

3D 视觉创新应用竞赛

作品材料

作品名称： RGB-D 相机结合机械臂的无序抓取系统

团队成员	姓名	学校	专业	年级	项目经历	邮箱
	赵晨昊	深圳大学	光学工程	研二	基于结构光的 三维重建	17722547920@163.com
	张曼英	深圳大学	光学工程	研二	高精度激光线 扫描系统； 人脸三维重建；	17724601021@163.com
	周制黔	深圳大学	光学工程	研二	二维云台搭载 测振仪及相机的 目标跟踪	457690373@qq.com
	郭宏希	深圳大学	光学工程	研二	红外图像融合； 嵌入式开发；	13128832184@163.com
	陈炎彬	深圳大学	光学工程	研二	基于 Scheimpflug 成像的大景深 三维视觉系统及 目标识别研究	836924042@qq.com

二〇二〇年十二月

目录

第 1 章. 作品方案.....	3
1.1 系统构成.....	3
1.2 关键技术创新点.....	4
1.3 算法设计.....	4
1.4 测试结果.....	4
第 3 章. 已知 bug 及问题记录.....	6
第 4 章. 软件设计与流程.....	8
第 5 章. 源代码示例.....	9
第 6 章. 摄像头及开发板问题描述.....	9

第 1 章. 作品方案

1.1 系统构成

在工业生产中，自动化抓取具有十分广阔的应用场景，如汽车装配、工件制造等过程。基于 2D 机器视觉的平面目标的抓取技术，在工业生产应用中已非常成熟。但这类应用对加工零件位置要求较为严格，要求加工零件必须在同一平面分开摆放，不能有重叠，此类技术只需定位零件的二维位置，无需将零件的三维姿态纳入考虑范围。然而，在实际的工业生产中，生产环境和作业要求较为复杂，经常需要从杂乱无章的物料框中抓取零件，待抓取的目标零件存在重叠、遮掩的情况，在该条件下，基于 2D 机器视觉的平面抓取技术显然无法顺利完成抓取工作。

因此，我们利用奥比中光科技集团股份有限公司提供的 Zora P1 开发板、RGBD 相机，以及深圳大学物理与光电工程学院智能光测研究院提供的川崎机器人（6 轴）开发出了一个 RGB-D 相机结合机械臂对任意摆放的复杂加工零件的三维重建及无序抓取系统，系统搭建如下图 1 所示，软件系统如图 2 所示。硬件系统主要包括以下两部分：

- （1）控制系统：川崎 6 轴机器人，气泵与气动夹具
- （2）视觉系统：奥比中光 RGBD 相机，开发板，高性能计算机

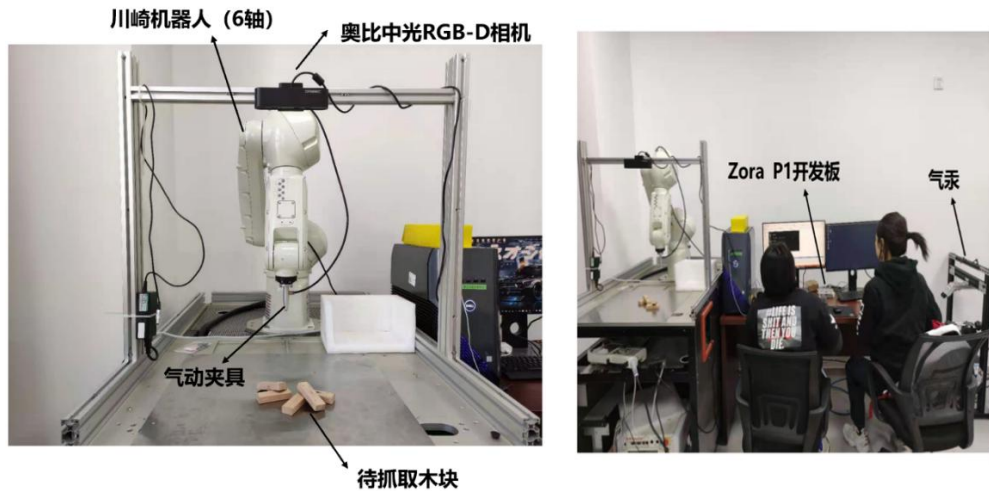


图 1. 系统构成

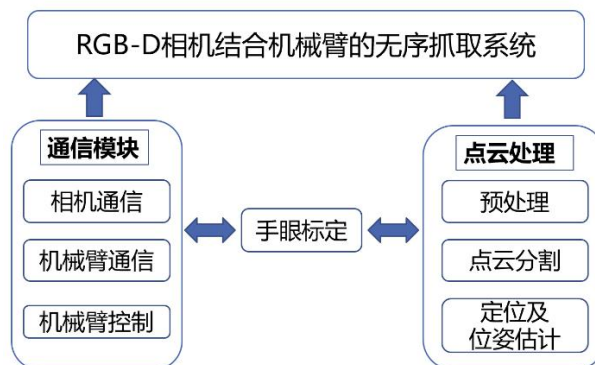


图 2. 软件系统

1.2 关键技术创新点

1. 采用一种快速 4 点手眼标定法，在没有强度信息下进行手眼标定。利用 4 点对 SVD 分解出 RT 矩阵
2. 可以带姿态进行无序抓取。即任意摆放都可以在平面的法线方向抓取。

1.3 算法设计

本项目的设计方案如图 3 所示，依据此方案进行算法的设计。

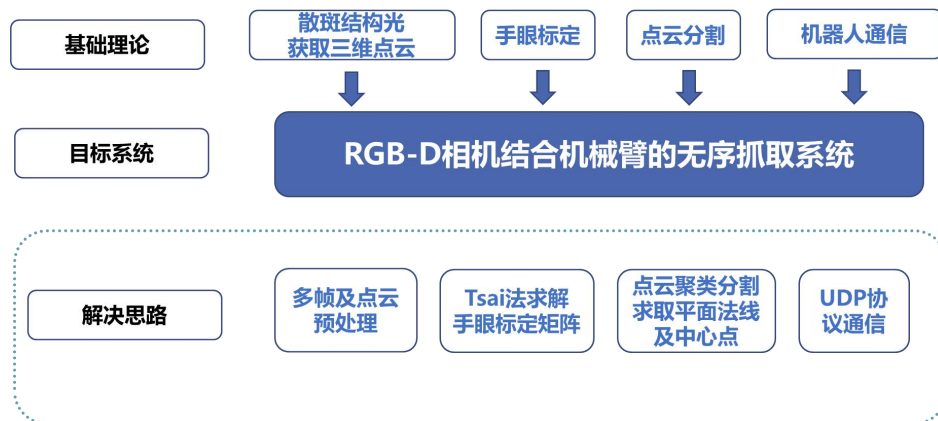


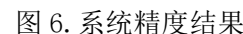
图 3.方案设计

1.4 测试结果

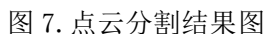
如图 5 所示，机械臂可以以小木块平面法线方向抓取，并且任意摆放的情况下可以顺利完成抓取任务。如图 6 所示：视觉系统的重建精度用一个标准平面的测量标准差来描述。Astra pro 在 620mm 工作距离内的平面标准差为 1.43mm。精度较差。是整个系统误差的最大来源。手眼标定精度用已知坐标转换点的平均误差来表示。收到视觉系统重建误差的影响，手眼标定的匹配误差均值为 1.5mm。抓取误差为每次抓取需要的补偿量的平均值表示，X，Y 方向为 0.5mm，Z 方向为 2mm。角度误差为 0.8 度。



图 5.带姿态进行无序抓取结果示意图



1. 相机输出深度图，转为点云图，并进行多帧，点云滤波等处理（直通滤波，体素滤波等）。
2. 6 轴机器人与计算机之间的通信（UDP 协议）
3. 相机与 6 轴机器人计算手眼标定矩阵，这里使用了 4 对点 SVD 分解法进行求解。得到视觉系统与机器人本体系统之间的转换矩阵。
4. 制作木块模板，并求出木块本体坐标系下抓取点和法线信息保存至文件中。
5. 利用相机拍得目标（木块）点云，得到点云进行点云滤波，下采样（实际中点云较少可以去掉）。
6. 点云分割（这里使用了聚类分割方法，由于木块是都是平面构成，聚类分割效果较好）。



7. 点云配准 (4PCS+ICP) 由于点云精度较差导致匹配结果较差, 且不鲁棒, 一致性差, 所以放弃此方案。

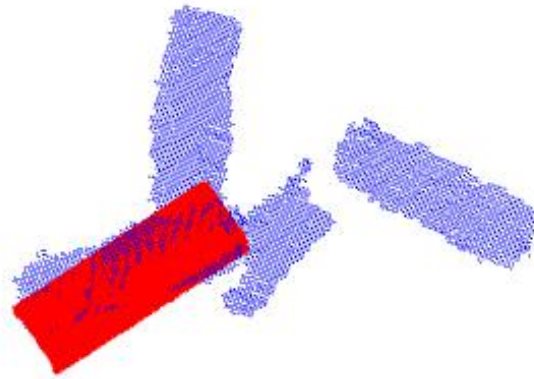


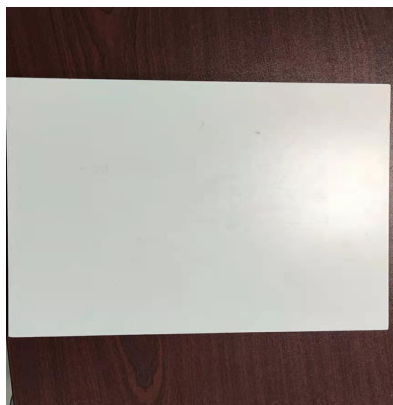
图 8. 点云配准结果图

8. 转为使用平面拟合及中心点的方案进行抓取。将点云分割出的最多点构成的面进行平面拟合，计算平面中心点和法向量
9. 利用法向量与机器人坐标系之间的关系，求解机器人的抓取姿态（OAT）
10. 输入坐标与姿态信息指令（X, Y, Z, O, A, T）给机器人实现抓取。
11. 开发板与相机，机械臂进行通信（QT 实现）
12. 开发板环境搭建（QT5, OPENCV320, PCL1.8）遇到 flann 库编译冲突并解决。
13. 开发板代码编译运行（由于开发板内存较小，无法正常运行）

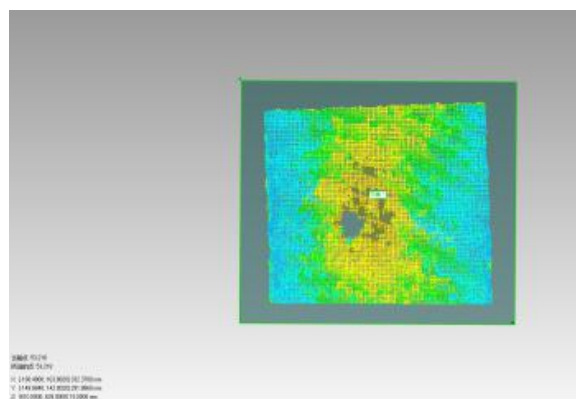
第 3 章. 已知 bug 及问题记录

1. 点云精度问题：

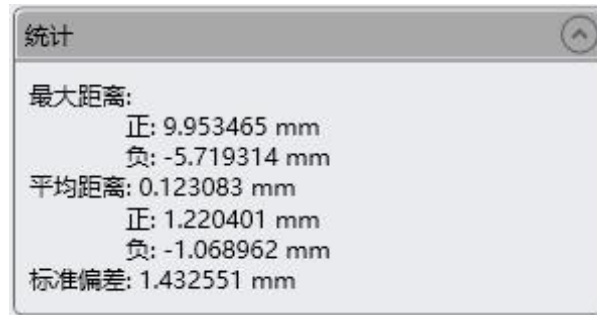
图 9 为工作距离（620mm）下拍白色背景板点云在 Geomagic 中拟合的平面，标准差 1.43mm，正负最大距离为 9mm，正负平均距离 1.22mm。数据可知在此工作距离下的平面标准差为 1.43mm，比较大，且点云会出现分层现象，导致了手眼标定时会产生误差，点云分割会导致同一个面分割成两个部分，以及点云配准时的不准确问题等等。也是本项目中最大误差产生的来源。我们通过点云滤波，取平均中心点和法线方向来平均掉误差，并且在控制系统中，让机械臂去在 x, y, z 方向补偿误差来进行抓取。



(a)



(b)



(c)

图 9. (a) (b) (c) 平面拟合精度分析图

2. 编译开发板过程中由于用到了 pcl 库和 opencv 库，遇到 flann 库错误，环境搭建等问题, 已解决。
3. 开发板由于内存较小，编译过程中卡死问题。采用英伟达开发套件进行交叉编译（同为 arm 架构，内核相同）编译失败，未找到原因。

第 4 章. 软件设计与流程

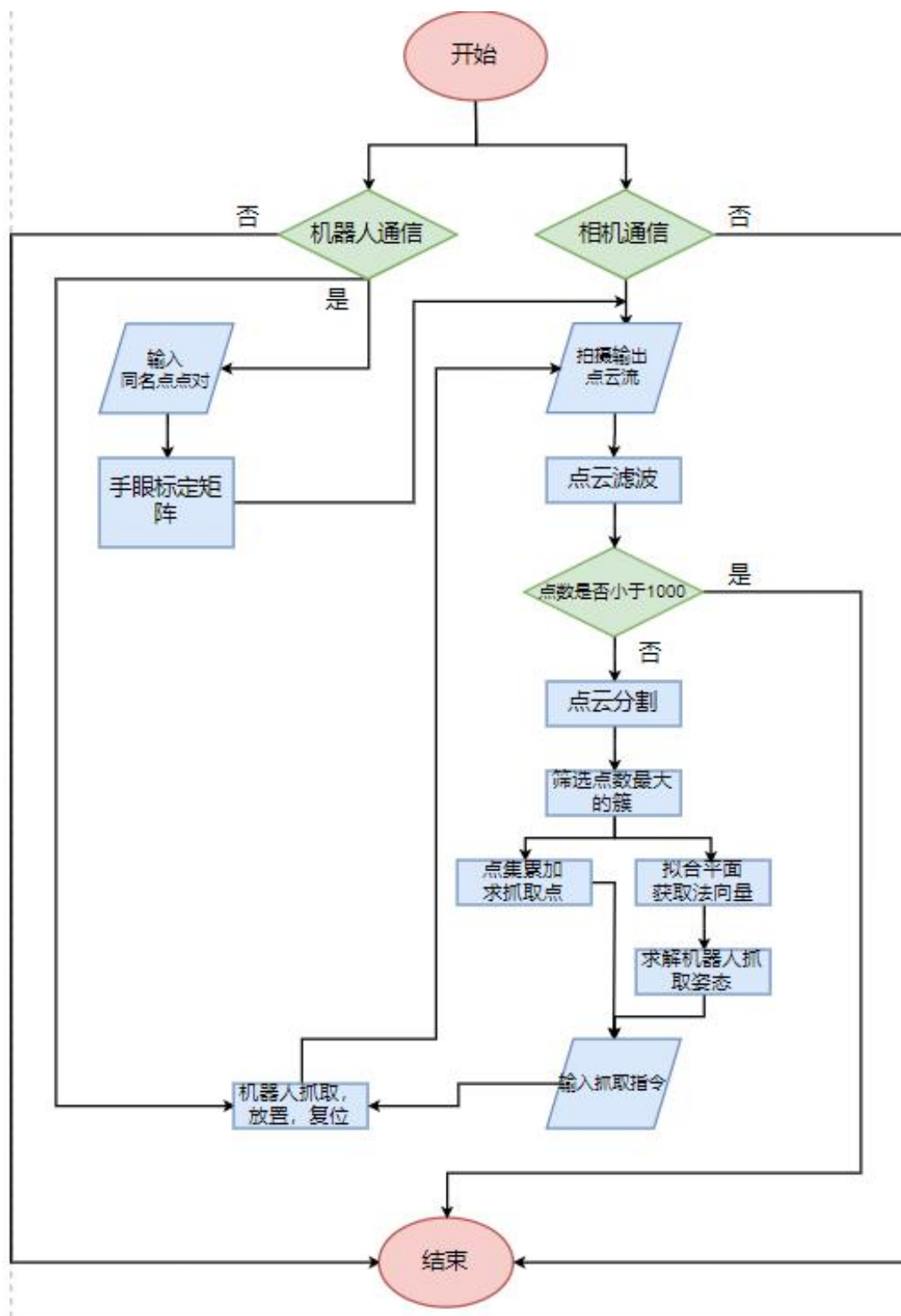


图 10. 软件设计流程图

第 5 章. 源代码示例

项目开源 github 网址: https://github.com/ManyingZhang/orbbec_robotCatch

```

148 void eyetohand()
149 {
150     int num = 4;
151     FILE *f1 = fopen("Robotxyz.txt", "r");
152     vector<Vec3d> M;
153     for (int i = 0; i < num; i++)
154     {
155         Vec3d Temp1;
156         float x1, y1, z1;
157         fscanf_s(f1, "%f %f %f\n", &x1, &y1, &z1);
158         Temp1[0] = x1; Temp1[1] = y1; Temp1[2] = z1;
159         M.push_back(Temp1);
160     }
161     fclose(f1);
162
163     FILE *f2 = fopen("PCLxyz.txt", "r");
164     vector<Vec3d> D;
165     for (int i = 0; i < num; i++)
166     {
167         Vec3d Temp2;
168         float x2, y2, z2;
169         fscanf_s(f2, "%f %f %f\n", &x2, &y2, &z2);
170         Temp2[0] = x2; Temp2[1] = y2; Temp2[2] = z2;
171         D.push_back(Temp2);
172     }
173     fclose(f2);
174
175     MatId R;
176     MatId T;
177     ComputeRT_3D(M, D, R, T);
178
179     MatId RT = MatId::eye(4, 4);
180     R.copyTo(RT(Rect(0, 0, 3, 3)));
181     RT(0, 3) = T[0][0];
182     RT(1, 3) = T[1][0];
183     RT(2, 3) = T[2][0];
184     cout << RT << endl;
185     system("pause");
186     RT = RT.inv();
187     WriteData("ethRTinv.txt", RT);
188 }

```

图 11. 部分开源代码

第 6 章. 摄像头及开发板问题描述

1. 相机问题:

(1) 本项目中: 工作距离 (620mm) 精度较差, 标准差 1.4mm, 对于目标抓取有很大的影响。应使用条纹结构光相机进行测量。

(2) 无法单独找到深度相机接口, 控制红外散斑开关进行拍照。

2. 开发板问题: 内存较小, 在编译过程中容易死机