

引用格式:徐慧,张合新,姚二亮,等.动态场景下基于场景流的闭环检测算法[J].电光与控制,2019,26(8):37-42. XU H, ZHANG H X, YAO E L, et al. A loop closure detection algorithm based on scene flow in dynamic scenes[J]. Electronics Optics & Control, 2019, 26(8):37-42.

动态场景下基于场景流的闭环检测算法

徐 慧, 张合新, 姚二亮, 宋海涛, 朱建文
(火箭军工程大学,西安 710025)

摘要:针对传统闭环检测算法在动态场景下不稳定、易失败的问题,提出了一种在动态场景下能够准确检测到闭环的算法。首先,改进基于场景流区分动、静特征点算法,使得到的动静点划分更为准确;其次,剔除动态特征点,进行聚类,将图像在视觉词典树各个节点的 TF-IDF 熵作为图像在该视觉单词的得分权重,从而构造得分向量对场景进行描述;最后,采用负指数幂函数作为计算两幅图像的相似性得分函数,计算当前帧与候选关键帧的相似性得分,经过最后的闭环确认环节,得到最终的与当前帧发生闭环的关键帧。实际场景的实验表明,所提算法在动态场景下能够有效检测到闭环。

关键词: 闭环检测; 动态场景; 场景流; TF-IDF 熵; 相似性得分函数

中图分类号: TP242 文献标志码: A doi:10.3969/j.issn.1671-637X.2019.08.008

A Loop Closure Detection Algorithm Based on Scene Flow in Dynamic Scenes

XU Hui, ZHANG He-xin, YAO Er-liang, SONG Hai-tao, ZHU Jian-wen
(Rocket Force University of Engineering, Xi'an 710025, China)

Abstract: To solve the problem that the traditional Loop Closure Detection (LCD) algorithms are unstable and easy to fail in dynamic scenes, an algorithm that can accurately detect the loop closure under dynamic scenes is proposed. First, the algorithm for distinguishing the dynamic from the static features based on the scenes flow is improved so that the dynamic and static points are more accurately determined. Then, the dynamic feature points are removed and clustering processing is performed. The TF-IDF entropy of each node of the image in the visual dictionary tree is used as the weight of the image in the visual word, and a score vector is constructed to describe the scenes. Finally, the negative exponent power function is used as the similarity score function to calculate the similarity scores of the two images. The similarity score between the current frame and the candidate key frame is calculated, and after the final loop closure confirmation, the final key frame forming a loop closure with the current frame is obtained. The experiment of the actual lab scenes shows that the proposed algorithm can effectively detect the loop closure in the dynamic scenes.

Key words: loop closure detection; dynamic scene; scene flow; TF-IDF entropy; similarity score function

0 引言

同时定位与地图构建(Simultaneous Localization and Mapping, SLAM)是机器人实现自主导航的关键^[1-2]。闭环检测(Loop Closure Detection, LCD)即能否准确且有效地判别出机器人位于之前已经到达的场景。作为SLAM技术的基础问题,闭环检测能够减少机器人运动

带来的累积误差,并对构建准确的地图起重要作用^[3]。

目前,在视觉SLAM问题研究中,很多研究者采用词袋(Bag of Words, BOW)^[4]的方法进行闭环检测。该方法通过训练大量图片,提取图像的特征,进行聚类得到视觉单词,从而构成视觉词典;当有查询图像到来时,BOW方法会利用单词的统计直方图来描述场景;通过计算当前场景的图像帧与历史场景图像帧的相似性,来进行闭环检测。

近几年,在BOW方法的基础上,出现了多种取得良好闭环检测效果的改进算法。CUMMINS等^[5]考虑

收稿日期:2018-09-21

修回日期:2018-10-08

作者简介:徐 慧(1995—),男,陕西西安人,硕士,研究方向为SLAM和计算机视觉。

单词之间的相关性,利用 Chow-Liu 树的方法得到闭环检测概率;李博等^[6]提出一种基于视觉词典树的金字塔 TF-IDF 得分匹配方法,有效区分了视觉词典树的不同层次节点的影响。闭环检测对实时性以及准确率要求较高并且需要有效地减少感知歧义。IAB-MAP^[4]以及 FAB-MAP^[7]方法虽然能够较好地解决大规模环境下的闭环检测问题,但是检测时间长,且准确率不高;RTAB-MAP^[8]方法虽然具有较好的实时性,但是仅利用部分历史帧信息进行闭环检测,准确率不高,易出现误正闭环(把错误的闭环当作正确闭环);李永锋等^[9]通过改进金字塔 TF-IDF 得分匹配方法,进一步减少了闭环检测中的感知歧义;李维鹏等^[10]在进行闭环检测时,加入空间位置不确定性约束,有效地排除多歧义场景中的大部分感知歧义。

上述方法进行闭环检测时,应用场景多为静态不变的,当场景中有不断运动的物体时,得到的场景信息也是不断变化的,不断有场景信息出现或被遮挡,从而造成了本应该是产生闭环的关键帧,算法却认为是到达新的场景,导致闭环检测失败。

TAN 等^[11]使用 SIFT 描述子,利用 GPU 进行加速实现了动态场景下的单目 SLAM 算法,但是对动态场景约束条件较多,不具有较好的推广性;JOHANSSON 等^[12]提取 SURF 或者 BRIEF 特征点进行单词训练,在长时间、长距离的闭环检测实验中取得一定的成效;林志林等^[13]提出一种把场景流和虚拟地图点相结合的方法,有效区分场景中运动物体上的特征点和静态特征点,但未进行闭环检测方面的工作。

为提高动态场景下闭环检测方法的准确性,在文献[12]的基础上,结合 BOW 方法,提出一种动态场景下的闭环检测算法。首先,改进文献[12]中选取区分动静点阈值的方法,得到较为精确的动静点划分;然后,利用 BOW 方法,剔除动态特征点后进行聚类,得到仅包含关键帧静态信息场景的描述向量,改进相似度计算函数,提高两帧之间相似性判别的精度;最后,通过时间连续一致性以及对极几何约束,进一步剔除错误闭环,提高动态场景下闭环检测的准确性。实际场景实验表明,本文提出的相似度计算函数在相似程度高的环境下依然有着较好的效果,提出的闭环检测算法在动态场景中能够有效地检测闭环。

1 基于场景流的动静特征点的区分

基于场景流的动静特征点区分方法可主要划分为以下 3 部分:特征点场景流的计算、虚拟地图点的构造和运动点的判别。

1.1 特征点场景流的计算

假设相邻两帧之间的旋转变换矩阵为 \mathbf{R} , 平移变换向量为 \mathbf{t} , 这两帧之间形成匹配点对 (u, v) 和 (u', v') , 特征点对应的深度分别为 h 和 h' , 此处的相机由深度相机直接获取, 则可以获得它们对应的点在当前帧坐标系中的坐标分别为

$$[X \ Y \ Z]^T = h \cdot \mathbf{R} \cdot \mathbf{K}^{-1} \cdot [u \ v \ 1]^T + \mathbf{t} \quad (1)$$

$$[X' \ Y' \ Z']^T = h' \cdot \mathbf{K}^{-1} \cdot [u' \ v' \ 1]^T \quad (2)$$

式中, \mathbf{K} 为相机内参矩阵。该特征点对应的 3D 运动矢量即场景流为

$$\mathbf{M} = \begin{pmatrix} X' - X \\ Y' - Y \\ Z' - Z \end{pmatrix} \quad (3)$$

由于存在匹配误差、投影误差及其他因素的影响, 静态物体特征点的场景流也不一定为零, 所以这一运动矢量不能直接用于区分场景中的运动物体及静态物体。文献[12]中, 利用高斯混合模型方法, 依据场景流角度信息, 对关键帧的特征点进行分类, 得到各特征点的高斯值 ψ_p 。将高斯值较高的特征点列入待选运动点集合 $\alpha = \{p | \psi_p > \psi_{th}\}$ 中, 其中, ψ_{th} 为高斯值阈值。

但是模值不同的矢量, 角度也有可能是相同的, 因此, 不能简单地只利用场景流的角度信息来判断特征点的状态。本文提出加权平均的方法, 综合衡量场景流的角度及模值, 得到更为准确的动静点区分阈值, 详见下文。

1.2 虚拟地图点的构造

记当前时刻为 k 时刻, 上一帧时刻为 $k-1$ 时刻, 下一帧时刻为 $k+1$ 时刻。利用 $k-1$ 时刻运动物体的运动模型和 k 时刻相机的位姿, 估计出运动物体对应地图点在 $k+1$ 时刻的 3D 位置, 这些估计出来的地图点组成了虚拟地图点集。

具体地, 由式(1)、式(2)可得 $k-1$ 时刻和 k 时刻的地图点的 3D 坐标, 进而得到运动物体在两个时刻间的运动变化

$$\mathbf{M}_k = \mathbf{P}_k - \mathbf{P}_{k-1} \quad (4)$$

式中, $\mathbf{P}_k, \mathbf{P}_{k-1}$ 分别是 k 时刻和 $k-1$ 时刻特征点对应的地图点。由于相邻帧之间的间隔较小, 可近似认为运动物体是匀速运动的, 即可用 k 时刻与 $k-1$ 时刻的运动变化来近似表示 $k+1$ 与 k 时刻的运动变化, 用 $\mathbf{P}_{k+1}^{(v)}$ 表示得到的 $k+1$ 时刻的虚拟地图点, 即

$$\mathbf{P}_{k+1}^{(v)} = \mathbf{P}_k + \mathbf{M}_k \quad (5)$$

图 1 为虚拟地图点构造的原理示意图, 图 1b 中空心的点为估计出的虚拟地图点。

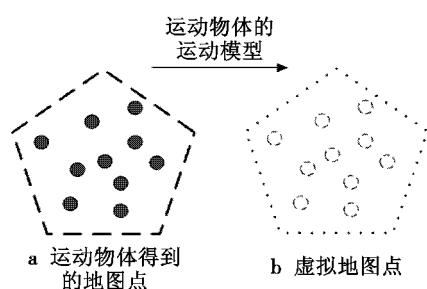


图1 虚拟地图点的构造

Fig. 1 The construction of virtual map points

1.3 运动点的判别

从1.1节中可以获得特征点初步筛选得到的待选运动点集合,但是由于存在匹配误差等影响,这样的粗划分会导致部分静态点被认作是动态特征点,或者部分动态特征点被认为是静态特征点。因此,将待选运动点集合 α 与虚拟点集合 v 进行匹配,以 ϕ_p 表示待选运动点 p 匹配的结果,即

$$\phi_p = \begin{cases} \text{true} & \text{匹配成功} \\ \text{false} & \text{匹配失败} \end{cases} \quad (6)$$

最后,用 φ_p 表示特征点 p 的运动状态,将点的状态分为 moving, unknown, static, 即

$$\varphi_p = \begin{cases} \text{moving} & \psi_p > \psi_{th} \text{ and } \phi_p = \text{true} \\ \text{unknown} & \psi_p > \psi_{th} \text{ and } \phi_p = \text{false} \\ \text{static} & \psi_p < \psi_{th} \end{cases} \quad (7)$$

1.4 改进的基于场景流动静点区分算法

由于仅利用场景流的角度信息来区分动静特征点容易造成特征点状态的误判,本文提出一种加权平均的方法,同时考虑场景流的角度及模值,得到一个相对较为准确的区分动静点的阈值。

首先,计算每一对匹配特征点场景流的模值 c 和角度 θ ;然后,求解高斯混合模型各参数,根据场景流角度对当前帧的特征点进行聚类,计算每一类特征点场景流模值的平均值。平均值最小的被认为是静态点最有可能存在的一类点,记为集合 $S = \{p_n, n = 1, 2, 3, \dots, k\}$, k 为集合 S 中特征点的个数。利用加权平均的方法,对于每一个特征点构造新的评判指标 H_n ,即

$$H_n = \frac{c_n}{\sum_{i=1}^k c_i} + \frac{|\theta_n|}{\sum_{i=1}^k |\theta_i|} \quad n = 1, 2, 3, \dots, k \quad (8)$$

式中: c_n 为集合 S 中第 n 个特征点对应场景流的模值; θ_n 为第 n 个特征点对应场景流的角度。利用集合 S 中特征点新评判指标的均值作为区分动静点的阈值 ψ_1 ,即

$$\psi_1 = \frac{\sum H_n}{k} \quad n = 1, 2, 3, \dots, k \quad (9)$$

则对于能计算场景流的特征点,计算对应的评判指标

H_n , 小于 ψ_1 即属于静态特征点;否则,归于待选运动点集,等待进一步与虚拟地图点匹配判别状态。可得特征点 p 的状态为

$$\varphi_p = \begin{cases} \text{moving} & \psi_p > \psi_1 \text{ and } \phi_p = \text{true} \\ \text{unknown} & \psi_p > \psi_1 \text{ and } \phi_p = \text{false} \\ \text{static} & \psi_p < \psi_1 \end{cases} \quad (10)$$

2 改进的基于BOW闭环检测算法

传统的基于BOW的闭环检测算法不能够在含有运动物体的场景中正常工作,为提高动态场景下闭环检测的准确率,本文提出了一种基于BOW改进的闭环检测算法,主要包括3个方面:剔除动点后场景描述、改进的金字塔TF-IDF得分匹配方法、闭环确认。

2.1 剔除动点后的场景描述

针对静态场景,传统的BOW算法先从图像中提取特征进行聚类,从而构造视觉单词,然后采用单词的统计直方图对场景进行描述。

本文考虑场景中运动物体上会有部分特征点,在使用第1章中的方法区分出动静特征点的基础上,剔除动态的特征点,然后进行聚类,利用图像在视觉词典树各个节点的TF-IDF熵作为图像在该视觉单词的得分权重,从而得到单词的得分向量对场景进行描述。

假设树的分支数为 k ,层数为 L ,对每一分支递归调用 K 均值聚类算法,从而得到 k 个子分支,直到第 L 层,如图2所示。将每个分支特征的聚类中心作为该分支的节点,其描述向量作为视觉单词,从而建立视觉词典树。

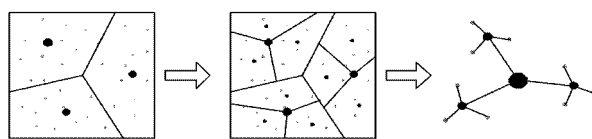


图2 视觉词典树生成方法

Fig. 2 Generation of visual vocabulary tree

假设从图像 X 中提取 n 个 d 维特征: $X = \{x_1, x_2, \dots, x_n\}$, $x_i \in \mathbf{R}^d$,按照图2方法构建视觉词典树,将图像在各个树节点的TF-IDF熵作为图像在该视觉单词的得分权重。定义TF-IDF熵为

$$\omega_i^l(X) = \frac{n_i}{n} \ln \frac{N}{N_i} \quad (11)$$

式中: $l \in \{0, 1, \dots, L\}$; $i \in \{1, 2, \dots, k^l\}$; $\omega_i^l(X)$ 表示图像 X 在视觉词典树的第 l 层的第 i 个节点 O_i^l 处的投影得分; N 表示待处理图像总数; n_i 表示图像 X 投影到节点 i 的特征数; N_i 表示至少有1个特征投影到节点 i 的图像数。

记图像 X 的场景描述向量为

$$\mathbf{W}(X) = (\mathbf{W}^1(X), \mathbf{W}^2(X), \dots, \mathbf{W}^l(X), \dots, \mathbf{W}^L(X)) \quad (12)$$

式中, $\mathbf{W}^l(X) = (\omega_1^l(X), \omega_2^l(X), \dots, \omega_k^l(X))$ 表示图像 X 在视觉词典树的第 l 层的得分向量。

2.2 改进的金字塔 TF-IDF 得分匹配方法

通过计算 2 帧图像的视觉单词得分向量之间的相似性来进行闭环检测,判断闭环是否发生。但是实际场景中会出现很多相似性场景,引起感知歧义,导致误正闭环,最终会导致地图的一致性遭到破坏。

根据文献[5],图像 X 和 Y 在单个节点 O_i^l 的相似性得分可通过最小值函数求得

$$S_i^l(X, Y) = \min\{\omega_i^l(X), \omega_i^l(Y)\}。 \quad (13)$$

文献[8]中对此方法的不合理性进行了分析,假设对于 $\omega_i^l(Y_1), \omega_i^l(Y_2)$, 满足 $\omega_i^l(Y_1) > \omega_i^l(Y_2) > \omega_i^l(X)$, 则根据式(13), 存在 $S_i^l(X, Y_1) = S_i^l(X, Y_2)$, 即在节点 O_i^l 处 X 和 Y_1 的相似性得分与 X 和 Y_2 的相似性得分相同, 也就是表明 X 与 Y_1, Y_2 在节点 O_i^l 处的相似性相同。得分越相近的 2 幅图像相似性越高, 显然在节点 O_i^l 处 X 和 Y_2 相似性更高, 因此文献[5]中的公式不能有效避免这种感知歧义的情况, 文献[8]就利用反比例函数代替 2 幅图像之间的相似性得分函数, 即

$$S_i^l(X, Y) = \frac{1}{|\omega_i^l(X) - \omega_i^l(Y)| + 1}。 \quad (14)$$

但是, 对于相似性程度高的场景, 即当 $\omega_i^l(Y_1) > \omega_i^l(Y_2) > \omega_i^l(X)$ 而 $|\omega_i^l(X) - \omega_i^l(Y_1)|$ 与 $|\omega_i^l(X) - \omega_i^l(Y_2)|$ 相差较小时, $S_i^l(X, Y_1)$ 与 $S_i^l(X, Y_2)$ 相差几乎可以忽略不计, 因此式(14)得到的相似度分数区分度较低, 不能解决这种高度相似场景导致的感知歧义问题。本文提出一种负指数幂函数作为 2 幅图像之间的相似性得分函数, 即

$$S_i^l(X, Y) = e^{-|\omega_i^l(X) - \omega_i^l(Y)|}。 \quad (15)$$

负指数幂函数与反比例函数具有同样的优点: 一方面, 负指数幂函数使得 X 和 Y 在节点 O_i^l 的相似性得分与二者的 TF-IDF 熵之差成反比, 即两者在该节点的 TF-IDF 熵越接近, 其相似性得分越高; 另一方面, 保证 2 帧图像在单个节点的相似性得分范围控制在 $(0, 1]$ 内, 避免相似性得分受某一节点的影响过大, 提高了 2 帧图像相似性得分的可靠性。另外, 对于高度相似场景导致的感知歧义问题, 由于式(15)中负指数幂函数在 $|\omega_i^l(X) - \omega_i^l(Y)|$ 接近零时斜率变化比反比例函数大, 故 $|\omega_i^l(X) - \omega_i^l(Y_1)|$ 与 $|\omega_i^l(X) - \omega_i^l(Y_2)|$ 相差较小时, 式(15)得到的相似性得分 $S_i^l(X, Y_1)$ 与 $S_i^l(X, Y_2)$ 差别明显, 区分度较好, 有利于排除此种感知歧义问题。

2.3 闭环确认

针对候选闭环中存在的误正闭环问题, 本文利用

时间连续性和对极几何约束^[14]来剔除误正闭环。

机器人获得的场景图具有时间连续性, 因此闭环一般在连续的几帧图像中均会发生。假设 k 时刻图像与之前 r 时刻发生闭环, 则接下来几个时刻会依次发生闭环(如 $k+1$ 时刻与 $r+1$ 时刻, $k+2$ 时刻与 $r+2$ 时刻也发生闭环), 一旦出现不满足时间连续性要求的情况, 如检测到 $k+1$ 时刻与 $r-2$ 时刻发生闭环, 则认为该闭环为误正闭环, 可予以剔除。

闭环发生时, 两幅图像通常是对同一场景的不同视角成像, 应当满足对极几何约束。利用发生闭环的两幅图像间的位姿, 进行投影计算内点数, 判断二者是否满足对极几何约束。若不满足, 则认为该闭环为误正闭环, 应从闭环候选中删除。

3 动态场景下闭环检测算法

本文针对动态场景下闭环检测不稳定、常发生失败的问题, 提出一种能适用于动态场景下的闭环检测算法。图 3 为本文算法的示意图。

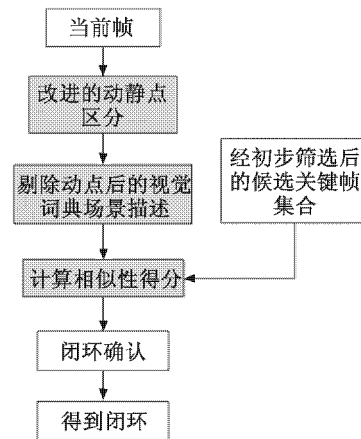


图 3 动态场景下闭环检测算法示意图

Fig. 3 Schematic of the LCD algorithm in the dynamic scenes

图中阴影部分即为本文的创新部分。值得说明的是, 经过初步筛选后的候选关键帧集合, 是采用 ORB-SLAM2 中候选闭环帧获得方法得到的。

候选关键帧的选取条件主要有: 1) 与当前帧的共有单词数超过一定阈值的帧, 但不包括当前帧的共视帧; 2) 与当前帧的相似性得分大于当前帧与其共视帧的最低得分。这里的图像帧都是经过动静点区分, 然后剔除动点后得到场景描述的。通过共有单词数量的判断, 能筛选出大多数无关的帧, 极大地减少了比较次数, 从而提高了闭环检测的实时性。

4 实验与分析

本文以实验室中的模拟室内场景为实验场景, 该场景范围为 $8.0 \text{ m} \times 5.0 \text{ m}$ 。实验使用 Xtion 传感器,

在场景中巡视一周。实验所用电脑配置 CPU 为 I7 处理器,主频 2.5 GHz,内存 4 GiB,不使用 GPU 加速,系统为 Ubuntu14.04。

图 4 展现了 ORB-SLAM2 与本文算法提取的特征点。因为 ORB-SLAM2 算法没有对静态点和运动物体上的特征点进行区分,所以从图 4a 中可以看出,有很大一部分特征点在人身上。本文算法采用了改进的基于场景流的动静特征点区分方法,所以对动静特征点进行了区分,图 4b 中红色点即为本文算法得到的动态特征点,可以看出,本文采用的动静点的区分算法作用明显,为后续得到闭环检测环节打下良好基础。

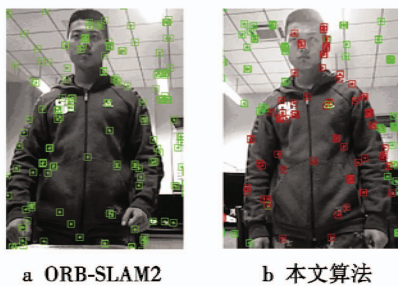


图 4 不同算法提取特征点

Fig.4 Feature points extracted by different algorithms

为了展示本文提出的相似度得分函数的作用,特地设置高度相似场景下的实验。选取走廊这种相似程度较高的场景,相机的运动轨迹如图 5 红线所示。



图 5 高度相似场景

Fig.5 Scenes with a high degree of similarity

实验中分别以反比例函数和本文提出的负指数幂函数作为相似性得分函数,进行闭环检测。图 6 是在两种相似性得分函数下闭环检测的耗时对比,因为实验中只检测到发生一次闭环,所以得到的闭环检测时间只有一次,其余关键帧到来时,未检测到闭环发生,所以耗时为零。图 6 中进行闭环检测时,本文算法用时约为 743.64 ms,对比算法的用时约为 745.08 ms,耗时相差不大,但本文算法在第 457 个关键帧时就能够判断出发生闭环,而对比的反比例函数的算法是在第 468 个关键帧才得到同样的结果,充分说明负指数幂函数作为得分函数,在场景相似度高,区分度更好,更有利于及早发现闭环,排除错误闭环发生的隐

患。图 7 和图 8 分别是两种方法得到的闭环检测结果。从图 7a、图 8a 中比较出两次实验都能检测到闭环发生,但是通过图 7b 和图 8b、图 7c 和图 8c 比较发现,本文提出的负指数幂函数作为相似性得分函数,找到的闭环帧更接近闭环查询帧。这是因为在相似度较高的场景中,负指数幂函数得到的相似性得分区别度更高,使得找到的闭环关键帧更为准确。

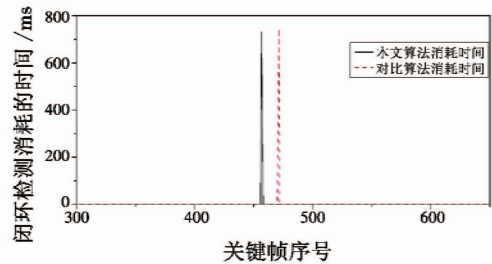


图 6 高度相似场景下闭环检测耗时对比

Fig.6 Time-lapse comparison of LCD in a high-similarity environment

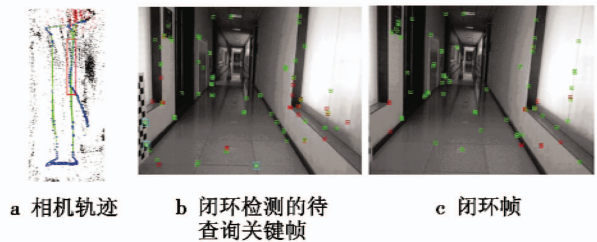


图 7 采用反比例函数得到的闭环检测结果

Fig.7 LCD result of inverse proportional function

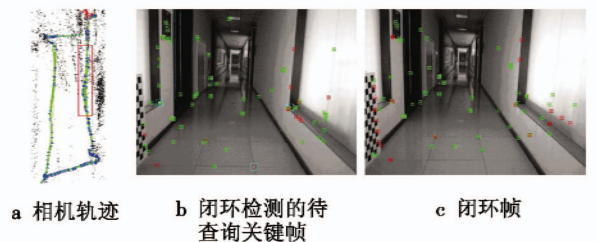


图 8 采用负指数幂函数得到的闭环检测结果

Fig.8 LCD result of negative exponential power function

设置动态场景的实验场景:相机装置围绕实验室场景进行运动,同时场景中不存在不断移动的人。在动态场景下,ORB-SLAM2 算法因受到运动的人的干扰未能检测到闭环,本文提出的基于场景流的改进闭环检测算法,因划分出动态的特征点,排除了动态特征点的干扰,因此能够检测到闭环发生,耗时约 763 ms。

图 9 中 4 幅子图分别展示动态场景下本文算法及 ORB-SLAM2 对比算法得到的轨迹。图中红色矩形框部分是相机回到之前到达的区域,即会有闭环发生的阶段,并且在实验场景中含有正在行走的人。

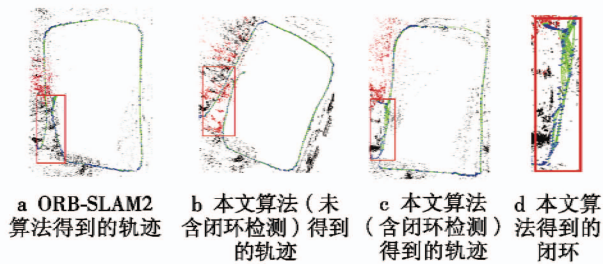


图 9 实验室动态场景下闭环检测

Fig. 9 LCD under dynamic scenes in the lab

从图 9a 中可以看出, ORB-SLAM2 算法在红色矩形区域内并未检测出闭环, 并且由于场景中存在运动的人, ORB-SLAM2 算法未对运动物体上的特征点进行区分, 导致该算法得到的轨迹误差较大, 与实际矩形的实验室场景相比, 轨迹有较大的变形。图 9b 是仅用本文算法(未包含闭环检测部分)得到的相机轨迹, 可以看出虽未发生闭环, 但是经过动静点区分后的算法得到的轨迹更接近矩形, 变形比 ORB-SLAM2 算法小, 得到的轨迹更精确。图 9c 是由本文算法(包含闭环检测部分)得到的相机轨迹。从图中可以看出, 本文算法在红色矩形部分即动态场景区域检测到了闭环, 并对相机轨迹进行了优化, 使得到的轨迹更为准确。图 9d 为图 9c 中红色矩形部分的细节, 其中绿色的横向线条表示 2 帧之间发生了闭环。图 9 中 4 幅图像的比较结果表明本文算法在动态场景下依然能够顺利检测出闭环, 从而证明了本文算法的有效性。

5 结束语

本文针对动态场景下闭环检测不稳定、易失败的问题, 提出了一种动态场景下闭环检测算法。通过改进场景流的方法, 使得动静特征点的划分更加准确; 剔除动态特征点后, 使用得到的 BOW 得分向量对动态场景进行描述; 最后利用负指数幂函数计算 2 帧图像之间的相似性得分, 经过闭环确认后得到最终检测到的闭环。通过实际场景中与 ORB-SLAM2 算法进行对比, 证明了本文算法的有效性。下一步计划使用哈希表进行图像搜索, 加快闭环候选帧的筛选, 提高闭环检测的实时性, 并利用检测到的闭环信息进一步优化得到的地图与相机位姿。

参考文献

[1] 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.

- [2] 朱平, 甄子洋, 覃海群, 等. 基于立体视觉和光流的无人机避障算法研究[J]. *电光与控制*, 2017, 24(12): 31-35.
- [3] CHEN J B, LI J, XU Y, et al. A compact loop closure detection based on spatial partitioning[C]//*The 2nd International Conference on Image, Vision and Computing*, 2017: 371-375.
- [4] ANGELI A, FILLIAT D, DONCIEUX S, et al. Fast and incremental method for loop-closure detection using bags of visual words[J]. *IEEE Transactions on Robotics*, 2008, 24(5): 1027-1037.
- [5] WILLIAMS B P, CUMMINS M, NEIRA J, et al. An image-to-map loop closing method for monocular SLAM[C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008: 2053-2059.
- [6] 李博, 杨丹, 邓林. 移动机器人闭环检测的视觉字典树金字塔 TF-IDF 得分匹配方法[J]. *自动化学报*, 2011, 37(6): 665-673.
- [7] CUMMINS M, NEWMAN P. Highly scalable appearance-only SLAM—FAB-MAP 2.0[C]//*Robotics, Sciences & Systems*, 2009: 1-8.
- [8] LABBE M, MICHAUD F. Appearance-based loop closure detection for online large-scale and long-term operation[J]. *IEEE Transactions on Robotics*, 2013, 29(3): 734-745.
- [9] 李永锋, 张国良, 王峰, 等. 一种基于历史模型集的改进闭环检测算法[J]. *机器人*, 2015, 37(6): 663-673.
- [10] 李维鹏, 张国良, 姚二亮, 等. 基于空间位置不确定性约束的改进闭环检测算法[J]. *机器人*, 2016, 38(3): 301-310, 321.
- [11] TAN W, LIU H M, DONG Z L, et al. Robust monocular SLAM in dynamic environments[C]//*IEEE International Symposium on Mixed and Augmented Reality*, 2013: 209-218.
- [12] JOHANNSSON H, KAESSE M, FALLON M, et al. Temporally scalable visual SLAM using a reduced pose graph[C]//*Proceedings of IEEE International Conference on Robotics and Automation*, 2013: 54-61.
- [13] 林志林, 张国良, 姚二亮, 等. 动态场景下基于运动物体检测的立体视觉里程计[J]. *光学学报*, 2017, 37(11): 1115001-1-1115001-9.
- [14] 李卓, 刘洁瑜, 周伟. 基于几何约束的视觉闭环检测和位姿优化[J]. *电光与控制*, 2018, 25(5): 79-83.