基于深度学习和传统视觉 SLAM 融合的单目视觉 LSLAM

重庆大学硕士学位论文

（学术学位）

学生姓名：陈剑斌

指导教师：李　军　副教授

专　　业：控制科学与工程

学科门类：工　学

重庆大学自动化学院

二 O 一八年四月

Deep Learning and Traditional Vision SLAM Based Monocular SLAM

A Thesis Submitted to Chongqing University

in Partial Fulfillment of the Requirement for the

Master's Degree of Engineering

By

Chen Jianbin

Supervised by As. Prof. Li Jun

Specialty：Control Engineering

College of Automation of Chongqing University,

Chongqing, China

April 2018

摘　要

SLAM（Simultaneous Localization and Mapping）即时定位与地图构建技术对机器人实现环境和自我感知具有十分重要的作用，对该问题的研究对实现机器人走向高度智能化具有十分重要的理论意义和实际应用价值。SLAM 技术要求机器人依赖自身的传感器获取的数据建立环境地图信息，同时确定自身在场景中的位姿，并依赖后续获取的数据建立新的状态并更新修正前面的状态。由于单目视觉传感器独特的优势，单目视觉 SLAM 技术得到了国际学者的广泛关注，成为了 SLAM 领域中的重要研究方向。

当前，单目视觉 SLAM 技术可以分为间接法和直接法。间接法依赖稀疏的场景关键点的重投影误差建立约束对系统状态向量进行更新；直接法基于场景显著点灰度不变假设利用灰度误差建立约束。单目视觉 SLAM 技术目前在实验室取得了很好的实验效果，但是在实际应用中由于环境的复杂性，单目视觉 SLAM 技术还存在很多问题，例如其初始化尺度的估计，尺度漂移，闭环检测等。基于深度学习解决 SLAM 中的子问题当前也是计算机视觉领域研究的热点，而且取得了很多成果，同时如何依赖深度学习促进 SLAM 技术的发展也是当前 SLAM 技术中最前沿的研究。

本文就深度学习与传统视觉 SLAM 融合解决其当前存在的问题，并提高算法在复杂场景中的表现进行了研究。基于此设计了一个算法，其基于优化框架将深度卷积场景深度预测网络 FCRN 与传统 DSO SLAM 算法融合转化为各自获取的信息的优化融合。引入深度学习使本文设计的 SLAM 算法具有学习性，所以本文称之为 Learning SLAM（简称 LSLAM）算法。算法在单目视觉 SLAM 的初始化阶段，算法通过引入深度卷积网络对场景深度的预测信息解决了无法估计初始场景尺度的问题；在相机运动估计阶段，引入预测的场景深度信息对尺度进行监督和更正，从而有效的减少了系统运行中的尺度漂移；在场景关键桢初始化中加入预测的深度信息对其深度参数进行初始化，从而改善后续优化的效果。

在论文的第一部分就视觉 SLAM 技术和基于深度学习的 SLAM 技术的发展进行了阐述，并对本文设计的算法框架进行了说明。第二部分对传统视觉 SLAM 中的关键核心部分就直接法和间接法两个角度进行了原理的介绍，并对其中隐藏的问题进行了分析。第三部分对当前基于深度卷积神经网络进行场景深度预测的经典网络设计方案做了详细的介绍。第四部分对本文设计的 LSLAM 算法各个部分进行了详细说明，并对各个部分做了合理性验证的实验。第五部分对本文的硬件平台—机器人“甲鲁普”、软件平台—ROS 机器人操作系统进行了介绍，然后使用实

验对本文设计的算法进行了性能的验证。

为了验证 LSLAM 算法的性能，本文共设计了 3 个综合实验对其进行验证。第一二个实验选取了 4 个数据集分别就 SLAM 问题中最核心的机器人定位进行了与传统 DSO SLAM 算法和 RGB-D SLAM 算法的对比实验。第三个实验在实验室环境下就机器人两种常见的运动模式（四边形，圆形）与 DSO 算法就机器人绝对定位误差进行了比较。第四个实验在室内过道大场景下，进行了与 DOS 算法的综合对比实验。四个实验都说明了本文设计的 LSLAM 算法在机器人定位精度上的提升，在面对复杂场景时算法具有更好的表现，证明了本文提出的融合框架的有效性。

关键词：单目视觉 SLAM，直接稀疏里程计，场景深度预测卷积神经网络，优化

ABSTRACT

Simultaneous Localization and Mapping (SLAM) plays a very important role in the robot's realization of environment and self-awareness. Research on this issue is of great theoretical and practical value for making the robot to be highly intelligent. SLAM methods requires the robot to establish environmental map based on the data acquired by its own sensors, and determine its position in the scene. Next time it need update the previous state and build a new state by the subsequently acquired data. Due to the unique advantages of the monocular vision sensor, monocular SLAM has received extensive attention from international scholars and has become an important research direction in the field of SLAM.

Currently, monocular SLAM approaches can be divided into indirect method and direct method. The indirect method relies on the reprojection error of the sparse image feature points extracted to add the constraint to the state of the system, but the direct method is based on the assumption that the pixel gray value of a particular view spot is invariant in two images. Now monocular SLAM methods have got good experimental results in the laboratory, but in practical applications, because of the complexity of the environment, there are still many problems in the monocular SLAM, such as the initialization scale, the scale drift, the loop closure detection and so on. Resolving sub-problems in SLAM based on deep learning is currently a highly active research in the field of computer vision, and has obtained many results. At the same time, how to rely on deep learning to promote the development of SLAM technology is also the most cutting-edge research in SLAM.

In this paper, we study the combination of deep learning and traditional visual SLAM to solve its current problems and improve the performance of algorithms in complex scenes. Based on this, an algorithm is designed, which transform the combination of the deep convolution scene depth prediction network FCRN and the traditional DSO SLAM algorithm into the optimal fusion of Information obtained from each other based on the optimization framework. The introduction of deep learning makes the SLAM algorithm designed in this paper have the ability of learning, so this is called Learning SLAM (LSLAM) algorithm. In the initial stage of Monocular SLAM, the algorithm solves the problem of inability to estimate the initial scale by introducing the depth information obtained by the depth prediction network; In the phase of the

motion estimation, the predicted depth information is used to supervise and correct the scale, thus effectively reducing the scale drift in the system. In addition, the depth information of key point in key frame is initialized under the help of the predicted depth information so as to improve the effect of follow-up graph optimization.

In the first part of this paper, the visual SLAM and its development based on deep learning are expounded, and the algorithm framework designed in this paper is explained. The second part introduces the principle of the core parts of traditional visual SLAM from two view of direct and indirect methods, and analyzes the hidden problems. The third part gives a detailed introduction to the classic deep network architecture for scene depth prediction based on convolutional neural network. In the fourth part, the various parts of LSLAM algorithm are described in detail, and the verification experiment of rationality are performed for each part. In the fifth part, the hardware platform of this paper—the robot "Jappeloup" and the software platform—ROS (robot operating system) are introduced, and then the performance of the algorithm designed in this paper is verified by experiments.

In order to verify the performance of LSLAM algorithm, we design 3 experiments. In the first two experiments, four data sets were selected to perform comparison experiments with the traditional DSO SLAM algorithm and the RGB-D SLAM algorithm on the most essential robot positioning. In the third experiment, the absolute translation error is compared with the DSO algorithm on two common motion patterns (quadrilateral and circle) of robot in the laboratory. The fourth experiment carried out a comprehensive contrast experiment with DOS algorithm in the indoor corridor. The four experiments all show that the LSLAM algorithm designed in this paper improves the positioning accuracy of the robot. In the face of complicated scenes, the algorithm has a better performance, which proves the effectiveness of the proposed optimal fusion framework.

Keywords: Monocular SLAM, Direct Sparse Odometry，Depth prediction convolution neural network, Optimization

目　录

1 绪论

1.1 研究背景与研究意义

1.1.1 课题来源

本课题来源于导师承担的国家自然科学基金项目“基于认知学习的智能机器人控制系统关键问题的研究”（项目编号：61473052）。

1.1.2 研究背景与意义

机器人的定义是：“A robot is a machine which can be programmed to perform some tasks under automatic control”，即“机器人是一种通过编程在自动控制下可完成某些任务的机器”[1]。机器人已经在工业应用中取得了显著的成果，同时随着社会的发展和机器人技术的革新，社会对于机器人在各个领域的应用需求不断攀升，机器人在社会劳动中扮演的角色越来越重要。机器人在近几十年来已经从传统的装配、焊接、喷涂、搬运等工厂制造环境领域拓展到了海洋、航天、军事、农业、服务、娱乐等非制造环境领域[2]。各个国家对于机器人的发展越来越重视，对于智能机器人的研究也成为了国际研究的热点，其中对于移动机器人的研究最为广泛。当前移动机器人的研究成果在各个领域已经得到了广泛的应用。例如在探测未知星球中发挥重要作用的美国 NASA 的火星探测机器人“勇气号”和“机遇号”；在地震中进行废墟搜索与辅助救援的救援机器人；在安防中的用来解决安全隐患，巡逻监控及灾情预警的安防机器人；在仓库里分拣货物，运输的仓储物流机器人；在家里扫地的扫地机器人；以及当前最热门的无人驾驶汽车。可以说机器人的应用我们随处可见，移动机器人的需求日益显著，未来大部分的社会劳动将会被机器人所取代。国家也将多项移动机器人的研究纳入了国家 863 计划[3]，在机器人的研究上已经开展了大量的工作，对于移动机器人的研究具有重要的科研意义，也具有广阔的应用前景。

移动机器人是一个复杂的综合系统，其容纳了环境和自我感知，行为规划、控制与执行等多种功能[4]。其与一般工业机器人不一样的是其工作环境通常是非结构化的，具有可移动、适应性强、智能化水平高等特点。移动机器人处于高复杂的作业环境中，所以自主导航是实现其完全自主的必要也是最重要的能力。其也是移动机器人研究领域中最关键的技术之一。对于移动机器人实现自主导航，其要解决的问题可以归结为：（1）“where am I now ?” 即是指机器人的定位问题——机器人如何通过传感器获取的数据确定自身的位置和姿态；（2）“What is the structure of the environment ?” 即是指环境地图的构建——机器人如何通过传感器获取的数据感知环境；（3）“How can I get that target position ?” 即是移动机器

人的导航——机器人在当前环境和姿态下如何在某些限制下到达特定位置[5]。前面两个问题的实质其实就是移动机器人同步定位与地图构建（simultaneous localization and mapping）即 SLAM 问题，而后一个问题依赖于 SLAM 问题的解决。SLAM 问题就是指系统模型已知，初始状态未知的移动机器人在具有某些特质的环境中运动时，利用自身所带的传感器感知机器人及环境的信息，实时的确定自身的位置坐标，同时建立起环境的增量式地图。所以对于 SLAM 问题的研究是实现移动机器人自主导航的关键，也是真正实现自主的重要条件之一[6]。积极研究 SLAM 技术对于提高移动机器人的智能化、自主性和适应性有着十分重要的作用和意义。

当前，SLAM 技术从使用的传感器类别可以划分为：激光 SLAM[7,8,9,10]，视觉 SLAM[11,12,13,14,15,16,17]。激光 SLAM 即采用激光雷达获取环境数据，而视觉 SLAM 即是采用视觉传感器构建 SLAM 系统。近些年来，随着视觉传感器的迅速发展，视觉传感器因其获取的信息丰富、价格低廉等优势在 SLAM 系统中得到了广泛地应用。对于视觉 SLAM 的研究也是近年来计算机视觉研究中的热门研究方向。视觉 SLAM 中采用的视觉传感器主要分为三类：单目摄像头、双目摄像头、深度摄像头。三类传感器的优缺点如表 1.1 所示。单目视觉 SLAM 因其硬件结构简单、灵活方便、可拓展领域广泛等特点，更是成为了视觉 SLAM 领域中研究的焦点，本文也选择了对其进行研究。

表 1.1 视觉传感器对比

Tab1.1 Visual sensor contrast

视觉传感器	优缺点
单目摄像头	结构简单、成本低、便于推广、室内室外都可、尺度不确定、运动测距
双目摄像头	立体视觉、室内室外都可、标定复杂、计算量大
深度摄像头	直接获取深度、计算量小、深度测量窄、易受光照影响、室外难应用、成本高

当前，视觉 SLAM 已经取得了一定的进展，视觉 SLAM 技术已经在众多的产品中出现了其身影，例如可穿戴设备，增强现实技术（AR）、三维重建、智能家居等，但是大部分的研究成果还仍然停留在实验室阶段，其主要原因有两点：（1）运行环境的复杂性；（2）系统运行的实时性。所以对于研究高实时性和高鲁棒性的 SLAM 系统具有十分重要的意义和价值。深度学习算法是当前计算机视觉领域

最活跃的算法，其利用多层神经网络学习图像更深层次的特征表达。在众多的任务中与传统方法相比都取得了更好的效果，例如场景分割、物体识别等。其高抗干扰能力尤其突出。最近，基于深度学习实现单张图像恢复场景深度、视觉里程计、SLAM 闭环检测、语义 SLAM 等方面的研究得到了广泛地关注，并且已经取得了一些很好的成果。同时对于深度学习和传统 SLAM 算法融合的研究也成为了 SLAM 最前沿的研究热点。本文旨在研究融合深度学习和传统单目视觉 SLAM 解决当前单目视觉 SLAM 还存在的一些难题。

1.2 研究现状概述

1.2.1 传统视觉 SLAM 研究现状

SLAM 技术具有重要的理论和实际应用价值。在移动机器人界，其被广泛认为是实现移动机器人真正完全自主的核心，甚至被称为“圣杯(Holy grail)”[7]。SLAM 同时定位与地图构建最先由 Smith、Self 和 Cheeseman 提出[18]。最初的工作是由 Smith[19]和 Durrant-Whyte 等人[20]在文献中描述了机器人位置和环境路标的空间几何不确定性和两者的相互关系的表达。1989 年 P. Moutarlier and R. Chatlia 等人提出了如何融合机器人位置和环境路标的不确定性[21]。1990 年，R. Smith, M. Self, and P. Cheeseman[22]等人在文献中正式提出了如何根据获取的信息来估计机器人中的空间位置同时增量式的建立环境地图。至此，Smith and Durrant-Whyte 等人的研究成果奠定了 SLAM 理论研究的框架，同时基于卡尔曼滤波建立起了最初的 SLAM 问题的解决方案。2001 年，Paul Newman[7]在文献中证明了 Smith 等人提出的 SLAM 框架中系统的各个状态更新的收敛性，验证了 SLAM 框架的理论可行性。在该基础上，Paul Newman 基于扩展卡尔曼滤波（EKF[23]）也提出了 SLAM 问题的一种解决方案。至此，SLAM 技术的基础概率理论基本成熟，对于 SLAM 的研究热潮也就此打开。前面的两种解决方案都将环境路标的位置以及机器人的当前位姿表示为系统当前状态向量 X，存在的一个最大问题就是：环境特征的增加使协方差中元素成平方增加，计算复杂度会达到 O(n3)[24]，n 为环境中的路标数量。当地图规模比较大时，会耗费大量的计算资源，在保证算法实时性下，极大的限制了算法在实际应用中的路标数量。2002 年，Montemerlo and Thrun 等人基于粒子滤波提出了 Fast SLAM[9]解决了对于路标数量的限制。其采用粒子群代表机器人某一时刻的位置，在每个粒子上使用卡尔曼滤波器处理建图问题。该算法在一定程度上提高了 SLAM 技术应用的地图规模，但是也引入了粒子耗散、粒子退化等问题，而且在传感器不够准确的情况下，需要比较多的粒子数，这样也加剧了计算量。

基于关键桢的方法被公认为视觉 SLAM 技术发展过程中一个重要的里程碑,其

由 Thrun 等人[25]提出。在此之前，视觉 SLAM 技术采用传统的滤波技术消除系统运行中的积累误差，而基于关键桢的方法研究采用误差最小化数学优化的方法寻找系统最优的参数值。该方法也被称为基于图优化的方法。对于这一方法采用一般的优化求解方法计算成本很高，但是后来随着稀疏代数理论的兴起，研究者发现了 SLAM 优化问题中雅克比矩阵和海塞矩阵的稀疏性[26]。利用稀疏矩阵分解来求解线性化的 SLAM 问题极大的提高了算法的实时性，使 SLAM 算法向大规模场景的应用成为可能。G Klein and D Murray 等人在文献[15]中提出的平行追踪与建图的 PTAM 算法（Parallel Tracking and Mapping for Small AR Workspaces）是其中最具代表性的方法之一。该算法框架将图像桢中某些包含较多显著且新的信息的图像桢选择作为关键桢，并计算该图像桢中机器人的位姿以及图像中环境地图点的位置信息作为一个计算节点。在算法后端采用数学优化的方式消除积累误差。在 PTAM 算法中，更是将算法前端追踪相邻桢的位姿变换和后端的环境地图构建及各状态量的优化分隔成两个平行运行的线程，这大大的提高了算法运行的效率，在文中作者将该算法在一般的手机上进行了应用。该算法框架也是当前主流的视觉 SLAM 算法框架。2012 年，Hauke Strasdat, J.M.M. Montiel, Andrew J. Davison et al.[27]就视觉 SLAM 中基于图优化和滤波的 SLAM 技术进行了比较，得出了在相同的计算量下基于图优化的 SLAM 方法能获取到更高的精度。在此之后，在视觉 SLAM 领域掀起了基于图优化的热潮。当前，主流视觉 SLAM 算法都采用基于图优化的方法，也是 SLAM 研究者研究的热点方向。

基于关键桢的视觉 SLAM 早期采用基于图像特征点的方式构建系统[12,14,15,28]。对于每桢图像，首先选取图像中的关键点作为环境路标点，并且计算每个关键点的对应特征描述子作为后期图像关键点匹配的基础，然后使用关键点匹配计算两桢图像之间的位姿变换矩阵。该方法类似于计算机视觉里的 BA（Bundle adjustment[29]）算法。虽然基于图像特征点的方式能在一定程度上消除图像本身给系统带来的干扰，但是提取图像特征点需要耗费大量计算资源，系统的实时性较难保证。所以最近几年研究者提出了直接使用图像像素点构建系统，基于图像灰度不变的假设计算图像桢间的位姿变换矩阵。该方法去除了提取图像特征点的环节，实时性上得到了提高，但是系统易受图像噪声的影响，特别是光照给图像带来的影响[30]。当前，前面一种方法被研究者称为间接法，后面一种方法被称为直接法。间接法使用了图像的特征点来构建系统，所以基于不同的图像特征点也就衍生出了多种 SLAM 算法[14,15,31,32,33]。其中 2014 年在文献[14]中，Rau Mur-Artal 等人基于图像二进制 orb 特征建立的 ORB-SLAM 是当下最流行的间接法 SLAM 解决方案。直接法也由图像像素点使用的稠密程度被分类为稠密法[30,34]、半稠密法[35,36]、稀疏法[16,17]。其中 2016 年在文献[17]中，Jakob Engel 等人基于稀疏直接法

的 DSO（Direct Sparse Odometry）算法是当下最流行的直接法 SLAM 解决方案。直接法和间接法各有各的优缺点，两种算法框架都是研究者研究的热点。

在国内，在 SLAM 领域近年来才逐渐开展研究，主要集中在以下四个方面：（1）如何改进算法提高实时性，以及如何设计不变性更强且效率更高的图像特征点[37,38]；（2）在图像特征点匹配中对于数据关联部分的改进[39]；（3）在基于滤波技术的 SLAM 算法中对卡尔曼滤波以及粒子滤波的改进[40,41]；（4）多机器人协作 SLAM 算法的研究[42]。总的来说，国内研究主要以跟踪国外技术为主，国内在这方面的研究与国外还有不少的差距。

1.2.2 基于深度学习的 SLAM 研究现状

随着深度学习的兴起，其在传统的多个领域中都取得了惊人的表现。深度学习在视觉 SLAM 领域的应用也成为了研究者研究的热点，而且已经获得了一些成果。当前，研究者主要在场景深度与运动估计[43-48]、闭环检测[49-52]、语义 SLAM[53-55]等方向进行研究。

如何从单张图像中恢复场景深度也是计算机视觉领域研究的热点。早期传统的方法多采用人工的图像特征来获取场景的深度，其对于场景的几何特性一般要做出很强的假设[56]。最近出现了很多采用深度学习来完成该任务的研究，其都取得了比采用传统方法更好的效果。视觉里程计是视觉 SLAM 算法中的核心组成部分，一般多采用图像特征点匹配、对极几何、ICP（Iterative Closest Points[57]）等算法来计算位姿转换矩阵，而深度学习将其转化为了一个端到端的学习问题。当前，主流的算法将场景深度恢复和视觉里程计结合成一个整体一起学习，学习得到的场景深度和视觉里程计信息相互监督解决了标定训练数据困难的问题。

闭环检测是指检测机器人是否之前到过某个地方，类似于场景识别。在 SLAM 系统中，如果闭环检测成功，那么可以根据闭环的结果对系统中的状态量进行修正更新，从而大大减少积累误差。闭环检测是 SLAM 技术在大规模场景应用中，消除积累误差十分有效的一种方式，其也是 SLAM 问题研究中的核心点。闭环检测中多采用 BOW（Bog of Words[58]）算法对图像进行量化比较，从而判断是否产生闭环[60]。当前，深度学习在闭环检测上的研究主要集中在两个方面：（1）如何基于深度学习提取图像的特征进行闭环检测；（2）如何直接将其转化为学习问题。基于深度学习的闭环检测当前获得了很好的精度。实际当中由于对于实时性的要求，基于深度学习的在线学习方法将是 SLAM 闭环检测问题很好的解决方向。

传统 SLAM 技术主要实现了智能体对环境的几何信息的理解，但是缺乏对环境语义信息的理解。语义 SLAM 是指不仅构建环境地图实体，还对环境中的物体进行识别，融入物体的语义信息。早期的语义 SLAM 多采用形状比较的方式，而随着深度学习在场景分割和物体识别上取得的发展，基于深度学习建立语义 SLAM

系统成为了一个几乎最好的选择。

在 2016 年 ICCV 研讨会上就 SLAM 的研究方向进行了探讨。其中主要包括以下三点：（1）如何进一步提高 SLAM 的实时性和鲁棒性；（2）传统 SLAM 与深度学习如何更好的融合；（3）语义 SLAM。当前，深度学习应用到传统 SLAM 中去已经成为了国际学者研究的热点方向。在文献[55]中，CAMP-TU Munich 研究使用深度学习获取的场景深度融入到 LSD-SLAM 中去，显著的改善了单目视觉 SLAM 在纯旋转运动下建图和定位的精度。在文献[54]中，Xuanpeng LI 将深度学习获取的语义信息融入 SLAM 算法，最终生成了包含语义信息的语义地图。基于深度学习的 SLAM 技术的研究对于提高算法的运行效率及其实际应用中必需的鲁棒性都将具有十分重要的推动作用。

1.3 SLAM 技术应用领域

目前，SLAM 技术主要被运用于无人机、无人驾驶、移动机器人、VR/AR 等领域，其被研究者认为是这些当前新兴的热门应用的关键技术之一[60]。

① 无人机

对于无人机的研究主要集中在无人机的长续航、无人机的定位、物体追踪、智能避障、路径规划等方面。无人机的定位过程就是对 SLAM 技术的成功应用，其也可以帮助构建局部地图，辅助无人机进行自主避障、规划路径。美国“全球鹰”无人机（如图 1.1 所示）、国内中航集团研制的“翔龙”无人机（如图 1.2 所示）、大疆机器人研发的娱乐无人机“精灵 PHANTOM4 PRO”（如图 1.3 所示）及其研发的农业植保机 MG-1P 系列（如图 1.4 所示）。

图 1.1 美国全球鹰无人机

Fig1.1 Global Hawk

图 1.2 中航“翔龙”无人机

Fig1.2 AVIC Xianglong UAV

图 1.3 大疆“精灵 PHANTOM4 PRO”

Fig1.3 phantom4 pro

图 1.4 大疆农业植保机 MG-1P 系列

Fig1.4 Fog Machine MG-1P

② 无人驾驶

无人驾驶是近年来很火的话题之一，Google、Uber、百度等企业都在加速研发无人驾驶相关技术，抢占先机，同时一些传统车企也将无人机驾驶作为企业未来发展的重要方向。随着人工智能，物联网技术的发展，无人驾驶必是出行的大势所趋。无人驾驶的核心技术就是定位和环境地图的构建，以及后面的规避障碍、路径规划。SLAM 在无人驾驶上的应用一般采用高精度的雷达，同时多传感器数据的融合也是无人驾驶技术研究的热点。Google Driverless Car（如图 1.5 所示）由谷歌公司的 Google X 实验室研发，目前正在测试，已驾驶了 48 万公里。在 2016 年，Google 开源了激光雷达 SLAM 算法 Cartographer。2017 年百度的无人车 Apollo（如图 1.6 所示）正式上路进入公众视野。

图 1.5 Google 无人车

Fig1.5 Google driverless car

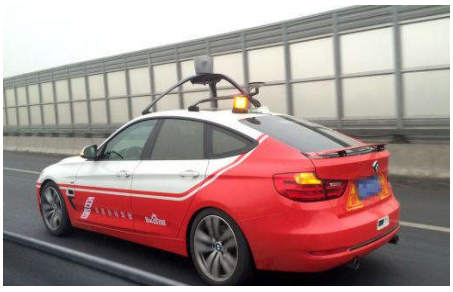


图 1.6 百度无人车 Apollo

Fig1.6 Baidu Apollo

③ 移动机器人

SLAM 最早被提出就是为了解决移动机器人的定位和地图构建问题，其在移动机器人中的应用也最普遍。在各行各业的移动机器人领域，基本上都应用到了 SLAM 技术。图 1.7 中，美国发射的“机遇号”探测器正在火星执行火车探测任务。图 1.8 中，浙江国自机器人技术有限公司研究的变电站巡检机器人正在电站进行日常巡检。图 1.9 中，中国安防技术有限公司开发的 SPR 安保机器人正在为人们保

驾护航。图 1.10 中，iRobot 公司设计的扫地机器人正在为主人打扫房间。

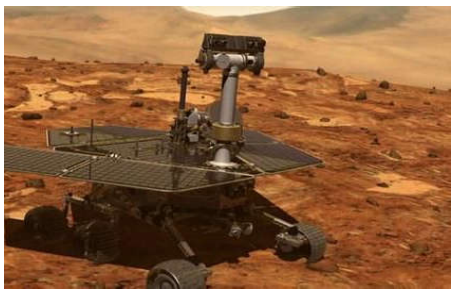


图 1.7 “机遇号”火星探测器

Fig1.7 Mars probe Opportunity

图 1.8 变电站巡检机器人

Fig1.8 Substation inspection robot

图 1.9 SPR 安保机器人

Fig1.9 Security robot SPR

图 1.10 iRobot 扫地机器人

Fig1.10 iRobot Sweeping robot

④ VR/AR

VR/AR 研究的是如何实现真实世界和虚拟世界之间的无缝连接，而 SLAM 正是连接虚实世界的一项技术。SLAM 之于 VR/AR 的重要性在于只有知道眼镜的空间坐标和相对于室内各种障碍物的位置，同时识别出各种室内摆件的形状，才能实现人机交互的各种应用。目前基于 SLAM 技术开发的代表性 VR/AR 产品有微软的 Hololens（如图 1.11 所示）和谷歌的 Project Tango（如图 1.12 所示）。

图 1.11 Google 的 Project Tango 平板

Fig1.11 Google Project Tango tablet

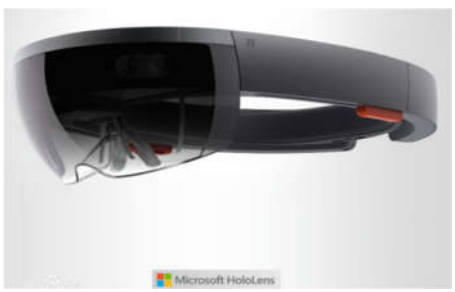


图 1.12 微软 MR 眼镜 Hololens

Fig1.12 Microsoft Hololens

1.4 论文主要研究内容及组织结构

当前，视觉 SLAM 技术被广泛的应用于各行各业，同时各应用对于优良的单目视觉 SLAM 技术的需求也越来越高，但是单目视觉 SLAM 技术还存在很多问题。基于深度学习的视觉 SLAM 取得了一些成果，本文旨在研究融合传统视觉 SLAM 和场景深度预测卷积网络。通过融合预测网络对场景深度的预测信息提升单目视觉 SLAM 算法在如下关键难题中的表现：（1）目前单目视觉 SLAM 无法解决的场景初始尺度的确定；（2）单目视觉 SLAM 运行中场景尺度的漂移。最终综合提升单目视觉 SLAM 在复杂场景下建图和定位的精度。本文设计的该算法融入了深度学习，使得算法具有了学习性，所以本文将其称为 Learning SLAM，简称 LSLAM。

LSLAM 算法框架如图 1.13 所示。场景深度预测网络选择了文献[46]中设计的 FCRN 网络，传统单目视觉 SLAM 框架选择了直接法 DSO 算法。整个系统的输入为场景的 RGB 图像桢。系统初始化时采用深度网络预测的场景深度作为初始地图点深度初值进入迭代优化，初始化完毕后依赖对场景的网络预测深度的统计对初始场景进行尺度化；在算法前端运动估计部分，融合网络预测的场景深度与成熟地图点的投影深度提升运动估计的准确性，减小系统运行中尺度的漂移；同时在深度追踪部分引入网络预测场景深度初始化关键桢，提升关键桢中未成熟地图点的追踪精度和效率。整个 SLAM 系统在运行的鲁棒性上有了显著的提升。最后，本文基于机器人操作系统 ROS 完成了整个系统的软件设计，并在实验室机器人“甲鲁普”上进行了实现。

论文的章节内容安排如下：

第一章 “绪论”阐述了视觉 SLAM 研究的背景和意义，指出了视觉 SLAM 发展的瓶颈及研究方向，并引出了本文设计的目的和方法。在此之上，对传统 SLAM 和基于深度学习的 SLAM 的研究现状分别进行了概述，然后对 SLAM 在无人机、无人驾驶、移动机器人、VR/AR 等领域的应用进行了举例说明。最后针对单目视觉 SLAM 中存在的问题，提出了融合场景深度预测网络与传统视觉 SLAM 算法的方案。

第二章 “单目视觉 SLAM 算法”对传统视觉 SLAM 进行了原理的剖析，建立起了其数学概率模型。并在此基础上，对单目视觉 SLAM 的三个核心部分：场景特征提取与匹配、视觉里程计、地图优化分别从间接法和直接法的角度进行了阐述。

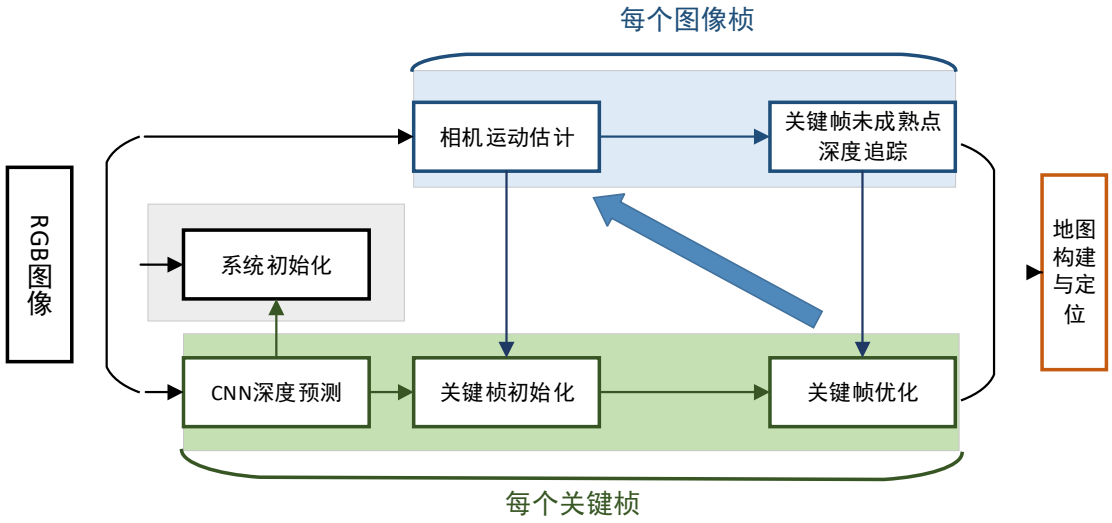


图 1.13 LSLAM 算法框架

Fig1.13 LSLAM algorithm framework

第三章“基于深度学习的场景深度预测”，首先对深度学习中最重要的网络结构卷积神经网络从原理上进行了说明，这里主要从其生物学依据以及其实际的应用结构和特点两方面进行了阐述。本章然后介绍了深度网络应用到场景深度预测的发展及当前流行的网络结构设计，对其的阐述主要分成了两个阶段：（1）单张图像深度恢复；（2）图像对深度恢复。

第四章“LSLAM 算法的设计与实现”主要对本文设计的 SLAM 算法框架中的关键部分的实际实现细节进行了详细说明。首先阐述了如何选择出最适合本文设计的深度预测网络，然后叙述了如何引入场景深度预测网络进入算法的初始化，继而介绍了融合场景深度预测网络进行相机运动估计的方法，最后对算法中关键桢的初始化及其优化进行了说明。

第五章“实验结果与分析”首先对实验的硬件平台机器人“甲鲁普”和软件平台机器人操作系统 ROS（Robot Operating System）分别进行了说明，然后设计实验对算法的性能进行了测试，并将实验结果与传统 DSO SLAM、RGB-D SLAM 算法进行了比较。实验主要针对机器人的定位精度，通过对比验证了本文提出的基于优化框架融合场景深度预测网络与传统 DSO SLAM 的 LSLAM 算法的优越性。

第六章“总结与展望”对本文设计的 LSLAM 算法进行了总结，并对算法中还存在的不足，以及未来需要进一步开展的工作进行了说明。

2　单目视觉 SLAM 算法

2.1 引言

早期 SLAM 技术多采用激光、雷达作为系统的输入媒介。当前基于激光 SLAM 的方法的定位精度也最高，但是，高精度的激光雷达成本高。反观视觉传感器不仅价格低廉，而且能获取到环境中更丰富的信息，视觉 SLAM 技术的发展具有更高的社会需求，所以视觉 SLAM 成为了当前 SLAM 研究的热点。单目 SLAM 更是因其硬件的简易性、易推广性受到了国际研究者的青睐，本文正是基于这些原因对单目视觉 SLAM 算法进行了研究。本章将首先叙述 SLAM 问题的数学模型，然后重点阐述单目视觉 SLAM 中的经典算法流程框架及其各个核心环节。

2.2 SLAM 问题的数学模型

SLAM 问题可从概率学的角度进行分析，分为定位与建图两个部分。定位与建图两部分相互依赖，交替进行互为条件，寻求条件概率的最大化。SLAM 问题从实质上来说就是一个迭代的状态估计问题[19]。机器人定位一般可以依靠机器人的运动模型进行估计，但是由于运动模型中存在误差，所以需要通过获取环境数据对其进行修正。在定位的基础上，机器人将对环境地图进行构建。

在 SLAM 数学模型[61]中，我们假设系统 k 时刻的状态向量为 $x(k)$，在噪声的影响下，其运动模型如式 2.1 所示：

$$x(k)=f(x(k-1),u(k))+\omega(k) \tag{2.1}$$

其中，$x(k)$ 包含了机器人当前的位姿及所有环境中的路标信息，$u(k)$ 为系统 $k-1$ 时刻的输入量，使系统达到状态 $x(k)$，$\omega(k)$ 为零均值高斯白噪声，这里假设其包含了机器人运动中的各类误差。在运动中，机器人在 k 时刻的观测量这里设为 $z(k)$，观测模型用于表达机器人的观测量与状态向量之间的关系，其数学表示如式 2.2 所示：

$$z(k)=h(x(k))+\xi(k) \tag{2.2}$$

其中 $\xi(k)$ 为观测量噪声，本文假定其为零均值高斯白噪声，表示传感器测量噪声以及观测模型本身的误差。

从 SLAM 问题的数学模型中，我们看出运动中这种增量式的迭代过程会产生积累误差，而如何消除积累误差就是 SLAM 技术要解决的最主要的原始问题。当前各种 SLAM 技术方案不管是基于滤波的还是基于图优化的其最基本的概率学理论依据就是贝叶斯原理。从贝叶斯的角度讲，SLAM 问题就是最大似然估计[61]，SLAM 过程可用如下数学表示式表示，运动模型如式 2.3 所示：

$$p(x(k)\,|\,x(k-1),u(k)) \tag{2.3}$$

观测模型如式 2.4 所示：

$$p(z(k)\,|\,x(k)) \tag{2.4}$$

运动更新如式 2.5 所示：

$$\begin{aligned} &p(x(k)\,|\,z_{0:k-1},u_{0:k},x_0)= \\ &\int p(x(k)\,|\,x(k-1),u(k))*p(x(k-1)\,|\,z_{0:k-1},u_{0:k-1},x_0)dx(k-1) \end{aligned} \tag{2.5}$$

测量更新如式 2.6 所示：

$$\begin{aligned} &p(x(k)\,|\,z_{0:k},u_{0:k},x_0)= \\ &\frac{p(z(k)\,|\,x(k))*p(x(k)\,|\,z_{0:k-1},u_{0:k},x_0)}{p(z(k)\,|\,z_{0:k-1},u_{0:k})} \end{aligned} \tag{2.6}$$

其中$z_{0:k}$表示从 0 到k时刻机器人的观测量，$u_{0:k}=\{u(1),u(2),\ldots u(k)\}$表示 0 到$k-1$时刻机器人的输入量。运动更新主要获取对系统状态向量的先验估计，然后在测量更新中结合新的观测值得到系统状态向量完整的后验估计。图 2.1 对 SLAM 过程中利用贝叶斯原理来估计地图和位姿进行了展现。对于纯视觉的基于关键帧的 SLAM 技术，前端采用追踪技术获取系统状态向量的初始值，然后在后端采用图优化最小化误差对状态变量进行更新，其实就是最大化贝叶斯网络里的条件概率 $p(z(k)\,|\,x(k))$。

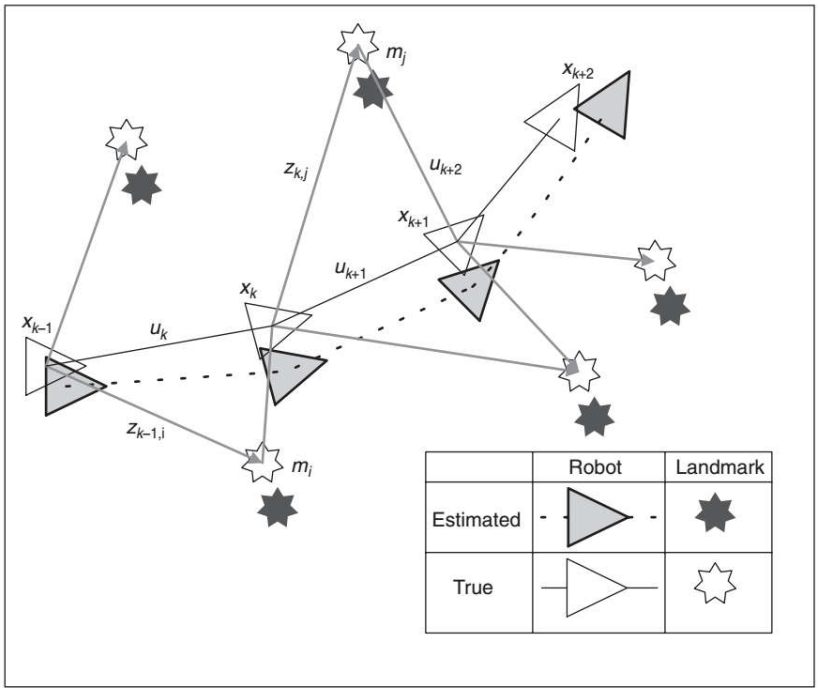


图 2.1 SLAM 问题的贝叶斯网络模型

Fig2.1 The Bias model of SLAM problem

2.3 单目视觉 SLAM 算法总体框架

单目视觉 SLAM 算法一般采用纯视觉，这有利于算法的应用。主流的单目视觉 SLAM 主要分为前端图像和后端优化两部分：前端主要包括特征提取与匹配、视觉里程计、闭环检测等，后端主要包括直接法里的深度追踪或是间接法里的位姿优化、地图优化等环节。其总体框架如图 2.2 所示：

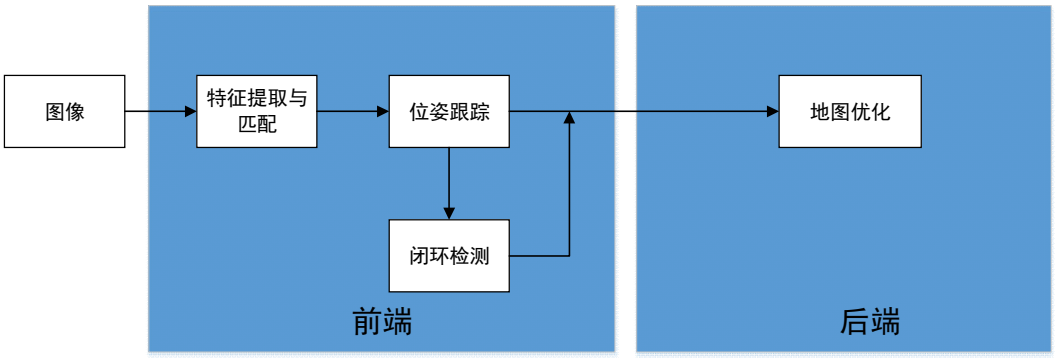


图 2.2 单目视觉 SLAM 算法流程图

Fig2.2 Monocular SLAM flowchart

每当系统获取到一桢图像，首先提取图像中的特征点，计算每个特征点的描述子，然后通过与前一桢图像中的特征点进行特征匹配建立起特征点之间的一一对应关系。在直接法中，直接基于灰度不变的假设建立起两桢图像像素点之间的关联。视觉里程计在特征匹配或灰度不变假设建立的约束下，通过对极几何求解两桢图像之间相机的位姿变换。地图优化利用新引入的图像桢信息对系统状态向量进行更新。本文这里的地图优化是广义的地图优化，其包含了间接法里的纯位姿优化和直接法里的地图点深度追踪。

2.4 特征提取与匹配

2.4.1 特征提取

良好的特征提取算法对于高精度的数据匹配具有十分重要的意义，而高精度的数据匹配几乎决定了整个 SLAM 系统的性能表现。特征提取包括图像关键点的检测和描述两个方面。关键点一般是图像局部中具有某种特殊性的点，其描述子就是这种特殊性的表达。对于一个好的特征提取应具备以下几个特征[62]：

① 不变性：特征点的检测和描述需要具有对图像平移、旋转，以及外部光照条件等的不变性；

② 区分性：不同特征点的描述子需要具有高度的差异性以易于区分和检测；

③ 数量多：在图像中检测出的特征点应该足够丰富，便于为匹配提供更多的

信息；

④ 高效性：特征点的提取需要考虑其运行的效率，特征点提取算法时间复杂度不能过高；

当前在单目视觉 SLAM 系统中最常用的特征点提取算法为 SIFT[63]、SURF[64]、ORB[65]特征。SIFT 特征是物体上的一些局部外观的兴趣点，其具有旋转、平移、尺度、一定程度仿射等不变性外,具有极高的区分性,特征匹配的正确率非常高。虽然 SIFT 特征具有很好的特性，但是其计算较复杂，提取速度慢，SURF 是对 SIFT 的改进算法。SURF 标准版本其特征点的数量减少了，但是提取速度比 SIFT 要快数倍。ORB 特征是 Ethan 等人 2011 年提出的一种性能出色的二进制特征提取算法。视觉 SLAM 正确的特征匹配对算法后端优化收敛具有决定性的作用，所以当前无疑 SIFT 特征是最适合间接法视觉 SLAM 应用的特征点之一。下文也将对图像 SIFT 特征的提取做详细的介绍。

SIFT 算法的本质是在不同尺度空间上查找一些十分突出的关键点，并计算其方向。这些关键点对光照，仿射变换和噪音等因素具有很强的鲁棒性，其中例如角点、边缘点、暗亮区域里的孤立点等。SIFT 算法是 Lowe 在 1999 年提出，其将 SIFT 算法分解为如下 4 步[63]：

1）尺度空间极值检测：为了获取图像在各个尺度上都稳定的关键点，首先采用高斯核构建图像的尺度空间，一个图像的尺寸空间 $L(x,y,\sigma)$ 定义为原始图像 $I(x,y)$ 与一个可变尺寸的二维高斯函数 $G(x,y,\sigma)$ 进行卷积运算。该过程如下式所示：

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \tag{2.7}$$

其中，$*$ 表示进行卷积运算，二维空间高斯函数数学表达式如下式所示：

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2} \tag{2.8}$$

其中 σ 为尺度因子，决定图像的平滑程度，σ 越小，图像就越清晰，σ 越大，图像就越模糊。

在此基础上，通过使用图像高斯金字塔构建图像的多尺度空间。图像高斯金字塔的建立可拆分为两部分：（1）采用不同尺度对每组图像做高斯模糊；（2）对图像进行降采样形成新的一组图像。其中，降采样时，高斯金字塔上一组图像的最底层图像由前一组图像隔点采样得到，这里选取上一组的倒数第三张图像。图像高斯金字塔如图 2.3 所示。

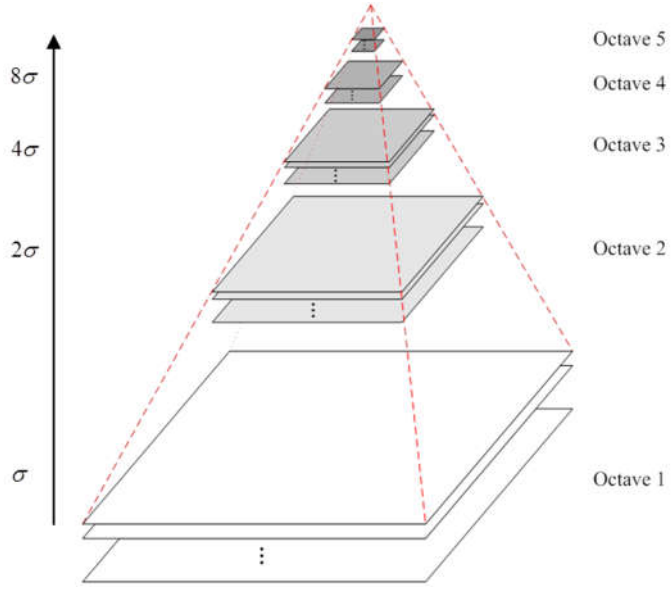


图 2.3 图像高斯金字塔

Fig2.3 Image Gauss Pyramid

尺度空间极值检测通过寻找高斯差分尺度空间中的极值点获取。高斯差分方程简称 DOG 算子，其与归一化的高斯拉普拉斯函数非常近似，能够产生最稳定的图像特征。图像高斯差分尺度空间可从下式得出：

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \tag{2.9}$$

在算法实现中采用图像高斯金字塔中每组图像中相邻图像相减得到图像的高斯差分尺度金字塔，如图 2.4 所示。在寻找尺度空间中的极值点时，每一个图像点都要和它邻域的所有 26 个点（本尺度空间 8 个点，上下相邻尺度空间各 9 个点）相比较，如果此点为最大值或最小值点，该点即为图像在该尺度初步探查的特征点。

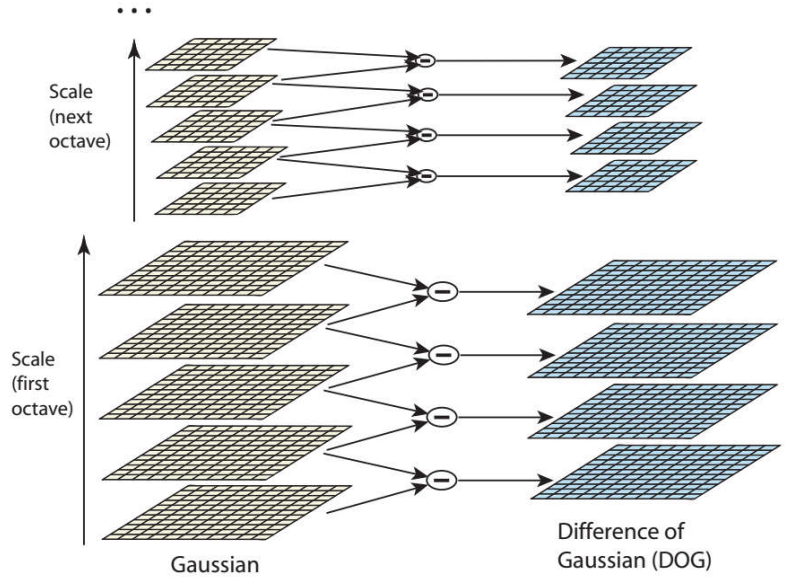


图 2.4 图像高斯差分金字塔

Fig2.4 Image DOG Pyramid

2）关键点精确定位：通过以上的方法可以发现候选关键点，下一步就是与附近的数据点进行拟合来精确确定关键点的位置和尺度，同时去除低对比度或是边缘附近的点（DOG 算子会产生较强的边缘响应）。这里通过利用三维二次函数与本地样本点对尺度空间 DOG 函数进行曲线拟合。

$D(x,y,\sigma)$ 的泰勒展开式如下式所示：

$$D(x)=D+\frac{\partial D^T}{\partial x}x+\frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x \tag{2.10}$$

其中，$x=(x,y,\sigma)^T$，这里令 $D'(x)=0$ 得到极值点的偏移量 $\hat{x}$：

$$\hat{x}=-\frac{\partial D^T}{\partial x}(\frac{\partial^2 D}{\partial x^2})^{-1} \tag{2.11}$$

加入偏移量就是对应的极值点，极值点方程的值为：

$$D(\hat{x})=D+\frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x} \tag{2.12}$$

其中，$\hat{x}=(x,y,\sigma)$，如果其在任一维度上的偏移量大于 0.5 时（即 x 或 y 或 σ ），表示该极值点已经偏移到它的邻近点上了，所以必然需要纠正该极值点的位置。同时，当 $|D(\hat{x})|\le 0.03$ 时，该候选极值点对噪声敏感，会因为干扰而不稳定，所以应当去除。DOG 算子需要去除不稳定的边缘响应点。首先获取特征点处的 Hessian 矩阵如下式所示：

$$H=\begin{bmatrix}D_{xx} & D_{xy}\\ D_{xy} & D_{yy}\end{bmatrix} \tag{2.13}$$

若$\frac{Tr(H)^2}{Det(H)}<\frac{(r+1)^2}{r}$，则将该关键点保留，否则删除，$r$为一阈值，一般取 10。

3）关键点方向指定：在精确定位关键点后，根据关键点的尺度值得到最接近这一尺度值的高斯图像。计算以$3\times1.5\sigma$为半径的圆形范围内图像梯度的幅角和幅值，公式如下：

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \quad (2.14)$$

$$\theta(x,y)=\arctan\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \quad (2.15)$$

在完成计算后，邻域内像素的梯度和方向通过使用直方图进行统计。在梯度直方图中，0~360 度的方向范围被均分为 36 个柱(bins)，也就是每柱 10 度。关键点的主方向由直方图的峰值方向决定。

4）关键点描述子：通过以上步骤，每一个关键点获取到了三个信息：位置、尺度以及方向。接下来就是用一组向量将这个关键点描述出来，也就是该关键点的描述子。这个描述子不仅容纳了关键点的信息，也包含了周围对其有贡献作用的像素点。首先将关键点附近划分成$d\times d$个子区域，每个子区域包含$m\sigma$个像元（$d=4$，$m=3$）。为了保证特征矢量具有旋转不变性，需要对区域旋转到特征点的方向。然后，在每个子区域内，统计 8 个方向的梯度直方图的累加值，得到一个种子点。这样共形成 16 个种子点，所以最终得到一个 128 维的向量作为该关键点的 SIFT 特征描述子。

对于直接法 SLAM 中，一般稠密法直接选择所有或是像素点一阶导数比较大的像素点作为场景的特征点。在稀疏法中，考虑特征点分布的均匀，一般将图像平均划分为单元格，在每个单元格里寻找一阶导数比较大的像素点作为场景的特征点。

从上面的介绍中可以看出，间接法特征点提取过程较复杂，虽然特征点在抗干扰能力上有了一定提升，但是最后算法实时性有所损失。这也是本文选取直接法作为 LSLAM 算法的基础框架的原因之一。同时，也可以看出直接法选择特征点的条件相比于间接法更宽松，场景中特征点更多，这有利于系统在特征点少的场景中运行。场景深度预测网络整体预测精度较高，但是对于场景细节的预测比较粗糙，所以特征点多也能更充分的发挥预测网络对场景深度的预测作用。

2.4.2 特征匹配

特征匹配即是将两桢图像中的关键点进行一一对应，建立约束，是后期视觉里程计和地图优化的基础。在间接法中，好的特征匹配即要保证匹配的准确率，又要保证其匹配的效率。在准确率上，SIFT 特征能保证较高的准确率。基本的匹配方法采用穷举法，即是一一对特征点的特征向量进行比对，通过计算两特征向

量之间的距离来衡量两者的相似程度。常用的距离有欧式距离、汉明距离和马式距离等。这里假设两个特征点的描述子为：

$$D_1 = (x_1, x_2, \dots, x_n) \tag{2.16}$$

$$D_2 = (y_1, y_2, \dots, y_n) \tag{2.17}$$

① 欧式距离

欧式距离是特征匹配中最常用的距离度量方法，其表示的是两个向量在n维空间中的距离，其数学表示为：

$$\begin{aligned} d(D_1, D_2) &= \sqrt{(D_1 - D_2)(D_1 - D_2)^T} \\ &= \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \end{aligned} \tag{2.18}$$

② 汉明距离

汉明距离是指在两个相同维数的向量中，两向量对应位置对应元素不同的数量，一般应用于二进制的图像特征点（例如 ORB 特征）。该距离计算采用异或运算，计算效率高，其数学表示为：

$$d(D_1, D_2) = \sum_{i=1}^{n} x_i \oplus y_i \tag{2.19}$$

③ 马式距离

马式距离在概率统计计算中最常用，在视觉 SLAM 特征匹配中也有该距离的应用，其表示的是向量间的协方差距离，其数学表示为：

$$d(D_1, D_2) = \sqrt{(D_1 - D_2)S^{-1}(D_1 - D_2)^T} \tag{2.20}$$

其中，S为协方差矩阵。

在特征匹配中，虽然穷举法简单，但是时间复杂度高，所以在视觉 SLAM 中一般采用高维的数据结构加速搜索过程，例如 KD-TREE[66]搜索算法。在 KD-TREE 算法中，首先利用每一桢中的所有特征点构建 KD-TREE，然后在数据比对时在 KD-TREE 中搜索与其距离最近的 KD-TREE 节点。KD-TREE 搜索的理论时间复杂度为$O(\log n)$，而穷举法的时间复杂度为$O(n)$，在实际应用中，KD-TREE 搜索比穷举法具有更高的匹配效率。

在直接法中，特征匹配基于场景灰度不变假设建立场景特征点之间的约束。为了提高匹配的准确性，一般将特征点周围的几个点作为一个整体，该整体基于灰度不变假设进行匹配。在实际应用中，其实现一般转化为一个图优化问题，将在后面章节做详细介绍。

2.5 视觉里程计

视觉里程计即是计算图像序列相邻两桢图像相机位姿的变换矩阵，从而恢复相机运动，实现位姿跟踪。对于单目视觉 SLAM 间接法和直接法的区别主要就在于视觉里程计采用的方法的不同。视觉里程计恢复的相机运动作为后面地图优化的初始状态，其对于优化的收敛有极其重要的作用。

2.5.1 摄像机模型

视觉 SLAM 利用单目摄像头获取场景的图像信息，由此来感知、识别机器人周围的环境特征。相机的模型就是建立三维地图点与图像中的二维像素点的对应关系。在视觉 SLAM 中，常用的摄像机有单目摄像头、双目摄像头、深度摄像头、全景摄像头等。本文研究的是单目视觉 SLAM，使用单目摄像头，下面就单目摄像头中最简单的针孔模型进行介绍，如图 2.5 所示。

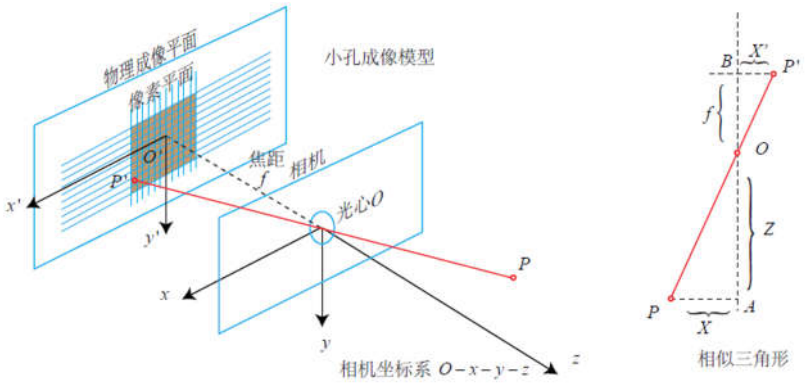


图 2.5 相机针孔模型

Fig2.5 The pinhole model of camera

相机针孔模型就是小孔成像模型，三维空间物理点从相机坐标系到成像平面的数学表达式如下式所示：

$$\begin{aligned} x' &= f_x \frac{x}{z} \\ y' &= f_y \frac{y}{z} \end{aligned} \tag{2.21}$$

其中 f_x, f_y 为相机在 x, y 方向上的焦距。

成像平面转换到相机像素平面的数学表示式如下式所示：

$$\begin{aligned} u &= x' + c_x = f_x \frac{x}{z} + c_x \\ v &= y' + c_y = f_y \frac{y}{z} + c_y \end{aligned} \tag{2.22}$$

其中 c_x, y_y 为相机光轴与成像平面的交点，称为基准点。 f_x, f_y, c_x, c_y 称为相机

的内参。转化为矩阵形式如下式所示：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z}\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} \triangleq \frac{1}{z}KP \tag{2.23}$$

三维空间物理点从世界坐标转化到相机平面的数学表式如下：

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix}\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} = T_{cw}P_w \tag{2.24}$$

其中，T_{cw}表示相机中心到世界坐标的转换矩阵，称为相机的外参，包含了两部分：表示旋转的旋转矩阵和表示平移的平移向量。最后整理得出：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z}\begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2.25}$$

2.5.2 运动姿态估计

运动姿态优化通过最小化误差来调整系统状态向量，但是该优化函数一般存在很多的局部最优点，所以为了获取好的优化结果甚至是收敛，系统状态向量需要有一个好的初始值。对于运动姿态的估计有一个从粗到细的过程。间接法和直接法对于运动姿态的估计采用不同的方法，下面将分别对其进行介绍：

① 间接法：

在间接法中，初始姿态的估计依赖特征匹配和对极几何建立变换关系。对极几何指对于同一物体，其在两幅图像中投影点的相对位置存在一定的约束关系，这种约束关系在立体视觉中被称为对极几何[67]，如图 2.6 所示。

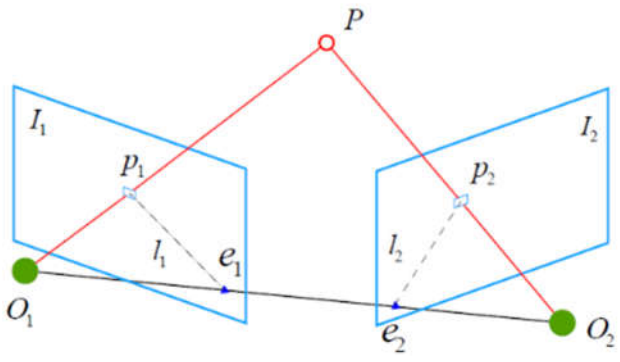


图 2.6 对极几何

Fig2.6 Polar geometry

其中，O_1,O_2 表示相机的光心，I_1,I_2 表示相机的两个成像平面，直线 O_1O_2 称为基线，直线 l 称为对极线，对极几何描述的是以基线为轴的平面簇与两成像平面相交的几何关系。P 在两个成像平面上的坐标满足以下几何约束：

$$P_1^T E_{12} P_2 = 0 \tag{2.26}$$

其中，$E_{12} = R_{12} t_{21}^{\wedge}$ 称为本质矩阵，加入相机内参则可得到：

$$p_1^{'T} F_{12} p_2^{'} = 0 \tag{2.27}$$

其中，$p_1^{'}, p_2^{'}$ 分别表示 P 在两个像素平面上的像素坐标，$F_{12} = K^{-1^T} E_{12} K^{-1}$ 称为基础矩阵。

对极几何中的本质矩阵最少可通过 5 个匹配的特征点唯一确定，然后对本质矩阵进行奇异值分解可得出摄像头在两帧图像间的运动参数：旋转矩阵 R 和平移向量 t。但是，从中也可以看出对极几何约束，存在一个没有限制的自由度，即等式两边乘以一个倍数等式仍然成立，所以采用对极几何无法估计场景尺度[68]。这也是单目视觉 SLAM 在初始化时无法估计场景尺度因子的根本原因。

从上面可知，最少 5 个特征匹配点就可求出相机运动参数，但是一般图像中匹配的特征点含有几百个，而且其中含有错误的匹配，所以需要选择出最合理的模型。一般采用随机抽样一致算法（RANSAC[68]）估计模型的参数。RANSAC 算法可以通过迭代的方式利用数据估计模型的合理参数，其数据中可以允许包含一些“局外点”。RANSAC 算法的一般流程为：（1）随机选取几个数据点计算出模型的参数；（2）利用其它的数据点对模型进行验证；（3）如果模型容纳的数据点超过一定阈值则说明估计的模型参数合理，反之，则另取数据点计算模型参数，再进行模型验证。在视觉 SLAM 中采用 RANSAC 算法能较好的去除错误的匹配点，获得良好的相机运动参数。

通过以上方法可得到间接法中系统初始化中相机初始运动姿态的估计，然后利用其获取初始地图点的深度信息建立初始地图。在后面的运动姿态估计中，一般利用所有的特征匹配点及当前已知的地图信息，基于匹配点对间的重投影误差将该问题转化为一个优化问题，这里一般常采用迭代最近点算法 ICP。ICP 算法通过对式 2.28 的目标函数求优化来求解位姿运动参数 R 和 t：

$$\min_{R,t} \frac{1}{N} \sum_{i=1}^{N} \left\| q_i - K(RK^{-1}(p_i, d_p) + t) \right\|^2 \tag{2.28}$$

其中，N 表示特征匹配点对的总数，q_i, p_i 分别表示第 i 对匹配点在两个图像桢中的像素坐标。d_p 为该特征点在关键桢中的深度值，在获取运动姿态的条件下通过三角化进行恢复。

② 直接法：

在直接法 SLAM 中，初始化和后续的相机运动估计都采用类似的操作，即采用图像金字塔逐层优化和运动假设模型估计机器人当前相对于参考关键桢的运动。运动假设模型假设的是当前桢与上一桢之间的位姿转换，然后累加上一步求取的图像桢到参考关键桢的位姿转换矩阵，求出当前桢相对于参考关键桢的位姿转换矩阵的最初估计[17]。参考关键桢的构建主要是其中参考地图点的构建，其利用当前优化窗口中的所有关键桢中的特征点向参考关键帧进行投影获得。运动假设模型一般包含三个方向：（1）机器人做匀速运动；（2）机器人保持不动；（3）机器人做纯旋转运动。基于这些假设，利用图像金字塔逐层优化对假设进行优化和验证。

图像金字塔（如图 2.2 所示）这里直接采用对相机采集的原始图像降采样的方法获取，金字塔最底层是原图像，每升高一层的图像都是由其下一层图像降采样得到的，最高层是分辨率最低的图像。在对灰度进行降采样时，对深度值也进行降采样。基于图像金字塔由上往下对运动姿态进行估计。运动姿态估计自始自终优化的函数如下式所示：

$$E=\sum_{p\in N_p}\omega_p\left\|I_j[p']-I_i[p]\right\|_\gamma \tag{2.30}$$

其中，N_p 表示在优化窗口中所有关键桢中的成熟关键点投影后在参考关键桢中的关键点集合；$\|\cdot\|_\gamma$ 表示 Huber 正则化，如下式所示：

$$\|a\|_\gamma=\begin{cases}\dfrac{1}{2}a^2, & for\,|a|\le\gamma\\ \gamma.(|a|-\dfrac{1}{2}\gamma), & otherwise\end{cases} \tag{2.31}$$

p' 表示参考关键桢 i 中的关键点 p 向当前桢 j 投影后所得到的点，其数学表示式如下：

$$p'=K(RK^{-1}(p,d_p)+t) \tag{2.32}$$

$$\begin{bmatrix}R & t\\ 0 & 1\end{bmatrix}=T_jT_i^{-1} \tag{2.33}$$

其中，K 为相机内参，R,t 表示当下桢相对于参考关键帧的旋转和平移参数，也是需要估计的参数。优化总误差通过对每个特征点投影后灰度误差的带权重统计来获取，权重 ω_p 一般依赖于该特征点像素灰度的导数，其一种设计方案如下：

$$\omega_p=\frac{c^2}{c^2+\|\nabla I_i(p)\|_2^2} \tag{2.34}$$

其中，c为一常数因子。

最后通过在图像金字塔从上到下逐层求对该误差函数求优化。从上层获取的优化参数作为下层参数优化的初始值，由粗到细完成对运动姿态的估计。该方法

在场景特征点比较丰富，外界光照干扰小的情况下能获得精度很高的运动姿态估计，但是其易受关照变化的影响。而且由于参考关键桢中的参考地图点深度存在误差，所以最终场景尺度在系统运行中会出现漂移现象，本文引入深度学习的目的之一就是为了改善系统在这一点的表现。

2.6 地图优化

SLAM 算法基于相邻桢间运动的增量式迭代计算，其会导致机器人位姿估计存在积累误差的问题，特别是在大规模地图中该问题尤其突出。在视觉 SLAM 中，一般采用图优化的方式对局部范围内的机器人位姿及环境地图点进行更新调整。图是由节点和边组成，SLAM 问题转化为图优化问题主要分两步：（1）构建图；（2）优化图。在间接法 SLAM 和直接法 SLAM 中存在不同的构造方式。图优化本质就是求使目标损失函数最小的参数，当前比较流行的优化框架有 g2o 和 Google Ceres Solver。

间接法：在间接法中误差建立的依据还是特征匹配点对的重投影误差。假设机器人在位置 x_i 处获取到特征点 p_j 的观测值为 z_{ij}，这里的观测值即是特征点在图像上的二维坐标，利用相机投影模型则可建立如式 2.35 的误差项，也是整个优化图中的一条二元边，如图 2.7 所示。

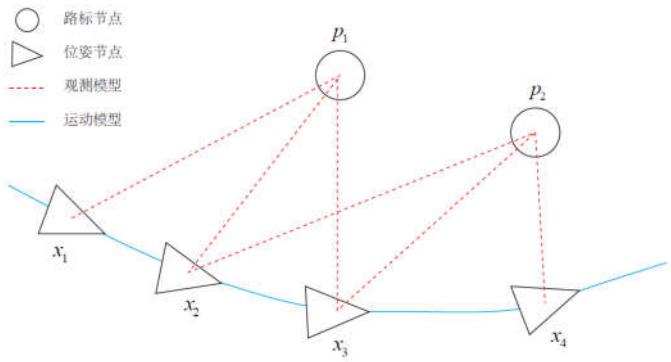


图 2.7 间接法 SLAM 图优化示意图

Fig2.7 Indirect SLAM optimization diagram

$$e_{ij} = z_{ij} - \hat{z}_{ij}(x_i, p_j) \tag{2.35}$$

通过对优化窗口中的所有观测数据的堆积，即可完成对整个优化图的构建，也即是所有误差累积的总和，其数学定义如下：

$$F(X) = \sum_{i} \sum_{j \in P_i} e_{ij}^{T} \Omega_{ij} e_{ij} \tag{2.36}$$

其中，P_i 表示第 i 桢关键帧的特征匹配点点集，Ω_{ij} 为该误差的权重，则通过优化方法求出使目标函数最小的系统状态值为：

$$X^* = \arg\min_X F(X) \tag{2.37}$$

其中，X^*即为经过优化后系统状态向量的值。

对于间接法里的纯位姿优化，其优化目标函数跟地图整体优化目标函数（如式 2.36 所示）一致，只是通过控制地图点的位置状态值保持不变，只对优化函数里的机器人位姿进行优化。这样减少了优化变量的数量，有利于优化的快速收敛。间接法中一般在地图整体优化之前和之后都会先使用纯位姿优化。

直接法：深度追踪是对关键帧中的地图点的深度进行离散优化，如图 2.8 所示。其依赖与该关键帧形成短基线双目模式的图像桢，在相机投影模型的基础上建立全局误差函数。

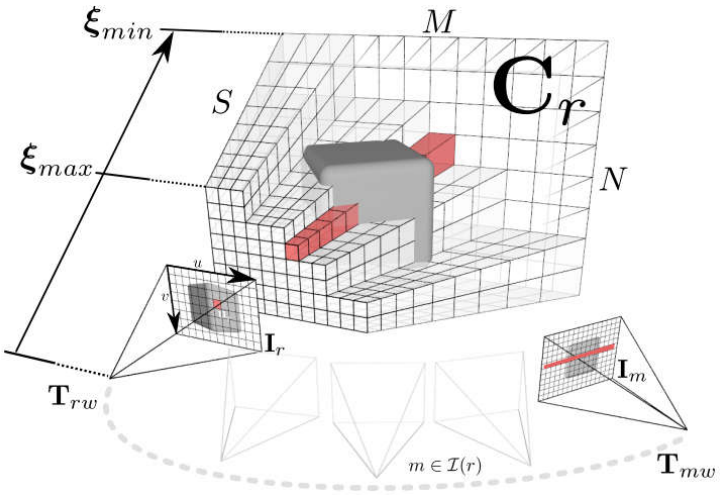


图 2.8 地图点深度追踪示意图

Fig2.8 Depth tracking diagram

图中T_{rw}为关键帧r到世界坐标的转换矩阵，T_{mw}为图像桢m到世界坐标的转换矩阵，I(r)为与该关键帧形成短基线双目模式的所有图像桢集合，$\xi_{\min},\xi_{\max}$为该地图点的最小最大逆深度，C_r为光度误差量。$C_r(u,d)$为某一地图点的平均光度误差，由下式可得：

$$C_r(u,d) = \frac{1}{|I(r)|}\sum_{m\in I(r)} \left\| \rho_r(I_m,u,d) \right\|_1 \tag{2.38}$$

其中，$d \in [\xi_{\min},\xi_{\max}]$，单个的投影光度误差由下式可得：

$$\rho_r(I_m,u,d) = I_r(u) - I_m(\pi(KT_{mr}\pi^{-1}(u,d))) \tag{2.39}$$

其中，π表示相机的投影模型，K为相机内参，T_{mr}为两图像桢间的位姿转换矩阵。

最后，通过在区间$[\xi_{\min},\xi_{\max}]$进行离散搜索求取使误差最小的深度值。在实际

的应用中，还会向误差函数里添加修正量来实现深度的平滑。

从上面的原理剖析中，可以看出深度追踪的效率和准确率依赖于初始的 $\xi_{\min},\xi_{\max}$ 的给定。本文通过引入预测网络对地图点深度的预测来初始化这两个参量，提升了深度追踪的效率和准确度。

对于整体地图优化，在直接法中误差建立的依据是灰度不变假设，即场景关键点在投影到新桢后灰度保持不变，单个误差项由两个机器人的位姿和一个特征点的逆深度构成，如式 2.30 所示。通过对每个误差项进行堆积，整个优化图如图 2.9 所示，其数学表达式如下式所示：

$$E=\sum_{i\in F}\sum_{p\in P_i}\sum_{j\in obs(p)}E_{pj} \tag{2.40}$$

其中，F 表示优化窗口内的所有关键帧，P_i 表示第 i 桢中的特征点集合，该桢也称为这些特征点的主桢，$obs(p)$ 表示特征点向其它关键桢投影后在其图像内的所有关键帧的集合，也称为该特征点的投影目标桢集合。

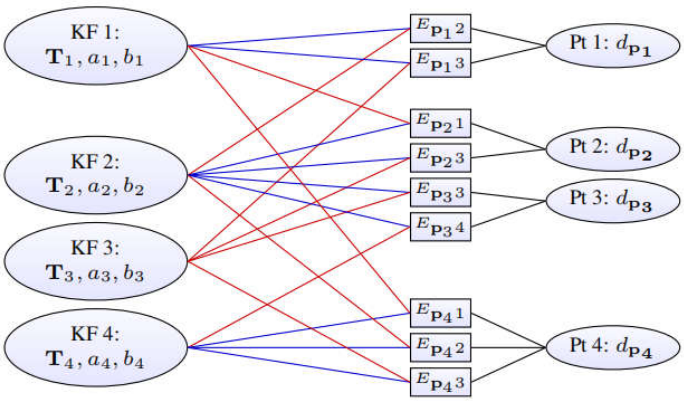


图 2.9 直接法 SLAM 图优化示意图

Fig2.9 Direct SLAM optimization sketch map

在图 2.9 中，作者还引入了相机光度标定参数 a,b 作为系统优化的状态变量，提升了系统对场景光照干扰的鲁班性[17]。本文借鉴了原作者的这一想法，同时在进行优化时引入了深度卷积神经网络对场景深度的预测来对特征点的逆深度初始值进行了更正，提升了优化的速度和精度。

在大规模场景中，闭环检测对减少长时间运动所产生的积累误差以及对于机器人追踪中的丢失恢复都具有重要意义。闭环检测即是识别是否曾经到过某个地方。如果闭环检测成果，则可以通过将对应的两个机器人位姿进行对齐，向图优化中加入新边，形成闭环。优化中将新的约束进行反向传播修正该局部的系统状态变量，极大的消除积累误差。闭环检测当前主要采用词袋算法（Bag of Word，BoW）对场景进行描述。在文献[69]中，我们也设计了基于图像分割的 BoW 算法，

提升了 SLAM 系统闭环检测的精度。

2.7 本章小结

本章主要是完成对 SLAM 问题的建模，同时对传统单目视觉 SLAM 算法进行了介绍。SLAM 问题的模型分别从概率和图优化两个方向进行了说明。对单目视觉 SLAM 算法的介绍首先描述了视觉 SLAM 算法的常用框架，然后对其各个核心部分就间接法和直接法两个角度分别进行了详细的说明，并对其中存在的问题进行了分析。核心部分主要包括特征点的提取与匹配、视觉里程计、地图优化等。本章的介绍为后面 LSLAM 算法的设计提供了理论基础。

3　基于深度学习的场景深度预测

3.1 引言

深度学习在众多传统领域都取得了比传统方法更好的效果，在 SLAM 问题中，同样也有研究者做了相关的工作，且获得很多突出的成果。立足于深度学习在 SLAM 技术上的发展上，本文借鉴研究者在场景深度预测上获得的一些现有成果，研究将其与传统单目视觉 SLAM 算法相融合，克服传统单目视觉 SLAM 算法的某些难点，最终取得了较好的效果。下文将就深度卷积神经网络的原理以及基于深度学习的场景深度预测做详细的介绍。

3.2 深度卷积神经网络设计原理

卷积神经网络是一种特殊的深层神经网络，它也是深度学习中最具代表性的网络结构，同时也是生物启发人工智能的最成功的案例。本节将详细介绍卷积神经网络算法原理，首先介绍卷积神经网络的生物学依据，然后详细阐述卷积神经网络算法的工作原理。

3.2.1 卷积神经网络的生物学依据

卷积神经网络的历史始于神经科学实验。神经生物学家 David Hubel and Torsten Wisel 经过多年对哺乳动物视觉系统的研究，提出了视觉信息处理层级模型[70]，他们凭借这一系列成就获得了 1981 年的诺贝尔医学奖。在 Hubel and Wiesel 的视觉层级模型中，其神经网络具有一个层级结构：外侧膝状体（Lateral Geniculate Nucleus, LGN）→简单细胞→复杂细胞→低阶超复杂细胞→高阶超复杂细胞。低阶和高阶超复杂细胞之间与简单和复杂细胞之间的神经网络具有类似的网络结构。在这种层状结构中，较高级别的细胞通常会有这样的倾向——对刺激模式的更复杂的特征进行选择性响应，同时也具有一个更大的接收域，同时对刺激模式位置的移动更不敏感。

2001 年 Simon Thorpe 等人[71]在《科学》杂志上发表了关于猴子在分类物体时大脑皮层活动的研究。他们发现在分类过程中，视觉皮层中的腹侧视觉路径（Ventral Pathway）是多阶段的，如图 3.1 所示，信息从视网膜经 LGN 流到 V1，然后到 V2，V4，之后是颞下皮层（Inferior Temporal , IT），而 IT 中包含对特定类别物体产生反应的神经元。

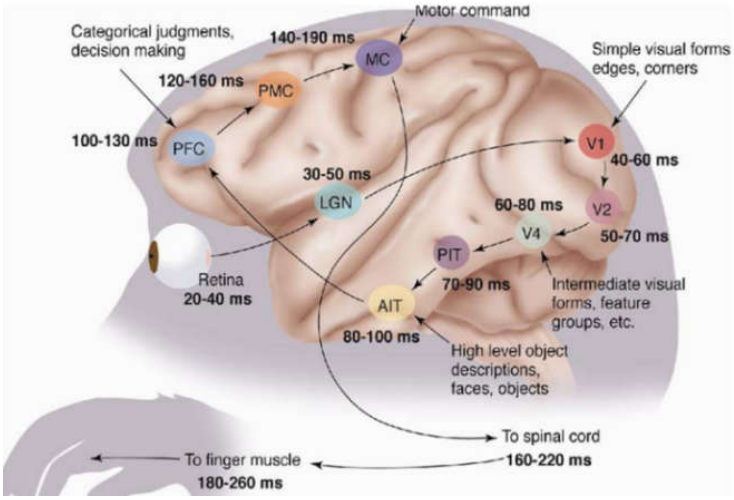


图 3.1 猴脑视觉系统处理分类任务时的工作图

Fig3.1 Monkey brain's vision system in classification task

上述过程是在观察对象的前 100ms 内发生的，如果继续观察对象更久，那么信息将开始回流，因为大脑使用自上而下的反馈来更新较低级脑区中的激活。然而，如果我们打断注视，并且只关注前 100ms 内的大多数前向激活导致的放电率，那么 IT 被证明与卷积神经网络非常相似。卷积神经网络可以预测 IT 放电率，并且在执行对象识别任务时与灵长类动物非常类似。

3.2.2 卷积神经网络的结构设计

将从生物学得到的关于大脑神经处理图像数据的知识具体化，产生了卷积神经网络的模型结构。卷积神经网络由多个处理特征的卷积网络叠加而成，对特征进行逐层的抽象和提取，一个卷积网络层的主要组件包括卷积层、激活层和池化层，如下图 3.2 所示。卷积层是其最主要的内容，下面将主要介绍卷积层结构设计中包含的原理。

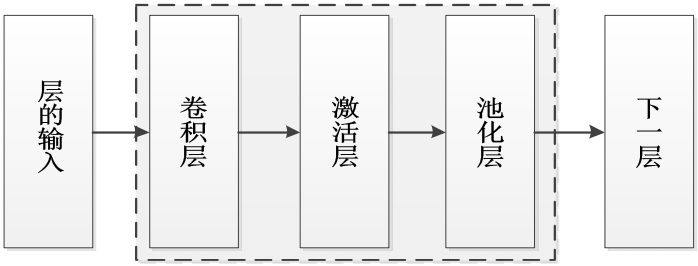


图 3.2 典型卷积网络层的组件

Fig3.2 Components of a typical convolution network layer

卷积神经网络，顾名思义是对输入信息进行卷积操作的神经网络，对一个二维输入进行卷积处理的直观例子如图 3.3 所示。通过对每一层输入的数据用多个卷

积核进行线性映射处理，能够提取到该输入的特征信息。随着卷积神经网络的不断加深，得到的特征图越来越抽象，视野范围也越来越大，得到的语义信息也越来越全局化。

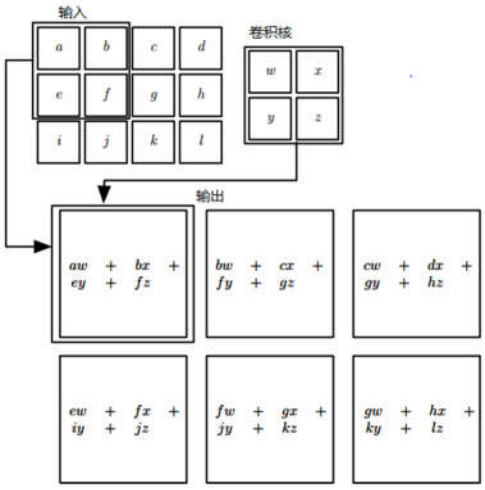


图 3.3 二维卷积操作示意图

Fig3.3 2D convolution operation

相比于传统的人工神经网络，卷积神经网络具有很多创造性的设计方法，其中，主要的改进来自于三个重要的思想：稀疏连接（Sparse interaction）、参数共享（Parameter sharing）和池化操作（Pooling operation）。

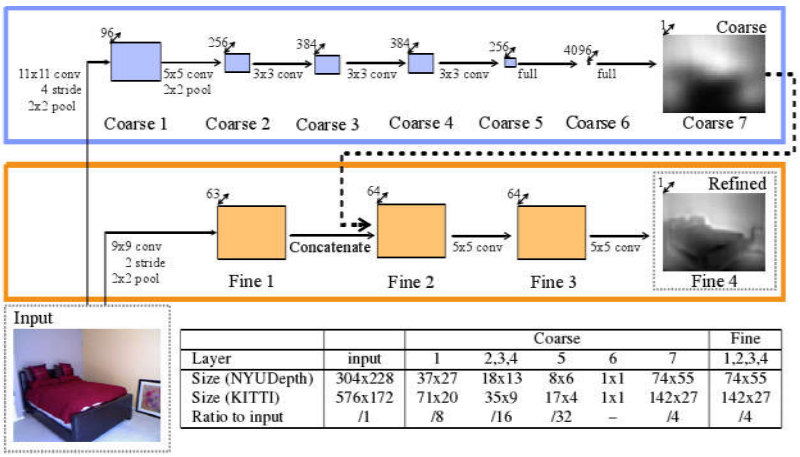
3.3 场景深度估计

场景深度恢复在传统 SLAM 中常采用深度离散区间搜索、三角化等方法获取，在深度学习中从图像恢复深度也是计算机视觉的一个研究热点。当前，基于深度学习的场景深度与运动估计一般融合成一个学习问题，相互监督，使整个问题变成了一个半监督问题，解决了训练数据难获取的问题。运动估计也称为视觉里程计（visual odometry），其通过关联图像之间的信息，利用多视图几何关系来确定机器人位姿，其一般作为视觉 SLAM 的前端。在本节将就基于深度学习实现场景深度恢复做详细介绍。

3.3.1 单张图像深度恢复

对于单张图像场景深度预测，最经典的算法是 David Eigen 等人[44]提出的深度预测网络。作者提出了一个多尺度的深度神经网络解决深度预测问题，整个网络，如图 3.4 所示，包含一个粗网络和一个细化网络。粗网络直接使用了经典的 AlexNet 网络结构，只是将原网络最后一层的分类器换成了粗预测的深度估计图；细化网络则采用大步长的卷积核将图片的大小变小之后，采用步长为 1，大小为 5 的卷积

核进行图像细节特征的提取。最终细化网络的输出与粗网络融合得出最终的预测结果。



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

图 3.4 深度预测网络网络结构

Fig3.4 Depth prediction network structure

Iro Laina 等人[45]利用全卷积神经网络设计的网络是当前实现从单张图像中恢复场景深度的网络中表现很突出的深度网络结构，如图 3.5 所示。在网络的设计上，作者使用了全卷积神经网络，一个方面使得网络对输入图像的大小没有了限制，另一方面使得网络的参数大大减少，网络运行实时性得到提高。作者在网络的前端使用了最近很流行的 ResNet50 的网络结构，在后端使用了类似于逆卷积的结构，使得网络最终的输出与输入的图像具有相差不大的大小。作者也利用了 Pre-train 的网络结构用来将高级的特征返回到和原图的大小，同时也采用了更深的网络。本文从网络预测的精度和效率考虑，最终也选择了该网络引入传统视觉 SLAM 系统。

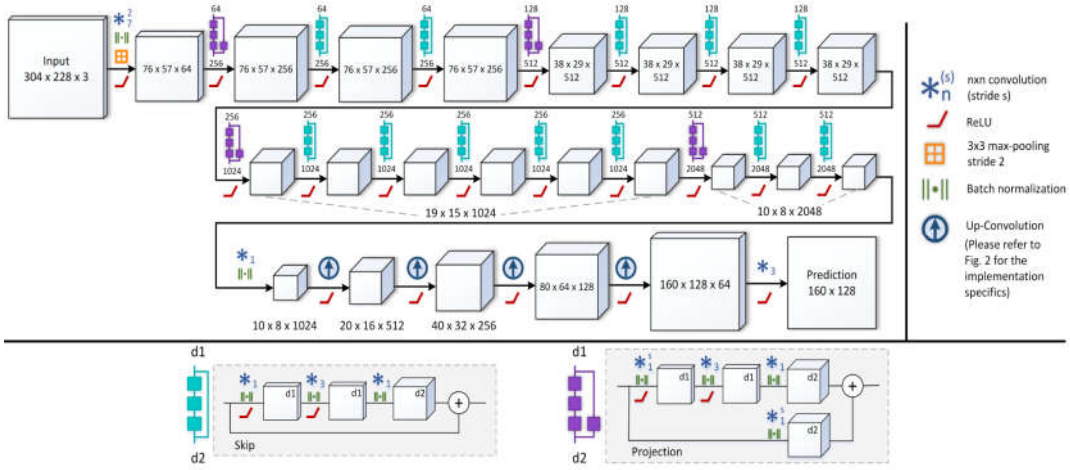


图 3.5 FCRN 深度预测网络网络结构

Fig3.5 FCRN network structure

3.3.2 图像对深度恢复

虽然研究者利用深度网络从单桢图像中恢复了场景的深度，但是该问题本身就是有病态性质的，其必须依赖于对场景的先验和语义的理解。所以，当前大多数研究者采用图像对来实现对场景的深度估计，同时完成对相机运动的估计。Benjamin Ummenhofer and Huizhong Zhou[46]提出卷积神经网络 DeMoN 从一对图像中估计场景深度和相机运动。网络使用位姿和深度作为监督信息来估计场景深度和相机运动。整个网络结构如图 3.6 所示，网络共由 3 部分组成：bootstrap net、iterative net、refinement net，其每个部分都由编码器-解码器这一种计算单元组成。bootstrap net 输入为图像对，第一个编码器-解码器输出为光流的估计及其置信度，最终输出对深度和相机运动的粗糙估计；iterative net 用于改善前面网络的输出结果，其将深度与运动估计转化为了一个光流场；refinement net 用于将缩小的预测深度图扩展成与原始图像相同分辨率的深度图，同时获取对相机运动的估计。

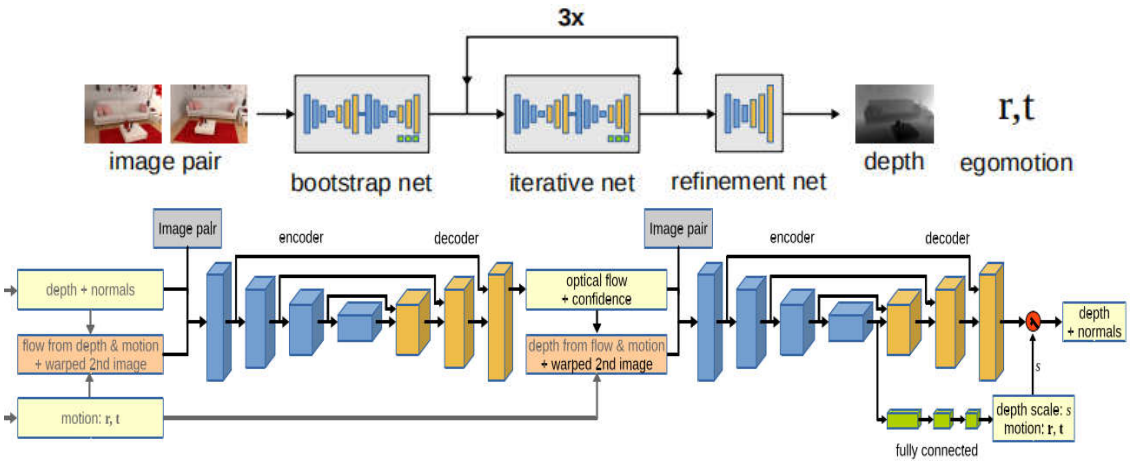


图 3.6 DeMoN 网络结构示意图

Fig3.6 DeMoN network structure

训练数据的标定是深度网络训练的一个难点，对于获取图像对的位姿和深度更是困难，所以有研究者提出了采用半监督甚至完全非监督的网络结构来完成该任务，克服训练数据难获取的难题。Sudheendra Vijayanarasimhan 等人提出了一个 SfM-Net[47]，其可仅仅依赖视频对网络进行训练，其网络结构如图 3.7 所示。网络设计的核心思想与直接法 SLAM 类似，也是基于光度不变假设。整个网络通过预测的深度和相机运动对图像进行转换，然后计算图像对之间的光度误差作为网络反向的监督信息。

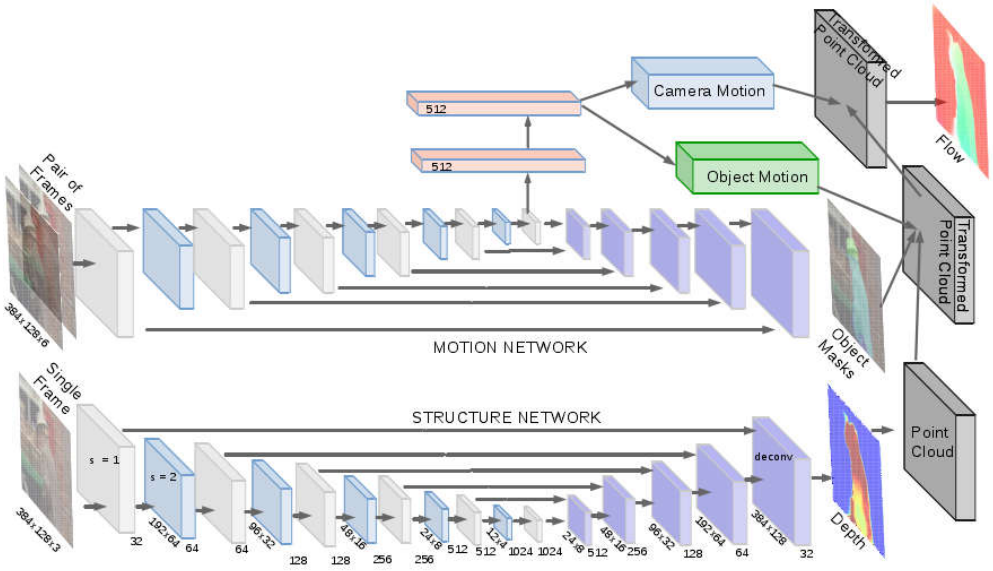


图 3.7 SfM-Net 网络结构示意图

Fig3.7 SfM-Net network structure

3.4 本章小结

本章首先描述了卷积神经网络的生物依据及其应用网络结构模型的设计，然后对其在场景深度估计上取得的进展进行了说明。对于场景深度估计的进展主要分单张图像深度预测和图像对深度预测两个方面进行了描述，选取了几个经典的网络对其网络结构和设计思路进行了详细说明。

4 LSLAM 算法的设计与实现

4.1 LSLAM 算法总体框架

本节将对本文设计的 LSLAM 算法做总体的说明。算法的核心思想是基于优化的框架将深度卷积网络预测的场景深度与传统单目视觉 SLAM 获取的场景深度信息、机器人运动信息相融合，解决当前单目视觉 SLAM 存在的一些问题，最终获得对整个 SLAM 系统状态变量的更好估计，达到提升系统运行的精度和鲁棒性的目的。具体实现上主要：（1）在单目视觉 SLAM 的初始化阶段，算法通过引入深度卷积网络对场景深度的预测信息解决无法估计初始场景尺度的问题；（2）在相机运动估计阶段，引入预测的场景深度信息对尺度进行监督和更正，从而有效的减少了系统运行中的尺度漂移；（3）在场景关键桢初始化中加入预测的深度信息对其深度参数进行初始化，从而改善后续优化的效果。引入深度学习一方面受启发于当前国际研究者在引入深度学习解决 SLAM 问题上的成果，一方面也考虑到了深度网络其强的鲁棒性、而这正是传统视觉 SLAM 算法缺失的。同时，在我们设计的 LSLAM 算法中场景深度预测网络运行于 GPU，系统优化融合运行于 CPU，计算资源互不抢占反而实现了机器人计算资源更好的利用，引入深度预测网络不会影响算法运行的效率。本文设计的 LSLAM 算法总体框架如图 1.13 所示。

传统单目视觉 SLAM 算法框架本文选择了基于直接法的 DSO SLAM 算法。其是当前基于直接法的视觉 SLAM 中效果很好的算法，但是其也存在着前面提到的单目视觉 SLAM 算法中的那些问题。场景深度预测网络在对几个候选网络进行实验比较后本文选取了利用全卷积深度神经网络进行场景深度预测的 FCRN 网络模型。整个 LSLAM 算法的设计主要包括以下三个关键部分：（1）系统初始化；（2）针对每桢图像的相机运动估计；（3）针对每个关键桢的关键帧初始化及滑动优化窗内关键桢的优化。下面将就场景深度预测 CNN 的选择过程以及在 LSLAM 算法设计中的三个关键部分进行详细说明。

4.2 场景深度预测 CNN

本文引入场景深度预测网络主要目的是为系统优化提供良好的系统状态初始值，所以在深度网络的选择上主要考虑了网络在场景深度预测上的精度与效率。本文共选取了三种候选深度神经网络进行实验比较：（1）Benjamin Ummenhofer 等人提出的 DeMoN 网络结构；（2）Sudheendra Vijayanarasimhan 等人提出的 SfM-Net 网络结构；（3）Iro Laina 等人提出的全卷积神经网络 FCRN 网络。

首先对三种神经网络在场景深度预测的精度上进行了比较，本文选取了 TUM

RGB-D SLAM 数据集[72]中的图像序列进行精度的比较，这里去除了尺度对深度的影响，对各种方法得到的地图点深度都乘上了一个尺度因子。本文选取了其中两张图像在三种网络的预测结果进行了展示，如下图所示：

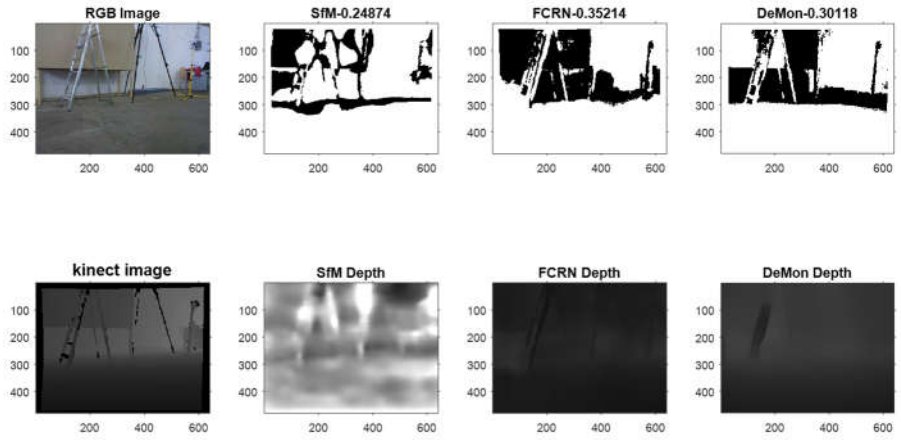


图 4.1 image1 预测结果比较

Fig4.1 Results comparison of Image1

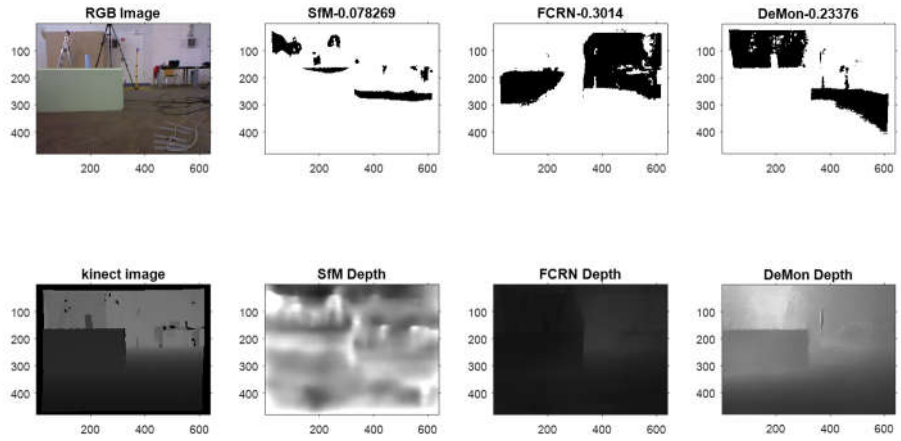


图 4.2 image2 预测结果比较

Fig4.2 Results comparison of Image2

在实验结果图中，第一排表示预测结果，其中黑色像素点表示深度预测正确，白色像素点表示深度预测错误；第二排是真实深度图及三种网络的预测深度图。在评判某个像素点的预测深度时，若预测深度与真实深度的差值的绝对值在实际深度的 20%之内，则认为该像素点深度预测正确，反之则认为预测错误。从图中，我们可以看出 FCRN 深度预测网络具有更高的预测准确度。

本文选择了单张图像深度预测所用的平均时间来比较三个深度网络的预测效率。本文选取了 TUM RGB-D SLAM 数据集中的图像序列中的 100 张图像计算平

均耗时，最后的实验结果如下表所示：

表 4.1 场景深度预测网络实时性比较

Tab4.1 Result comparison in real-time

场景深度预测网络	DeMoN	SfM-Net	FCRN
单张平均耗时（s）	1.71	2.80	0.46

从表中可以看出 FCRN 网络需要的计算资源更少，具有更高的效率。结合前面预测精度的比较结果，明显 FCRN 是这三种网络中的最佳选择。FCRN 网络预测的场景深度以带权重的方式引入系统，本文对 FCRN 预测网络预测的场景深度值与真实深度比值的范围进行了估计。选取了 2 张图像，获取真实与预测深度比值，若该值低于阈值，同时大于阈值的倒数则该点像素为黑色，反之，则为白色。实验结果如下图所示：

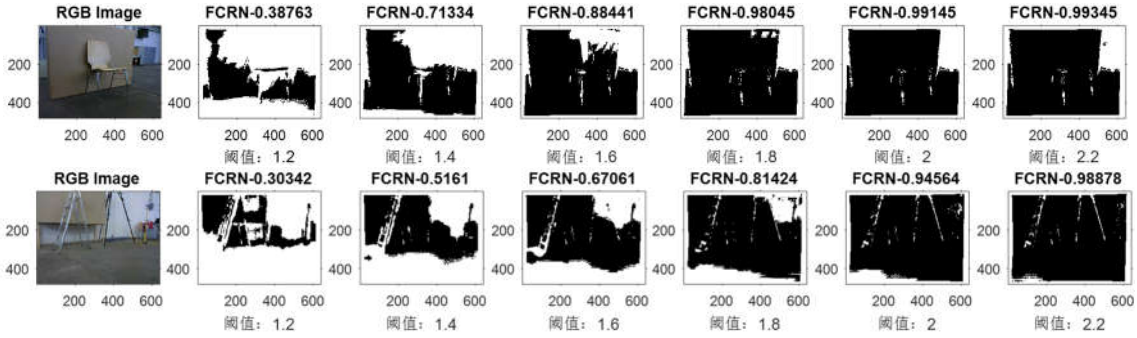


图 4.3 不同阈值 FCRN 网络预测结果展示

Fig4.3 FCRN prediction result with different threshold

图中数字表示黑色像素占的比例，从实验结果可以看出，本文选取的 FCRN 网络预测的深度值与真实值的比值范围近似在 0.5~2 之间。

4.3 系统初始化

由前面的分析可知，通过两桢图像估计相机运动必须有图像中某些关键点的深度信息作为基础，否则对相机运动估计的平移部分将与真实值相差一个尺度因子，所以单目视觉 SLAM 系统都需要进行地图初始化。在间接法中主要依赖对极几何完成，而在直接法中主要依赖图像金字塔实现，但是都无法估计场景的尺度。同时，当场景中的特征点信息较少时，系统有可能出现初始化失败。

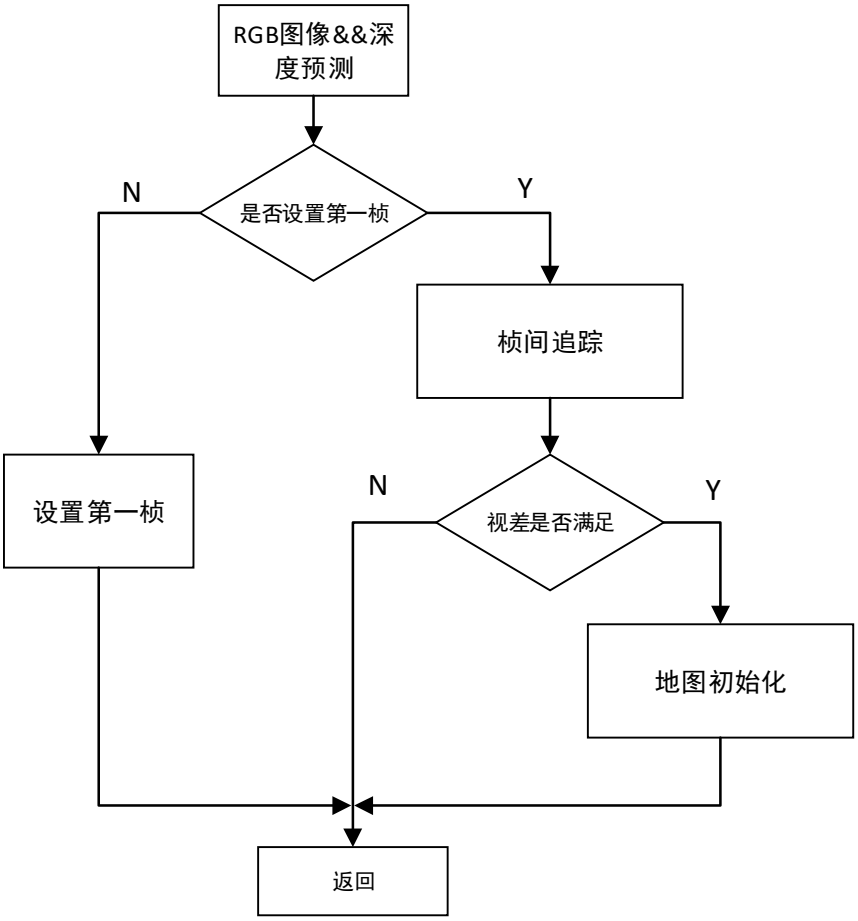


图 4.4 系统初始化流程图

Fig4.4 System initialization flow chart

本文设计的 LSLAM 算法在初始化时通过引入 FCRN 网络预测的深度信息辅助系统进行初始化，一方面解决了单目视觉 SLAM 初始化无法估计场景尺度的问题，另一方面提高了初始化的精度。整个算法流程图如图 4.4 所示，系统初始化主要包括如下三个核心部分：（1）设置第一桢；（2）桢间追踪；（3）地图初始化。

设置第一桢即是为系统设置第一桢初始关键帧，该桢的相机坐标位置也被作为地图和机器人世界坐标的原点。设置第一桢的第一步是图像中关键点的提取。关键点的提取首先获取每个像素点的图像灰度梯度的大小，然后将图像平均分割成小块，在每个小块里选取梯度最大的像素点作为图像中的关键点。设置第一桢的第二步是对提取出的关键点的逆深度初始值进行估计。采用逆深度主要是其可以很方便的表示无穷远的点。在一般的单目视觉系统中，由于对特征点逆深度信息没有先验知识，一般选择随机初始化，或是选择所有初始值为 1。在本文的算法中我们引入了深度网络对场景深度的预测作为特征点的初始深度值，这有利于在桢间追踪中保证优化收敛，同时获取到更好的追踪结果。

系统初始化中的桢间追踪主要是利用新的图像桢对第一桢中关键点的深度进

行优化，同时估计新图像桢中相机的运动。算法中两个问题整合成一个优化问题，优化目标函数如式 2.30 所示，整个优化过程也利用了图像金字塔由粗到细，逐步求解。算法中首先构造图像的灰度金字塔，和图像的深度金字塔，然后从金字塔顶层开始逐步求解，并将上层的关键点的深度信息和估计的相机运动参数向下层传递。在算法中，图像的深度金字塔本文通过 FCRN 网络预测的深度图进行降采样获取。

地图初始化主要将前面通过桢间追踪获取到的好的关键点初始化为关键桢中的初始地图点，为后面图像桢间的运动追踪建立条件。但是由于最初没有先验的深度信息，所以利用初始化中的桢间追踪估计获取到的也只是相对深度，其与真实场景深度相差一个尺度因子。在一般的单目视觉 SLAM 算法中，一般人为选择一个尺度因子对估计的深度进行修正。在本文的 LSLAM 算法中，首先统计出深度网络预测的深度的平均值，这里假设为d_p，然后统计出经过初始化桢间估计获取的所有关键点深度的平均值，这里假设为d_i，则系统尺度修正因子s为：

$$s = \frac{d_p}{d_i} \tag{4.1}$$

对于视差是否满足，主要通过间隔的追踪成功的图像桢的帧数来衡量，本文选取的帧数阈值为 5。整个算法如下所示：

算法：系统初始化

给定 原始 RGB 图像I_i，第一桢设置标志 Setfirst=0，判断视差是否满足的标志 Parallax=0.

1.获取 RGB 图像I_i的灰度图GI_i和场景深度图D_i.

2.设置图像桢F_i:

$$\mathrm{F}_i = addactiveframe(GI_i, D_i)$$

3.判断是否设置过第一桢，若设置了则跳过。否则设置第一桢 Firstframe，然后算法返回，等待下一帧图像：

$$\mathrm{Firstframe} = setfirst(GI_i, D_i)$$

$$Setfirst = 1$$

4.若已经设置了 Firstframe，则利用新图像桢与 Firstframe 进行桢间追踪，恢复 Firstframe 里地图点的深度，并判断是否满足视差：

$$Parallax = trackframe(\mathrm{Firstframe}, F_i)$$

5.若视差不满足则返回，否则进行地图的初始化：

$$Iinit = initializefrominitializer(\mathrm{Firstframe}, F_i)$$

算法 1 系统初始化

Algorithm 1 system initialization

为了验证本文设计的引入 FCRN 网络预测深度进行单目视觉 SLAM 初始化的合理性，本文选取了 TUM RGB-D SLAM 数据集中 4 个图像序列——Fre2_slam、Fre2_xyz、Fre1_xyz、Fre1_room 进行实验。对于每个序列首先获取本文设计的 LSLAM 方法和原始的 DSO 方法在初始化后关键帧中的所有初始地图点的平均逆深度，然后将其与使用 RGB-D 真实深度进行初始化后所有初始地图点的平均逆深度进行比较，获取的实验结果如下：

表 4.2 初始化后的平均逆深度

Tab4.2 Average inverse depth after initialization

方法 \ 数据集	Fre2_slam	Fre2_xyz	Fre1_xyz	Fre1_room
RGBD	0.1457	0.1749	0.2283	0.1768
LSLAM	0.2342	0.3121	0.3366	0.0771
DSO	1.2193	1.2960	1.3699	0.9523

从表中可以看出，本文设计的 LSLAM 算法初始化后的尺度相比于无监督的 DSO 算法获取的尺度有了质的提升，而且与采用真实 RGBD 深度图初始化的结果接近，说明了引入深度网络预测深度图是解决单目视觉 SLAM 系统尺度初始化问题的一个很好途径。但是也可以看出该算法估计场景的真实尺度依赖于深度预测网络预测深度的精度，其有待提高。

4.4 相机运动估计

相机运动估计主要通过优化的方式获取当前图像桢相对最后一帧关键桢的相机运动参数。该参数为后面滑动窗口中的关键帧中的未成熟地图点的深度追踪提供地图点投影参数。同时如果该新图像桢被选作为关键桢则为新关键桢联合优化提供初始的相机运动参数。在算法中，相机运动估计部分的算法框图如下图所示：

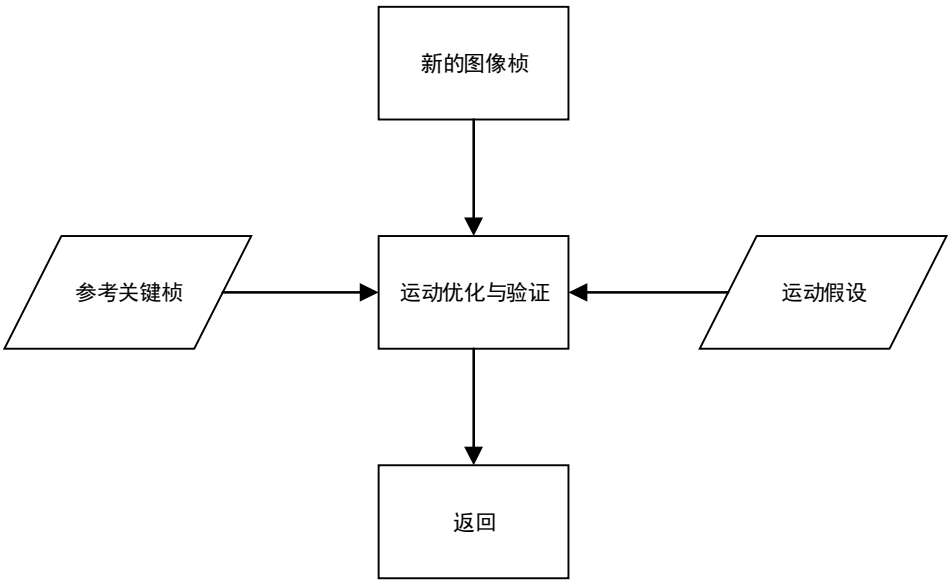


图 4.5 相机运动估计程序控制框图

Fig4.5 Block diagram of motion estimation

相机运动估计的实质是利用带有场景特征点深度信息的参考关键帧和一帧新的图像桢来估计两桢图像之间相机的相对位姿转换参数值。从前面的分析中可知，好的相机运动参数估计结果依赖于参考关键帧中特征点的数量及其深度的准确度。如果场景中特征点少，关键桢中的参考特征点就会变少，容易导致估计变差，甚至产生估计失败。同时如果场景深度信息有误差会导致估计结果的尺度发生波动。由于该估计的结果会作为后面优化的初始值，最终优化的结果会用于后续参考关键桢中地图点深度的估计，所以这种波动会积累，这就是系统尺度漂移的重要原因之一。

本文引入FCRN网络预测的场景深度值一方面当参考关键帧中的特征点少时，临时为参考关键帧添加特征点，从而提高运动估计在特征点较少的场景中的成功率；另一方面，通过对参考关键帧中准确度低的特征点的深度利用预测深度进行修正，从而改善估计的准确率，减少尺度的漂移。

如控制框图 4.4 所示，整个相机运动估计主要包括参考关键桢的构建、运动假设的提出、运动优化和验证三个关键方面，下面将对这三个方面进行详细介绍。

参考关键桢构建建立起估计的参考标准，极大的影响着最终的估计结果。本文中的参考关键桢构建主要分为如下 5 个步骤：

第一步：首先提取在优化窗口里所有的投影目标桢包含当前参考关键帧的所有特征点，同时去除在优化后为“外点”的投影关联。然后将提取的特征点向当前参考关键桢进行投影，构造参考关键桢的参考特征点。同时对于一个特征点对应多个投影的时候，对于该特征点的逆深度估计采用加权平均，每个投影的权重为最

终优化后的海塞矩阵对角线上对应的值。

第二步：统计该参考关键帧的 FCRN 网络预测的逆深度的平均值 d_p 为：

$$d_p = \frac{1}{n}\sum_i \frac{1}{D(i)} \tag{4.2}$$

其中 $D(i)$ 为像素点 i 预测的深度值，n 为总的像素点个数。

第三步：统计当前参考关键帧中所有特征点逆深度的平均值 $\tilde{d}$，及权重的平均值 $\tilde{w}$。

第四步：利用获取的预测尺度（近似为逆深度的平均值）和当前参考关键桢的尺度对每个特征点的逆深度进行修正，其修正方法如下式所示：

$$s = \frac{d_p * w_p + \tilde{d} * \tilde{w}}{\tilde{d}(w_p + \tilde{w})} \tag{4.3}$$

$$d_i = ((w_p + \tilde{w})/2 * s + \tilde{w}_i) * \tilde{d}_i / (\tilde{w}_i + (w_p + \tilde{w})/2) \tag{4.4}$$

其中，w_p 为 FCRE 网络预测深度的初始权重，本文算法中其值为 6，$\tilde{d}_i, d_i$ 分别为第 i 个特征点修正前和修正后的逆深度值。

第五步：判断关键桢中的特征点个数是否超过一定阈值。若没有，则从参考关键帧中选取关键点，使用 FCRN 网络预测深度对其逆深度进行初始化，从而为参考关键帧添加参考地图点。

运动假设主要为相机运动估计提供几个假设，同时也为优化提供了初值，在运动假设部分，算法中采用了前面第二章所述的方法。运动优化与验证部分主要通过最小化投影灰度误差总和来获取更精确地运动估计，同时也可通过查看目标函数优化后的结果验证该运动假设是否合理，整个优化方法如第二章所述。

整个算法如下所示：

算法：相机运动估计

给定 新的图像桢F_i，参考关键桢LastRef.

1.从运动假设模型中得到运动候选集TS.

2.从TS选取一个运动假设T_i，进行验证优化：

$$\text{Flag}\ , T_i = trackNewestCoarse(F_i, T_i, LastRef)$$

3.若运动假设验证通过则返回，否则返回到算法第二步继续验证优化；若所有运动假设都没有验证通过，则粗糙追踪失败，整个系统停止.

算法 2 相机运动估计

Algorithm 2 camera motion estimation

为了验证本文设计的相机运动估计算法的合理性，本文做了一个实验。在 4 个图像序列中，统计获取每张图像的 FCRN 网络预测的深度图的平均值及其真实深度图的平均值，然后得到两者的比值，实验结果如下图所示：

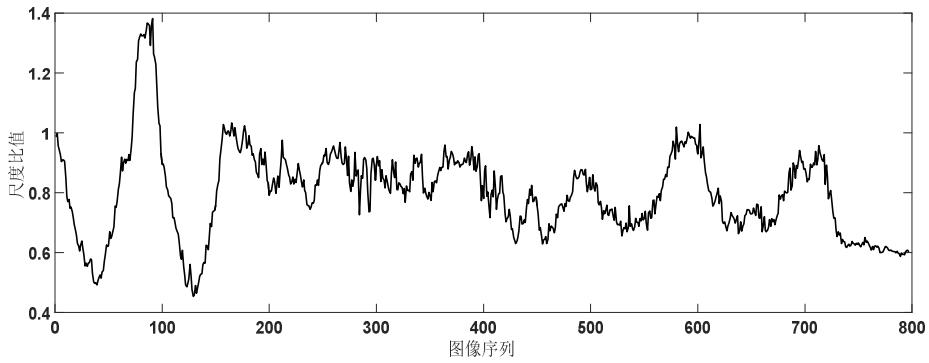


图 4.6 Fre1_xyz 数据集尺度比值

Fig4.6 Fre1_xyz scale ratio

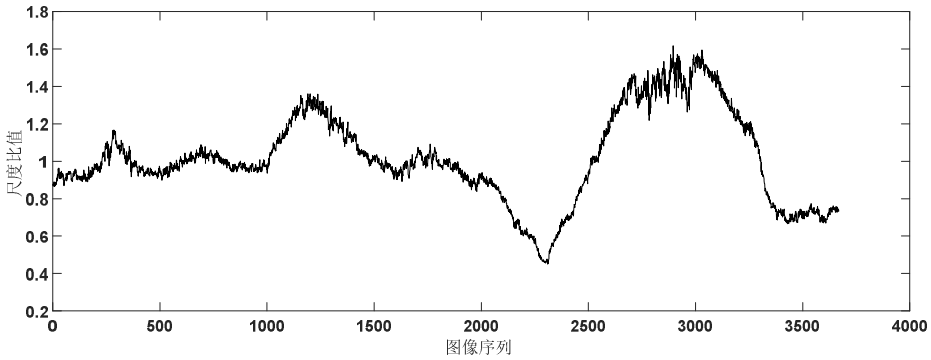


图 4.7 Fre2_xyz 数据集尺度比值

Fig4.7 Fre2_xyz scale ratio

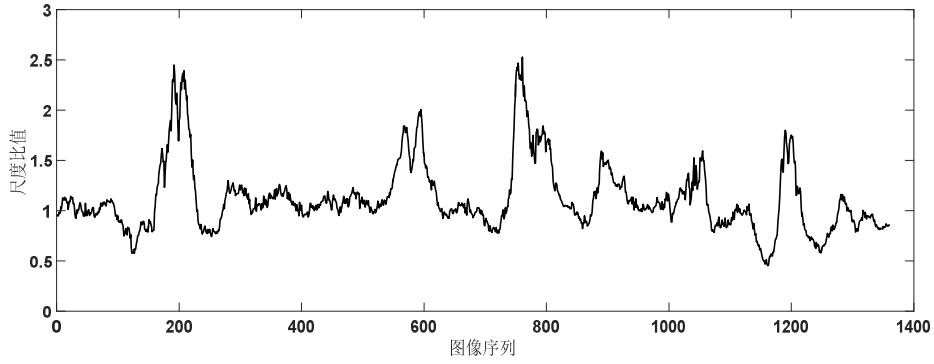


图 4.8 Fre1_room 数据集尺度比值

Fig4.8 Fre1_room scale ratio

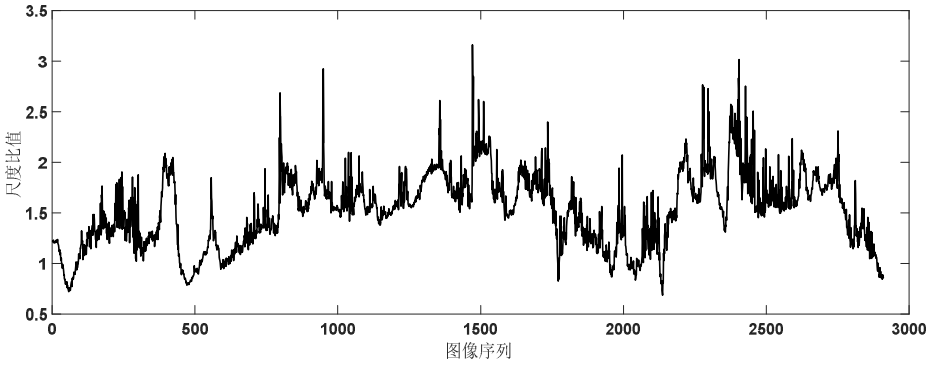


图 4.9 Fre2_slam 数据集尺度比值

Fig4.9 Fre2_slam scale ratio

从实验结果图中可知，对于简单场景（前两个数据集）网络对场景尺度的预测较稳定，对于复杂场景（后两个数据集）网络对场景尺度的预测波动较大，但是预测网络对场景深度的预测在尺度上总体不会出现导致崩溃的漂移，而且最重要的是不会产生尺度误差的积累。同时，为了改善尺度的稳定性，本文设计的 LSLAM 算法还对引入的预测尺度因子与当前系统内的尺度因子进行了权重滤波，改善这种不稳定性。总体来说，引入预测深度有助于改善单目视觉 SLAM 系统的尺度漂移问题。

为了展示引入预测深度后，对减少系统尺度漂移的作用，我们就本文设计的 LSLAM 算法和传统的 DSO 算法、RGB-D SLAM 算法在 4 个数据集上进行了对比实验。实验中首先将通过三种算法获取的机器人轨迹与真实的机器人轨迹按时间戳进行匹配，然后获取相同时间段里，算法估计的相机平移量与真实相机平移量的比值。实验结果如下图所示：

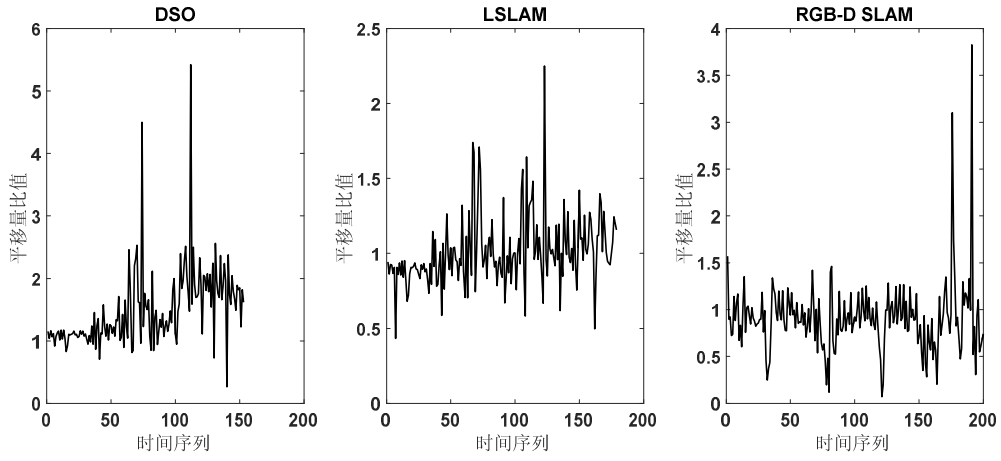


图 4.10 Fre1_xyz 数据集平移量比值

Fig4.10 Fre1_xyz translation ratio

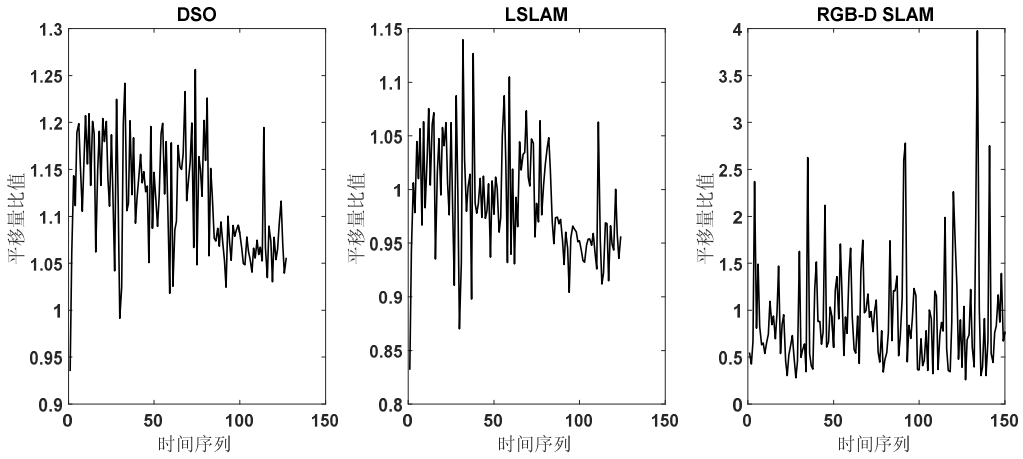


图 4.11 Fre2_xyz 数据集平移量比值

Fig4.11 Fre2_xyz translation ratio

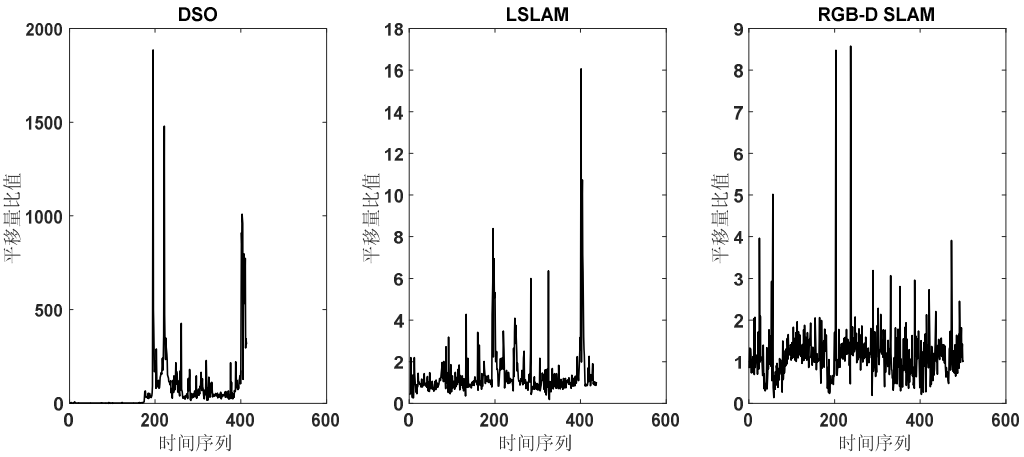


图 4.12 Fre1_room 数据集平移量比值

Fig4.12 Fre1_room translation ratio

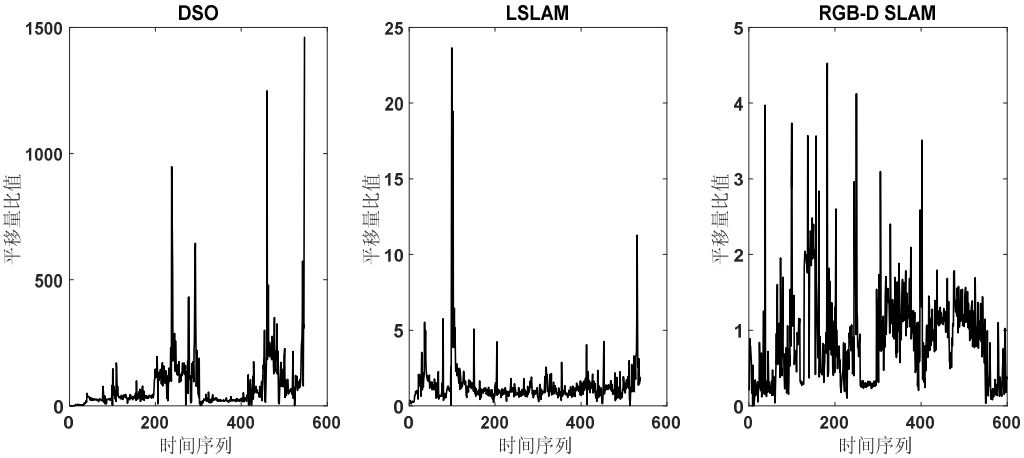


图 4.13 Fre2_slam 数据集平移量比值

Fig4.13 Fre2_slam translation ratio

从图中可以看出，传统 DSO 算法在简单场景上尺度漂移较小，但是对于复杂的场景尺度漂移十分严重，甚至会导致构图失败。LSLAM 算法通过引入 FCRN 网络预测的深度信息改善系统的尺度漂移问题，维持尺度的稳定。在简单场景中其提升作用较明显，而在复杂的场景中其对于尺度稳定的维持具有极其显著的作用，与 RGB-D SLAM 算法在尺度稳定上达到了几乎同一水平。这说明了引入网络预测的深度信息是解决系统尺度漂移问题的一个很好的途径。

4.5 关键帧的初始化与优化

关键帧初始化主要实现图像中初始未成熟地图点的选择及其逆深度的初始化。关键桢优化主要包含两部分：第一部分是当非关键帧进入系统，对关键帧中的未成熟地图点的深度进行离散追踪，同时追踪结果好的未成熟点初始化为关键帧中的地图点；第二部分是当关键帧进入系统，除了进行非关键帧进入时的操作，还利用新进入的关键帧对滑动窗口里的所有关键帧的地图和位姿参数进行优化更新。该部分结构框图如下图所示：

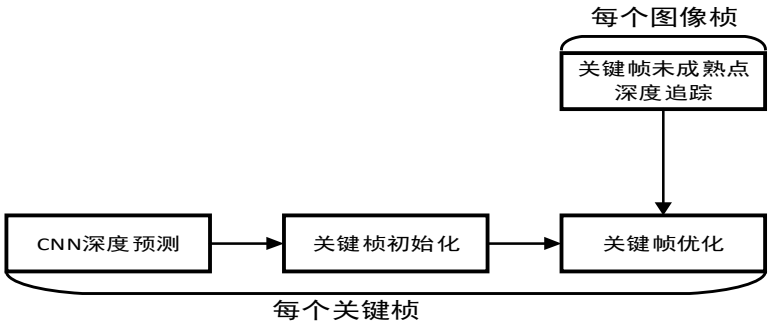


图 4.14 关键桢初始化与优化框图

Fig4.14 Block diagram of key frame initialization and optimization

关键帧初始化中初始关键点的选择采用系统初始化中图像平均分割找最大灰度图像梯度的像素点的方法。对于其逆深度的初始化，传统的 DSO 算法中对于未成熟点的逆深度初始化中采用了两个变量：逆深度最大值和逆深度最小值。对于其各自的初始值分别取最大数和 0。这种初始化方法在进行未成熟点深度追踪时搜索范围大。本文引入了 FCRN 网络的预测对未成熟点的深度的两个参量进行了估计。从 4.2 节中对 FCRN 网络预测效果的分析中可知，预测深度与真实深度的比值在 0.5~2 之间，所以本文设置了逆深度最大量为预测逆深度的 5 倍，逆深度最小量为预测逆深度的 0.2 倍，这样减少了离散搜索的范围，有利于提升速度和精度。整个算法如算法 3 所示。

同时本文还将窗口中所有的关键桢中的海塞地图点向当前需初始化的关键帧

进行投影，获取前面关键帧对当前关键帧某些未成熟点的约束，与 FCRN 预测深度带权重融合从而更好的初始化未成熟点的深度参量。对于初始化的精度本文对其进行了实验验证，利用预测的最大最小逆深度与真实逆深度值比较，获取在预测范围内的未成熟点比例，我们在 4 个图像序列上进行了实验，统计出了在每个图像序列上的平均值，实验结果如表下所示：

表 4.3 未成熟点逆深度的初始化正确率

Tab4.3 Initialization accuracy of the inverse depth

数据集	Fre1_xyz	Fre2_xyz	Fre1_room	Fre2_slam
正确率	100%	99.82%	100%	97.00%

算法：关键桢的初始化与优化

给定 新的图像桢F_i，优化窗口里的所有关键帧$frameHessians$，是否需要关键帧的标志NeedKF.

1.对所有优化窗口内的关键桢的未成熟点进行深度追踪：

$$traceNewCoarse(F_i)$$

2.若NeedKF为 0 则算法返回，否则进行关键帧的优化：

$$optimize(frameHessians, F_i)$$

3.构造新的关键桢NewKF:

$$NewKF = makeKeyFrame(F_i)$$

算法 3 关键桢的初始化与优化

Algorithm 3 the initialization and optimization of key frame

从实验结果中可以看出，未成熟点初始化的精度很高。关键桢优化中的未成熟点深度追踪采用在最大逆深度和最小逆深度之间进行离散搜索来更新参量，然后对于单个未成熟点建立灰度误差进行优化。如最终优化误差小则表示追踪成功，则将该未成熟点转化为关键帧中的地图点，该地图点的初始逆深度值d_h由下式可得：

$$d_h = \frac{d_{\max} + d_{\min}}{2} \tag{4.5}$$

其中，$d_{\max}, d_{\min}$分别表示未成熟点里的最大最小逆深度参量。

关键帧优化中的滑动窗口优化利用优化窗口里所有关键桢的所有地图点里的内点建立总误差函数，如式 2.40 所示，图形化如图 2.9 所示。最后利用传统优化算法对其进行优化。

4.6 本章小结

本章对本文设计的 LSLAM 算法进行了详细的说明。首先对 LSLAM 算法的总体框架进行了说明，然后对系统中四个关键点：场景深度预测 CNN、系统初始化、相机运动估计、关键帧的初始化和优化做了详细的介绍。场景深度预测 CNN 主要是深度预测网络的选择，后面三个关键点的介绍是本文提出的基于优化框架融合网络预测深度和传统视觉 SLAM 的设计方案的核心呈现。

5　实验设计与分析

第二章和第三章详细介绍了本文设计的 LSLAM 算法的理论基础；第四章对LSLAM 算法的各个核心部分的具体算法流程进行了说明，同时做了实验分析。本章将对LSLAM算法的硬件、软件实现平台进行说明，同时通过与传统DSO、RGB-D SLAM 算法做对比，对本文设计的 LSLAM 算法做整体性能表现的实验分析与总结。实验数据集选取了 TUM RGB-D SLAM 数据集中的 4 个图像序列——Fre2_xyz、Fre1_xyz、Fre2_slam、Fre1_room。其中，前面两个为简单场景，后面两个为较复杂场景。

5.1 硬件平台

本文 LSLAM 算法实现硬件平台采用实验室购买的“贾鲁普”机器人移动平台。其基于 ROS 开发，配备了丰富的惯性和视觉传感器。处理器采用 i7-4500U 酷睿双核处理器，1.8GHz CPU 睿频 3.0GHz，具有良好的计算性能。机器人最大移动速度为$0.8\mathrm{m/s}$，最大加速度$1.5\mathrm{m}/s^2$，最大角速度$230\mathrm{deg/s}$，最大角加速度$660\mathrm{deg}/s^2$，运动性能较好。其操作系统使用Ubuntu14.04，系统稳定而且便于进行算法开发。其也配备了各种传感器的驱动程序方便传感器数据的获取及周边硬件的扩展。远程端通过 SSH 远程登陆服务远程控制系统，便于使用机器人进行实验。同时，在手机端配备了遥控图传 app 安卓版，在电脑端配备了遥控图传 windows 客户端，也可使用手柄控制机器人运动，所以对于机器人“甲鲁普”的控制十分简易、方便。这些特点都为算法设计和实验验证提供了很好的条件，省去了很多前期算法设计的准备工作，十分便于 SLAM 算法的设计和开发。“贾鲁普”机器人外形如图 5.1 所示，其主要硬件组成如图 5.2 所示。

图 5.1 机器人“贾鲁普”

Fig5.1 Robot Jappeloup

	详细参数
i7主机	8G运存；64Gssd硬盘；无线网卡及天线；1个5A充电器；
底盘	两个主动轮；两个电机（带编码器）；1块ARM驱动板；一个从动减震万向轮；两个大功率稳压器；20Ah固体锂离子电池
1080P单目相机	广角170°
二层多孔位平台支架	支持机械臂、激光雷达、kinect等
kinect体感	
红外	距离可调；带固定底座
IMU九轴陀螺仪	

图 5.2 机器人“贾鲁普”硬件组成

Fig5.2 Robot Jappeloup hardware

本文的算法设计使用的单目摄像头，采用的是机器人“贾鲁普”上配备的 1080P 高清广角相机，其视角为170^{o}，如图 5.3 所示。

图 5.3 1080P 高清广角相机

Fig5.3 Wide angle camera

5.2 软件平台

本文设计的 LSLAM 算法基于机器人操作系统 ROS（robot operating system）完成其整个系统的设计，系统内的各个 ROS 节点模块使用 C++语言编程实现。

机器人操作系统 ROS 在 2007 年由斯坦福大学与机器人技术公司 Willow Garage 合作开发，2010 年正式发布开源机器人操作系统 ROS（robot operating system）。ROS 的出现给机器人研究者搭建起了交流的桥梁，促进了机器人的发展。ROS 操作系统已经经历了多个版本的更新，形成了十分系统的框架，当前最新版本 Kinetic Kame 为 ROS 的第十个官方版本。机器人操作系统 ROS 是一种次级操作系统。其提供类似操作系统所提供的功能，包含硬件驱动、程序间的消息传递、程序发行包管理等。机器人操作系统 ROS 的首要设计目标是在机器人研发领域提高代码的复用率，较少重复造轮子的工作，从而提高算法开发设计的效率，同时也有利于算法的推广应用。ROS 总体框架分为三个级别：社区级、文件系统级、计算图级。社区级是研究者进行网络代码发布的一种形式，这方便研究者交流、共享，这也是 ROS 发展的重要推动力量。文件系统级主要指基于 ROS 开发的源代码在文件中的组织形式。计算图级主要指 ROS 系统处理数据的一种点对点的网络形式。计算图级是进行算法设计的主要层级，其主要包括了节点、消息、主题、服务、ROS 控制器等。

本文设计的 LSLAM 算法在 ROS 下的组织架构如图 5.4 所示，其包括了图像获取节点 CamImage、场景深度预测节点 DepthCnn、定位建图节点 LSLAM 三个数据处理节点。其中，CamImage 节点通过主题 rgb_image 与节点 DepthCnn、LSLAM 进行通信。DepthCnn 节点发布场景深度预测的服务 depth_image，当 LSLAM 节点需要该服务时可向节点 DepthCnn 请求服务，从而获得场景的网络预测深度图。

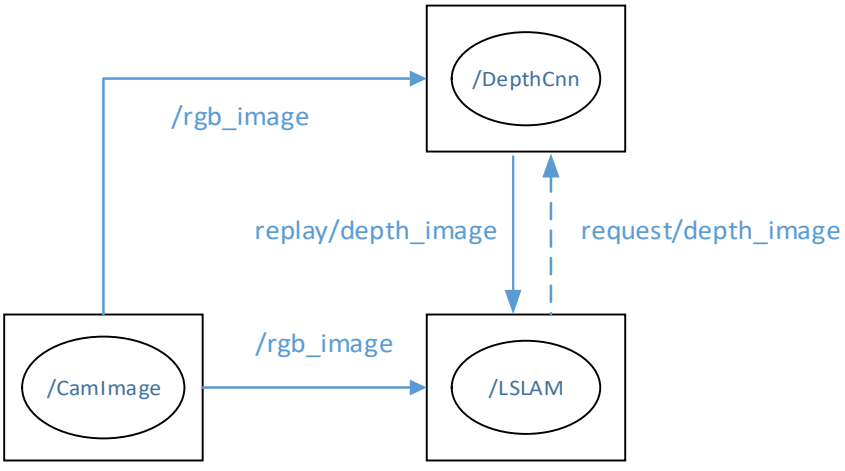


图 5.4 ROS 下 LSLAM 算法组织架构图

Fig5.4 Block diagram of LSLAM in ROS

5.3 实验与分析

5.3.1 性能评价标准

本文评价系统性能的指标借鉴了在文献[72]中叙述的两个评价指标，分别是机器人的绝对定位误差和相对定位误差。SLAM 问题其实最核心的就是机器人定位问题。对于一个 SLAM 算法，机器人绝对定位的误差决定了一个 SLAM 算法的优良，相对定位误差对于衡量系统尺度的漂移具有十分有效的作用。

绝对定位误差为算法获取的机器人轨迹上的所有节点的三维坐标与其对应的真实轨迹上的节点的三维坐标之间的距离，节点的配对利用节点的时间戳进行配对。其数学表示如下式所示：

$$e_{atei} = \sqrt{(X_{ei} - X_{ti})^T (X_{ei} - X_{ti})} \tag{5.1}$$

其中，X_{ei} 表示算法估计的第 i 个节点的三维坐标，X_{ti} 表示第 i 个节点对应的真实轨迹中的节点的三维坐标。

相对定位误差计算中首先将算法获取的机器人轨迹上的每一个节点及与其具有某种固定距离的节点组成一对。距离可以为节点坐标之间的距离、节点位姿角度之间的差值、节点之间的时间间隔等。本文选择了节点之间固定的时间，且设定为 $1s$，然后将两个节点与真实机器人轨迹按时间戳进行配对，获取对应的真实节点组对。分别计算算法获取的和真实的节点组队之间的位姿变换矩阵，最后计算出两个位姿变换矩阵之间的变换矩阵，将该变换矩阵中的平移量作为最终的相对定位误差，其数学表达如下式所示：

$$T_{ij} = T_i^{-1} T_j \tag{5.2}$$

$$T_{ij}^{'} = T_i^{'-1} T_j^{'} \tag{5.3}$$

$$e_{rpeij} = translation.norm(T_{ij}^{-1} T_{ij}^{'}) \tag{5.4}$$

其中 T_i, T_j 表示节点组队中算法获取的两个时刻的机器人位姿，$T_1^{'}, T_2^{'}$ 为对应的机器人真实位姿。

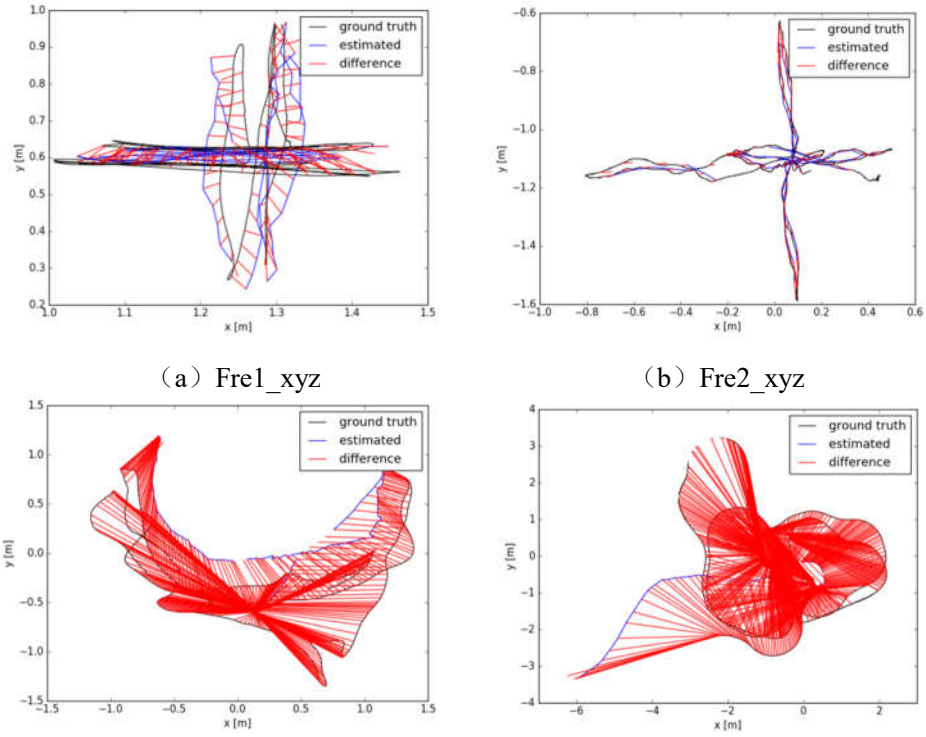
5.3.2 实验 1

为了验证本文设计的 LSLAM 算法在机器人定位上的表现，本文设计了一个实验对 LSLAM 算法和 DSO 算法的机器人绝对定位误差进行了比较。在数据集中的 4 个图像序列上，分别运行 LSLAM、DSO、RGB-D SLAM 算法，获取机器人的运动轨迹，然后统计出机器人绝对定位误差的平均值及其标准差。实验结果如表 5.1 所示，数据单位为 m。同时还对算法获取的机器人轨迹、机器人真实轨迹及其差异在二维平面进行了可视化，如图 5.5、5.6、5.7 所示。

表 5.1 机器人绝对定位误差

Tab5.1 Absolute translation error

方法 \ 数据集	Fre1_xyz	Fre2_xyz	Fre1_room	Fre2_slam
DSO	0.05763±0.02907	0.01175±0.00584	0.75390±0.33218	2.10903±0.66453
LSLAM	0.03996±0.01837	0.01071±0.00575	0.47935±0.18396	0.99944±0.41296
RGB-D	0.01347±0.00607	0.01897±0.00901	0.10117±0.07070	1.56933±0.66889

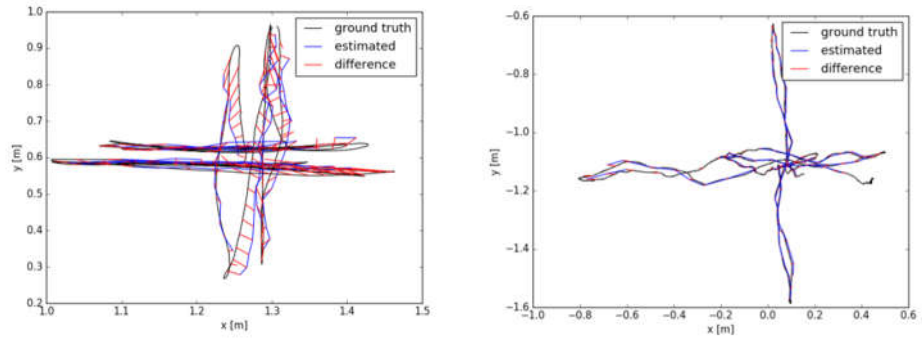


（a）Fre1_xyz　（b）Fre2_xyz

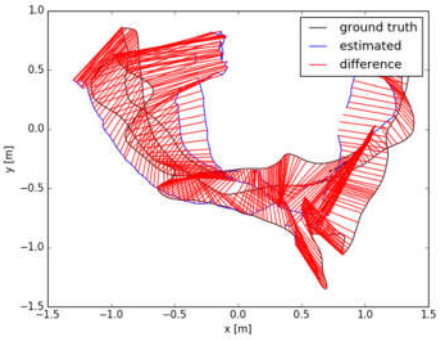
（c）Fre1_room　（d）Fre2_slam

图 5.5 DSO 算法下机器人轨迹可视化

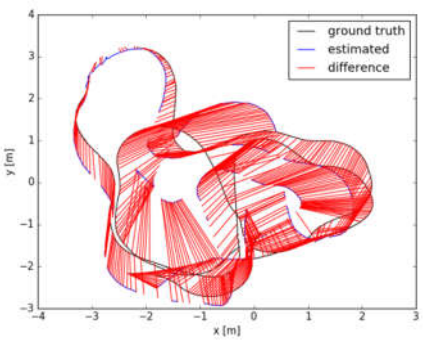
Fig5.5 Robot trajectory with DSO algorithm



（a）Fre1_xyz　（b）Fre2_xyz



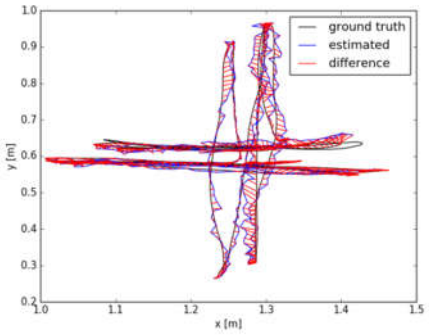
（c）Fre1_room



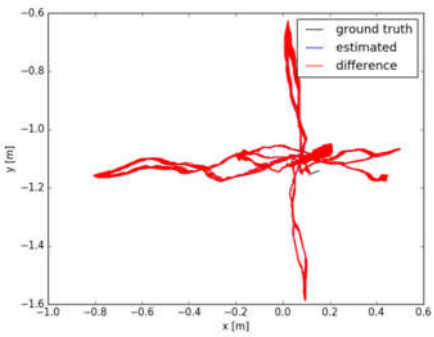
（d）Fre2_slam

图 5.6 LSLAM 算法下机器人轨迹可视化

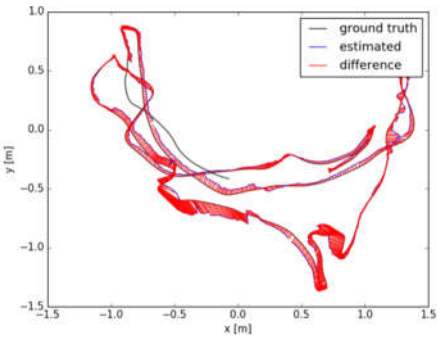
Fig5.6 Robot trajectory with LSLAM algorithm



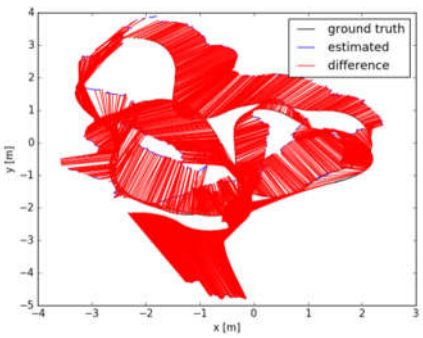
（a）Fre1_xyz



（b）Fre2_xyz



（c）Fre1_room



（d）Fre2_slam

图 5.7 RGB-D SLAM 算法下机器人轨迹可视化

Fig5.7 Robot trajectory with RGB-D SLAM algorithm

从表中可以看出，LSLAM 算法相对于 DSO 算法在机器人定位精度上有了明显的提升，特别是对于复杂的场景。从可视化图中也可以看出，对于简单的场景 LSLAM 算法定位精度比 DSO 算法更高，对于复杂场景，DSO 算法尺度漂移严重，基本上定位失败，LSLAM 算法尺度维持的较稳定，定位效果明显提升。这都说明了本文设计的基于优化融合 FCRN 深度预测网络与传统视觉 SLAM DSO 算法解决

单目视觉 SLAM 问题的框架具有很强的合理性，引入 FCRN 深度预测网络对于提升传统视觉 SLAM 算法的鲁棒性具有很好的作用。深度学习与传统视觉 SLAM 算法结合是发展高性能 SLAM 算法的一个很好方向。RGB-D SLAM 算法在小场景（a）、（b）、（c）下表现突出，但是在大场景（d）中表现较差。这显露了基于深度摄像头的 SLAM 算法的缺点，其得益于也受限于深度摄像头对场景深度的测量。深度摄像头测量深度具有一定范围，同时精度也随着测量范围的变化而变化。单目视觉 SLAM 具有更好的场景普适性。

5.3.3 实验 2

为了进一步说明本文设计的 LSLAM 算法在解决系统运行中尺度漂移问题上的良好性能，本文对机器人定位的相对定位误差进行了评价。实验还是选取了实验 1 数据集中的 4 个图像序列。最后获取整个轨迹上的相对定位误差的分布，如图 5.8、5.9、5.10 所示。相对定位误差的平均值及其标准差如 表 5.2 所示。

表 5.2 机器人相对定位误差

Tab5.2 Relative position error

方法 \ 数据集	Fre1_xyz	Fre2_xyz	Fre1_room	Fre2_slam
DSO	0.04999±0.02628	0.00705±0.00350	0.23223±0.13700	0.25611±0.07971
LSLAM	0.02802±0.01422	0.00378±0.00206	0.10095±0.07023	0.09234±0.09268
RGB-D	0.01952±0.00831	0.00531±0.00300	0.05422±0.07846	0.15886±0.21389

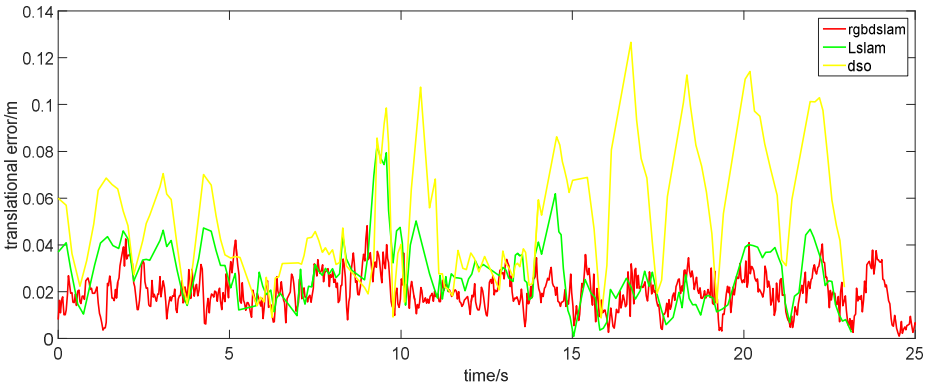


图 5.8 数据集 Fre1_xyz 上机器人相对定位误差

图 5.8 Relative position error in Fre1_xyz

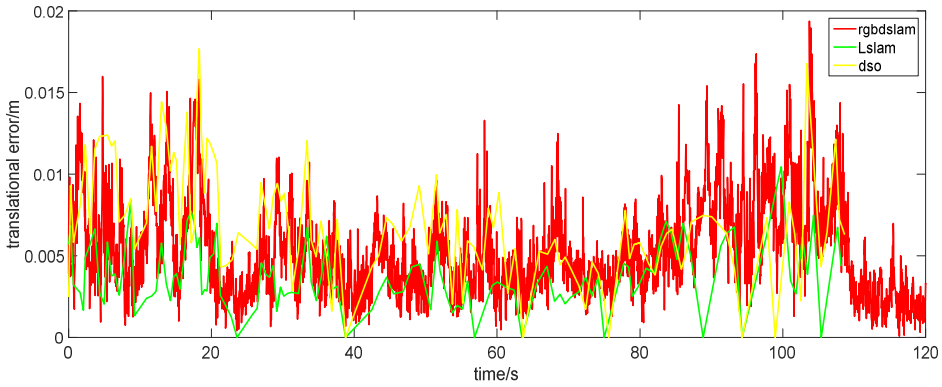


图 5.9 数据集 Fre2_xyz 上机器人相对定位误差

图 5.9 Relative position error in Fre2_xyz

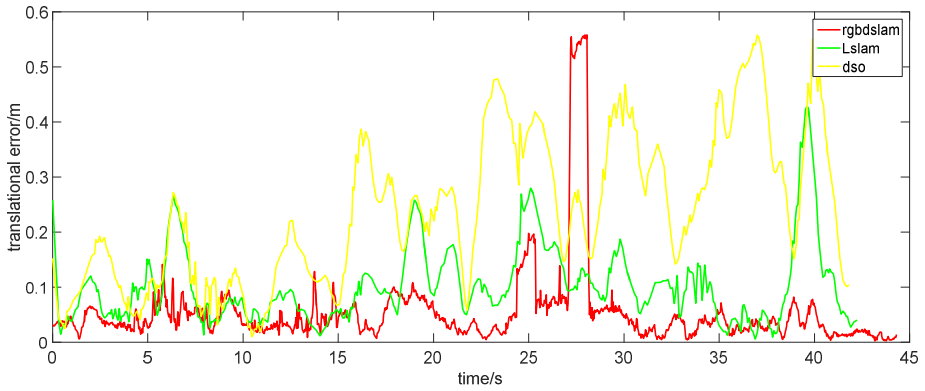


图 5.10 数据集 Fre1_room 上机器人相对定位误差

图 5.10 Relative position error in Fre1_room

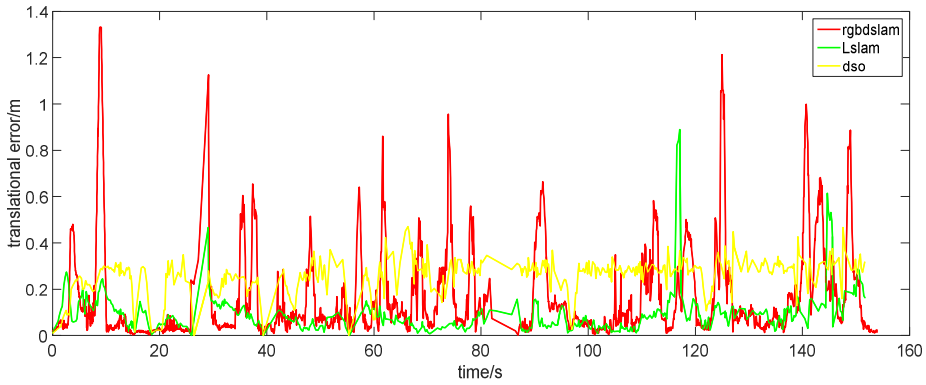


图 5.11 数据集 Fre2_slam 上机器人相对定位误差

图 5.11 Relative position error in Fre2_slam

从表 5.2 中可以看出，本文设计的 LSLAM 算法相对于 DSO 算法在尺度漂移上提升了两倍，在复杂的场景中其提升更明显。在大场景中，LSLAM 算法的表现甚至超过了 RGB-D SLAM 算法。这说明了本文设计的 LSLAM 算法框架的高效性。

从实验结果图中也可以看出，LSLAM 算法相对于 DSO 算法，机器人相对定位误差都更小且更稳定，这也说明了引入深度网络预测的深度信息对于解决尺度漂移具有很好的作用。

5.3.4 实验 3

在实验室场景下，基于机器人“贾鲁普”硬件平台，本文设计了一个实验来比较本文设计的 LSLAM 算法与 DSO 算法在定位精度上的表现。在实验中选取了 2 种运行模式，分别是四边形、圆形运动；实验场景展示选取了一些实验室场景关键桢进行了显示，如图 5.12 所示。实验结果如图 5.13、5.14 所示，算法的绝对定位误差（单位 m）的平均值和标准差统计如下表 5.3 所示。

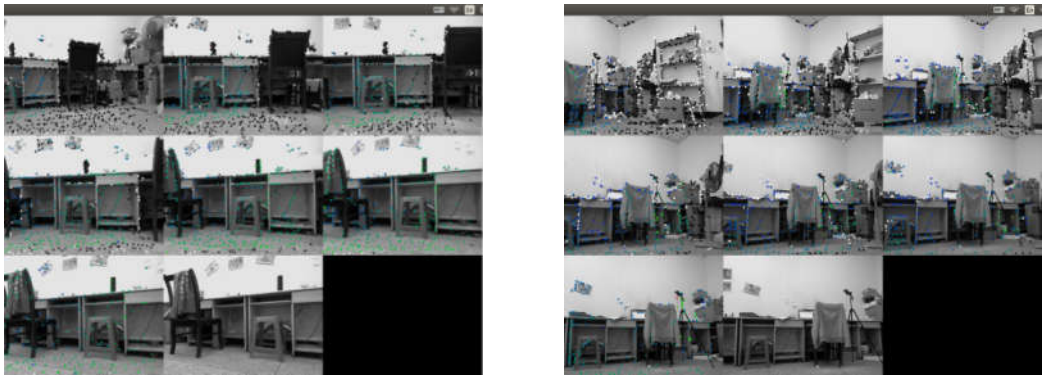
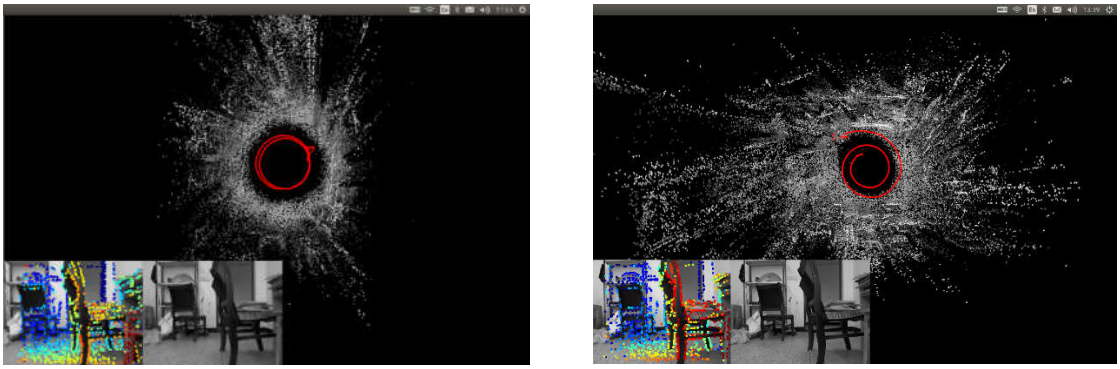


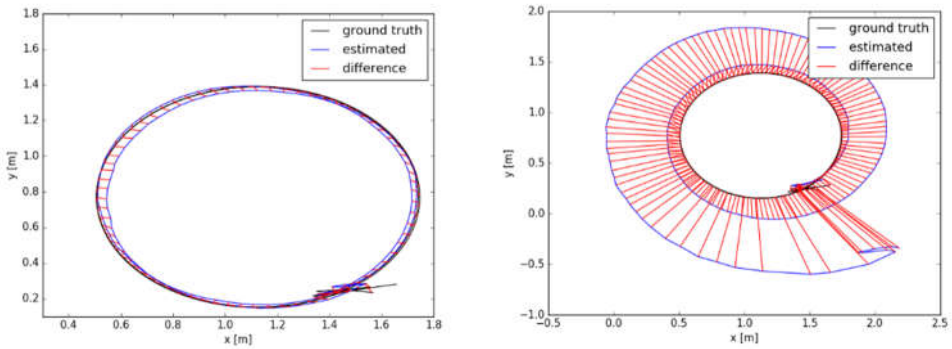
图 5.12 实验场景展示

Fig5.12 Experimental scene display



(a) 算法运行结果可视化（左：LSLAM，右：DSO）

(a) Visualization of running result (Left: LSLAM , Right: DSO)

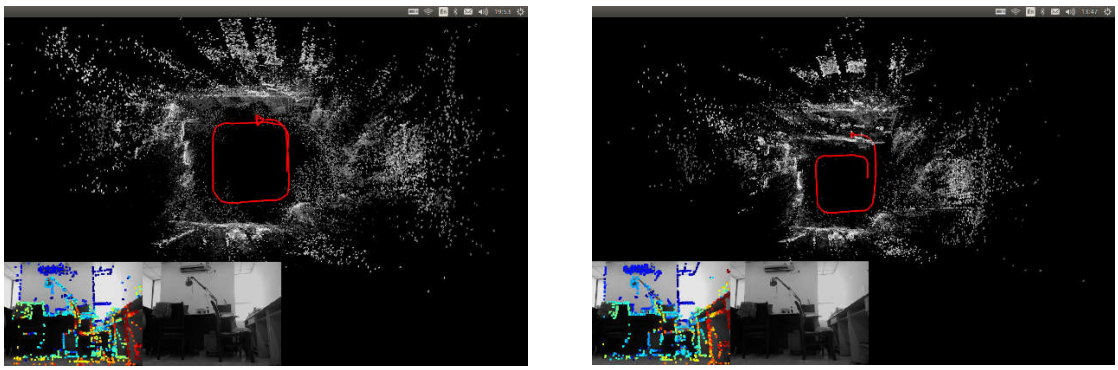


(b) 机器人定位结果图（左：LSLAM，右：DSO）

(b) Result of robot positioning (Left: LSLAM , Right: DSO)

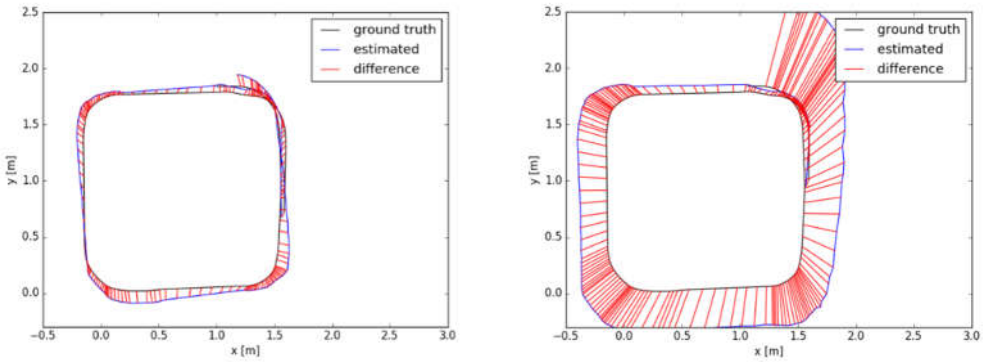
5.13 圆形运动模式下机器人定位实验结果

Fig5.13 Result of robot positioning under circular motion mode



（a）算法运行结果可视化（左：LSLAM，右：DSO）

(a)Visualization of running result (Left: LSLAM , Right: DSO)



（b）机器人定位结果图（左：LSLAM，右：DSO）

(b)Result of robot positioning (Left: LSLAM , Right: DSO)

5.14 四边形运动模式下机器人定位实验结果

Fig5.14 Result of robot positioning under Quadrilateral motion mode

表 5.3 不同运动模式下算法绝对定位误差

Tab5.3 Absolute translation error

方法 \ 模式	四边形	圆形
LSLAM	0.06828±0.02482	0.05335±0.01672
DSO	0.35868±0.20174	0.36264±0.21441

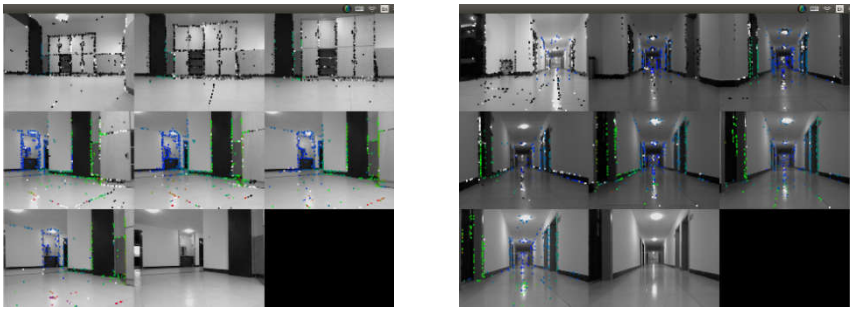
从实验中获取的绝对定位误差数据，以及机器人定位可视化中都可以明显的看出，本文设计的 LSLAM 算法在机器人定位精度上明显优于 DSO 算法。对于圆形的运动模式，机器人基本处于做匀速圆周运动，DSO 算法随着系统运行，漂移越来越大，而从实验结果中可以看出 LSLAM 算法对其进行了有效的控制，获取到了良好的定位效果；对于四边形的运动模式，由于存在纯旋转的运动，DSO 算法定位精度更差，而 LSLAM 算法具有了明显的改善。

5.3.5 实验 4

为了验证本文设计的 LSLAM 算法在实际应用中的室内大场景下的定位和构图精度，本文设计了一个与 DSO 算法的对比实验。实验场景为室内过道，过道结构示意图如图 5.15 所示，过道关键场景图如图 5.16 所示。分别采用 LSLAM 算法和传统的 DSO 算法对过道进行地图 构建同时完成机器人的定位，实验结果如图 5.17、5.18 所示，整个机器人运行轨迹的绝对定位误差的平均值和标准差如表 5.4 所示。

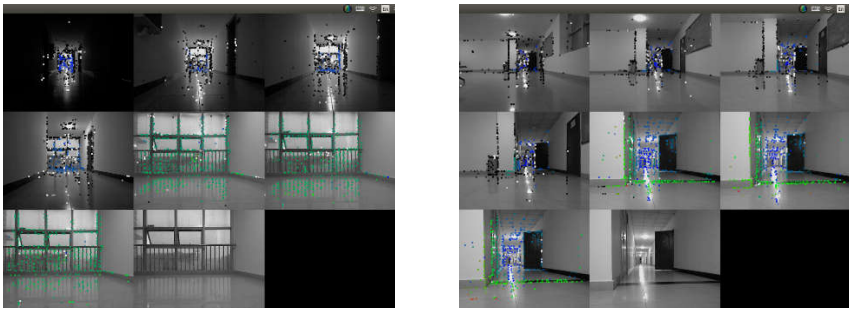
图 5.15 室内过道结构示意图

Fig5.15 Diagram of indoor aisle structure



（a）过道关键场景 1、2

（a）Key scene 1,2

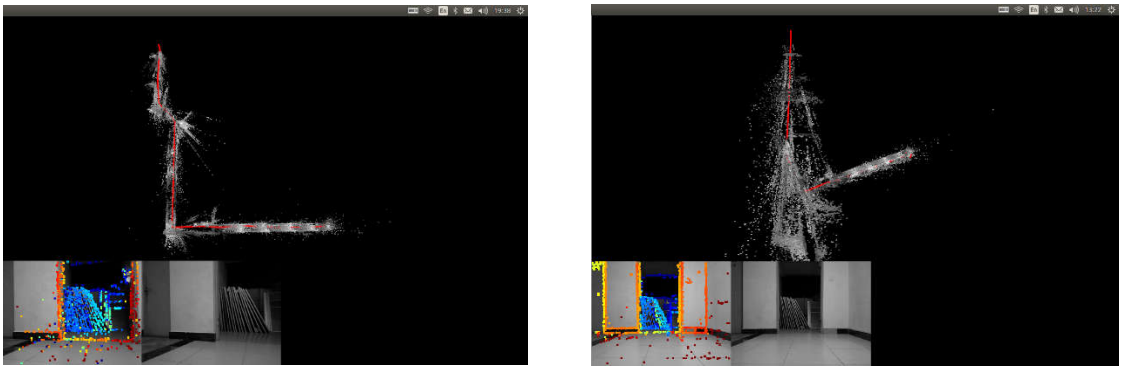


（b）过道关键场景 3、4

（b）Key scene 3,4

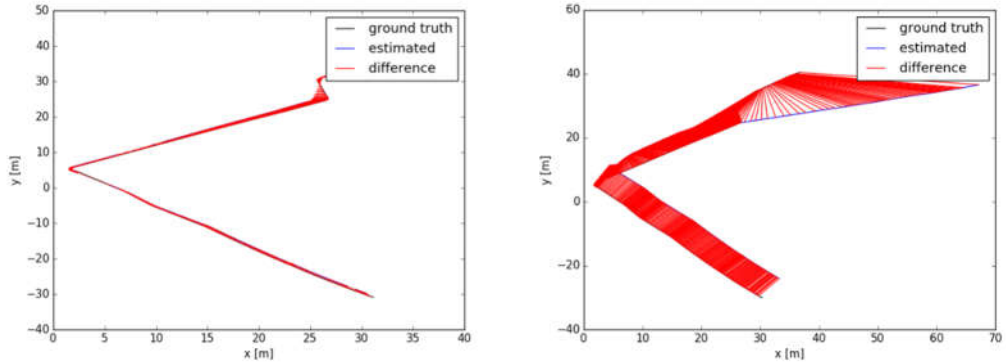
图 5.16 过道关键场景

Fig5.16 Key scenes of aisle



(a) 算法运行结果可视化（左：LSLAM，右：DSO）

(a) Visualization of running result (Left: LSLAM , Right: DSO)



(b) 机器人定位结果图（左：LSLAM，右：DSO）

(b)Result of robot positioning (Left: LSLAM , Right: DSO)

图 5.17 室内过道实验结果

Fig5.17 Experimental result in indoor aisle

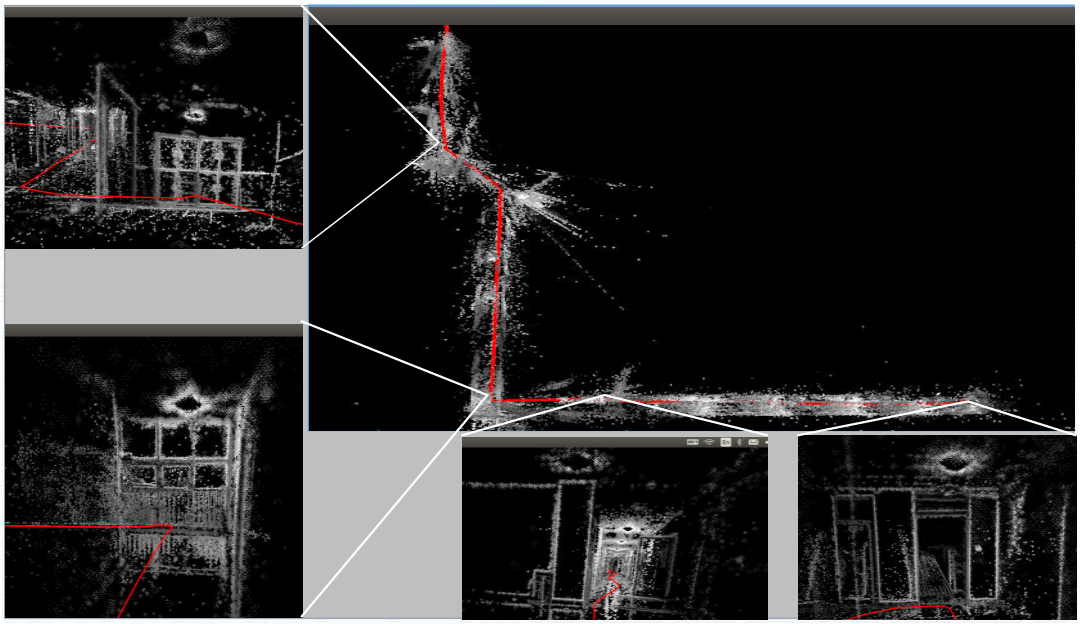


图 5.18 LSLAM 算法下过道地图局部放大

Fig5.18 Aisle local map zoom in under LSLAM

表 5.4 室内过道实验绝对定位误差

Tab5.4 Absolute translation error

方法	LSLAM	DSO
绝对定位误差	1.04377±0.53253	9.85752±4.58804

在本实验下，从机器人定位结果的可视化图和机器人整个轨迹的绝对定位误差统计表中可以明显得看出，本文设计的 LSLAM 算法相对于传统的 DOS 算法在实际应用的室内场景下对于克服 SLAM 系统的尺度漂移，系统抗干扰，机器人定位精度上都有了很大的提升。从 LSLAM 构建的过道地图局部放大中也可以看出在该场景下本文算法构建的地图具有很高的精度。这些都说明了本文基于优化的框架引入深度网络对场景深度的预测信息的方案的高效性，也说明了引入深度学习是发展高性能 SLAM 的一个很好方向。

5.4 本章小结

本章详细介绍了 LSLAM 算法实际实现中使用的硬件平台——机器人“贾鲁普”，及其软件平台——ROS。然后为了说明本文设计的 LSLAM 算法的有效性，本文设计了 4 个对比实验。第一个实验与 DSO、RGB-D SLAM 算法就定位绝对误差进行了比较；第二个实验就相对定位误差进行了比较。第三个实验在实验室场景下就机器人在两种闭环运动模式下机器人的绝对定位误差进行了比较。第四个实验在常见的实际应用场景——室内过道下，对机器人的表现进行了评估对比。四个实验都显示了本文设计的 LSLAM 算法相对于原始的 DSO 在机器人定位精度上有了很大的提升，引入场景深度预测网络对于提高传统视觉 SLAM 的鲁棒性具有很好的作用。

6 总结与展望

SLAM 技术是移动机器人的自主导航中的核心技术，同时其也是当前比较火的 AR/VR、无人驾驶等技术中的核心，当前其正是国际学者研究的热点。视觉 SLAM 因其获取的信息丰富、硬件成本价格低廉、应用需求大等特点得到了研究者的追逐。单目视觉 SLAM 技术更是因其硬件的简易、计算代价较低、易推广应用的特点成为热点中的热点，工业应用中对其的需求也越来越大。当前，单目视觉 SLAM 技术虽然在简单环境下已经取得了很好的效果，但是在实际中的应用还存在一些问题。一是实际应用中算法运行的实时性，系统的简易性；二是系统在实际应用中算法抗干扰能力。当前，如何引入深度学习提升传统视觉 SLAM 算法在复杂场景中的性能是当前研究者研究的热点，也是本文研究的重点。本文基于优化设计了 FCRN 场景深度预测网络与传统单目视觉 DSO SLAM 算法融合的 LSLAM 算法，并且基于机器人平台“贾鲁普”对算法进行了集成实现。在数据集上和实验室实际场景下进行了相关的实验，实验结果验证了本文设计的算法在机器人定位精确性和鲁棒性上的改善。

本文的研究重点是如何融合场景深度预测网络和传统视觉 SLAM 技术，所做的主要工作和取得的研究成果如下：

① 使用深度神经网络预测场景深度一直是计算机视觉领域研究的热点，本文对当前主流网络设计方案进行了文献检索，并选取了最具代表性的 3 种网络进行了实验分析，从而选取出最适合本文算法设计的网络。

② 本文基于优化设计出了一个场景深度预测网络与传统视觉 DSO SLAM 技术融合的算法框架，并在实验中验证了本文设计的 LSLAM 算法的有效性。

③ 针对单目视觉 SLAM 无法依赖纯视觉估计系统初始化尺度的问题，本文创新性的引入网络预测的场景深度信息从而较好的解决了系统尺度初始化的问题。从尺度初始化实验中，可以看出该方法具有较好的效果。

④ 单目视觉 SLAM 中一个很严重的问题就是在追踪过程中系统的尺度漂移问题。针对该问题，本文提出了在相机运动估计中融合网络预测的场景深度，从而维持系统尺度的稳定，从实验结果中可以看出本文设计的方法在尺度漂移问题上有了很好的提升，特别是在复杂场景中作用更明显。

⑤ 在关键桢初始化中，本文创新性的使用预测的深度信息对关键桢中的关键点进行深度初始化，而不是采用统一的假定参数，从而提高了初始化的精度，提升后期优化的效率和效果。实验结果中也能看出本文相对于 DSO 算法在机器人定位精度上的提升。

在本论文中，本文对深度学习与传统视觉 SLAM 技术相融合进行了研究、尝试，主要出于解决其存在的一些问题，从而提升传统视觉 SLAM 算法的效果。虽然从最后实验结果中可以看出取得了一些成果，但是仍然还有很多问题有待进一步研究，具体包括以下几个方面：

① 从实验中可以看出，预测网络对场景深度预测的精度对整个融合系统的性能具有很大的影响，而当前的预测精度还有待提高。所以对于预测精度更高，效率更高的场景深度预测网络的设计将是要进一步研究的重点。

② 从实验中可以看出，LSLAM 算法虽然对于复杂场景有提升，但是定位建图效果还是不理想，主要因为是：直接法视觉 SLAM 对于光照噪音敏感，有待进一步研究对其进行改善；在本文设计的算法框架中，预测网络对系统具有较强的监督作用，算法较依赖于预测网络的深度预测精度，传统视觉 SLAM 算法在后期优化中对预测的深度信息进行修正有待进一步改善，所以还需对算法框架进行调整，从而获取到更好的效果。

③ 在大规模场景下，闭环检测是解决 SLAM 问题中消除积累误差十分有效的方法。本文当前系统中并没有包含这部分，所以对于如何利用深度学习进行高效的闭环检测从而改善算法在大规模场景中的表现需要进一步的研究。

④ 当前设计的框架较繁杂，算法实时性还有待进一步提高，还存在很多需要优化的环节。

致　谢

时光匆匆，在重庆大学的三年的研究生学习生涯即将结束。在这三年的学习生涯中因为有那么些人，那么些事让我三年的求学生活过得充实而有趣。个人也在这三年里不管是 学术研究上的能力还是处理生活事物的能力都有了很大的进步。感谢这一路走来引导过我、帮助过我，鼓励过我的人。在此毕业论文完成之际，谨向你们致以我最诚挚的感谢。

首先我要感谢我导师李军老师，感谢你的引导，感谢你的教诲，感谢你的肯定。在研究生学习期间，李军老师对我进行了悉心的教导，指导我如何进行课题项目的研究，是我研究生学习成长路上的指路明灯。在论文开题、课题问题研究、论文定稿上，李军老师都给了细心的指导和帮助。李军老师严谨的教学态度和科研精神，渊博的学识给我留下了深刻的印象，对我产生了深远的影响，让我受益匪浅。

感谢在研究生期间给予了我很多帮助的王斌老师，谢谢你的关心和指导。感谢高杨健、叶云波师兄，许阳、沈广田、靳为东同学，也感谢 525 实验室让我们相遇。因为你们我的学习生活里充满了友爱和欢乐，想想那是一段人生多么美好的回忆，感谢你们给予我的关心和帮助。同时，在学校我还认识了很多热心的老师、知心的朋友，你们给予了我很多帮助，在此表示衷心的感谢。

特别感谢父母对我无私的爱，因为有你们的关心和支持，才有了此刻的我，愿你们永远健康快乐。

最后，衷心感谢在百忙之中抽出时间对论文进行评阅并提出宝贵意见的专家，教授。

陈剑斌

二 O 一八年四月　于重庆

参考文献

[1] 王耀南. 机器人智能控制工程[M]. 科学出版社, 2004.

[2] 迈向二十一世纪的中国机器人——国家八六三计划智能机器人主题十五年辉煌历程[J]. 高科技与产业化, 2001(1):28-30.

[3] 863 计划智能机器人主题专家组. 863 计划对中国机器人发展的巨大作用和深远影响[J]. 机器人技术与应用, 2001(3):4-6.

[4] Nilsson N J. A mobius automation: an application of artificial intelligence techniques[C]// International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1969:509-520.

[5] Thrun, Sebastian, Burgard, et al. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)[C]// The MIT Press, 2005:120-135.

[6] Leonard J J, Durrant-Whyte H F. Simultaneous Map Building and Localization for an Autonomous Mobile Robot[C]// Ieee/rsj Int. Workshop on Intelligent Robots and Systems. 1991:1442-1447 vol.3.

[7] Dissanayake M W M G, Newman P, Clark S, et al. A solution to the simultaneous localization and map building (SLAM) problem[J]. IEEE Trans Ra, 2001, 17(3):229-241.

[8] Steux B, Hamzaoui O E. tinySLAM: A SLAM algorithm in less than 200 lines C-language program[C]// International Conference on Control Automation Robotics & Vision. IEEE, 2011:1975-1979.

[9] Montemerlo M. FastSLAM : A factored solution to the simultaneous localization and mapping problem with unknown data association[J]. Ph.d.thesis Carnegie Mellon University, 2003, 50(2):240-248.

[10] Grisettiyz G, Stachniss C, Burgard W. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling[C]// IEEE International Conference on Robotics and Automation. IEEE, 2005:2432-2437.

[11] Civera J, Grasa O G, Davison A J, et al. 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry[J]. Journal of Field Robotics, 2010, 27(5):609–631.

[12] Endres F, Hess J, Sturm J, et al. 3-D Mapping With an RGB-D Camera[J]. IEEE Transactions on Robotics, 2017, 30(1):177-187.

[13] Mei C, Sibley G, Cummins M, et al. A constant time efficient stereo SLAM system[C]// British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings.

DBLP, 2009.

[14] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.

[15] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C]// IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007:1-10.

[16] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[J]. 2014, 8690:834-849.

[17] Engel J, Koltun V, Cremers D. Direct Sparse Odometry[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, PP(99):1-1.

[18] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainly[M]. Sage Publications, Inc. 1986.

[19] Smith R, Self M, Cheeseman P. A stochastic map for uncertain spatial relationships[C]// International Symposium on Robotics Research. 1988:467-474.

[20] Durrantwhyte H F. Uncertain geometry in robotics[J]. IEEE Journal on Robotics & Automation, 1988, 4(1):23-31.

[21] Moutarlier P, Chatila R. Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modelling[C]// Iser. 1989.

[22] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[M]// Autonomous Robot Vehicles. Springer New York, 1990:435-461.

[23] Kalman R E, Bucy R S. New Results in Linear Filtering and Prediction Theory[C]// Trans. ASME, Ser. D, J. Basic Eng. 1961:109.

[24] 刘艳丽. 融合颜色和深度信息的三维同步定位与地图构建研究[D]. 中南大学, 2014.

[25] Thrun S, Montemerlo M. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures[J]. International Journal of Robotics Research, 2006, 25(5):403-429.

[26] Stefan Leutenegger, Simon Lynen, Michael Bosse, et al. Keyframe-based visual-inertial odometry using nonlinear optimization[J]. International Journal of Robotics Research, 2015, 34(3):314-334.

[27] Strasdat H, Montiel J M M, Davison A J. Real-time monocular SLAM: Why filter?[C]// IEEE International Conference on Robotics and Automation. IEEE, 2010:2657-2664.

[28] Mur-Artal R, Tardós J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2016, 33(5):1255-1262.

[29] Zhao L, Huang S, Sun Y, et al. ParallaxBA: bundle adjustment using parallax angle feature

parametrization[J]. International Journal of Robotics Research, 2015, 34(4-5):493-516.

[30] Newcombe R A, Lovegrove S J, Davison A J. DTAM: Dense tracking and mapping in real-time[C]// IEEE International Conference on Computer Vision. IEEE, 2011:2320-2327.

[31] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-Time Single Camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007, 29(6):1052-1067.

[32] Scaramuzza D, Fraundorfer F. Visual Odometry: Part I: The First 30 Years and Fundamentals[J]. IEEE Robotics & Automation Magazine, 2011.

[33] Fraundorfer F, Scaramuzza D. Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications[J]. IEEE Robotics & Automation Magazine, 2012, 19(2):78-90.

[34] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2014:2100-2106.

[35] Engel J, Sturm J, Cremers D. Semi-dense Visual Odometry for a Monocular Camera[C]// IEEE International Conference on Computer Vision. IEEE, 2014:1449-1456.

[36] Schöps T, Engel J, Cremers D. Semi-dense visual odometry for AR on a smartphone[C]// IEEE International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2014:145-150.

[37] 武二永. 基于视觉的机器人同时定位与地图构建[D]. 浙江大学信息科学与工程学院 浙江大学, 2007.

[38] 熊斯睿. 基于立体全景视觉的移动机器人 3D SLAM 研究[D]. 哈尔滨工业大学, 2015.

[39] Zhang X, Rad A B, Huang G, et al. An optimal data association method based on the minimum weighted bipartite perfect matching[J]. Autonomous Robots, 2016, 40(1):77-91.

[40] Li L, Yang M, Wang C, et al. Rigid Point Set Registration Based on Cubature Kalman Filter and Its Application in Intelligent Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, PP(99):1-12.

[41] 宋宇, 李庆玲, 康轶非,等. 平方根容积 Rao-Blackwillised 粒子滤波 SLAM 算法[J]. 自动化学报, 2014, 40(2):357-367.

[42] 苑全德. 基于视觉的多机器人协作 SLAM 研究[D]. 哈尔滨工业大学, 2016.

[43] Konda K, Memisevic R. Learning Visual Odometry with a Convolutional Network[C]// International Conference on Computer Vision Theory and Applications. 2015:486-490.

[44] Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network[J]. 2014:2366-2374.

[45] Laina I, Rupprecht C, Belagiannis V, et al. Deeper Depth Prediction with Fully Convolutional Residual Networks[C]// Fourth International Conference on 3d Vision. IEEE Computer Society, 2016:239-248.

[46] Ummenhofer B, Zhou H, Uhrig J, et al. DeMoN: Depth and Motion Network for Learning Monocular Stereo[J]. 2016:5622-5631.

[47] Vijayanarasimhan S, Ricco S, Schmid C, et al. SfM-Net: Learning of Structure and Motion from Video[J]. 2017.

[48] Zhou T, Brown M, Snavely N, et al. Unsupervised Learning of Depth and Ego-Motion from Video[J]. 2017:6612-6619.

[49] Chen Z, Lam O, Jacobson A, et al. Convolutional Neural Network-based Place Recognition[J]. Computer Science, 2014.

[50] Hou Y, Zhang H, Zhou S. Convolutional Neural Network-Based Image Representation for Visual Loop Closure Detection[J]. 2015, 15:2238-2245.

[51] Bai D, Wang C, Bo Z, et al. Matching-range-constrained real-time loop closure detection with CNNs features[J]. Robotics & Biomimetics, 2016, 3(1):15.

[52] Arandjelovic R, Gronat P, Torii A, et al. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, PP(99):1-1.

[53] Mccormac J, Handa A, Davison A, et al. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks[J]. 2016:4628-4635.

[54] Li X, Belaroussi R. Semi-Dense 3D Semantic Mapping from Monocular SLAM[J]. 2016.

[55] Tateno K, Tombari F, Laina I, et al. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction[J]. 2017:6565-6574.

[56] Saxena A, Sun M, Ng A Y. Make3D: Learning 3D Scene Structure from a Single Still Image[M]. IEEE Computer Society, 2009.

[57] Besl P J, Mckay N D. A Method for Registration of 3-D Shapes[M]. IEEE Computer Society, 1992.

[58] Filliat D. A visual bag of words method for interactive qualitative localization and mapping[C]// IEEE International Conference on Robotics and Automation. IEEE, 2007:3921-3926.

[59] Cummins M. FAB-MAP : Probabilistic localization and mapping in the space of appearance[J]. Int.j.robot.res, 2008, 27(6):647-665.

[60] 曲丽萍. 移动机器人同步定位与地图构建关键技术的研究[D]. 哈尔滨工程大学, 2013.

[61] Bailey T. Mobile Robot Localisation and Mapping in Extensive Outdoor Environments[J]. 2002.

[62] Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey[J]. Foundations & Trends® in Computer Graphics & Vision, 2008, 3(3):177-280.

[63] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal

of Computer Vision, 2004, 60(2):91-110.

[64] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[C]// European Conference on Computer Vision. Springer-Verlag, 2006:404-417.

[65] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]// IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571.

[66] Brown R A. Building a Balanced k-d Tree in O(kn log n) Time[J]. Computer Science, 2014.

[67] Andrew A M. Multiple View Geometry in Computer Vision[J]. Kybernetes, 2004, 30(9/10):1865 - 1872.

[68] Civera J, Grasa O G, Davison A J, et al. 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry[J]. Journal of Field Robotics, 2010, 27(5):609–631.

[69] Chen J, Li J, Xu Y, et al. A compact loop closure detection based on spatial partitioning[C]// International Conference on Image, Vision and Computing. IEEE, 2017:371-375.

[70] Hubel D H, Wiesel T N. Brain and visual perception[M]. Oxford University Press, 2005.

[71] Thorpe S J, Fabre-Thorpe M. Seeking Categories in the Brain[J]. Science, 2001, 291(5502):260-3.

[72] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2012:573-580.

附　录

A. 作者在攻读学位期间发表的论文目录：

［1］Chen J, Li J, Xu Y, et al. A compact loop closure detection based on spatial partitioning[C]// International Conference on Image, Vision and Computing. IEEE, 2017:371-375.

[2] Xu Y, Li J, Chen J, et al. A novel approach for visual Saliency detection and segmentation based on objectness and top-down attention[C]// International Conference on Image, Vision and Computing. IEEE, 2017:361-365.

[3] Shen G, Li J, Chen J, et al. Motion control of manipulator based on K-Q algorithm[C]// Chinese Automation Congress. 2017:293-298.

B. 作者在攻读学位期间取得的科研成果目录：

[1] 李军，陈剑斌，沈广田. 基于 CR^2 神经网络的图像-文本双编码机理实现模型. 中国国家专利. 专利号：CN201710322410.1.

[2] 李军，沈广田，陈剑斌. 一种基于策略梯度的机器人学习控制方法. 中国国家专利. 专利号：CN201710321632.1.