
目录

论文思路整理	2
1. 概述	2
2. 手眼标定	2
3. 点云处理	2
3.1 技术选择	2
待解决问题	3
手眼标定方案讨论:	4
1. 标定球方案-空间定点 ^[23-10-26]	4
1.1 尝试自动标定方法	4
2. 非空间定点	5
手眼标定实验	6
1. 基于标定板正交属性的手眼标定 – 无误差分析	6
1.1 结果:	6
1.2 关于 PSO – 慎用!	7
1.3 结果分析	7

论文思路整理

1. 概述

首先，论文最终想要呈现的效果是一个系统，由【视觉引导的机器人系统】，名字待定。其包括手眼标定 -> 自动标定 -> 点云处理[依赖高精度手眼标定进行的配准，依赖无监督小样本的点云分割]。

为什么要进行手眼标定 -

为什么要使用深度学习的方法来进行点云处理：

传统方法不具有足够的鲁棒性，针对环境的改变没有很强的适应性。例：目前使用的点云提取方法中，噪点是否提取成功的重要因素。

2. 手眼标定

3. 点云处理

首先，想要拿一个其他数据集所训练的网络直接取分割工件是不可能。得到的网络只是数据集中输入和其标签之间的映射关系，虽然大多数网络和算法都可以做到提取局部特征和全局特征，但在训练学习的过程中，是通过预测值和真实标签之间的关系建立的损失函数，来构建网络的。

3.1 技术选择

无监督学习 + 有监督 & 小样本 + 迁移学习 + PointNN 提速训练 + 传统方法提取工件目标区域作为标签

基于 graphTER，虽然这个网络自称是无监督网络，但是在进行解码时还需要利用标签来训练分类器(解码器)实现标签和提取到特征的对应关系。不足：在工业应用中，为工件点云打标签是不显示的。改进：利用由传统方法提取出的 ROI 作为标签

- 使用多模态，加入由点云不同的随机视角生成的 2D 图像的特征提取，然后在跨模态间进行对比学习。
- 借助 segmenta-anything
- 点云特征提取器{Point-NN, GraphTER, PointGLR}
- 图像特征提取器，应该就可以使用 segmenta-anything

[【点云系列】PointGLR: Unsupervised Structural Representation Learning of 3D Point Clouds-CSDN 博客](#)

[AAAI 2022 | 一种 3D 场景多模态对比学习新方法 SimIPU - 知乎 \(zhihu.com\)](#)

[CVPR 2022 | CrossPoint: 3D 点云理解的自监督跨模态对比学习 - 知乎 \(zhihu.com\)](#)

3.2 待解决问题

1. 如何处理噪点问题
2. 网络部署在哪里，部署在何时 -> 测试阶段，在系统开发阶段就部署网络，利用有监督学习生成的网络和测试阶段的小样本数据进行微调，样本标签可使用传统方法提取出的目标点云。
3. 针对有监督学习网络的训练，应该多多贴近实际需求，目前大多数点云分割任务所做的是将一个点云的不同部件全部分割出来。而在实际项目中，我们只需要将目标区域的点云分割出来即可！

以 shapeNet 为例，其将飞机主体、机翼、尾翼等全部分割，其与我们要处理的任务并不完全相同，可能我们只需要飞机的尾翼，而其他部位是否分割成功，我们并不关心。那么在进行网络设计时，我们可能需要圈定 ROI，然后寻找点云与 ROI 之间的关系。

但是这样看起来好像并没有把任务简单化。原任务是把点云进行拆分，而现在的任务变成提取指定 ROI。

手眼标定方案讨论：

1. 标定球方案-空间定点^[23-10-26]

硬件：标准球

标定原理：扫描标准球，通过圆拟合求出截面的圆心，根据勾股定理求出标准球球心，进行 ${}^B_C T_O P = {}^B_E T_O P$ 的坐标转换，根据误差方程 $\Delta_i = A b_i - c_i$ 求出手眼矩阵A。

改进方法：根据线激光相机光学成像原理，激光线在物体表面的高度差越大误差越大，所以在标定过程中改变位姿只关注拟合圆半径最大的地方，越接近球体半径效果越好。**存疑：后续应进行相应的实验进行误差分析。**

最优化方程-1: $\Delta = \sum_i \Delta_i$, ($\Delta_i = A_O^C P_i - {}^B_E T_O^E P_i$)

主要误差：

- 硬件误差： E_P ，标定物的硬件误差。
- Δ 的最优化误差

最优化方程-2：利用标定物和法兰盘的相对不变。 $({}^E_B T_C^B T_O^C P)_1 = ({}^E_B T_C^B T_O^C P)_2 = \dots$

主要误差：

- 相较于上式，减少了硬件误差，两者最优化方程的误差比较需进行实验尝试。

1.1 尝试自动标定方法

- 已知 E_P – 最优化方程 1:

随便将标定球移至激光线下(尽量靠近球心位置)，然后开始自动标定。

标定程序：第一步，求解粗糙精度手眼矩阵，首先拟合当前激光线下的圆，求出圆心坐标，然后求出球心坐标。调整机器人位姿，使机器人重新到达距离球心坐标 5%的误差范围内，根据求得位姿结果求解手眼矩阵。利用手眼矩阵求得当前拟合圆心在基坐标系下的坐标位置，然后将球心移至此坐标位置。然后控制机器人以不同位姿到达此空间定点，然后求解手眼矩阵。

- 已知 E_P – 自行建立外部基坐标系 – 最优化方程 1:

利用 tcp 在激光线处建立基坐标系(x, y 轴方向尽量与相机坐标系保持一致)，然后安置标定球，将标定球球心移至外部基坐标系原点。拟合圆，根据半径大小，

判断需要沿 y 轴哪个方向运动，然后找到最大半径，确定此定点位置坐标，然后改变机器人各轴位姿重新达到此位置坐标(可再次进行最大半径确认)。

- 未知 5_0P – 自行建立外部基坐标系– 最优化方程 2：效果也许比使用最优化方程 1 好
-

2. 非空间定点

非空间定点的方法，例如标定点，要求出相机的外参及标定物相对于相机的位姿，采用空间定点的方法理论上比非空间定点的方法要多一些求解步骤。

手眼标定实验

1. 基于标定板正交属性的手眼标定 – 无误差分析

1.1 结果:

对于正交结果没有取绝对值，并且-1 操作在了总和上，而不是正交结果上

[0.35916387647123915, -0.16341298928510636, 0.14733343822295228,

0.09098256646261173, 2052.058693479467, -581.7415735667876,

578.7794487773392] 31.60478334478031

[0.522084587820512, -0.8234775924389033, 1.0157965448543802,

-0.5170586066122033, 1302.4908465351934, -2161.112279317044,

1.5715959684077796] 243.1123204376505

[-0.2854214179659117, 0.13447477645743564, 1.2918097686678305,

0.44229037379147157, 2749.4538316265603, 778.6709631111136,

2080.9564076019246] -0.01057838886273288

修正后的结果(只使用了前两组数据)

[-0.1956222937825032, -0.766691234111081, -1.3069606486078362,

0.4611925079381167, -18.499284071773456, 137.47627475210075,

1130.7134499266062] 7.80849483352845

第一组数据

[-0.0991413135466297, 0.2990517623102416, 0.4023198355907962,

0.20760125872645924, -34.83984877057516, 166.67872691599973,

1160.3935743510986] 3.8762244552348335e-05

修正：对机器人位姿求逆

[-0.3303837255305453, 1.776886984456582, 1.7709967204580646, 0.8,

-53.70164663739292, 142.53531107860917, 1145.8717964358702]

0.00010950245283147808

1.2 关于 PSO – 慎用！

基于空间定点的手眼标定方案还没有想好初值的问题，但是 PSO 的求解范围对求解的精度提升有非常大的帮助。

1.3 结果分析

最后呈现的结果很差，通过求解出的手眼标定矩阵，将标定板上的点转换至机器人末端后，进行直线拟合，发现任意两条直线都没在一个平面。

原因尚未找到。