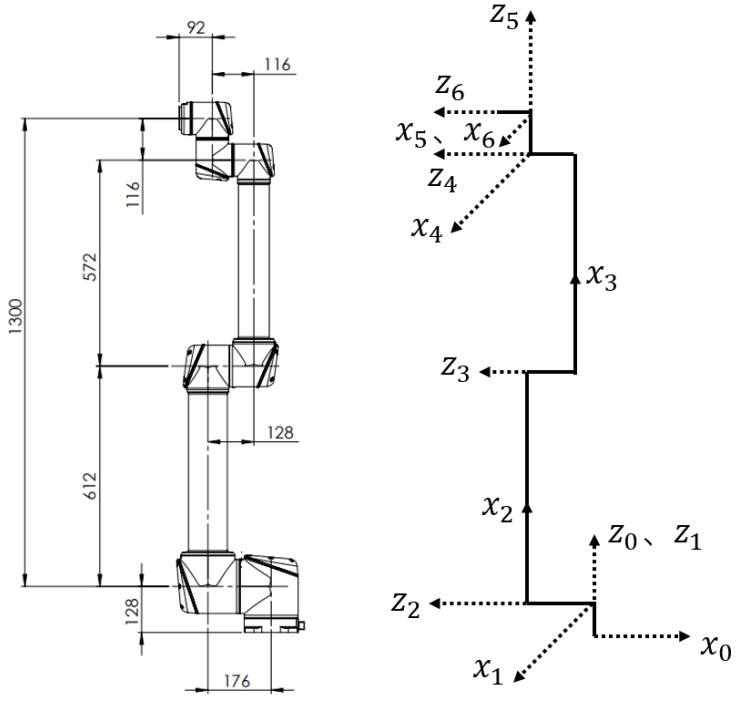
$${}^{i-1}_{i}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3\text{-}2)$$

设末端执行器对应的坐标系为 z_n 坐标系。则末端执行器所在坐标系到基坐标系（即世界坐标系）的齐次变换矩阵${}^{0}_{n}T$可由下式求解

$${}^{0}_{n}T = {}^{0}_{1}T{}^{1}_{2}T \ldots\ldots {}^{n-2}_{n-1}T{}^{n-1}_{n}T \quad (3\text{-}3)$$

3.2.2 UR10 机械臂正向运动学

由 UR10 机械结构及尺寸信息，建立 DH 坐标系，确定 DH 参数



(a)UR10机械结构图　　(b)DH坐标系

图 3-1 UR10 实际尺寸参数及 DH 坐标系

表 3-1 DH 参数表

T	a_{i-1}/mm	d_i/mm	α_{i-1}/°	θ_i/°
${}^0_1T$	0	128	0	θ_1
${}^1_2T$	0	176	90	θ_2
${}^2_3T$	612	-128	0	θ_3
${}^3_4T$	572	116	0	θ_4
${}^4_5T$	0	116	-90	θ_5
${}^5_6T$	0	0	90	θ_6

根据 DH 参数以及式，即可得到机械臂正向运动学的解。

3.3 机械臂逆向运动学

3.3.1 机械臂逆向运动学求解

对于满足 Pieper 准则的机器人，逆向运动学存在解析解。但不同的 DH 坐标系影响解析解的难易程度。本文探讨了 DH 坐标的选择对逆向运动学求解的难度的影响，设定了 DH 坐标系选择的准则，并完成了 UR10 机器人逆向运动学求解。

为表达简洁，用c_i表示$\cos(\theta_i)$，用s_i表示$\sin(\theta_i)$，用s_{ijk}表示$\sin(\theta_i+\theta_j+\theta_k)$，用$c_{ijk}$表示$\cos(\theta_i+\theta_j+\theta_k)$。

当末端执行器的位置和姿态已知，即${}^0_6T$已知。则可以通过下式，进行逆向运动学求解。

$$ {}^1_6T = ({}^0_1T)^{-1}{}^0_6T \tag{3-4} $$

其中

$$ \begin{cases} {}^1_6T = \begin{bmatrix} {}^1n_x & {}^1o_x & {}^1a_x & {}^1p_x \\ {}^1n_y & {}^1o_y & {}^1a_y & {}^1p_y \\ {}^1n_z & {}^1o_z & {}^1a_z & {}^1p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{cases} \tag{3-5} $$

若在${}^1_6T$的元素中存在常数项，则可以通过该常数项求解出θ_1，大大简化逆向运动学的求解。若不存在常数项，则可以选择角度变量最少的一项，通过坐标系之间沿 x 轴或 z 轴的平移，消除角度项前的系数，从而获得常数项，求解出θ_1。

本文所构建的 DH 坐标系，是经过调整后，将末端执行器的坐标系沿 z 轴平移至原点与前一个坐标系的原点重合，获得常数项后的结果。

现按照已经建立 UR10DH 坐标系，完成逆向运动学求解。

$$\begin{cases} {}^1n_x = c_{234}c_5c_6 - s_{234}s_6 \\ {}^1n_y = s_5c_6 \\ {}^1n_z = c_{234}s_6 + s_{234}c_5c_6 \\ {}^1o_x = -s_{234}c_6 - c_{234}c_5s_6 \\ {}^1o_y = -s_5s_6 \\ {}^1o_z = c_{234}c_6 - s_{234}c_5s_6 \\ {}^1a_x = c_{234}s_5 \\ {}^1a_y = -c_5 \\ {}^1a_z = s_{234}s_5 \\ {}^1p_x = a_3c_{23} + a_2c_2 - d_5s_{234} \\ {}^1p_y = -d_2 - d_3 - d_4 \\ {}^1p_z = a_3s_{23} + a_2s_2 + d_5c_{234} \end{cases} \tag{3-6}$$

由 ${}^1p_y$为常数项，可得求解θ_1的一元方程：

$$-s_1p_x + c_1p_y = -d_2 - d_3 - d_4 \tag{3-7}$$

解得：

$$\theta_1 = atan2(p_y, p_x) - atan2(E, \pm\sqrt{F}) \tag{3-8}$$

式中，$E = -d_2 - d_3 - d_4$，$F = {p_x}^2 + {p_y}^2 - E^2$

然后观察到 ${}^1n_y$、${}^1o_y$、${}^1a_y$包含的角度变量较少，由此可以解出θ_5、θ_6。对应的二元方程组为：

$$\begin{cases} -s_1n_x + c_1n_y = s_5c_6 \\ -s_1o_x + c_1o_y = -s_5s_6 \\ -s_1a_x + c_1a_y = -c_5 \end{cases} \tag{3-9}$$

可以观察到当$s_5 = 0$时，UR10 产生奇异，无法求解出θ_6。若$s_5 \neq 0$，解得：

$$\begin{cases} s_5 = \pm\sqrt{(-s_1n_x + c_1n_y)^2 + (-s_1o_x + c_1o_y)^2} \\ \theta_5 = atan2(s_5, s_1a_x - c_1a_y) \\ \theta_6 = atan2(\frac{s_1o_x - c_1o_y}{s_5}, \frac{-s_1n_x + c_1n_y}{s_5}) \end{cases} \tag{3-10}$$

可以观察到 ${}^1a_x$、${}^1a_z$所包含的角度变量较少（可将$\theta_2 + \theta_3 + \theta_4$看作一个整体），方程组为：

$$\begin{cases} a_xc_1 + a_ys_1 = c_{234}s_5 \\ a_z = s_{234}s_5 \end{cases} \tag{3-11}$$

可以观察到当$s_5 = 0$时，UR10 发生奇异，与之前的奇异位置相同。若$s_5 \neq 0$，解得：

$$\theta_2 + \theta_3 + \theta_4 = \text{atan2}(\frac{a_z}{s_5}, \frac{a_xc_1 + a_ys_1}{s_5}) \tag{3-12}$$

最后观察到包含θ_2、θ_3、θ_4信息的其他元素只有 ${}^1p_x$、 ${}^1p_z$，联立，可以解得：

$$\begin{cases} \theta_2 = atan2(B, -A) - atan2(C, -\sqrt{A^2 + B^2 - C^2}) \\ \theta_3 = atan2\left(\frac{M - a_2s_2}{a_3}, \frac{N - a_2c_2}{a_3}\right) - \theta_2 \\ \theta_4 = \text{atan2}\left(\frac{a_z}{s_5}, \frac{a_xc_1 + a_ys_1}{s_5}\right) - atan2\left(\frac{M - a_2s_2}{a_3}, \frac{N - a_2c_2}{a_3}\right) \end{cases} \tag{3-13}$$

式中：

$$\begin{cases} M = p_z - d_1 - d_5 c_{234} \\ N = p_x c_1 + p_y s_1 + d_5 s_{234} \\ A = 2a_2 M \\ B = 2a_2 N \\ C = {a_2}^2 - {a_3}^2 + M^2 + N^2 \end{cases} \tag{3-14}$$

至此，得到 UR10 机械臂逆解。

3.3.2 机械臂多解性的选择

由 3.3.1 可得，UR10 机械臂逆解存在多解性。对于工业机器人路径规划，其逆向运动学主要用于确定目标构型，即终止位置时各自由度对应的值。对于终止位置所对应的状态空间的多个解，通过是否与环境发生不必要的碰撞等，删除不符约束的解；通过优化目标，如用时最短等，从满足约束的解中选择最优的解作为目标构型。从理论分析和实验结果来看，通过对机械臂逆解多解的选择，可以对路径规划结果起到显著的优化效果，同时根据是否存在满足约束的解，可以快速判断机械臂是否能够达到目标位置。若无法达到，则说明需要调整任务目标，提高了反馈应激性。

第四章 多自由度机器人路径规划

4.1 引言

路径规划是指为机器人规划一条从起始位置到达终止位置，尽可能达到某一目标（如路径长度最短、所用时间最短、代价最少等）的避开障碍物的运动路径。

路径规划的算法可以按照空间的维度进行划分，主要可以分为二维、三维和多维。

人工势场法是将地图信息和目标位置转换成力的作用，在环境中人工生成势场，将机器人置于起始位置，将沿着势能减小的方向运动，从而最终达到目标位置。其构造人工势场的方法主要是将目标位置作为引力点，将障碍物作为斥力点。人工势场法规划的路径具有平滑性的优点，但容易陷入局部最小点。若引力以及斥力的大小选择不当，在环境中产生局部势能极小点，而该极小点并不是终止位置，则机器人可能会进入该极小点而停止。人工势场法的难点在于设定引力与斥力的参数，构建没有局部最小点的人工势场，并且能够更接近优化目标。

基于图搜索的方法是指将机器人所在的状态空间分成一个个的栅格，从而形成图状结构，将路径规划问题转化成基于某个优化目标的图搜索问题。图搜索的算法主要有 Dijkstra 算法、A*算法、D*算法。Dijkstra 算法是指从起始位置开始，不断向相邻的栅格扩展，分别计算到各栅格的距离，并不断更新到各栅格的最小代价，最终达到目标位置所在的栅格，结束搜索。其中，代价可以指路径长度、所耗费的时间、资源消耗等。该算法由于没有方向性，不断向已经搜索过的栅格的外围拓展，其效率十分低下。算法虽然能够找到最优路径，但随着搜索次数的增加，向目标位置的拓展速度呈指数型下降。A*算法是 Dijkstra 算法的改进版，是一种启发式的算法。该算法设定了一个启发函数$f(n)$，n代表节点序号。$f(n) = g(n) + h(n)$，$g(n)$表示从起始位置到节点n的最小代价，$h(n)$表示从节点n到目标位置的估计代价。$g(n)$可以通过不断地根据已搜索的节点与新增节点之间的代价，更新节点n最小代价来确定。$h(n)$则是一个启发函数，当$h(n)$小于实际从节点n到目标位置的代价，则总能找到最优路径；当$h(n)$等于实际从节点n到目标位置的代价，则此时搜索沿着最优路径，效率最高，且能找

到最优路径；当$h(n)$大于实际从节点n到目标位置的代价，则虽然效率提高，但不能保证路径是最优的。A*算法进行扩展时，是对已搜索节点的外围节点中选择$f(n)$最小的节点，向周围栅格进行拓展，因此其拓展具有方向性，效率高于 Dijkstra 算法。Dijkstra 算法和 A*算法主要适用于静态环境中的路径规划，而 D*算法主要适用于动态环境中的路径规划，是 Dijkstra 算法和 A*算法或其他算法的结合。D*算法是通过 Dijkstra 算法在已知环境中规划最优路径，若环境发生改变，则在局部进行 A*算法或其他路径规划算法，获得当前环境下的最优路径。

基于图搜索的路径规划算法虽然在路径优化上具有很大优势，但当机器人维度提高，如在 UR10 机械臂的六维状态空间进行图搜索，则栅格的数量将随着维度呈指数型增长，搜索效率十分低下，在有障碍物的情况下，更为明显。人工势场法在高纬的应用则需要将势场转换到高纬环境中，参数增加，能否成功完成路径规划，依赖于参数的选择。在目前路径规划的首要目标是在有限时间内成功完成路径规划的环境下，上述两种类型的路径规划算法均存在一定的不适用性。

基于随机采样的路径规划算法则在牺牲优化性能的情况下，保证路径规划的效率以及成功率。该类路径规划算法主要思想为在机器人状态空间进行随机采样，将采样点作为路径经过点，通过在路径经过点之间的连线中，寻找一条连接起始位置和终止位置的路径。若路径不存在，则增加采样点。基于随机采样的路径规划算法总能快速找到一条满足约束的路径，其应对高维、复杂地图的能力十分优秀，逐渐成为机器人路径规划的常用方式。

4.2 基于随机采样的路径规划算法

基于随机采样的路径规划是在机器人构型空间的一种路径规划算法。对于 n 自由度的机器人，其状态可以用 n 维列向量来表示，是 n 维构型空间中的一点，将其定义为构型 p。构型空间可以分为自由构型空间和障碍物空间。机器人路径规划问题可以转化为在自由构型空间中，寻找一条尽可能达到优化目标的从起始构型到达目标构型的路径。障碍物空间的建立是一个 PSPACE-HARD 问题。从障碍物空间建立的复杂程度上看，需要将三维或二维的障碍物信息转换到多维空间中，多维空间的描述以及判断是否属于障碍物空间的难度均较大。因此，判断构型是否处于自由构型空间，仅需对该构型进行碰撞检测，若与障碍物发生碰撞，则属于障碍物空间。传统的基于随机采样的路径规划算法有扩张空间树法（Expansive Space Tree, EST）、随机路标法（Probabilistic Road Map, PRM）、快速扩展随机树法（Rapidly-exploring Random Tree, RRT）等。这里主要介绍在机器人路径规划领域应用较广、优势较为明显的 PRM 与 RRT 算法。

随机路标法主要是在构型空间中随机采样，生成一系列随机采样点，若处于障碍物空间（碰撞检测等），则删去，直至生成足够个数的处于自由构型空间的随机采样点。对每一个采样点进行 K 近邻搜索，搜索到与该点代价小于设定值的点，将其连接，判断该路径是否会碰到障碍物，若碰到则删除该路径，否则则保留。最终生成采样点对应构型，边代表无碰撞路径的图状结构。最后通过图搜索算法，获得该图的最优路径。随机路标法适用于静态的场景，而且算法是概率完备。若随机采样点选取不合适、数量不够多，则存在无法生成从起始构型到目标构型的路径，需要增加采样点数量，重新进行路径规划，而且其优化效果也较为一般。

快速随机扩展随机树法是一种树扩展算法，其主要思想为从起始构型开始构建随机扩展树，以起始构型作为根节点，设定d为随机扩展树扩展代价，在构型空间中随机采样获得某一随机构型P_{rand}，搜索随机扩展树中离该随机构型代价最小的节点P_{near}，若$||P_{rand}-P_{near}|| < d$（$||\cdot||$表示两个构型之间的代价），若$P_{rand}$属于只有构型空间，则将$P_{rand}$作为随机扩展树新节点，同时将$P_{rand}$与$P_{near}$相连；若$||P_{rand}-P_{near}|| > d$（$||\cdot||$表示两个构型之间

的代价），则将取在P_{near}到P_{rand}方向上代价为d的构型作为P_{new}，若P_{new}属于自由空间，则将P_{new}作为随机扩展树新节点，同时将P_{new}与P_{near}相连。重复上述过程，直至新节点恰好为目标构型或者离目标构型的代价小于d。快速随机扩展随机树法有以下优势：能够向未知区域进行快速扩展，路径搜索效率较高；其扩展的无序性能够很好适应复杂地图信息；快速随机扩展随机树法是概率完备的，只要足够的时间，总能找到一条从起始构型到达目标构型且无碰撞的路径。

由于 RRT 算法相较其他随机采样算法具有一定优势，目前机器人路径规划所采用的算法多为 RRT 算法及其优化算法。

RRT-P 是对随机采样的无序性进行一定的约束，以提高随机扩展树向目标构型的扩展的速度。其主要思想是取定一个概率P，表示随机采样点有概率P会取到目标构型，从而约束随机扩展树的扩展方向，提高路径规划的效率。其弊端在于若环境较为复杂，则反而可能降低路径规划的效率。但在多维空间中或环境不那么复杂的情况下，其路径规划的效率和优化效果会得到提升。

RRT-P 相较于 RRT 的效率提升并不是十分的明显，优化效果也不佳，经常出现会在某一空间内重复行走的情况。RRT-CONNECT 则可以更显著地提升计算效率，并且大幅提高优化效果。其主要思想为在起始构型与目标构型分别生成随机扩展树，并且向另一棵树的方向进行扩展，直至两棵树完成连接。其主要步骤如下：

（1）设起始构型为P_{start}，对应的随机扩展树$Tree_{start}$，目标构型为P_{goal}，对应的随机扩展树$Tree_{goal}$，假设$Tree_{start}$是第一随机扩展树$Tree_1$，$Tree_{goal}$是第二随机扩展树$Tree_2$，设定d为随机扩展树扩展代价；

（2）从$Tree_1$开始扩展，在构型空间中随机采样得到P_{rand}，在$Tree_1$节点中找到离P_{rand}代价最小的节点P_{near}，根据扩展代价d以及碰撞检测，获得新节点P_{new}，加入$Tree_1$并连线；若新节点不存在，则重新采样，直到新节点存在；

（3）将P_{new}作为$Tree_2$扩展的随机采样点，即P_{rand}'，在$Tree_2$节点中找到离P_{rand}'代价最小的节点P_{near}'，根据扩展代价d以及碰撞检测，获得$Tree_2$的新节点P_{new}'，加入$Tree_2$并连线。再以P_{rand}'作为$Tree_2$的随机采样点，以P_{new}'作为离P_{rand}'代价最小的节点，以扩展代价d向外扩展，不断获得新节点，加入$Tree_2$并连线，直至碰撞检测不通过或$Tree_1$、$Tree_2$相互连接。

（4）若碰撞检测不通过，则根据$Tree_1$与$Tree_2$节点数的比较，交换$Tree_1$与$Tree_2$，使总是是节点数较小的随机扩展树向节点数较多的随机扩展树扩展，然后重复步骤（2）。比如，若$Tree_2$节点数比$Tree_1$节点数，则交换$Tree_1$与$Tree_2$。

（5）若$Tree_1$、$Tree_2$相互连接，即$Tree_2$的新节点到$Tree_2$的随机采样点的距离小于扩展代价d，则停止扩展，将$Tree_2$的新节点到$Tree_2$的随机采样点相连，并合并两棵树即可。

RRT-CONNECT 通过两棵树在起始构型和目标构型区域的同时扩展大幅提高了空间扩展的效率，并且由于两棵树互相向对方扩展，方向性更加明显，优化效果更好。但同时 RRT-CONNECT 并没有彻底解决在同一空间附近重复经过的问题，仅仅通过有方向的扩展减少了重复经过问题的发生。所以有学者提出了 RRT-STAR 算法。RRT-STAR 算法主要思想是计算根节点到各节点的最小代价，并且通过寻找P_{new}附近代价半径为d_0（$d_0 > d$）超维球内的节点集，比较节点集中各节点本身的代价加上从该节点到P_{new}之和，选择最短的节点作为P_{new}的父节点，将两者相连，更新P_{new}的最小代价，然后再比较节点集中各节点本身的代价与先到P_{new}，再到该节点的代价大小，若经过P_{new}的路径代价更小，则更新该节点的最小代价，并将该节点的父节点改为P_{new}。由此可以在该随机扩展树中，找到相对次优的路径。其主要步骤如下：

（1）设起始构型为P_{start}，随机扩展树$Tree$，目标构型为P_{goal}，设定d为扩展代价，d_0

（$d_0 > d$）为超维球半径，设$g(P_i)$是从P_{start}到P_i的最小代价，$l(P_i, P_j)$是从P_i到P_j的代价；

（2）在构型空间中随机采样得到P_{rand}，在$Tree$节点中找到离P_{rand}代价最小的节点P_{near}，根据扩展代价d以及碰撞检测，获得新节点P_{new}，取位于代价半径为d_0超维球内部的节点$Pset\{i\}$，在通过碰撞检测后，取$g(Pset\{i\}) + l(Pset\{i\}, P_{new})$中最小的节点作为$P_{new}$的父节点，并将两者相连；在通过碰撞检测后，比较$g(Pset\{i\})$与$g(P_{new}) + l(P_{new}, Pset\{i\})$，若前者大，则更新$g(Pset\{i\}) = g(P_{new}) + l(P_{new}, Pset\{i\})$，并将$Pset\{i\}$的父节点更改为$P_{new}$，将两者相连。

（3）重复步骤（2），直至到达目标构型P_{goal}。

RRT-STAR 能够找到相对次优的路径，保证在代价半径为d_0的超维球构型空间中不会发生重复经过的情况，但其存在以下的缺陷：

（1）计算所消耗的时间较大，需要额外地遍历随机扩展树的节点，寻找代价半径为d_0超维球内部的节点，并且需要额外进行比较和加法运算；

（2）存储空间较大，需要额外存储各节点对应的最小代价；

（3）随机扩展树的扩展呈现无序状态，没有向目标构型扩展的趋势。

4.3 RRT-CS 与 RRT-CSD 算法

4.3.1 RRT-CS 算法

基于传统的 RRT 算法的优劣势，本文提出了 RRT-CS 与 RRT-CSD 两种改进的 RRT 算法，在机器人路径规划中具有很大的优势。

RRT-CS 算法相对于 RRT-CONNECT 在优化上有显著的提升，但计算效率低于 RRT-CONNECT；相对于 RRT-STAR 在计算效率上有显著提升，在优化上各有优势与劣势。其主要思想为分别在起始构型与目标构型分别生成随机扩展树，并且不断向对方的方向进行扩展，扩展的同时在新节点附近进行局部路径规划，直至两棵树完成连接。RRT-CS 能够快速生成相对次优的路径，其主要步骤如下：

（1）设起始构型为P_{start}，对应的随机扩展树$Tree_{start}$，目标构型为P_{goal}，对应的随机扩展树$Tree_{goal}$，假设$Tree_{start}$是第一随机扩展树$Tree_1$，$Tree_{goal}$是第二随机扩展树$Tree_2$，设定d为随机扩展树扩展代价，d_0（$d_0 > d$）为超维球半径，设$g(P_i)$是从P_{start}到P_i的最小代价，$l(P_i, P_j)$是从P_i到P_j的代价；

（2）从$Tree_1$开始扩展，在构型空间中随机采样得到P_{rand}，在$Tree_1$节点中找到离P_{rand}代价最小的节点P_{near}，根据扩展代价d以及碰撞检测，获得新节点P_{new}，取位于代价半径为d_0超维球内部的节点$Pset\{i\}$，在通过碰撞检测后，取$g(Pset\{i\}) + l(Pset\{i\}, P_{new})$中最小的节点作为$P_{new}$的父节点，并将两者相连；在通过碰撞检测后，比较$g(Pset\{i\})$与$g(P_{new}) + l(P_{new}, Pset\{i\})$，若前者大，则更新$g(Pset\{i\}) = g(P_{new}) + l(P_{new}, Pset\{i\})$，并将$Pset\{i\}$的父节点更改为$P_{new}$，将两者相连。设上述获得新节点节点$P_{new}$之后的操作为最小代价更新和父节点更新。若新节点不存在，则重复步骤（2）；

（3）将P_{new}作为$Tree_2$扩展的随机采样点，即P_{rand}'，在$Tree_2$节点中找到离P_{rand}'代价最小的节点P_{near}'，根据扩展代价d以及碰撞检测，获得$Tree_2$的新节点P_{new}'，加入$Tree_2$，然后进行最小代价更新和父节点更新。再以P_{rand}'作为$Tree_2$的随机采样点，以P_{new}'作为离P_{rand}'代价最小的节点，以扩展代价d向外扩展，不断获得新节点，加入$Tree_2$并进行最小代价更新和父节点更新，直至碰撞检测不通过或$Tree_1$、$Tree_2$相互连接。

（4）若碰撞检测不通过，则根据$Tree_1$与$Tree_2$节点数的比较，交换$Tree_1$与$Tree_2$，使总是是节点数较小的随机扩展树向节点数较多的随机扩展树扩展，然后重复步骤（2）。比如，若$Tree_2$节点数比$Tree_1$节点数，则交换$Tree_1$与$Tree_2$。

（5）若$Tree_1$、$Tree_2$相互连接，即$Tree_2$的新节点到$Tree_2$的随机采样点的距离小于扩

展代价d，则停止扩展，在$Tree_2$新节点处进行最小代价更新以及父节点更新，并以新节点为球心，以d_0为代价半径构造超维球，取包含在该超维球内部$Tree_1$、$Tree_2$的节点，并在属于$Tree_1$的节点和属于$Tree_2$的节点中各取一个，两者代价之和即为从起始构型到目标构型的代价，取代价之和最小的组合，将其相连。

4.3.2 RRT-CSD 算法

RRT-CS 算法相对于 RRT-CONNECT 在牺牲一定计算效率的情况下，能够显著减少代价消耗，更能达到优化目标。而相对于 RRT-STAR，能够显著提高计算效率，并且具有方向性，其代价消耗大概率少于 RRT-STAR，优化效果更佳。虽然 RRT-CS 相较传统 RRT 算法具有显著优势，但仍然存在一定的缺陷。主要体现在随机采样的无序性。这也是所有 RRT 路径规划算法存在的通病。随机采样的无序性将导致随机扩展树的无效节点的生成，严重影响路径规划及优化的算法效率。

本文提出解决随机采样的无序性的两种算法：

（1）目标构型方向法。目标构型方向法的主要思想是每次扩展生成多个随机采样点，然后生成每个随机采样点对应的最近点P_{near}和新节点P_{new}，得到P_{near}到P_{new}的方向向量$\overrightarrow{n_{rand}}$（为单位向量），取$\overrightarrow{n_{rand}}\cdot\overrightarrow{n_{goal}}$（$\overrightarrow{n_{goal}}$为$P_{near}$到$P_{goal}$）最大值对应的$P_{near}$和$P_{new}$作为本次扩展。

（2）启发式算法。启发式算法的主要思想同样是每次扩展生成多个随机采样点，然后生成每个随机采样点对应的最近点P_{near}和新节点P_{new}，只是随机采样点的选择标准有所区分。设$g(P_i)$是从P_{start}到P_i的最小代价，$l(P_i,P_j)$是从P_i到P_j的代价，$h(P_i)$是从P_i到P_{goal}的代价估计。取$g(P_{near})+l(P_{near},P_{new})+h(P_{new})$最小值对应的$P_{near}$和$P_{new}$作为本次扩展。$h(P_i)$的代价估计若小于等于实际代价，则选取的$P_{near}$和$P_{new}$总是最优的；若大于实际代价，则选取的$P_{near}$和$P_{new}$不一定最优。考虑到实际构型空间中总是存在障碍物，因此直接取$h(P_i)=l(P_i,P_{goal})$，即可保证估计代价小于实际代价，从而选取的P_{near}和P_{new}是最优的。

目标构型方向法当环境复杂的情况下，会严重影响路径生成效率，因为扩展方向可能大概率是错误的，陷入障碍物的死角，且没有考虑到已经消耗的代价，优化目标不够全面；启发式算法充分考虑已经消耗的代价以及未来的代价，具有显著的优化效果的提升，同时其方向性由于考虑到已经消耗的代价，因为大概率朝向目标构型扩展而陷入障碍物死角的情况发生的可能性大大减小，效率相较目标构型方向法更高。因此，选择启发式算法解决随机采样无序性问题，并开创 RRT-CSD 算法。其主要步骤如下：

（1）设起始构型为P_{start}，对应的随机扩展树$Tree_{start}$，目标构型为P_{goal}，对应的随机扩展树$Tree_{goal}$，假设$Tree_{start}$是第一随机扩展树$Tree_1$，$Tree_{goal}$是第二随机扩展树$Tree_2$，设定d为随机扩展树扩展代价，d_0（$d_0>d$）为超维球半径，设$g(P_i)$是从P_{start}到P_i的最小代价，$l(P_i,P_j)$是从P_i到P_j的代价，$h(P_i)=l(P_i,P_{goal})$；

（2）从$Tree_1$开始扩展，在构型空间中随机采样得到一组$Pset_{rand}\{i\}$，对每个随机采样点$Pset_{rand}\{i\}$在$Tree_1$节点中找到离其代价最小的节点P_{near}，根据扩展代价d以及碰撞检测，获得新节点P_{new}。在这一组P_{near}、P_{new}中选择$g(P_{near})+l(P_{near},P_{new})+h(P_{new})$最小值所对应的$P_{near}$、$P_{new}$。取位于以$P_{new}$为球心、代价半径为$d_0$超维球内部的节点$Pset\{i\}$，在通过碰撞检测后，取$g(Pset\{i\})+l(Pset\{i\},P_{new})$中最小的节点作为$P_{new}$的父节点，并将两者相连；在通过碰撞检测后，比较$g(Pset\{i\})$与$g(P_{new})+l(P_{new},Pset\{i\})$，若前者大，则更新$g(Pset\{i\})=g(P_{new})+l(P_{new},Pset\{i\})$，并将$Pset\{i\}$的父节点更改为$P_{new}$，将两者相连。设上述获得新节点节点$P_{new}$之后的操作为最小代价更新和父节点更新。若新节点不存在，则重复步骤（2）；

（3）将P_{new}作为$Tree_2$扩展的随机采样点，即P_{rand}'，在$Tree_2$节点中找到离P_{rand}'代价最小的节点P_{near}'，根据扩展代价d以及碰撞检测，获得$Tree_2$的新节点P_{new}'，加入$Tree_2$，然

后进行最小代价更新和父节点更新。再以P_{rand}'作为$Tree_2$的随机采样点，以P_{new}'作为离P_{rand}'代价最小的节点，以扩展代价d向外扩展，不断获得新节点，加入$Tree_2$并进行最小代价更新和父节点更新，直至碰撞检测不通过或$Tree_1$、$Tree_2$相互连接。

（4）若碰撞检测不通过，则根据$Tree_1$与$Tree_2$节点数的比较，交换$Tree_1$与$Tree_2$，使总是是节点数较小的随机扩展树向节点数较多的随机扩展树扩展，然后重复步骤（2）。比如，若$Tree_2$节点数比$Tree_1$节点数，则交换$Tree_1$与$Tree_2$。

（5）若$Tree_1$、$Tree_2$相互连接，即$Tree_2$的新节点到$Tree_2$的随机采样点的距离小于扩展代价d，则停止扩展，在$Tree_2$新节点处进行最小代价更新以及父节点更新，并以新节点为球心，以d_0为代价半径构造超维球，取包含在该超维球内部$Tree_1$、$Tree_2$的节点，并在属于$Tree_1$的节点和属于$Tree_2$的节点中各取一个，两者代价之和即为从起始构型到目标构型的代价，取代价之和最小的组合，将其相连。

4.4 基于随机采样的 A*算法

RRT 算法是在牺牲一定优化效果的情况下，保证路径规划的完备性的一种规划算法。A*算法是一种基于图搜索的路径规划算法，其能够保证生成一条最优的路径。当面临多维状态空间的路径规划问题时，A*算法的计算复杂度随着维度增加而呈指数型增加。但由于其路径优化的效果，本文开发了基于随机采样的 A*路径规划算法，提高了 A*算法在多维空间的适用性。

A*算法在多维空间应用的难点在于扩展方向与多维栅格的划分。A*算法主要应用于二维地图中，其前进方向多为“上下左右”四个方向或者再加上斜向四个方向，栅格主要通过将地图划分正方形格获得。在多维地图中，前进方向无法简单地通过类似“上下左右”来定义，若简单地沿某一维度前进来定义方向，容易发生多余的碰撞，且其优化效果较差，计算复杂度随着维度的增加而呈指数型上升；栅格的划分无法简单地根据多维空间在每一维度上的等分来获得，若根据维度等分来获得，则将消耗大量存储空间，增加计算复杂度；若根据固定的前进方向以及维度等分的栅格获得了一条路径，还需根据这条路径进行平滑优化，因为前进方向存在突变。基于随机采样的 A*算法的优势在于前进方向依据随机采样的方式确定，并且前进方向的数量可控，降低了计算复杂度的维度，无需进行栅格的划分，获得的路径突变处显著减少，简化平滑优化。其主要步骤如下：

（1）设起始构型为P_{start}，目标构型为P_{goal}，d为扩展代价（前进代价），d_0（$d_0 > d$）为超维球半径，设$f(P_i) = g(P_i) + h(P_i)$，$g(P_i)$是从P_{start}到P_i的最小代价，$l(P_i, P_j)$是从P_i到P_j的代价，$h(P_i)$是从P_i到P_{goal}的代价估计，openlist 为已经搜索过的节点集的外围序号，初始状态 openlist 中为P_{start}序号，；

（2）选取 openlist 中f最小的节点，作为扩展原点$P_{initial}$，同时在 openlist 中删除该节点序号。在构型空间中随机采样得到一组$Pset_{rand}\{i\}$，以构型空间中心点作为原点，确定随机采样点所对应的方向向量$\overrightarrow{n_{rand}\{i\}}$，由扩展原点、扩展方向$\overrightarrow{n_{rand}\{i\}}$、扩展代价以及碰撞检测，获得新节点集$P_{new}\{i\}$。对每一个$P_{new}\{i\}$，取位于以$P_{new}\{i\}$为球心、代价半径为$d_0$超维球内部的节点$Pset\{i\}$。对每一个通过碰撞检测的$Pset\{i\}$，计算$Pset\{i\} + l(Pset\{i\}, P_{new})$，作为$g(P_{new})$。

（3）重复步骤（2）直至$l(P_{initial}, P_{goal}) < d$，同样经过超维球最小代价检测与更新后，完成路径规划。其中不在 openlist 中的节点集即为路径节点集。

4.5 路径规划算法性能对比

本文使用已有的 RRT 算法以及本文所开发的三种基于随机采样的路径规划算法，解决同一路径规划任务，对比各类算法的性能。

4.5.1 路径规划算法部分性能的提升

本文所实现和开发的几种路径规划算法，部分方面的性能呈现递进式的提升。本小节主要以图示的方式直观地介绍性能提升的效果。

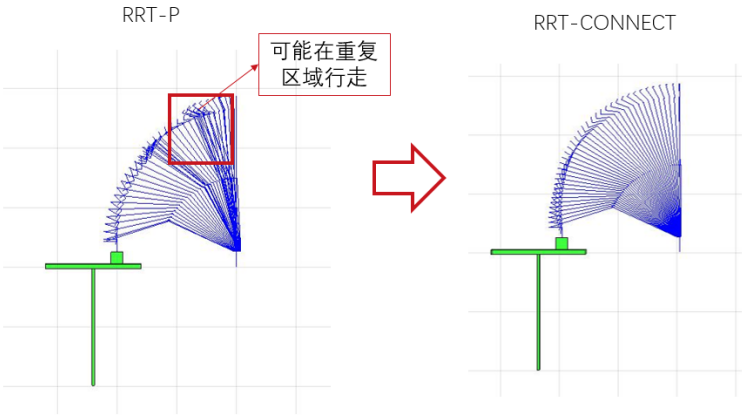


图 4-1 RRT-P 与 RRT-CONNECT

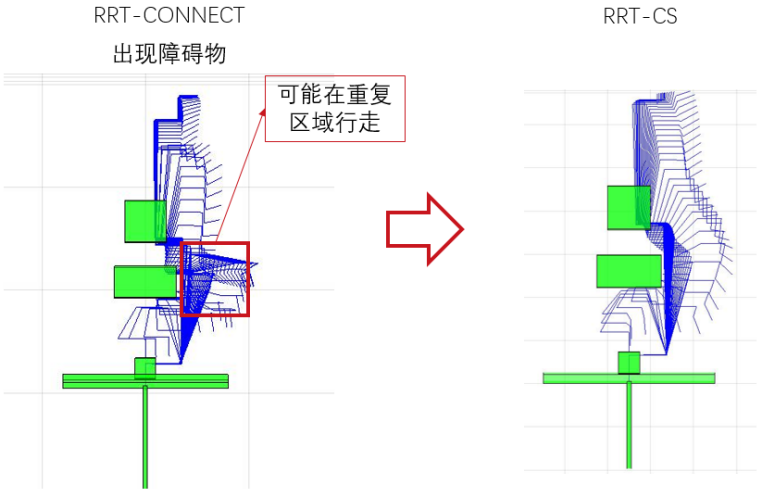


图 4-2 RRT-CONNECT 与 RRT-CS

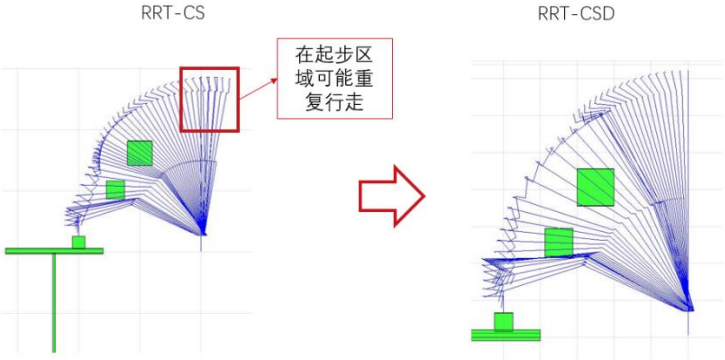


图 4-3 RRT-CS 与 RRT-CSD

4.5.2 路径规划算法优劣对比

本文选择两种不同的路径规划任务，分别采用 RRT-P、RRT-CONNECT、RRT-STAR、RRT-CS、RRTCSD 和基于随机采样的 A*算法实现 UR10 机械臂末端执行器从初始位置到待抓取物体正上方，对比其路径规划用时以及代价。路径规划优化目标为用时最短，具体衡量指标定义如下：由路径规划得到一组构型P_1、P_2……P_{n-1}、P_n，定义代价$\mathrm{W} = \sum_{i=1}^{n-1} ||P_i -$

P_{i+1}||（||·||表示二范数）。定义用时超过 120s 为路径规划失败。

（1）针对障碍物较少的场景（桌子和待抓取物体同样属于障碍物），如图 4-4，分别使用不同的路径规划算法，重复 10 次，取用时及代价的平均数作为性能衡量指标，得到表 4-1。

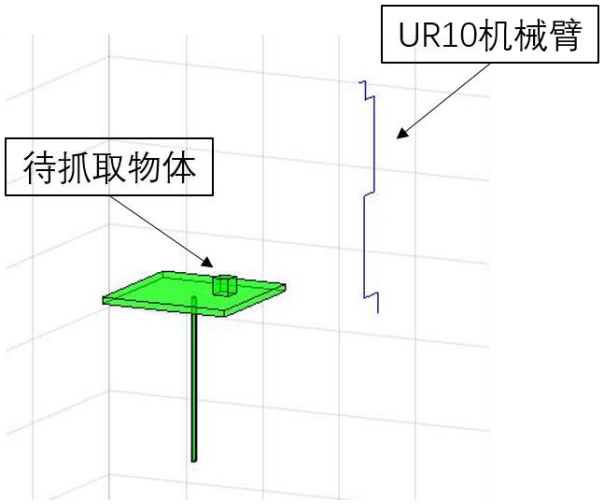


图 4-4 任务一：障碍物较少

表 4-1 任务一：各路径规划算法性能对比表

	用时/s	代价/°
RRT-P	1.678	253.12
RRT-CONNECT	0.633	212.79
RRT-STAR	5.112	225.66
RRT-CS	1.137	201.91
RRT-CSD	1.155	202.05
基于随机采样的 A*算法	61.319	440

（2）针对障碍物较多的场景，如图 4-5，分别使用不同的路径规划算法，重复 10 次，取用时及代价的平均数作为性能衡量指标，得到表 4-2。

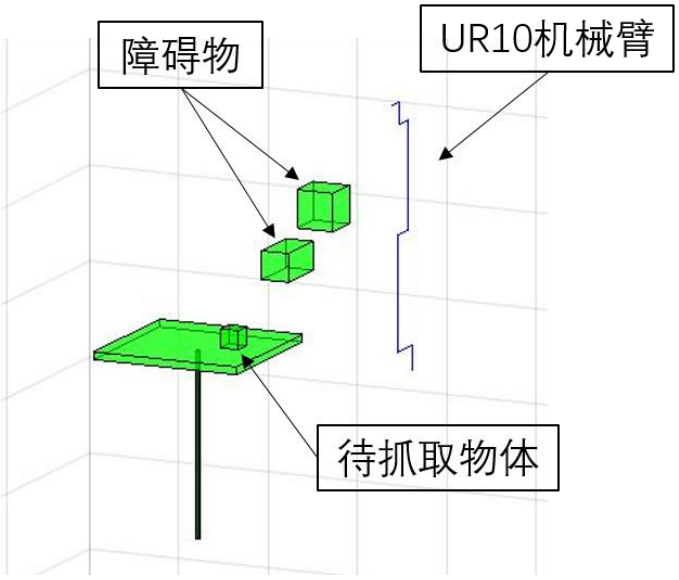


图 4-4 任务二：障碍物较多

表 4-2 任务二：各路径规划算法性能对比表

	用时/s	代价/°
RRT-P	*	*
RRT-CONNECT	2.756	283.65
RRT-STAR	*	*
RRT-CS	8.652	258.52
RRT-CSD	13.587	235.79
基于随机采样的 A*算法	*	*

注：*表示路径规划失败

从上述各路径算法的对比，以及 4.5.1 的直观展示，可以得出以下结论：

（1）在障碍物较少的情况下，RRT-CONNECT 路径规划效率最高，且代价较少；RRT-CS 与 RRT-CSD 两者性能差别不大，在规划效率上略低于 RRT-CONNECT，但它们的代价少于 RRT-CONNECT；RRT-P 在效率与代价上均没有显著的优势；RRT-STAR 规划效率比较低，且代价也较高，这是因为 RRT-STAR 是在 RRT 基础上改写的，其扩展的无序性导致了规划效率的低下，但其代价相对于 RRT-P 更少；基于随机采样的 A*算法在各项性能上均较差，主要原因是通过随机采样方式栅格划分以及扩展，不具完备性，得到的路径非最优。

（2）在障碍物较多的情况下，RRT-P、RRT-STAR、基于随机采样的 A*算法耗时均超过 120s，规划效率很差；在障碍物较多的情况下，RRT-CS 与 RRT-CSD 显示出其在路径优化上的强大性能，其代价消耗明显低于 RRT-CONNECT，而 RRT-CSD 由于对随机采样有了一定的选择，其大概率能够规划一条比 RRT-CS 更优的路径，但其牺牲了一定的规划效率；RRT-CONNECT 具有最好的规划效率。

（3）RRT-CONNECT 虽然具有计算效率极高的优点，但从规划结果上看，大概率存在在某一区域附近来回重复行走的现象，这对工业机器人路径平滑、减少代价等要求不符；RRT-CS 与 RRT-CSD 在牺牲一定计算效率的情况下，能够减少这类问题的发生，在实际工业机器人路径规划中具有很大的优势。

第五章 最优抓取姿态

5.1 抓取姿态与路径规划

工业机器人在确定目标构型从而进行路径规划时，会发现存在多个目标构型均能完成目标任务。因此，选择最优的抓取姿态，从而使路径规划的效率提升或代价减少，具有重要的经济意义和现实意义。

本文所采用的抓取姿态优化方法主要根据用时最短的优化目标，确定影响优化目标的影响因素$||P_{start} - P_{goal}||$（P_{start}为初始构型，P_{goal}为终止构型，$||\cdot||$表示二范数），通过抓取姿态与P_{goal}的对应关系，进行抓取姿态的优化。

5.2 抓取姿态的优化

本文所研究的任务是垫片的分拣，本课题设计的抓取垫片的机械手为二指型，抓取时机械手位于垫片正上方。由于垫片呈圆环形，机械手完成抓取任务只需保证机械手轴向与垫片

法向平行，且距离在一定范围内即可，机械手沿自身轴向旋转角度不影响抓取任务的实现。因此，完成抓取垫片任务所对应的末端执行器，即机械手的姿态存在多解。路径规划的优化目标是用时最短，即代价$\mathrm{W}=\sum_{i=1}^{n-1}||P_i-P_{i+1}||$（$||\cdot||$表示二范数，$P_1$、$P_2$……$P_{n-1}$、$P_n$为路径规划得到的一组构型）。$||P_{start}-P_{goal}||$（$P_{start}$为初始构型，$P_{goal}$为终止构型）直接影响W大小。优化流程如下：

（1）由垫片位置$[x,y,z]^T$，可以获得对应的可能的抓取姿态$[x,y,z+d,\pi,0,\theta]^T$（d为机械手执行抓取任务时末端执行器所在坐标系原点与垫片在z方向上的距离，θ表示未知角度）。

（2）由抓取姿态$[x,y,z+d,\pi,0,\theta]^T$，可计算末端执行器对应的旋转矩阵R和坐标系原点位置p，由此可以计算出齐次变换矩阵${}_6^0T=\begin{bmatrix}R & p\\ 0 & 1\end{bmatrix}$，然后通过逆向运动学获得对应的目标构型$P_{goal}$，从中选择$||P_{start}-P_{goal}||$最小值$d(\theta)$。

（3）在$\theta\in[0,\pi]$，中选择使$d(\theta)$最小的θ值以及对应的逆向运动学的解，作为最终的目标构型P_{goal}。

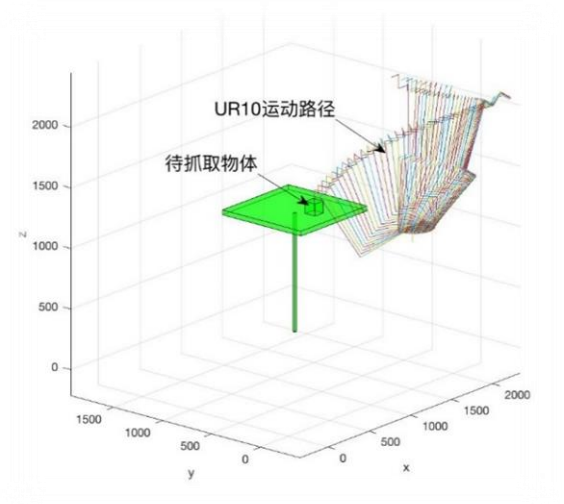


图 5-1 一种抓取姿态对应的路径规划结果

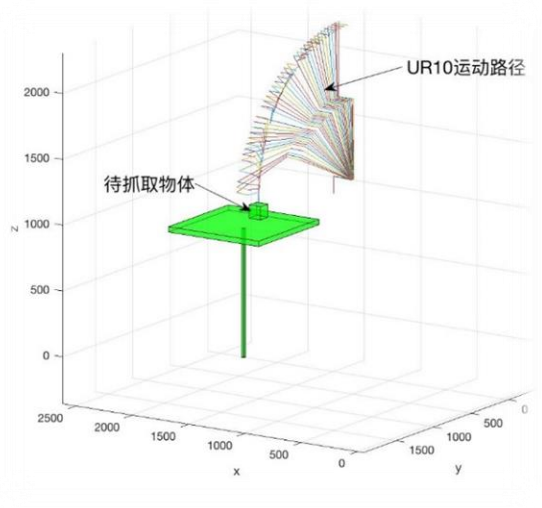


图 5-2 抓取姿态优化后路径规划结果

从路径规划结果上看，抓取姿态的优化能够显著提升路径规划的效果。

第六章 工业机器人路径规划实验

6.1 工业机器人路径规划系统简介

本文搭建了工业机器人基于 Kinect 传感器以及 Matlab 软件平台的工业机器人路径规划系统，实现了 UR10 机械臂执行垫片分拣任务中避障抓取并优化用时的路径规划，并能够控制 UR10 机械臂执行分拣任务。其系统流程框图如下：

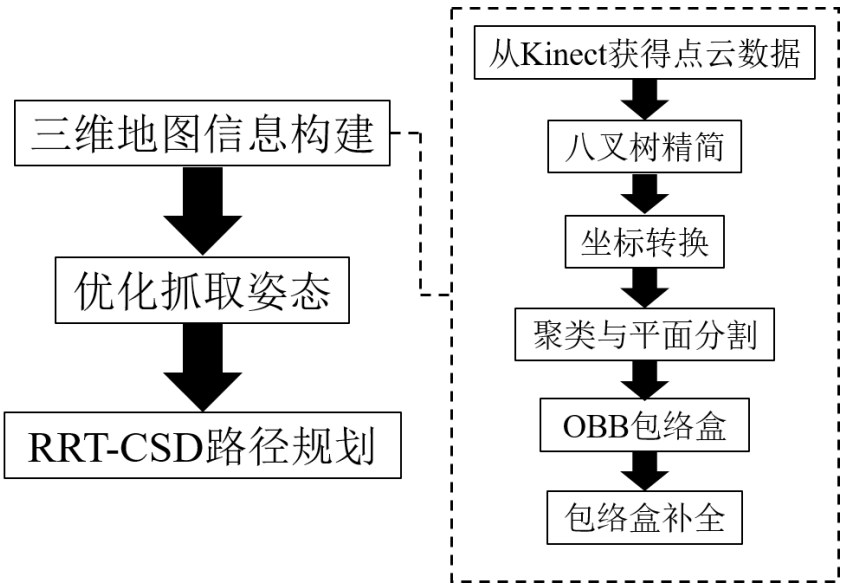


图 6-1 工业机器人路径规划系统流程框图

该系统相较传统的示教式路径规划算法，具有调试快、用时短、操作简单、路径规划结果好的优势。系统的设计及算法的选择和改进，充分考虑了实际工业场景，具有一定的工业应用价值。

6.2 路径规划系统实验

本文基于所研究的具体工业任务，即操作 UR10 机械臂实现垫片的分拣，使用该工业机器人路径规划系统，进行实验。执行任务目标为操作 UR10 机械臂将不同尺寸大小的垫片抓取置对应的纸盒中，与桌面上的障碍物不发生碰撞，并且尽量使任务时长最短。任务场景如图 2-7。

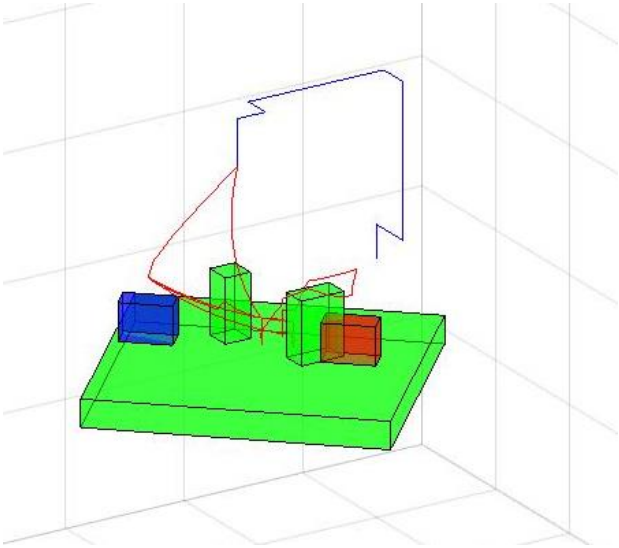


图 6-2 路径规划结果。红线表示机器人末端执行器轨迹。

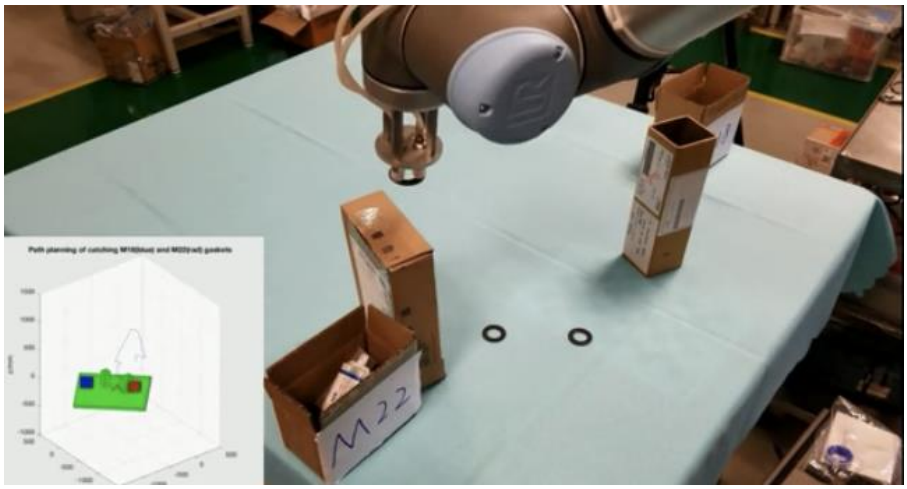


图 6-3 实际控制 UR10 机械臂实现分拣任务。
图示为抓取第一个垫片，并越过障碍物。

图 6-2 为对该任务路径规划的结果，红线表示机器人末端执行器轨迹。机器人末端执行器并不指机械手与垫片接触的位置，而是处于偏上一点的位置。这是为了保证机器人能够完成抓取任务而保留的一定冗余。图 6-3 是实际控制 UR10 完成垫片分拣任务的截图。图示为 UR10 抓取垫片并越过第一个障碍物。左下角为对应的路径规划对应的位置。

由图 6-2 与图 6-3 所示，以及多次执行其他任务的情况结果看，本文所设计的工业机器人路径规划系统具有良好的适用性，相对于示教法具有相当的优势，提高了工业机器人的智能水平。

第七章 总结与展望

7.1 研究总结

工业机器人在路径规划领域仍然以示教为主。如何快速准确地规划一条无碰撞、且尽可能满足工业需要（如耗时最短等）的路径，使工业机器人迅速投入生产实践，具有重要的研究意义，同时也面临着很多挑战。本文基于 Kinect 传感器，针对 UR10 机械臂执行抓取、分拣垫片的生产任务，完成了三维地图构建、机械臂运动学求解、路径规划、抓取姿态优化等工作，实现了控制 UR10 抓取指定垫片的任务。主要工作分为以下几个方面：

第一，实现了基于 Kinect 传感器获得的点云数据构建三维仿真环境。通过设定 Kinect 获得的点云数据的坐标系与 UR10 机械臂末端执行器位置和姿态的对应关系，将三维地图信息转换到 UR10 坐标系，完成三维地图信息的构建。对比了多种障碍物探测和描述算法的优劣，设计了一套兼顾计算效率与障碍物信息保真性的仿真环境构建算法，实现从点云数据到 UR10 坐标系下的障碍物信息的构建算法。

第二，实现了 UR10 机械臂正向运动学和逆向运动学的求解。根据 UR10 机械臂实际尺寸参数，通过 DH 坐标法，完成 UR10 机械臂正向运动学求解，同时通过 DH 坐标系的正确选择，简化逆向运动学解析解求解的计算步骤，完成逆向运动学求解，并提出一定的求解策略。

第三，实现了多种基于随机采样的路径规划算法，开发了三种新型路径规划算法，对比其性能。通过实现并比较传统的路径规划算法及其改进算法，开发出 RRT-CS、RRT-CSD、基于随机采样的 A*算法，前两者在面对多障碍物的复杂环境中，其优化效果明显优于传统的 RRT-CONNECT 算法。通过仿真测试，RRT-CSD 具有更好的综合性能。

第四，实现了对抓取姿态的优化。针对本文所研究的特定任务，选择直观清晰的优化目标，实现了对抓取姿态的优化。结果表明，抓取姿态的优化对路径规划效果有显著提升。

第五，完成了从点云数据到路径规划的算法流程搭建，其输入为垫片位置，其输出为 UR10 机械臂路径，能够控制 UR10 机械臂在不碰撞障碍物的情况下，实现抓取任务。

7.2 主要贡献与创新点

本文的主要贡献如下：

（1）针对工业机器人路径规划的需求，搭建了基于单一 Kinect 传感器的路径规划智能算法系统，其所需参数少，规划用时短，优化效果好，适用能力强，各项性能优于传统示教法，具有一定的工业应用价值。

本文主要创新点如下：

（1）开发了 RRT-CS 和 RRT-CSD 两种综合性能较好的新型路径规划算法。其对多自由度机器人的路径规划具有良好的适用性，面对复杂的环境具有显著的计算效率以及优化效果上的优势。

7.3 研究展望

由于作者的时间和学识有限，所获得的研究成果仍然有许多可以改进的地方，在广度和深度上仍然有较大的拓展空间。基于本文所涉及的研究内容，未来工作的展望主要有以下几

个方面：

（1）三维地图信息构建有待进一步提升

本文所述的三维地图信息构建算法所适用的环境结构较为简单，对复杂环境适用性不高。由于所采用的 Kinect 传感器仅有一台，所获得的点云数据不足以完整描述障碍物信息。所使用的 AABB 包络盒，存在空隙较大的问题。可以考虑使用多台 Kinect 传感器，将点云数据转换到同一个坐标系下，从而获得完整的三维地图信息；可以深入研究各包络盒以及直接使用八叉树直接进行障碍物建模的优劣，并对其进行改进。

（2）抓取顺序的优化

本文所研究的任务是垫片的分拣，多个垫片的分拣涉及到抓取顺序的问题。如何选择抓取的顺序，从而使整个分拣过程的用时最优，涉及到多个领域的交叉和融合。

参考文献

[1] Zhang Q, Ma J C. Liu Q. Path planning based quadtree representation for mobile robot using hybrid-simulated annealing and ant colony opimization algorithm[C]. Proceedings of the World Congress on Intelligent Control and Automation. Beijing: IEEE Press, 2012: 2537-2542

[2] LaValle S M and Kuffner J J. Randomized kinodynamic planning [J]. The International Journal of Robotics Research, 2001, 20(5): 378-400.

[3] Kuffner J J and LaValle S M. RRT-Connect: an efficient approach to single-query path planning [C]. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, 2000: 995-1001

[4] Kacraki L, Svestka P, Latombe J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces [J]. IEEE Transactions on Robotics and Automation, 1996, 12(4): 566-580.

[5] Reif J H. Complexity of the mover's problem and generalizations [C]. Proceedings of the 20th Annual Symposium on Foundations of Computer Science, 1979: 421-427.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths in graphs. IEEE Trans. Syst. Sci. and Cybernetics, SSC-4(2):100-107, 1968

[7] Stentz A. The focussed D^* algorithm for real-time replanning[C]//IJCAI. 1995, 95: 1652-1659.

[8] 辛煜. 无人驾驶车辆运动障碍物检测、预测和避撞方法研究[D].中国科学技术大学,2014.

[9] 王荣本,李琳辉,金立生,郭烈,赵一兵.基于双目视觉的智能车辆障碍物探测技术研究[J].中国图象图形学报,2007(12):2158-2163.

[10] 霍艳艳,黄影平.基于立体视觉和光流的障碍物探测方法[J].信息技术,2013,37(01):125-127.

[11] 崔坤征,罗均,谢少荣,龚振邦.一种用于动态障碍物探测的超声波系统的研制[J].机电工程,2005(06):44-48.

[12] 赵亮. 基于 Kinect 的障碍物探测及三维场景重建研究与实现[D].西安科技大学,2015.

[13] 张琦. 移动机器人的路径规划与定位技术研究[D].哈尔滨工业大学,2014.

[14] 杨扬. 基于机器视觉的服务机器人智能抓取研究[D].上海交通大学,2014.

[15] 任忠杰. 基于点云模型虚拟手术训练系统的碰撞检测及力反馈算法研究[D].南昌大学,2015.
[16] 莫建文,芦爱余,张彤.结合图像分割和点云分割的障碍物检测算法[J].计算机工程与设计,2015,36(07):1855-1858+1907.
[17] 张国飚,张华,刘满禄,余慧.基于空间剖分的碰撞检测算法研究[J].计算机工程与应用,2014,50(07):46-49+55.
[18] 彭飞. 约束条件下的船舶装配拆卸随机采样路径规划研究[D].华中科技大学,2013.
[19] 张建英,赵志萍,刘暾.基于人工势场法的机器人路径规划[J].哈尔滨工业大学学报,2006(08):1306-1309.
[20] 马登武,叶文,李瑛.基于包围盒的碰撞检测算法综述[J].系统仿真学报,2006(04):1058-1061+1064.
[21] 朱庆保,张玉兰.基于栅格法的机器人路径规划蚁群算法[J].机器人,2005(02):132-136.
[22] 张颖,吴成东,原宝龙.机器人路径规划方法综述[J].控制工程,2003(S1):152-155.
[23] 蔡自兴. 机器人学[M]. 北京: 清华大学出版社, 2000.

谢辞

感谢学校丁烨导师和各企业导师在研究过程以及本文撰写中的指导，提供了很多宝贵的意见。感谢丁烨导师、徐凯老师、吴建华老师在我们中期答辩时给予我们的经验和教训。感谢全体组员的辛勤付出和合作，正是我们的相互帮助、理解，才让我们获得了“最佳展示奖”的荣誉，让每个人有所收获。

最后感谢学校给我们一个良好的科研环境，一个校企合作的平台，培养我们的科研素养，提升我们的科研能力。

STUDY ON INDUSTRIAL ROBOT PATH PLANNING ALGORITHM

With the introduction of "Industry 4.0", enterprises are increasingly demanding smart manufacturing and acquire diverse functions. Industrial robots at this stage are gradually unable to meet market demands, and in particular, higher requirements are placed on robot path planning. The path planning of industrial robots refers to the planning of a path from the starting position to the ending position, reaching a certain goal (such as the minimum path length, the shortest time, the least cost, etc.) and avoiding obstacles. At present, teaching is main method of industrial robot path planning. Robot teaching refers to operate robot, storage robot movement processes, and reproduce the movement process. When the movement process of the robot is complex or there are many obstacles in the working space, robot teaching takes a long time and the labor cost is high; when the task for robot to perform changes, robot needs to be taught again. As the demand for production efficiency continues to deepen, higher requirements are placed on path planning. In robot teaching, the selection of the shorter path can only be judged intuitively by humans. The path obtained by robot teaching often consumes a lot of costs (such as time, energy, etc.). The biggest disadvantage of robot teaching is that when an unsuccessful operation occurs, for example, the robot collides with an obstacle, it is necessary to trace back to the previous operation, delete the retained part of the movement processes, and then teach again. When the environment is more complex, such as automobile assembly and welding, those downsides can be more obvious and the cost of robot teaching will increase dramatically. This paper carries out gasket sorting task for UR10 manipulator and designs a set of general-purpose algorithms for path planning of industrial robots. The work of this article mainly includes the following aspects:

(1) According to the point cloud data, an octree-based hierarchical envelope box method is used to construct three-dimensional map information. The octree tree method divides the cube space into eight subcubes and encodes the smallest cubes. All the point cloud data are located in a certain smallest subscube, and they can be expressed as the centroid of the subscube they are located. So, the point cloud data from Kinect sensor can be simplified. After using the octree method to reduce the point cloud data, the point cloud coordinate is transferred to the robot coordinate system through the homogeneous transformation matrix, and then the clustering algorithm clusters the point cloud data into different obstacles. And then, a hierarchical envelope box method is used to create a minimal envelope that can encompass each set of point clouds representing obstacles. Because there is only one Kinect, and the actual industrial scenarios often cannot meet the requirement of arranging multiple sensors to construct entire three-dimensional map information, the obtained point cloud data are incomplete, hardly rebuilding obstacles completely. In this paper, by measuring the actual size of the obstacles, the generated envelopes are compensated, and finally the envelopes that can surround the real obstacles are obtained, thereby improving the fidelity of the obstacle information. The entire set of algorithmic systems takes into account the computational efficiency and the fidelity of the obstruction information, which has certain advantages in industrial applications. Path

planning problems inevitably involve collision detection problems. The collision detection efficiency between the point and the envelope is much higher than the collision detection efficiency between the envelope and the envelope. Therefore, in this paper, the axes of the UR10 manipulator is used as a collision detection model, and the volume envelop is expanded in turn. Collision of the axes and expanded envelopes can simulate actual collision between the UR10 and the obstacles, which greatly improves the collision detection efficiency.

(2) The forward kinematics and inverse kinematics analysis of the UR10 manipulator are performed. In this paper, forward kinematics analysis is executed by DH coordinate method. According to the actual size parameters of UR10, a DH parameter table is established to realize the forward kinematic analysis of the UR10 robot arm. Because the UR10 manipulator meets the criteria of Pieper, which is to say the three adjacent joint axes intersect at a point or their axes are parallel, an analytical solution to the inverse kinematics exists. In this paper, by establishing a suitable DH coordinate system, the computational steps for solving the inverse solution are simplified, and the inverse kinematics analysis of the UR10 manipulator is completed. The solution to the analytical solution of the inverse kinematics of the n-degree-of-freedom robot is based on the correct selection of the DH coordinate system, which means there is a constant term in the homogeneous transformation matrix ${}_{n-1}^{0}\boldsymbol{T}$, so that one variable can be quickly solved and the solution step can be simplified. .

(3) Traditional path planning algorithms based on random sampling is implemented. Three new path planning algorithms, RRT-CS, RRT-CSD and A* algorithm based on random sampling, have been developed and tested. Most industrial robots are multi-degree-of-freedom robots, and their path planning goals are generally the minimum time cost. The stroke length in an industrial robot's state space determines the time spent, not the stroke length of the end effector. Therefore, path planning needs to be performed in the robot state space. Traditional path planning algorithms, such as Dijkstra algorithm, A* algorithm, intelligent algorithm, etc., are suitable for low-dimensional space, and they will fall into dimension disaster in high-dimensional space. The random sampling-based path planning method and its optimization algorithm significantly increase the efficiency of path planning in multi-dimensional space, though they have negative effect on optimization. This paper implements and tests existing path planning algorithms based on random sampling, such as Rapidly-exploring Random Tree, RRT-P, RRT-CONNECT, and RRT-STAR. RRT-CONNECT enhances the directionality of random tree expansion, and it has high computational efficiency in a simple environment. RRT-STAR optimizes the branches and leaves of random trees to generate a suboptimal path, which improves the optimization effect at the expense of computational efficiency. This paper presents the RRT-CS algorithm, which combines the advantages of both RRT-CONNECT and RRT-STAR algorithms, taking into account both the computational efficiency and the optimization effect. RRT-CS, like existing RRT algorithms, has the problem of sampling's disorder. This paper proposes two solutions to solve random sampling disorder. The first is to randomly sample multiple points at one expanding time and to select the point whose expanding direction is closer to the target configuration P_{goal} to extend. The second method is a heuristic algorithm. Each expanding, multiple points are randomly sampled, and then corresponding extension points are generated. Each extension point can generate cost function. Choose the smallest value of the cost function, and expand the tree. The cost function of extension point P_i is defined as $f(P_i) = g(P_i) + h(P_i)$. $g(P_i)$ is the minimum cost from the initial configuration P_{start} to P_i. $h(P_i)$ is the estimated cost from P_i to P_{goal}. When the environment is complex, the first method

will seriously affect the efficiency of route generation, because the direction of expansion may be wrong, and expanding will fall into the blind corner of the obstacles. Besides, it does not consider the cost consumed, and the optimization goal is not comprehensive enough; The heuristic algorithm fully considers the cost consumed and the cost of the future, has a significant improvement in the optimization effect, and at the same time it decreases the probability of expanding into an obstacle blind situation for its directionality takes into account the cost consumed, which has higher efficiency than the first method. Therefore, the heuristic algorithm is chosen to solve the problem of random sampling disorder, and the RRT-CSD path planning algorithm is formed. A* algorithm based on random sampling is to apply A* algorithm to multi-dimensional space path planning avoiding dimensional disasters through random sampling. Through the path planning test for the same task, RRT-CSD has the best overall performance and is more applicable in complex environments. Under the condition of ensuring the calculation efficiency, the optimization effect is significant.

(4) The influence of different grab gestures on the path planning results is studied and optimized. The grab posture corresponding to the minimum two-norm value between the target configuration of the different grab postures and the initial configuration is selected, and path planning is performed. The result of the optimization is significantly improved for path planning.

The three-dimensional map information construction, forward kinematics and inverse kinematics analysis of industrial robots, path planning algorithms, and grasping posture optimization in this paper covers the major technical issues involved in the path planning of industrial robots. Research results of this paper can solve path planning problems of industrial robots in practice efficiently, and have certain advantages in time cost, labor cost, and economic cost. The main contribution of this paper is to propose several new path planning algorithms, their overall performances such as computational efficiency and optimization effect are better than traditional path planning algorithms. Research results of this paper can reduce the cost of layout and debugging of industrial robots to some extent, and can expand the application of industrial robots.