

分类号_____

编 号_____

U D C_____

密 级_____



南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

本科生毕业设计（论文）

题 目： 智能机器人抓取仿真系统

姓 名： 周兴雨

学 号： 12012004

院 系： 计算机科学与工程系

专 业： 计算机科学与技术

指导教师： 杨剑 史玉回

2024 年 6 月 7 日

诚信承诺书

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。
2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。
3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。
4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名：

周兴雨

2024 年 6 月 7 日

智能机器人抓取仿真系统

周兴雨

(计算机科学与工程系 指导教师：杨剑 史玉回)

[摘要]：智能机器人在工业生产、物流仓储、医疗护理等领域扮演着重要角色。机器人抓取技术作为其中关键的一环，需要在不同场景下实现准确抓取目标物体的能力。机器人抓取是一项高精度要求的技术，在正式使用前需要进行大量测试，以提前验证抓取算法的正确性，减少正式使用的成本与风险，因此使用仿真系统对算法的正确性进行测试尤为重要。为此，本文以 Pybullet 平台为基础设计了一个智能机器人抓取仿真系统，实现了仿真环境下机械臂的控制。该系统还实现了可操作的控制界面和自定义场景功能，提供了丰富的用户交互模块，方便用户进行抓取测试。同时，本文还设计了一个简单抓取实验，测试 REGNet 这一抓取位姿估计算法，以证明本仿真系统的测试能力。实验结果表明，本仿真系统能够有效适配机器人抓取相关算法，为机器人抓取相关研究的测试工作提供了一定便利。

[关键词]：机器人抓取；仿真；算法测试；Pybullet；REGNet

[ABSTRACT]: Intelligent robots play crucial roles in various fields such as industrial production, logistics warehousing, and medical care. Robot grasping, as a key component, requires the ability to accurately grasp target objects in different scenarios. Robot grasping is a highly precise technique that necessitates extensive testing before formal deployment to validate the correctness of grasping algorithms and reduce costs and risks associated with formal usage. Therefore, this paper designs an intelligent robot grasping simulation system based on Pybullet, achieving control of robotic arms in a simulated environment. The system also provides an interactive control interface and customizable scene functions, offering user interaction modules for testing. Additionally, a grasping experiment is designed to test the REGNet grasping pose estimation algorithm, demonstrating the testing capability of this simulation system. Experimental results indicate that this simulation system effectively adapts to robot grasping-related algorithms, providing certain convenience for testing work in robot grasping-related research.

[Keywords]: robot grasping; simulation; algorithm test; Pybullet; REGNet

目录

1. 绪论	1
1.1. 背景及研究意义	1
1.2. 国内外研究现状	3
1.2.1. 开发平台	3
1.2.2. 仿真系统及算法仿真测试	4
1.3. 论文结构	5
2. 研究内容	6
2.1. 系统设计及工作流程	6
2.2. 系统功能概况	7
2.2.1. 动作检测算法	7
2.2.2. 机械臂控制	9
2.2.3. 抓取场景控制及自定义	10
2.2.4. 虚拟相机成像	12
3. 抓取实验	13
3.1. 实验设计	13
3.2. 实验结果及分析	14
4. 总结与展望	16
参考文献	16
致谢	17

1. 绪论

1.1. 背景及研究意义

通常来说，机器人是一种模仿人类行为模式进行工作的机器，自 20 世纪中叶以来，机器人技术一直处于不断演进的过程中。最初的工业机器人主要用于重复性任务，但随着时间的推移，机器人的功能不断扩展，涵盖了更多领域，包括医疗、军事、物流等。在机器人技术的诸多应用中，机器人抓取技术无疑是至关重要的一环。机器人抓取技术是指使机器人能够根据视觉信息，自动地识别、定位和抓取物体的技术，它是机器人学、计算机视觉和人工智能等多个领域的交叉研究课题，具有重要的理论价值和实际意义。机器人抓取技术可以应用于工业生产、物流仓储、医疗护理、家庭服务等多个场景，提高生产效率、降低人力成本、改善人类生活质量，机器人的抓取能力直接影响其在各种任务中的表现和效率。然而，要实现有效的抓取，需要克服许多挑战，如视觉感知、物体识别、运动规划等。

机器人抓取作为一项高精度要求的技术，在正式使用前需要进行大量测试，测试方式分为实景测试和仿真测试两种，其中实景测试在工业应用之前必须大量进行，但是真实机器人的数据收集十分繁琐且复杂，需要大量成本[1]。仿真测试虽然不能提供如实景测试一般最为可靠的数据[2]，但是其收集相对于真实机器人数据而言更为简便，而且在算法测试中，仿真测试相比于实景测试具有一定的优越性：

首先，仿真环境能够提供多样化的场景和目标物体，从简单的几何体到复杂的真实物体，模拟或真实的环境和机器人的建模，包括执行器，传感器和触点的建模[3]，都可以在仿真中进行模拟，可以在不同情景下评估抓取算法的适用性和鲁棒性，而无需实际物体。其次，仿真环境具备快速调整的能力。当需要改进抓取算法或调整机器人参数时，可以直接在仿真环境中进行，无需等待实际物体和场地的准备，加快开发周期，使得算法和机器人系统可以更快地响应需求和改进。第三，仿真环境提供了精确的物理模拟和传感器模拟，仿真平台可以准确地模拟物体的形状、质地和物理特性，以及传感器的工作原理和噪声，在仿真中进行更真实的抓取模拟和传感器数据生成，为算法和系统的验证提供了可靠的基础。

此外，仿真环境还能够模拟各种不同的情况和挑战，如光照变化、物体遮挡、动态障碍物等，这有助于评估抓取算法的鲁棒性和可靠性。在仿真环境中进行这些测试，可以更安全地探索各种极端情况，避免了对真实系统的潜在损坏或人员安全的风险。同时，在应用仿真测试之后，如果需要现实应用，还可以利用仿真测试的数据进一步进行实景测试，得到真实场景中更为可靠的数据。

机器人抓取技术具有丰富的现实应用场景，需要智能机器人能够处理各种复杂情况的抓取，并针对抓取物体的形状、质量、位姿等信息调整抓取的方式。针对智能机器人抓取的现实运用需求，G. Du, K. Wang 等人将智能机器人抓取分解为三个问题：抓取方式和估计、目标定位和目标位姿估计[4]。

目前智能机器人的抓取主要使用二维平行夹持器进行研究[5]，抓取方式主要可分为两种：二维平面抓取和六自由度抓取。二维平面抓取是指目标物体位于平面工作空间上，抓取只受一个方向的约束。在这种情况下，夹持器的高度是固定的，夹持器的方向垂直于一个平面。六自由度抓取指的是夹持器可以自由地变换 $x, y, z, \text{roll}, \text{pitch}, \text{yaw}$ 这六个坐标，使得机器人能够以更丰富的姿态对不同位姿的物体进行抓取[6]。

抓取估计是指在相机坐标中估计 6D 抓取器的姿态。如前所述，抓取可以分为二维平面抓取和 6DoF 抓取。对于 6DoF 抓取，抓取器可以从不同的角度抓取物体，6DoF 抓取器的姿态是进行抓取的必要条件，因此在抓取前需要对抓取的姿态进行估计[7]。

在执行抓取之前，智能机器人需要先通过传感器图像识别抓取物体和夹爪的相对位置，进行目标定位。目标定位具有三种精度等级：不分类的目标定位、目标检测和目标实例分割。不分类的目标定位表示只用边界框分隔出各个对象，而不关心每个对象的形状、大小等属性，是最低精度的目标定位；目标检测的任务是检测某一类对象的实例，可以看作是定位任务加分类任务，提供目标对象及其类别的边界框，但无法精确描述物体的形状，且无法分辨同一种类对象的不同个体，目标检测是许多其他计算机视觉任务的基础[8]；目标实例分割在目标检测的基础上，进一步提供目标对象的像素级或点级区域及其类别，同时将同类别的不同对象也进行区分[9]，是最高精度的目标定位，满足 6DoF 抓取的要求。

目标位姿估计是在计算机视觉和机器人领域中的一项关键任务，旨在确定目

标物体在空间中的位置和方向。这项任务通常涉及从传感器数据中提取特征并使用各种算法来推断目标的姿态。其中，传感器可以是摄像头、激光雷达、惯性测量单元（IMU）等。在 2D 平面抓取和 6DoF 抓取场景中，大多需要估计 6D 物体姿态，这有助于机器人感知目标物体的三维位置和三维方向[10]。

大量智能机器人抓取的相关算法都需要机器人抓取仿真环境进行准确度和正确性测试，因此，本项目拟完成一个智能机器人抓取仿真系统，模拟真实物理环境的机器人抓取，并结合智能抓取方法进行仿真实验，为机器人抓取领域的相关算法提供一个仿真测试工具。

1.2. 国内外研究现状

1.2.1. 开发平台

机器人抓取技术在自动化生产和服务机器人领域扮演着至关重要的角色。为了有效地设计、测试和验证抓取算法和控制方案，研究人员和工程师们通常依赖于各种仿真开发平台[11]。当前已有大量智能机器人仿真开发平台，为机器人仿真实验提供了便利，其中最常用的有 Pybullet 模块、ROS 库的 Gazebo 环境、MuJoCo 引擎等。

Pybullet 是基于开源物理 bullet 开发的一个 Python 模块，专为机器人仿真和机器人控制任务而设计。它提供了强大的物理仿真功能，支持多种机器人模型和环境设置，包含了大量机器人抓取相关功能的接口，用户可以方便地使用这些接口快速地开发和测试抓取算法。

Robot Operating System(ROS)是帮助创建机器人应用程序的打包库和工具，有来自世界各地的许多贡献[12]。Gazebo 提供了一个高度可定制的 3D 仿真环境，支持多机器人协同操作和各种传感器模拟[13]。通过 Gazebo，用户可以模拟真实世界中的复杂场景，进行抓取算法的验证和性能评估。在 ROS 和 Gazebo 平台上模拟了一些著名的机器人，如 PR2、Care-O-bot、TurtleBot 等[14]。

MuJoCo 是一个专为模拟关节生物力学系统和机器人系统而设计的物理引擎[15]。它被广泛用于机器人学、生物力学、计算机图形学和机器学习等领域的研究和开发中。MuJoCo 具有以下几个关键特点：首先，它能够提供高度准确的复杂动力系统模拟，能够模拟机器人、仿人机器人、操作器和其他关节结构，准确

模拟关节约束、摩擦、接触和动力学。其次，MuJoCo 具有高效的计算性能，可以有效地模拟具有许多自由度的系统，适用于实时控制和优化任务。再者，MuJoCo 易于使用，采用简单的基于 XML 的文件格式描述要模拟的系统的物理属性，这使得指定复杂场景和定制模拟变得容易。

PyBullet、Gazebo 和 MuJoCo 在机器人抓取方面各有优劣。PyBullet 作为开源项目，具有灵活性和与机器学习框架的良好集成，但精度和性能可能不如商业软件 MuJoCo。Gazebo 作为 ROS 的一部分，具有与 ROS 集成紧密和多种传感器支持的优势，但在计算资源消耗和学习曲线方面存在挑战。而 MuJoCo 以其高精度的物理仿真和出色的计算效率脱颖而出，但需要购买许可证，且定制性较差。

根据项目需求，本项目选用 Pybullet 作为仿真系统的开发平台，相比其他开发平台而言，Pybullet 的优势在于可使用 Tensorflow 和 Pytorch 包训练数据，操作简单，易于上手，同时提供了可视化界面进行用户交互，适合本项目智能机器人抓取仿真系统的开发。

1.2.2. 仿真系统及算法仿真测试

真实机器人抓取实验的设备成本较高，且难以模拟或复现复杂的抓取场景，因此许多机械臂相关的研究都在仿真环境下进行实验以获取数据，以提前验证算法的正确性，为后续实景测试提供理论基础。

来自谷歌的研究人员在 Transporter Networks 这份工作提出了名为 Transporter Nets 的简单模型架构[16]，用于学习基于视觉的物体整理工作，其使用 Ravens 这个模拟抓取场景集合，在 Pybullet 环境下进行实验获取仿真数据，以验证所提出架构的正确性和有效性。

KitchenShift 这份工作的目标是证明当前基于模仿的策略学习方法不具有零概率泛化能力，缺乏对训练环境中微小变化的鲁棒性[17]。为了证明这一点，研究人员设计了 KitchenShift，一个在 MuJoCo 开发环境下的多机器人厨房操作模拟，使用仿真模拟多机器人在厨房进行作业，以验证其猜想的正确性，并鼓励解决这一问题。

OpenGRASP 是一个包含机器人抓取和操作的仿真工具包[18]，除了可扩展性，可操作性和公共可用性之外，OpenGrasp 还提供了一个整体的环境，用于处理仿真的各种方面，包括新算法的开发和测试，环境和机器人的建模等。

OpenGRASP 基于模块化架构，支持新功能的创建和添加，以及现有和广泛使用的技术和标准的集成。此外，还为生成和迁移这些模型创建了指定的编辑器。

A.T. Miller 等人提出了一种名为 GraspIt! 的机器人抓取模拟器，用于解决五指抓握模型拾取和放置类型的任务[19]。GraspIt! 包含一个可交互的用户界面，自带抓取分析和可视化方法，具有碰撞检测和接触确定系统，使用户能够评估抓取并计算最佳抓取位姿和力度。

1.3. 论文结构

接下来的三个章节所述内容如下：

在第二章，本文将介绍智能机器人抓取仿真系统的设计，包含功能设计图和工作流程图以及具体功能实现，包含用户交互模块的 GUI 控制界面和基于 Pybullet 平台基础上修改的仿真图形界面。

在第三章，本文参考了一个抓取位姿候选算法进行模拟实验，通过测试抓取成功率的实验以验证本项目设计的仿真系统具有足够的算法测试能力。

第四章是本项目的总结及展望，总结本项目的成果和不足之处，并对仿真系统的相关工作进行展望。

2. 研究内容

如前文所述，真实机器人抓取实验的设备成本较高，且难以模拟或复现复杂的抓取场景。现有的机器人抓取仿真系统，如 OpenGrasp (2003)，GraspIt! (2004) 这两个仿真系统，年代较为久远，难以与目前流行的物理仿真平台结合测试，在算法测试层面具有一定的局限性，而目前使用仿真测试的机器人相关算法大多是使用仿真平台对自设的特定场景进行测试，缺少用户的交互。因此，本文在 Pybullet 机器人仿真平台的基础上，设计了一个交互性强，具有泛用性，满足用户便捷测试需求的一个智能机器人抓取仿真系统。

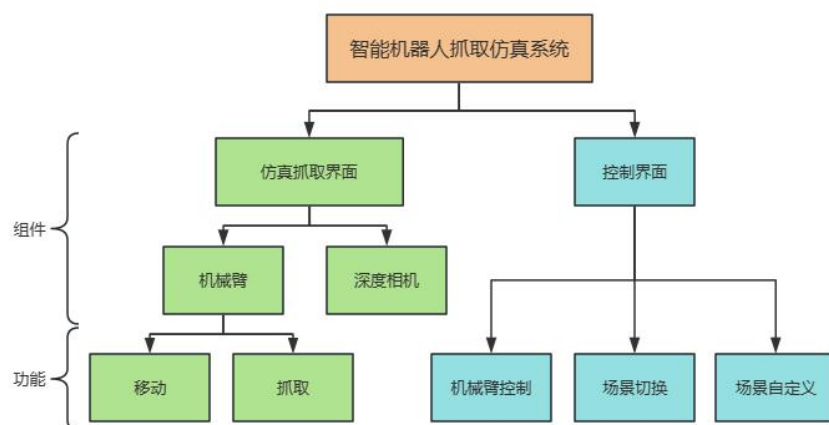
本章将具体介绍智能机器人抓取仿真系统的功能设计、操作流程和具体实现方法。

2.1. 系统设计及工作流程

基于机器人抓取的相关需求，绘制系统架构图和流程图如图 2.1、图 2.2。

如图 2.1 所示，智能机器人抓取仿真系统包含仿真抓取界面和控制界面两大主要组件，其中仿真抓取界面还包含机械臂和深度相机两个组件。功能方面，机械臂组件具有移动和抓取功能，控制界面具有机械臂控制、场景切换和场景自定义功能。

如图 2.2 所示，抓取流程首先是用户选择抓取场景和抓取对象，由虚拟相机拍摄场景和抓取对象的相关信息，再通过相关算法计算获取合适的抓取位姿，由机械臂根据数据执行抓取。



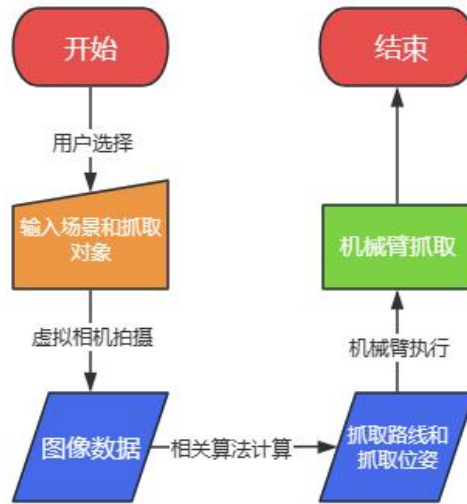


图 2.2 智能机器人抓取仿真系统流程图

2.2. 系统功能概况

本项目已完成如系统设计的智能机器人抓取仿真系统的所有功能，包括动作检测算法，机械臂的手动控制、自动抓取，虚拟深度相机的成像、点云建模，场景控制模块的场景切换和自定义场景模块的添加抓取目标、导入/导出场景描述文件功能。下文将详细介绍上述功能的实现方法。

2.2.1. 动作检测算法

在 Pybullet 仿真物理环境中，所有物体的状态都在实时更新，以达到模拟真实物理场景的效果。这样的更新机制使得机械臂会实时接受指令并改变状态，机械臂的移动需要时间，但指令会持续不断的发送，这就导致当机械臂在执行指令的过程中接受了另一个指令时，它会放弃继续执行当前指令而立即执行新的指令，这显然不合理。为解决这一问题，使机械臂能够正确且按顺序完成预设的移动和抓取动作，本系统设计了一个动作序列，用于分步执行每一步操作，实现了移动检测和抓取检测两个模块，用于检测移动和抓取操作是否完成，相关伪代码及分析如下：

算法 1: 移动检测算法

输入：目标位置 `target_position`，当前位置 `position`

输出：是否进行下一步操作

变量：位置误差值 `diff`，最大误差值 `DIFF`，失败次数 `error`，最大失败次数

MAX_WAITING_TIME

```
1  diff ← absolute_value(target_position − position)
2  if error ≤ MAX_WAITING_TIME and diff ≥ DIFF then
3      error ← error + 1
4      return False
5  else
6      error ← 0
7      return True
8  end if
```

移动检测算法输入当前位置和当前步骤需要移动到的位置，比较它们的差值‘diff’，如果差值大于等于设定的差异值‘DIFF’代表未移动完成，检测失败。为防止机械臂移动过程受阻导致算法进入死循环，此处还添加了最大失败次数，当检测失败次数超出阈值后，会立即中断此步操作进行下一步操作。

算法 2: 抓取检测算法

输入：目标位置 position， 左右夹爪位置 left_position, right_position

输出：是否进行下一步操作

变量：失败次数 error，最大失败次数 MAX_WAITING_TIME

```
1  left_min, left_max ← getAABB(left_position)
2  right_min, right_max ← getAABB(right_position)
3  min, max ← getAABB(target)
4  left_judge ← check_collision(left_min, left_max, min, max)
5  right_judge ← check_collision(right_min, right_max, min, max)
6  if (left_judge and right_judge) or error > MAX_WAITING_TIME then
7      error ← 0
8      return True
9  else
10     error ← error + 1
11     return False
12  end if
```

抓取检测算法输入目标位置和左右夹爪位置，通过‘getAABB’方法获取它们碰撞立方体的极大和极小坐标，检测抓取目标是否同时和两个夹爪碰撞，用以判断是否抓取到目标。

2.2.2. 机械臂控制

仿真机械臂具有移动和抓取两个基本操作，移动操作需要机械臂多轴协同配合完成使抓手移动到指定位置，抓取指控制抓手开闭。本项目实现了手动控制机械臂移动和转向、根据指定位置自动抓取两个机械臂基本操作，并将功能整合到一个 GUI 页面上，如图 2.3 所示。



图 2.3 机械臂控制界面

（1）手动控制移动及转向：Pybullet 库提供的‘calculateInverseKinematics’方法可以通过计算关节角得到每个关节的移动规划，‘setJointMotorControl2’方法可以根据上述方法计算得出的移动规划，控制机械臂每个关节的移动，使得抓手能调整到指定位姿。机械臂控制页面包含移动、转向、抓取、指定位姿移动四个子模块。在移动模块中，用户可点击按钮使机械臂抓手向指定方向移动固定步长；在转向模块中，用户可点击按钮操作机械臂按 roll、pitch、yaw 方向进行旋转，每次旋转固定角度；抓取模块中，用户可点击按钮控制夹爪开启/闭合。上述三

个控制模块主要用于用户与系统交互，指定位姿移动模块可用于算法测试。指定位姿移动模块需要输入[x, y, z, row, pitch, yaw]格式的坐标，在点击“确认”按钮后，机械臂会执行按指定朝向移动到目标位置的操作。

（2）自动抓取：此功能是二维平面自动抓取功能，实现原理是将指定物体的坐标设定为抓取位置，然后在动作队列中依次加入五个动作：1. 张开夹爪 2. 改变朝向为竖直向下 3. 移动到抓取位置 4. 闭合夹爪 5. 移动到初始位置。实现抓取指定物体的操作。

2.2.3. 抓取场景控制及自定义

为满足测试需求，本项目实现了场景切换及场景自定义模块，用户可在场景控制台界面切换抓取场景（如图 2.4），也可通过导入现有场景或自行设计场景物品进行场景自定义（如图 2.5）。



图 2.4 场景切换界面



图 2.5 场景自定义界面

(1) 场景设计：本项目目前设置了四个抓取场景，如图 2.6 所示，场景 1（左上）为 Pybullet 场景；场景 2（右上）为自设固定抓取场景；场景 3（左下）为小车抓取场景；场景 4（右下）为空白场景，主要用于自定义场景。

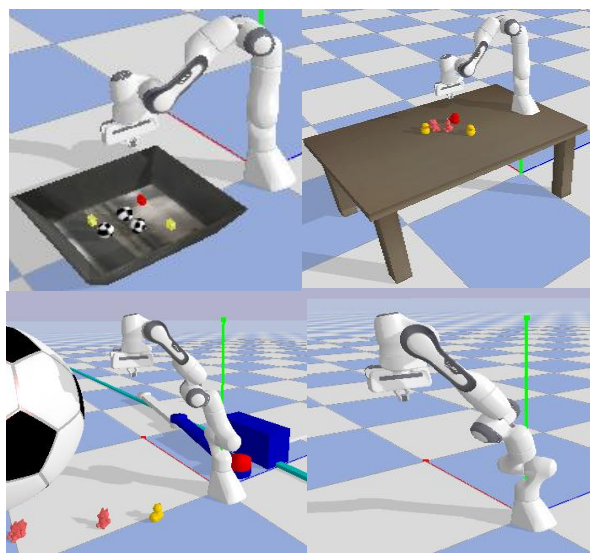


图 2.6: 抓取场景示例

(2) 自定义场景：为满足用户在不同场景下进行抓取测试的要求，本项目设计了一个自定义场景功能，主要用于添加自定义模型以自行设计新场景。添加

自定义模型功能需要用户提供模型的.urdf 模型描述文件，输入需要放置模型的位置坐标和方向坐标，根据.urdf 文件目录为当前场景添加模型。

(3) 场景描述文件：为方便用户导出通过自定义场景模块设计的新场景，本系统将抓取场景设计为.json 描述文件进行保存，用户可通过格式相同的.json 文件导入已有的抓取场景。本系统的.json 文件储存了机械臂和抓取物体的文件地址、位姿向量等内容，导入.json 文件时，系统会读取机械臂和各抓取物体的地址和位姿，进行逐个加载，以完成场景的导入。

2.2.4. 虚拟相机成像

为实现对目标检测及目标位姿估计算法的测试，本项目设计并模拟了一个装在机械臂上的仿真深度相机以获取抓取目标图像。

(1) 相机参数设定：Pybullet 库提供的‘computeViewMatrix’和‘computeProjectionMatrixFOV’方法可以通过给定参数计算相机的视图矩阵和投影矩阵。其中视图矩阵需要输入相机的位置、朝向及视图方向，投影矩阵需要输入相机的视野角度、远近拍摄距离及拍摄图像长宽比。投影矩阵视野角度设定为 45 度，远近距离经多次尝试设定为 0.1 至 5.0 单位长度，图像长宽比设为 1。

(2) 相机位置选择：参考 Robotiq Wrist 腕部相机的安装位置及 Denso 机械臂的相机位置，相机位置可设定在机械臂的腕部，目前考虑到测试便利，减少遮挡，相机暂设定在抓取位置，即抓手中心。

(3) 深度图成像和点云生成：仿真系统利用模拟深度相机的 RGBD 数据以及相机参数，使用 open3d 库的‘create_from_rgbd_image’方法生成单帧点云，便于目标位姿估计相关算法的测试，效果图如图 2.7。

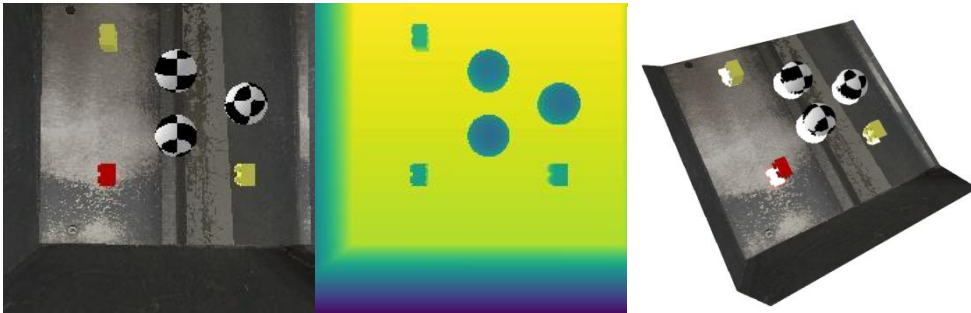


图 2.7：深度相机成像，从左至右分别为 RGB 图、深度图、生成点云

3. 抓取实验

本项目使用 Franka Panda 机械臂进行抓取实验。Franka Panda 机械臂是一个七轴平行夹爪机械臂，满足六自由度抓取的需求，且 Pybullet 库自带其.urdf 模型描述文件，适配于 Pybullet 的机械臂控制接口，故本项目选择 Franka Panda 机械臂用于仿真抓取测试。

REGNet 提出了一种以单视点云为输入的端到端抓取检测网络，生成包含抓取位姿候选模型的点云，在真实世界的抓取中实现了 79.34%的成功率和 96%的完成率[20]。因此，本项目使用 REGNet 的抓取检测网络进行测试，输入仿真环境的生成点云，测试抓取成功率，以检验 REGNet 的算法正确性，同时验证此仿真系统的算法测试能力。

3.1. 实验设计

本实验在自设抓取场景中测试，使用 3 个 Pybullet 库自带抓取模型素材，使用仿真系统生成单帧点云。将其输入 REGNet 网络，得到包含抓取位姿候选的点云，根据 REGNet 的输出——包含每个抓取位姿候选模型的位置、旋转矩阵和评分——选择评分最高的 4 个抓取模型(如图 3.1)，进行抓取测试。

抓取成功的评判标准是：机器人能够正确按指定位姿将物体抓取并保证物体不掉落。每个模型重复实验 15 次（在实验中，由于仿真平台会模拟所有实验物体的物理状态，导致它们的状态并不是完全固定的。而且，由于物理引擎设置，机械臂无法完全移动到指定的位置，数值上会存在微小的偏差，因此每次实验的状态均有所不同，故通过多次实验减少误差），记录抓取成功率数据为表 3.1。

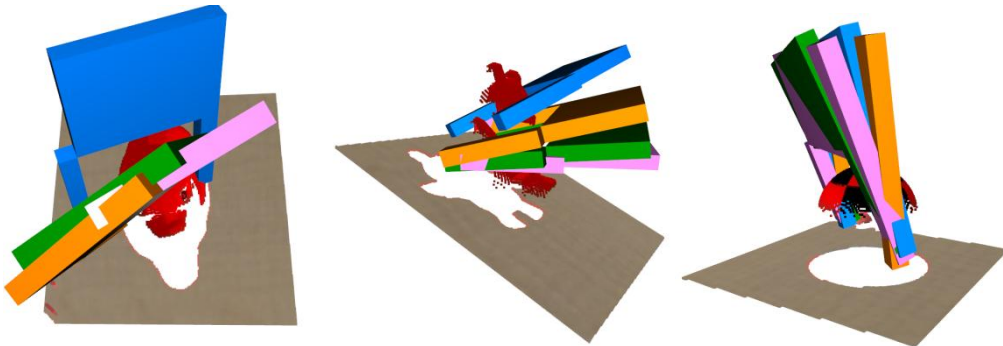


图 3.1 REGNet 的输出点云实例

3.2. 实验结果及分析

表 3.1 抓取实验数据

模型名称	位姿	Score	成功	失败	成功率
duck	1	0.598	14	1	93.3%
	2	0.589	14	1	93.3%
	3	0.578	12	3	80.0%
	4	0.577	13	2	86.7%
teddy	1	0.550	0	15	0.0%
	2	0.536	13	2	86.7%
	3	0.534	12	3	80.0%
	4	0.527	0	15	0.0%
sphere	1	0.553	13	2	86.7%
	2	0.548	12	3	80.0%
	3	0.545	13	2	86.7%
	4	0.541	12	3	80.0%

表格 1-4 列分别记录了抓取的.urdf 模型名称、实验的位姿编号（其中位姿 1、2、3、4 分别对应图 3.1 中橙色、蓝色、粉色、绿色抓取位姿候选模型）、REGNet 对该位姿的评分以及 15 次抓取实验成功率。

本次实验测试了 REGNet 抓取位姿候选算法的正确性，在 3 个模型，12 个抓取位姿，180 次抓取实验中，总成功次数为 128 次，总成功率为 71.1%。在 12 个抓取位姿中，在第二个泰迪熊模型的测试中有两个位姿是完全抓取失败的，其主要原因是机械臂在移动到指定位姿的过程中会触碰到抓取物体，导致其偏离初始位置，使得抓取失败，这是由于路径规划和位姿估计没有进行配合的原因，和抓取位姿估计的正确性无关。将两个冲突模型剔除后，在 150 次测试中成功次数为 128 次，成功率为 85.3%。

第一个鸭子模型的实验是成功率最高的，在 60 次实验中成功率为 88.3%，机器人在大部分情况下能够成功执行抓取操作。出现了少部分失败情况，部分原因是抓手移动过程中的触碰（如图 3.2），导致抓手没有稳定抓持住目标，在移动过程中物体掉落。

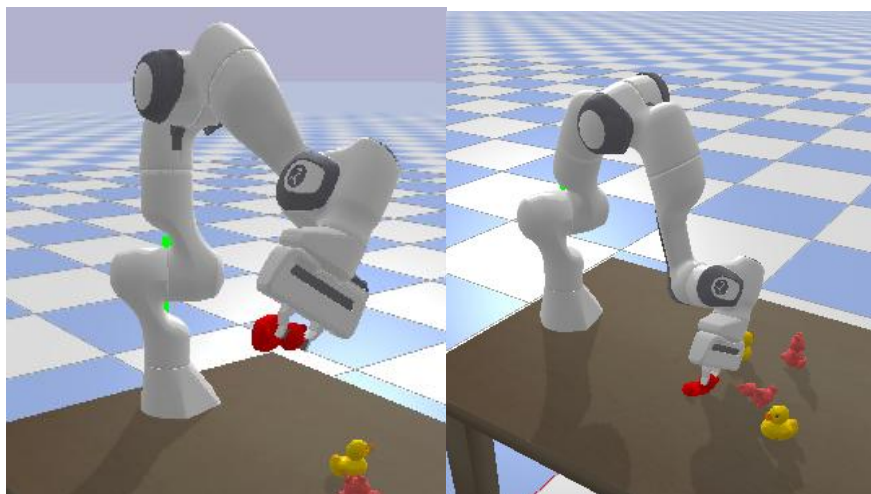


图 3.2 六自由度抓取测试，左为成功示例，右为失败示例

第二个泰迪熊模型效果较差，其主要原因如上文所述，是碰撞导致的物体偏移。对另外两个成功的抓取位姿，也存在侧面抓取不稳定导致物体滑落的问题，导致部分测试失败。

第三个模型使用小球模型，由于其表面较光滑，在实验前对其系数进行了一些调整，总抓取成功率为 83.3%。其中失败的实验均是由于碰撞扰动影响。

本实验获得了 85.3%（剔除错误组前 71.1%）的抓取成功率，和 REGNet 在真实世界抓取中获得的 79.34%成功抓取率相符合，因此可以认定本仿真系统对 REGNet 进行了一次成功的测试，验证了其算法的正确性。

总体来说，本次实验成功证明了 REGNet 生成抓取位姿的正确性，也证明了本仿真系统可以很好地适配已有的机器人抓取检测算法进行测试，实现六自由度抓取。但在本次实验中，仅测试了抓取位姿估计的 REGNet 算法，并未考虑机械臂的路径规划等问题，导致产生了一部分失败的抓取测试，这也很好地说明了机器人抓取是一项高精度要求的、需要多种算法结合的一项技术，仍需要多种算法的协同测试。

4. 总结与展望

本项目完成了一个智能机器人抓取仿真系统，基于 Python 编程，在 Pybullet 物理引擎的基础下，实现了机械臂的基本控制、深度相机的成像、抓取场景的切换及自定义等功能。在算法测试方面，本项目设计的仿真系统实现了模拟深度相机的 RGBD 成像及生成点云，提供测试算法的输入数据；提供了根据 6D 坐标控制机械臂移动的接口，以跟据测试算法的输出控制机械臂进行六自由度抓取。

在完成算法测试要求的同时，仿真系统提供了可操作的控制 GUI，用户可通过 GUI 便捷地对仿真机械臂进行操作。同时，系统提供的自定义场景模块为用户提供了自由设计抓取场景或导入已有场景的自由，便于多场景的抓取测试。

最后，本项目利用完成的仿真系统设计了一个简单的抓取实验，实验结果表明，本项目实现的仿真系统可以很好地适配已有的机器人抓取检测算法，实现六自由度抓取。

就未来应用而言，本项目实现的仿真系统还需要进行更多算法的协同测试，以适配更为复杂的抓取场景，为现实的抓取提供可靠数据。

参考文献

- [1] James, Stephen, et al. "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
- [2] Bousmalis, Konstantinos, et al. "Using simulation and domain adaptation to improve efficiency of deep robotic grasping." 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018.
- [3] Cheng, Hu, Yingying Wang, and Max Q-H. Meng. "A vision-based robot grasping system." IEEE Sensors Journal 22.10 (2022): 9610-9620.
- [4] Du, G., Wang, K., Lian, S. *et al.* Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artif Intell Rev* **54**, 1677–1734, 2021. <https://doi.org/10.1007/s10462-020-09888-5>
- [5] Mahler, Jeffrey, et al. "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics." arXiv preprint arXiv:1703.09312 (2017).
- [6] Song, Shuran, et al. "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations." IEEE Robotics and Automation Letters 5.3 (2020): 4978-4985.
- [7] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326 – 336, 2012. Autonomous Grasping.
- [8] Zou, Zhengxia, et al. "Object detection in 20 years: A survey." Proceedings of the IEEE 111.3 (2023): 257-276.
- [9] Hariharan, Bharath, et al. "Simultaneous detection and segmentation." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. Springer International Publishing, 2014.
- [10] Xiang, Yu, et al. "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes." arXiv preprint arXiv:1711.00199 (2017).
- [11] Žlajpah, Leon. "Simulation in robotics." *Mathematics and Computers in Simulation* 79.4 (2008): 879-897.
- [12] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

- [13] Koenig, Nathan, and Andrew Howard. "Design and use paradigms for gazebo, an open-source multi-robot simulator." 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566). Vol. 3. Ieee, 2004.
- [14] Qian, Wei, et al. "Manipulation task simulation using ROS and Gazebo." 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). IEEE, 2014.
- [15] Todorov, Emanuel, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control." 2012 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2012.
- [16] Zeng, Andy, et al. "Transporter networks: Rearranging the visual world for robotic manipulation." Conference on Robot Learning. PMLR, 2021.
- [17] Xing, Eliot, et al. "Kitchenshift: Evaluating zero-shot generalization of imitation-based policy learning under domain shifts." NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications. 2021.
- [18] León, Beatriz, et al. "Opengrasp: a toolkit for robot grasping simulation." Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference, SIMPAR 2010, Darmstadt, Germany, November 15-18, 2010. Proceedings 2. Springer Berlin Heidelberg, 2010.
- [19] Miller, Andrew T., and Peter K. Allen. "Grasplit! a versatile simulator for robotic grasping." IEEE Robotics & Automation Magazine 11.4 (2004): 110-122.
- [20] Zhao, Binglei, et al. "Regnet: Region-based grasp network for end-to-end grasp detection in point clouds." 2021 IEEE international conference on robotics and automation (ICRA). IEEE, 2021.

致谢

在完成本论文的过程中，我要衷心感谢我的导师杨剑老师和史玉回老师。感谢您在整个研究过程中给予我的指导和支持。您的专业知识和悉心指导使我能够克服各种困难，顺利完成了论文的写作。

同时，我也要感谢陈薪宇师兄对本文的技术支持和审阅。您的帮助对于我完成实验、撰写论文、修改论文并提高论文质量起到了关键作用。

此外，我还要感谢我的家人和朋友，是你们在我学习和研究过程中的支持和理解，让我能够全身心地投入到论文工作中。

最后，我要感谢所有在我学业生涯中给予过我帮助和支持的人，你们的支持是我前进的动力。