激光与光电子学进展 先进成像

面向室内动态场景的视觉同时定位与地图构建语义八叉树地图构建方法

张荣芬*，袁文昊，卢金，刘宇红
贵州大学大数据与信息工程学院，贵州 贵阳 550025

摘要 针对传统视觉同时定位与地图构建(vSLAM)系统在动态场景中无法有效去除运动物体及缺少可用于更高层应用的语义地图等问题，提出了一种可有效去除动态物体并构建表征室内静态环境的语义八叉树地图的vSLAM系统方法。首先，使用Fast-SCNN作为语义分割网络提取图像的语义信息，同时，利用金字塔光流法对特征点进行跟踪匹配。然后，使用步进随机抽样一致算法(Multi-stage RANSAC)通过多次执行不同尺度的RANSAC流程对特征点进行步进采样，再利用对极几何约束并结合Fast-SCNN提取的语义信息进行视觉里程计动态特征点剔除。最后，通过体素滤波降低点云冗余后构建纯静态环境的语义八叉树地图。实验结果表明：所提方法在公用数据集TUM RGB-D的8个RGB-D高动态序列中测试的相机相对位移误差、相对旋转误差和全局轨迹误差相较于ORB-SLAM2系统有94%以上的提升，全局轨迹误差仅为0.1 m；相较于同类DS-SLAM系统，动点剔除总耗时有21%的缩减。建图性能方面，经体素滤波后构建的语义点云地图与语义八叉树地图分别占据9.6 MB、685 kB的存储空间，相较于17 MB的原始点云，语义八叉树地图仅占用其4%的存储空间并因含有语义可用于更高层次的智能交互任务。

关键词 同步定位与地图构建；动态点剔除；语义分割；步进随机抽样一致算法；体素滤波；语义八叉树地图
中图分类号 TP391；TP18 **文献标志码** A **DOI**：10.3788/LOP202259.1811003

Visual Simultaneous Localization and Mapping Method of Semantic Octree Map Toward Indoor Dynamic Scenes

Zhang Rongfen*, Yuan Wenhao, Lu Jin, Liu Yuhong
College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, *Guizhou, China*

Abstract Aiming at the problems that traditional visual simultaneous localization and mapping (vSLAM) systems cannot remove moving objects in dynamic scenes effectively and lack semantic maps for high-level interactive applications, a vSLAM system scheme was proposed. The scheme can remove moving objects effectively and build semantic octree maps representing indoor static environments. First, Fast-SCNN was used as a semantic segmentation network to extract semantic information from images. Meanwhile, a pyramid optical flow method was used to track and match feature points. Then, for step sampling of the feature points, a stepping random sampling consistent algorithm (Multi-stage RANSAC) was used to perform the RANSAC process on different scales several times. Later, the epipolar geometry constraint and semantic information extracted from the Fast-SCNN were combined to remove the dynamic feature points of the visual odometer. Finally, the semantic octree map representing the static indoor environment was built by the point cloud after using voxel filtering to reduce redundancy. Experimental results show that the performance indicators of a camera, including relative displacement, relative rotation, and global trajectory errors in the 8 RGB-D high dynamic sequence of common datasets TUM RGB-D, are improved by more than 94% compared with the ORB-SLAM2 system, and the global trajectory error is only 0.1 m. Compared with a similar DS-SLAM system, the total time for eliminating a moving point is reduced by 21%. After voxel filtering, the semantic point cloud and octree maps occupy 9.6 MB and 685 kB storage space, respectively, in terms of map construction performance. Compared with the original point cloud of 17 MB, the semantic octree map occupies only 4% of the storage space; therefore, it could be used for high-level intelligent interactive applications due to its semantics.

Key words simultaneous localization and mapping; moving point elimination; semantic segmentation; stepping random sampling consistent algorithm; voxel filtering; semantic octree map

收稿日期：2021-05-21；**修回日期**：2021-06-16；**录用日期**：2021-06-16
基金项目：贵州省科学技术基金(黔科合基础-ZK[2021]重点001)
通信作者：*rfzhang@gzu.edu.cn

1 引 言

随着机器人技术的快速发展，移动机器人已初步应用于航空航天、医疗健康、智慧农业、家庭护理、自动驾驶、智能家居等领域。要在未知环境中实现自主导航、路径规划乃至于更高层次的人机交互任务，机器人需要对所在环境进行信息感知与理解并构建对应的模型。视觉同时定位与地图构建（vSLAM）是解决这一问题的主流技术方案之一。

早在 2007 年，Klein 等[1]提出的 parallel tracking and mapping（PTAM）模型首次将 vSLAM 系统分为前端和后端，前端负责相机跟踪，后端负责建图优化。直到 2015 年，Mur-Artal 等[2]提出了 ORB-SLAM，在 PTAM 的基础上进一步将系统分为跟踪、局部建图、闭环检测等三个线程。三线程的架构及 ORB 特征的使用保证了 ORB-SLAM 的实时性。随后 Mur-Artal 等[3]于 2017 年又提出了 ORB-SLAM2，其完备的系统架构与代码设计将视觉 SLAM 的发展推向了新的高峰。2021 年，Campos 等[4]提出了 ORB-SLAM3，ORB-SLAM3 将视觉特征信息与惯性传感器信息紧密耦合，仅依靠最大后验估计便能在小型和大型室内外场景中实时稳定运行。尽管上述基于视觉的 ORB-SLAM2、ORB-SLAM3 等经典 SLAM 框架在静态刚体环境中拥有杰出的表现，但在动态环境下却无法去除运动物体带来的干扰。并且由于缺乏语义信息，搭载这些 vSLAM 系统的机器人无法完成更高层次的环境感知和智能交互任务。近年，随着深度神经网络（DNNs）在场景理解和语义分割方面的发展，一些学者将深度学习与传统 vSLAM 相结合以提升 vSLAM 系统在动态环境下的鲁棒性和精确度，如 DS-SLAM[5]、DynaSLAM[6]、DP-SLAM[7]等，但仍然存在较大的提升空间。并且在地图构建方面，这类与深度学习相结合的框架只构建了环境的稀疏点云的表征形式，而点云地图存在占用存储空间大、无法有效表示空间的占有状态的问题，导致难以应用于大型场景、路径规划及高层次的人机交互等方面，精简的语义八叉树地图（Octomap）能为解决上述问题提供潜在的思路与方法。

针对上述问题，本文提出了一种基于 ORB-SLAM2 架构的 vSLAM 方法。一方面，在使用金字塔 LK 光流法对特征点进行跟踪匹配的同时基于步进随机抽样一致算法（Multi-stage RANSAC）进行特征点筛选并结合 Fast-SCNN 语义分割网络获取的语义信息与对极几何剔除动态特征点。另一方面，将所得语义信息与 vSLAM 的建图模块结合，高效构建室内静态环境的语义八叉树地图。在视觉里程计的追踪线程，对 Multi-stage RANSAC 算法的两个阈值 τ_1 与 τ_2 进行了更充分更细致的实验测试优化，最终将第 1 个较大的阈值 τ_1 设置为 1，第 2 个较小的阈值 τ_2 设置为 0.2，使得在尽量保留内点的同时剔除外点的效果达到最优。在语义分割线程，引入 Fast-SCNN 作为语义分割网络，相较于经典 DS-SLAM 架构及文献[8]中使用的 SegNet，其语义分割的精度有较大的提升。在地图构建线程，体素滤波的方法在保留有效点云的同时极大地降低了全局点云的冗余，也极大地减少了点云存储占用的空间。实验结果表明，得益于 Multi-stage RANSAC 算法与 Fast-SCNN 语义分割网络的引入，所提方法对动态特征点的剔除更加全面，保留的静态特征点更为纯净，从而可构建更精简、精确度更高、泛化能力更强的语义八叉树地图。语义信息的加入，也使得机器人具备了高层次交互的能力，为执行智能路径规划导航与其他交互任务奠定了基础。

2 系统框架

经典的 vSLAM 框架由数据采集、前端视觉里程计、后端优化、回环检测、地图构建 5 部分组成，但这些框架是针对静态场景构建地图的，若实际环境中存在动态物体（比如行人等）则无法去除，导致无法构建准确的地图以表征环境。

所提 vSLAM 系统在 ORB-SLAM2 框架的基础上进行扩展，在增加动点剔除功能的同时，融合了语义信息实时构建可表征静态环境的三维（3D）语义八叉树地图。所提系统架构如图 1 所示。

受到文献[5]中 DS-SLAM 的启发，所提系统对相机采集的每一帧 RGB 图像进行多线程处理。每一帧图像将同时送入语义分割线程和追踪线程。在语义分割线程中，利用 Fast-SCNN 提取图像语义信息。在追踪线程，先提取图像的 ORB 特征点，利用金字塔 LK 光流法将这些特征点与上一帧进行跟踪匹配。接着，利用 Multi-stage RANSAC 算法对特征点进行筛选，去除干扰点或噪声点，计算出较为可靠的基础矩阵 $\boldsymbol{F}$。最后，利用对极几何约束对所有匹配特征点对进行运动一致性验证。经上述处理后再结合语义分割线程输出的语义信息剔除图像中的动点。在建图方面，当图像中的特征点匹配对数量达到初设的阈值时，系统将生成关键帧并送入建图线程和回环。建图线程中，系统首先将关键帧对应的深度图与附有语义信息的关键帧相结合并利用相机内参将当前帧的静态特征点投影到空间坐标中逐像素生成 3D 语义点云地图，然后再利用概率模型[9]与 OctoMap 库将语义点云地图转换成语义八叉树地图。

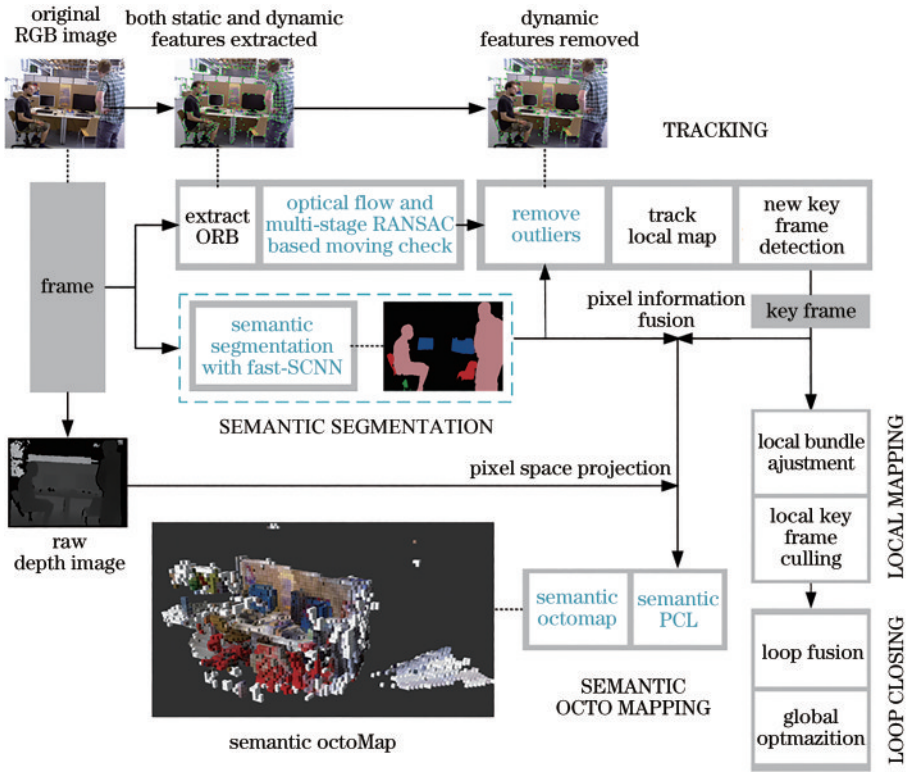

图 1　所提 vSLAM 系统结构

Fig. 1　Structure of proposed vSLAM system

2.1　动态场景下的视觉里程计

2.1.1　动态特征点检测

在语义分割线程，本实验组采用 Fast-SCNN 提取原始图像语义信息。Fast-SCNN 是东芝欧洲研究中心 Poudel 等[10]提出的轻量型语义分割网络，其网络架构如图 2 所示。该网络主体采用与 BiSeNet[11]类似的双支（two-branch）结构，受编码与解码结构的 SegNet[12]等网络的启发，Fast-SCNN 在双支结构的前面添加了类似于编码与解码结构中残差连接层（skip connection）的学习降采样层（learning to down-sample），具有良好的实效性、鲁棒性和灵活性。

为了将上述 Fast-SCNN 应用于所提 vSLAM，本实验组在 Pascal VOC 数据集[13]上对网络进行了训练。图 3 为 Fast-SCNN 语义分割效果，其中不同颜色代表不同的类别，且人体和椅子重叠部分已完整地分割开来，为改善动点剔除性能提供了保障。

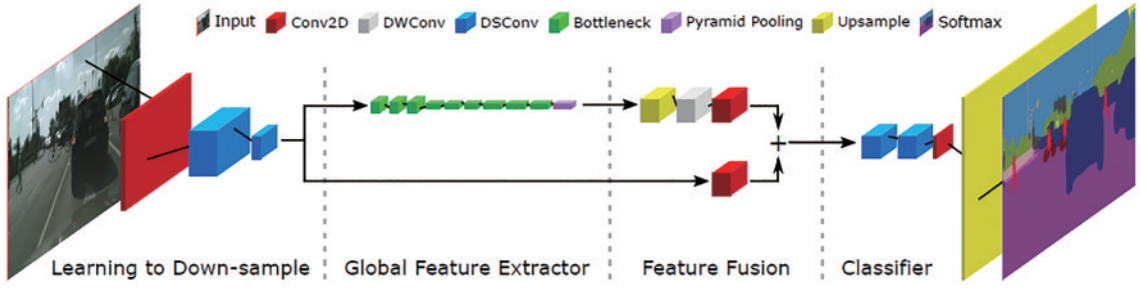

图 2　Fast-SCNN 的网络架构[10]

Fig. 2　Network structure of FAST-SCNN[10]

在追踪线程，传统 ORB 描述子的提取与匹配较为复杂且效率不高，区别于点线特征匹配的方法[14-16]，本实验组采用金字塔 LK 光流法对特征点进行追踪匹配，再利用文献[8]的 Multi-stage RANSAC 对特征匹配对进行筛选。Multi-stage RANSAC 是针对标准 RANSAC 存在当动态物体占据相机大部分视野时精度会明显降低的缺陷[17]而提出的，首先设置一个较大的阈值 τ_1 对特征点执行 RANSAC 流程，然后在已经被判定为内点的点集内再次执行一次较小阈值 τ_2 的 RANSAC 流程，以达到再次优化的目的。Multi-stage RANSAC 算法如表 1 所示。

关于阈值 τ_1 与 τ_2 的选择，本实验组对标准 RANSAC 和 Multi-stage RANSAC 算法设置不同的阈值进而对输出的内点数、外点数与整体时间消耗

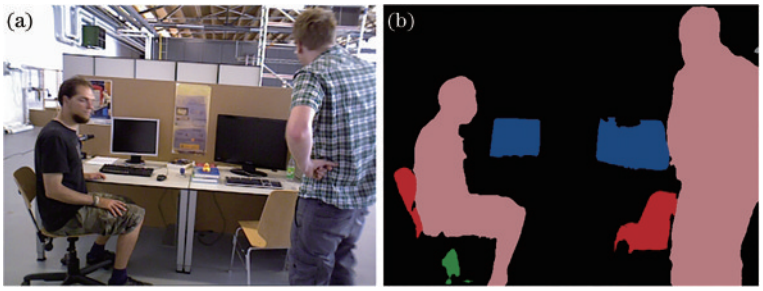

图 3　语义分割效果。(a)原图;(b)分割结果

Fig. 3　Result of semantic segmentation. (a) Original drawing; (b) segmented output

表 1　Multi-stage RANSAC 算法

Table 1　Multi-stage RANSAC algorithm

Algorithm Multi-stage RANSAC Algorithm
Input: Previous Frame, F_1; KeyPoints of F_1, K; Current Frame, F_2; The minimum number of points required, m
Output: The outliers, O; The inliers, I
1: assign K to I
2: **for** the current stage number i is less than n
3:　　compute the maximum number required for iterations, N
4:　　select randomly m points from I
5:　　**while** the number of iterations is less than N
6:　　solve for F
7:　　determine the portion of inliers among K using F
8:　　**if** the portion of inliers is larger than τ_i
9:　　**leave the loop**
10:　　**end if**
11:　　**end while**
12:　　assign new inliers to I
13: **end for**

进行了统计,结果如表 2、3 所示。在表 2 的表头中 Ins. 表示剩余内点数目,Outs. 表示残留的外点数目,Time 表示实验消耗的时间。在表 3 的表头中,Ins.$_1$、Ins.$_2$分别表示经过阈值τ_1和阈值τ_2后剩余的内点数目。从表 2 可知:阈值τ越小,RANSAC 对于动点的剔除效果越好,而将内点误判为外点的概率也就越大,同时所需的计算量也越大;当阈值为 0.1 时,图像中得到保留的内点数为 168,仍残留在图中的外点数目为 6,而最终所消耗时间为 8.2 ms。从表 3 第 1 行可以看出,若对所有特征点先进行一次阈值为 1 的 RANSAC 流程,再对已经判定为内点的特征点执行阈值为 0.1 的 RANSAC 流程,则最终可节省将近一半的时间,这样会达到更好的外点剔除效果,但是同时也会导致更多的内点被剔除;而将第 2 步的阈值增加为 0.2,则仅耗费约 1/10 的时间。所提 Multi-stage RANSAC 可以在保留更多内点的同时,使得残留在图中的外点更低,如表 3 第 2 行所示。实验结果表明,Multi-stage RANSAC 算法在将第 1 个较大的阈值τ_1设置为 1 且第 2 个较小的阈值τ_2设置为 0.2 时,在尽量保留内点的同时剔除外点的效果达到最优。

表 2　RANSAC 实验结果

Table 2　Test results of RANSAC algorithm

τ	Ins.	Outs.	Time /ms
0.1	168	6	8.22814
0.2	228	12	1.02447
0.3	260	13	0.60222

表 3　Multi-stage RANSAC 实验结果

Table 3　Test results of Multi-stage RANSAC algorithm

τ_1	τ_2	Ins.$_1$	Ins.$_2$	Outs.	Time /ms
1	0.1	294	159	5	4.99953
1	0.2	294	210	4	0.83729
1	0.3	294	238	11	0.48140
2	0.1	327	159	7	9.83135
2	0.2	327	211	9	1.63077
2	0.3	327	238	11	0.69094

2.1.2　动态特征点剔除

动点剔除算法如表 4 所示,经过 Multi-stage RANSAC 与对极几何的运动状态判定后可以得到外点集O,进一步结合语义分割信息得到运动物体的语义标签M,然后遍历外点集O中的每一个点,若该点位于运动物体的语义标签范围内则判定为动点并将其

表 4　动点剔除算法

Table 4　Moving point removing algorithm

Dynamic Points Removing Algorithm
Input: Dynamic Points, O; Semantic mask of most likely moving objects, M; KeyPoints, K
Output: The set of inliers, I
1: **if** M not empty **then**
2: **for** point o, m in O, M **do**
3:　　**if** $o = m$ **then**
4:　　remove M from K
5:　　**leave the loop**
6:　　**end if**
7: **end for**
8: **end if**

剔除，反之则继续保留。

动点剔除效果如图 4 所示。从图中可以看出，该算法对位于人体身上绝大部分的特征点进行了剔除，仅剩下静态特征点用于后续的地图构建。

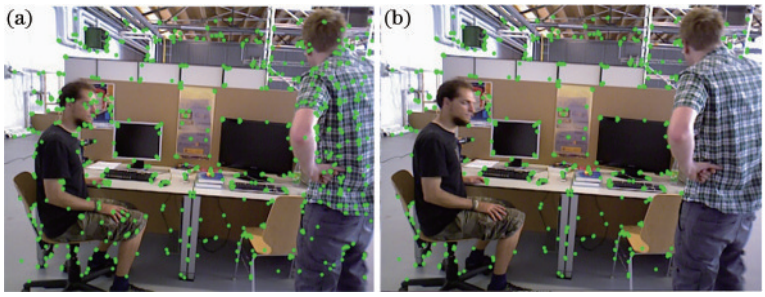

图 4　动点剔除。(a)所有的特征点；(b)剔除动点后的效果

Fig. 4　Outliers removing. (a) All feature points; (b) effect after removing moving points

2.2　语义八叉树地图构建

地图构建是 vSLAM 的两项主要任务之一，针对传统的点云地图存在占用存储体积大、难以更新和优化、不包含语义信息且无法直接用于导航等更高层次的应用等问题，本实验组提出了一种含有语义信息的八叉树地图构建方法。

八叉树地图(Octomap)的核心思想是利用八叉树的数据结构形式构建地图。八叉树是一种三维空间的分层数据结构，基于此结构可以将空间进行无限划分[18-19]。八叉树中的每个节点表示一个体素，其值为该体素所占空间的大小，每一个体素可被分割为八个体积大小一致的体素，以此循环操作直至达到可表示空间的对应体积大小，如图 5 所示，其中白色表示空间未被占据，黑色表示空间被占据不可再分割。

所提语义八叉树地图构建流程如图 6 所示，为了降低冗余只对关键帧进行地图构建，将关键帧与其深度图对应后结合 Fast-SCNN 获取的语义信息首先构建语义点云地图，然后再利用概率模型转换为语义八叉树地图。

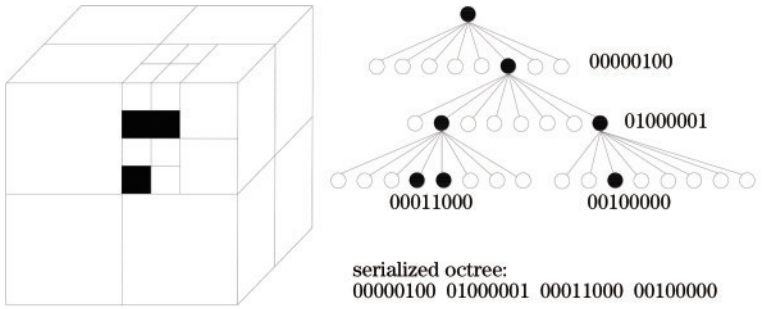

图 5　八叉树结构示意图

Fig. 5　Diagram of octree map

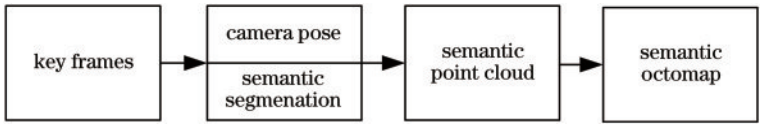

图 6　语义八叉树地图构建流程

Fig. 6　Semantic octree map building process

2.2.1　单帧语义点云构建

当建图线程接收到关键帧时，将其深度图相对应并结合语义分割线程输出的语义信息逐像素进行点云构建。构建过程首先需要利用相机内参将当前帧点云投影到空间坐标中。假设某点 P 的像素坐标为 (u,v)，其对应的深度为 d，该点在三维空间坐标系内坐标为 (X,Y,Z)。根据相机的针孔模型[20]，将空间中坐标点投影到相机成像平面的变换式为

$$\begin{bmatrix} u \\ v \\ d \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{1}$$

式中：等号右边的第一个矩阵为相机内参；f_x、f_y、c_x、c_y 分别表示相机在空间坐标系 X 轴和 Y 轴方向的焦距及像素坐标原点相对空间坐标原点在 X 轴和 Y 轴方向的相对平移量。根据式(1)可知，只需对等式两边同时左乘一个相机内参矩阵的逆矩阵即可得到像素点对应的空

间坐标点位置信息，变换式和最终变换结果为

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ v \\ d \end{bmatrix}, \tag{2}$$

$$\begin{cases} X = d(u - c_x)/f_x \\ Y = d(v - c_y)/f_y \\ Z = d \end{cases}。 \tag{3}$$

对于每一个深度点，首先需要按照结构光相机的最大量程和最小精度判断该点是否有效，如果该点深度值超过了相机最大量程或小于最小精度则判为无效点并舍弃。深度点进行有效性判断后还需判断其所属类别，若某深度点被判断为动态物体比如人类或是其他动物等则不对其构建点云图，若判断所属类别为动态物体以外的其他类别，则构建其点云图并融入对应类别的颜色标签。

2.2.2　语义点云的拼接与滤波

为了使系统构建的地图能够正确表征环境，需要将单帧语义点云地图进行拼接构成全局点云地图，具体拼接实现形式可描述为

$$\boldsymbol{m} = \sum_{k=0}^{n} \boldsymbol{T}_k \boldsymbol{C}_k, \tag{4}$$

式中：$\boldsymbol{m}$ 为所有关键帧拼接的全局点云地图；$\boldsymbol{C}_K$ 表示第 k 帧构建的点云图；$\boldsymbol{T}_K$ 表示第 k 帧相机位姿。

尽管本实验组只选取了关键帧构建地图，关键帧之间仍然存在冗余现象，构建的全局点云地图会出现相互重叠的现象，使得地图无法准确地表征实际环境，同时又增大了点云的存储空间。对于上述问题，采用体素滤波器对全局点云中的重叠点进行滤除。体素滤波器将整个空间分为若干个大小一样且不重合的立方体，只保留位于立方体重心上的点，由此可以保留有效点云的同时极大地降低全局点云的冗余，也极大地减少了点云存储空间的占用。

2.2.3　语义八叉树地图构建与测试

在八叉树空间中，最底层的子叶表示地图的最低分辨率，不同层的节点会保存空间占据信息。对于空间信息最为简单的表示方法是用“0”表示空间未占据，用“1”表示空间被占据。由于语义分割精度的影响，仍会存在动点剔除不完整的迹象，造成子叶空间占据信息的判别不准确，为此本实验组采用多次观测的联合概率统计数值来判定该点的状态。假设用概率 $P \in [0, 1]$ 表示某点的初始状态，对其进行多次观测，若每一次观测该点为被占据则对 P 执行累加操作，反之执行减操作。由于概率 P 可能出现超出 $[0,1]$ 的情况，这里引用概率对数值来表示，即

$$l = \mathrm{log\,it}(p) = \log\left(\frac{p}{1-p}\right)。 \tag{5}$$

假设某节点 n，其对应观测数据为 z，若将该点从初始时刻到 t 时刻的观测概率对数值用 $L(n|z_{1:t})$ 表示，则从初始时刻到 $t+1$ 时刻该节点的概率对数值为

$$L(n|z_{1:t+1}) = L(n|z_{1:t}) + L(n|z_{t+1}), \tag{6}$$

$$L(n|z_{t+1}) = \begin{cases} \tau_{\mathrm{occup}}, & \text{node } n \text{ is observed} \\ 0, & \text{node } n \text{ is not observed} \end{cases}, \tag{7}$$

式中：τ_{occup} 为预先设定的参数，用于计算节点某时刻的概率对数值。

首先使用 Silberman 等在文献[21]中提供的 Dining Rooms 中的 5 张 RGB 图像序列作为构建地图的测试集。测试结果如图 7 所示，其中第 1 行为原始的 RGB 图像，第 2 行为语义分割结果，最后结合相机位姿和对应的深度图构建的原始点云如[图 8(a)]所示，该点云图包含了 1081843 个点云，占用存储空间约 17 MB。按照前文所述的体素滤波方法以 0.01 分辨率对点云进行下采样滤波，最终得到的点云地图包含 627996 个点云，占用存储空间仅为 9.6 MB，滤波后点云如[图 8(b)]所示。体素滤波在有效去冗余的同时将其体积压缩近一倍。

图 7　Dining Rooms 图像序列及其语义分割结果

Fig. 7　Dining Rooms image sequence and its semantic segmentation results

上述滤波后点云图进一步融入语义信息构建的语义点云地图如[图 9(a)]所示，最后转换得到的语义八叉树地图如[图 9(b)]所示，二者分别占据 9.6 MB、685 kB 的存储空间，相较于最初的 17 MB 点云，语义八叉树地图仅占用了其 4% 的存储空间，并且因含有语义使得地图可用于更高层的智能任务。

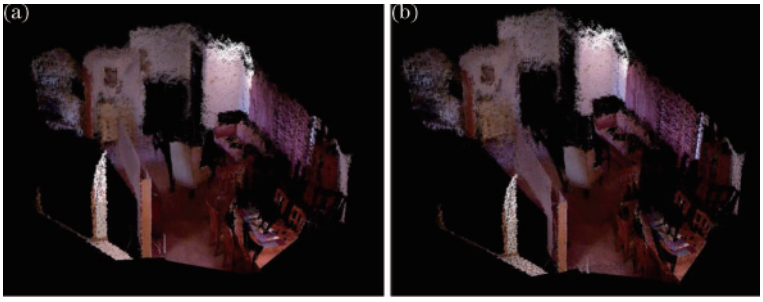

图 8　点云。(a)原始点云;(b)滤波后的点云

Fig. 8　Point cloud. (a) Original point cloud; (b) filtered point cloud

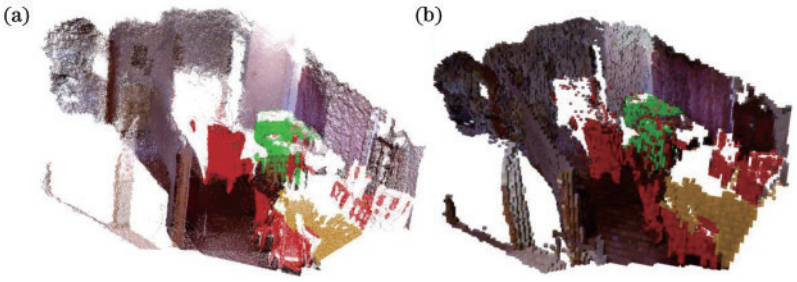

图 9　语义点云和语义八叉树地图。(a)语义点云;(b)语义八叉树地图

Fig. 9　Semantic point cloud and semantic octree map. (a) Semantic point cloud; (b) semantic octree map

3　实验测试与分析

3.1　实验数据集与环境说明

为验证所提系统的实时有效性,选取德国慕尼黑工业大学发布的 TUM RGB-D 数据集[22]中的 8 个 RGB-D 动态帧序列作为测试数据集。所用 CPU 为 Intel I7 10750H,显卡为 NVIDIA RTX 2060,内存为 16 GB;系统为 Ubuntu 16.04。

TUM RGB-D 数据集的 8 个 RGB-D 动态帧序列分别是 fr3_walking_xyz、fr3_walking_rpy、fr3_walking_halfsphere、fr3_walking_static 及 fr3_sitting_xyz、fr3_sitting_rpy、fr3_sitting_halfsphere、fr3_sitting_static。“fr3”表示数据采集的相机代号,“sitting”和“walking”表示物体“坐着”和“行走”两种运动状态,相机分别产生四种不同运动方式,即“平移运动”“旋转运动”“半球形运动”和“静止”,分别用“xyz”“rpy”“halfsphere”“static”表示。为了便于记录,用“WX、WR、WH、WS、SX、SR、SH、SS”分别表示上述 8 个动态帧序列。

评价指标包含相机位姿误差(RPE)和全局轨迹误差(ATE)。RPE 对里程计的相对位移误差和相对旋转误差进行统计,而 ATE 对相机的全局一致性运动轨迹与真实轨迹进行比较。实验过程中对两项评价指标均采用均方根误差(RMSE)和标准差(S.D)进行统计,前者关注估计值与真实值的差距,后者关注自身的离散程度。

3.2　公共数据集测试

3.2.1　与 ORB-SLAM2 的对比

首先将所提 vSLAM 系统与经典 ORB-SLAM2 系统的视觉里程计性能进行对比评估,然后对比其建图效果。表 5～7 分别给出了两个系统在相对位移误差、

表 5　相对位移误差对比

Table 5　Typical value of translation

Seq.	ORB-SLAM2		Proposed system		Improvement	
	RMSE /m	S.D /m	RMSE /m	S.D /m	RMSE /%	S.D /%
WX	0.825981	0.449478	0.019991	0.010374	97.58	97.69
WH	0.502363	0.299615	0.025955	0.012036	94.83	95.98
WR	1.212279	0.676205	0.062167	0.041967	94.87	93.79
WS	0.585281	0.423743	0.009044	0.004097	98.45	99.03
SX	0.013602	0.006688	0.012649	0.006140	7.01	8.19
SH	0.040732	0.023205	0.017660	0.008521	56.64	63.28
SR	0.030898	0.018059	0.021174	0.011048	31.47	38.82
SS	0.012007	0.005570	0.007237	0.003720	39.73	33.21

表 6　相对旋转误差对比

Table 6　Typical value of rotation

Seq.	ORB-SLAM2		Proposed system		Improvement	
	RMSE /deg	S. D /deg	RMSE /deg	S. D /deg	RMSE /%	S. D /%
WX	14. 812930	8. 086117	0. 618651	0. 376314	95. 82	95. 35
WH	13. 379170	7. 277847	0. 749063	0. 354548	94. 40	95. 13
WR	22. 021472	12. 858064	1. 201734	0. 787774	94. 54	93. 87
WS	10. 334787	7. 523754	0. 256180	0. 110816	97. 52	98. 53
SX	0. 578052	0. 299133	0. 503881	0. 274636	12. 83	8. 19
SH	1. 030726	0. 456638	0. 652451	0. 330930	36. 70	27. 53
SR	0. 882169	0. 434188	0. 748790	0. 362180	15. 12	16. 58
SS	0. 336292	0. 144834	0. 259707	0. 116066	22. 77	19. 86

表 7　全局轨迹误差对比

Table 7　Typical value of ATE

Seq.	ORB-SLAM2		Proposed system		Improvement	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0. 565505	0. 200691	0. 014932	0. 007969	97. 36	96. 03
WH	0. 327989	0. 177225	0. 025357	0. 013169	92. 27	92. 57
WR	0. 817879	0. 430206	0. 047672	0. 033180	94. 17	92. 29
WS	0. 409268	0. 175114	0. 006957	0. 003244	98. 30	98. 15
SX	0. 009275	0. 004796	0. 010674	0. 004922	−15. 08	−2. 63
SH	0. 027882	0. 013692	0. 014091	0. 006767	49. 46	50. 58
SR	0. 021513	0. 014181	0. 016052	0. 009607	25. 38	32. 25
SS	0. 007698	0. 003655	0. 005568	0. 003047	27. 67	16. 63

相对旋转误差和全局轨迹误差上的量化结果，表头中 ORB-SLAM2 表示在相同实验环境中 ORB-SLAM2 系统的测试结果，Proposed system 表示所提系统的测试结果，Improvement 指所提系统在与其对比的系统上的提升效果，结果以百分数表示。

表 5～7 中的数据表明，相较于 ORB-SLAM2，所提系统在几个高动态环境中各个序列都有 94% 以上的提升。虽然 ORB-SLAM2 发展已经足够成熟，但所提系统在低动态运动环境中除去 SX 序列外仍有不同程度的提升，其中对于 RMSE 有最低 15%、最高 49% 的提升，S. D 有最低 16%、最高 50% 的提升。为了更直观的表示，将高动态帧集的 WH 序列数据进行了可视化，结果如图 10、11 所示。从图 10 可以看出，ORB-SLAM2 测试的轨迹与真实轨迹相差严重，而所提系统测试的轨迹与真实轨迹相差无几。从图 11 则可以看出，ORB-SLAM2 测试的最大位移误差已超过 1. 2 m，而所提系统测试的最大位移误差仅约为 0. 1 m。

在地图构建方面，ORB-SLAM2 仅构建了环境的稀疏点云地图。并且由于该系统缺乏动态物体的判别

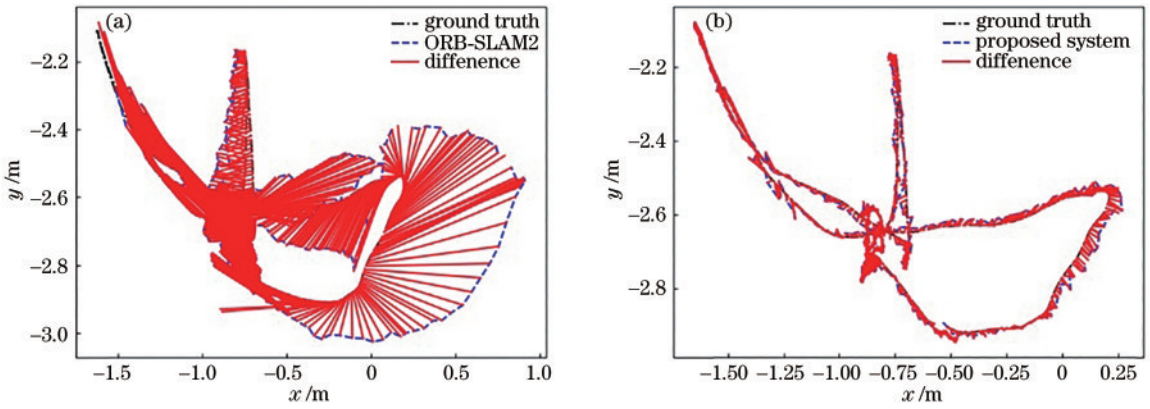

图 10　全局路径误差对比。(a) ORB-SLAM2；(b)所提系统

Fig. 10　Comparison of ATE. (a) ORB-SLAM2; (b) proposed system

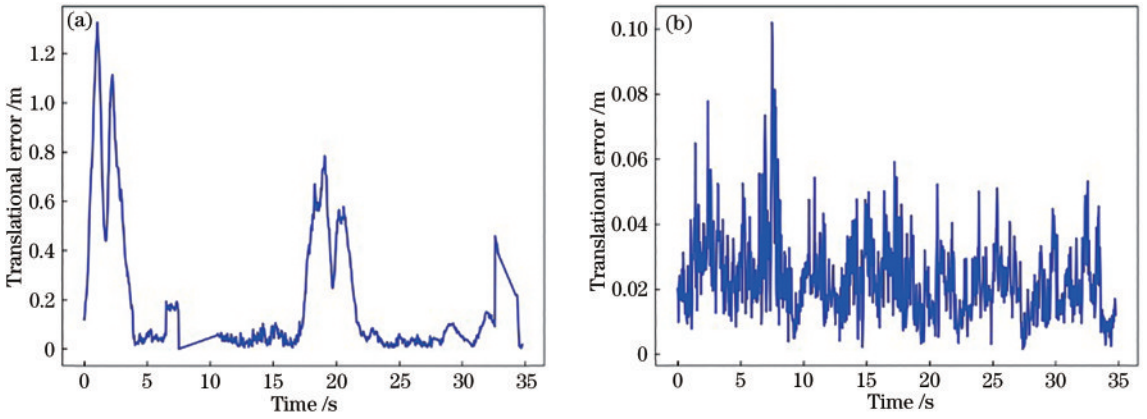

图 11　相对平移误差对比。(a) ORB-SLAM2;(b)所提系统

Fig. 11　Comparison of relative translation error. (a) ORB-SLAM2; (b) proposed system

机制,动态物体也被构建到了地图之中。所提系统进一步对上述数据集的 8 个动态帧进行了含有语义信息的八叉树地图构建,结果如[图 12(b)]所示。从地图效果与原始帧序列的对比可以看到,不同类别的物体被赋予了不同颜色,处于运动状态的人体并未构建到地图中。

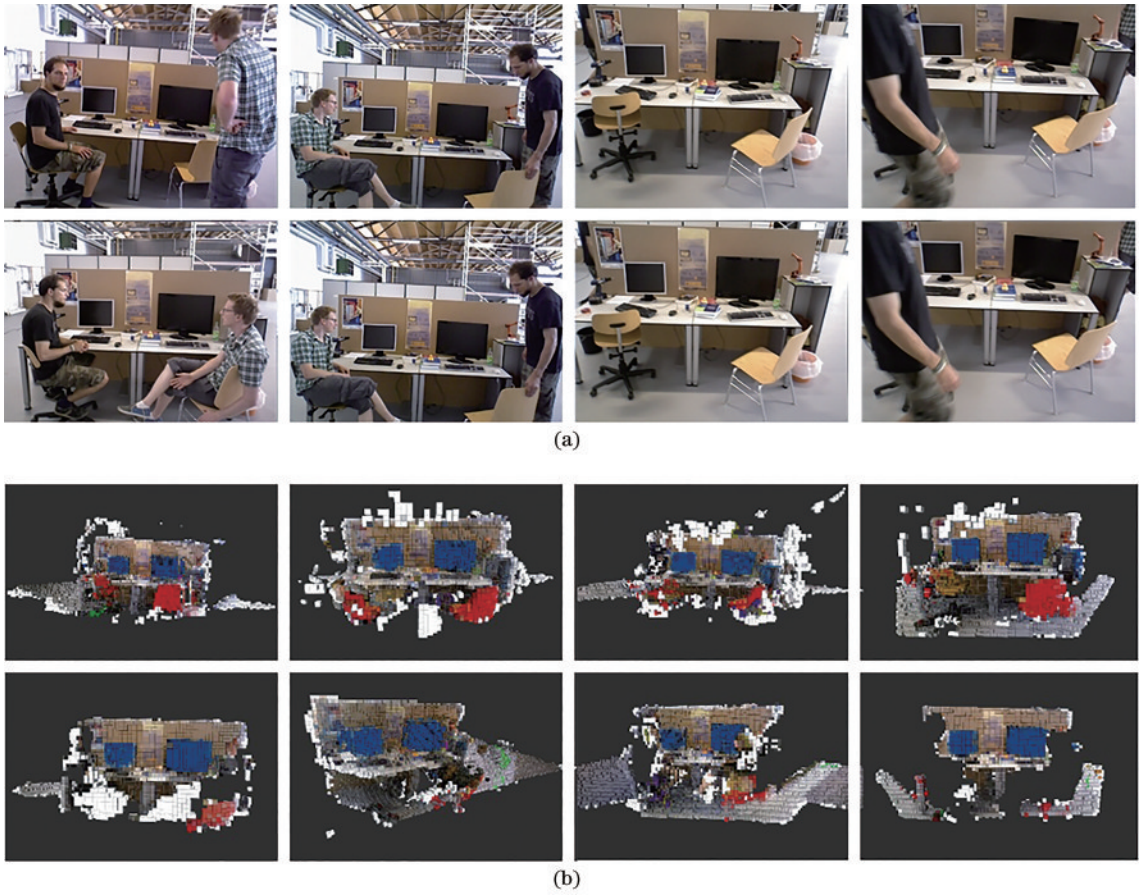

图 12　语义八叉树地图构建效果。(a)原始帧序列;(b)语义八叉树地图

Fig. 12　Semantic octree map building effect. (a) Original frame sequence; (b) semantic octree map

3.2.2　与 DS-SLAM 的对比

同样基于 ORB-SLAM2 提出的经典 DS-SLAM 与所提系统相似,主要致力于动态场景的图像处理,不同的是 DS-SLAM 采用标准 RANSAC 抽样方案对特征点的运动一致性进行判断并且在语义分割线程使用 SegNet 提取语义信息。将所提系统与 DS-SLAM 在同一硬件平台上进行了测试对比,对比结果如表 8~10 所示。

从表中可看出:在前 4 组高动态帧序列中,所提系统相较于 DS-SLAM 都有不同程度的提升,尤其在 WR 旋转维度与 WX 位移维度表现非常明显;对于 RMSE 最高有 89% 的提升,对于 SD 最高有 85% 的提

表 8 相对位移误差对比

Table 8 Typical value of translation

Seq.	ORB-SLAM2		Proposed system		Improvement	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0.030283	0.019977	0.019991	0.010374	33.99	48.07
WH	0.031893	0.016458	0.025955	0.012036	18.62	26.87
WR	0.152189	0.116961	0.062167	0.041967	59.15	64.12
WS	0.009347	0.004358	0.009044	0.004097	3.24	5.99
SX	0.012692	0.006664	0.012649	0.006140	0.34	7.86
SH	0.017403	0.007808	0.017660	0.008221	−1.48	−5.29
SR	0.025757	0.014738	0.021174	0.011048	17.79	25.04
SS	0.007327	0.003564	0.007237	0.003520	1.29	1.23

表 9 相对旋转误差对比

Table 9 Typical value of rotation

Seq.	ORB-SLAM2		Proposed system		Improvement	
	RMSE /deg	S. D /deg	RMSE /deg	S. D /deg	RMSE /%	S. D /%
WX	0.766283	0.529380	0.618651	0.376314	19.27	28.91
WH	0.846296	0.422084	0.749063	0.354548	11.49	16.00
WR	3.043619	2.335232	1.201734	0.787774	60.52	66.27
WS	0.255013	0.106953	0.256180	0.110816	−0.46	−3.61
SX	0.495392	0.270148	0.493881	0.274636	0.31	−1.66
SH	0.630755	0.300657	0.628451	0.290930	0.37	3.24
SR	0.843054	0.468068	0.748790	0.362180	11.18	22.62
SS	0.262147	0.117104	0.259707	0.116066	0.93	0.89

表 10 全局轨迹误差对比

Table 10 Typical value of ATE

Seq.	ORB-SLAM2		Proposed system		Improvement	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0.022180	0.014402	0.014932	0.007969	32.68	44.67
WH	0.032083	0.017715	0.025357	0.013169	20.96	25.66
WR	0.433820	0.228252	0.047672	0.033180	89.01	85.46
WS	0.007709	0.003275	0.006957	0.003244	9.75	0.95
SX	0.010339	0.005377	0.010674	0.004922	−3.24	8.46
SH	0.014816	0.006672	0.014091	0.006767	4.89	−1.42
SR	0.020242	0.012680	0.016052	0.009607	20.70	24.24
SS	0.006273	0.003085	0.005568	0.003047	11.24	1.23

升。由于DS-SLAM系统本身运用于动态环境，在后4组低动态帧序列中，所提系统表现虽不及高动态帧序列中的表现，但在各序列上也有略微提升，其中较为可观的是SR序列性能提升明显，其最高有25%、最低有11%的性能增加。对于SX与SH序列，即使存在性能下降的情况，其下降的幅度仅在5%以内处于一个较为平稳的状态。综上所述，所提系统对于高动态场景的处理表现对比于DS-SLAM框架有一定的优势，在低动态场景中可保持与DS-SLAM相近甚至略高的性能表现。

另外，为了更好地表现Multi-stage RANSAC算法与Fast-SCNN相较于DS-SLAM框架的性能优势，分别对Multi-stage RANSAC算法、Fast-SCNN及结合Multi-stage RANSAC和Fast-SCNN对特征点提取、运动一致性判断、语义分割的具体耗时进行了消融实验。

首先，在仍使用SegNet作为分割网络的情况下，将Multi-stage RANSAC嵌入DS-SLAM中，实验结果如表11～13所示，表头中的Multi指嵌入Multi-stage RANSAC后的vSLAM系统。得益于Multi-stage RANSAC算法，其设置的不同的判别阈值与步进型抽样机制，嵌入DS-SLAM中仍有明显的性能提升，在WX、SR序列提升明显，均有10%以上的提升，甚至个别序列性能提升了25%～30%。

表 11 相对位移误差对比

Table 11 Typical value of translation

Seq.	DS-SLAM		Multi		Improvement	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0. 030283	0. 019977	0. 024652	0. 013576	18. 59	32. 04
WH	0. 031893	0. 016458	0. 029321	0. 016007	8. 06	2. 74
WR	0. 152189	0. 116961	0. 138137	0. 110548	9. 23	5. 48
WS	0. 009347	0. 004358	0. 009599	0. 004705	−2. 70	−7. 96
SX	0. 012692	0. 006664	0. 012615	0. 006442	0. 61	3. 33
SH	0. 017403	0. 007808	0. 017283	0. 007227	0. 69	7. 44
SR	0. 025757	0. 014738	0. 021212	0. 011222	1. 76	24. 1
SS	0. 007327	0. 003564	0. 007315	0. 003871	0. 16	−0. 03

表 12 相对旋转误差对比

Table 12 Typical value of rotation

Seq.	DS-SLAM		Multi		Improvement	
	RMSE /deg	S. D /deg	RMSE /deg	S. D /deg	RMSE /%	S. D /%
WX	0. 766283	0. 529380	0. 683158	0. 440730	10. 85	16. 75
WH	0. 846296	0. 422084	0. 824658	0. 447438	2. 56	−6. 01
WR	3. 043619	2. 335232	2. 797477	2. 189081	8. 09	6. 26
WS	0. 255013	0. 106953	0. 261905	0. 122951	−2. 70	−14. 96
SX	0. 495392	0. 270148	0. 488689	0. 259726	0. 14	3. 86
SH	0. 630755	0. 300657	0. 618015	0. 289567	0. 20	3. 69
SR	0. 843054	0. 468068	0. 681521	0. 333695	19. 16	28. 71
SS	0. 262147	0. 117104	0. 265180	0. 110150	−1. 16	5. 94

表 13 全局轨迹误差对比

Table 13 Typical value of ATE

Seq.	DS-SLAM		Multi		Improvement	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0. 022180	0. 014402	0. 019148	0. 010682	13. 67	25. 83
WH	0. 032083	0. 017715	0. 028845	0. 015730	10. 09	11. 21
WR	0. 433820	0. 228252	0. 407781	0. 205247	6. 00	10. 08
WS	0. 007709	0. 003275	0. 007302	0. 003459	5. 28	−5. 62
SX	0. 010339	0. 005377	0. 009962	0. 005129	3. 65	4. 61
SH	0. 014816	0. 006672	0. 014589	0. 006602	1. 53	1. 10
SR	0. 020242	0. 012680	0. 016531	0. 010268	18. 33	19. 02
SS	0. 006273	0. 003085	0. 006142	0. 003216	2. 09	−4. 25

其次，在保持RANSAC不变的情况下，使用Fast-SCNN替换DS-SLAM的SegNet，实验结果如表14～16所示，表头中的Semantic表示将Fast-SCNN嵌入DS-SLAM后的vSLAM系统。由于语义分割网络精度的提升，Semantic方案整体在各方面性能提升明显，尤其在WR序列中，Semantic方案在相对位移误差与相对旋转误差都降低了50%以上，全局轨迹误差降低了81. 81%。同样，个别序列比如WS在相对位移误差与相对旋转误差有5%以内的增加，处于一个相对平稳的状态。

最后，结合Multi-stage RANSAC算法和Fast-SCNN对特征点提取、运动一致性判断、语义分割的具体耗时进行了测试对比，实验结果如表17所示。从实验数据可知：在特征点提取阶段两个系统耗时一致；在运动一致性判定阶段，所提系统将耗时降到了13 ms左右；而在语义分割阶段耗时降到了21 ms左右。从总的耗时来看，DS-SLAM的追踪线程需要约25 ms，语义分割线程需要约28 ms，相较而言所提系统节省了约21%的时间。

表 14 相对位移误差对比

Table 14 Typical value of translation

Seq.	DS-SLAM		Semantic		Improvement	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0.030283	0.019977	0.020390	0.010275	32.67	48.57
WH	0.031893	0.016458	0.027500	0.014119	13.77	14.21
WR	0.152189	0.116961	0.072958	0.051138	52.06	56.28
WS	0.009347	0.004358	0.009172	0.004383	1.87	−0.57
SX	0.012692	0.006664	0.012570	0.006306	0.96	5.37
SH	0.017403	0.007808	0.017659	0.007570	−1.47	3.05
SR	0.025757	0.014738	0.020002	0.010545	22.34	28.45
SS	0.007327	0.003564	0.007121	0.003518	2.81	1.29

表 15 相对旋转误差对比

Table 15 Typical value of rotation

Seq.	DS-SLAM		Semantic		Improvements	
	RMSE /deg	S. D /deg	RMSE /deg	S. D /deg	RMSE /%	S. D /%
WX	0.766283	0.529380	0.614950	0.379957	19.75	28.23
WH	0.846296	0.422084	0.799037	0.402792	5.58	4.57
WR	3.043619	2.335232	1.471557	1.021622	51.65	56.25
WS	0.255013	0.106953	0.260383	0.114594	−2.11	−7.14
SX	0.495392	0.270148	0.486103	0.263159	1.88	2.59
SH	0.630755	0.300657	0.649584	0.314669	−2.99	−4.66
SR	0.843054	0.468068	0.667244	0.323382	20.85	30.91
SS	0.262147	0.117104	0.258282	0.113205	1.47	3.33

表 16 全局轨迹误差对比

Table 16 Typical value of ATE

Seq.	DS-SLAM		Semantic		Improvements	
	RMSE /m	S. D /m	RMSE /m	S. D /m	RMSE /%	S. D /%
WX	0.022180	0.014402	0.015859	0.008289	28.50	42.45
WH	0.032083	0.017715	0.026138	0.013565	18.53	23.43
WR	0.433820	0.228252	0.057395	0.041521	86.77	81.81
WS	0.007709	0.003275	0.007095	0.003276	7.96	−0.03
SX	0.010339	0.005377	0.010258	0.004892	0.78	9.02
SH	0.014816	0.006672	0.014299	0.006334	3.49	5.07
SR	0.020242	0.012680	0.016159	0.009816	20.17	22.59
SS	0.006273	0.003085	0.005903	0.002946	5.90	4.51

表 17 特征点提取、运动一致性判定、语义分割的耗时对比

Table 17 Time comparison of feature point extraction, motion consistency judgment, and semantic segmentation unit: s

vSLAM	ORB extract	Moving check	Segmentation
Proposed	0.008118	0.012913	0.020810
DS-SLAM	0.008118	0.017299	0.027627

3.3 实际环境测试

为了验证所提系统在实际环境中的表现，最后按照 TUM 数据集格式对实验室的两个日常场景进行了采集。在两个场景中，所提系统特征提取阶段的可视化效果如图 13 所示，位于人体内的特征点也进行了匹配。结合图 14 的语义分割结果，所提系统对运动人体上的动态特征点进行了剔除，剔除动点后的效果如图 15 所示。

上述两个实际场景的语义八叉树建图效果如图 16 所示。一方面，结合语义分割获得的语义信息对

图 13　所提系统在两个场景中的特征点提取

Fig. 13　Feature points extraction of proposed system in two scenes

图 14　所提系统在两个场景中的语义分割结果

Fig. 14　Semantic segmentation results of proposed system in two scenes

图 15　所提系统在两个场景中的动点剔除效果

Fig. 15　Moving point removing effect of proposed system in two scenes

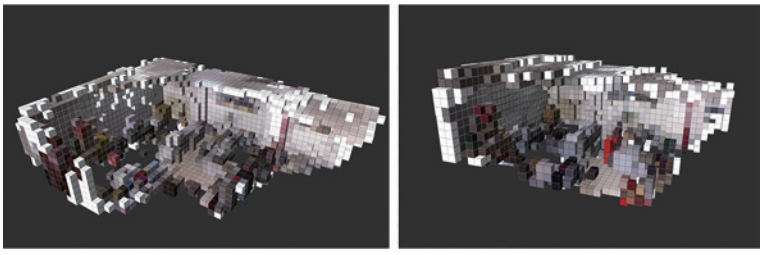

图 16　所提系统地图构建效果

Fig. 16　Map building effect of proposed system

运动物体从图像中进行了有效剔除，另一方面，将静态物体对应的颜色标签融合到体素中，构建了不包含动态物体的室内环境语义八叉树地图。同时，得益于其轻量性和灵活度，该Octomap可用于大型场景建图。

4　结　　论

以将深度学习引入传统vSLAM系统为主线，对经典vSLAM在特征点匹配、动点剔除及地图的构建等几个环节进行了优化：在特征匹配环节，采用金字塔LK光流法以提升特征匹配的速度；在外点剔除环节，基于Multi-stage RANSAC与对极几何验证的运动一致性验证方案，对Multi-stage RANSAC中的两个阈值τ_1与τ_2进行了充分、细致的测试优化，并结合Fast-SCNN线程获取的语义信息来实现较为完整的动态特征点剔除；在地图构建环节，构建了全局一致的静态环境的语义八叉树地图以提升系统的智能性。在TUM

RGB-D数据集的8个RGB-D动态帧序列测试下，所提系统在高动态序列中对比ORB-SLAM2系统各项指标均有94%以上的性能提升，全局轨迹误差仅为0.1 m。较DS-SLAM系统，对动点剔除的总耗时有21%的缩减。在地图构建方面，经过体素滤波后的语义点云地图与语义八叉树地图分别占据9.6 MB、685 kB的存储空间，相较于原始17 MB的点云，语义八叉树地图仅占用其4%的存储空间，轻量且灵活，可满足用于更高层次应用的标准与需求。综合而言，所提vSLAM构图方法具有较高的实时性和可靠性。在接下来的工作中将继续探索机器人在语义八叉树地图中路径规划、导航、避障等运用与场景理解问题。

参 考 文 献

[1] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]//2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, November 13-16, 2007, Nara, Japan. New York: IEEE Press, 2007: 225-234.

[2] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.

[3] Mur-Artal R, Tardos 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.

[4] Campos C, Elvira R, Rodriguez J J G, et al. ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[5] Yu C, Liu Z X, Liu X J, et al. DS-SLAM: a semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 1-5, 2018, Madrid. New York: IEEE Press, 2018: 1168-1174.

[6] Bescos B, Facil J M, Civera J, et al. DynaSLAM: tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

[7] Li A, Wang J K, Xu M, et al. DP-SLAM: a visual SLAM with moving probability towards dynamic environments[J]. Information Sciences, 2021, 556: 128-142.

[8] 卢金，刘宇红，张荣芬. 面向动态场景的语义视觉里程计[J]. 激光与光电子学进展, 2021, 58(6): 0611001.
Lu J, Liu Y H, Zhang R F. Semantic-based visual odometry towards dynamic scenes[J]. Laser & Optoelectronics Progress, 2021, 58(6): 0611001.

[9] 赵宏，刘向东，杨永娟. 基于RGB-D图像的室内机器人同时定位与地图构建[J]. 计算机应用, 2020, 40(12): 3637-3643.
Zhao H, Liu X D, Yang Y J. Indoor robot simultaneous localization and mapping based on RGB-D image[J]. Journal of Computer Applications, 2020, 40(12): 3637-3643.

[10] Poudel R P K, Liwicki S, Cipolla R. Fast-SCNN: fast semantic segmentation network[EB/OL]. (2019-02-12)[2021-05-04]. https://arxiv.org/abs/1902.04502.

[11] Yu C Q, Wang J B, Peng C, et al. BiSeNet: bilateral segmentation network for real-time semantic segmentation[M]//Ferrari V, Hebert M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11217: 334-349.

[12] Badrinarayanan V, Kendall A, Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[13] Everingham M, van Gool L, Williams C K I, et al. The pascal visual object classes (VOC) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.

[14] 李运舵，车进，薛澄. 基于点线特征匹配的实时定位及地图重建方法[J]. 激光与光电子学进展, 2022, 59(2): 0210003.
Li Y D, Che J, Xue C. Simultaneous localization and mapping based on point and line feature matching[J]. Laser & Optoelectronics Progress, 2022, 59(2): 0210003.

[15] 方琪，王晓华，苏杰. 一种基于分组策略的点线融合特征SLAM算法[J]. 激光与光电子学进展, 2021, 58(14): 1415003.
Fang Q, Wang X H, Su J. A visual SLAM algorithm through the combination point and line segments based on grouping strategy[J]. Laser & Optoelectronics Progress, 2021, 58(14): 1415003.

[16] 陈兴华，蔡云飞，唐印. 一种基于点线不变量的视觉SLAM算法[J]. 机器人, 2020, 42(4): 485-493.
Chen X H, Cai Y F, Tang Y. A visual SLAM algorithm based on point-line invariant[J]. Robot, 2020, 42(4): 485-493.

[17] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.

[18] Wilhelms J, van Gelder A. Octrees for faster isosurface generation[J]. ACM Transactions on Graphics, 1992, 11(3): 201-227.

[19] 席志红，王洪旭，韩双全. 基于ORB-SLAM2系统的快速误匹配剔除算法与地图构建[J]. 计算机应用, 2020, 40(11): 3289-3294.
Xi Z H, Wang H X, Han S Q. Fast mismatch elimination algorithm and map-building based on ORB-SLAM2 system[J]. Journal of Computer Applications, 2020, 40(11): 3289-3294.

[20] 高翔，张涛，刘毅. 视觉SLAM十四讲：从理论到实践[M]. 北京：电子工业出版社, 2017.
Gao X, Zhang T, Liu Y. 14 lectures on visual SLAM: from theory to practice[M]. Beijing: Publishing House of Electronics industry, 2017.

[21] Silberman N, Hoiem D, Kohli P, et al. Indoor segmentation and support inference from RGBD images

[M]//Fitzgibbon A, Lazebnik S, Perona P, et al. Computer Vision-ECCV 2012. Lecture notes in computer science. Heidelberg: Springer, 2012, 7576: 740-760.

[22] 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-Algarve, Portugal. New York: IEEE Press, 2012: 573-580.