

# 《机器人学导论》课程设计

付焘 肖书奇

2021 年 6 月 6 日

# 目录

## ① 任务

- 原理与技术分析

## ② 原理简介

- 路径规划

## ③ 软件系统

- 系统设计
- 系统实现
- 图形用户界面的演示

## ④ 总结

# 任务:堆积木

- Intel Realsense D455 相机
- QKM SI7400 开放式六轴串联机器人

要求:

- ① 积木随机散落桌面上;
- ② 可以指定积木的搭建方法;
- ③ 越高越好,越快越好。



# 任务分析

## 理论

- 刚体机器人的正逆运动学；
- 路径规划；
- 轨迹规划；
- 相机的手眼标定；
- 物体识别；
- TCP,FTP,Socket 网络通信；

## 技术

- C++ 多文件编程与项目构建
- 线性代数开源库 Eigen 的使用
- 视觉开源库 OpenCV 的使用
- 计算机网络开源库 Poco 的使用
- 图形用户界面的开发

# 路径规划: 围绕最优搭建中心的分段直线

## 最优搭建中心的确定

- 与各个积木初始位置的距离之和最小
- 不可与积木初始位置重叠

$$p = [x, y, z, r, p, y]^T, \quad f(p) = \sum_{i=1}^N \|p - q_i\|$$

$$s.t. \quad \|p - q_i\| \geq k$$

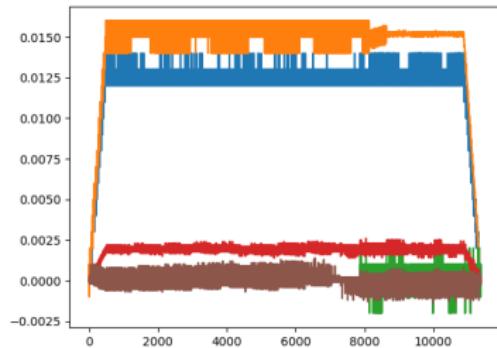
$$\min f(p)$$

含简单约束的凸优化问题, 用数值方法求解。

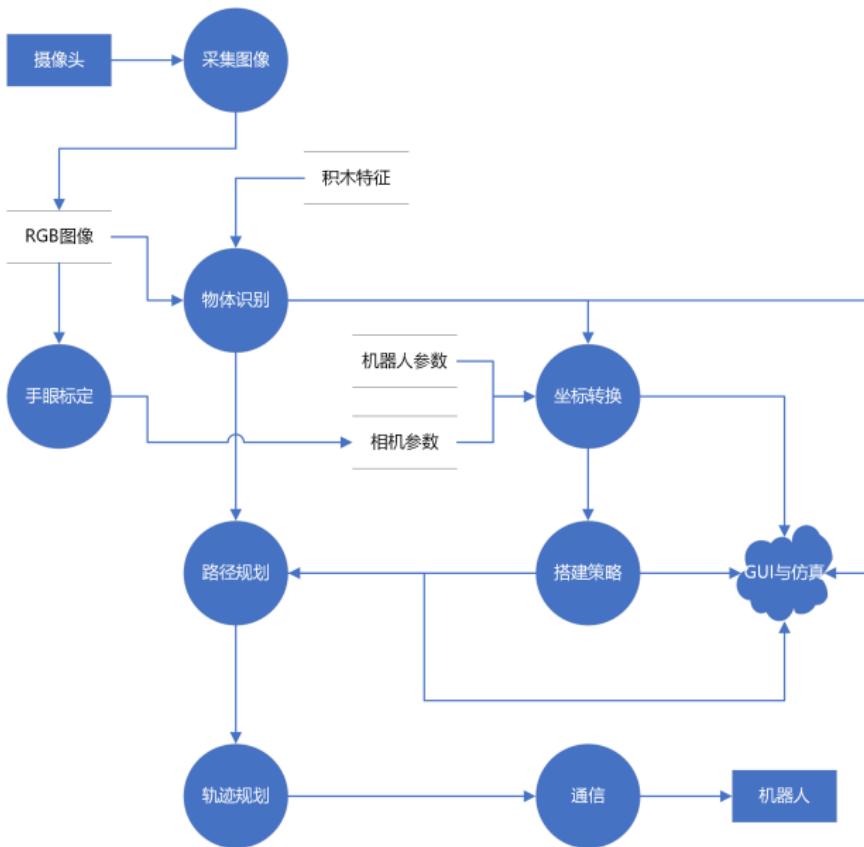
```
void GeometricMedian(const vector<Vector2d> &points, Vector2d  
&center, double &min_dist, const double &forbidden_zone_radius  
const double &lower_limit);
```

# 轨迹规划:操作空间,梯形速度

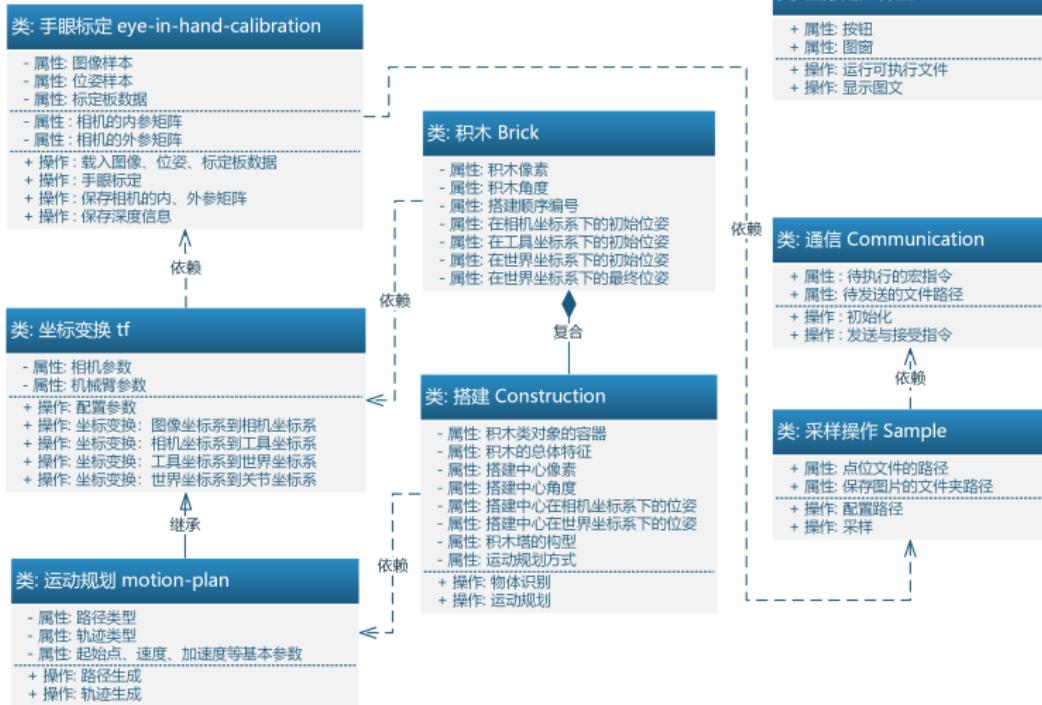
$$\vec{x}(n+1) - \vec{x}(n) = \begin{cases} \frac{1}{2} a_{acc} \vec{k} t_s^2 (2n + 1) & 0 < n < N_{acc} \\ \vec{v k t_s} & N_{acc} < n < N_{dec} \\ \vec{v k t_s} - \frac{1}{2} a_{dec} \vec{k} t_s^2 (2n + 1 - 2N_{dec}) & N_{dec} < n < N \end{cases}$$



# 设计:UML 数据流图



# 设计:UML 类图



# 实现

- C++17 标准, GCC 编译器构建;
- 图形用户界面使用 Python 编写。

```
// Sampling for eye-in-hand calibration
sample_calibration_sample("./share/eye-in-hand-calibration/src");
sample_calibration_sample("./share/eye-in-hand-calibration/src/rpy.txt");
calibration_sample<Sample>();

// Calibration
sysman.calibration.calibration("./share/eye-in-hand-calibration/src", 11, 8, 7, 7);
calibration.Calibrate("./share/eye-in-hand-calibration/src");
calibration.SystemicCalibration();
calibration.SystemicCalibration();

// Sampling for construction
sample_src_sample("./share/sample",
                 "./share/sample/rpy.txt");
src_sample<Sample>();

// Detection, Generating trajectories, log
construction.Detect();
construction.AverageTrajectory();
construction.Solution();
construction.log();

// Ping-Pong buffer application
Communication ppa_application;
ppa_application.SendInstruction("1# System.Home [1]");

ppa_application.SendInstruction("1# Eye-in-hand camera [1]");

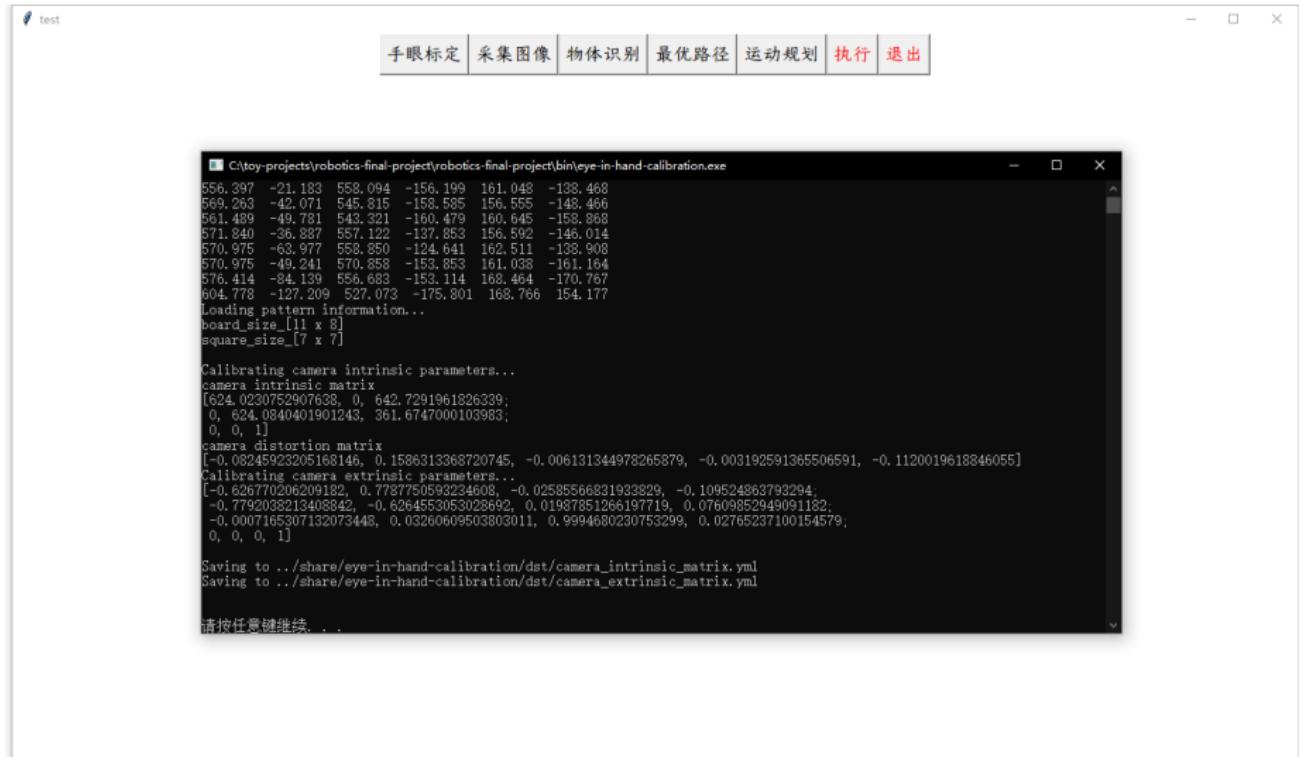
ppa_application.SendInstruction("1# Robot.Home [1]"); // joint space

int traj_num = 0; // the number of trajectories
for (const auto entry : std::filesystem::directory_iterator("./share/motion-plan/0st"))
{
    if (entry.path().extension() == ".txt")
    {
        std::cout << entry.path() << std::endl;
        rtpcontrol::RtpControl::Handle(<string>"192.168.19.101", "data", entry.path().string(), std::to_string(traj_num) + ".txt");
    }
    traj_num++;
}

for (int i = 0; i < traj_num; i++)
{
    ppa_application.SendInstruction("1# FileReadFile 1,/data/" + std::to_string(i) + ".txt");
    // i - move to the first point in the file, 0 - right-hand frame, 1 - rpy representation
    ppa_application.SendInstruction("2# FileInterpreter 1,0,1");
    ppa_application.SendInstruction("1# FileCloseFile 1");
    ppa_application.SendInstruction("1# MotionPlan 1");
    ppa_application.SendInstruction("1# MotionPlan 1");
}
```



# GUI: 手眼标定



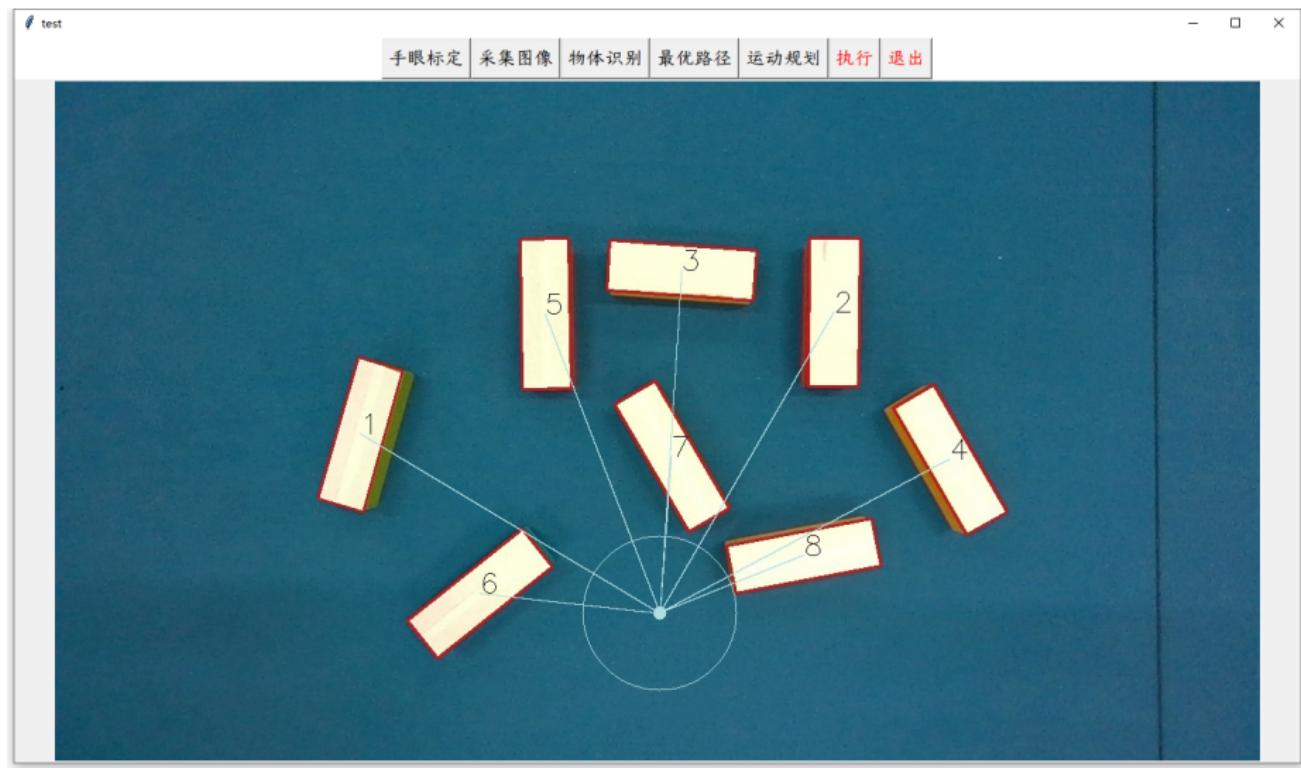
# GUI:采集图像



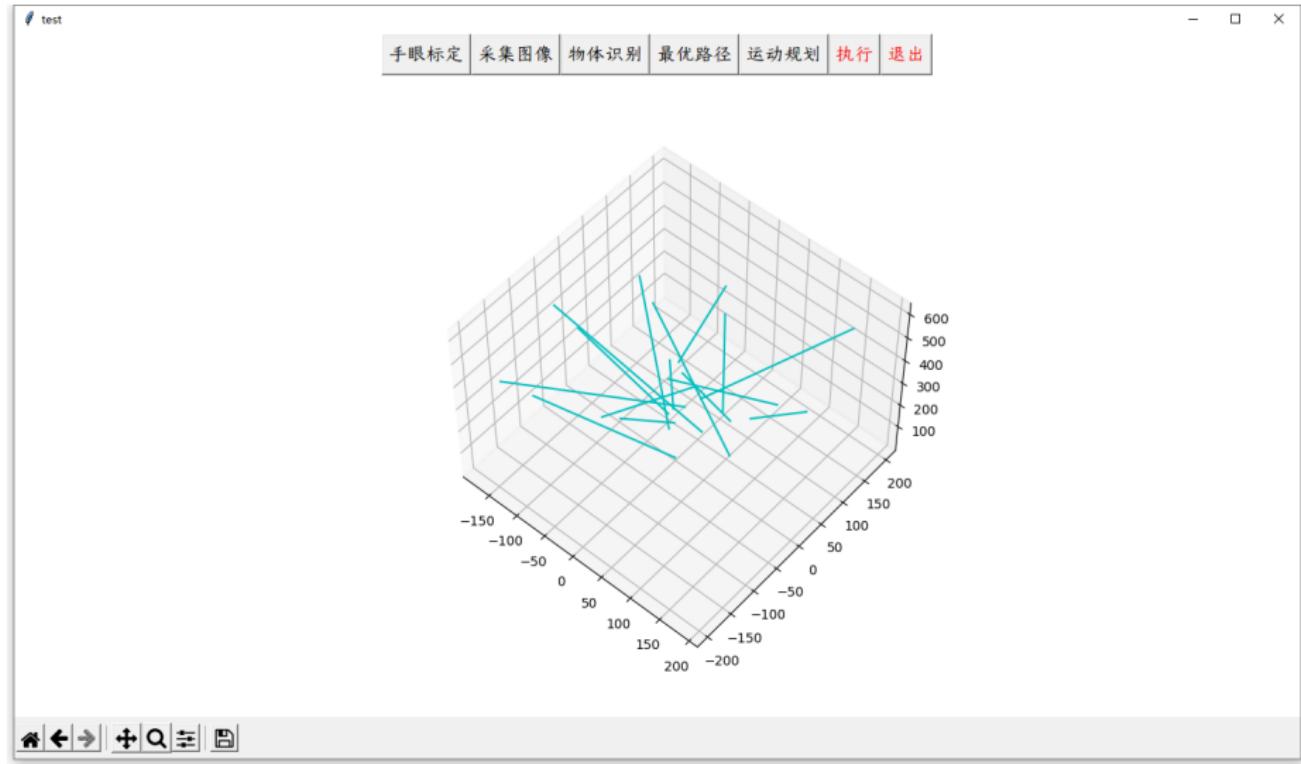
# GUI: 物体识别



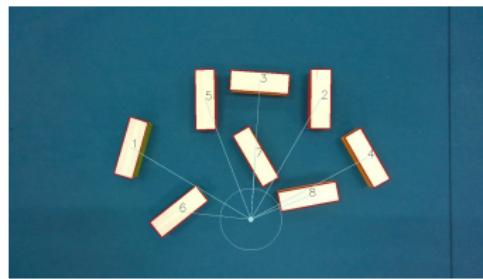
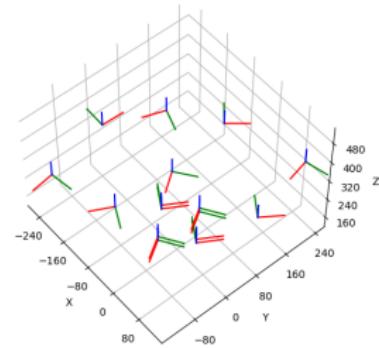
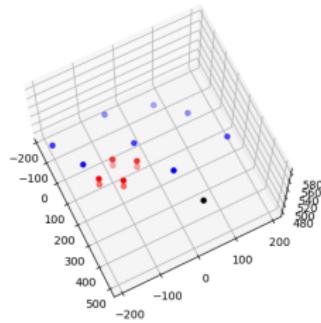
# GUI: 最优搭建中心



# GUI:路径规划



# GUI: 起点终点的刚体位姿仿真



# 总结

## 待改进

### 特色

- ① 面向对象的系统设计与实现；
- ② 路径规划：凸优化；
- ③ 刚体位姿的可视化仿真；
- ④ 图形用户界面；

- ① 优化轨迹规划，物体识别等方法；
- ② 利用 Realsense 相机的深度信息提升精度；
- ③ 项目的可维护性；
- ④ 完善仿真功能；
- ⑤ 多试验；

# The End