华南理工大学硕士学位论文

LaTeX 模板使用说明

作者姓名

指导教师：xxx 教授

华南理工大学

2025 年 1 月 6 日

目　录

插图目录

表格目录

第一章　相关工作

本文主要研究语言视觉激光多模态融合的机器人导航方法的研究及实现，研究方向是多模态融合的目标物体导航方法。本章则主要介绍这个研究内容所设计到的技术，其中包括 ROS 系统及传感器间的通讯方法、联合标定、目标检测网络、多模态特征融合网络、点云聚类算法。

1.1　ROS 系统的通信与导航

ROS 的全称是机器人操作系统 (Robot Operating System)，它最初由斯坦福大学人工智能实验室 (SAIL) 于 2007 年开发，旨在为机器人开发提供一个灵活、模块化的通用框架，帮助工程师、研究人员和教育工作者等开发人员快速构建复杂的机器人系统。2013 年，ROS 由开源机器人基金会 (Open Source Robotics Foundation, OSRF) 维护和推广。经过多年的发展，ROS 已经成为机器人领域最流行的开发框架之一。

ROS 系统由节点管理者 (ROS Master)、发布者 (Publisher)、订阅者 (Subscriber) 和话题 (Topic) 共同组成如图1-1所示。其中，节点是 ROS 中的基本计算单元，负责执行如传感器数据采集、运动控制或算法处理等特定的任务，而一个完整的机器人系统通常由多个节点组成。话题是 ROS 中实现数据交换的核心机制。节点可以通过节点管理者发布消息到话题，或者通过订阅话题来接受消息，当发布节点和订阅节点都已经注册完成且话题名称一致时，消息就会从发布节点传输给订阅者，实现进程间的通讯。

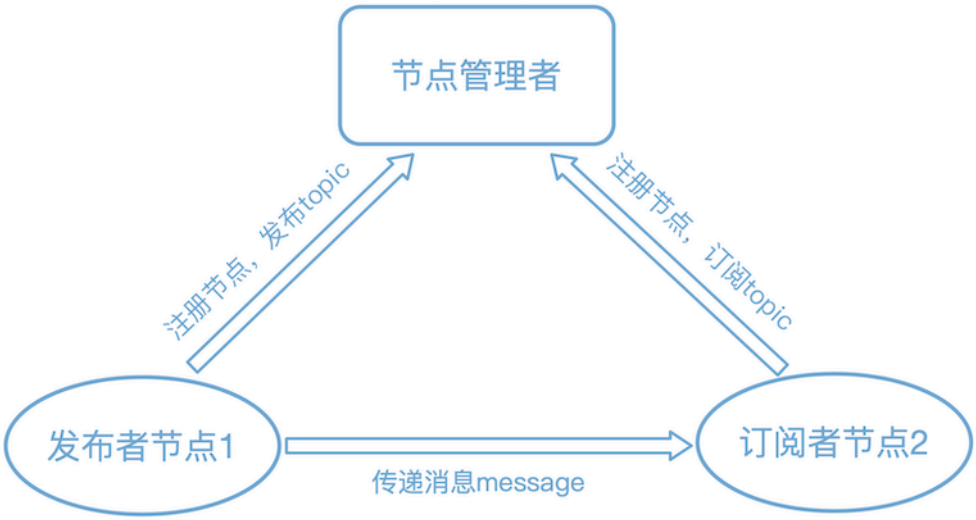

图 1-1 ROS 系统组件图

在用户侧，ROS 提供了一套机器人导航框架如图1-2所示。该框架接收其它组件提

供的地图、传感器信息、定位坐标关联、里程计信息，根据用户指定的导航目标点发布机器人执行的速度指令以运动到目标点。具体来说，ROS 使用 SLAM 算法，通过获取传感器数据 (如激光雷达或摄像头数据) 和机器人运动信息来实时构建环境地图，然后，在创建的二维栅格地图的基础上，通过自适应蒙特卡洛 (Adaptive Monte Carlo Localization, AMCL) 等定位算法和传感器数据来实时估计机器人的位姿，最后，提供了 move_base 作为导航框架主体，由局部、全局代价地图、全局规划期、局部规划期、失败恢复行为等组件来执行导航任务。

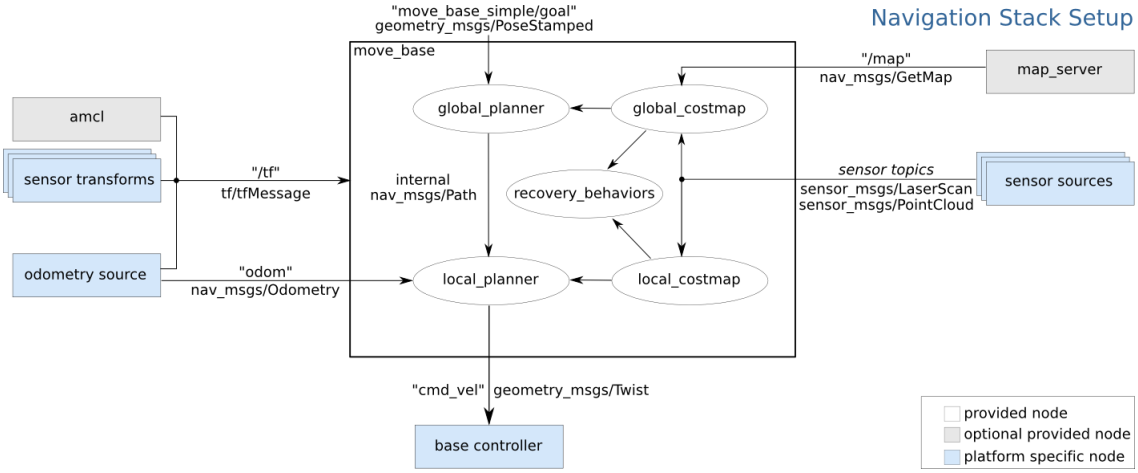

图 1-2 ROS 导航框架图

除了上述的核心功能以外，ROS 还提供了丰富的工具和生态系统，如可以实时显示机器人传感器数据、地图、路径等信息的可视化工具 RViz、允许开发者在虚拟环境中测试机器人算法的高保真物理仿真环境 Gazebo、用于记录机器人运行时的传感器数据工具 ROS Bag，来帮助开发者更高效地开发和调试机器人系统。本文采用 ROS 系统的通信机制、导航方法和仿真环境来实现目标物体导航。

1.2 联合标定

激光雷达和单目相机在移动机器人上有着各自的坐标系，为了让激光雷达感知到的物体和单目相机认知到的物体能够对应起来，需要对它们进行联合标定工作。这里使用张正友标定法[1] 进行联合标定，具体的过程如下图1-3所示。图中存在三个坐标系，分别是以标定板左下角为原点的世界坐标系 $\text{Point}_W = [X_W, Y_W, Z_W]^T$，以单目相机为原点的相机坐标系 $\text{Point}_C = [X_C, Y_C, Z_C]^T$ 和以激光雷达为原点的激光雷达坐标系 $\text{Point}_L = [X_L, Y_L, Z_L]^T$。

联合标定的目的就是要找到相机坐标系和激光雷达坐标系之间的旋转矩阵和平移

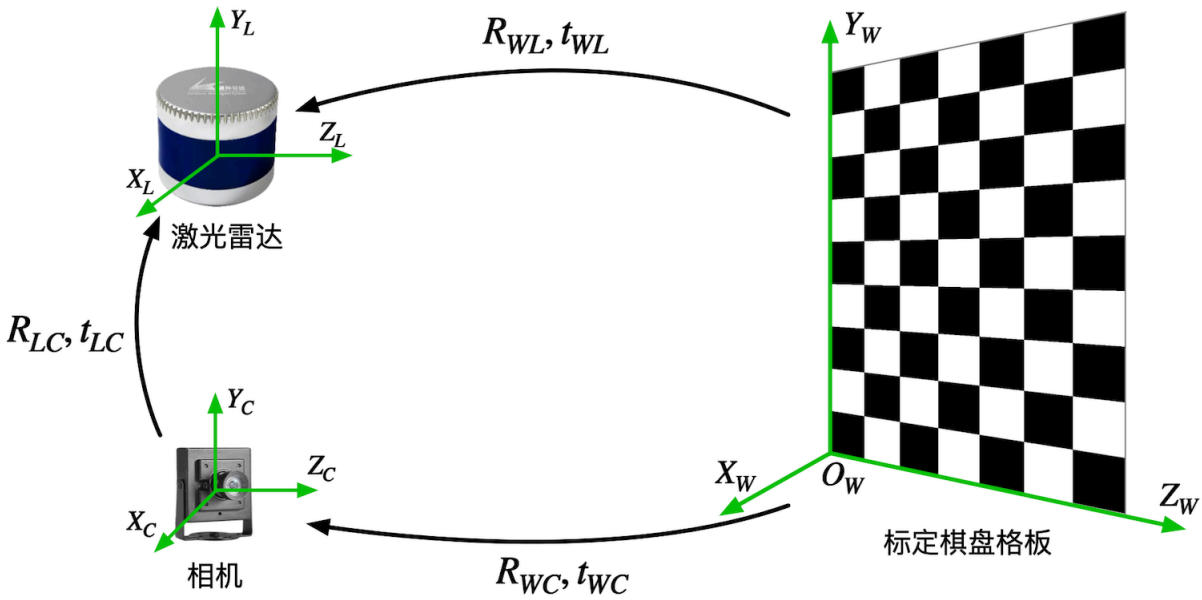

图 1-3 激光雷达和单目相机联合标定

矩阵，即图中的 R_{LC} 和 t_{LC}，使得激光雷达的点云能够映射到图像之中，将雷达感知到的物体与相机认知识别到的物体对应起来。为此需要借助棋盘格标定板来求解它们直接的变换关系矩阵。

首先，相机坐标系中的三维坐标点 Point_C 与像素平面上的二维坐标点 P_{uv} 之间的转换关系可以表示为：

$$ZP_{uv} = Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = K \cdot \mathrm{point}_C \tag{1-1}$$

然后，在激光雷达坐标系下空间的任意一点都可以通过旋转矩阵 R_{LC} 和平移矩阵 t_{LC} 在另一坐标系中进行表示, 它们之间的转换关系可以表示为：

$$\mathrm{Point}_C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \mathrm{R}_{LC} \cdot \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} + \mathrm{t}_{LC} = \mathrm{R}_{LC} \cdot \mathrm{Point}_L + \mathrm{t}_{LC} \tag{1-2}$$

将上式左项移动到右侧，变形可得：

$$\delta\left(R_{LC}, t_{LC}, \mathrm{point}_C, \mathrm{point}_L\right) = R_{LC} \cdot \mathrm{point}_C + t_{LC} - \mathrm{point}_L \tag{1-3}$$

因此，借助棋盘格标定板找到多组相机坐标系和激光雷达坐标系下对应的点，使

得上式中的 $\delta\left(R_{LC}, t_{LC}, \mathrm{point}_C, \mathrm{point}_L\right)$ 取得最小，即可得到旋转矩阵 R_{LC} 和平移矩阵 t_{LC}, 即：

$$\left(R_{LC}, t_{LC}\right)=\underset{R_{LC}, t_{LC}}{\arg\min} \frac{1}{2} \sum\left\|\delta\left(R_{LC}, t_{LC}, \mathrm{point}_C, \mathrm{point}_L\right)\right\|^2. \tag{1-4}$$

上述非线性最小二乘方程可以使用 Levenberg-Marquadt 算法[2] 将其化成线性方程进行求解，且具有较快的收敛速度。获得的旋转平移矩阵将用于后续第四章的局部路径部分。

1.3 目标检测网络

在目标物体导航任务中，部署模型的智能体通过目标检测网络对视觉观测进行编码，以提取周围环境的局部特征，即识别环境中的物体语义信息及其空间位置。这些关键信息用于精准定位感兴趣的目标，并辅助智能体确定导航方向、规划通向目标的最优路径，同时在行进过程中有效规避障碍物。

目标检测是计算机视觉领域的一项基础性任务，其目标是从图像或视频中定位并识别出感兴趣的目标物体。与图像分类任务不同，目标检测不仅需要识别出图像中存在的物体类别，还需要精确地用锚框 (Anchor) 的形式定位出每个物体的位置信息及其对应的置信度分数。在深度学习兴起之前，目标检测主要依赖于手工设计的特征提取方法和机器学习分类器[3]。但这类传统方法存在特征设计复杂、泛化能力有限等问题，难以应对复杂场景下的目标检测任务。

随着深度学习技术的快速发展，特别是卷积神经网络 (Convolutional Neural Network，CNN)[4] 在图像分类的任务中取得的巨大成功，目标检测领域也迎来了革命性的突破。目前，目标检测可以分为两大类别：单阶段 (One-stage) 检测算法和两阶段 (Two-stage) 检测算法。前者将目标检测任务视为一个统一的回归问题，一次性直接在图像上进行目标分类和锚框预测而不需要额外的区域提案生成步骤。因此这种方法通常具有较快的推理速度，适用于实时目标检测的场景，但在跟踪小物体或密集目标时效果较差。经典的 One-stage 目标检测模型包括：

(1) YOLO[5]：早期 YOLO 系列的算法采用网格划分策略，它将输入图像分割成等尺寸的单元，让每个单元同时预测若干预设锚框的空间坐标、类别概率及置信度分数。在生成密集预测结果后，系统再通过非极大值抑制技术筛选最优边界框以去除冗余检测。这类方法的显著特点是检测效率高。得益于其单阶段检测

架构，它无需区域建议等预处理步骤即可完成目标识别。然而，YOLO 过于依赖非极大值抑制进行后处理以得到正确的检测框，这阻碍了其的端到端部署，并且对推理延迟产生了不利影响。除此之外，YOLO 的各种组件设计缺乏全面深入的检查，导致明显的计算冗余，限制了模型的能力。与两阶段的目标检测算法相比，在面对小尺度目标或背景复杂的图像时 YOLO 的检测精度存在一定局限。

(2) SSD[6]：SSD 使用预训练的卷积神经网络在特征提取网络的不同层次获取多个尺度的特征图，这些特征图分别用于检测不同大小的目标。但这种方法需要人工设置预测边框的初始尺度和长宽比的值而不能直接通过学习获得，这导致调试过程十分依赖经验，并且对小尺寸的目标识别效果仍然较差，存在特征提取不充分的情况。

Two-stage 目标检测算法则将目标检测任务分为两个阶段。首先利用区域提案网络 (RPN) 生成一组候选目标区域，然后对每个候选区域进行分类和边界框回归。Two-stage 算法通常在准确性上表现更好，尤其擅长处理小目标和复杂场景，但相对于 One-stage 算法，可能需要更多的计算资源和时间。代表性的 Two-stage 目标检测模型包括：

(1) RCNN 系列：包括 RCNN[7]、fastRCNN[8]、FasterRCNN[9] 等。这类算法首先通过候选框生成网络 (Region Proposals Network, RPN) 来生成一系列可能包含目标的候选框，然后针对每个候选区域都会通过一个预训练的卷积神经网络中提取特征或者对每个候选区域使用 RoI、citedoukas2007region 池化，将每个候选区域都提取并缩放至固定大小的特征，最后使用全连接层进行目标分类和边界框回归。但这类检测方法的实时性不足。

(2) 基于 Transformer：DETR[10] 是 Transformer 目标检测算法的开篇之作。它通过引入 Transformer 架构将目标检测过程视为一个由图形到集合的预测问题，消除了如锚框生成和非极大值抑制等后处理过程，通过二分匹配和一个转换器、编码器、解码器的结构和端到端的方式来进行目标的预测和类别的区分。但这类方法的训练时间长，模型的收敛速度慢，且在小物体的检测时性能较差。

为了满足整个导航过程中目标检查所需的实时性和可靠性，本文使用的目标检测网络是 YOLOV10 网络[11]，如图1-4所示。YOLOV10 与 YOLOV8[12] 相比在整体的网络结构上基本保持一致，网络分为骨干网络 (Backbone)、颈部 (Neck)、头部 (Head) 三个部分。首先将图片输入骨干网络中提取图像的全局和局部特征，在颈部通过 Upsampling、

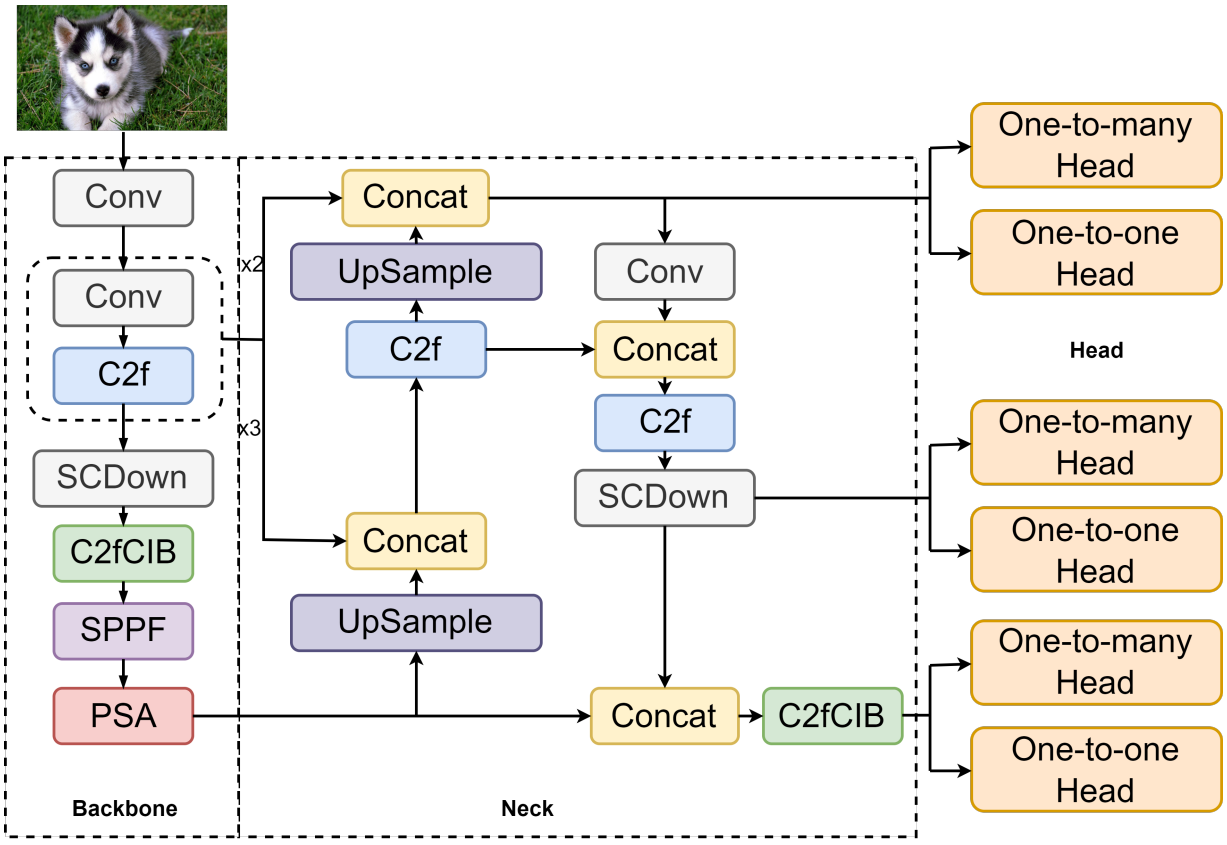

图 1-4 YOLOV10 网络

Concat 和注意力机制卷积网络增强特征的表达能力，实现有效的特征提取和融合，并在头部将颈部提取的特征映射到最终的输出空间，生成网络的最终预测结果。然而，相较于后者，YOLOV10 为了实现更加轻量化的端到端部署而做出了几点重要的优化：

(1) 轻量化分类头 (Lightweight Classification Head)：在 YOLOV8 网络结构中 Head 部分的分类头的参数量和计算量比回归头更大，但后者对检测精度的影响更大，因此减少分类头的卷积参数量以达到轻量化模型的目的。

(2) 空间-通道分离下采样 (Spatial-Channel decoupled downsampling, SCDown)：YOLOV8 使用一个标准卷积时实现空间下采样和通道变换，SCDown 将这两种操作进行解耦，先通过逐点卷积调节通道维度，然后通过深度卷积进行空间下采样，保证降低计算成本的同时最大限度保留信息。

(3) 精度驱动的模型设计：在小模型规模的深层阶段使用大核卷积 (Large-kernel Conv) 来扩大感受野，增强模型能力；针对计算开销过大的自注意力机制设计了一种高效的部分自注意力 (Partial self-attention, PSA)，对分辨率最低的特征的一半进行计算，将对于全局的学习能力以较小的计算成本融入到网络中。通过这些方法可以在不显著增加计算成本的情况下提升模型的性能。

(4) 基于秩的块设计：提出了一种紧凑的倒置块 (CIB) 结构，它采用廉价的深度卷积进行空间混合，通过成本效益高的点卷积进行通道混合，解决了简单地为所有阶段应用相同的块设计导致计算冗余的问题。

1.4 多模态特征融合网络

多模态特征融合是指整合多种来源或形式的数据信息，通过协同作用提升系统的认知能力或执行效果。这种技术旨在充分发挥各模态间的优势互补特性，从而优化系统在诸如模式辨识、类别划分以及内容生成等任务中的综合表现。

在后续的目标物体导航过程中，将利用多模态特征提取网络来融合词嵌入网络的目标特征、目标检测网络编码的局部特征和卷积神经网络编码的全局特征，以得到可以指导机器人进行导航的动作决策。现阶段不同的多模态融合方法按照融合的阶段可以分为以下三种类型：

(1) 特征级融合：这种融合方法是在神经网络的核心处理模块之前，通常是在数据输入环节，将多种模态的特征进行整合。比如，在数据输入阶段就将视觉信息和文本信息进行融合。

(2) 模型级融合：这种融合方式选择在神经网络的中间层级进行多模态信息的整合。具体做法可以是将各模态经过独立学习后的特征表示进行合并，然后再进行后续的网络处理。

(3) 决策级融合：该策略在完成各模态的独立处理后进行融合，整合过程发生在决策或输出层。在每个模态的数据都经过单独处理后，最终将各模态的输出结果进行综合以形成最终决策。这种方法的优势在于其灵活性，能够兼容经过预训练的单模态模型。

本文将采用第三种融合方法，即使用 Transformer 网络进行后期的决策级融合，Transformer 网络的操作流程如图1-5所示。Transformer 网络的核心创新在于采用了多头自注意力机制[13]。自注意力的核心原理是通过计算序列中各元素间的关联度，生成相应的注意力权重，并基于这些权重对序列元素进行加权整合。多头机制则通过并行使用多个注意力单元，使每个单元能够学习到不同的权重分布，从而让模型能够在多个特征子空间中对输入序列进行多样化表征。这种设计使 Transformer 能够突破传统模型的局部窗口限制，实现对序列全局信息的有效捕捉。多个注意力单元可以分别聚焦于不同类

型的信息特征，显著增强了网络在处理多模态数据时的特征融合能力。具体来说：

$$\begin{aligned} Attention\left(Q,K,V\right) &= softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \\ head_i &= Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \\ MultiHead\left(Q,K,V\right) &= Concat\left(head_1,\ldots,head_n\right)W^O \end{aligned} \tag{1-5}$$

其中，查询 (Q)、键 (K) 和值 (V) 由输入序列通过三个线性变换获得的矩阵，W_i^Q、W_i^K 和 W_i^V 分别是 Q、K 和 V 的权重矩阵，d_k 表示特征维度，$head_i$ 表示第 i 个注意力头，W^O 表示连接多个注意力头输出的权重矩阵 Attention 表示子注意力机制，MultiHead 是多头自注意力机制。

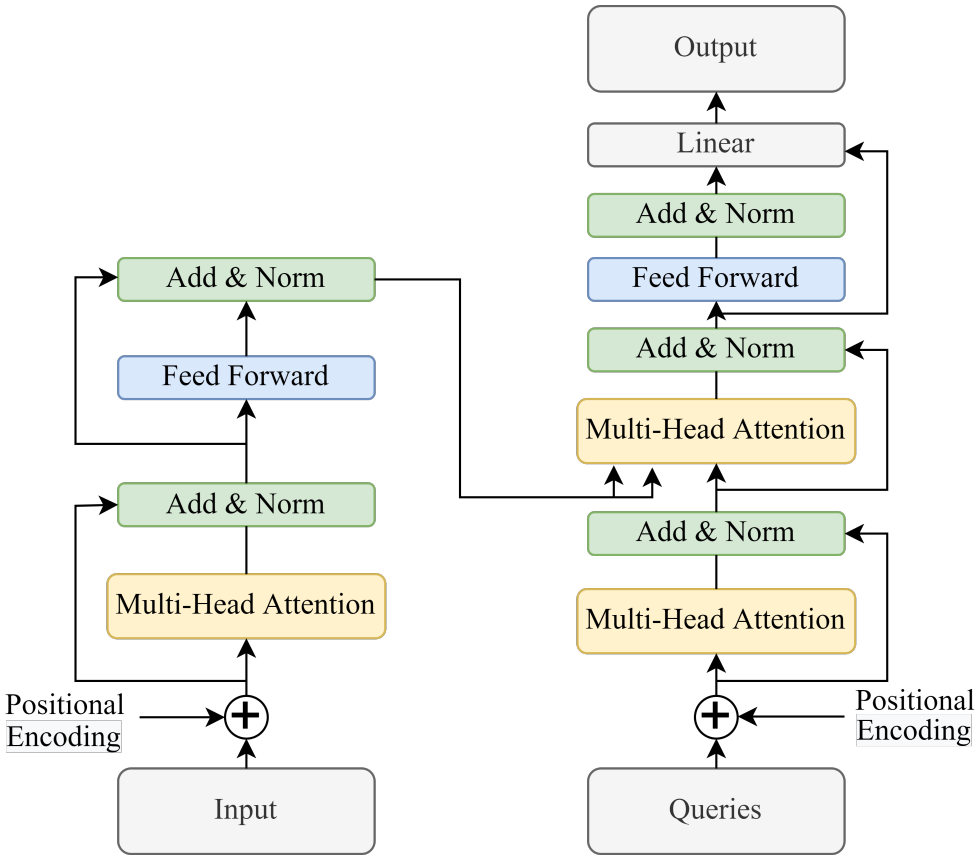

图 1-5 Transformer 网络结构

1.5 点云聚类

三维激光雷达所获取得点云信息在空间中是由一堆离散的点进行表示得，如何区分每个点属于哪个物体是聚类算法需要解决的问题。点云通常利用其特征属性进行聚类，对每个点空间或局部空间进行特征的提取或者转换以得到多种属性，如法向量、密度、距离、强度等，将不同属性的点云分割开来[14]。常见的点云处理方法有欧式聚类、密度

聚类、超体聚类等，针对不同的场景各个算法在耗时和准确率方面各有优势。在目标物体导航的过程中要求系统需要更强的实时性，因此本文采用欧式聚类[15] 方法对三维点云进行预处理。

欧式聚类法在点云密集的情况下需要进行额外的优化以保证其实时性，这里采用点云栅格化和 kd 树对算法进行加速优化[16]。栅格化方法首先将扫描区域划分为若干网格单元，将三维点云投影至二维平面，保持 z 轴数值不变的同时，将每个网格内点的 x、y 坐标统一为该网格中心点坐标。随后进行去重处理，对于 x、y 坐标相同且 z 值相近的点只保留一个代表性点，通过栅格化方法对点云数据进行处理能够显著减少计算复杂度。除此之外，本文利用 PCL 库中的 KdTree->setInputCloud() 函数将栅格化后的点云构建为 k 维二叉树结构，借助 kd 树结构优化近邻搜索过程，进一步提升欧式聚类算法的效率。

距离阈值是用来区分不同簇点云的重要参数，当一个点集 $P_m = \{p_m \in P\}$ 与另一个点集 $P_n = \{p_n \in P\}$ 之间的最小距离大于给定的距离阈值 d_{th}，则可以认为这两个点集为两簇不同的点集。因此，点云簇为两簇不同的点云的条件可以表示为

$$\min \|p_m - p_n\| \geq d_{th} \tag{1-6}$$

激光雷达所产生的点云具有发散的特点，即随着与原点的距离的增加，点云的密度会逐渐下降。如图1-6所示，随着车辆与激光雷达之间的距离增大，其反射生成的点云分布密度显著降低。传统的欧式聚类算法依赖于固定的距离阈值来划分障碍物，这种方法难以应对点云密度随距离变化的情况。针对这种发散的特性，文本根据距离激光原点的远近程度设计了一种改进的欧式聚类方法，如(1-7)所示，当点云离原点距离越远时所采用的距离阈值参数也越大。

$$d_{th} = \begin{cases} 5cm & 0 < Range \leq 1.5m \\ 10cm & 1.5m < Range \leq 3.0m \\ 15cm & 3.0 < Range \leq 5m \\ 20cm & 5m < Range \end{cases} \tag{1-7}$$

文本所采用的是 16 线激光雷达，在离目标距离超过 5 米时竖直方向上的点过于稀疏，导致目标物体检测的效果不理想，因此超过 5 米范围的距离阈值则不再进行更新。本文改进的欧式聚类算法的伪代码见 Algorithm1。

点云聚类算法的结果用于后文第四章所介绍的点云映射中，将点云感知信息与图像认知信息的结果相融合，以获得目标检测算法认知到的目标物体的精确位置信息。

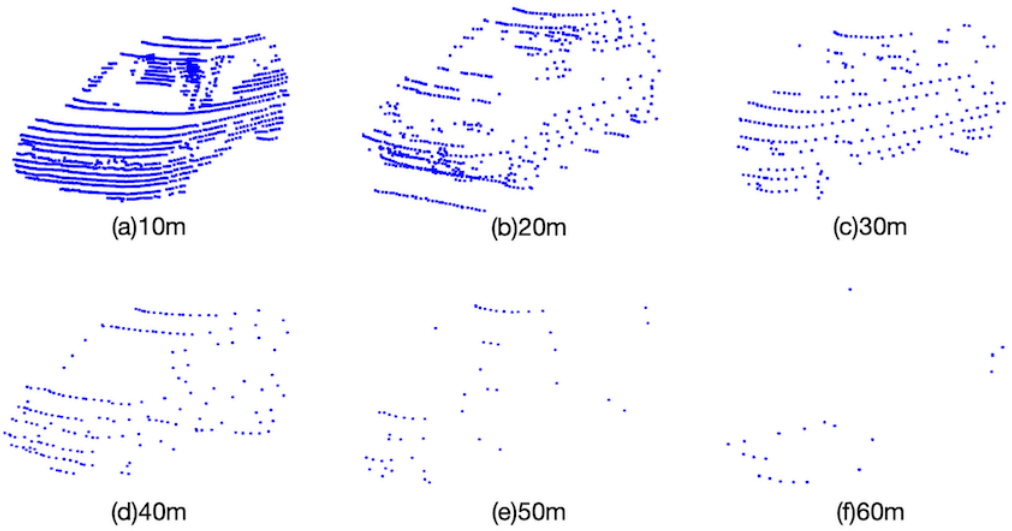

图 1-6 障碍物点云随着距离增加变稀疏

Algorithm 1 可变距离阈值的欧式聚类算法

Input: 激光点云 P

Output: 点云簇集合 C

Grid Downsample and create KD-tree for P; ▷ 对点云进行栅格化和 kd 树预处理
create cluster list C
create cluster c
for $p_i \in P$ **do**
  que.push(p_i) ▷ 遍历点云 P 中的每个点，并将当前点 p_i 加入队列
  while !que.empty() **do**
    ThresholdGet(p_i)$\rightarrow d_{th}$ ▷ 计算当前点对应的距离阈值
    KdtreeSearch(P,p_i,d_{th})$\rightarrow P_i^k$ ▷ 使用 kd 树寻找 p_i 的邻近点
    for $p_j \in P_i^k$ **do**
      que.push(p_j) ▷ 遍历 p_i 的临近点，并将其加入队列中
    end for
    $c = c \cup p_i$ ▷ 将找到的这簇点云加入结果集中
    que.pop()
  end while$c = \emptyset$
end for

1.6 本章小结

本章主要内容是目标物体导航过程中涉及到的一些常用的算法原理和关键技术，主要介绍了 ROS 系统及传感器间的通讯方法、联合标定、目标检测网络、多模态特征融合网络、点云聚类算法。

参考文献

[1] Zhang Z. A flexible new technique for camera calibration[J]. IEEE Transactions on pattern analysis and machine intelligence, 2002, 22(11): 1330-1334.

[2] Moré J J. The Levenberg-Marquardt algorithm: implementation and theory in numerical analysis[J]. Lecture notes in mathematics, 1977, 630: 105-116.

[3] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001: vol. 1. 2001: I-I.

[4] Taye M M. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions[J]. Computation, 2023, 11(3): 52.

[5] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[6] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. 2016: 21-37.

[7] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[8] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[9] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149.

[10] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]//European conference on computer vision. 2020: 213-229.

[11] Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection[J]. Advances in Neural Information Processing Systems, 2025, 37: 107984-108011.

[12] Varghese R, Sambath M. Yolov8: A novel object detection algorithm with enhanced performance and robustness[C]//2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). 2024: 1-6.

[13] Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning[J]. Neurocomputing, 2021, 452: 48-62.

[14] Himmelsbach M, Hundelshausen F V, Wuensche H J. Fast segmentation of 3D point clouds for ground vehicles[C]//2010 IEEE Intelligent Vehicles Symposium. 2010: 560-565.

[15] Liu H, Song R, Zhang X, et al. Point cloud segmentation based on Euclidean clustering and multi-plane extraction in rugged field[J]. Measurement Science and Technology, 2021, 32(9): 095106.

[16] Guo Z, Liu H, Shi H, et al. KD-tree-based euclidean clustering for tomographic SAR point cloud extraction and segmentation[J]. IEEE Geoscience and Remote Sensing Letters, 2023, 20: 1-5.