



自适应窗隔匹配与深度学习相结合的RGB-D SLAM算法

余东应,刘桂华,曾维林,冯 波,张文凯

(西南科技大学 信息工程学院,四川 绵阳 621010)

摘 要:在动态场景的SLAM系统中,传统的特征点法视觉SLAM系统易受动态物体的影响,使得图像前后两帧的动态物体区域出现大量的误匹配,导致机器人定位精度不高。为此,提出一种结合自适应窗隔匹配模型与深度学习算法的动态场景RGB-D SLAM算法。构建基于自适应窗隔匹配模型的视觉SLAM前端算法框架,该框架筛选图像帧后采用基于网格的概率运动统计方式实现匹配点筛选,以获得静态区域的特征匹配点对,然后使用恒速度模型或参考帧模型实现位姿估计。利用深度学习算法Mask R-CNN提供的语义信息进行动态场景的静态三维稠密地图构建。在TUM数据集和实际环境中进行算法性能验证,结果表明,该算法在动态场景下的定位精度和跟踪速度均优于ORB-SLAM2及DynaSLAM系统,在全长为6.62 m的高动态场景中定位精度可达1.475 cm,平均跟踪时间为0.024 s。

关键词:动态场景;自适应窗隔匹配;静态区域特征匹配;深度学习;静态三维稠密地图构建

开放科学(资源服务)标志码(OSID):



中文引用格式:余东应,刘桂华,曾维林,等.自适应窗隔匹配与深度学习相结合的RGB-D SLAM算法[J].计算机工程,2021,47(8):224-233.

英文引用格式:YU D Y, LIU G H, ZENG W L, et al. RGB-D SLAM algorithm combining adaptive window interval matching and deep learning[J]. Computer Engineering, 2021, 47(8): 224-233.

RGB-D SLAM Algorithm Combining Adaptive Window Interval Matching and Deep Learning

YU Dongying, LIU Guihua, ZENG Weilin, FENG Bo, ZHANG Wenkai

(School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China)

[Abstract] In dynamic scenes, the traditional visual SLAM systems based on the feature point method are easily affected by dynamic objects, producing a large number of mismatches in the dynamic object areas between the previous frame and the next frame, so the positioning accuracy of robots is significantly reduced. To solve the problem, an RGB-SLAM algorithm that combines an adaptive window interval matching model and a deep learning algorithm is proposed for dynamic scenes. A framework for the front-end algorithm of visual SLAM is constructed based on the adaptive window interval matching model. The framework selects the image frames, and uses grid-based probabilistic motion statistics to filter the matched points, so the matched feature point pairs in static areas are obtained. Then the constant speed model or reference frame model is used to achieve position estimation. On this basis, the semantic information provided by the deep learning algorithm Mask R-CNN is used to construct a static 3D dense map of the dynamic scenes. The algorithm is tested on the TUM data set and in the real-world environment, and the results show that the positioning accuracy and tracking speed of the algorithm are better than that of ORB-SLAM2 and DynaSLAM in dynamic scenes. The positioning accuracy of the algorithm reaches up to 1.475 cm in the highly dynamic scenes that are as wide as 6.62 m, and the average tracking time of the algorithm is 0.024 s.

[Key words] dynamic scene; adaptive window interval matching; static region feature matching; deep learning; static 3D dense map construction

DOI:10.19678/j.issn.1000-3428.0058956

基金项目:国防科工局核能开发科研项目“核应急处置机器人关键技术研究”([2016]1295);四川省科技厅重点研发项目“智能AGV车辆的SLAM建图与自主导航避障方法”(2021YFG0380)。

作者简介:余东应(1995—),男,硕士研究生,主研方向为视觉SLAM、三维重建、图像处理;刘桂华(通信作者),教授、博士;曾维林、冯 波、张文凯,硕士研究生。

收稿日期:2020-07-16 修回日期:2020-09-06 E-mail:yudong2817@sina.com

0 概述

SLAM(Simultaneous Localization and Mapping)在机器人位姿估计及环境感知过程中起着至关重要的作用,主要应用于机器人的导航和避障任务以及自动驾驶^[1-3]和增强现实^[4-5]等领域。视觉传感器体积小、使用方便,并且图像存储了丰富的环境纹理和色彩信息,从而使得V-SLAM(Visual-Simultaneous Localization and Mapping)得到广泛关注和研究。虽然目前的V-SLAM系统框架比较成熟,解决了V-SLAM任务中的很多问题,许多先进的SLAM算法^[6-8]也相继被提出,但是,这些算法均在静态的场景或数据集上进行实验测试,可以达到很好的定位精度与鲁棒性,但并未解决场景中存在动态物体的问题。当场景中存在动态物体,如人、车等,会导致V-SLAM出现定位误差较大或跟踪失败的问题,因此,在动态环境中进行位姿估计与地图构建依然是一个挑战。目前用于消除动态物体影响的方法主要有直接筛选剔除法和语义分割法2种。

直接筛选剔除法可以分为2类,一类是根据图像帧提取的关键点,如Shi-Tomasi^[9]、FAST^[10]和ORB^[11]等,直接进行关键点筛选,最具代表性的算法是RANSAC^[12],该算法针对低动态场景具有较好的鲁棒性,但是,当场景中存在运动的车或人时,依然会导致位姿估计错误或跟踪失败;另一类方法根据连续图像帧之间的运动关系,采用光流法进行动态物体检测,如文献[13]结合最优估计和均匀采样的方式进行动态物体检测,相比于传统的光流法,其速度快,但是精度较低,文献[14]采用RGB光流法进行场景中的动态物体分割,文献[15]根据双目立体的场景流进行动态物体检测,但是上述2个系统均需要提供动态物体的先验信息,难以适用于长期的位姿跟踪。

语义分割法使用深度学习的方式,检测场景中的动态物体,并对其所处区域进行标注与分割,基于该方法已取得诸多研究成果^[16-18]。文献[18]在ORB-SLAM2的基础上,采用Segnet^[19]网络进行语义分割,获得图像中的语义信息,同时结合运动一致性检查剔除动态物体上的特征点,该算法构建了带有语义信息的八叉树地图,具有较好的实时性,但是定位精度不高。文献[20]采用Mask R-CNN^[21]进行图像的实例分割,对于算法中的RGB-D相机部分,采用多视角几何检测动态特征点并将其移除,同时结合图像修复技术实现静态三维稠密地图构建,该算法具有较高的定位精度,但是系统实时性不高。文献[22]采用稀疏场景流和语义分割进行动态目标识别,采用双目立体匹配构建静态的稠密三维地图,但是该算法没有对定位精度进行评价,并且系统运行速度仅为2.5 Hz。

现有的SLAM系统在动态环境下普遍存在实时性较差或定位误差较大的问题,导致其无法应用于现实场景中。针对以上问题,本文结合自适应窗隔匹配模型及深度学习算法,将两者应用于SLAM系统中。在本文系统前端,采用自适应窗隔匹配模型进行匹配点筛选,得到静态区域的特征匹配点对,实现V-SLAM系统的初始化。在位姿跟踪部分,通过最小化重投影误差进行当前帧位姿跟踪,并且仅对关键帧实现特征匹配,同时向局部地图中添加静态三维点,更新稀疏点云地图。在系统后端,将前端获取的关键帧位姿信息以及RGB-D相机获得的图像信息传送到深度学习模块进行语义分割,以消除场景中的动态物体,实现静态三维稠密地图构建。

1 算法框架

动态场景下的RGB-D SLAM算法结构框图如图1所示,该算法主要包括初始化、位姿跟踪、局部地图构建、闭环检测、深度学习语义分割及三维地图构建等模块。

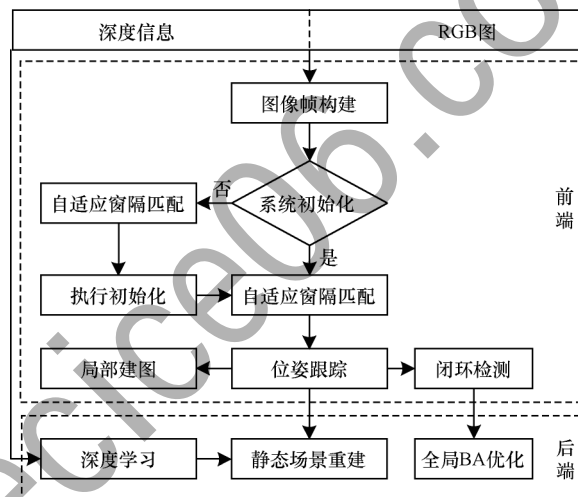


图1 RGB-D SLAM算法结构框图

Fig.1 Block diagram of RGB-D SLAM algorithm structure

在RGB-D SLAM算法中,首先进行图像帧的构建,图像帧的构建方式和ORB-SLAM2^[6]一致,将每一帧的图像信息和提取的ORB特征点保存在系统中,用于初始化、位姿跟踪和地图优化等。

若系统未初始化,则采用自适应窗隔匹配的方式实现动态场景中静态特征点的匹配,然后根据BA(Bundle Adjustment)优化得到最终的初始化静态3D地图点。在初始化成功之后,将当前帧作为初始位姿,根据构建的静态局部地图实现连续图像帧的位姿估计。

在跟踪过程中同时进行关键帧的选择,以保存机器人定位过程中的关键信息。通过关键帧之间的特征匹配,向局部地图中添加三维点,用于位姿跟踪、闭环检测及重定位等。同时,对关键帧中的图像信息进行

语义分割,消除场景中动态物体的影响,构建静态的三维稠密地图,从而实现地图的可重复使用。

2 系统前端设计

在本文算法的前端,首先对每幅图像进行均匀分布的ORB特征提取,然后使用自适应窗隔匹配模型消除场景中动态物体的影响,得到静态区域的特征匹配点对,最后进行位姿跟踪与关键帧筛选。

2.1 ORB特征提取

首先将原始图像按照尺度为1.2的大小构成层数为8的图像金字塔,将每层图像分成 30×30 的网格区域;然后设置FAST^[10]角点提取的阈值为9,在每层图像上提取FAST角点,并且生成256维的BRIEF^[23]特征点描述子。如果当前网格区域没有提取到特征点,减少FAST角点提取的阈值继续提取特征点,阈值最小值为5。将所有层提取到的特征点进行保存得到当前图像帧的ORB特征点。

2.2 自适应窗隔匹配模型

普通的特征点法SLAM系统一般采用连续的两帧图像进行特征匹配,以实现系统的初始化和位姿跟踪。但是,当场景中存在动态物体时,由于相机帧率较高,对于连续的两帧图像,运动物体并没有发生明显的移动,难以消除动态物体的影响。另外,对于非连续的两帧图像,传统的匹配方式会造成大量的误匹配。连续图像帧及非连续图像帧的匹配区别如图2所示。

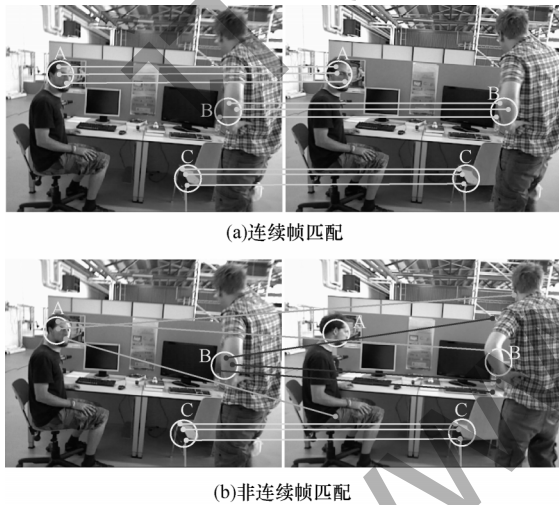


图2 图像帧之间的特征匹配

Fig.2 Feature matching between image frames

在图2中,A和B为场景中的部分运动区域,C为场景中的静态区域。图2(a)描述了连续图像帧之间的特征匹配,由于动态区域移动较小,因此A、B和C区域均能得到较好的特征匹配,但是会造成SLAM系统累计误差持续增大的问题,影响定位精度。图2(b)所示为非连续图像帧之间的特征匹配,由于动态区域移动较大,使A和B区域的特征匹配发生

错误,同样会影响系统的定位精度。

针对以上问题,为方便理论分析,本文定义如下的集合关系式,记图像中全局特征点集合为 U ,静态背景及物体上的特征点集合为 S ,动态物体上的特征点集合为 D ,三者的关系为:

$$U = D \cup S \quad (1)$$

所有图像帧集合表示为 $F = \{f_1, f_2, \dots, f_N\}$,固定窗口图像帧集合为 $W = \{f_1, f_2, \dots, f_8\}$,其中, $W \subset F$ 。

为消除动态物体的影响,引入自适应窗隔匹配模型,该模型通过帧间筛选策略将窗口中首帧图像与第 $x(x \leq 8)$ 帧图像进行特征匹配, $x-1$ 为窗隔大小,窗口大小设置为固定值8。自适应窗隔匹配模型算法步骤如下:

1)选取窗口 W 中两帧图像 f_1, f_x ,提取均匀分布的ORB特征点,并使用暴力匹配法获得初步的匹配关系,然后进行图像区域划分。图像区域划分主要流程如图3所示。

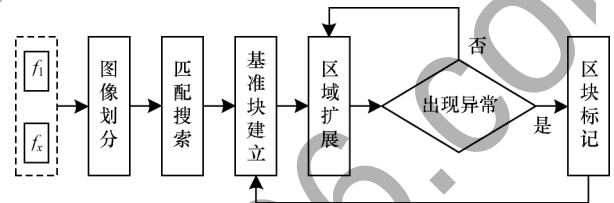


图3 图像区域划分流程

Fig.3 Procedure of image area division

将图像帧划分为 16×16 大小的像素块,以 f_x 图像左上角像素块为起点,在 f_1 图像中执行匹配搜索,搜索步长设置为8,优先选择含匹配角点的像素块。匹配正确之后以该像素块为基准,在横向和纵向依次进行匹配扩展,逐渐扩大区域范围,若该区域块能够占据图像区域的绝大部分,则视为静态区域块,记作 T 区域。匹配搜索过程中检测到不连续区块,则将其单独标记,并对该区域进行局部扩展,若该区域占据图像范围较小,视其为动态区域块,记为 F 区域。在实际应用中,设定 T 区域必定出现且唯一, F 区域可为0个或者多个。

2)图像中的场景随着相机运动而不断更新,两帧图像间的变换可视为仿射变换。记第一帧图像中某区域第 i 个像素块坐标的向量形式为 v_{il} ,第 x 帧($x \leq 8$)图像中与之对应的像素块坐标向量为 v_{ip} ,则有如下关系:

$$v_{il} = A v_{ip} + t, \quad i = 1, 2, \dots, m \quad (2)$$

其中, A 为二阶可逆矩阵, t 为二维平移向量, m 为对应区域中像素块的总数。可以通过最小化均方根误差来求解 A, t ,其公式为:

$$\varphi(A, t) = \argmin_{A, t} \left(\frac{1}{m} \sum_{i=1}^m \|v_{il} - (A v_{ip} + t)\|_2^2 \right)^{\frac{1}{2}} \quad (3)$$

考虑到区域中所有的像素块,将式(2)写成矩阵形式可得:

$$[\mathbf{v}_{1l}, \mathbf{v}_{2l}, \dots, \mathbf{v}_{ml}] = \mathbf{A}[\mathbf{v}_{1p}, \mathbf{v}_{2p}, \dots, \mathbf{v}_{mp}] + \mathbf{t} \quad (4)$$

记 $\mathbf{V}_l = (\mathbf{v}_{1l}, \mathbf{v}_{2l}, \dots, \mathbf{v}_{ml})$, $\mathbf{V}_p = (\mathbf{v}_{1p}, \mathbf{v}_{2p}, \dots, \mathbf{v}_{mp})$, 则式(4)可以记作 $\mathbf{V}_l = \mathbf{A}\mathbf{V}_p + \mathbf{t}\mathbf{L}^T$, $\mathbf{L}$ 为 m 维单位列向量。因此,对式(3)的求解可以等价为无约束极值问题,如式(5)所示:

$$\psi(\mathbf{A}, \mathbf{t}) = m \left(\frac{1}{m} \sum_{i=1}^m \left\| \mathbf{v}_{il} - (\mathbf{A}\mathbf{v}_{ip} + \mathbf{t}) \right\|_2^2 \right)^{\frac{1}{2}} = \text{tr}((\mathbf{V}_l - (\mathbf{A}\mathbf{V}_p + \mathbf{t}))(\mathbf{V}_l - (\mathbf{A}\mathbf{V}_p + \mathbf{t}))^T) \quad (5)$$

其中, $\psi(\mathbf{A}, \mathbf{t})$ 是关于 $\mathbf{A}$ 和 $\mathbf{t}$ 的矩阵函数, $\text{tr}()$ 运算符表示求矩阵的迹。在函数 $\psi(\mathbf{A}, \mathbf{t})$ 取极小值时,对应的参数 $\mathbf{A}^*$ 、 $\mathbf{t}^*$ 满足如下关系式:

$$\left. \frac{\partial \psi(\mathbf{A}, \mathbf{t})}{\partial \mathbf{A}} \right|_{\mathbf{A}=\mathbf{A}^*} = 0 \quad (6)$$

$$\left. \frac{\partial \psi(\mathbf{A}, \mathbf{t})}{\partial \mathbf{t}} \right|_{\mathbf{t}=\mathbf{t}^*} = 0 \quad (7)$$

令 $\mathbf{J} = \mathbf{I} - \frac{1}{m} \mathbf{L}\mathbf{L}^T$, $\mathbf{I}$ 为单位矩阵, 则有 $\mathbf{J}\mathbf{J}^T = \mathbf{J}^T\mathbf{J} = \mathbf{J} = \mathbf{J}^T$, 对式(6)、式(7)求解可得:

$$\mathbf{t}^* = \frac{1}{m} (\mathbf{V}_l - \mathbf{V}_l \mathbf{J} \mathbf{V}_p^T (\mathbf{V}_p \mathbf{J} \mathbf{V}_p^T)^{-1} \mathbf{V}_p) \mathbf{L} \quad (8)$$

$$\mathbf{A}^* = \mathbf{V}_l \mathbf{J} \mathbf{V}_p^T (\mathbf{V}_p \mathbf{J} \mathbf{V}_p^T)^{-1} \quad (9)$$

由式(8)、式(9)可以得到 T 区域及 F 区域对应的仿射变换参数。由于相机运动是平滑的,为简化运算,对窗口中的图像只分析平移向量 $\mathbf{t}$ 的变化。

在窗口图像帧集合 W 中,将图像帧 f_1 与图像帧 $f_2 \sim f_8$ 逐一进行匹配搜索来确定窗隔的大小。定义 $\mathbf{t}_T$ 为 T 区域的平移向量, $\mathbf{t}_F$ 为 F 区域的平移向量。存在如下3种情况:

(1) 图像帧中仅存在 T 区域,则将其视为静态场景,若 $\|\mathbf{t}_T\|_2 \geq 20$,表示场景在某段时间内没有出现动态物体且相机已经发生明显移动,则认定两帧图像符合筛选条件。

(2) 当图像帧中存在一个 F 区域,判断 $\mathbf{t}_T$ 与 $\mathbf{t}_F$ 的大小,若 $0 \leq \|\mathbf{t}_F\|_2 \leq 0.2\|\mathbf{t}_T\|_2$ 或 $1.8\|\mathbf{t}_T\|_2 \leq \|\mathbf{t}_F\|_2 \leq 3\|\mathbf{t}_T\|_2$,表示图像帧中动态物体区域发生明显变化,则认为两帧图像符合筛选条件。

(3) 当图像帧中出现多个 F 区域,仅对平移向量较小的 T 区域做出判断,若其满足 $0 \leq \|\mathbf{t}_F\|_2 \leq 0.4\|\mathbf{t}_T\|_2$ 或 $1.2\|\mathbf{t}_T\|_2 \leq \|\mathbf{t}_F\|_2 \leq 2\|\mathbf{t}_T\|_2$,则认为两帧图像符合筛选条件。

窗隔图像帧的更新策略为:若图像帧 f_1 在初始窗口图像帧中没有匹配到符合筛选条件的图像帧,则保留 f_1 ,将 $f_2 \sim f_8$ 帧移出,再向窗口中移入7帧图像,重新进行匹配搜索。若图像帧 f_1 与窗口中第 x 帧图像符合筛选条件,则窗隔大小即为 $x-1$,然后将窗口中前 $x-1$ 帧图像移出,并将第 x 帧图像设置为 f_1 帧,再向窗口中补充 $x-1$ 帧图像,并开启新一轮匹配搜索。

3) 在筛选出符合条件的图像帧之后,本文使用改进的GMS^[24]特征匹配算法来获得高质量的匹配点对,该算法是一种高效且实时的匹配点筛选算法,其基于运动网格统计模型对原始的特征匹配进行筛选,得到静态区域的特征匹配。

根据GMS特征匹配算法的原理,得出如下关系:

$$s = |X| - 1 \quad (10)$$

其中, s 表示两帧图像中对应区域得分, $|X|$ 表示该区域中特征匹配点对的数量,如果一个特征匹配是正确的,那么它的周围依然存在其他特征匹配点对,此时 s 的值较大。

对于连续的2个图像帧,如图2(a)所示,区域A、B、C中均可得到较多的匹配点对,从而使得动态区域A、B造成假得分的情况。假设A、B和C区域中的特征匹配点对得分为 s_A 、 s_B 和 s_C ,那么有如下关系:

$$s_A \approx s_B \approx s_C \quad (11)$$

在这种情况下,无法通过设定合适的阈值将动态区域(区域A和B)和静态区域(区域C)的特征匹配区分开。因此,GMS算法不适用于场景中动态物体移动较小的情况。

本文系统采用自适应窗隔匹配模型,得到场景中动态物体发生明显移动的两帧图像。以图2(b)为例,静态区域C有较高的匹配得分,但是,区域A和B受到动态物体的影响发生了明显变化,使得这些区域的特征匹配出现较大的错误, s_A 和 s_B 的值很小,则有如下关系:

$$s_A \approx s_B \ll s_C \quad (12)$$

这样就可以通过设置合适的阈值,消除动态物体上特征匹配点的影响,解决假得分的问题,获得静态区域的特征匹配点集 S 。

为提高特征匹配点对的筛选速度,将图像进行网格划分。由网格的可分性度量值 $P \propto \sqrt{KM}$ 可知, P 正比于网格数量 K 及网格区域中特征点数量 M ,且 K 与 M 互补。因此,综合考虑特征匹配点对的筛选速度和可分精度,采用增大网格数量 K ,即缩小网格大小的方式,将网格大小设置为 10×10 。然后,直接判断每个网格中所有特征匹配点对的正确性,如图4所示。

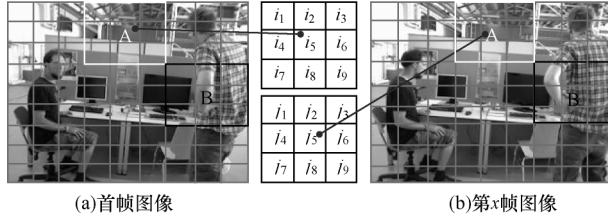


图4 图像网格划分

Fig.4 Grid division of image

图4为经过窗隔匹配筛选的两帧图像,图4(a)为窗隔模型中的首帧图像,图4(b)为第 x 帧图像,在以网格A为中心的 3×3 网格中,统计左右图像对中区域A的匹配得分 s_{ij} ,其定义如下:

$$s_{ij} = \sum_{k=1}^9 |X_{i_k j_k}| \quad (13)$$

其中, $|X_{i_k j_k}|$ 表示网格对 $\{i_k, j_k\}$ 中对应的特征点匹配数量,根据运动的平滑性及特征点匹配的一致性, s_{ij} 值越大,表示网格A中特征匹配的正确性越高,反之,则正确性越低。此时,设置阈值 t 来对网格A中的特征匹配是否正确进行判断:

$$t = \alpha \sqrt{n} \quad (14)$$

其中, α 为常量(一般设置为6), n 表示以A为中心的 3×3 网格中特征匹配的数量。

对于静态区域,如图4中白色方框所示,网格A可以获得较高的匹配得分 s_{ij} ,而对图4中黑色方框所示的动态区域,由于存在运动物体,使得左图像提取的特征点无法在右图像中找到对应的匹配点,甚至会在右图像中找到错误的匹配点,因此该区域的匹配得分 s_{ij} 较低。由式(14)能够看出,可以根据区域内匹配特征点的数量设置自适应阈值。若 $s_{ij} \geq t$,则区域匹配正确,保留匹配点;若 $s_{ij} < t$,则区域匹配错误,舍弃匹配点。通过自适应阈值判断匹配点对,可以剔除动态区域的误匹配,得到静态区域的匹配点对,从而提高全局的匹配正确率。

2.3 位姿跟踪

使用自适应窗隔匹配模型进行系统的初始化,初始化成功后得到当前帧的位姿信息,将当前帧设置为关键帧和参考帧。当新的图像帧创建后,得到 n 组匹配成功的静态点对,其对应于三维空间点 P_i , p_{i1} 、 p_{i2} ($i \in n$)为该点在参考帧和当前帧中的投影, $\hat{p}_{i2}$ 为在初始位姿下空间点 P_i 在当前帧中的投影, e_i 为重投影误差,如图5所示。然后,可通过BA优化方式,即最小化重投影误差求解相机位姿。引入李代数中 ξ 表示相机位姿,有如下关系:

$$\hat{p}_{i2} = \exp(\xi^\wedge) p_{i1}, \forall i \in n \quad (15)$$

其中, $^\wedge$ 运算符号表示将李代数元素转换为变换矩阵。该点重投影误差可表示为:

$$e_i = p_{i2} - \exp(\xi^\wedge) p_{i1} \quad (16)$$

将 n 组匹配点的误差项进行平方求和,构建最小二乘问题,并使其最小化:

$$\xi^* = \operatorname{argmin}_{\xi} \frac{1}{2} \sum_{i=1}^n \|p_{i2} - \exp(\xi^\wedge) p_{i1}\|_2^2 \quad (17)$$

使用列文伯格-马夸尔特优化算法求解式(17),可求得最优相机位姿 ξ^* 。

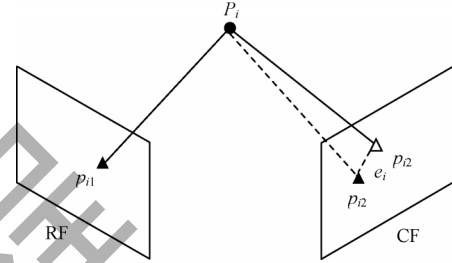


图5 重投影误差示意图

Fig.5 Schematic diagram of reprojection error

在进入跟踪模块后,使用参考帧模型或恒速模型对当前帧位姿进行估计。对于参考帧模型,设置最新的关键帧作为参考帧,对参考帧和当前帧进行窗隔匹配,如果参考帧匹配的特征点在局部地图中存在对应的三维点,就将这些三维点投影到当前帧。将上一帧的位姿信息作为当前帧位姿的初始值,根据BA优化得到当前帧的位姿信息与观测到的三维地图点。与参考帧模型不同,恒速模型假设两帧之间的运动与之前两帧相同,以此来设定BA优化的位姿初始值。

关键帧的筛选策略不同于ORB-SLAM2,本文首先采用自适应窗隔匹配法得到两图像帧,再对静态区域的匹配点数进行判断,若匹配点数大于50且距离上一次所加入的关键帧超过8帧,则加入新的关键帧。该方法既可确保图像中的动态物体出现显著运动,又可避免引入过多的冗余信息,而且在为局部地图添加新的地图点时,能够消除场景中动态物体的影响。

3 系统后端设计

为消除场景中的动态物体,构建可供实际应用的静态稠密地图,本文系统采用Mask R-CNN进行图像的语义分割,然后将分割得到的动态物体对应的深度信息从该帧的深度图中移除,从而达到仅对静态场景进行稠密地图构建的目的。

3.1 图像语义分割

本文采用Mask R-CNN对图像帧进行语义分割,使用在COCO^[25]数据集上训练的深度模型,该模型可以分割常见的运动物体(如人、自行车、汽车、火车、卡车、鸟、马、羊等)。Mask R-CNN算法的输入是原始的RGB图像,输出图像中包含矩形选择框、类别标签以及实例的掩膜区域。然而,在本文系统的应用中,仅需用到实例的掩膜信息,因此,在算法中

移除了关于类别预测的模块,分割结果如图6所示,该图来自COCO数据集。



图6 Mask R-CNN分割示意图

Fig.6 The segmentation diagram of Mask R-CNN

由于Mask R-CNN是在Faster R-CNN^[26]的框架下进行设计的,因此相对于其他语义分割算法具有较快的预测速度,在GPU设备上预测COCO数据集上一张图片的语义信息仅需200 ms。

V-SLAM系统均有很高的实时性要求,为了提高系统运行速度,将语义分割部分作为一个单独的线程运行。假设输入图像大小为 $m \times n \times 3$,则输出矩阵的大小为 $m \times n \times l$,其中, l 表示图像中目标的数目。对图像的每一个通道进行语义分割,最终得到单帧图像中所有的动态目标。

3.2 静态三维地图构建

利用深度学习算法得到动态物体的掩膜信息,根据对应关系可移除动态物体在深度图中的深度数据,然后结合SLAM系统得到的关键帧位姿信息进行静态三维稠密地图构建。

为了保证系统的运行速度,将可视化点云信息界面作为一个单独的线程运行,当图像帧为关键帧时,将此时的深度图和RGB图像结合相机内部参数恢复得到相机坐标系下的三维点云坐标 (x, y, z) ,如式(18)所示:

$$\begin{cases} z = d \\ x = z(u - c_x)/f_x \\ y = z(v - c_y)/f_y \end{cases} \quad (18)$$

其中, d 与深度图在像素坐标 (u, v) 的值成正相关,其余参数均为相机内参,由相机自身决定,可以通过相机标定的方式得到。

4 实验结果与分析

本文使用TUM^[27] RGB-D动态序列数据集进行特征匹配、定位精度及跟踪时间的对比分析,并在动态数据集及实际环境中进行场景稠密重建测试。实验环境配置为4核i7处理器,8 GB内存,GTX 1060,Ubuntu 16.04操作系统。

4.1 特征匹配效果分析

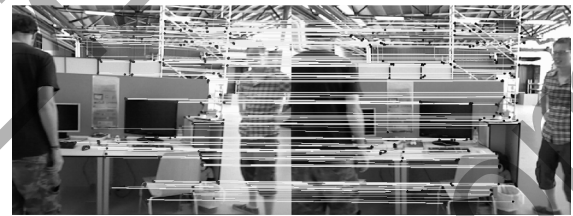
基于TUM的fr3_w_xyz动态数据集,对本文系统的自适应窗隔匹配策略与BF、FLANN匹配方式作对比分析,在前后两帧中均提取1 000个特征点进行匹配,匹配结果如图7所示。



(a)BF匹配



(b)FLANN匹配



(c)自适应窗隔匹配

图7 匹配效果对比

Fig.7 Comparison of matching effect

从图7(a)、图7(b)可以看出,BF匹配和FLANN匹配不能有效移除动态物体上的匹配点对,同时导致静态区域部分出现较多的误匹配。图7(c)为自适应窗隔匹配算法的匹配效果,该算法不仅移除了动态物体上的匹配点对,也使静态区域特征匹配的正确率得到提升。

对窗隔匹配算法、BF算法及FLANN算法的匹配耗时及匹配点对个数进行对比分析,结果如表1所示。

表1 匹配耗时及匹配数量对比

Table 1 Comparison of matching time and matching quantity

匹配算法	匹配耗时/s	匹配点对个数
窗隔匹配算法	0.005 7	302
BF算法	0.004 5	386
FLANN算法	0.018 1	116

由表1可知,FLANN算法的匹配耗时最长,且匹配点对数量最少,BF算法匹配耗时最短,且匹配点对个数最多,但是该算法会带来大量的误匹配。因此,综合考虑匹配速度及匹配正确性,自适应窗隔匹配算法达到最优。

4.2 定位精度及实时性分析

使用TUM RGB-D数据集评估本文算法的定位精度和实时性,分析包含多动态对象的低动态(fr3_s_*)和高动态(fr3_w_*)场景序列的测试结果,并与ORB-SLAM2及DynaSLAM系统进行对比。以下均以

OUR_SYSTEM来指代本文系统,实验结果如表2和表3所示。在表2中使用绝对轨迹误差(ATE)中的平方根误差(RMSE)对定位精度进行评估,表3中使用图像帧平均跟踪时间进行实时性能评估,图像帧跟踪时间为从构建图像帧到获取其位姿的耗时,图像帧平均跟踪时间为所有图像帧跟踪时间总和取平均值。其中,“*”表示跟踪丢失且重定位失败。

表2 系统定位精度对比

Table 2 Comparison of system positioning accuracy cm

数据集	OUR_SYSTEM	DynaSLAM	ORB-SLAM2
fr1_desk	1.493	1.655	1.506
fr2_desk_person	0.581	*	0.597
fr3_s_hs	1.445	2.503	1.737
fr3_s_rpy	1.677	4.809	1.828
fr3_s_static	0.676	0.628	0.825
fr3_s_xyz	0.833	1.453	0.927
fr3_w_hs	2.500	2.902	75.561
fr3_w_rpy	4.072	4.876	72.208
fr3_w_static	0.708	0.747	42.126
fr3_w_xyz	1.475	1.543	78.292

表3 系统跟踪时间对比

Table 3 Comparison of system tracking time s

数据集	OUR_SYSTEM	DynaSLAM	ORB-SLAM2
fr1_desk	0.026	1.688	0.032
fr2_desk_person	0.027	*	0.038
fr3_s_hs	0.037	1.106	0.040
fr3_s_rpy	0.024	0.810	0.030
fr3_s_static	0.021	0.679	0.028
fr3_s_xyz	0.025	0.757	0.029
fr3_w_hs	0.023	1.185	0.041
fr3_w_rpy	0.024	0.972	0.042
fr3_w_static	0.022	1.228	0.042
fr3_w_xyz	0.024	1.170	0.042

由表2可得,在静态场景(fr1_*)及单一动态对象序列(fr2_*)中,本文系统(OUR_SYSTEM)定位精度最高。在低动态的场景序列(fr3_s_*)中,OUR_SYSTEM以及其他SLAM系统定位误差均较小。但是,在多数数据集上,OUR_SYSTEM达到了最高的定位精度。在高动态场景序列(fr3_w_*)中,OUR_SYSTEM的定位精度不仅远高于ORB-SLAM2系统,而且也优于DynaSLAM系统。

由表3可得,在静态场景序列及全部动态场景序列中,OUR_SYSTEM的跟踪速度快于ORB-SLAM2及DynaSLAM,并且跟踪速度比DynaSLAM系统快10倍以上。计算表中OUR_SYSTEM跟踪时间的均值,可得平均时间为0.025 s,即跟踪速率为

40 Hz,满足视觉SLAM的实时性要求。

综合以上关于定位精度和跟踪速度的分析,OUR_SYSTEM在静态及动态场景中均可实现较高的定位精度,而且具有较好的实时性,总体效果优于ORB-SLAM2和DynaSLAM系统。

为进一步验证OUR_SYSTEM的定位精度,绘制绝对位姿误差APE(Absolute Pose Error)的分布示意图,如图8所示。

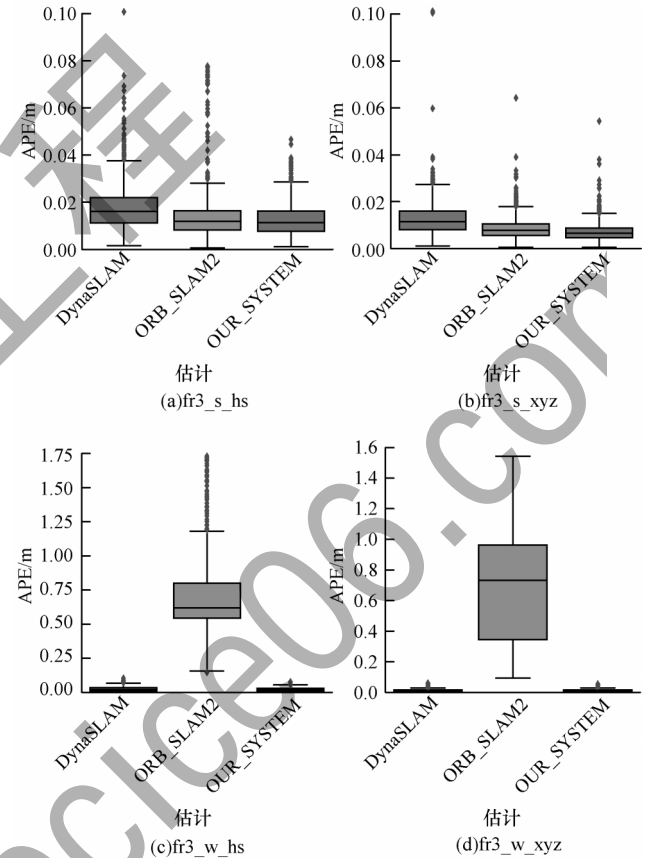


图8 绝对位姿误差分布

Fig.8 APE distribution

图8所示为位姿跟踪过程中OUR_SYSTEM、ORB-SLAM2及DynaSLAM在动态数据集fr3_s_hs、fr3_s_xyz、fr3_w_hs及fr3_w_xyz上的APE分布,其中,矩形区域表示3/4的APE数据分布,其余部分表示剩余的APE数据分布。从图8可以看出,在低动态数据集(fr3_s_*)的定位过程中,这3种SLAM算法均可达到较小的绝对位姿误差,并且OUR_SYSTEM绝对位姿误差最小。在高动态场景数据集(fr3_w_*)中,原始ORB-SLAM2系统的绝对位姿误差最大,而OUR_SYSTEM在最大限度上消除了运动物体的影响,达到了最高的定位精度。

OUR_SYSTEM在数据集fr3_s_hs、fr3_s_xyz、fr3_w_hs及fr3_w_xyz中实际运动轨迹与真实轨迹的热力对比如图9所示。

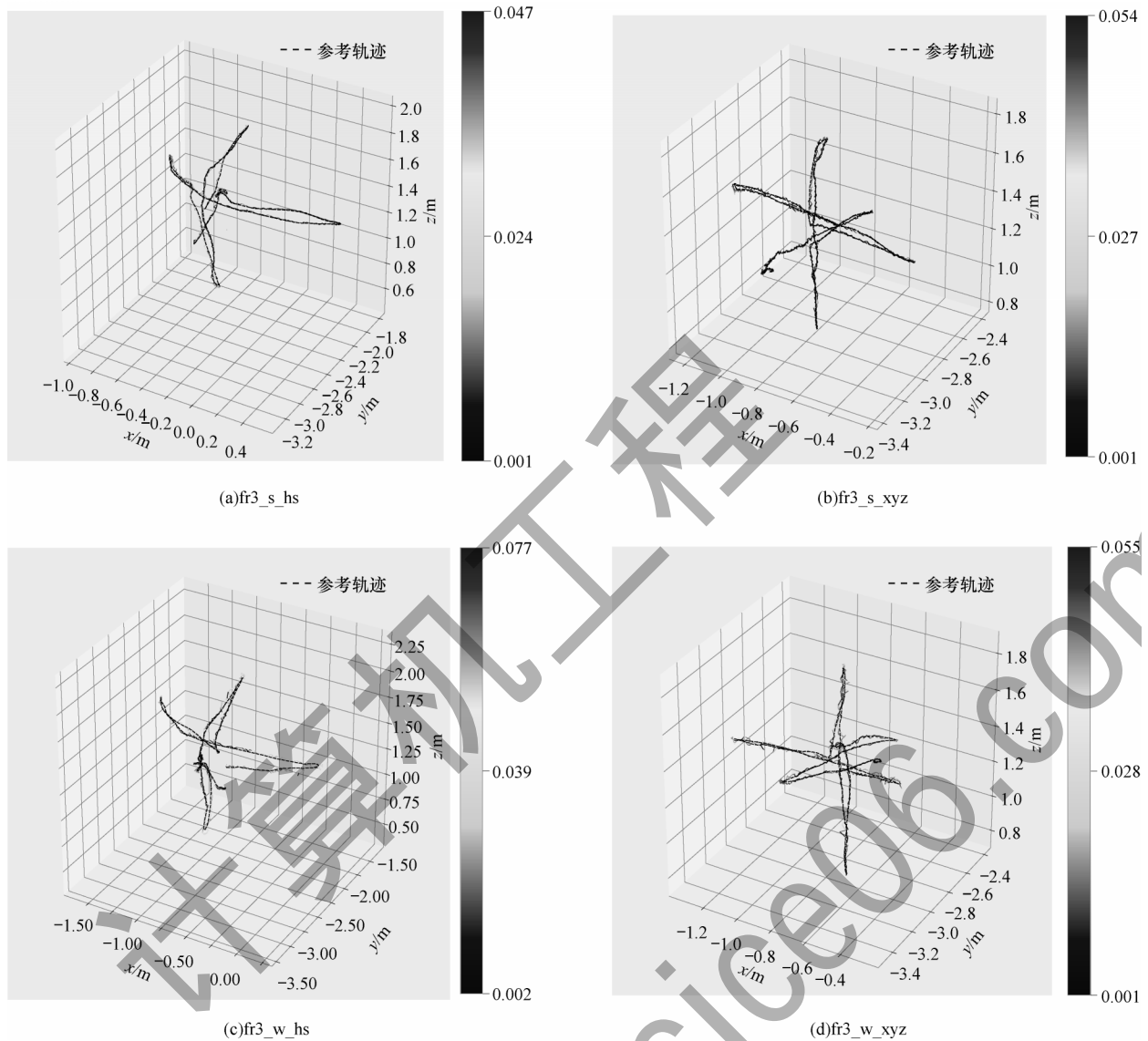


图9 3D轨迹热力图

Fig.9 3D trajectory thermal map

从图9可以看出,OUR_SYSTEM在动态环境下能够实现对相机位姿的精确估计。

4.3 静态地图构建及算法有效性分析

在TUM动态数据集和实际场景中分别进行三维场景重建效果测试,并在实际环境下测试OUR_SYSTEM特征提取及定位的效果。TUM动态数据集上的实验结果如图10所示。图10(a)为原始RGB图像,对原始图像通过深度学习算法进行动态目标剔除,得到图10(b),同时,将对应的深度图中动态目标的深度信息也剔除,得到图10(c)。使用不含动态目标的RGB图像及深度图像,结合图像帧的位姿信息进行静态三维稠密地图构建,结果如图10(d)所示,该地图不含动态物体,可以重复使用,有较高的实用性。

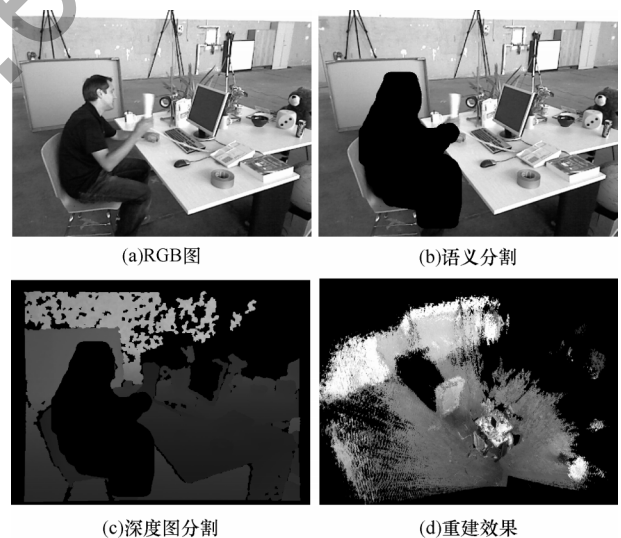


图10 TUM数据集的三维地图构建效果

Fig.10 3D map construction effect of TUM dataset

为进一步验证算法的有效性,使用RGB-D相机(乐视Letv)在动态的实验室环境中进行场景三维重建。在场景中存在移动人员的情况下,通过RGB-D相机连续采集图像进行实验,使相机足够平稳且出现闭环。图11所示为ORB-SLAM2与OUR_SYSTEM提取ORB特征的对比效果,图11(a)为ORB_SLAM2算法效果,其中有大量的特征点分布在行走的人员身上,图11(b)为OUR_SYSTEM效果,其中绝大多数的特征点分布在背景中。



图11 特征点提取效果对比

Fig.11 Effect comparison of feature points extraction

图12所示为ORB_SLAM2与OUR_SYSTEM相机预测轨迹对比结果,图中实线轨迹比较平滑,且有闭环出现,这与相机的真实运动轨迹相符,表明OUR_SYSTEM在实际动态场景中精度较高。而黑色虚线轨迹由于受到动态物体的影响,出现了大幅度的偏差,定位误差较大,虽然有闭环出现,但未能实现全局轨迹矫正。

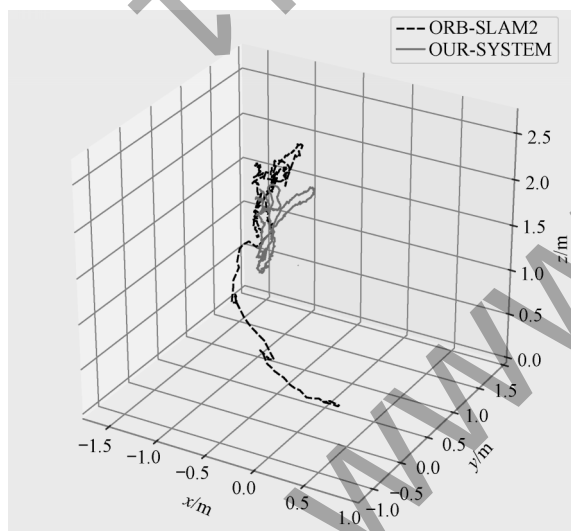


图12 相机预测轨迹对比

Fig.12 Comparison of camera prediction trajectory

与图10类似,图13(a)、图13(b)、图13(c)为视序列中某一帧图像的处理效果,图13(d)为最终的重建效果。可以看出,最终效果消除了实际场景中的动态物体,构建动态场景的静态三维稠密地图能

够实现地图的可重复使用。



图13 实际场景三维地图构建效果

Fig.13 Construction effect of 3D map of actual scene

5 结束语

本文针对现有RGB-D视觉SLAM系统在动态环境下存在的精度不足、实时性较差等问题,提出一种动态场景下的RGB-D SLAM算法,该算法采用自适应窗隔匹配模型来实现动态环境下的运动估计。在定位过程中,相比于ORB-SLAM2及DynaSLAM方法,本文算法在实时性和定位精度上均有所提升。在场景重建过程中,本文算法使用Mask R-CNN检测运动物体,根据所提供的掩膜信息消除动态目标。实验结果表明,该算法可以对场景进行完整的三维重建。但是,当相机移动过快或者处于低纹理场景中时,本文算法容易出现跟踪丢失的情况。因此,下一步将考虑添加IMU传感器,使用多传感器融合的方法来实现更加鲁棒的位姿跟踪。

参考文献

- [1] NISTÉR D, NARODITSKY O, BERGEN J. Visual odometry for ground vehicle applications [J]. Journal of Field Robotics, 2006, 23(1): 3-20.
- [2] HOWARD A. Real-time stereo visual odometry for autonomous ground vehicles [C]//Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. Washington D. C., USA: IEEE Press, 2008: 3946-3952.
- [3] JOO S-H, MANZOOR S, ROCHA Y G, et al. A realtime autonomous robot navigation framework for human like high-level interaction and task planning in global dynamic environment[EB/OL]. [2020-06-01]. <https://arxiv.org/ftp/arxiv/papers/1905/1905.12942.pdf>.

- [4] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces[C]//Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. Washington D. C., USA: IEEE Press, 2007: 1-10.
- [5] 赵越, 李晶皎, 王爱侠, 等. 基于IEKF-SLAM的未知场景增强现实跟踪注册算法[J]. 计算机工程, 2016, 42(1): 272-277. ZHAO Y, LI J J, WANG A X, et al. Tracking and registration algorithm of augmented reality on unknown scene based on IEKF-SLAM[J]. Computer Engineering, 2016, 42(1): 272-277. (in Chinese)
- [6] 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.
- [7] ENGEL J, KOLTUN V, CREMERS D. Direct sparse odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(3): 611-625.
- [8] GOMEZ-OJEDA R, MORENO F-A, ZUÑIGA-NOËL D, et al. PL-SLAM: a stereo SLAM system through the combination of points and line segments [J]. IEEE Transactions on Robotics, 2019, 35(3): 734-746.
- [9] SHI J. Good features to track [C]//Proceedings of 1994 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 1994: 593-600.
- [10] ROSTEN E, DRUMMOND T. Machine learning for high-speed corner detection [C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Springer, 2006: 430-443.
- [11] RUBLEE E, RABAU D V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF [C]//Proceedings of the 2011 International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2011: 1-2.
- [12] RAGURAM R, FRAHM J-M, POLLEFEYS M. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus [C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Springer, 2008: 500-513.
- [13] FANG Y, DAI B. An improved moving target detecting and tracking based on optical flow technique and Kalman filter [C]//Proceedings of 2009 International Conference on Computer Science & Education. Berlin, Germany: Springer, 2009: 1197-1202.
- [14] WANG Y, HUANG S. Motion segmentation based robust RGB-D SLAM [C]//Proceeding of the 11th World Congress on Intelligent Control and Automation. Berlin, Germany: Springer, 2014: 3122-3127.
- [15] ALCANTARILLA P F, YEBES J J, ALMAZÁN J, et al. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments [C]//Proceedings of 2012 IEEE International Conference on Robotics and Automation. Washington D. C., USA: IEEE Press, 2012: 1290-1297.
- [16] 杨雪, 范勇, 高琳, 等. 基于纹理基元块识别与合并的图像语义分割[J]. 计算机工程, 2015, 41(3): 253-257. YANG X, FAN Y, GAO L, et al. Image semantic segmentation based on texture element block recognition and merging [J]. Computer Engineering, 2015, 41(3): 253-257. (in Chinese)
- [17] 夏胡云, 叶学义, 罗霄哈, 等. 多尺度空间金字塔池化PCANet的行人检测[J]. 计算机工程, 2019, 45(2): 270-277. XIA H Y, YE X Y, LUO X H, et al. Pedestrian detection using multi-scale principal component analysis network of spatial pyramid pooling [J]. Computer Engineering, 2019, 45(2): 270-277. (in Chinese)
- [18] YU C, LIU Z, LIU X J, et al. DS-SLAM: a semantic visual slam towards dynamic environments [C]//Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Washington D. C., USA: IEEE Press, 2018: 1168-1174.
- [19] 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.
- [20] BESCOS B, FÁCIL 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.
- [21] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN [C]//Proceedings of the IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2017: 2961-2969.
- [22] BÂRSAN I A, LIU P, POLLEFEYS M, et al. Robust dense mapping for large-scale dynamic environments [C]//Proceedings of 2018 IEEE International Conference on Robotics and Automation. Washington D. C., USA: IEEE Press, 2018: 7510-7517.
- [23] CALONDER M, LEPETIT V, STRECHA C, et al. Brief: Binary robust independent elementary features [C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Springer, 2010: 778-792.
- [24] BIAN J, LIN W-Y, MATSUSHITA Y, et al. GMS: grid-based motion statistics for fast, ultra-robust feature correspondence [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 4181-4190.
- [25] LIN T-Y, MAIRE M, BELONGIE S, et al. Microsoft coco: common objects in context [C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Springer, 2014: 740-755.
- [26] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C]//Proceedings of Advances in Neural Information Processing Systems. Berlin, Germany: Springer, 2015: 91-99.
- [27] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems [C]//Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Washington D. C., USA: IEEE Press, 2012: 573-580.