
摘要

摘要

das

关键词：无人驾驶，激光雷达，障碍物检测，深度学习，多传感器融合

ABSTRACT

ABSTRACT

With the widespread engineering applications ranging from broadband signals and non-linear systems, time-domain integral equations (TDIE) methods for analyzing transient electromagnetic scattering problems are becoming widely used nowadays. TDIE-based marching-on-in-time (MOT) scheme and its fast algorithm are researched in this dissertation, including the numerical techniques of MOT scheme, late-time stability of MOT scheme, and two-level PWTD-enhanced MOT scheme. The contents are divided into four parts shown as follows.

.....

Keywords: time-domain electromagnetic scattering, time-domain integral equation (TDIE), marching-on in-time (MOT) scheme, late-time instability, plane wave time-domain (PWTD) algorithm

目 录

第1章 绪论	1
1.1 研究工作的背景与意义	1
1.2 无人车三维障碍物检测的国内外研究历史与现状	2
1.3 本文的主要贡献与创新	2
1.4 本论文的结构安排	2
第2章 三维感知传感器机构设计	3
2.1 三维感知传感器机构的机械结构设计	3
2.1.1 机械结构运动原理	4
2.1.2 曲柄连杆机构的设计	5
2.2 三维感知传感器机构的电路设计	7
2.2.1 驱动器	7
2.2.2 执行机构	7
2.2.3 传感器	8
2.2.3.1 角度传感器	8
2.2.3.2 激光雷达	8
2.2.4 主控板	8
2.2.5 电路拓扑	9
2.3 三维感知传感器机构的软件设计与运动控制	10
2.4 本章小结	10
第3章 点云的多帧融合与激光雷达和相机标定	12
3.1 激光雷达点云的多帧融合	12
3.1.1 一种朴素的多帧融合策略	12
3.1.2 点云的运动畸变的形成与矫正	12
3.1.2.1 点云的运动畸变	13
3.1.2.2 运动畸变的矫正	14
3.1.3 纠正运动畸变后的多帧融合策略	14
3.2 激光雷达与相机的标定	16

目录

3.2.1 标定板	16
3.2.2 相机坐标系中的三维特征点提取	17
3.2.3 LiDAR坐标系中的三维特征点提取	18
3.2.4 刚体变换求解	18
3.3 本章小结	20
第4章 基于视觉激光融合的三维障碍物检测方法	21
4.1 相机与激光雷达标定	21
4.2 基于yolo的视觉、激光结合的三维障碍物检测	21
4.3 本章小结	21
第5章 实验验证与结果分析	22
5.1 全文总结	22
5.2 后续工作展望	22
参考文献	23
致 谢	23
外文资料原文	24
外文资料译文	26

缩略词表

主要符号表

第1章 绪论

1.1 研究工作的背景与意义

近几年来，自动驾驶技术取得了长足的进步，而其中关键的技术就是多传感器的环境感知与融合。环境感知的一个重要环节便是障碍物检测。目前，虽然基于图像的障碍物检测已经取得了卓有成效的进步，然而相较于三维障碍物检测，二维障碍物检测有以下缺陷：

1. 基于单目相机的障碍物检测没有尺度信息，无法恢复出目标的三维坐标。
2. 基于双目相机的障碍物检测，当基线较短时，测量距离较长（5m以上）的物体时计算出来的距离信息很不准确，而当基线较长时，近处物体的检测又容易出现在两个相机的视野盲区之中，从而导致无法三角化而得出距离信息。

基于上述原因，越来越多的目光聚焦在了基于激光雷达（LiDAR）的三维障碍物检测。LiDAR是Light Detection And Ranging的缩写，中文译作“激光探测与测量”，一般指多线数的三维激光雷达传感器。相较于相机图像，激光雷达的点云拥有以下几点优势：

1. 测量范围广。目前的激光雷达的测量有效距离基本都在0.5-100米左右，远高于双目相机三角测距的适用范围。
2. 测量精度高。激光雷达的测距误差可达厘米级，同样优于双目相机的测距结果。

目前最常见的旋转式激光雷达，其本质是多个激光束旋转后对每个时刻的测距结果进行保留与叠加，最后再以点云的形式发布出去。决定激光雷达的分辨率的一个参数为其激光束的个数，一般称之为激光雷达的线数。目前常用的激光雷达线数有16线、32线、64线等，其中由于低线数的激光雷达生成的点云在测量远距离物体时密度较低。举例来说，当使用16线激光雷达检测到20米处的障碍物时，其16线激光束两两之间的距离可以达到70cm，相当于检测20m处的人时，只能够有两线激光束能够返回距离。因此，低线数激光雷达点云的稀疏性较大地制约了三维障碍物检测任务的准确率。

通常在自动驾驶的无人车系统中会在车的四周装上多个16线的激光雷达进行点云融合，或者直接采用线数更高的激光雷达来做障碍物检测的任务。然而目前三维激光雷达造价不菲，无人车系统中光是64线激光雷达的成本就奖金十万美金，如此高昂的成本在一定程度上限制了无人驾驶汽车的普及与推广。

鉴于上述存在在问题，本文希望能够提出一种基于多帧融合的低线数激光雷达感知机构，使其能够通过增加一个在垂直方向的往复运动，并将该机构上激光雷达的多帧点云融合发布来提高三维点云的稠密性，借而解决低线数激光雷达在障碍物检测问题上由于点云的稀疏性而造成的困难。并且，本文还希望通过融合上述机构发布的点云信息以及相机的图像信息，发挥各个传感器的优势从而提高三维障碍物检测任务的准确率与实现三维障碍物的多类别检测。

1.2 无人车三维障碍物检测的国内外研究历史与现状

根据本文的主要研究方向，下面将对基于激光雷达点云的三维目标分割与检测（3D object segmentation and detection）的研究现状进行调研。

针对三维障碍物的分割问题，目前主要有三类方法来实现

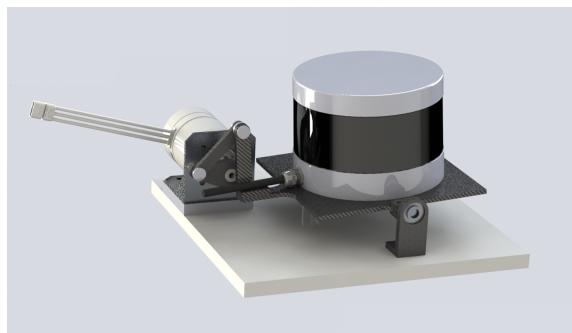
1.3 本文的主要贡献与创新

1.4 本论文的结构安排

第2章 三维感知传感器机构设计

为了能够解决低线数激光雷达在障碍物检测问题上由于点云的稀疏性而造成的困难，本章提出了一种三维感知传感器机构，通过增加三维激光雷达在垂直方向的往复旋转，同时融合多帧激光点云，来增加激光雷达在铅垂方向上的分辨率，实现类似于高线数激光雷达的稠密点云。

2.1 三维感知传感器机构的机械结构设计



(a)



(b)

图 2-1 机构总图(a)solidworks渲染图;(b)实物图

2.1.1 机械结构运动原理

本章所述的机构结构如图2-1(b)所示，其中3508电机提供驱动转矩，曲柄连杆装置将电机的旋转运动转化为激光雷达底座在航向角(yaw)方向上的往复运动，同时绝对值磁编码器记录激光雷达在航向角上的角度变化，以供多传感器融合时使用。

在著名的激光SLAM算法LOAM^[?]中，由于当时的条件限制，其文章作者没有三维激光雷达来进行SLAM，而是用一个舵机给一个二维激光雷达进行竖直方向的往复旋转来增加线数，如图2-2所示。

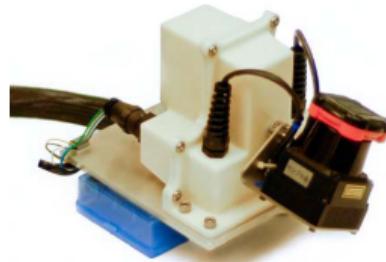


图 2-2 LOAM中的机构设计

该机构最大的优点就是结构简单，这也是本文初次采用的机构设计。然而该机构有以下几个缺点：

- 1.在往复运动中，当运动方向发生改变时，由于舵机控制精度的问题，很难做到平滑换向，并且经常伴随有较大的震动，给之后的传感器融合算法带来了困难。
- 2.LOAM中采用的是二位激光雷达，重量较轻，而本文需要带动三维激光雷达进行往复运动，重量较重（近1kg），长时间使用舵机带动会使舵机产生较为明显的回程间隙，影响角度的测量与后续的传感器融合的效果。

因为上述原因，我们没有采用这种结构设计，而是采用了之前提到的曲柄连杆机构来针对三维激光雷达进行往复运动，相较于上述机构，曲柄连杆结构有以下几个优点：

- 1.换向平滑。执行电机只需要一直向同一方向旋转，曲柄连杆机构就能够自动换向，并且输出的角度曲线近似正弦曲线。
- 2.对执行机构负担小。仅需要较小并且较为恒定的转矩就能够驱动较大的负载做往复运动。

3.对执行机构的控制要求低。在该机构中，无刷电机只需要输出恒定的转矩就能够完成三维激光雷达在偏航角方向上的往复运动，并且经过验证，其角度输出近似正弦曲线，而若采用上述的舵机机构，要想得到相近的角度曲线，则对舵机的软件控制提出了较高的要求。

综上，本文选择曲柄连杆机构作为该机构的驱动机构。

2.1.2 曲柄连杆机构的设计

该三维感知机构的一个难点在于如何设计与电机相连的曲柄连杆机构。这里参照《机械设计基础》^[2]一书中的相应章节对曲柄四连杆机构的连杆长度进行求解。

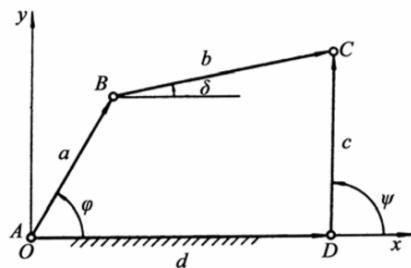


图 2-3 四杆机构的数学模型

如图2-3所示，假设已知该铰链四杆机构两连架杆AB和CD所形成的角度 ψ_1 和 ϕ_1 在三个不同位置下的角度，要求连杆a、b、c、d的尺寸。则根据向量向x、y轴投影，有

$$a \cos \phi + b \cos \delta = d + \cos \phi$$

$$a \sin \phi + b \sin \delta = c \sin \phi$$

将上两式先进行移项，然后作平方和相加，从中消去 δ 后整理有

$$b^2 = a^2 + c^2 + d^2 + 2cd \cos \psi - 2ad \cos \phi - 2acc \cos(\phi - \psi)$$

我们设

$$\begin{cases} R_1 = (a^2 + d^2 + c^2 - b^2) \\ R_2 = d/c \\ R_3 = d/a \end{cases}$$

代入，则上一个式子可以化简为

$$R_1 - R_2 \cos \phi + R_3 \cos \psi = \cos(\phi - \psi)$$

这个式子即为铰链四连杆机构的角位置方程，该方程有三个待定参数 R_1 、 R_2 、 R_3 。故应有三组对应的 ψ_1 和 ϕ_1 角才能得出这个方程的解。将三组 ψ_1 和 ϕ_1 角代入求解该方程后，可以得到四个构件之间的长度关系为

$$\begin{cases} a = d/R_3 \\ c = d/R_2 \\ b = \sqrt{a^2 + c^2 + d^2 - 2acR_1} \end{cases}$$

则根据机构的具体设置情况，知道 a, b, c, d 中的任何一条边的长度后，便可知剩下四条边的长度。

在实际设计中，我们已知 ψ_1 和 ϕ_1 的三组对应角度为

$$\begin{cases} \psi_1 = 30^\circ \quad \phi_1 = 36.3^\circ \\ \psi_1 = 60^\circ \quad \phi_1 = 43.87^\circ \\ \psi_1 = 120^\circ \quad \phi_1 = 35.75^\circ \end{cases}$$

并且根据我们的机构设置，构件 d 的长度为 105.72mm。将这些已知量代入公式中可得

$$\begin{cases} a = 31.6\text{mm} \\ b = 49.18\text{mm} \\ c = 108.37\text{mm} \end{cases}$$

由此，便得到了曲柄机构的连杆构件设计参数。

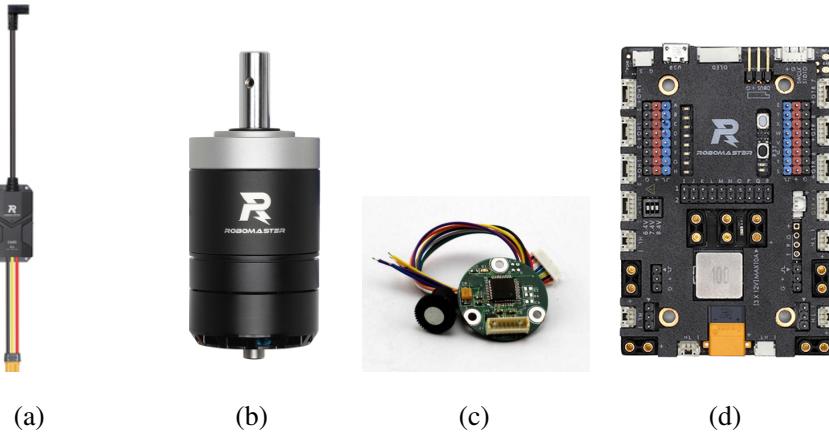


图 2-4 驱动器与传感器(a)C620 电调;(b)M3508无刷电机;(c)磁编码器;(d)RM A型开发板

2.2 三维感知传感器机构的电路设计

2.2.1 驱动器

该三维感知机构采用的驱动器为DJI C620电调，如图2-4(a)所示。该电调支持50-500Hz的PWM（脉宽调制）信号控制以及CAN总线指令控制，最高支持20A的持续电流，支持对CAN总线上的电调快速设置ID，支持通过CAN总线获取电机温度、转子位置和转子速度等信息，切换电机时可无需进行位置传感器的参数校准。

2.2.2 执行机构

该三维感知机构采用的执行机构为DJI M3508无刷电机，如图2-4(b)所示。该电机可搭配上文所述C620电调实现正弦驱动，相比传统方波驱动具有更高的效率、机动性和稳定性。其最高可持续输出力矩为2.8Nm，满足驱动曲柄四连杆机构的需求。

2.2.3 传感器

2.2.3.1 角度传感器

该三维感知机构采用的角度传感器为傲蓝13线磁编码器，如图2-4(c)所示。该编码器采用RS485方式通信，其单圈分辨率为8192cpr，精度为 ± 0.1 度。

该编码器为绝对值式编码器，其相对于增量式编码器不同点在于，增量式编码器以上电时的位置为零点，每次使用都要机械对位；而绝对值式编码器能够记录机构的唯一位置，即单圈内编码器的每一个示值，都唯一对应了空间中机构的位置与角度。考虑到我们曲柄连杆机构的特性，显然绝对值式编码器更加符合我们的要求。

2.2.3.2 激光雷达



图 2-5 速腾16线激光雷达

该三维感知机构采用的激光雷达为速腾聚创的RS-LiDAR-16，如图2-5所示。该激光雷达为16线激光雷达，其测距范围为50cm-150m，精度误差为 $\pm 2cm$ 。垂直视场角为30度，其角分辨率为2度；水平视场角为360度，其角分辨率为0.09-0.36度（对应的点云频率为5Hz-20Hz）。

2.2.4 主控板

该三维感知机构采用的主控板为DJI Robomaster A型开发板，如图2-4(d)所示。该开发板具备类型丰富的接口，包括12V、5V、3.3V电源接口、CAN接口、

UART接口、可变电压PWM接口、SWD接口等。同时该开发板拥有电源输入的防反接、过压保护、缓启动、12V电源输出过流保护、PWM端口的ESD等多重保护。

2.2.5 电路拓扑

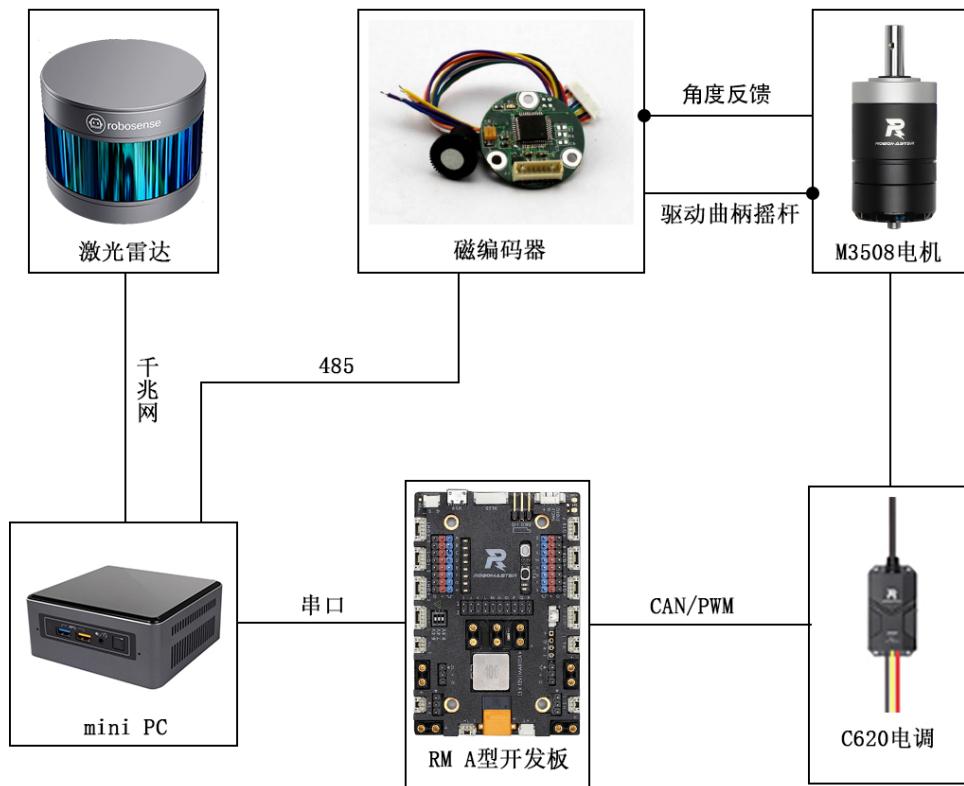


图 2-6 电路拓扑

该三维感知机构的电路拓扑如图2-6所示。激光雷达通过千兆网接口将点云传输到mini PC上，磁编码器通过485转USB与mini PC通信，同时主控通过PWM控制电调输出，调节M3508电机的转速，M3508电机提供曲柄四连杆机构的驱动力矩，而磁编码器又将曲柄机构作用在激光雷达底座上的旋转通过485通信输出到mini PC。

2.3 三维感知传感器机构的软件设计与运动控制

根据上文所述的机械设计以及电路设计，该三维感知机构的软件设计主要实现了以下几个任务：

1. 实现了各个传感器、主控到Mini PC的通信，同时将数据以ROS（Robot Operating System）话题的方式发布出去，以供第二章节提到的多帧融合算法使用。
2. 实现了电机的多档调速功能。为了应对不同的场景，在主控中实现了多档调速功能，以调节曲柄机构的往复运动频率。
3. 实时检测电调的温度信息，提供了基于温度检测的堵转保护（温度过高自动切断控制）。

此外，本文还记录了在电机输出恒定转速情况下的曲柄连杆机构的输出的角度信息，如图2-7所示。该图纵坐标为角度制的输出角度。从图中可以看出，本章所设计的三维感知机构其输出角度近似正弦曲线，并且没有较大的换向震动，相比起上文所提到的舵机的结构拥有稳定可靠的优势。

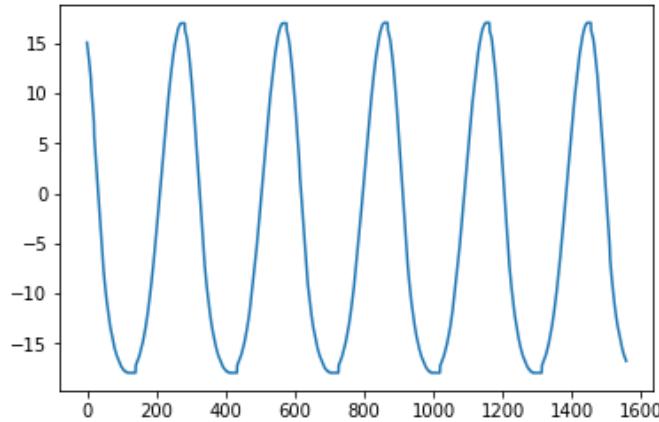


图 2-7 曲柄连杆机构输出角度

2.4 本章小结

本章提出了一种新的三维感知机构，并从该机构的机械设计、电路设计以及软件设计和运动控制三个方面介绍了该机构。在机械方面，本文提出使用无刷电机+曲柄摇杆机构来代替简单的舵机给三维激光雷达提供一个竖直航向角方向上

的往复运动，这种结构的优势在于机构换向流畅、控制简单以及对机构负载小，能够为本文后续章节提到的激光雷达的多帧融合提供结构上的稳定与可靠性。在电路方面，本章利用绝对值式磁编码器对曲柄机构运动的角度进行了记录与输出，相较于增量式编码器，磁编码器不需要保证每次上电时机构都在同一个位置，为机械结构的设计提供了便利。在软件方面，本章实现了各个传感器与主控以及mini PC的通信，主控对无刷电机的多档控制以及对电机的堵转保护，并且绘制了输出角度，验证了机构的可行性。本文的后续章节将利用该机构输出的点云信息以及角度信息来进行点云的多帧融合稠密化，并且在融合的点云上进行三维障碍物的检测与分类。

第3章 点云的多帧融合与激光雷达和相机标定

根据本文第二章所提到的机构，能够将三维激光雷达在其yaw角方向上提供一个有规律的正弦往复运动。本文提出该机构的主要目的为将激光雷达在时间轴上的多帧点云进行融合，进而增加激光雷达在竖直方向的分辨率，达到近似于给激光雷达增加线数的效果。本章将对上述机构得到多帧激光雷达点云进行融合，并且在目前机构的基础上进行激光雷达与相机的标定，从而使得融合后的点云能够应用于第四章提到的基于视觉与激光融合的三维障碍物检测方法。

3.1 激光雷达点云的多帧融合

3.1.1 一种朴素的多帧融合策略

最为直观的策略就是，读取绝对值编码器返回的角度 α ，将激光雷达每帧点云沿着航向角方向旋转 $-\alpha$ 的角度，然后注册多帧的激光雷达点云并发布。

在实际的实现过程中，编码器返回角度的频率约为30Hz，而点云发布的频率约为10Hz，在将点云旋转 $-\alpha$ 角度时，对 α 角进行了线性插值以便获得更加精确的结果。同时，根据计算曲柄机构的角度是增加还是减少，来判断曲柄机构的运动方向，并且将曲柄机构运动角度为一个正弦周期内的点云融合为一帧新的点云输出。

然而这种朴素的融合策略在实际中效果不好，体现在融合后的点云所显示的物体轮廓失真严重，如图3-3(a)所示，原因是没有考虑激光雷达的运动对激光雷达点云生成的影响。

3.1.2 点云的运动畸变的形成与矫正

在激光雷达点云的多帧融合中，如果只是进行简单的历史点云叠加（如上文所示），那么融合后的点云相较于真实情况会有很严重的失真，其原因就在于第二章所提到的三维感知机构在给激光雷达在偏航角方向上的往复运动时，点云会产生不可忽视的运动畸变。本章节首先介绍什么是激光雷达点云的运动畸变，然后提出一种通过插值的方式矫正激光雷达的运动畸变。

3.1.2.1 点云的运动畸变

激光雷达的点云的形成本质上是由激光雷达内部的多个激光测距器将一个旋转周期内的各个测量值记录下来并同时发布后得到的。因此点云中的每个点并不是在同一时刻被测量出来的。如果激光雷达在测量的过程中也在运动，那么激光雷达的点云可能会发生畸变^[?]。

下面以二维激光雷达为例，介绍激光雷达点云运动畸变的形成。

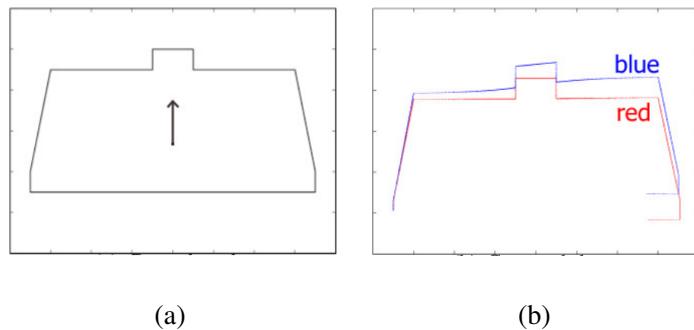


图 3-1 激光雷达运动畸变(a)ground truth;(b)采集得到的数据

图3-1(a)中的黑色的线条表示二维激光雷达处在的真实环境的轮廓图，箭头表示二维激光雷达的运动方向。图3-1(b)中的蓝色的线条表示二维激光雷达的原始数据。注意到其已经发生了畸变，因为二维激光雷达内的激光测距器通过逆时针的方向旋转，使得右上角的数据优先得到，而左上角的数据在激光雷达向箭头方向运动了一段距离之后才进行测量，自然导致了运动畸变的产生。

值得一提的是上图表示的二维激光雷达发生的运动畸变是当激光雷达运动方向为水平运动方向时造成的，而当激光雷达在空间中有垂直方向的旋转运动时，其造成的运动畸变远比水平运动严重。这是因为激光雷达旋转一周的时间普遍在0.1秒左右，其水平运动的距离往往很小可以忽略不计。而当激光雷达有竖直方向的旋转时，即使在0.1s内只有2度的航向角的旋转（这在本文的机构中并不算很快），在测量20m处的物体时，其运动造成的点云畸变可使得点云的同一线上的第一个点与最后一个点的垂直相差将近70cm。

综上所述，对于第二章所述的机构，由于其施加了在激光雷达航向角方向上的旋转，因此导致其在竖直方向上的运动畸变不可忽视，从而简单的叠加点云会导致在做激光雷达的物体检测时的失真。

3.1.2.2 运动畸变的矫正

对于本文所提到的三维感知机构在航向角方向上的旋转所产生的运动畸变的矫正，一个较为朴素的方法是，依次遍历激光雷达每帧点云中的每个点，计算其产生的时间戳 t ，对磁编码器的角度进行插值，计算出在时间戳 t 上的角度 α ，然后将该点绕原点在航向角方向旋转 $-\alpha$ 的角度。这个方法最为直观，然而激光雷达每帧点云高达数十万个点，如果对每个测量得到的点进行插值，则在一秒内要进行近百万次的插值与旋转操作，显然对于无人驾驶汽车上的移动处理器平台来说这是不现实的。

因此本文提出一个假设，设点云中第一个点产生的时间戳为 t_0 ，最后一个点产生的时间戳为 t_k ，则将 t_0 到 t_k 之间的时间均匀分为 n 份，每份长为 Δt 。本文假设 t_0 至 $t_0 + \Delta t$ 、 $t_0 + \Delta t$ 至 $t_0 + 2\Delta t$... $t_k - \Delta t$ 至 t_k 这些时间段，每个时间段内的激光雷达的测距点的产生的时间都是相同的，为其第一个点的产生时间。根据这个假设，每个时间段内的所有点都只要进行相同角度变换即可进行运动畸变的矫正。遵循该假设，则每帧点云只需要进行 n 次角度插值即可，极大地减小了运算量。同时虽然该假设认为同一时间段内的点云是同一时间产生的，每个时间段内的点仍有运动畸变的影响，然而在实验过程中发现，只要当 n 取一个不太小的值（ $n \geq 50$ ），则该假设的所产生的时间段内的运动畸变产生的影响很小，可忽略不计。

在实际的程序实现中，由于激光雷达每帧点云是分段传输的，如图3-2所示。RS-LiDAR-16激光雷达采用UDP协议向PC传输点云信息，而UDP协议相较于TCP协议，发送数据之前不需要双方建立链接，并且发送的数据没有校验，也没有丢包的检测，因此不适合一次性发送大量数据给PC。该激光雷达将一帧点云分为84个UDP包(UDP Packet)，每个包中的点云都是激光雷达旋转 $360/84 = 4.28$ 度后得到的16线的点云的集合。当激光雷达的驱动程序接收到84个UDP Packet之后，将这84个Packet合并成一帧点云输出。

本文修改了RS-LiDAR-16的ROS 驱动程序，将每个UDP Packet不经过合并直接发布出去，同时在多帧融合的程序里，对每个Packet（而不是每帧）分别进行编码器角度的插值与点云的旋转，最后再将旋转后的84个Packet进行合并发布。

3.1.3 纠正运动畸变后的多帧融合策略

如上文所言，对单帧点云进行多次插值之后的多帧融合算法流程如Algorithm 1所示；。

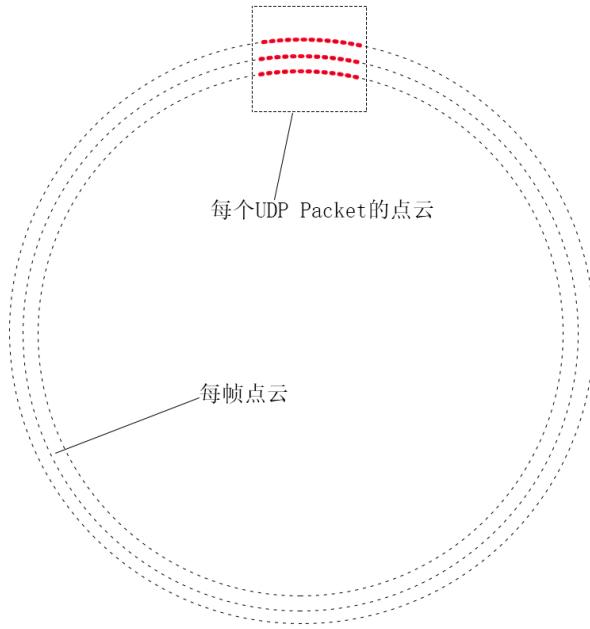


图 3-2 点云的传输

Algorithm 1 Improved multiple pointcloud fusion algorithm

```

1: for packet ∈ point cloud do
2:    $\alpha_1, \alpha_2 = \text{FindClosestAngles}(\text{encoderAngles}, \text{packet.timestamp})$ 
3:    $\alpha = \alpha_1 + (\alpha_2 - \alpha_1) \times \frac{\text{packet.timestamp} - \alpha_1.\text{timestamp}}{\alpha_2.\text{timestamp} - \alpha_1.\text{timestamp}}$ 
4:   rectifiedPacket = RotatePointCloudByYaw(packet,  $-\alpha$ )
5:   rectifiedPointcloud += rectifiedPacket
6: return rectifiedPointcloud

```

在Algorithm 1中，`FindClosestAngles`函数查找在所有的编码器角度中，时间上离`packet`的时间戳最近的两帧编码器的角度值，随后对这两个角度进行线性插值得到`packet`时间戳下的机构的角度。`RotatePointCloudByYaw`函数将特定的点云绕激光雷达坐标系原点旋转指定的角度。最后将每个矫正后的`packet`合并到一个新的矫正后的点云中并发布出去即得到了矫正因机构而产生的运动畸变后的点云。

在图3-3(b)中展示了消除运动畸变后的激光雷达测得的轿车的点云图案。相较于图3-3(a)，其点云图案没有出现明显的失真，并且通过机构进行多帧融合后的点云，能够明显的看出轿车的轮廓细节信息，包括车的反光镜、前挡风玻璃等。

而矫正畸变前的轿车点云则很难分辨出这些细节，并且由于运动畸变的作用，其体积明显比矫正后的点云更大一些。由此证明了本文提出的多帧融合算法拥有较好的矫正畸变的效果。

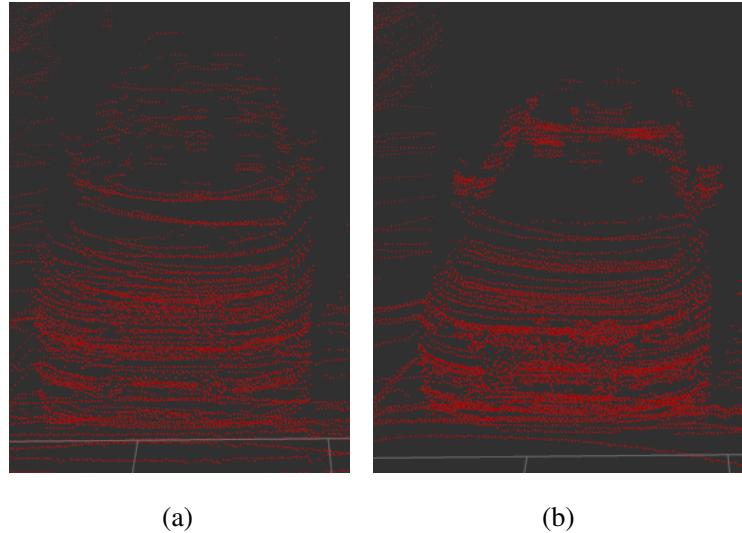


图 3-3 多帧融合后的轿车点云(a)矫正畸变前;(b)矫正畸变后

3.2 激光雷达与相机的标定

自动驾驶无人车是一个多传感器的系统，多传感器信息的融合可以使整个无人车系统的决策更加智能。激光雷达虽然能够获取较为精确的点云信息，然而点云信息只包含了三维距离信息。而相机可以通过图像获得大量信息诸如颜色、纹理信息等，但是其受光照与天气条件影响严重，并且从单目图像中无法获取三维结构信息。为了同时收集三维信息与物体的颜色与纹理信息，激光雷达与相机经常进行传感器的数据融合，来为多传感器系统提供更稳定的数据支持。为了进行数据的融合，首先得知道相机坐标系与激光雷达坐标系之间的旋转与平移关系，因此，激光雷达与相机的标定就显得尤为重要了。

本章节后续将介绍一种文献^[?]提到的，利用两张贴有ArUco Marker^[?]的标定板所提供的3d-3d特征匹配的方法，来进行相机与激光雷达的标定。

3.2.1 标定板

本文参照文献^[?]提到的方法，制作了两块长约40cm，宽约27cm的长方形硬纸

板材质的标定板，并且将两块ArUco Marker粘在硬纸板的固定位置上，如图3-4所示。虽然一块标定板已经可以得到四组3d-3d匹配点来解决标定问题，本文仍然采用了两块标定板，目的是构造多于四对的匹配点来减小标定误差。

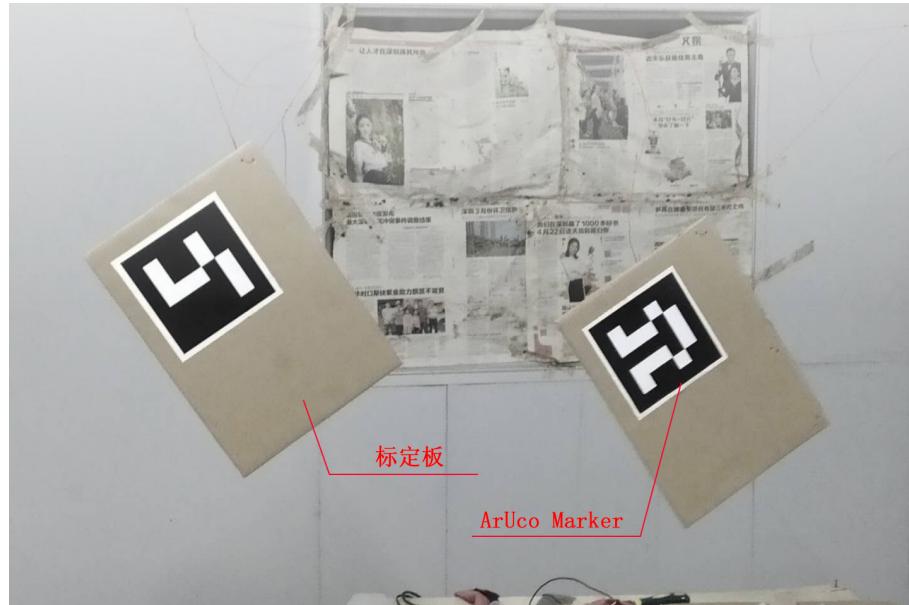


图 3-4 标定环境

3.2.2 相机坐标系中的三维特征点提取

ArUco markers是一种经过特定编码的二维码图案，用以实现对二维码自身的定位与畸变矫正。更多细节可以参考文献^[?]。该文献提出，通过特定的机器视觉算法检测到marker的四个角点后，可以对marker上的二维码进行解码运算，进而求得二维码的id与四个角点的顺序。而通过输入marker的边长后，还能够通过PnP^[?]求解出相机坐标系到marker自身坐标系的转换。



图 3-5 ArUco markers

在本文中，将ArUco marker粘在硬纸板的矩形标定板上，并且测量得出硬纸板的四条边长以及marker在硬纸板中的位置，即可得到硬纸板的四个角点在marker坐标系中的位置。而本文通过ROS中aruco_ros以及aruco_mapping^[?]两个程序包可以检测ArUco marker的位置，进而得到相机坐标系到marker坐标系的转换，从而得到相机坐标系下的硬纸板的四个角点的位置。相机坐标系下角点位置的计算公式为

$$\begin{pmatrix} x_{camera} \\ y_{camera} \\ z_{camera} \\ 1 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{aruco} \\ y_{aruco} \\ z_{aruco} \\ 1 \end{pmatrix}$$

其中， x_{camera} 是相机坐标系中角点的坐标， x_{aruco} 是ArUco marker坐标系中的角点坐标，而上式中的 $[R|t]$ 矩阵则是相机坐标系相对于marker坐标系的欧式变换矩阵。

3.2.3 LiDAR坐标系中的三维特征点提取

本文所参照的标定方法，其在LiDAR坐标系中的三维特征点提取是通过直线拟合的方法进行的。如图3-6所示。图中显示的点为激光雷达的点云投影到相机图像上之后进行边缘检测后所形成的点。在得到该幅图像后，需要手动框选标定板的每条边上的点，随后标定程序会对这些点进行直线拟合，每两条直线的交点即为所求的LiDAR坐标系中的三维点。

3.2.4 刚体变换求解

在得到两个坐标系下的三维特征匹配点之后，两个坐标系之间的 $[R|t]$ 刚体变换矩阵可以通过使用迭代最近点（Iterative Closest Point, ICP）^[?]算法求得。假设 P , Q 为 \mathbb{R}^3 中的一组对应点，ICP算法尝试使经过刚体变换后的 P 点集与 Q 点集的重合误差最小。其求解问题可以表述为式3-1所示。

$$\arg \min_{R \in SO(3), t \in \mathbb{R}^3} \| (RP + t) - Q \|^2 \quad (3-1)$$

一般来说，ICP问题认为对于点集 P 中的每个点，在点集 Q 中与之对应的点为其距离最近的点，依次确认匹配关系后，该算法通过减小两个点集的欧氏距离来对齐两个点集。

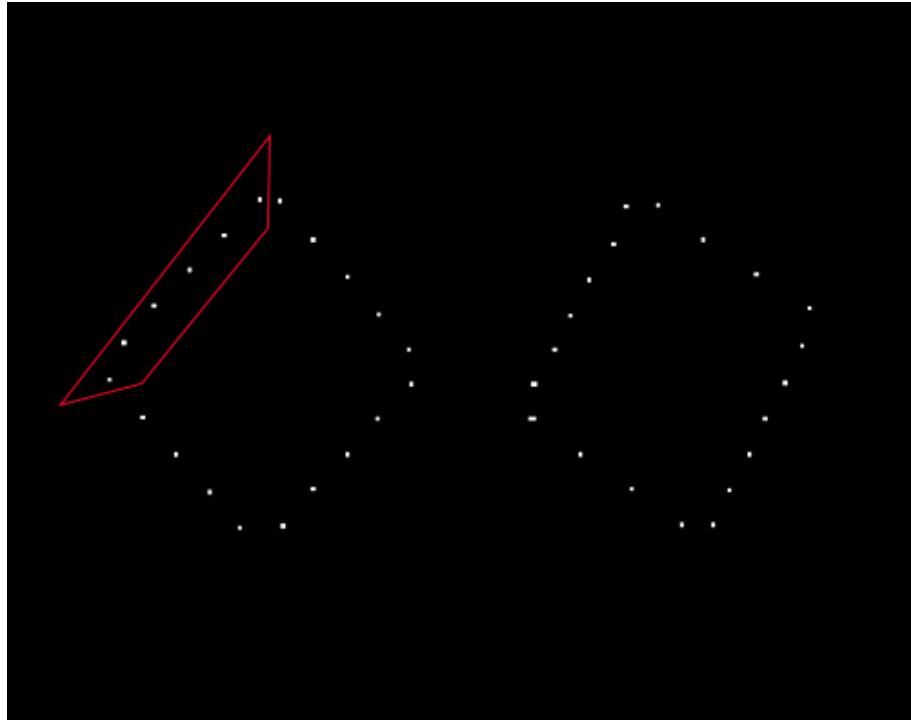


图 3-6 直线拟合提取特征点

ICP的方法找到两组点的匹配关系，在没有良好的初始值的情况下，极易发生误匹配，导致算法最后没办法收敛到一个正确的解。考虑到在本文的标定环境下，两组点的对应关系的是已知的，则两组点之间的 $[R|t]$ 即存在一个闭式解。Kabsch算法^{[?][?]}提供了找到两组对应点间的旋转矩阵的闭式解的方法，而两组点间的平移量可以在进行旋转对齐后求得。下面介绍Kabsch算法的主要步骤。

首先假设旋转已知，求两个点集 P 与 Q 之间的平移量，设

$$F(t) = \sum_{i=1}^n \| (RP_i + t) - Q_i \|^2 \quad (3-2)$$

对上式进行求导，令两边等于0，得

$$\frac{\partial F(t)}{\partial t} = 2 \sum_{i=1}^n ((RP_i + t) - Q_i) = 0 \quad (3-3)$$

因为

$$\frac{\partial F(t)}{\partial t} = 2R \sum_{i=1}^n P_i + 2nt - 2 \sum_{i=1}^n Q_i \quad (3-4)$$

可得

$$t = \frac{1}{n} \sum_{i=1}^n Q_i - R \frac{1}{n} \sum_{i=1}^n P_i \quad (3-5)$$

$$t = \overline{Q} - R\overline{P} \quad (3-6)$$

将式3-6的结果代入式3-2中，有

$$R = \arg \min_{R \in SO(3)} \sum_{i=1}^n \| (R(P_i - \overline{P}) - (Q_i - \overline{Q})) \|^2 \quad (3-7)$$

令

$$X = P_i - \overline{P}, X' = RX, Y = Q_i - \overline{Q}$$

则目标函数变为

3.3 本章小结

第4章 基于视觉激光融合的三维障碍物检测方法

4.1 相机与激光雷达标定

本文以时域积分方程方法为研究背景，主要对求解时域积分方程的时间步进算法以及两层平面波快速算法进行了研究。

4.2 基于yolo的视觉、激光结合的三维障碍物检测

时域积分方程方法的研究近几年发展迅速，在本文研究工作的基础上，仍有以下方向值得进一步研究：

4.3 本章小结

第5章 实验验证与结果分析

5.1 全文总结

本文以时域积分方程方法为研究背景，主要对求解时域积分方程的时间步进算法以及两层平面波快速算法进行了研究。

.....

5.2 后续工作展望

时域积分方程方法的研究近几年发展迅速，在本文研究工作的基础上，仍有以下方向值得进一步研究：

.....

致 谢

在攻读博士学位期间，首先衷心感谢我的导师 XXX 教授，……

.....

The Name of the Game

1.1 xxx

1.1.1 xxx

1.1.1.1 xxxx

1.2 xxx

1.2.1 xxx

1.2.1.1 xxxx

English words like ‘technology’ stem from a Greek root beginning with the letters $\tau\epsilon\chi\dots$; and this same Greek word means *art* as well as technology. Hence the name $\text{T}_{\text{E}}\text{X}$, which is an uppercase form of $\tau\epsilon\chi.\text{TeX}$ (actually $\text{T}_{\text{E}}\text{X}$), meaning of $\tau\epsilon\chi$

Insiders pronounce the χ of $\text{T}_{\text{E}}\text{X}$ as a Greek chi, not as an ‘x’, so that $\text{T}_{\text{E}}\text{X}$ rhymes with the word bleechhh. It’s the ‘ch’ sound in Scottish words like *loch* or German words like *ach*; it’s a Spanish ‘j’ and a Russian ‘kh’. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that $\text{T}_{\text{E}}\text{X}$ is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document—something acceptable and basically readable but not really beautiful—a simpler system will usually suffice. With $\text{T}_{\text{E}}\text{X}$ the goal is to produce the *finest* quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you’ll be able to take special pride in the finished product.

On the other hand, it’s important to notice another thing about $\text{T}_{\text{E}}\text{X}$ ’s name: The ‘E’ is out of kilter. This logo displaced ‘E’ is a reminder that $\text{T}_{\text{E}}\text{X}$ is about typesetting, and it distinguishes $\text{T}_{\text{E}}\text{X}$ from other system names. In fact, TEX (pronounced *tecks*) is the admirable *Text EXecutive* processor developed by Honeywell Information Systems.

Since these two system names are Bemer, Robert, see TEX, ASCII pronounced quite differently, they should also be spelled differently. The correct way to refer to T_EX in a computer file, or when using some other medium that doesn't allow lowering of the 'E', is to type '—TeX—'. Then there will be no confusion with similar names, and people will be primed to pronounce everything properly.

此名有诗意

1.1 xxx

1.1.1 xxx

1.1.1.1 xxxx

1.2 xxx

1.2.1 xxx

1.2.1.1 xxxx

英语单词“technology”来源于以字母 $\tau\epsilon\chi\dots$ 开头的希腊词根；并且这个希腊单词除了 technology 的意思外也有 art 的意思。因此，名称 **T_EX** 是 $\tau\epsilon\chi$ 的大写格式。

在发音时，T_EX 的 χ 的发音与希腊的 chi 一样，而不是 “x”，所以 T_EX 与 blecchhh 押韵。“ch” 听起来象苏格兰单词中的 loch 或者德语单词中的 ach；它在西班牙语中是 “j”，在俄语中是 “kh”。当你对着计算机正确读出时，终端屏幕上可能有点雾。

这个发音练习是提醒你，T_EX 主要处理的是高质量的专业书稿：它的重点在艺术和专业方面，就象希腊单词的含义一样。如果你仅仅想得到一个过得去——可读下去但不那么漂亮——的文书，那么简单的系统一般就够用了。使用 T_EX 的目的是得到最好的质量；这就要在细节上花功夫，但是你不会认为它难到哪里去，并且你会为所完成的作品感到特别骄傲。

另一方面重要的是要注意到与 T_EX 名称有关的另一件事：“E” 是错位的。这个偏移 “E”的标识提醒人们，T_EX 与排版有关，并且把 T_EX 从其它系统的名称区别开来。实际上，TEX(读音为 tecks)是 Honeywell Information Systems 的极好的 Text EXecutive 处理器。因为这两个系统的名称读音差别很大，所以它们的拼写也不同。在计算机中表明 T_EX 文件的正确方法，或者当所用的方式无法降低 “E” 时，就要写作 “TeX”。这样，就与类似的名称不会产生混淆，并且为人们可以正确发音提供了条件。