基于深度学习的语义 SLAM 研究与实现

摘要

视觉 SLAM 技术是移动机器人、自动驾驶、AR\VR 应用中的关键技术。现有的比较成熟的 SLAM 大多使用传统的特征提取和匹配的方法，该方法依赖于特征提取和匹配速度，且大多是稀疏地图，无法融入场景语义，不利于机器人与场景的交互。近来深度学习在计算机视觉领域的应用取得巨大成就，如何有效地将深度学习应用在 SLAM 技术中受到广大计算机视觉研究者的广泛关注。本文使用深度学习方法实现了一个深度估计网络和位姿估计网络联合训练的无监督框架，使用 KITTI 数据集的里程计子集进行训练，并融合了图像语义实现了一个语义 SLAM 系统。

关键词：视觉 SLAM，特征提取和匹配，计算机视觉，联合训练，无监督

Research and Implement of Semantic SLAM Based on Deep Learning

ABSTRACT

Visual SLAM is a key technology in mobile robots, autonomous driving, and AR\VR applications. Most of the existing SLAM use traditional feature extraction and matching methods. This method is limited by the speed of feature extraction and matching, and most of them build sparse maps. Recently, great achievements have been made in the application of deep learning-based methods in the field of computer vision. How to effectively apply deep learning to SLAM technology has attracted widespread attention from computer vision researchers. This article uses deep learning to implement an unsupervised framework for joint training of deep estimation networks and pose estimation networks, uses odometer split of the KITTI dataset for training, and combines image semantics to implement a semantic SLAM system.

Key words: visual SLAM, feature extraction and matching, computer vision, joint training, unsupervised

目 录

1 课题背景.....1
1.1 问题起源.....1
1.2 SLAM 应用领域.....1
1.3 视觉 SLAM.....1
1.4 SLAM 的未来.....2
2 相关知识.....3
2.1 SLAM 问题描述.....3
2.2 刚体运动和相机模型.....3
2.3 地图的形式.....5
2.3.1 地图需求和相应形式.....5
2.3.2 点云地图.....5
2.3.3 八叉树地图.....5
2.3.4 其他地图.....6
3 相关工作.....7
3.1 图像语义分割.....7
3.2 视觉 SLAM.....7
3.3 语义 SLAM 和深度学习.....8
4 技术路线.....9
4.1 视图合成监督信号.....9
4.2 单视图深度估计网络.....10
4.3 位姿估计网络.....10
4.4 语义分割网络.....12
4.5 语义与深度融合方法.....12
4.6 数据预处理和数据增强.....13
5 实验.....14
5.1 训练细节.....14
5.2 深度估计网络.....14
5.3 位姿估计网络.....15
5.4 语义分割网络.....17
6 结论和展望.....19
6.1 结论.....19
6.2 不足.....19
6.2.1 尺度不确定性.....19
6.2.2 不能有效利用计算资源.....19
6.2.3 无法进行回环检测和重定位.....19
6.3 进一步研究计划.....19
6.3.1 解决尺度不确定性问题.....19
6.3.2 尝试共用特征提取器.....19
6.3.3 回环检测和重定位.....19
6.3.4 尝试使用生成对抗网络方法.....20
附录.....25
附录 1 深度指标计算公式.....25
附录 2 ATE 计算公式.....25
谢辞.....26

1 课题背景

1.1 问题起源

SLAM 是“Simultaneous Localization And Mapping”的缩写，可译为同步定位与建图。SLAM 问题最早由 Durrant Whyte 提出在 1986 年的 IEEE Robotics and Automation Conference 大会上提出,希望能将估计理论方法(estimation-theoretic methods)应用在构图和定位问题中。SLAM 最早被应用在机器人领域，其目标是在没有先验知识的情况下，根据传感器数据实时构建周围环境模型，同时根据这个环境模型推测自身的位置[1]。

1.2 SLAM 应用领域

在机器人领域，智能服务机器人逐渐走上了行业的风口浪尖。我们的生活中也出现了越来越多移动机器人的身影。随着传感器技术、人工智能技术和计算技术等的不断提高，智能移动机器人逐渐的融入了人们的学习、生活和工作。无论是服务型移动机器人还是工厂里流水线上的移动装配机器人、物流行业的自动分拣机器人，都离不开 SLAM 实时定位与地图构建这一环节。如何让机器人清晰、高效的知道自己在哪里、周围有什么就成为了制约移动机器人工作能力和运行效率的一个重要条件。

目前，SLAM 技术已经普遍应用于机器人定位导航、VR/AR、无人驾驶等领域，SLAM 可以辅助机器人执行路径规划、自主探索、导航等任务。如无人驾驶技术中车辆需要通过 SLAM 观测周围环境、确定自己位置并做出相应的驾驶行为。在 VR/AR 领域，SLAM 技术用于改善视觉效果，构建视觉效果更为真实的地图。在服务机器人的应用中，机器人需要借助 SLAM 技术构建高质量的语义地图，并在精确的定位和语义地图的帮助下完成一系列的任务。在无人机领域，SLAM 可以快速构建局部 3D 地图，并与地理信息系统（GIS）相结合，可以辅助无人机识别路障并自动避障规划路径。

1.3 视觉 SLAM

SLAM 中常用的传感器有激光雷达、声纳、深度相机（RGB-D）和惯性传感器（IMU）等，其中仅使用相机作为传感器的 SLAM 系统叫做视觉 SLAM。视觉 SLAM 常用的技术包括图像特征的检测、描述、匹配，图像的识别、检索等。

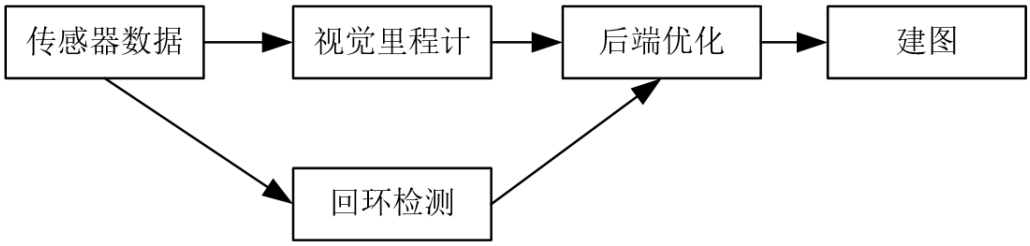

图 1.1 经典视觉 SLAM 流程图

如图 1 所示，经典的视觉 SLAM 系统一般包含视觉里程计、后端优化、闭环检测、建图等几个主要部分[2]。视觉里程计（Visual Odometry）模块利用当前输入数据和历史输入数据估

计机器人运动状态；回环检测（Loop Closing）模块通过机器人运动过程中传感器的输入信息判断机器人是否发生轨迹闭环，即判断自身是否进入历史同一地点；后端优化模块接收视觉里程计估计的相机位姿以及回环检测模块的输入，通过优化技术使轨迹和环境模型具有一致性；建图模块根据估计的轨迹和路标位置建立环境模型；除上述模块外，在部分应用中还有丢失重定位的需求，在系统发生奔溃后要求机器人根据历史地图迅速定位并开始导航。

1.4 SLAM 的未来

虽然当前已经出现很多比较成熟的使用传统方法的 SLAM，但是在具体应用场景中往往很难满足性能需求。主要有两个方面的原因：应用场景的多样性和复杂性、计算能力与性能需求的不匹配。

在实际应用过程中往往会出现各种各样的场景，这些场景中常常包含对 SLAM 算法挑战比较大的区域，比如场景中出现运动物体、低纹理区域、遮挡\解除遮挡和非刚体等，出现这些情况时 SLAM 算法出现误差的可能性比较大，如何设计鲁棒性好、可靠性高的 SLAM 算法非常重要。

在许多需要使用 SLAM 的设备往往对性能有较高的要求但是自身计算资源非常有限，这就需要轻量化、小型化的 SLAM。多传感器融合是 SLAM 小型化的一个方向，实际应用中机器人通常携带多种传感器，如何利用从多个传感器获得的数据提高 SLAM 精度和可靠性、降低对计算资源的需求是一个关注热点。

随着智能机器人的发展，对 SLAM 的要求不再只是定位和建图，还要能够用于智能机器人与环境的交互，这就需要结合语义的 SLAM。语义和 SLAM 结合主要有两个方面：语义帮助 SLAM 和 SLAM 帮助语义[51]。语义帮助 SLAM 一般单独生成物体语义标签，然后将语义标签用于回环检测、优化等任务中提高 SLAM 的性能。SLAM 帮助语义是利用移动机器人从物体各个角度观测物体生成高质量的语义训练数据，这样能够很大程度上加速分类器训练过程。

在深度学习（Deep Learning）普遍应用前只能利用支持向量机、条件随机场等方法进行图像语义分割。近年来由于深度学习的快速发展，尤其是全卷积神经网络 (Fully Convolutional Network，FCN)的出现，实现了从图像像素到像素类别的变换，结合语义的 SLAM 技术得到了快速的发展。深度学习不仅用于生成图像语义，还出现了将深度学习用于基本 SLAM 任务（如深度估计、位姿估计、回环检测和丢失重定位等）的探索。

本选题是希望在这样的背景下，通过分析研究当前各类算法框架的特点，探索将深度学习应用在 SLAM 中的方法，实现一个基本的语义 SLAM。

2 相关知识

2.1 SLAM 问题描述

SLAM 两大任务是估计相机运动状态同时构建环境模型。用 x 表示移动机器人的位置，在某一段时间内的离散时刻 $t=1,...,\mathrm{K}$ 移动机器人经过的位置为 $x_1,...,x_K$，这便是移动机器人的轨迹。假设地图由 N 个路标(Landmark) $y_1,...,\mathrm{y}_N$ 组成，在上述每一个离散时刻，移动机器人自身所携带的传感器（激光雷达、声纳或相机等）从当前所在位置能观察到一部分路标点，构成传感器的观测数据。这样移动机器人在环境中的运动可由以下两个事件描述：

（1） **运动** 考虑从 $k-1$ 时刻到 k 时刻，移动机器人的位置 x 怎样变化。用数学语言描述：

$$x_k = f(x_{k-1}, u_k, \omega_k) \tag{2.1}$$

其中 u_k 表示传感器的输入，一般由移动机器人自身携带的传感器读取获得，ω_k 代表噪声。上述方程叫做**运动方程**。

（2） **观测** 移动机器人在位置 x_k 上看到路标 y_i，产生了一个观测数据 $z_{k,i}$，用抽象函数 h 描述：

$$z_{k,i} = h(y_i, x_k, \mu_{k,i}) \tag{2.2}$$

其中 $\mu_{k,i}$ 表示观测里的噪声。上述方程叫做**观测方程**。

SLAM 任务中估计相机运动状态便是要估计上述运动方程中的 x_k，建图即通过传感器输入 $z_{k,i}$ 提取观察到路标 y_i，并将提取的路标 y_i 保存并用于以后的定位和导航。

2.2 刚体运动和相机模型

SLAM 中常常设定一个惯性坐标系（也叫世界坐标系），使用物体在惯性坐标系中的坐标描述该物体在地图中的位置。如图 2.1，将坐标系 $x_w y_w z_w$ 作为惯性坐标系，相机则是一个移动坐标系 $x_c y_c z_c$ 的原点。用 p_c 表示在移动坐标系中某个点 P 的坐标，则该点在移动坐标系中的

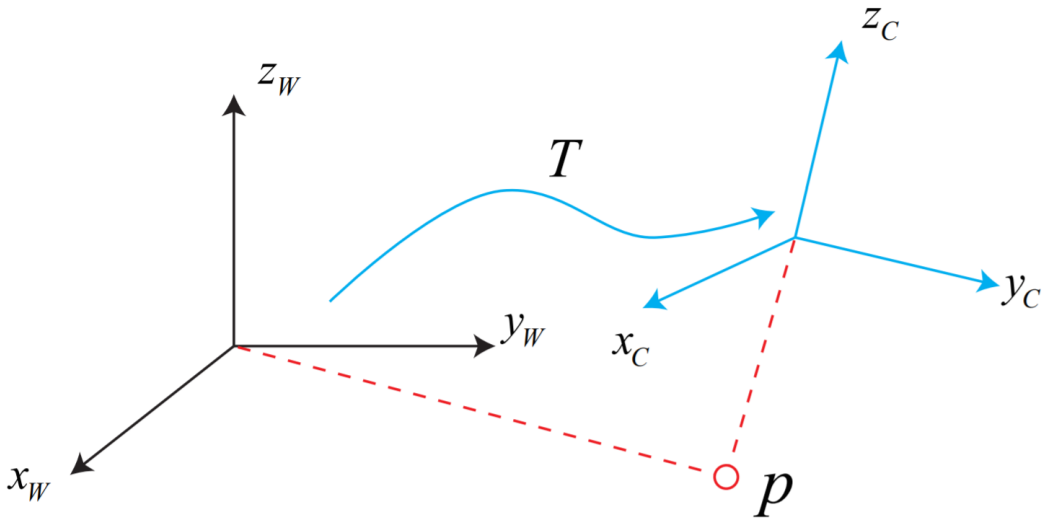

图 2.1 坐标变换示意图

坐标 p_c 可由该点在惯性坐标系中的坐标 p_w、移动坐标系 $x_c y_c z_c$ 相对于惯性坐标系 $x_w y_w z_w$ 的

旋转矩阵 R 和平移向量 t 表示：

$$p_c = Rp_w + t \tag{2.3}$$

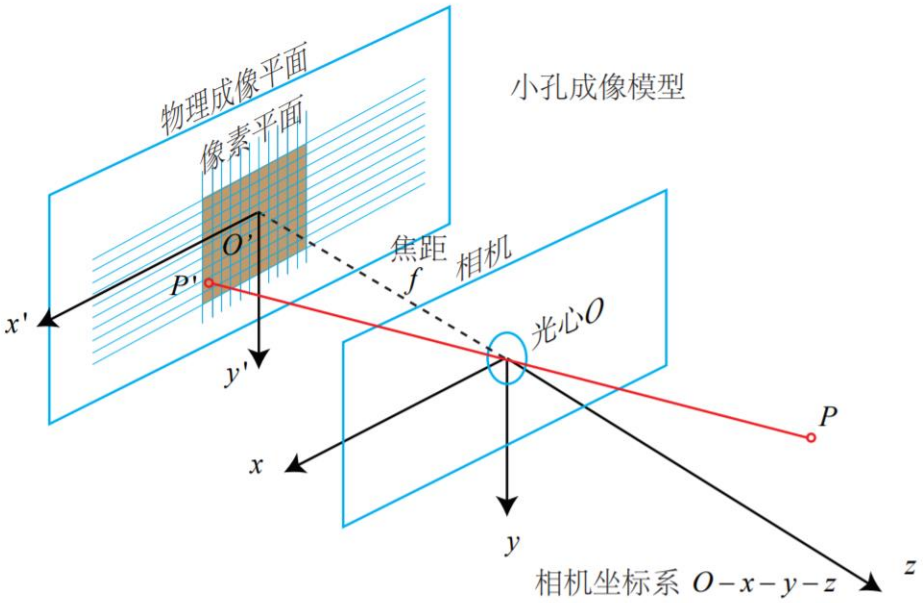

图 2.2 小孔成像模型

图 2.2 是一个简单的小孔成像模型。设 $O-x-y-z$ 为相机坐标系，O 为摄像机光心。现实世界中的点 P 经过小孔 O 投影后落在物理平面 $O'-x'-y'$ 中坐标为 p' 的点 P' 上，设 p_c 和 p' 分别为 $[X_c, Y_c, Z_c]^T$ 和 $[X', Y', Z']^T$，相机成像平面到小孔的距离为 f（焦距），按照相似三角形边长的关系，有：

$$\frac{Z_c}{f} = \frac{X_c}{X'} = \frac{Y_c}{Y'} \tag{2.4}$$

由于相机坐标原点 o' 通常在图像左上角，且成像平面通常不在焦平面，所以经过了缩放和平移的像素坐标 $[u, v]^T$ 与坐标 P' 之间的关系表示为：

$$u = \alpha X' + c_x, v = \beta Y' + c_y \tag{2.5}$$

将等式（2.4）带入等式（2.5），并将 αf 合并为 f_x、βf 合并为 f_y，得到：

$$u = f_x \frac{X_c}{Z_c} + c_x, v = f_y \frac{Y_c}{Z_c} + c_y \tag{2.6}$$

将上式写成矩阵形式并变换为其次坐标：

$$Z_c \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = \begin{Bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} X_c \\ Y_c \\ Z_c \end{Bmatrix} = Kp_c \tag{2.7}$$

等式（2.7）中的矩阵 K 叫做相机内参（Camera Intrinsics），该参数通常在相机出厂后不再变化。

由于相机在运动，由等式（2.3）可知点 P 在相机坐标系中的坐标 p_c 可由世界坐标与相机运动 R 和 t，将等式（2.3）带入等式（2.7）后可得

$$Z_c \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = K(Rp_w + t) = KTp_w \tag{2.8}$$

等式（2.8）中T是用于描述相机运动的变换矩阵（Transform Matrix），Z是图像深度，$[u,v]^T$是像素坐标，可见在有相机内参的情况下可用图像深度Z和相机运动T估算像素点在世界坐标系$x_w y_w z_w$下的坐标。

相机或机器人的运动除了可以使用上述变换矩阵表示外，还可以旋转分量和平移分量表示。其中旋转有多种表示方法，在所有表示方法中欧拉角是最符合人的直观感受的表示方法。欧拉角使用三个绕坐标轴的旋转分量来共同描述一次旋转。欧拉角最常见的一种就是用“偏航-俯仰-滚转”（yaw-pitch-roll）三个角度来描述一个旋转:

（1） 物体绕Z轴旋转，得到偏航角 yaw；

（2） 物体绕旋转后的Y轴旋转，得到俯仰角 pitch；

（3） 物体绕旋转后的Z轴旋转，得到滚转角 roll。

这样使用向量$[r,p,y]^T$可以描述任何旋转。

旋转还有旋转向量和四元数等表示方法，这些表示方法之间可以互相转换，欧拉角和上述变换矩阵的旋转分量也可以互相转换。

2.3 地图的形式

2.3.1 地图需求和相应形式

目前出现的建图方法有很多种，不同的方式构建的地图表现形式也有所不同。最简单的建图方式就是根据当前相机相对于世界坐标系的位姿和像素深度值，求出像素点在惯性坐标系中的坐标，再在点云（Point Cloud）把该点的坐标和像素值表示出来，这样生成的是一个由离散点组成的点云地图（Point Cloud Map）。除此之外，使用三角网格（Mesh）或者面素（Surfel）可以构建包含物体表面信息的地图，使用体素（Voxel）建立的占据网格地图（Occupancy Map）可以包含当前环境中准确的空间信息，这种地图常用于导航任务。

2.3.2 点云地图

点云地图就是一种使用离散的点组成的地图，最基本的只需要给出离散点的x,y,z三维坐标信息，也带有r,g,b色彩信息。点云地图是一种最基本的表示方法，短板也很明显：点云地图通常会占据非常大的存储空间，因为每一个像素点都与地图中的一个点对应，这样建立的地图含有很多冗余信息；其次点云对于地图中出现的重叠很难处理，而这种由于位姿估计或者深度估计出现误差引起的重叠又非常常见，这是点云地图在应用过程中很难处理的一个问题；最后是点云地图中离散点提供的信息过于分散，很难及时有效地将地图中的信息应用于实际任务中，比如利用点云地图进行导航需要精心地设计算法。点云地图之所以有这些不足是因为点云地图是一种接近传感器读取的原始数据的一种地图形式，它的处理方式也是最简单的。

2.3.3 八叉树地图

针对导航任务可以从点云出发构建占据网格地图（Occupancy Grid），占据网格地图中以某个最小体积为基本单位记录该空间的占据信息。八叉树（Octo map）[55]是一种灵活的、压缩的又能够随时更新的占据地图。如图3.1所示，八叉树地图使用立方体来表示空间是否被占据，每个立方体又能够切分成 8 个更小的立方体，最小的立方体中存储网格占据信息。建图过程中当某个方块的所有子节点都不被占据时该节点就没必要展开，由于实际场景中被物体占据的空间是连续的，不被物体占据的空间也是连续的，通常根据实际环境建立的八叉树地图中许多节点不需要展开，所以八叉树地图比点云地图节省了大量存储空间。

八叉树的每一个叶节点存储该叶节点对应空间被占据的概率$x(x\in[0,1])$，如果连续观察

到某个网格被占据，就增加 x 的值，反之观察到网格不被占据就减小 x。因为 x 是一个概率，不便于处理，实际利用的是 x 的概率对数值（Log-odds）y 来表示网格被占据的概率，x 为 0 到 1 之间的概率，那么他们之间的变换可用 logit 描述为：

$$y = \log it(x) = \log(\frac{x}{1-x}) \tag{2.9}$$

反过来变换为：

$$x = \log it^{-1}(\mathrm{y}) = \frac{\exp(y)}{\exp(\mathrm{y})+1} \tag{2.10}$$

这样当 y 从 $-\infty$ 变化到 $+\infty$ 时，x 相应地从 0 变化到 1。设观察数据中某个节点 n 为占据概率为 z。设从 1 时刻到 t 时刻某节点的概率对数值为 $L(n \mid z_{1:t})$，那么 $t+1$ 时刻为：

$$L(n \mid z_{1:t+1}) = \mathrm{L}(\mathrm{n} \mid \mathrm{z}_{1:t-1}) + L(n \mid z_t) \tag{11}$$

写成概率的形式就是：

$$P(n \mid z_{1:T}) = [1 + \frac{1-P(n \mid z_T)}{P(n \mid z_T)} \frac{1-P(n \mid Z_{1:T-1})}{P(n \mid \mathrm{z}_{1:T-1})} \frac{P(n)}{1-P(n)}]^{-1} \tag{12}$$

使用上述八叉树地图的对数形式，可根据机器人的运动状态对整个地图进行更新。

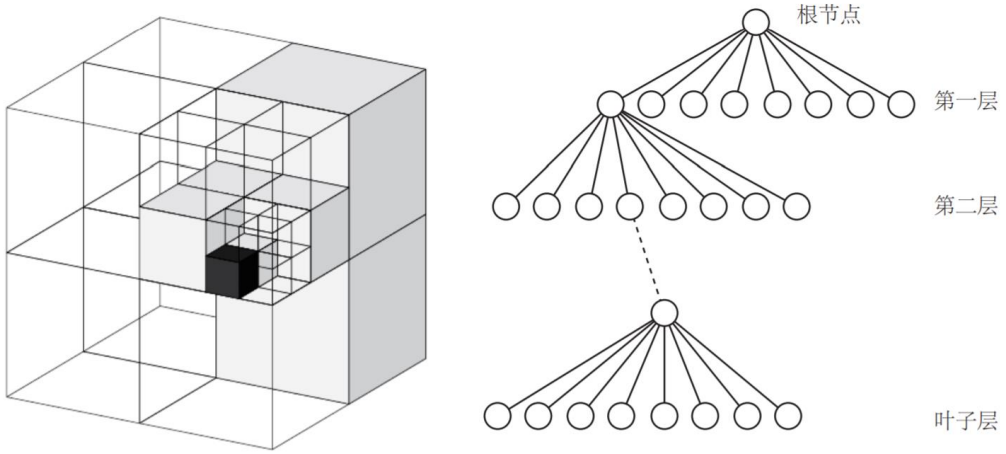

图 2.3 八叉树示意图

2.3.4 其他地图

随着计算式视觉的发展以及 AR\VR 的发展，许多应用场景要求算法能够重建高质量的地图，而定位居于次要地位，这种任务有运动恢复结构（Structure from Motion, SfM）和实时三维重建。SfM 中常用泊松重建能够利用点云地图重建物体网格地图[3]，得到物体的表面信息。除泊松重建外，Surfel 亦能够表达物体表面信息，以面素作为地图的基本单位，能够建立符合人的直观感受的地图[4]。实时三维重建任务把重建高质量地图作为主要目标，在地图质量、实时性方面比 SfM 要求更高，基本都需要使用 GPU 进行加速，甚至使用多个 GPU 并行加速，建图结果也精细得多。这些技术主要有 Fusion[5, 6, 7, 8]系列和 TSDF（Truncated Signed Distance Function）等，这里与本文内容相关性不大，不做过多赘述。

3 相关工作

3.1 图像语义分割

语义分割是一项重要的密集预测任务，其预测的目标是图像中每个像素的已知类的后验分布。目前出现的全部方法可大抵分为传统方法和深度学习方法两大类。

传统图像语义分割根据像素灰度值、色彩、图像纹理、物体几何形状等特征把图像划分成若干个区域，并使用马尔科夫随机场(Markov Random Fields, MRFs)和条件随机场(Conditional Random Fields, CRFs)来构建概率图模型，然后使用概率图的方法来求解[9, 10]。这些方法只利用了有限的图像的浅层视觉特征，由于实际场景往往复杂多变，而这些方法能够提取到的图像特征非常有限，所以传统语义分割方法鲁棒性往往不高，在具体的应用中经常需要针对应用场景进行精细的算法设计。

神经网络模型最早由 Lecun 在 1998 年提出，他设计的 LeNet 网络含有卷积层、池化层和非线性层[1]。Hinton 研究组于 2012 年年提出了 AlexNet，该网络在 LeNet 网络架构的基础上加深了网络深度，使用了 Relu 非线性激活、Dropout 和数据增强等提高模型性能的方法，并使用了 GPU 进行训练，该网络模型在当年的 ImageNet 竞赛上表现远超传统方法[12]。从此深度学习成为计算机视觉领域的主流方法，研究者们开始摸索将深度学习模型引入到计算机视觉的各项任务中。在图像语义分割中采用卷积神经网络方法学习目标特征并训练分类器，对目标区域进行分类，从而实现目标区域的自动语义标注。2015 年，Karen Simonyan 等提出 VGG 神经网络，VGG 网络在第一层使用了感受域更小的卷积层，使得模型的参数更少，非线性更强，也因此使得决策函数更具区分度，模型更好训练[13]。Jonathan Long 等提出的全卷积网络(Fully Convolutional Networks, FCN)在神经网络的全连接层的位置使用卷积层，并使用了反卷积(deconv)和跳级(skip)结构，使得网络在直接输出语义预测结果的同时拥有更好的鲁棒性和精确性[14]。Liang-Chieh Chen 等使用深度神经网络(Deep Convolutional Nets)和全连接条件随机场(Fully Connected CRFs)获得比较精细的分割结果[15]。Kaiming He 等提出的 Mask R-CNN 在 Faster R-CNN 的基础上加入了语义分割结构，并取得非常好的实验效果[16, 17]。

3.2 视觉 SLAM

视觉 SLAM 的发展分为古典阶段（1986-2004）、算法分析阶段（2004-2015）和现阶段三个主要时期[18]。古典阶段的 SLAM 大多使用概率方法，例如扩展卡尔曼滤波(Extended Kalman Filter，EKF)、粒子滤波器（Particle Filter）、极大似然（Maximum Likelihood，ML）和最大期望(Expectation-Maximization，EM)等[19]。当要联合考虑机器人和环境模型不确定性的时候，扩展卡尔曼滤波和最大期望是比较好的选择，但在大场景中的导航和回环检测的能力有限。Guivant 最早提出以增量的方式构建地图，他还给出了随机地图的概念，并使用 EKF 方法提升了 SLAM 算法的精度[20]。由 Davison 教授于 2007 年提出的 MonoSLAM 是第一个实时单目 SLAM 系统，MonoSALM 以相机当前状态和所有路标点为状态量，使用扩展卡尔曼滤波更新环境模型均值和协方差，该方法因只跟踪少量的特征点能够实现在线运行[21]。

算法分析阶段研究 SLAM 的基本属性，例如可观察性、收敛性和一致性等。这一阶段许多非线性优化方法被应用于求解 SLAM 问题，比如 Bundle Adjustment、位姿图（Pose Graph）等。2007 年，Klein 等人在 PTAM（Parallel Tracking and Mapping）中将 SLAM 中的追踪和建图分别使用前端和后端两个进程完成，这是视觉 SLAM 中首次将跟踪和建图分离，PTAM 中

还首次使用非线性优化进行后端优化，并使用关键帧方法提高优化效率[22]。ORB-SLAM 是现在出现的 SLAM 中十分完善、易用的 SLAM 系统，整个 ORB-SLAM 围绕 ORB 特征进行计算，ORB 特征在保持优良的旋转和缩放不变性的同时具备很高的效率，ORB-SLAM 将 SLAM 任务分为实时跟踪特征点（Tracking）、局部优化（Bundle Adjustment）和全局回环检测和优化（Pose Graph），ORB-SLAM 非常卓越的回环检测模块以及良好的代码设计都是 ORB-SLAM 获得普遍认可的基础，另外，它支持单目、双目和 RGB-D 三种模式[23]。J.Engle 等人于 2014 年提出的 LSD-SLAM（Large Scale Direct monocular SLAM）针对像素使用直接法在 CPU 上实现了半稠密重建，然而回环检测模块还是依赖于特征点计算[24]。SVO（Semi-direct Visual Odometry）跟踪关键点并像直接法一样估计相机运动和关键点的位置，由于 SVO 不需要计算特征点，它能够在低端计算平台上实现实时[25]。

受到生物启发而构建的 SLAM 系统也是一类常见的视觉 SLAM 解决方法。Milford 模拟鼠类提出的 RatSLAM 使用单个相机生成复杂环境的模型[26]，RatSLAM 在室内和室外环境中性能表现都很突出[27, 28]。Collett 研究了蚂蚁在戈壁如何通过视觉路标点进行引导，这种视觉路标点引导的方法在机器人系统上是可行的[29]。

与 SLAM 任务相似的另一个研究方向叫做运动恢复结构（Structure from Motion， SfM），运动恢复结构比 SLAM 更注重于精确的环境模型的构建。SfM 起源于图形学和计算机视觉，一般的计算流程是在输入图像上进行特征提取和匹配，并使用非线性优化来最小化重投影误差从而获得环境模型[30, 31]。Pollefeys 在 2004 年提出了从运动恢复结构(Structure from Motion，SfM)技术从图像序列中计算相机位姿和场景的三维结构[32]。

3.3 语义 SLAM 和深度学习

将场景语义引入到 SLAM 中一直是研究人员不懈的追求，已有相关论文将物体语义标签与视觉 SLAM 结合起来构建带语义的地图[33, 34]。把标签信息引入到 BA(Bundle Adjustment)或优化端的目标函数和约束中，可以结合特征点的位置与标签信息进行优化[35]。Flint 等人提出了应用于室内场景的在线 SLAM 模型，该模型利用曼哈顿世界假定进行重要平面的分割[36]。2015 年，斯坦福的 Vineet 等人使用增量的方法进行大场景语义重建，该系统几乎能够实时进行建图和语义分割，证明了语义 SLAM 有可能性应用到实际应用中[37]。Bowman 引入了最大期望(Expectation Maximization, EM)估计来把语义 SLAM 转换成概率问题，其优化目标还是重投影误差[38]。

随着深度学习的发展，研究者们开始将神经网络应用于 SLAM 领域，现在深度学习在 SLAM 中的应用有位姿估计、深度估计、重定位、回环检测以及图像语义生成等方面[39, 40, 41, 42]。Yasin Almalioglu 等使用生成对抗网络实现了位姿估计网络和深度估计网络的联合训练[43]。Keisuke Tateno 等将深度学习方法与传统方法融合，利用神经网络进行图像深度预测和图像语义生成，实现了一个实时单目 SLAM[44]。Michael Strecke 等用 SDF 表示将多目标跟踪表示为 RGB-D 图像的直接对准,并使用概率方法直接进行数据关联和遮挡处理，并使用期望最大化(Expectation Maximization, EM)框架将 Mask R-CNN 的识别和分割结果融合到 SLAM 里，该方法在动态的室内场景取得较好效果[44]。

4 技术路线

本节将详细介绍语义 SLAM 系统中使用的相关技术，主要有使用与监督方法联合训练的单视图深度估计网络（depth CNN）和位姿估计网络（pose estimation CNN）[45]、参考 SegNet[46]实现的语义分割网络。

这里网络结构中输入大小均为128×416的无损 png 格式图像，深度估计网络和位姿估计网络的输出分别对应于多个尺度的预测数据，最终合成语义地图时使用了与输入大小一样的输出结果。

4.1 视图合成监督信号

深度估计网络和位姿估计网络的监督信号来自于视图合成模块。利用相机在某个位姿下观测图像序列作为输入（某一帧作为目标视图I_t，其他作为源视图I_s），根据网络输出的源视图与目标视图的相对位姿$\hat{T}_{t\to s}$以及合成目标视图中各个像素点的深度值$\hat{D}_t$合成目标视图$\hat{I}_s$，视图合成模块可以用几何的方法合成新视图。视图合成用一种可微的方式实现，合成视图和目标视图之间的误差用于训练深度估计网络和位姿估计网络。

用$<I_1,...,I_N>$代表一个训练样例，其中一帧作为目标视图I_t，其他帧作为源视图$I_s(1\le s\le N, s\ne t)$。合成视图与目标视图之间的像素差叫做像素损失，用下面公式表示：

$$\mathcal{L}_{vs}=\sum_{1\le s\le N,s\ne t}\sum_{p}|\mathrm{I}_t(\mathrm{p})-\hat{\mathrm{I}}_s(\mathrm{p})| \qquad (4.1)$$

其中p表示像素坐标，$\hat{I}_s$表示源视图I_s经过视图合成模块投影到目标位姿后合成的视图。视图合成模块使用 CNN 预测的深度图$\hat{D}_t$、4×4 相机旋转矩阵$\hat{T}_{t\to s}$和源视图I_s作为输入。

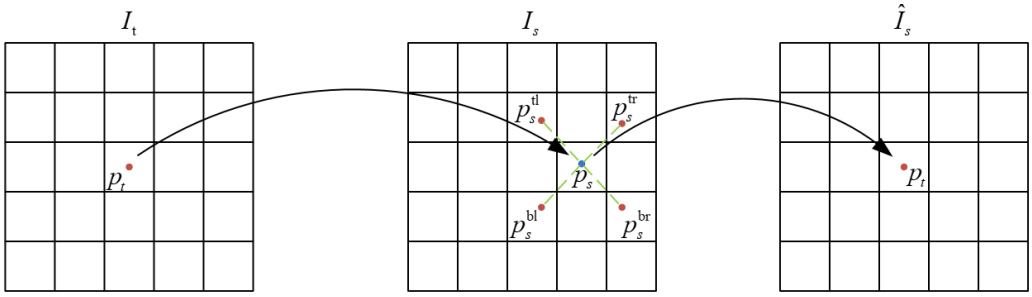

图 4.1 视图合成过程示意图

图 4.1 是视图合成过程示意图。p_t表示目标视图中某个像素点的坐标，要求合成视图$\hat{I}_s$在p_t点的像素值$\hat{I}_s(\mathrm{p}_t)$，先将坐标p_t投影到源视图I_s得到坐标p_s，再用p_t周围的像素值（p_s^{tl}、p_s^{tr}、p_s^{bl}、p_s^{br}分别表示投影点p_s左上、右上、左下、右下像素点的坐标）求双线性插值的结果作为$\hat{I}_s(\mathrm{p}_t)$的值。

坐标投影过程需要使用相机内参K、深度估计网络预测的目标视图深度$\hat{D}_t$和位姿估计网络的输出的相对位姿$\hat{T}_{t\to s}$，坐标投影过程用下面公式表示：

$$p_s\sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t \qquad (4.2)$$

投影坐标p_s是连续值（p_s是坐标值是小数，不是某一个像素的坐标），为了获得源视图在

p_s 处的像素值 $I_s(p_s)$，需要使用双线性采样方法，双线性采样方法用与 p_s 相隔最近的四个像素（左上、右上、左下、右下）的像素值的双线性采样结果作为 $I_s(p_s)$ 的近似值。双线性采样过程用如下公式表示：

$$\hat{I}_s(p_t) = I_s(p_s) = \sum_{i\in\{t,b\}, j\in\{l,r\}} w^{ij} I_s(p_s^{ij}) \qquad (4.3)$$

其中 w^{ij} 与 p_s 和 p_s^{ij} 的空间接近度成线性比例，且 $\sum_{i,j} w^{ij} = 1$。

4.2 单视图深度估计网络

单视图深度估计网络采用 DispNet[47]架构，DispNet 架构含有编码器-解码器单元，在编码器和解码器之间有跳级连接(skip connection)，能够进行多个尺度的深度。编码器单元有 7 个卷积-反卷积对，每一个卷积-反卷积对的尺寸大小相同。与编码器单元对应，解码器单元内含有 7 个反卷积-卷积对，编码器单元中每个卷积层对中的后一个卷积层的输出与解码器单元中对应尺寸的反卷积层的输出沿通道维堆叠，堆叠结果作为该反卷积层后面的卷积层的输入。解码器单元中卷积层的输出的是视差图，像素的视差求倒数后得到对应像素深度，使用某一层输出网络的对应尺度的估计结果，因为网络结构限制，这里使用的尺度数最多为 7。网络结构见图 4.2：

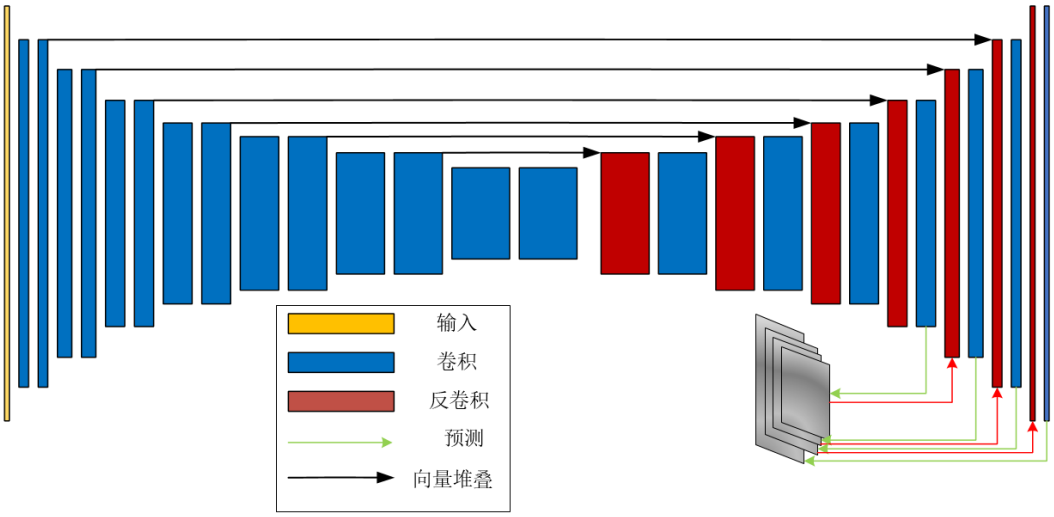

图 4.2 深度估计网络结构

如图 4.2 所示，单视图深度估计网络结构与编码器-解码器结构相似，网络前四个卷积层使用卷积核大小是分别是 7、7、5、5，其他层卷积核大小都是 3，全部卷积层都是用 ReLU 激活函数。解码器的反卷积过程可以进行多尺度的深度估计，为了保证深度估计结果在合理范围内，所有卷积层输出使用 $1/(\alpha * sigmoid(x) + \beta), (\alpha = 10, \beta = 0.1)$ 激活作为深度估计值。视差估计值取倒数并乘上相应的尺度因子后可得到目标视图深度的估计值 $\hat{D}_t$。

4.3 位姿估计网络

位姿估计网络是一个深度卷积网络，总共有 5 个用于特征提取的卷积层。输入数据是由目标视图 I_t 和源视图 $I_s (1 \le s \le N, s \ne t)$ 沿色彩通道串联而成，输出的一个分支预测目标视图与 $N-1$ 个源视图之间的 6-自由度（3 个欧拉角和 3 个平移向量）相对位姿 $T_{t\to s_i} (1 \le i \le N, i \ne t)$，6 个自由度的旋转向量中含 3 个欧拉角和 3 个平移向量，3 个欧拉角分别是绕 X 轴旋转的翻滚

角(roll)、绕Y轴旋转的俯仰角(pitch)和绕Z轴旋转的偏航角(yaw)，3 个平移向量分别是沿X轴平移t_x、Y轴平移t_y、Z轴平移t_z。另一个分支反卷积后的输出是多个尺度的可解释性掩模$\hat{E}_s$。网络结构见图 4.3：

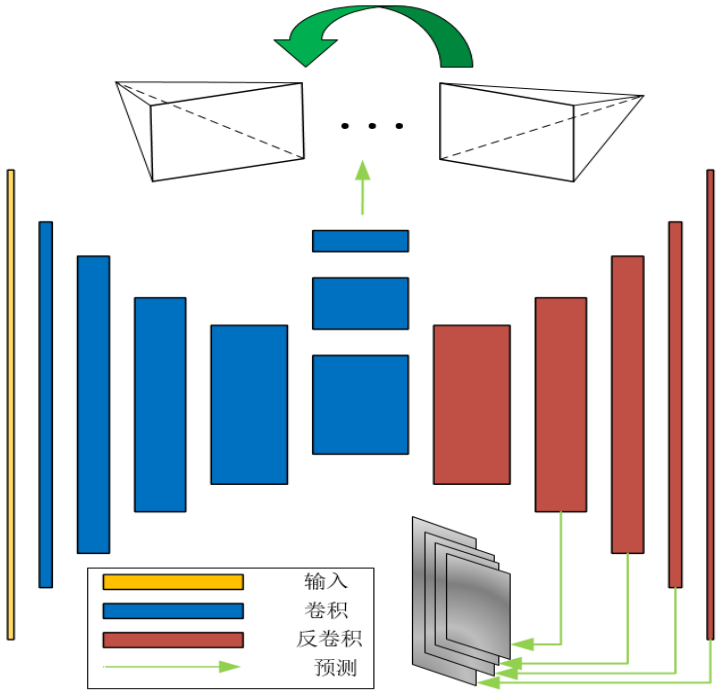

图 4.3 位姿估计网络结构

可解释性掩模$\hat{E}_s$用于减小因目标视图I_t和源视图$I_s(1\le s\le N, s\ne t)$之间出现遮挡、解除遮挡、光度变化或运动物体而造成的像素误差变化，上述情况会造成目标视图在源视图上投影点与源视图上相应点之间的像素差值不可微，因为这种误差不是网络预测的深度值或者相对位姿造成的。可解释性掩模作为相应源视图像素损失的权重，这样就能减小这种不可解释的误差。这样加权像素损失为：

$$\mathcal{L}_{vs}=\sum_{1\le s\le N, s\ne t}\sum_{p}\hat{E}_s(p)\,|I_t(p)-\hat{I}_s(p)| \qquad (4.4)$$

由于训练时对可解释性掩模$\hat{E}_s$没有特殊的监督信号，这样生成可解释掩模的网路总是将$\hat{E}_s$预测为 0 可使加权像素损失为 0。为了避免这种情况，在网络损失里添加一个正则项$\mathcal{L}_{reg}(\hat{E}_s)$使网络将$\hat{E}_s$预测为非 0 值，该正则项由可解释性掩模$\hat{E}_s$和标签 1 做交叉熵生成。

上述框架中一个重要问题是网络训练的损失主要来源于$\mathrm{I}(p_t)$和$\mathrm{I}(p_s)$四个相邻像素的差值，如果出现投影像素坐标p_s位于低纹理区域或者离当前估计值很远的情况这样的损失函数将会抑制训练过程。为了避免这种情况的出现，在损失函数里加入了多个尺度的平滑损失，这样网络损失可以来自更大的空间范围。平滑损失表达式：

$$\mathcal{L}_{smooth}=\sum_{p}s_t^u(\mathrm{p})\rho(\nabla_u\hat{D}_t(\mathrm{p}))+s_t^v(p)\rho(\nabla_v\hat{D}_v(p)) \qquad (4.5)$$

其中$\nabla_u\hat{D}_t(\mathrm{p})$和$\nabla_v\hat{D}_v(p)$是深度估计值沿竖直方向和水平方向的梯度，$s_t^u(\mathrm{p})$和$s_t^v(p)$是权重函数，与图像在对应方向的梯度有关，$\rho(x)=\sqrt{x^2+\xi^3}(\xi=0.01)$。

最终的损失函数变为：

$$\mathcal{L}_{final} = \sum_{l} \mathcal{L}_{vs}^{l} + \lambda_s \mathcal{L}_{smooth}^{l} + \lambda_e \sum_{s} \mathcal{L}_{reg}(\hat{E}_s^l) \quad (4.6)$$

其中 λ_s 和 λ_e 是损失的权重，需要通过实验来确定。

4.4 语义分割网络

语义分割网络采用 Vijay Badrinarayanan 等在 2016 年提出的 SegNet 结构[46]。与 DispNet[47]相似，SegNet 同样是形如编码器-解码器的架构，原文中 SegNet 使用了迁移学习方法，编码器共有 13 个卷积层（VGG-16 的前 13 个卷积层对应[13]）和 5 个最大池化层，解码器架构与编码器对应也有 13 个卷积层。这里对网络架构做了简化，编码器有 4 个卷积层和 4 个池化层，解码器有 4 个上采样层和 4 个卷积层组成。

语义分割网络对图像中所有点进行逐个分类，输入是需要生成语义的图像，输出是该图像中所有像素点类别的后验概率分布。

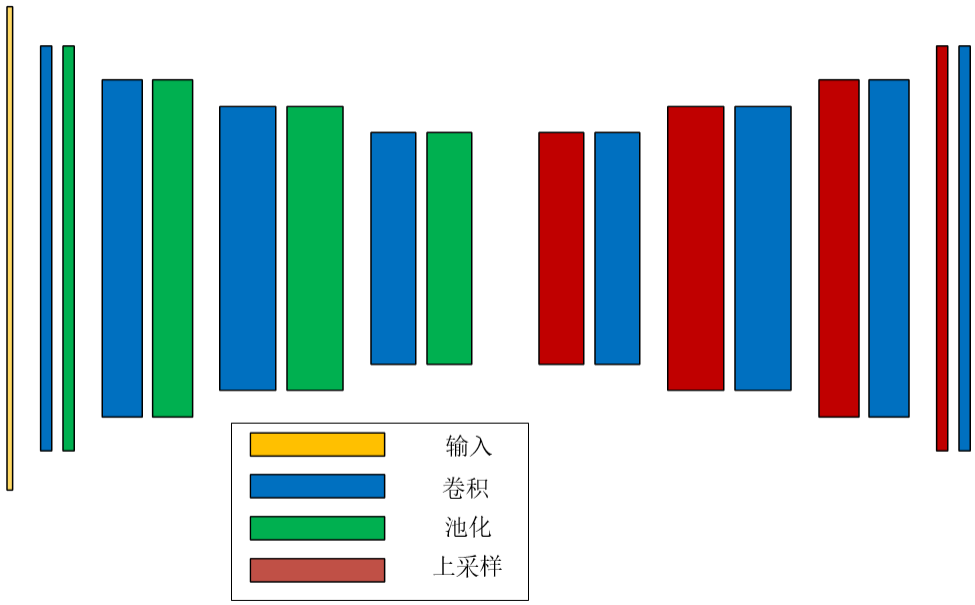

图 4.4 语义分割网络结构

4.5 语义与深度融合方法

标准的占据网格地图只存储网格占据概率的对数值，这里也需要存储网格的颜色值和标签概率分布，因为不同帧中的不同像素可能表示的是同一个网格，所以这里需要对颜色和标签分布进行融合然后再进行优化。对于颜色的融合，这里简单地取了不同位置观测像素值的平均值。标签的更新方法采用与网格占据概率相似的标准贝叶斯更新规则。用 z_t 表示某个网格在 t 时刻的标签概率分布，$l_{1:t}$ 表示从第一帧至今语义标签的观测值，对于新生成的语义标签图像 l_t 使用如下贝叶斯准则更新 3D 网格的语义标签：

$$p(z_t \mid l_{1:t}) = \frac{p(x_t \mid z_t)p(z_t)}{p(x_t)} \frac{p(x_{t-1} \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})}$$

$$\approx \frac{1}{Z} p(x_t \mid z_t) p(x_{t-1} \mid z_{1:t-1}) \qquad (4.7)$$

这里假设先验概率 $p(x_t)$ 是常数，并且 $p(z_t) / p(z_t \mid z_{1:t-1})$ 是一项归一化常量。这样对于每一个新到来的语义概率分布只需要乘上网格里存储的概率和一个常数即可。

4.6 数据预处理和数据增强

数据预处理分为调整图像大小和下采样。KITTI 数据集里程计子集中原始图像大小为 370×1226，为了使图像大小适应神经网络的输入尺寸，这里需要将图像大小调整为 128×416，图像大小调整后相机内参 K 也要做相应的调整。下采样是因为网络输出结果有多个尺度，为了将所有输出都用于计算损失训练网络，这里将输入图像下采样到相应的尺度。

数据增强是为了加快模型学习速度，在图像输入网络前对图像进行随机尺度变换和平移。随机尺度变换是在 1-1.15 之间随机取一个值作为尺度变换因子，然后将图像变换到相应尺度。平移与之相似，在图像的两个轴上小范围地移动图像，使网络输入数据更为多样化，加快网络学习速度。

5 实验

5.1 训练细节

深度估计网络和位姿估计网络使用了 KITTI 数据集的视觉里程计子集进行训练，KITTI 数据集是由德国卡尔斯鲁厄理工学院和丰田美国技术研究院联合推出的，该数据集使用包括多个传感器的移动数据采集平台采集，主要场景有市区、乡村和公路等多种场景，是现在国际上最大的自动驾驶数据集[48]。KITTI 数据集中包含有了深度估计、物体检测和里程计等多个子集，里程计子集总共有 22 个经过校准的图像序列，其中前 11 个图像序列提供了相应图像的位姿参数。这里用编号 0-8 共 9 个图像序列作为深度估计网络和位子估计网络的训练集，编号 9-10 的图像序列用于位姿估计网络测试。训练集 9 序列共包含 21039 张图像，由于这里设置序列长度 $N=3$，经过预处理后形成 21,021 个训练实例，其中 90%共 18,919 个实例用于训练，其余 10%用于测试。训练时设置了权重 $\lambda_s=0.5/l$（ l 是相应尺度的下采样因子)、$\lambda_e=0.2$，训练时除了输出层外所有层使用了 Batch-normalization，网络优化器使用了 Adam 优化器并且设置了 $\beta_1=0.9,\beta_2=0.999$，$batch\_size=4$，学习率为 $l=0.0002$。训练了大约 150,000 步后网络开始收敛。

语义分割网络的训练数据集由 CamVid 数据集和 KITTI 数据集的语义分割子集构成。CamVid 数据集是由剑桥大学 Brostow 教授的计算机视觉研究小组于 2008 年发布的用于图像语义研究领域的数据集，该数据集提供了一个道路场景视频序列的图像语义和相机位姿数据，其中语义标签由研究人员手动标定并进行复核。这里训练语义分割网络时使用了其中的 367 个图像序列作为训练数据，网络优化器为 SGD 优化器，设置冲量 $momentum=0.9$，$batch\_size=12$,学习率 $l=0.1$。在经过 100 次迭代后网络损失降到 0.2 左右，准确率约为 93.45%。

5.2 深度估计网络

使用 KITTI 数据集的原始数据(raw data)评估深度估计网络。这些图像序列是使用同一个数据采集平台在不同的场景（城市、居民区、道路和校区等）中采集的图像序列，除了含有四个相机捕捉的图像序列外，还带有从雷达读取的深度数据。这里从上述场景中各选一个序列用于深度估计网络的评估。

由于深度估计网络进行了多尺度的深度估计，在计算与雷达数据的误差之前，先将网络估计的深度值乘上一个尺度因子 $\hat{s}=median(D_{gt})/median(D_{pred})$ 使估计值的中位数与雷达测量数据匹配。深度估计值定量衡量指标分为误差指标和精确度指标，误差和精确度指标计算方式见附录 1。

表 5.1 深度估计网络定量指标

Scene	Seq Len	$\overline{s}$	Error metric				Accuracy metric		
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta<1.25$	$\delta<1.25^2$	$\delta<1.25^3$
City	69	3.25	0.150	1.247	6.807	0.226	0.786	0.933	0.977
Resident	474	2.87	0.191	1.622	7.488	0.303	0.697	0.868	0.943
Road	188	3.70	0.290	2.216	7.811	0.361	0.429	0.799	0.944
Campus	89	2.83	0.265	3.276	9.971	0.400	0.570	0.777	0.882

图 5.1 是深度估计网络的预测输出 $\hat{D}_t$ 可视化结果，深度图中颜色比较深的部分表明深度值较小，颜色比较浅的部分表明深度值较大。从图中能看出估计的深度基本能与图像中物体深度对应，前景与背景对比比较明显，远处天空区域预测的深度也比较准确。

图 5.1 深度估计结果

5.3 位姿估计网络

评估位姿估计网络时使用了预留作为测试集合的 09 和 10 两个图像序列，两个图像序列分别含有 1,591 和 1,201 张图像，测试时每个测试样例含 5 张图像，其中第三帧图像作为目标视图，其他图像作为源视图，位姿估计网络估计源视图与目标视图之间的相对位姿作为输出。

使用绝对轨迹误差（Absolute trajectory error ,ATE）指标（见附录 2）作为位姿估计网络的定性衡量指标，绝对轨迹误差是指在真实的尺度上（求误差之前将轨迹估计值与真实值之间进行对齐）轨迹估计值与真实值之间差值，ATE 是全部对应帧位姿误差的平均值[23]。表 5.2 是对位姿估计网络进行定量评估结果，可以看到绝对轨迹误差值在一个很小的范围内。

表 5.2 轨迹估计评估结果

评估序列	Seq. 09	Seq.10
ATE(m)	0.021 ± 0.017	0.020 ± 0.015

图 5.2 是使用位姿估计网络估计的位姿恢复出来的轨迹与轨迹测量值(ground-truth)对比图。中虚线是轨迹测量值（ground-truth），彩色线是轨迹预测值，彩色线条颜色表示预测值和测量值之间的误差。可以看根据估计的位姿恢复出来的运动轨迹与测量轨迹基本吻合，从图上来看，大部分时间预测轨迹与真实轨迹重合，只有在真实轨迹方向发生改变时预测值与真实值之间出现少许误差。

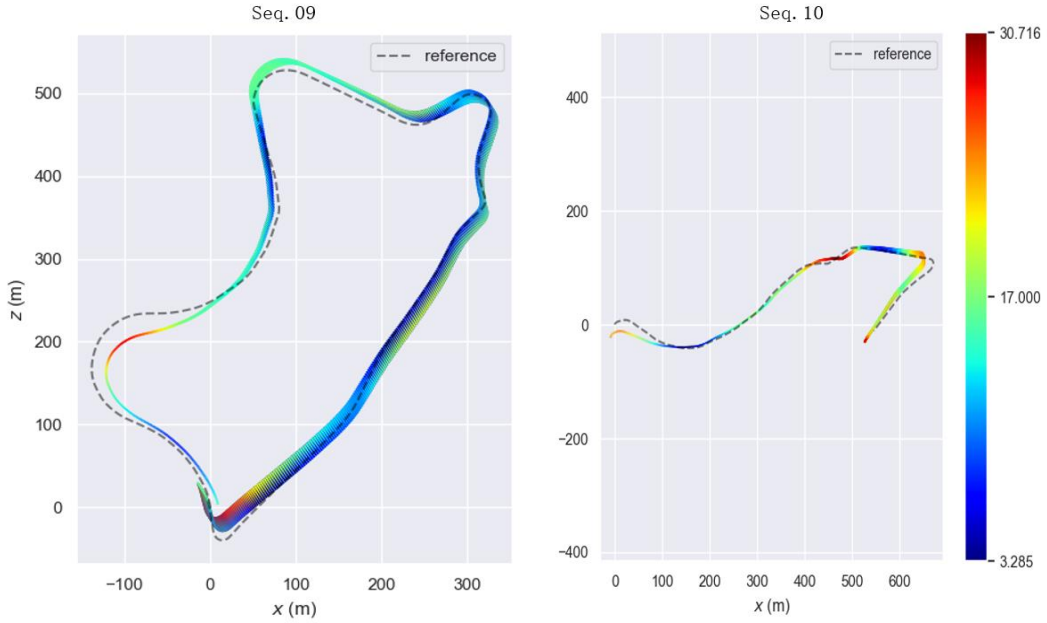

图 5.2 预测值和真实轨迹

图 5.3 是图像序列 9 和图像序列 10 各帧位姿估计值和测量值中平移分量的对照图。与图 10 一样，XYZ 三个轴向平移量的估计中沿 X 轴方向和沿 Z 轴方向的估计值和测量值基本吻合，在大部分时间内估计值和测量值一致，在真实轨迹发生转向的时候平移估计值与测量值出现少许偏差。沿 Y 轴平移量的估计值与测量值的走势相同，但是对应帧的沿 Y 轴平移量的估计值与测量值之间出现较多偏差。沿 Y 轴平移估计值与测量值出现较大误差是由于神经网络没有从数据集中学习到相关的模式，这是 KITTI 数据集的特点造成的，KITTI 以汽车作为移动数据采集平台，由人驾驶汽车在相应场景中行驶时采集数据，而这里采集数据的场景中道路几乎都是平稳的，这就导致汽车沿 Y 轴平移量很小；训练模型时 KITTI 数据图像经过处理后的宽:高≈4:1，采用这样图像比例虽然能够观察到更大的范围，但是这样也导致数据集中关于沿 Y 平移的模式信息被进一步压缩，神经网络在训练时在沿 Y 轴平移量估计上的误差没有产生相应的损失，这就导致了神经网络不能正确地学习到关于估计沿沿 Y 轴平移量的模式。

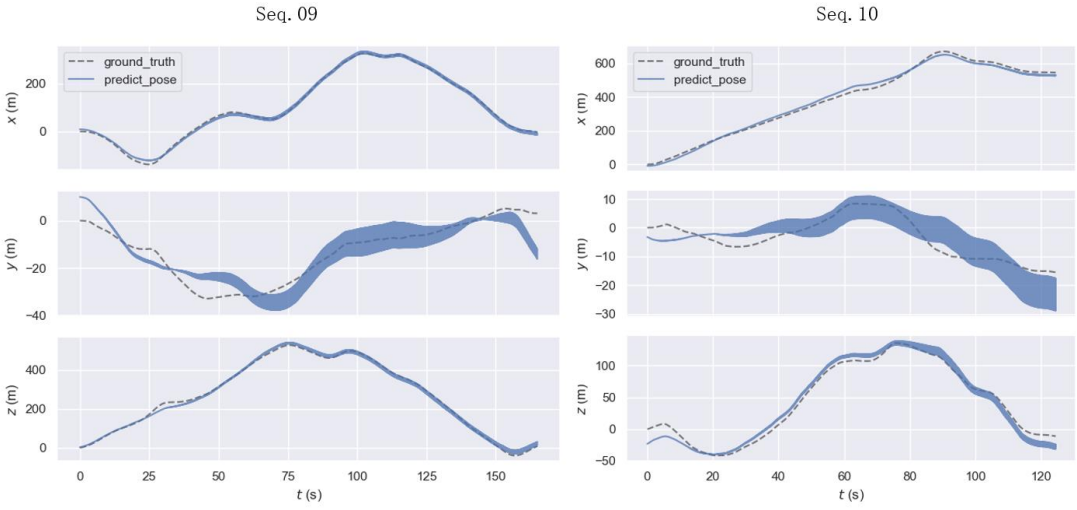

图 5.3 *XYZ* 轴平移估计值与真实值对比

图 5.4 是图像序列 9 和图像序列 10 各帧位姿估计值和测量值的旋转分量的对照图。

可以看到三个自由度的旋转估计上只有绕Y轴旋转角度(pitch)估计值比较准确，与真实值基本一致，绕X轴方向和绕Z轴方向的旋转角估计值和真实值有一些差距。出现这个现象同样是没能学习到估计绕X轴和绕Z轴旋转的模式，产生这样的差距还是与数据集和预处理相关。除此之外，从图上还可以看到当估计位姿中绕Y轴旋转角比较大的时候绕X轴方向的旋转角和绕Z轴方向的旋转角与测量值出现较大偏差，出现这个现象大概还与欧拉角这种旋转描述方式有关，欧拉角这种描述方式有万向锁问题，当第二次旋转角度为90°时，第一次旋转的轴和第三次旋转的轴重合，在第二次旋转角度接近90°时第一次和第三次旋转的角度可以取某个角或补角，其旋转效果相同。

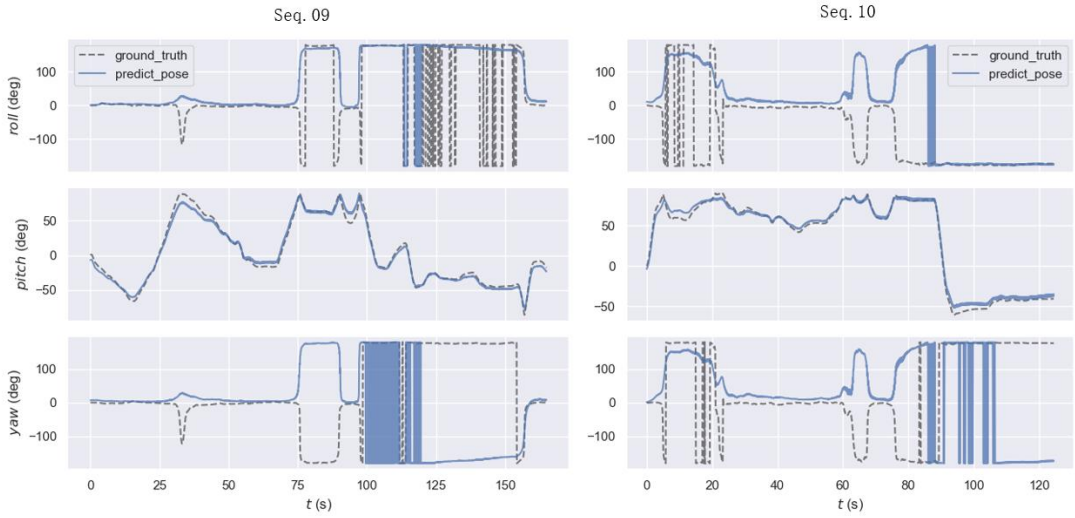

图 5.4 旋转向量（欧拉角）估计值与真实值对比

5.4 语义分割网络

语义分割网络利用 CamVid 中的语义图像和标签迭代了 100 次后网络损失函数已经很低，因为图像分辨率不高，为了防止出现过拟合情况，这里在迭代了 100 次后就没有再训练下去。图 5.5 是训练过程中网络的损失和准确率变化曲线。

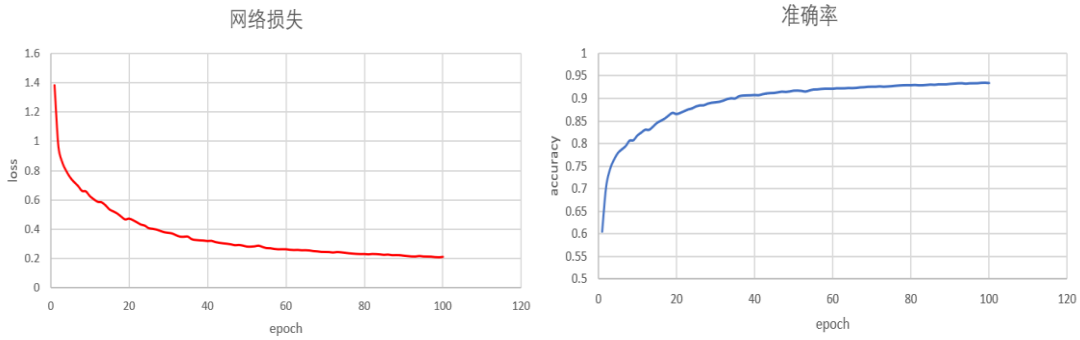

图 5.5 语义分割网络训练过程损失和准确率

图 5.6 是利用语义分割网络进行语义分割的结果，其中第一行是使用 CamVid 数据集迭代 100 次的模型在 CamVid 数据集上一个测试样例生成的结果，第二行是利用 KITTI 数据集中的一张图像作为测试样例的结果，第三行是利用 KITTI 数据集微调后的模型的结果。可以看到 CamVid 数据集训练的模型在 CamVid 数据集上的语义分割结果基本准确。但是把原模型应用

到 KITTI 数据集上语义分割效果并不是很理想，猜测这主要是因为两个数据集在场景上的差异引起的，CamVid 数据集是在城市道路上拍摄的，而 KITTI 大部分场景在居民区或者郊区拍摄。为了解决这个问题利用 KITTI 的语义分割数据集对模型进行了微调，再利用 KITTI 数据集进行语义分割的结果如下图，相对于原来只使用 CamVid 数据集的结果准确了很多。

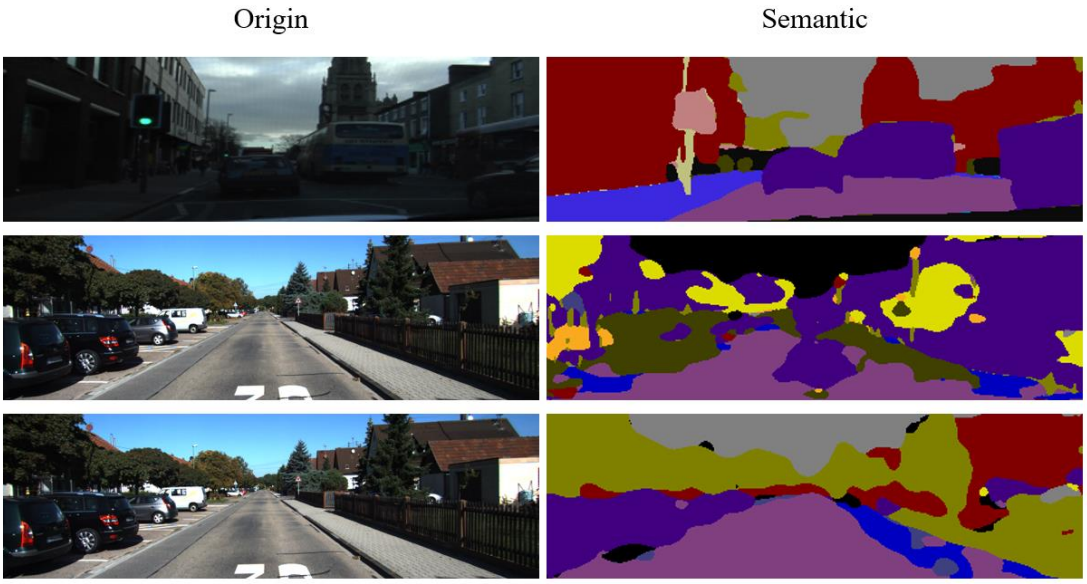

图 5.6 语义分割结果

5.5 融合语义的地图

使用上述方法生成网格占据地图，参照[50]的方法进行优化后将网格占据概率值和网格标签放入八叉树中，使用八叉树地图(Octomap)表示语义地图，这里设定的网格的分辨率为 0.05，即八叉树中最小的网格单元代表空间大小为 $0.05\times0.05\times0.05m^3$ 的实际空间。

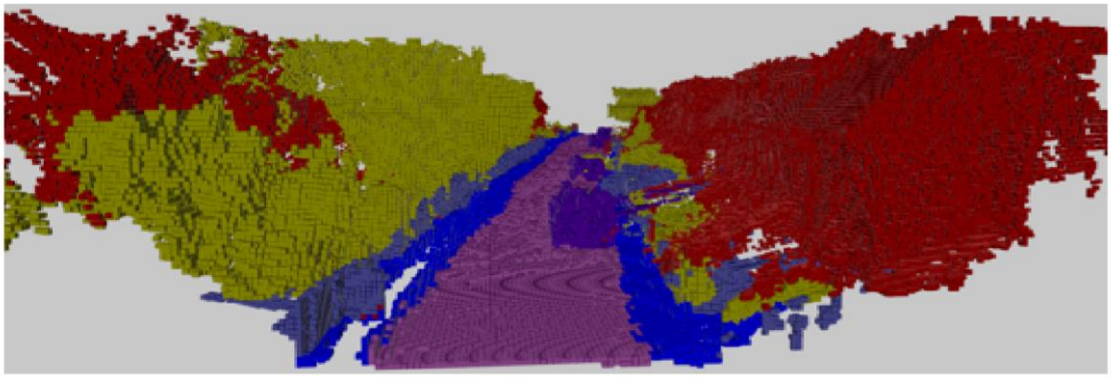

图 5.7 合成语义地图

从图 15 可以看出生成的语义地图中网格的空间位置和语义标签分布大致准确，物体空间外形保留了基本轮廓。但是也有一些不满意的地方，比如实际中有物体存在的空间在生成的地图是不被占据的，这主要是由于深度估计网络在估计大部分含有该点的图像深度的时候都产生了一个不合理的值，在进行融合的时候这类不合理的深度估计值被当作错误值丢弃（实际上除了丢弃也没有更好的办法），导致在合成的语义地图中相应占据网格没有被占据。

6 结论和展望

6.1 结论

本文尝试使用深度学习方法来完成语义 SLAM 中的几个基本任务，最终将网络输出结果用于构建语义地图。虽然神经网络在每一个任务中表现都不是非常突出，但是最终在一个比较好的优化方法的作用下得到的语义地图基本正确。说明在合适的方法的作用下，神经网络在复杂任务能够有比较好的表现，为我以后的研究探索找到了新的道路。

6.2 不足

本次实验探索虽然有一定成果，但是也还存在一些有待改进的地方。总结才能有收获，这里对于本次实验的不足做出如下几点总结。

6.2.1 尺度不确定性

本次实验由于训练时使用的是单个相机捕获的视频序列，与使用单目相机的方法相同，本实验中也有尺度不确定性问题。如前表 5.1 所示，深度估计网络估计的深度与真实深度之间有一个尺度因子的变换，且在不同场景的尺度因子还不相同。由于存在尺度不确定性问题，这个框架在要应用到实际中还需要很多辅助功能。

6.2.2 不能有效利用计算资源

实验中使用三个相对独立的神经网络结构，导致模型对于计算资源的需求比较高，且模型的结构具有相似性，存在不必要的资源浪费。实际上三个模型基本都是类似编码器-解码器结构，编码器用于提取图像特征，解码器生成相应输出，三个神经网络中编码器结构的重复出现是不必要的而且容易造成资源浪费。

6.2.3 无法进行回环检测和重定位

这两点是使用深度学习来完成 SLAM 任务的弊端。传统的 SLAM 中回环检测和丢失重定位非常依赖于特征的提取和匹配，在传感器观测到图像特诊与地图中存储的特征达到一定的匹配程度的时候就可认为传感器在地图中某个位置，而本实验中位姿、深度和语义都是使用神经网络直接输出，构建的地图中没有存储图像特征信息，这就使回环检测和重定位的难度非常大。

6.3 进一步研究计划

6.3.1 解决尺度不确定性问题

在构建用于导航和定位的地图过程中，确定尺度非常重要。传统 SLAM 技术中常常使用双目相机捕获的图像解决尺度问题，接下来计划参照传统的方法，使用双目相机捕获的图像序列来训练神经网络，并将使用神经网络输出深度进行左右投影产生的损失加入到神经网络的损失函数中，迫使神经网络学习与真实世界对应的尺度模式。

6.3.2 尝试共用特征提取器

6.2.2 节讲到三个神经网络结构相似且重复性较高，接下来将探索使用共同的特征提取层。一方面共用特征提取层可以提高计算效率，另一方面使用在大规模数据集上预训练的深度卷积网络（如 VGG[13]）作为特征提取器能提取更好的特征，为特征提取层后的任务网络提供更好的输入。

6.3.3 回环检测和重定位

探索回环检测和丢失重定位。这里已经有使用深度学习进行回环检测和重定位的相关工作[14, 42]，接下来也将参考相关论文尝试实现回环和丢失重定位。另外由于构建的地图中含有语义标签，下一步将尝试使用语义辅助回环检测和重定位。

6.3.4 尝试使用生成对抗网络方法

根据文献[49]，这里生成深度和图像语义的方法都是由图像-图像转换的任务。GAN 在这类任务中有非常好的表现，因为使用损失函数直接监督神经网络的方法中构建损失函数的步骤需要很多专业知识，很多时候手工构建的损失函数效果并不理想。而使用 GAN 只需要给判别器(Discriminator)构建简单的损失函数，再由判别器学习生成器损失函数。这种方法在很多图像-图像转换任务中表现比较好，下一步工作中尝试使用这种方法构建深度学习模型。

参考文献

[1] Durrant Whyte, H, and Bailey, Tim. “Simultaneous Localization and Mapping: Part I.” IEEE Robotics & Amp Amp Automation Magazine 13.2(2006):99 - 110.

[2] 陈常，朱华，由韶泽．基于视觉的同时定位与地图构建的研究进展［J/OL］．计算机应用研究, 2018,(03):1-9(2017-08-18).

[3] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in Proceedings of the fourth Eurographics symposium on Geometry processing, vol. 7, 2006.

[4] J. Stuckler and S. Behnke, “Multi-resolution surfel maps for efficient dense 3d modeling and tracking,” Journal of Visual Communication and Image Representation, vol. 25, no. 1, pp. 137–147, 2014. R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux,

[5] S. Hodges, D. Kim, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in 2011 10th IEEE international symposium on Mixed and augmented reality (ISMAR), pp. 127–136, IEEE, 2011.

[6] R. A. Newcombe, D. Fox, and S. M. Seitz, “Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 343–352, 2015.

[7] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, “Elasticfusion: Dense slam without a pose graph,” Proc. Robotics: Science and Systems, Rome, Italy, 2015.

[8] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al., “Fusion4d: Real-time performance capture of challenging scenes,” ACM Transactions on Graphics (TOG), vol. 35, no. 4, p. 114, 2016.

[9] Stan Z. Li. “Markov random field models in computer vision.” European conference on computer vision. Heidelberg: Springer,1994:361-370.

[10] John Lafferty， Andrew Mccallum， Fernando C.N. Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.” International Conference on MaG chine Learning. Wiliamstown: Morgan Kaufmann,2001:282-289.

[11] Lecun Y, Bottou L, Bengio Y, et al. “Gradient based learning applied to document recognition.” Procedings of the IEEE,1998,86(11):2278-2324.

[12] Alex Krizhevsky，I Sutskever，G Hinton.“Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. Nevada:ACM,2012:1097-1105.

[13] Karen Simonyan, Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv preprint arXiv:1409.1556,2014.

[14] Jonathan Long, Evan Shelhamer, Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation.” arXiv preprint arXiv:1411.4038v2.

[15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L.Yuille. “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.” arXiv preprint arXiv:1412.7062v4.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. “Mask R-CNN.” arXiv preprint arXiv:1703.06870v3.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun.“Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” arXiv preprint arXiv:1506.01497v3.

[18] C. Cadena and L. Carlone and H. Carrillo and Y. Latif and D. Scaramuzza and J. Neira and I. Reid and J.J. Leonard, “Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age”, in IEEE Transactions on Robotics 32 (6) pp 1309-1332, 2016.

[19] Montemerlo M, Thrun S, Koller D, et al. “FastSLAM: a factored solution to the simultaneous localization and mapping problem.” In: Proceedings of the AAAI National Conference on Artificial Intelligence, pp. 593-598.

[20] Guivant J.“Efficient simultaneous localization and mapping in large environments.” Dissertation, University of Sydney, Australia.

[21] A. Davison, I. Reid, N. Molton, and O. Stasse, “Monoslam: Real-time single camera SLAM,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

[22] G. Klein and D. Murray, “Parallel tracking and mapping for smallar workspaces,” in Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on, pp. 225–234, IEEE, 2007.

[23] R. Mur-Artal, J. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” arXiv preprint arXiv:1502.00956, 2015.

[24] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in Computer Vision–ECCV 2014, pp. 834–849, Springer, 2014.

[25] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on (rs,ed.), pp. 15–22, IEEE, 2014.

[26] Milford M, Wyeth G, Prasser D. “RatSLAM: a hippocampal model for simultaneous localization and mapping.” In: Proceeding of the IEEE International Conference on Robotics and Automation, 1:403-408.

[27] Milford M, Wyeth G.“Mapping a suburb with a single camera using a biologically inspired SLAM system.” IEEE Trans Robot, 24(5):1038-1053.

[28] Glover A, Maddern W, Milford M, et al. “FAB-MAP + RatSLAM: appearance-based slam for multiple times of day.” In: Proceedings of the IEEE International Conference on Robotics and Automation.

[29] Collett M.“How desert ants use a visual landmark for guidance along a habitual route.” In: Psychological and Cognitive Sciences, 107(25):11638-11643.

[30] Triggs B, Mclauchlan P, Hartley R, Fitzgibbon A.“Bundle adjustment – a modern synthesis.” In: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, pp. 298-375.

[31] Engels C, Stewénius H, Nistér D.“Bundle adjustment rules.” In: Photogrammetric Computer

Vision.

[32] Pollefeys M, Van L, Vergauwen M, et al.“Visual modeling with a hand-held camera.” Int J Comput Vis, 59(3): 207-232.

[33] A. Nüchter, J. Hertzberg. “Towards semantic maps for mobile robots.” Robotics and Autonomous Systems, vol. 56, no. 11, pp. 915–926, 2008.

[34] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, J. Montiel.“Towards semantic slam using a monocular camera.” in Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 1277–1284, IEEE, 2011.

[35] N. Fioraio and L. Di Stefano.“Joint detection, tracking and mapping by semantic bundle adjustment.” 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1538–45, 2013.

[36] Flint, D. Murray, I. D. Reid. “Manhattan Scene Understanding Using Monocular, Stereo, and 3D Features.” In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2228– 2235. IEEE, 2011.

[37] Vineet, O. Miksik, M. Lidegaard, M. Niessner, S. Golodetz, V. A. Prisacariu, O. Kahler, D. W. Murray, S. Izadi, P. Peerez, and P. H. S. Torr. “Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction.” In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 75–82. IEEE, 2015.

[38] Sean L. Bowman，Nikolay Atanasov，Kostas Daniilidis. “Probabilistic data association for semantic SLAM.” DOI：10.1109/ICRA.2017.7989203.

[39] K.Konda, R.Memisevic. “Learning visual odometry with a convolutional network.” in International Conference on Computer Vision Theory and Applications, 2015.

[40] Jamie Watson, Michael Firman, Gabriel J. Brostow, Daniyar Turmukhambetov. “Self-Supervised Monocular Depth Hints.” arXiv preprint arXiv:1909.09051v1.

[41] A.Kendall,M.Grimes,R. Cipolla.“Posenet: A convolutional network for realtime 6dof camera re localization.” in Proceedings of the IEEE International Conference on Computer Vision, pp. 29 38–2946, 2015.

[42] Y.Hou, H.Zhang, S.Zhou.“Convolutional neural networkbased image representation for visual l oop closure detection.” arXiv preprint arXiv:1504.05241.

[43] Yasin Almalioglu, Muhamad Risqi U. Saputra, Pedro P. B. de Gusmo, Andrew Markham, Niki Trigoni. “GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks.” arXiv preprint arXiv:1809.05786v3.

[44] Keisuke Tateno, Federico Tombari, Iro Laina1, Nassir Navab. “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction.”

[45] Tinghui Zhou, Matthew Brown, Noah Snavely, David G. Lowe. “Unsupervised Learning of Depth and Ego-Motion from Video.”

[46] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561v3 [cs.CV] 10 Oct 2016.

[47] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4040–4048, 2016. 4.

[48] Andreas Geiger, Philip Lenz, Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite.”

[49] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks.” arXiv:1611.07004v3 [cs.CV] 26 Nov 2018.

[50] Shichao Yang, Yulan Huang, Sebastian Scherer. “Semantic 3D Occupancy Mapping through Efficient High Order CRFs.” arXiv:1707.07388v1 [cs.CV] 24 Jul 2017.

[51] 高翔,张涛等.《视觉 SLAM 十四讲：从理论到实践》[M]. 北京:电子工业出版社. 2017:1-377。

[52] Richard Hartley,Andrew Zisserman. “Multiple View Geometry in Computer Vision.” Cambridge University Press. 2003.

[53] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects[J]. In 2013 IEEE Conference on Computer Vision and Pattern Recognition, pages 1352–1359, June 2013.

[54] Raul Mur-Artal,Juan D. Tardos. “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras.” IEEE Transactions on Robotics[J]. 2017(33): 1255 – 1262.

[55] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” Autonomous Robots, vol. 34, no. 3, pp. 189–206, 2013.

附录

附录 1 深度指标计算公式

公式 1 绝对相对误差

$$\mathrm{Abs.Re}l = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{\mathrm{D}}_i - \mathrm{D}_i|}{D_i}$$

公式 2 平方相对误差

$$\mathrm{Sq.Re}l = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{\mathrm{D}}_i - \mathrm{D}_i|^2}{D_i}$$

公式 3 均方根误差

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\hat{\mathrm{D}}_i - \mathrm{D}_i|^2}$$

公式 4 对数均方根误差

$$\log.RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\log\hat{\mathrm{D}}_i - \log\mathrm{D}_i|^2}$$

公式 5 精确度指标

$$\max(\frac{\hat{D}_i}{D_i}, \frac{D_i}{\hat{D}_i}) = \delta < T$$

注： $\hat{D}$：深度估计值 D：深度测量值 N：像素总数 T：阈值（$1.25, 1.25^2\ 1.25^3$）

附录 2 ATE 计算公式

$$ATE = \frac{1}{N}\sum p_i s\hat{p}_i$$

注： $\hat{p}$：位姿估计值 p：位姿测量值 s：相似变换矩阵，$s \in Sim(3)$

$Sim(3)$ 是将位姿估计值对齐到位姿测量值用的相似变换矩阵

谢辞

本论文的顺利完成离不开我的导师孙杳如教授的悉心指导。今年情况特殊，受到疫情影响开学不能返校，老师在获知学校不安排返校后主动打电话了解情况，询问在家里是否有网络、在家里能否完成毕业设计，并指导我完成毕业设计。老师的无微不至的关心让我如沐春风，专业的指导及时纠正了我思路上的偏差。老师在我毕业设计上倾注了无数心血，在此谨向老师表示最崇高的敬意和最衷心的感谢。

此外，本论文的完成也离不开实验室赵云皓学长和阚高远学长的帮助。这一路走下来遇到了不计其数的困难，每当我向赵学长求助的时候赵学长总是有求必应，帮我解决了很多困难，在此我要感谢赵云皓学长的帮助。在实验遇到困难难以继续下去的时候是阚高远学长在该我加油鼓劲，给我提供参考资料、帮我解决问题，阚学长待我像亲兄弟一样，阚高远学长的帮助我铭记于心。

在今年这种特殊情况下能按时完成毕业论文的工作和答辩同样也离不开电子与信息工程学院各位领导和老师辛勤工作，有了他们的无私付出，学生们才能在不能返校的情况下顺利完成学业，向学院领导和老师致以最真诚的敬意。