

基于 YOLOv3 和视觉 SLAM 的语义地图构建

邹斌^{1,2}, 林思阳^{1*}, 尹智帅^{1,2}

¹ 武汉理工大学现代汽车零部件技术湖北省重点实验室, 湖北 武汉 430070;

² 汽车零部件技术湖北省协同创新中心, 湖北 武汉 430070

摘要 以相机作为输入的视觉同时定位与建图(SLAM)系统在地图构建过程中虽然可以保留点云的空间几何信息,但是并没有完全利用环境中物体的语义信息。针对这个问题,对当前主流视觉 SLAM 系统和基于 Faster R-CNN、YOLO 等神经网络结构的目标检测算法进行研究。并提出一种有效的点云分割方法,该方法引入支撑平面以提升分割结果的鲁棒性。最后在 ORB-SLAM 系统的基础上,结合 YOLOv3 算法进行环境场景的物体检测并保证构建的点云地图具有语义信息。实验结果表明,所提方法可以构建几何信息复杂的语义地图,从而可应用于无人车或机器人的导航工作中。

关键词 图像处理; 视觉 SLAM; 神经网络; 目标检测; 点云分割; 语义地图

中图分类号 TP391

文献标志码 A

doi: 10.3788/LOP57.201012

Semantic Mapping Based on YOLOv3 and Visual SLAM

Zou Bin^{1,2}, Lin Siyang^{1*}, Yin Zhishuai^{1,2}

¹ Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan, Hubei 430070, China;

² Hubei Collaborative Innovation Center for Automotive Components Technology, Wuhan, Hubei 430070, China

Abstract Visual simultaneous localization and mapping (SLAM) systems that use cameras as input can retain the spatial geometry information of a point cloud in the map construction process. However, such systems do not fully utilize the semantic information of objects in the environment. To address this problem, the mainstream visual SLAM system and object detection algorithms based on neural network structures, such as Faster R-CNN and YOLO, are investigated. Moreover, an effective point cloud segmentation method that adds supporting planes to improve the robustness of the segmentation results is considered. Finally, the YOLOv3 algorithm is combined with ORB-SLAM system to detect objects in the environment scene and ensures that the constructed point cloud map has semantic information. The experimental results demonstrate that the proposed method constructs a semantic map with complex geometric information that can be applied to the navigation of unmanned vehicles or mobile robots.

Key words image processing; visual simultaneous localization and mapping; neural network; object detection; point cloud segmentation; semantic map

OCIS codes 100.4996; 100.3010; 100.4999

1 引言

实现无人车和机器人自主移动的关键是对环境场景的理解。但以融合方法^[1]作为思路的场景重建只能生成三维稠密地图,缺乏相关的语义信息,限制了移动端对环境物体信息的理解。为了既保留传统

同时定位与建图(SLAM)构建的几何信息,同时又能提供环境中的语义信息,语义 SLAM 领域的早期雏形源自于 McCormac 等^[2],他们利用神经网络生成类别概率图,从而在地图中添加预测结果进行语义融合。类似地,由 Girshick 等^[3]早期设计的 R-CNN 经典检测算法也受益于神经网络,进而衍生出

收稿日期: 2019-12-24; 修回日期: 2020-01-29; 录用日期: 2020-02-25

基金项目: 国家重点研发计划(2018YFB0105203)、新能源汽车科学与关键技术学科创新引智基地基金(B17034)

* E-mail: xyz5016@whut.edu.cn

Faster R-CNN、Mask R-CNN 等优秀架构^[4-5],它们替代选择搜索算法对图像中的目标进行选定,最终获取物体的分类信息。为了理解场景区域,使得分割后的各区域保留相似特征,点云分割相关算法得到十分迅速的发展,例如,Stein 等^[6]使用空间和法线信息设计的 LCCP 方法和近期热门的 PointNet 分割算法^[7],但是这些方法只处理了环境的几何信息。

事实上,在空间点云分割和地图构建中,主流方案大多选择图像或者点云作为单个输入,不能同时利用 2D、3D 信息。针对维度信息结合困难的问题,本文使用 ORB-SLAM2 算法完成点云的构建,并通过结合的方法将点云分割为类别明显的点云区段;同时使用 YOLOv3 算法对关键帧进行语义信息的标记,从而便于语义标签匹配点云区段,建立密集的语义地图;最后在 TUM 数据集上进行语义地图的构建,以验证所提算法的有效性。

2 视觉 SLAM 与目标检测算法

2.1 ORB-SLAM2 算法

ORB-SLAM2 算法由跟踪、局部建图、回环检测 3 个工作线程构成^[8]。其中,跟踪线程对输入的图像进行特征点提取,并且对其和邻近的关键帧进行比较,在获取特征点位置后对相机进行粗略的定位;局部建图线程的核心是构建局部 3D 地图,优化局部地图和关键帧的位姿,关键帧负责点云的生成,其他帧则获取相机的位姿,把深度图中的点转换到世界坐标系中;回环检测线程主要用于回环位置和关键帧的检测,通过位姿优化校正相机的漂移误差。通过上述线程得到的点云在标记后包含了语义地图的粗略信息。从性能方面看,ORB-SLAM2 算法在回环检测中采用 DBow2 词典模型以保证系统的鲁棒性,在丢失关键帧后可迅速找回原来的场景,此外,对关键帧的选择更为合理,因此性能提升较大。相对地,虽然 ORB-SLAM2 算法采用 ORB(Oriented FAST and Rotated BRIEF)作为视觉特征时鲁棒性良好,但是在特征点提取过多时会对每幅图像进行大量的计算,过少时又可能未达到匹配要求,进而影响跟踪线程。

针对上述情况,可以借鉴关注像素点灰度梯度的光流方法^[9],该方法节省了特征提取与匹配的时间,适用于重复纹理区域的检测。虽然该方法易受光照强度的影响,但是 ORB 视觉特征具有良好的光照不变性,因此,本文对两种方法进行结合,在 ORB 算法提取特征点后,使用光流法对特征点进行

跟踪,随后匹配有效特征点并计算相机的位姿。通过融合的方式进行优化,既保持了原有提取方法的鲁棒性,也通过光流法对特征点的运动进行了跟踪。

2.2 目标检测算法

目标检测算法由检测窗口选择、特征提取、分类器区分 3 个环节组成。通过选择的窗口去除无关区域,一般使用选择性搜索方法提取图像;提取特征点时,常用的视觉特征有 HOG、Harris、SIFT 等^[10];随后便利用支持向量机(SVM)、Adaboost 等分类器进行区分。虽然早期算法中各环节内容明确,但限制了环节间的优化学习。相比于深度学习网络,传统方法的性能较低。

如今主流的检测方法有 Faster R-CNN、YOLO、SSD^[11]等。为了检测细小物体的同时尽可能保证算法运行的时效性,选择 YOLO 算法进行目标检测从而获取物体的语义信息。其利用输入图像的信息来预测目标的类别并选定物体的边框。对应的算法流程如图 1 所示,先将输入图像划分为 $S \times S$ 个栅格;再在神经网络中预测各个栅格的完成类别概率和 B 个边框的置信度;最后通过非极大值抑制^[12]剔除多余预测。

作为 YOLO 目标检测算法更新的第三个版本,由 Redmon 等^[13]设计的 YOLOv3 算法在提升准确率的同时也保留了算法的实时性。同 YOLOv1, YOLOv2 网络结构相比,YOLOv3 使用多尺度方法来预测边框。在基础网络中,卷积层的预测框信息输出至尺度一;对卷积层进行上采样后,将其与 16×16 大小的特征图相加,因此尺度二是尺度一的 2 倍;类似地,尺度三使用的特征图大小为 32×32 。因此,网络最终输出的维度为 $S \times S \times [3 \times (B \times 5 + C)]$,其中 C 为对应数据集中物体类别的数量。三个尺度的预测保证了 YOLOv3 网络对微小物体的检测,其网络结构如图 2 所示。

除了多尺度预测的优势,YOLOv3 算法在进行类别预测时使用的损失函数为交叉熵损失函数,适用于场景的复杂检测^[13]。网络的损失函数表达式为

$$l = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in C} [p_i(c) - \hat{p}_i(c)]^2, \quad (1)$$

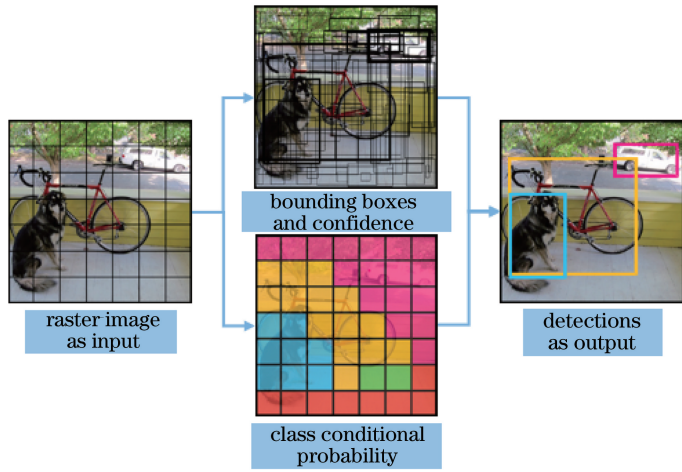


图 1 YOLOv3 算法流程
Fig. 1 Flow chart of YOLOv3 algorithm

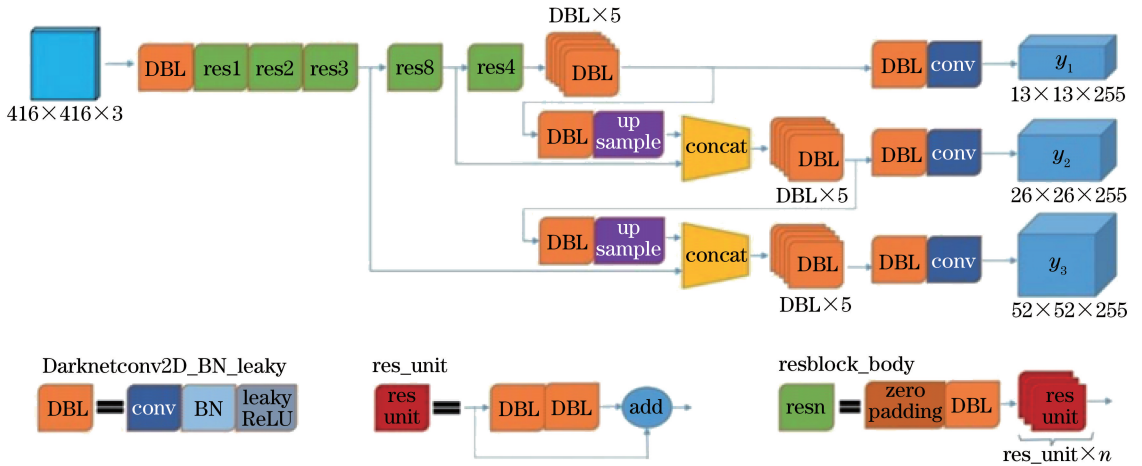


图 2 YOLOv3 结构
Fig. 2 Structure of YOLOv3

式中： I_{ij}^{obj} 为第 i 个栅格的第 j 个预测框检测到的物体； I_{ij}^{noobj} 为目标未落入预测框；对物体预测的惩罚权重 λ_{coord} 设为 5.0, λ_{noord} 设为 0.5, 以减小不包含物体栅格的交并比 (IoU) 误差； x_i, y_i 为预测中心的位置； w_i, h_i 分别为边框的宽和高； $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ 为坐标信息的标注值； C_i 为预测的置信度， \hat{C}_i 为对应的标注值； $p_i(c)$ 为预测框属于类别 c 的概率； $\hat{p}_i(c)$ 为标记框所属类别的真实值。通过 (1) 式可以看出，该损失函数由预测坐标、IoU 及预测类别 3 种误差组成^[13]。

3 构建三维语义地图

3.1 三维点云分割

通常把几何特征作为分割的依据,以减少与点云有关的纹理信息对分割结果的影响。为了优化分割

效果,一般在点云分割前进行预处理。按照点云提取信息的类型,把分割方法划分为基于点、剖面、体素三种类别方法^[14]。其中,基于点的分割方法大多根据 K 最邻近邻域关系完成点特征的确定,但是最优邻域选择有着较高的时间代价;基于剖面的方法通常和激光雷达生成的点云结合,但是剖面覆盖的邻域范围有所限制;因此考虑将体素作为基本单元的方法进行提取,从而可以较好地判定点云中线条、面或者球体分布的情况。超体聚类的示例如图 3 所示。

利用超体聚类进行预处理,根据点云中点的相似程度聚类以生成较小的表层面,同时降低了计算复杂度^[15]。预分割的结果以邻接图 $G = \{v, e\}$ 的形式显示, $v_x \in v$ 是表层面集合中的元素, e 完成相邻 v_x 和 v_y 两面的连接,预处理后,空间场景的分割便可以简化为图分割。因为聚类后的表层面包含一个

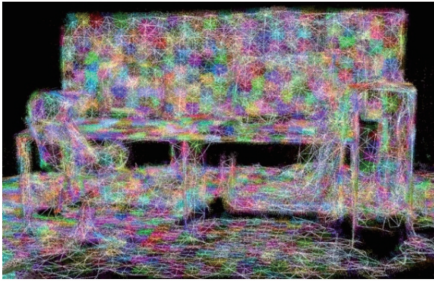


图3 超体聚类

Fig. 3 Supervoxel clustering

质心和一条法向量,将表层面作为节点单元,以保证所有节点都来源于同一物体,具体如图4红色标记所示,即依据节点的相似程度对相邻节点间的公共边进行区分。例如,常见的 Mean-shift 聚类算法^[16]通过计算节点间欧氏距离便可以反映表层面间的相似性,但计算量有所增加。上述聚类在高噪声比的点云中并不鲁棒,考虑到表层面上保留的法向量可以反映局部的凹凸信息,引入支撑平面进行优化。假定点云中存在 M 个支撑平面,预处理阶段获得的表层面和支撑平面集合同时存在,只需要把物体的支撑平面提取出来然后同对应表层面逐个匹配即可。一般地,定义变量 $\{p_i\}_1^N, p_i \in [0, M]$, 以 $p_i = M$ 表示属于支撑平面集合 S_M 的表层面, N 为分割物体对应的表层面总个数。

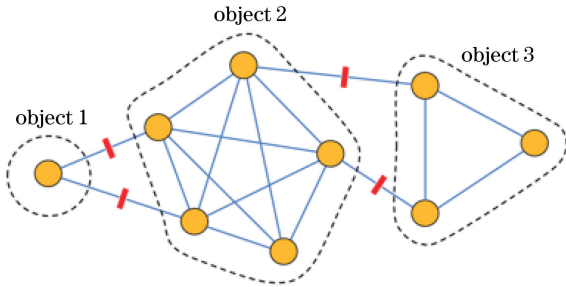


图4 表层面的图分割

Fig. 4 Graph segmentation of surface patches

获取表层面后,通过随机抽样处理表层面进而扩充为平面候选集,把 M 个平面构成的候选集记为 $H = \{p_1, p_2, \dots, p_m\}$, 然后计算表层面的质心 c_x 到平面 p_m 的距离,即 $d(c_x, p_m)$ 。引入阈值 σ , 通过设定参数得到和候选平面 p_m 距离 σ 以内的表层面, 以集合 $\Gamma_{p_m} = \{v_x \in V | d(c_x, p_m) < \sigma\}$ 包含所有符合条件的表层面。从 H 提取平面的问题可以用最小能量函数表示:

$$P^* = \operatorname{argmin} E(P), P \in H, \quad (2)$$

$$D(p_m) = \begin{cases} 1 - \frac{\Gamma_{p_m}}{\eta}, & \Gamma_{p_m} < \eta \\ \exp\left(1 - \frac{\eta - \Gamma_{p_m}}{\eta}\right), & \Gamma_{p_m} \geq \eta \end{cases}, \quad (3)$$

式中:能量函数 $E(P)$ 负责平面拟合程度的评定,其大小为候选平面集 $D(p_m)$ 的总和; $D(p_m)$ 负责平面精确度的评定; η 为边界阈值。在实验中,令 $\eta = 50, \sigma = 5.0$ cm, 这表明一个合格的平面至少有 50 个表层面,同时候选平面到表层面间的距离设定为 5.0 cm。完成上述设置后,得到平面集合 P 和表层面集合 M , 采用图割法使能量函数值最小^[17], 以保证表层面到平面集的分配。

3.2 语义信息标记

ORB-SLAM2 算法构建的点云地图由无序的空间点组成,并不包含物体的语义信息,因此需要系统构建语义标记的框架,具体如图5所示。初始地图由 SLAM 线程建立, YOLOv3 检测算法设置在另一线程中,并对 ORB-SLAM2 算法中的关键帧进行物体检测,如图5(a)所示。图5(b)中框定的目标即可作为图像中物体的标签,将其投影到点云中,使得标记后的点云包含坐标和彩色信息,如图5(c)所示。

此外, YOLOv3 对关键帧的检测对应 ORB-SLAM2 算法的跟踪线程,其图像中各物体的检测边框通过深度图映射到空间点云中,并设置颜色进行标注,进而构建粗略的语义地图。尽管在图像中,如显示器、键盘等物体的边框并不能够精准地反映物体实际,但构建的地图也大致包含了物体的语义信息。为了提升地图的精度,可以在点云分割的基础上,把点云片段和物体的语义标签结合。一般从 RGB 图像中获取原始语义标签,再利用图像标签和点云分割的结果对语义信息进行划分,使点云中每个物体所在的区域匹配到指定的语义信息。

4 实验结果分析

选择 TUM 数据集^[18] 对语义地图构建方法进行评价,并设置几种已知的物体来验证语义信息的有效性。考虑到训练 YOLOv3 的 coco 数据集较大,因此物体类别的选择应同实验场景的检测相符合。系统运行平台为 Ubuntu 16.04 64 位操作系统的笔记本电脑, i5-8300H (8 核 @ 2.30GHz) CPU, 8G DDR4 RAM 和 GTX 1060Ti。使用 GPU 加速后, YOLOv3 框架的检测速率能够达 8 frame/s, 同时 SLAM 模块实时性保持较好。

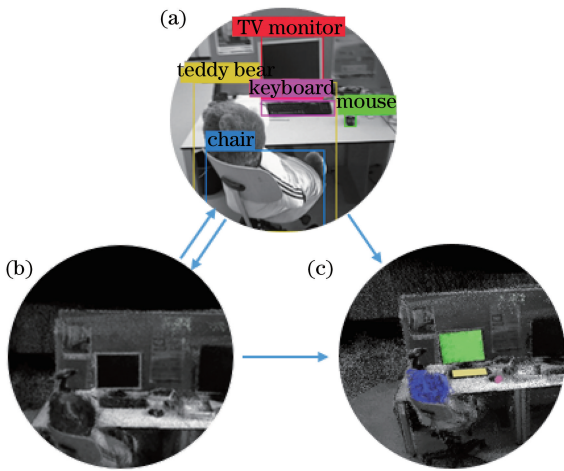


图5 物体信息的标记。(a)初始点云;(b)目标检测;(c)语义标记

Fig. 5 Object information label. (a) Original point cloud; (b) object detection; (c) semantic label

4.1 ORB 与光流融合

对 ORB 特征提取进行优化处理,融合过程如

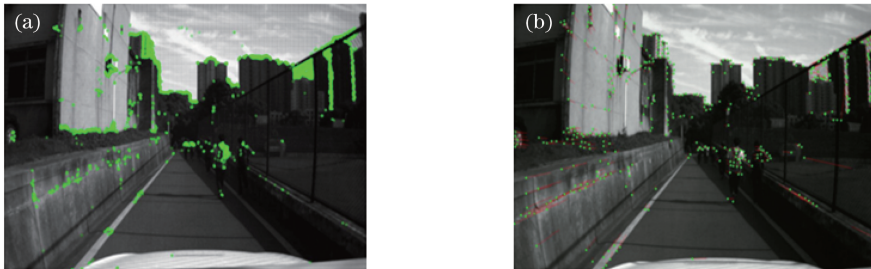


图6 ORB 特征点提取优化。(a)原始 ORB 特征点提取;(b)光流跟踪 ORB 特征点

Fig. 6 Optimization of ORB characteristic point extraction. (a) Original ORB characteristic point extraction; (b) optical flow tracking ORB characteristic point

4.2 有效点云分割

为了定量地评价分割方法,对所提算法和 PCL 点云库中常用的分割算法的分割结果进行比对,引入利用邻近信息的区域生长算法 1 和基于局部凹凸性的算法 2 作为对比项,从运行速率和点云分割效果两个方面进行分析。

表 1 不同算法的运行时间

Table 1 Running time of different algorithms

Algorithm	Running time /ms
Algorithm 1	18073
Algorithm 2	9825
Proposed algorithm	10531

从表 1 可以看出,所提算法的时间代价较低,比算法 1 消耗更少的时间。提供比对的点云由 ORB-SLAM2 算法生成,场景的选取来源于 TUM 数据集 freiburg3_sitting,其包括 800 对 RGB 图像和深

度图像;首先设置一个动态阈值 ξ ,保证在提取过程中可以获得较多的 FAST 角点;之后对输入图像进行 $s \times s$ 像素区域的划分,通过非极大值抑制对区域内的角点进行过滤;保留前 n 个具有最大 Harris 响应值的角点。每次跟踪特征点之后需要将结果同 ξ 进行判断,以防止相机大尺度漂移丢失特征点。在上述判定结束后,便停止对特征点的跟踪,重新利用 ORB 特征提取当前帧,以此作为循环单元。ORB 特征点优化结果如图 6 所示,选取校园辅路环境进行实验,并对融合后的方法和原有提取效果进行比对。

由图 6(a)发现,ORB 算法虽然保证了特征点提取的数量,但是在图像边缘区域冗余严重,在光线视角良好的区域中特征点聚集较少。与之对比,ORB 特征点提取与光流跟踪融合的实时分布效果显得更为合理,如图 6(b)所示,没有出现特征点堆积的情况,提取的特征点均匀分布在图像区域内,同时可以发现,优化后光流场的引导方向与运动方向一致,达到了实验设计的理想效果。

度图像,因此可以构建百万量级的空间点云。

在完成点云分割后,需要对处理后的点云进行分析。点云分割比较如图 7 所示,在相同的 RGB-D 图片数据集上将不同算法的分割结果展示出来,从而便于比较不同算法之间的细分特性。

图 7(a)中,算法 1 的点云分割实质是以邻近点间的相似信息作为判定标准,再通过设定的阈值进行聚类,但是噪声数据过多时会对分割结果产生负面影响,算法 1 在分割时未能很好地区分桌面和墙面。基于局部凹凸性的算法 2 尽管在分割时间上小于所提算法,但是从图 7(b)可以看出,其对场景中细小物体如键盘、鼠标之类的提取有一定的难度,这与局部的凹凸信息有关,最后分割的精确度降低。所提算法不仅降低了时间成本,而且使用支撑平面的几何信息降低噪声对分割的影响,从全局的角度完成分割性能的提升,如图 7(c)所示,所提算法可以完成点云中大部



图7 点云分割结果比较。(a)算法1;(b)算法2;(c)所提算法

Fig. 7 Comparison of point cloud segmentation results. (a) Algorithm 1; (b) algorithm 2; (c) proposed algorithm

分室内物体的细小物体分割,从初始点云中获取有效点云段,进而便于最终语义地图的构建。

4.3 语义地图构建结果

选用 TUM 数据集进行验证,并且在物体几何信息精确度和算法时间复杂度方面进行评估,从而对本系统有一个较为整体的评价。

实验选用 freiburg3_long_office 场景进行最终客观地图的建立,语义地图构建的运行示例如图 8(a)所示,可以实时记录相机轨迹的运动情况。ORB-SLAM2 算法获取的原始地图和系统构建的语义地图分别如图 8(b)和图 8(c)所示,通过对比可以看出,系统在建立语义地图的同时校正了部分点云的过分割、误分割,并且对 ORB-SLAM2 点云地图中的模糊细小物体进行了合理的标注。

完成原始地图和语义地图的几何信息比较后,对 TUM 数据集构建的地图信息进行时间方面的分析,并且添加与系统相似的主流方法作为参考,具体比对结果如表 2 所示。由于在语义信息标注前进行了点云分割处理,语义地图中所包含的空间点大致为 ORB-SLAM2 构建地图的 2 倍,同时语义地图的大小也明显大于原始地图,这反映了物体信息标签的映射和部分信息的冗余。最后,由于 YOLOv3 算法对关键帧的平面物体进行了检测和框定,系统对两个场景的单帧平均处理时间相较所提算法慢了 0.05 s 左右,但是比文献 [19]中的算法快。总体来说,语义地图构建系统牺牲了部分算法的运行速度,但是保证了信息标注的准确。



图8 语义地图验证。(a)系统实时运行;(b)原始地图;(c)语义地图

Fig. 8 Verifying semantic map. (a) System operating in real-time; (b) original map; (c) semantic map

表2 地图构建实验结果对比

Table 2 Comparison of experimental results of map construction

Algorithm	freiburg3_sitting			freiburg3_long_office		
	Number of points	Map size /Mbit	Time /s	Number of points	Map size /Mbit	Time /s
Proposed algorithm	2106545	73.5	0.0871	5467152	151.3	0.1318
ORB with YOLOv3	4503071	91.4	0.1295	9840873	190	0.1841
ORB with Mask R-CNN ^[19]	4382232	86.3	0.1734	9165934	177.8	0.2506

5 结 论

针对移动端 SLAM 传统算法在点云地图构建过程中缺乏语义信息的问题,在视觉 SLAM 的基础上,提出一种结合目标检测的语义地图构建方法。该方法利用优化后的 ORB-SLAM2 算法获取几何精确信息的同时,对其与 YOLOv3 网络识别的物体

语义信息进行融合;另外提出一种合理的点云分割方法,设置多组实验,验证了分割方法的有效性;最后建立物体信息标签和分割后点云段的映射匹配关系。将 RGB 信息和深度信息作为输入,在 TUM 数据集上进行实验,结果表明,系统可以建立全局一致的语义点云地图并且保留了键盘、鼠标等细小物体的语义信息。实验过程中设置多个场景进行比对,

验证了所提方法具有良好的鲁棒性。

参 考 文 献

- [1] Whelan T, Salas-Moreno R F, Glocker B, et al. ElasticFusion: real-time dense SLAM and light source estimation[J]. The International Journal of Robotics Research, 2016, 35(14): 1697-1716.
- [2] McCormac J, Handa A, Davison A, et al. SemanticFusion: dense 3D semantic mapping with convolutional neural networks[EB/OL]. (2016-09-28) [2019-11-10]. <https://arxiv.org/abs/1609.05130>.
- [3] Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(1): 142-158.
- [4] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.
- [5] He K M, Gkioxari G, Dollár P, et al. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386-397.
- [6] Stein S C, Schoeler M, Papon J, et al. Object partitioning using local convexity[C] // 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE, 2014: 304-311.
- [7] Charles R Q, Su H, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation[C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 77-85.
- [8] 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, 2017, 33(5): 1255-1262.
- [9] Oron S, Bar-Hillel A, Avidan S. Extended Lucas-Kanade tracking[M] // Fleet D, Pajdla T, Schiele B, et al. Computer vision - ECCV 2014. Lecture notes in computer science. 2014, 8693: 142-156.
- [10] Nixon M S, Aguado A S. Feature extraction and image processing[M]. Yang G B, Li S Y, Transl. 3rd ed. Beijing: Electronic Industry Press, 2010: 127-133.
Nixon M S, Aguado A S. 特征提取与图像处理[M]. 杨高波, 李实英, 译. 3 版. 北京: 电子工业出版社, 2010: 127-133.
- [11] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[M] // Leibe B, Matas J, Sebe N, et al. Computer vision - ECCV 2016. Lecture notes in computer science. 2016, 9905: 21-37.
- [12] Guo J X, Liu L B, Xu F, et al. Airport scene aircraft detection method based on YOLOv3[J]. Laser & Optoelectronics Progress, 2019, 56(19): 191003.
郭进祥, 刘立波, 徐峰, 等. 基于 YOLOv3 的机场场面飞机检测方法[J]. 激光与光电子学进展, 2019, 56(19): 191003.
- [13] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08) [2019-11-10]. <https://arxiv.org/abs/1804.02767>.
- [14] Li Y, Tong G F, Yang J C, et al. 3D point cloud scene data acquisition and its key technologies for scene understanding[J]. Laser & Optoelectronics Progress, 2019, 56(4): 040002.
李勇, 佟国峰, 杨景超, 等. 三维点云场景数据获取及其场景理解关键技术综述[J]. 激光与光电子学进展, 2019, 56(4): 040002.
- [15] Papon J, Abramov A, Schoeler M, et al. Voxel cloud connectivity segmentation - supervoxels for point clouds[C] // 2013 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2013, Portland, OR, USA. New York: IEEE, 2013: 2027-2034.
- [16] Derpanis K G. Mean shift clustering[EB/OL]. (2005-05-26) [2019-11-10]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.217.3313>.
- [17] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(11): 1222-1239.
- [18] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C] // 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, 2012, Vilamoura, Portugal. New York: IEEE, 2012: 573-580.
- [19] Wu H, Chi J X, Tian G H. Instance recognition and semantic mapping based on visual SLAM[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2019, 47(9): 48-54.
吴皓, 迟金鑫, 田国会. 基于视觉 SLAM 的物体实例识别与语义地图构建[J]. 华中科技大学学报(自然科学版), 2019, 47(9): 48-54.