
目录

关于写论文相关.....	2
开题和书写基本逻辑.....	2
博客_网络资源.....	4
手眼标定.....	4
相机标定.....	4
点云处理.....	4
深度学习_论文.....	4
Pointnet & Pointnet++	4
深度学习_技术_概念.....	4
maxPooling	4
0. 通用问题.....	5
0.1 坐标系转换计算.....	5
0.2 齐次矩阵-位姿	5
0.3 手眼标定矩阵等式求解.....	5
0.4 手眼标定矩阵等式.....	5
0.5 手眼标定中的思想-坐标系转换关系中的思想.....	7
0.6 齐次变换矩阵及其使用（原理不太懂，）	7
0.7 三维空间中的向量 – 旋转矩阵.....	9
1. ★ 误差分析(巧妙思想)-机器人激光检测手眼标定误差分析及优化.....	12
1.1 硬件设备信息.....	12
1.2 误差来源分析.....	13
1.3 手眼关系及误差表达式.....	13
1.4 *球心拟合误差（需要看一下论文 3 中线激光结构光相机的成像原理）	14
1.5 *手眼矩阵计算误差及改进.....	14
1.6 其他.....	14
2. ★ 0.0815(高精度)-基于标准球的机器人手眼标定方法改进研究.....	15
2.1 设备信息.....	15
2.2 手眼关系及标定流程.....	15

2.3 其他.....	16
3.学位论文-基于 3D 机器视觉的工业机器人跟踪涂胶系统_陈琳.....	17
4. 线激光器的手眼标定方法.....	18
4.1 设备信息.....	18
4.2 其他.....	18
5. Pointnet	19
5.1 解决的问题.....	19

关于读&写论文相关结构

1. 摘要

(所研究领域的)大背景 & 问题大类

-> (本文主要解决了哪方面问题) 当前的主要挑战

-> 主流做法 & 存在的问题 & 瓶颈

-> 本文提取的方法

-> 方法创新点及结果

- 2. $({}^E_B T_{(i+1)})^{-1} \times {}^E_B T_i \times {}^B_C T_i = {}^B_C T_{(i+1)} \times {}^C_O T_{(i+1)} \times ({}^C_O T_i)^{-1} X = {}^B_C T_i = {}^B_C T_{(i+1)}$

$$H_{gij} H_{gc} = H_{gc} H_{cij}$$

$$\begin{cases} R_{gij} R_{cg} = R_{cg} R_{cij} \\ (R_{gij} - I) T_{cg} = R_{cg} T_{cij} - T_{gij} \end{cases}$$

$$AX=XB$$

$$R_A R_X = R_X R_B$$

$$(R_A - I) T_X = R_X T_B - T_A$$

$$({}_0^ET)_i - ({}_0^ET)_{i+1}$$

2.

开题和书写基本逻辑

1. 看论文，最起码要了解的多一些，最终呈现出来的要覆盖广！

其次，还有论文分布在哪些期刊，哪些时间点，这些可以去支撑可行性分析，同时可以窥见当前的研究现状、近年来的研究进展和未来几年大概的发展方向和趋势。权威性！！

2. 论文肯定是解决问题的，从现实的小问题开始，解决之后解决大问题，问题越来越复杂，内容越来越丰富。

读论文

[谈一谈 CS.CV 方向如何阅读论文？ - 知乎 \(zhihu.com\)](#)

以后记录的最基本架构可参考如下：

1. 解决了什么问题 & 背景
2. 优点(为什么可以做到) & 主要贡献
3. 不足(有待改进的地方)

先 BFS 速读一遍论文，Abstract 和 Intro 要细，Related work 可以快速过甚至不看，Approach 稍快，关注一些特殊模块，experiment 图一乐。总体来讲第一遍要关注图和表，这样可以对文章形成初步的认识，并且判断是否值得你全文细读，有时可能只需要关键部分细读，就不必看全文。

博客_网络资源

手眼标定

相机标定

[相机标定目的及原理_相机标定的目的-CSDN 博客](#)

像素坐标系->物理坐标系->相机坐标系->世界坐标系

前两步的矩阵相乘的结果合称为相机内参

最后一步成为相加外参，也就是我们需要的标定物相对于相机的位姿。

这个所谓的标定物相对于相机的位姿不是必要条件，因为我们有标定物在相机坐标系中的坐标，以眼在手外为例，完全可以使用

$${}^E_oP = {}^E_BT \cdot {}^B_cT \cdot {}^c_oP$$

点云处理

[【3D 点云】分割算法总结（一）_3d 点云分割-CSDN 博客](#)

深度学习_论文

Pointnet & Pointnet++

[3D 分类与分割之 PointNet 论文笔记 - 知乎 \(zhihu.com\)](#)

深度学习_技术_概念

maxPooling

[对 Max Pooling 的理解_maxpooling-CSDN 博客](#)

0.通用问题

0.1 坐标系转换计算

(一) 关于手眼标定理论相关的笔记 [boss-dog 的博客-CSDN 博客](#)

结论:

$${}^A_B M = {}^B_A M^{-1}$$

$$\star \quad {}^1_n M = {}^1_2 M {}^2_3 M \dots {}^{n-1}_n M$$

0.2 齐次矩阵-位姿

${}^c_o T$ 是标定物坐标系到相机坐标系的齐次转换矩阵，实际含义是标定物在相机座标下的位姿，即以相机坐标系为基坐标系，标定物的位姿（标定物的坐标系）

$${}^b_o T = {}^b_e T \times {}^e_c T \times {}^c_o T$$

物体在基坐标系下的位姿 = 手在基坐标系下的位姿 * 相机在末端坐标系下的位姿 * 物体在相机中的位姿
这也就是“左乘基”的含义，每个位姿左乘它的基坐标系

0.3 手眼标定矩阵等式求解

求解手眼标定矩阵等式本质上就是最优化问题，因为列出等式的因式全部都是测量值，其在理论上是相等的，但是实际中是不可能完全相等的，所以涉及到了最优化问题，选择最优解。目前论文普遍使用最小二乘法进行简单筛选，目前无法确定使用最优化算法进行求解是否会带来精度的提升。

0.4 手眼标定矩阵等式

- 最常见的莫过于 $AX=XB$ ，就是把 $({}^e_b T \times {}^b_c T \times {}^c_o T)_i = ({}^e_o T)_i = ({}^e_o T)_{i+1} = ({}^e_b T \times {}^b_c T \times {}^c_o T)_{i+1}$ 中的手眼矩阵设为 X ，等式两边同右乘 ${}^c_o T^{-1}$ ，同左乘 ${}^e_b T^{-1}$ 即可。

- 使用空间顶点方法（标准球）的论文中更多使用

$P_B = {}^b_e T \times {}^e_c T \times P_C$ 然后在不同位姿下测量 P_B^i 例：

$$\begin{bmatrix} P_b \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_l \\ 1 \end{bmatrix} = B_t X \begin{bmatrix} P_l \\ 1 \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} x_b \\ y_b \\ z_b \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix}$$

$$B_{n1} X \begin{bmatrix} P_{n1} \\ 1 \end{bmatrix} = B_{n2} X \begin{bmatrix} P_{n2} \\ 1 \end{bmatrix} = \cdots = B_{nn} X \begin{bmatrix} P_{nn} \\ 1 \end{bmatrix}$$

然后使用四元数表示 X(手眼矩阵)联立方程直接求解。

- 也可以将 $P_B = {}^B T \times {}^E T \times P_C$ 转换为 $Ax=b$ 的问题使用非线性的方法进行求解。(最优化问题)

$$A = \begin{bmatrix} {}^B E r_{11} \cdot x_l & {}^B E r_{12} \cdot x_l & {}^B E r_{13} \cdot x_l & {}^B E r_{11} \cdot z_l & {}^B E r_{12} \cdot z_l & {}^B E r_{13} \cdot z_l & {}^B E r_{11} & {}^B E r_{12} & {}^B E r_{13} & {}^B E t_x \\ {}^B E r_{21} \cdot x_l & {}^B E r_{22} \cdot x_l & {}^B E r_{23} \cdot x_l & {}^B E r_{21} \cdot z_l & {}^B E r_{22} \cdot z_l & {}^B E r_{23} \cdot z_l & {}^B E r_{21} & {}^B E r_{22} & {}^B E r_{23} & {}^B E t_y \\ {}^B E r_{31} \cdot x_l & {}^B E r_{32} \cdot x_l & {}^B E r_{33} \cdot x_l & {}^B E r_{31} \cdot z_l & {}^B E r_{32} \cdot z_l & {}^B E r_{33} \cdot z_l & {}^B E r_{31} & {}^B E r_{32} & {}^B E r_{33} & {}^B E t_z \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$x = [{}^E L r_{11} \quad {}^E L r_{21} \quad {}^E L r_{31} \quad {}^E L r_{13} \quad {}^E L r_{23} \quad {}^E L r_{33} \quad {}^E L t_x \quad {}^E L t_y \quad {}^E L t_z \quad 1]^T, b = [x_b \quad y_b \quad z_b \quad 1]^T$$

$$\begin{cases} {}^E L r_{11}^2 + {}^E L r_{21}^2 + {}^E L r_{31}^2 = 1 \\ {}^E L r_{13}^2 + {}^E L r_{23}^2 + {}^E L r_{33}^2 = 1 \end{cases}$$

旋转矩阵为正交矩阵，以其向量为单位向量为约束条件。

- 除这些常规方法以外，还可以视情况而定去表示手眼关系等式。

$${}^B T_O^C X = {}^B T_O^E X \quad (1)$$

即：

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^E \\ y^E \\ z^E \\ 1 \end{bmatrix} \quad (2)$$

此例为手在眼外，且文中称可以根据标定物测量得出 ${}^E X$ 。

0.5 手眼标定中的思想-坐标系转换关系中的思想

- 首先是坐标系转换关系中的思想，为什么我们要做手眼标定，因为我们的手和眼不在一个维度（确切的说是不在同一个坐标系），那么我们就没有办法去让手干眼所看到的事情。所以我们要做手眼标定，即坐标系转换，而这个系统内部组成并不是静止不动的，也就是说针对每一个内部组成，其自身的坐标系原点有可能是在变化的，我们无法对其进行坐标变换。所以在进行坐标变换时，我们要找到两个系统内部相对静止的坐标系，一个是我们的目标坐标系转换矩阵即手眼矩阵，一个是用于罗列坐标转换矩阵等式用于求解。【将变量化解，使用定值求解】
 - Eye-to-hand:以眼在手外为例，相机和机器人基底的坐标系以及标定物和法兰盘坐标系明显相对静止，针对不同位姿有以下方程式： $({}^E T \times {}^B T \times {}^C T)_i = ({}^E T)_i = ({}^E T)_{i+1}$ $({}^B T \times {}^C T \times {}^O T)_{i+1}$ 。
B->E 可有机器人示教器读取，O->C 可由相机读取，由此可求手眼矩阵即 C->B 的坐标系转换矩阵。 $\underset{X}{\operatorname{argmin}} h(X) = \sum_i f(X)$
 - Eye-in-hand:以眼在手上为例，相机和法兰盘的坐标系以及标定物和机器人基底坐标系明显相对静止，针对不同位姿有以下方程式： $({}^B T \times {}^E T \times {}^C T)_i = ({}^B T)_i = ({}^B T)_{i+1}$ $({}^E T \times {}^C T \times {}^O T)_{i+1}$ 。
B->E 可有机器人示教器读取，O->C 可由相机读取，由此可求手眼矩阵即 C->E 的坐标系转换矩阵。

0.6 齐次变换矩阵及其使用（原理不太懂，—）

${}^B_E T$ 表示{E}坐标系到{B}坐标系的齐次变换矩阵。及 E 在基坐标系 B 下的位姿^[23-10-31]。

相机坐标中的一个点转换至基坐标系则有：

$$P_B = {}^B_E T \times {}^E_C T \times P_C \text{ (左乘才表示从下到上，从旧到新，从 L->E)}$$

首先要明确的是齐次变换矩阵是干嘛的，他是将空间中某一坐标系中的向量经过平移旋转至另一坐标系，那么在此基础上就不可能出现 $P(4*1) \times T(4*4)$ 的情况。

那么就有

$$P_L = P_B \times {}^B_E T \times {}^E_C T \text{ (右乘表示从上到下，从新到旧，从 E->L)}$$

即

$$P_E = \begin{matrix} {}^B \\ E \end{matrix} T \times P_E \times \begin{matrix} {}^B \\ E \end{matrix} T$$

不存在一个点右乘齐次变换矩阵，~~4x1~~去哪乘~~4x4~~去

0.7 三维空间中的向量 – 旋转矩阵

首先，旋转矩阵是正交矩阵。其次旋转矩阵是坐标系转换中新坐标系的基向量。更准确的，旋转矩阵是用原坐标系表示的新坐标系的基向量。证明见论文 7^[23-11-3]。

在三维空间中，两个向量的叉乘(外积)是一个垂直于这两个向量组成平面的向量(法向量)，向量大小为两个向量围城平行四边形的面积。

那么在旋转矩阵中，任意两个向量的叉乘等于另一向量。

向量点乘 $\mathbf{a} \cdot \mathbf{b}$ 的几何意义是 \mathbf{a} 在 \mathbf{b} 上的投影 $|\mathbf{a}| \cos \theta$ ，数值意义表示 \mathbf{a} 和 \mathbf{b} 的相似度(方向方面，数值越大方向越相似。)

[奇技淫巧系列：向量叉乘 - 知乎 \(zhihu.com\)](#)

叉乘计算：

[tempDocuments\向量叉乘.txt](#)

0.8 旋转向量、欧拉角、罗德里格斯参数(老迷糊了)

旋转矩阵是一个 3×3 的矩阵，它可以描述一个向量在三维空间中的旋转状态。具体来说，旋转矩阵的每个元素都代表了向量在旋转后在新坐标系中的投影。旋转矩阵的构造方式有多种，其中最常见的是使用三个互相垂直的单位向量来构造旋转矩阵。这三个单位向量通常被称为旋转矩阵的列向量，它们描述了向量在新坐标系中的方向。通过旋转矩阵，您可以方便地描述向量在三维空间中的旋转状态。

以下是一个绕 x、y、z 轴依次旋转 45 度的旋转矩阵：

```
R = [0.7071 0.7071 0;  
      -0.5    0.5    0.7071;  
      0.5   -0.5    0.7071]
```

将这个旋转矩阵转换成罗德里格斯参数的形式，我们可以使用以下的公式：

$\theta = \arccos((\text{trace}(\mathbf{R}) - 1) / 2)$

$\mathbf{k} = [\mathbf{R}(2,3) - \mathbf{R}(3,2); \mathbf{R}(3,1) - \mathbf{R}(1,3); \mathbf{R}(1,2) - \mathbf{R}(2,1)] / (2 * \sin(\theta))$

其中，`theta` 是旋转角度，`k` 是旋转轴向量，trace 是矩阵的秩。将上面的旋转矩阵代入公式，可以得到：

$\theta = 1.9106$

<p>$k = [0.4082; 0.4082; 0.8165]$</p> <p>因此，这个旋转矩阵可以表示为罗德里格斯参数 $[0.4082; 0.4082; 0.8165]$，其中旋转轴为 $[1; 1; 2]$，旋转角度为 109.47 度。</p> <p>将旋转矩阵转换成欧拉角的形式，我们可以使用以下的公式：</p> <p>$\phi = \text{atan2}(R(3,2), R(3,3))$</p> <p>$\theta = \text{atan2}(-R(3,1), \sqrt{R(3,2)^2 + R(3,3)^2})$</p> <p>$\psi = \text{atan2}(R(2,1), R(1,1))$</p> <p>其中，$\phi$、$\theta$ 和 ψ 分别是绕 z、y、x 轴旋转的欧拉角。将上面的旋转矩阵代入公式，可以得到：</p> <p>$\phi = 0.7854$</p> <p>$\theta = -0.7854$</p> <p>$\psi = 0.7854$</p> <p>因此，这个旋转矩阵可以表示为欧拉角 $[0.7854; -0.7854; 0.7854]$，其中旋转顺序为 Z-Y-X。</p>
<p>$R(\theta)v = v \cos \theta + (k \times v) \sin \theta + k(k \cdot v)(1 - \cos \theta)$ 罗德里格斯是描述一个向量 (v) 绕一个轴(k) 旋转一定角度(θ)</p> <p>这个公式是罗德里格斯公式的一种形式，它描述了向量绕着一个轴旋转一定的角度后的旋转结果。具体来说，这个公式可以将旋转向量 v 绕着旋转轴 k 旋转一个角度 θ，得到旋转后的向量 $R(\theta)v$。其中，$\cos \theta$ 和 $\sin \theta$ 分别代表旋转角度的余弦值和正弦值，$k \times v$ 表示向量 k 和 v 的叉积，$k \cdot v$ 表示向量 k 和 v 的点积，$1 - \cos \theta$ 表示旋转角度的余弦值。这个公式是罗德里格斯公式的一种形式，用于计算向量的旋转后的结果。</p>

0.9 TCP 标定和外部工具坐标系建立

0.10 相机成像

[UC3D 线激光轮廓传感器工作原理及参数说明 - 知乎 \(zhihu.com\)](https://zhuanlan.zhihu.com/p/44011011)

[工业相机（3D）主要参数详述_3d 相机速度计算公式图解-CSDN 博客](#)

[3D 视觉 之 线激光 3D 相机 - 知乎 \(zhihu.com\)](https://www.zhihu.com/question/35912111/answer/101100000)

0.11 结果误差分析

主流的结果误差分析采用的是以第一组数据为真值, 计算后续数据较之它的变换作为误差。

但是，开始就觉得这玩意有问题，如果第一组数值变差很大，后续数据与它的偏差很小，最后的误差看似很小，但是没有实际意义啊。所以对于最后的误差分析需要一个真正的定量真值，而不是假设的虚拟真值！！

下图中，在不同方法的情况下，最后得到的误差都在 0.24 左右，但是两个手眼矩阵之间的误差却很大，在 2mm 左右，他们与真值可能还存在一些误差。

TSAI误差分析							
	x(误差)	y(误差)	z(误差)	rx(误差)	ry(误差)	rz(误差)	
No.1	-0.46(0.0)	-0.54(0.0)	0.88(0.0)	-0.78(0.0)	1.5(0.0)	-1.06(0.0)	
No.2	-0.26(0.2)	-0.76(-0.22)	0.38(-0.5)	2.86(3.64)	-0.83(-2.32)	-2.95(-1.89)	
No.3	-0.08(0.03)	-0.66(-0.12)	0.79(-0.09)	-0.86(-0.08)	-0.79(-2.28)	-3.03(-1.97)	
No.4	-0.08(0.39)	-0.38(0.16)	0.9(0.02)	1.69(2.47)	0.76(-0.74)	-0.3(0.77)	
No.5	-0.56(-0.09)	-0.16(0.38)	0.91(0.03)	-2.98(-2.2)	-1.07(-2.56)	1.43(2.49)	
No.6	-0.23(0.23)	-0.88(-0.34)	0.56(-0.32)	-1.27(-0.49)	0.96(-0.53)	1.73(2.79)	
No.7	-0.09(0.38)	-0.78(-0.24)	0.72(-0.16)	-1.68(-0.9)	-0.46(-1.96)	-1.06(0.01)	
No.8	-0.39(0.08)	-0.79(-0.25)	0.66(-0.23)	0.31(1.09)	-0.64(-2.14)	-1.32(-0.26)	
No.9	-0.3(0.16)	-0.74(-0.2)	0.38(-0.51)	0.31(1.09)	-0.64(-2.14)	-1.32(-0.26)	
No.10	0.1(0.57)	-0.67(-0.13)	0.5(-0.39)	-0.66(-0.27)	-0.87(-2.37)	2.03(3.09)	
No.11	0.24(0.71)	-0.7(-0.16)	0.53(-0.35)	0.55(1.33)	-1.24(-2.73)	-1.14(-0.08)	
No.12	-0.27(0.19)	-0.73(-0.19)	0.33(-0.56)	2.16(2.94)	0.76(-0.73)	-2.96(-1.9)	
No.13	-0.18(0.28)	-0.76(-0.22)	0.57(-0.32)	-2.16(-1.38)	-1.46(-2.96)	-2.05(-0.98)	
平均值	0.28199778570207157	0.20096888177634056	0.2664016288584333	1.363901226067727	1.8049835836911163	1.2678238460492914	
最大值	0.7054136191667166	0.37788992185423737	0.5568070659684774	3.639878377470987	2.9601137046452486	3.0925747234538474	
最小值	0.0	0.0	0.0	0.0	0.0	0.0	
方差	0.03689458115499113	0.00867425355323771	0.03503005084307183	1.2125640182614736	0.8451696236421472	1.1834527696237325	

x、y、z误差的平均值: 0.2497894321122818

	x(误差)	y(误差)	z(误差)	rx(误差)	ry(误差)	rz(误差)
No.1	1.89(0.0)	-2.03(0.0)	0.75(0.0)	1.08(0.0)	0.58(0.0)	2.15(0.0)
No.2	1.87(-0.02)	-1.53(0.51)	0.46(-0.29)	2.32(1.23)	0.07(-0.51)	-1.33(-3.48)
No.3	2.08(0.19)	-1.67(0.36)	0.85(0.1)	-1.47(-2.55)	0.07(-0.5)	-1.43(-3.55)
No.4	1.93(0.04)	-1.78(0.26)	0.08(0.33)	2.58(1.5)	0.1(-0.47)	1.51(-0.64)
No.5	1.58(-0.31)	-2.16(-0.13)	0.95(0.2)	1.07(-0.01)	-0.93(-1.51)	-1.37(-3.52)
No.6	2.07(0.18)	-1.61(0.42)	0.49(-0.26)	-2.0(-3.08)	0.12(-0.45)	2.69(0.55)
No.7	2.13(0.24)	-1.62(0.41)	0.72(-0.02)	-0.75(-1.83)	0.01(-0.57)	0.1(-2.05)
No.8	1.99(0.1)	-1.8(0.24)	0.51(-0.24)	1.09(0.01)	0.07(-0.51)	-0.18(-2.33)
No.9	1.84(-0.05)	-1.56(0.48)	0.44(-0.3)	1.09(0.01)	0.06(-0.51)	-0.18(-2.33)
No.10	2.0(0.11)	-1.34(0.69)	0.78(0.03)	-2.11(-3.19)	-0.65(-1.23)	-1.76(-3.91)
No.11	2.09(0.2)	-1.26(0.77)	0.86(0.11)	1.55(0.47)	-0.39(-0.97)	-0.58(-2.72)
No.12	1.82(-0.07)	-1.5(0.53)	0.44(-0.31)	0.43(-0.65)	1.01(0.43)	-3.05(-5.2)
No.13	2.0(0.11)	-1.59(0.44)	0.6(-0.15)	-1.92(-3.0)	-0.45(-1.03)	-0.88(-3.03)
平均值	0.1249720235672804	0.40136145080322233	0.17976827127540987	1.34791457582551791	0.6698146957772375	2.562037596807782
最大值	0.30969637198669164	0.7693975359493144	0.3285312945664175	3.188466428273181	1.5094690030956537	5.2028317069009855
最小值	0.0	0.0	0.0	0.0	0.0	0.0
方差	0.008057602846853067	0.04171995813108394	0.012676715110352567	1.4798842500815708	0.14961025380551732	2.034388007325782

X、Y、Z误差的平均值: 0.23536724854863753

1. ★ 误差分析(巧妙思想)-机器人激光检测手眼标定误差分析及优化

Path: [Papers\over\手眼标定_23_10_9_★机器人激光检测手眼标定误差分析及优化_刘建春.pdf](#)

收获小结: eye-to-hand (平均精度 0.1495) || 首先提出手眼标定三大误差来源, 针对其中两种主要来源进行分析并优化。

求解手眼矩阵方法: 加权最小二乘法 (此外文中还提到了 SVD)

1.1 硬件设备信息

IRB ABB 6700 工业机器人

机器人重复定位精度为 0.05mm, 线性度为 0.096mm。机器人重复定位精度为 0.05mm 表示, 当机器人在同一位置执行多次任务时, 它们所到达的位置的最大偏差不超过 0.05mm。线性度为 0.096mm 表示, 机器人在执行线性运动时, 它们所到达的目标点与理论目标点之间的最大偏差不超过 0.096mm。

LJ-G5001 激光轮廓扫描仪 (官网没有这个型号)

两个标准球分别为半径 15mm, 圆度 0.0015mm 和半径 19.05mm, 圆度 0.002mm。

LJ-G5001基恩士日本原装正品 激光位移传感器 现货出售高精度2D激光位移传感器LJ-G5000 系列LJ-G5001 控制器型号LJ-G5001传感器兼容性测量模块无须工厂再校对即可互换可连接的传感器数量最大 2 个模块显示最小显示单位 $0.1\text{ }\mu\text{m}^*1$, 0.001 mm^2 , 0.01° 最大显示范围 $\pm 99999.9\text{ mm}$, $\pm 999999\text{ mm}^2$, $\pm 99999.9^\circ$ 输入端子座激光远程互锁输入无电压输入 *2输入端子台触发器输入对于感测头 A, 为无电压输入计时 1 输入无电压输入输入端子座自动归零 1 输入复位输入输出端子座模拟电压输出 $\pm 10\text{ V} \times 2$ 输出, 输出阻抗: $100\text{ }\Omega$ 总体判断输出NPN 集电极开路输出误差输出NPN 集电极开路输出 (N.C.)过程输出NPN 集电极开路输出触发器输入启用输出对于感测头 A, 为 NPN 集电极开路输出调整后误差输出扩展连接器计时 2 输入无电压输入自动归零 2 输入触发器输入对于感测头 B, 为无电压输入程序切换输入无电压输入, 4 个输入存储卡保存输入无电压输入激光关闭输入对于感测头 A/B, 为无电压输入判断/二进制输出3 等级判断输出: OUT1 至 OUT8、总体判断输出二进制输出: OUT1 至 OUT8 测量数据输出 (21 位)NPN 集电极开路输出*3选通脉冲输出NPN 集电极开路输出触发器输入启用输出对于感测头 B, NPN 集电极开路输出调整后误差输出模拟 RGB 显示器输出SVGA (800×600像素)RS-232C 接口测量数据输出与控制输入/输出 (最大波特率: 115200 位/秒, 可选)USB 接口支持USB2.0 HI-SPEED (兼容USB1.1 Full-SPEED)Ethernet 接口100BASE-TX/10BASE-T*4存储卡支持NR-M32 (32MB)、GR-M256 (256MB)、NR-M1G (1GB) (FAT32)主要功能感测头计算、轮廓调整、滤波器、平滑处理、平均、位置调整、OUT 名称更改、测量模式选择(高度、位置、缝隙、宽度、中心位置、截面积、交叉、角度、轮廓比较、轮廓跟踪)、缩放、平均、测量、测量值报警、容差设置、自动归零、存储 (数据/轮廓)、存储卡保存、程序存储、触发器模式更改、相互抗干扰、测量范围更改、校正、激光光线调整、采样时间设置、屏蔽、轮廓报警设置、倾斜调整、高度调整、显示语言切换、设置支持软件连接、触发间隔/测量时间显示等额定电源电压 $24\text{ VDC} \pm 10\%$ 、纹波 (P-P) 10% 以下最大消耗电流1 个感测头时为 800 mA 或更小/2 个感测头时为 1 A 或更小环境抗耐性环境温度 0 至 $+50^\circ\text{C}$ 相对湿度 35 至 $85\% \text{ RH}$ (无凝结)重量约 $1,050\text{ g}^*1$ 仅在连接 LJ-G015 或 LJ-G015K 时。连接其它传感器头时, 最小显示单位为 $1\text{ }\mu\text{m}^2$ (NPN 型) 无电压输入额定值: 电压 ON (开) 时 1 V 或更少, 电流 OFF (关) 时 0.6 mA 或更少。(PNP 型) 电压输入额定值: 电压 ON (开) 时 10.2 V 或更大 (最大 26.4 V), 电流 OFF (关) 时 0.6 mA 或更少 3 NPN 集电极开路输出的额定值: 最大 50 mA (最大 30 V), 残余电压最大 1 V 。PNP 集电极开路输出: 最大 50 mA (最大 30 V), 残余电压最大 1 V 。无电压输入的额定值: “打开” 电压 1 V 或更小, “关闭” 电流 0.6 mA 或更小 (触发器输入端子: “打开” 电压 1 V 或更小, “关闭” 电流 1.0 mA 或更小) 电压输入的额定值: 额定最大 26.4 V , “打开” 电压 10.8 V 或更小, “关闭” 电流 0.6 mA 或更小 (触发器输入端子: 额定最大 26.4 V , “打开” 电压 10.8 V 或更小, “关闭” 电流 1.0 mA 或更小)*4 专用于与 IV 系列 传感器的连接;

1.2 误差来源分析

论文将手眼标定误差总结为三个方面, 第一是设备带来的误差 (相机、机械臂、标定物等), 这个误差是固定的客观存在的, 不做考虑; 第二是在使用标准圆作为标定物时, 对于球心拟合的误差; 第三是利用手眼关系对式子进行求解时的误差。

本文在对误差作分析时, 使用实现数据并画出图标作为分析依据。

1.3 手眼关系及误差表达式

大部分论文都会把手眼标定关系转换为 $AX=XB$ 问题, 从而去求解手眼矩阵 X , 但是本文使用的是 ${}^b_0T_0T_0^bT_0^bT$, 其中 b_0T 表示相机到基底的转换矩阵 (即最终的手眼矩阵) 记为 A , 0_0T 表示通过拟合圆进而求出的球心坐标记为 b , ${}^b_0T_0^bT$ 前者表示法兰盘到基底的坐标变换, 后者表示标定物及球心坐标到法兰盘的坐标转换 (文中提到此项可以根据硬件设备信息求得, 也就是说对于眼在手外, 标定物的坐标可以作为已知条件), 两项乘积记为 c ,

则有 $Ab_i=c_i$ ，那么误差 $\Delta=Ab_i-c_i$ 。补充^[23-10-26]：本文中的原意应为 ${}^B_cT_oP={}^B_eT_oP$ ， P 是坐标， T 是转换矩阵，两者差异巨大。 ${}_oP$ 很好理解，就是拟合球心在相机坐标系下的坐标； B_oP 则是根据标定物的硬件尺寸推算出物理球心在法兰盘坐标系下的坐标。然后利用最优化求解误差方程，得出手眼矩阵 $A={}^B_cT$ 。

1.4 *球心拟合误差（需要看一下论文 3 中线激光结构光相机的成像原理）

根据线激光结构光原理，其成像误差受到入射角的影响，文中提到拟合圆半径越小，则激光光斑区域高度差越大，误差越大（此处应该也与入射角有部分关系）。同时，拟合圆所用到圆弧部分，两端和中间点的误差也很大（使用下面将要说到的优化算法来减小误差）。

在这一部分，文中的逻辑是：首先使用最小二乘法对球心坐标进行拟合，然后利用 $Ab-c$ 求出误差，分析出拟合半径越大，误差越小；然后，对拟合曲线上的点到拟合圆心的距离进行误差分析，得出曲线两端和中间段的误差较大，进而使用改进最小二乘法进行切面圆的拟合。而在实际操作中，选择 RANSAC 可能效果就和改进的最小二乘法相似。但是这一部分的思想很有意思。

1.5 *手眼矩阵计算误差及改进

1.6 其他

文中最后还是用不同的标准球进行验证。

表 2 标定误差		
优化方案	最大误差/mm	平均误差/mm
原始拟合圆 + 原始求解	0.533 5	0.297 0
加权拟合圆 + 原始求解	0.243 6	0.181 8
加权拟合圆 + 加权求解	0.228 4	0.149 5

2. ★0.0815(高精度)-基于标准球的机器人手眼标定方法改进研究

收获小结：终于了解了误差分析来源，验证了论文 1 中的优化方案，精度提高了一个数量级，最大误差仅为 0.0815mm

求解手眼矩阵方法：四元数

2.1 设备信息

FANUC 机器人公司的 Robot LRMate 200iD 型机器人

机器人重复定位精度为 0.02mm，线性度 无。

基恩士轮廓测量仪 LJ-V7060

型号			LJ-V7060
安装状态			漫反射
参考距离			60 mm
测量范围	Z 轴 (高度)		±8 mm (F.S.=16 mm)
	X 轴 (宽度)	近	13.5 mm
		参考距离	15 mm
		远	
光源	类型		蓝色半导体激光
	波长		405 nm(可见光)
	激光分类		2M 类激光产品 ^{*1} (IEC60825-1、FDA(CDRH) Part 1040.10 ⁻²)
	输出		10 mW
(参考距离上的)光点直径			约 21 mm x 45 μm
重复精度	Z 轴 (高度)		0.4 μm ^{*3,4}
	X 轴 (宽度)		5 μm ^{*3,5}
线性	Z 轴 (高度)		±0.1% 的 F.S. ^{*6}
轮廓数据间隔	X 轴 (宽度)		20 μm
取样频率(触发间隔)			最快 16 μs (高速模式)、最快 32μs (高性能模式) ^{*7}
温度特征			0.01% 的 F.S./°C
环境抗耐性	外壳防护级		IP67 (IEC60529) ^{*8}
	环境光照		白炽灯：最大 10,000 lux ^{*9}
	环境温度		0 至 +45 °C ^{*10}
	相对湿度		85%RH 以下 (无凝结)
	抗震性		10 至 57 Hz、双振幅 1.5 mm、X,Y,Z 方向各 3 个小时
	耐冲击性		15 G/6 ms
材质			铝
重量			约 450 g

^{*1} 请不要用皮肤直接接触激光光束。激光光束为Ⅱ类激光。眼镜镜片及面部防护镜等不能阻挡激光光束。

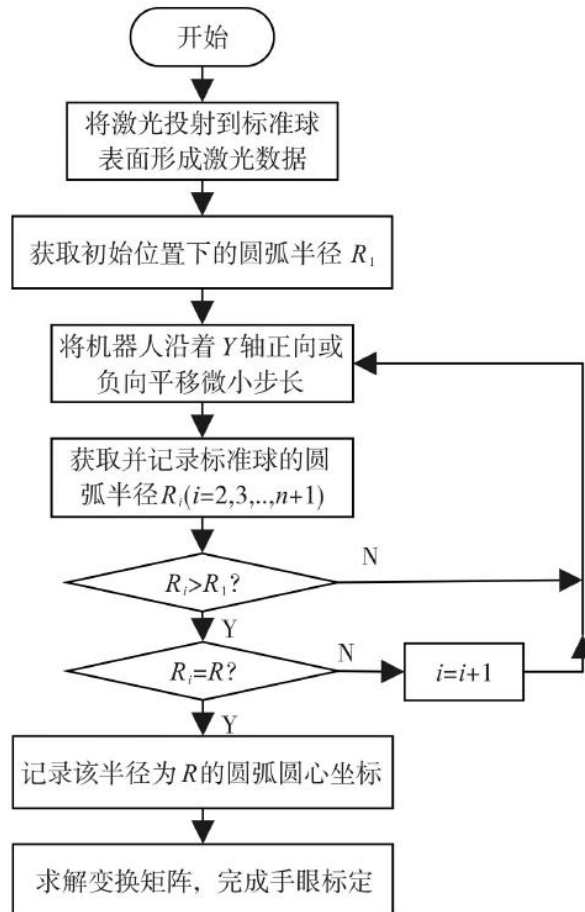
2.2 手眼关系及标定流程

此论文中将第一次测量计算得到的球心作为标准去计算后续球心与它的误差。

$$B_{t1} X \begin{pmatrix} x_{s1} \\ n_1 \cdot d \\ z_{s1} \\ 1 \end{pmatrix} = B_{t2} X \begin{pmatrix} x_{s2} \\ n_2 \cdot d \\ z_{s2} \\ 1 \end{pmatrix} = \dots = B_{tn} X \begin{pmatrix} x_{sn} \\ n_n \cdot d \\ z_{sn} \\ 1 \end{pmatrix}$$

始终的 $n_i \cdot d$ 表述不太正确，应该是 $(-a+b) \cdot d$ ，负方向移动 a 次，正方向移动 b 次。补充^[23-10-26]：此处误差会很大 $\{x1, n1d, z1\}$ 处的位姿和 $\{x1, 0, z1\}$ 处的位姿不同，所以应该直接读取半径最大处的球心坐标和机器人位姿。

此论文中直接表示对最大半径处进行扫描拟合圆可以更大程度上的减小误差。初次投射就应该将激光线基本对准最大半径处。



2.3 其他

这篇论文最亮的地方是最后的精度，同时基本继承了论文 1 中提到的误差改进方法。

3.学位论文-基于 3D 机器视觉的工业机器人跟踪涂胶系统_陈琳

收获小结：论文整体架构全面，项目所含技术全面。包含机器人运动学、相机成像原理等

4. 线激光器的手眼标定方法

收获小结：简单易操作，误差精度在 0.01~0.17mm(标准球)，0.5~5mm(空间定点)。使用几何法求解标准球切面拟合圆(圆上任意四点连线的垂直平分线的交点)。

4.1 设备信息

LMI Gocator2340

规格参数	
型号	Gocator 2340
品牌	LMI
传感器类型	线激光传感器
轮廓数据点数	1280
视野FOV (mm)	96~194
测量范围MR (mm)	210
净距离 (mm)	190
X方向分辨率 (μm)	95~170
Z方向分辨率 (μm)	13~37
Z方向线性度 (% of MR)	0.01
Z方向重复精度 (μm)	1.2
激光等级	2, 3R, 3B
扫描速率/时间	0.17 ~ 5 kHz
数据接口	GigE
电源	24 ~ 48 VDC, 13W
防护等级	IP67
机械尺寸	49 mm × 75 mm × 197 mm (顶部安装)
工作温度	0°C ~ +50°C
重量	0.94 kg

ABB IRB120

重复精度 0.01mm

4.2 其他

方法简单，精度还可以，手眼标定入门文章。

介绍了空间顶点和标准圆（球心作为空间定点）的标定方法。

5. Pointnet

5.1 解决的问题

可以对点云进行直接处理 – 网络结构简单 – 鲁棒性强 – 可完成多任务 – 复杂度低

6. (难)基于 3D 标定块的机器人与 3D 相机手眼标定研究

收获小结：矩阵运算和公式很多；多利用 3D 相机 z 轴精度高的因素；引言部分介绍许多求解手眼矩阵的方法；Tsai 方法的过程即所用公式

6.1 没有看懂的地方

1. 找寻特征点，利用最小二乘法建立并求解平方方程的地方没有看懂。
2. 利用特征点建立标定物坐标系的地方没有看懂，后面求解手眼矩阵的方法也就没看。

6.2 文章亮点

1. 文章中提到 z 轴精度高，误差小的因素，那在之后的方案设计中就需要考虑到这些因素。
2. 文中的引言介绍了许多求解手眼矩阵的方法，回头可以参考一下参考文献。
3. 文中大量使用了矩阵表达式和矩阵运算，以及一些思想方法的推导过程。

6.3 TSAI 的运算过程

$$\begin{cases} \mathbf{r}_{gij} = R_{\text{odrigues}}(A) \\ \mathbf{r}_{cij} = R_{\text{odrigues}}(B) \end{cases}, \begin{cases} \mathbf{N}_{rgij} = \frac{\mathbf{r}_{gij}}{\|\mathbf{r}_{gij}\|} \\ \mathbf{N}_{rcij} = \frac{\mathbf{r}_{cij}}{\|\mathbf{r}_{cij}\|} \end{cases}, \quad (4)$$

式中： $\mathbf{r}_{gij}, \mathbf{r}_{cij}$ 为代表机器人与标定块姿态的旋转向量； $R_{\text{odrigues}}(*)$ 为罗德里格斯旋转公式； $\mathbf{N}_{rgij}, \mathbf{N}_{rcij}$ 为归一化的旋转向量。接着使用修正的罗德里格斯参数表示姿态变化并计算初始旋转向量，之后再计算旋转向量，

$$S_{\text{kew}}(\mathbf{P}_{gij} + \mathbf{P}_{cij})\mathbf{P}'_{cg} = \mathbf{P}_{cij} - \mathbf{P}_{gij}, \quad (5)$$

式中：
$$\begin{cases} \mathbf{P}_{gij} = 2 \sin \frac{q_{gij}}{2} \mathbf{N}_{rgij} \\ \mathbf{P}_{cij} = 2 \sin \frac{q_{cij}}{2} \mathbf{N}_{rcij} \end{cases}; S_{\text{kew}} \text{ 为反对称运算}; \mathbf{P}_{gij},$$

$\mathbf{P}_{cij}, \mathbf{P}'_{cg}$ 为修正后的旋转向量。假设一个三维向量为 $\mathbf{V} = [v_x, v_y, v_z]$ ，则反对称矩阵为

$$\begin{cases} S_{\text{kew}}(\mathbf{V}) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \\ \mathbf{P}_{cg} = \frac{2\mathbf{P}'_{cg}}{\sqrt{1 + |\mathbf{P}'_{cg}|^2}} \end{cases}. \quad (6)$$

最后，先求出标定方程中 X 的旋转部分 (\mathbf{R}_{cg})，后求出平移部分 (\mathbf{T}_{cg}) 如下式完成标定，

$$\mathbf{R}_{cg} = \left(1 - \frac{|\mathbf{P}_{cg}|^2}{2}\right) \mathbf{I} +$$

$$\frac{1}{2} [\mathbf{P}_{cg} \mathbf{P}_{cg}^T + \sqrt{4 - |\mathbf{P}_{cg}|^2} S_{\text{kew}}(\mathbf{P}_{cg})], \quad (7)$$

式中： \mathbf{I} 为单位矩阵。

$$(\mathbf{R}_{gij} - \mathbf{I})\mathbf{T}_{cg} = \mathbf{R}_{cg}\mathbf{T}_{cij} - \mathbf{T}_{gij}. \quad (8)$$

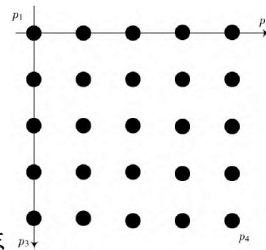
式中： \mathbf{T}_{cij} 为标定方程中 A 的平移部分； \mathbf{T}_{gij} 为标定方程中 B 的平移部分。

7. (好论文~~垃圾论文~~) 面向物料抓取机器人的新型手眼标定方法

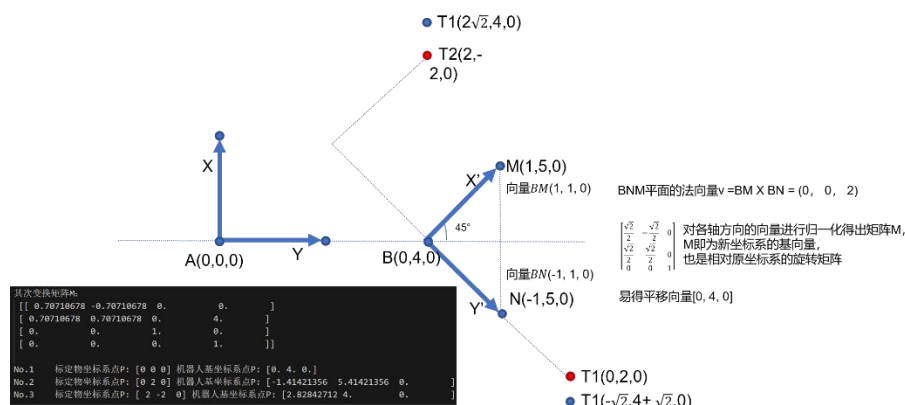
收获小结：更加了解空间坐标系的变换

7.1 方法(眼在手外)

总的来说，本文利用自制的标定板(规则排列的数个圆)，然后使用 ${}^B T_C T = {}^B T$ 直接获得手眼矩阵，方法简单。~~粗暴，但是精度异常可怜~~，首先是 ${}^B T$ ，其获取类似于机器人外部工具坐标系建立的方法，~~且没有改进，所以精度很低~~。所以，机器人的 TCP 标定以及标定物的坐标系建立成为影响精度的主要因素。



首先在标定板上建立标定板坐标系 $p_3 \downarrow$ p_4 , 然后使用 TCP 测量标定板上任意四个点在基坐标系(B)中的坐标, 根据四个点的坐标建立标定板坐标系的基向量直接作为旋转矩阵(文中并没有提, 但是需要进行归一化才可以作为旋转矩阵, 自己画图测试误差很大。没有问题, 很新颖, 对空间变换又多了一个理解)



之后根据线性插值法，求解了标定物到相机的转换矩阵 cT ，文中没有提什么相机，暂时理解为相机外参。

则手眼矩阵 ${}^B_cT = {}^B_oT_o^oT = {}^B_oT_o^oT^{-1}$

7.2 误差分析

本文误差在毫米级别，精度很差。从手眼矩阵方程式来看，误差主要来源于：

1. ${}^B T$ 使用 TCP 去触及标定板上的点获得坐标，那么 TCP 的精度和获取标定板上点坐标的误差就是 ${}^B T$ 的误差来源。
2. ${}^C T -$ 外参矩阵求解的误差

7.3 改进方案(待完善)

1. 标定板上坐标需要多获取几组，同时要保证机器人的多姿态
2. ${}^B T$ 的求解方法(如何从多组点中求出精度最高的坐标系向量)

8. 正交约束下的机器人线结构光手眼标定

收获小结：使用标定板测量，新的手眼矩阵关系

(多环境验证，平均精度不超过 0.15mm，但是论文结尾说系统误差在 0.3mm 左右)

8.1 设备信息(硬件太好)

运动机构采用 ABB 公司的 S7 - 1200 六轴工业机器人，重复定位精度 $\pm 0.01 \text{ mm}$;
线结构光传感器采用北京同舟兴业科技有限公司的 GF3 - 150, 测量精度 $\pm 0.03 \text{ mm}$; 使用高精度数控中心加工的方形不锈钢平板作为标定靶标，边长为 200 mm，直角精度优于 $\pm 0.05^\circ$ 。

8.2 方法(眼在手上)

使用带有正交关系的标定物，利用其正交关系进行手眼矩阵的求解。同时使用了新的手眼关系求解方程。 ${}^B\mathbf{T}_E \mathbf{x} \quad {}^E\mathbf{T}_C \mathbf{x} \quad \mathbf{P}_C = \mathbf{P}_B$, 然后进行运算整合有 $\mathbf{Ax} = \mathbf{b}$ (L 指相机坐标系):

$$\mathbf{A} = \begin{bmatrix} {}^B E'_{11} \cdot x_l & {}^B E'_{12} \cdot x_l & {}^B E'_{13} \cdot x_l & {}^B E'_{11} \cdot z_l & {}^B E'_{12} \cdot z_l & {}^B E'_{13} \cdot z_l & {}^B E'_{11} & {}^B E'_{12} & {}^B E'_{13} & {}^B E'_{1x} \\ {}^B E'_{21} \cdot x_l & {}^B E'_{22} \cdot x_l & {}^B E'_{23} \cdot x_l & {}^B E'_{21} \cdot z_l & {}^B E'_{22} \cdot z_l & {}^B E'_{23} \cdot z_l & {}^B E'_{21} & {}^B E'_{22} & {}^B E'_{23} & {}^B E'_{2y} \\ {}^B E'_{31} \cdot x_l & {}^B E'_{32} \cdot x_l & {}^B E'_{33} \cdot x_l & {}^B E'_{31} \cdot z_l & {}^B E'_{32} \cdot z_l & {}^B E'_{33} \cdot z_l & {}^B E'_{31} & {}^B E'_{32} & {}^B E'_{33} & {}^B E'_{3z} \\ & & & & & & & & & 1 \end{bmatrix}$$
$$\mathbf{x} = [{}^E L'_{11} \quad {}^E L'_{21} \quad {}^E L'_{31} \quad {}^E L'_{13} \quad {}^E L'_{23} \quad {}^E L'_{33} \quad {}^E L'_{1x} \quad {}^E L'_{2y} \quad {}^E L'_{3z} \quad 1]^T, \mathbf{b} = [x_b \quad y_b \quad z_b \quad 1]^T$$

控制系统在具有正交关系的边上分别取 N 个点，每两个点可组成一条向量且与另一条边正交，即：

$$v_p^i \cdot v_q^j = 0 \quad (5)$$

由式(3)与(5)相结合得：

$$[(A_p^{i_1} - A_p^{i_2}) x]^T \cdot [(A_q^{j_1} - A_q^{j_2}) x] = 0 \quad (6)$$

于是, 标定问题在公式(4)和(6)的共同限制下转化为公式(7)的最优化问题, 该问题可采用非线性迭代法^[17]计算求解：

$$f(x)_{\min} = \sum_{i^*=1}^N \sum_{j^*=1}^N [(A_p^{i_1} - A_p^{i_2}) x]^T \cdot [(A_q^{j_1} - A_q^{j_2}) x], \quad \text{s. t.} \begin{cases} {}^E_L R_{11}^2 + {}^E_L R_{21}^2 + {}^E_L R_{31}^2 = 1 \\ {}^E_L R_{13}^2 + {}^E_L R_{23}^2 + {}^E_L R_{33}^2 = 1 \end{cases}$$

求解可得手眼矩阵。

8.3 误差验证方法

使用现有手眼矩阵, 求解各点在机器人基坐标系下的坐标, 然后进行直线拟合和平面拟合计算手眼标定误差。

8.4 误差分析 & 方法迁移

手眼关系等式: ${}^B_E T x {}^E_C T x P_C = P_B$ 眼在手外则为 ${}^B_C T x P_C = P_B$

对于眼在手上而言误差主要来源于在标定物上取点和最优化方程的求解。而对于眼在手外而言无法直接使用这个方法, 因为如果把标定物放在机械臂上, 则无法保证采集到的点所在的边是正交关系。但是可以将标定物固定, 使用 TCP 去有正交关系的边上取点, 眼在手外也可以使用这个方法, 因为对于眼在手外而言, 并不需要改变机器人的位姿去提高手眼矩阵的精度, 其要求的转换关系中的两个坐标系都是固定不动的, 受到额外误差的影响很少。使用该方法的话, 直接套就好了。只是本文的方法中矩阵 A 是已知参数, 向量 x 是未知参数, 我们的唯一一个矩阵就是我们要求的手眼矩阵。

9. PointNN_CVPR2023_Starting_From_Non-Parametric_Networks_for_3D_Point_Cloud_Analysis_CVPR_2023_paper

0. 细节不太懂-无参数-可作插件-性能可

1. 总结

1.1 动机

自从 Pointnet 开始使用深度学习直接处理点云后，大家都在追求更高的性能的同时，网络的参数和需要的算力都越来越高。

2.2 做了什么

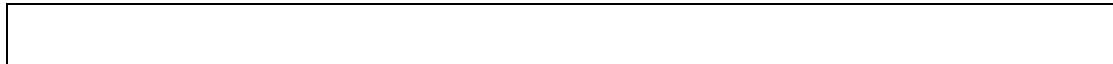
提出了一种无参数的模块，此模块主要利用 FPS、K 紧邻、池化以及 transformer 中的三角变换来实现。

同时，此模块可通过添加简单的线性层来提高性能，也可以作为插件，让已经训练好的网络进一步提升性能。

2.3 取得的成绩

详见论文

3. 技术路线



10. ECCV_2020 PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding

10.1 点云深度学习网路现状

1) 缺乏大规模和高质量的数据: 与 2D 图像相比, 3D 数据更难收集, 标记成本更高, 并且传感设备的多样性可能会带来巨大的领域差距;2) 缺乏统一的骨干网架构: 与 2D 视觉相比, ResNets 等架构已被证明是成功的预训练和网络调优的骨干网, 点云网络架构设计仍在不断发展;3) 缺乏一套全面的数据集和用于评估的高级任务。

10.2 论文主要贡献 – 预训练网络

在相同世界坐标中对齐生成的两个视图[使用 KDTree 或者八叉树] (x_1 和 x_2), 即它们在相同的世界坐标系下表示相同的空间位置。确保两个视图之间的点能够有意义地进行对比和匹配。

使用对应关系映射 M 建立两个视图之间的点的对应关系。具体来说, 如果 $(i, j) \in M$, 则表示在两个视图中的点 x_{1i} 和 x_{2j} 是匹配的, 即它们在相同的物体表面上。

该预训练任务的框架是为了学习点云的表示, 其中点云是三维空间中的一组点。首先, 从分布中采样一个点云 x , 然后生成两个在相同世界坐标中对齐的视图 x_1 和 x_2 。接着, 计算这两个视图之间的对应关系映射 M , 以获取匹配的点对。为了增加任务的难度, 引入了随机的刚性几何变换 T_1 和 T_2 , 如旋转、平移和缩放, 将两个视图进一步变换。最后, 通过对两个视图中的点应用对比损失来训练模型, 以使匹配的点更加接近, 而不匹配的点更加远离。这个方法与 2D 图像的对比学习相似, 但在这里对比学习是在点级别进行的, 目的是学习点云的紧凑表示, 特别是用于几何局部描述符的度量学习。

10.2.1 损失函数

Hardest-Contrastive Loss The first loss function, hardest-contrastive loss we try, is borrowed from the best-performing loss design proposed in FCGF [10], which adopts a hard negative mining scheme in traditional margin-based contrastive learning formulation,

$$\mathcal{L}_c = \sum_{(i,j) \in \mathcal{P}} \left\{ [d(\mathbf{f}_i, \mathbf{f}_j) - m_p]_+^2 / |\mathcal{P}| + 0.5 [m_n - \min_{k \in \mathcal{N}} d(\mathbf{f}_i, \mathbf{f}_k)]_+^2 / |\mathcal{N}_i| + 0.5 [m_n - \min_{k \in \mathcal{N}} d(\mathbf{f}_j, \mathbf{f}_k)]_+^2 / |\mathcal{N}_j| \right\}$$

$$\mathcal{L}_c = - \sum_{(i,j) \in \mathcal{P}} \log \frac{\exp(\mathbf{f}_i \cdot \mathbf{f}_j / \tau)}{\sum_{(\cdot, k) \in \mathcal{P}} \exp(\mathbf{f}_i \cdot \mathbf{f}_k / \tau)}$$

第二个名为 PointInfoNCE 参考了 2D 处理中的 InfoNCE(对比学习)

10.2.2 基于 U-Net 的 SRU-Net 网络

Here we show the SR-UNet architecture that is used as a shared backbone in our paper for both the pre-training and the fine-tuning phases. This U-Net architecture was originally proposed in [9] for ScanNet semantic segmentation.

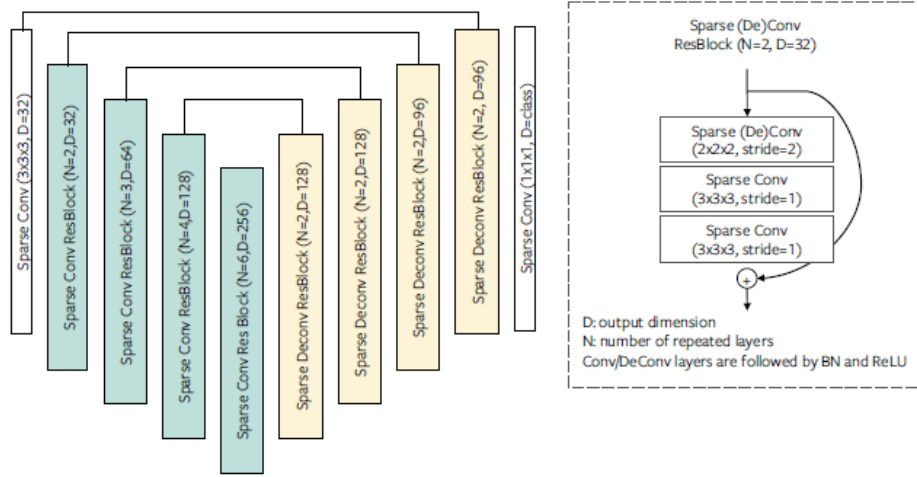


Fig. 3: SR-UNet architecture we used as a shared backbone network for pre-training and fine-tuning tasks. For segmentation and detection tasks, both the encoder and decoder weights are fine-tuned; for classification downstream tasks, only the encoder network is kept and fine-tuned.

11. CVPR_2020_GraphTER_Unsupervised_Learning_of_Graph_Transformation_Equivariant_Representations_via_Auto-Encoding_CVPR_2020_paper

0. (无监督)与论文 10 异同

同为无监督网路处理点云，论文 10 中表达的思想是进行预训练 – 通过两个不同的视点进行网络训练

在本文中，有异曲同工之处，论文 10 可以看作是对整点云进行(旋转)变换，而在本文中也是对点云进行仿射变换，但是是在点级的基础上，且不仅限于旋转，包含平移、旋转和裁剪。

1. 算法流程

算法的流程如下：

步骤 1：给定一个包含 N 个节点的图形，将其表示为图信号 $X = \{x_1, x_2, \dots, x_N\}$ 。然后，在训练的每个迭代中，从图中随机选择一个节点子集 S 。节点子集可以通过全局采样或局部采样的方式获取。局部采样是在图中的局部节点集合进行采样，而全局采样是在整个图中进行随机采样。这样做的目的是将全局和局部的图结构性质以不同尺度进行建模，并提高计算效率[4a]。

步骤 2：在子集 S 的每个样本 x_i 上绘制一个节点变换 t_i ，这可以是各向同性或各向异性的。对应地，与变换图相关的图 \tilde{A} 也会在实际应用样本节点变换之后在 A 的基础上进行等变换。节点变换 t 的选择会根据节点的特征和图的连接情况来确定[5a]。

步骤 3：构造一个基于 k 最近邻的图，以利用节点之间的连接。这个图的邻接矩阵 A 在应用了特定节点变换之后会发生变化。然后，使用多层常规边卷积将其堆叠起来，形成最终的编码器。边卷积通过边权重 $\tilde{a}_{i,j}$ 将节点的特征与相邻节点的特征聚合起来。由于底层图的边信息与节点的变换相对应，边卷积可以从原始和变换后的边信息中提取高级特征[5b][5c]。

步骤 4：使用一个自动编码器（auto-encoder）学习图信号的表示和节点变换的解码。编码器将节点特征映射到连续表示，解码器从这个连续表示中重建原始和经过变换的图的特

征。节点变换 t 通过解码器来估计，然后反向传播损失来更新编码器和解码器的参数 [4b][4c][5d]。

步骤 5：通过最小化损失函数来端到端地训练整个网络[4d]。损失函数的期望是在每次训练的迭代中，通过对采样子集 S 上的节点变换进行评估而计算的[4d]。

12. CVPR_2022_CrossPoint_Self-Supervised _Cross-modal_Contrastive_Learning_for_3D_P oint_Cloud_Understanding

12.1 文献阅读

浅读 CVPR2022-CrossPoint - 知乎 (zhihu.com)
再读 CVPR2022-CrossPoint（附代码） - 知乎 (zhihu.com)