

基于 YOLO v3 和传感器融合的机器人定位建图系统

陈文峰 张学习 蔡述庭 熊晓明

(广东工业大学自动化学院, 广东 广州 510006)

摘要: 场景中的动态物体影响移动机器人定位算法的精度, 使机器人无法建立蕴含场景信息的高精度地图, 降低定位建图系统在复杂场景中的鲁棒性。针对目前主流动态 SLAM 技术受限于系统需求和硬件性能, 无法兼顾移动机器人定位精度和系统实时性的问题, 提出一种基于 YOLO v3 和传感器融合的机器人定位建图系统。首先, 建立融合编码器和视觉传感器的机器人运动模型, 求解移动机器人位姿; 然后, 利用深度学习技术剔除复杂场景中的动态物体, 并针对 YOLO v3 目标检测网络特点, 采用多视图几何方法进行性能优化; 最后, 经测试, 本系统相比 DS_SLAM 具有更优的轨迹精度, 耗时更短。

关键词: 传感器融合; 目标检测; 动态物体; 定位; 多视图几何

中图分类号: TP830.1

文献标识码: A

文章编号: 1674-2605(2021)02-0007-06

DOI: 10.3969/j.issn.1674-2605.2021.02.007

0 引言

同步定位和地图构建 (Simultaneous Localization and Mapping, SLAM) 是一种利用传感器获取移动机器人在环境中的运动信息和构建未知场景地图的技术^[1], 广泛应用于机器人、未知领域 (行星、空中、陆地、海洋等) 探索、高风险地区搜索救援任务、虚拟现实和自动驾驶等领域^[2]。

近年来, 视觉 SLAM (vSLAM) 系统由于传感器成本低廉、性能不俗, 受到研究人员的广泛关注。经典的 vSLAM 系统在理想室内静态场景内, 已经相当成熟。目前, 动态场景下 vSLAM 系统的定位和建图问题已成为国内外研究的热点。采用多传感器融合替换单一传感器, 常见的是融合 IMU 传感器的 SLAM 系统, 如港科大团队发布的 VINS-Mono^[3]和最新的 ORB_SLAM3^[4]都采用这种方案; 融合深度学习方法的 SLAM 系统可解决动态物体对机器人建图的干扰, 如 ClusterSLAM^[5]利用 K-means 算法对像素点分簇, 计算不同簇的运动模型, 恢复物体运动; DS_SLAM 采用语义分割方法分离图像的前景和背景, 利用帧间几何一致性判断前景是否为动态物体; KinectFusion^[6]和 Static Fusion^[7]通过聚类对图像像素点分簇, 为每一簇构造独立的运动模型, 然后进行三维重建, 并将三

维重建的投影与采集图像进行比对和优化。

本文在已有研究成果的基础上, 采用深度相机和编码器采集数据, 通过非线性优化的方式融合传感器数据; 利用 YOLO v3 网络分离关键帧中的动态物体, 通过帧间几何一致性判别是否为动态物体; 利用多视图几何重投影方法, 减少目标检测次数, 提高目标检测线程的性能。

1 移动机器人运动模型分析

1.1 编码器模型分析

利用编码器数据, 根据运动模型计算两帧之间机器人位姿的相对变化, 构建移动机器人运动模型。假设移动机器人在二维平面运动, A 点是 t 时刻机器人的坐标, B 点是 $t+1$ 时刻机器人的坐标, 已知移动机器人轮距 b 以及 t 和 $t+1$ 时刻编码器的读数差值 Δe_{lr} 。从 A 点到 B 点, 根据机器人两轮运动速度的不同, 有 3 种模型描述机器人轨迹。

1) 切线模型, 当机器人左右轮速度相同时, 轨迹是一条直线。假设机器人从 A 点到 B 点的轨迹是一条直线, 可得出 t 时刻位姿 $\partial_t = [x_t, y_t, \theta_t]^T$ 和 $t+1$ 时刻位姿 $\partial_{t+1} = [x_{t+1}, y_{t+1}, \theta_{t+1}]^T$ 关系为

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \Delta s \cdot \cos(\theta_t) \\ \Delta s \cdot \sin(\theta_t) \\ \Delta \theta \end{bmatrix} \quad (1)$$

式中, 机器人位移 Δs 和机器人转角 $\Delta \theta$, 可以通过左右轮的位移量 Δs_{lr} 求解:

$$\begin{cases} \Delta s = \frac{\Delta s_r + \Delta s_l}{2} \\ \Delta \theta = \frac{\Delta s_r - \Delta s_l}{b} \end{cases} \quad (2)$$

式中, 位移量 Δs_{lr} 可通过编码器读数差 Δe_{lr} 计算得到:

$$\begin{cases} \Delta s_r = k_r \cdot \Delta e_r + \delta_r, \delta_r \sim \mathcal{N}(0, \|K \cdot k_r \cdot \Delta e_r\|^2) \\ \Delta s_l = k_l \cdot \Delta e_l + \delta_l, \delta_l \sim \mathcal{N}(0, \|K \cdot k_l \cdot \Delta e_l\|^2) \end{cases} \quad (3)$$

式中, $k_{r/l}$ 和 K 为比例系数; $\delta_{r/l}$ 为高斯噪声, 噪声主要来源车轮形变、编码器自身误差、滑动误差和传动误差等;

2) 圆弧模型, 当机器人左右轮差速运动时, 轨迹是圆弧。假设机器人从 A 点到 B 点的轨迹是圆弧, 圆心在圆弧的中垂线上, 可得出 t 和 $t+1$ 时刻位姿关系为

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \frac{\Delta s}{\Delta \theta} \cdot (\sin(\theta_t + \Delta \theta) - \sin \theta_t) \\ \frac{\Delta s}{\Delta \theta} \cdot (-\cos(\theta_t + \Delta \theta) + \cos \theta_t) \\ \Delta \theta \end{bmatrix} \quad (4)$$

3) 割线模型, 圆弧模型计算较为复杂, 实际中使用最多的是割线模型。假设机器人沿圆弧的割线方向移动, 得到 t 和 $t+1$ 时刻位姿关系为

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \Delta s \cdot \cos(\theta_t + 0.5\Delta \theta) \\ \Delta s \cdot \sin(\theta_t + 0.5\Delta \theta) \\ \Delta \theta \end{bmatrix} \quad (5)$$

1.2 编码器和视觉传感器融合

单目相机是利用针孔模型成像的一种传感器, 假

设 P 点坐标为 $[X, Y, Z]^T$, 经过光心 O 之后投影在成像平面上 P' , P' 在像素坐标系下的坐标为 $[u, v, 1]^T$, 假设成像平面与光心 O 距离为 f , 根据小孔成像原理, 可得

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq KP \quad (6)$$

式中, $f_{x/y}$ 为成像平面到像素坐标的缩放比例; $c_{x/y}$ 为成像平面到像素坐标的位移; K 为相机内参, 通过相机厂家数据或标定得到。

为融合编码器和相机数据, 选取机器人初始位姿建立世界坐标系, SLAM 系统的地图点构建在世界坐标系下。假设当前帧有一像素点 u , 若从像素点 u 得到地图点 wP , 首先, 将像素点转化为相机坐标下的 P 点; 然后, 利用机器人外参矩阵将 P 点转化为机器人坐标; 最后, 利用已知的编码器数据将机器人坐标转化为世界坐标 (第一帧机器人坐标):

$$P_w = {}^wT_{r_k} {}^r_kT_c {}^cT \cdot \pi(u) \quad (7)$$

同理, 地图点 wP 重投影到像素坐标的公式为

$$u = \left({}^wT_{r_k} {}^r_kT_c {}^cT \right)^{-1} \cdot \pi^{-1}(P_w) \quad (8)$$

式中, $\pi(\cdot)$ 为针孔模型下像素坐标到三维相机坐标的转换函数; cT 为移动机器人外参, 描述机器人坐标和相机坐标的相对位姿, 可通过手眼标定方法得到; r_kT 为当前帧机器人坐标到第 K 帧关键帧机器人坐标的变换矩阵; ${}^wT_{r_k}$ 是第 1 帧机器人坐标到第 K 帧机器人坐标的变换矩阵, 这 2 个变换矩阵可通过编码器数据计算得出。

采用非线性优化方法实现编码器和相机数据的融合, 需求解当前帧在世界坐标下的位姿 wT_c 。由地图点重投影到像素坐标下的过程可知, 误差主要来源于编码器误差和相机重投影误差 2 部分。通过最小化编码器误差和相机重投影误差得到当前帧在世界坐

标下的位姿。

$${}^wT_{r_c}^* = \arg \min_{{}^wT_{r_c}} (E_{\text{enc}} + E_{\text{proj}}) \quad (9)$$

其中编码器误差为

$$E_{\text{enc}} = \rho \left((\mathbf{e}^e)^T \begin{pmatrix} r_k \\ r_c \end{pmatrix} \Sigma^e \begin{pmatrix} r_k \\ r_c \end{pmatrix} \right) \quad (10)$$

$$\mathbf{e}^e = \begin{pmatrix} r_k \\ r_c \end{pmatrix} \zeta^e - G \left(\left(\begin{pmatrix} {}^wT_{r_k} \\ {}^wT_{r_c} \end{pmatrix} \right)^{-1} {}^wT_{r_c} \right) \quad (11)$$

式中， $\rho(\cdot)$ 为 Huber 损失函数； $G(\cdot)$ 将 4×4 矩阵转化为 3×1 向量。

编码器误差协方差矩阵可由编码器噪声模型得到。

2 目标检测网络与 SLAM 系统融合

2.1 深度学习与 SLAM 系统融合

本文采用目标检测方法，兼顾实时性和识别精度；采用 YOLO v3 目标检测网络，在开源 COCO 数据集进行训练。COCO 数据集是一个大型的、具有丰富场景的目标检测和语义分割数据集，可提供 80 个类别的分类标签。在传统 SLAM 系统框架中增加目标检测线程，YOLO v3 网络可处理图片 20 张/s。Kinect 深度相机帧率为 20~30，如果对每一帧图像都进行目标网络检测，目标检测线程会成为系统性能瓶颈，因此，本文采取多视图几何法优化目标网络检测。

2.2 多视图几何法优化目标网络检测

动态像素主要通过 2 方面影响特征点法 SLAM 系统的轨迹精度：1) 物体运动造成的特征点误匹配影响求解位姿精度；2) 大量动态特征点成功特征匹配，影响随机抽样一致性估算的位姿值。

本文采用编码器和相机传感器融合的 SLAM 方案，关键帧遴选通过编码器数据决定。当编码器计算机器人位移累计达到 0.3 m，转角达到 0.5 rad 时，将当前帧判定为关键帧。首先，对相邻 2 个关键帧进行目标检测，形成多个框住动态物体的长方形框；然后，利用中心像素的深度值将长方形框投影到世界坐标系，设第 k 帧关键帧长方形框 4 个像素点坐标为 p_i ，对应世界坐标为 P_i ，第 $K+1$ 帧关键帧像素点坐标为 p'_i ，对应世界坐标为 P'_i ，可得

$$P_i = T_{c_k}^w T_c^r \cdot \pi(p_i) \quad (12)$$

$$P'_i = T_{c_{k+1}}^w T_c^r \cdot \pi(p'_i) \quad (13)$$

利用投影到三维的长方形框剖面表征动态物体。假设运动物体在关键帧之间的运动是匀速直线的（由于关键帧时间短，此假设在大部分场景下成立），可计算普通帧对应的每一个时刻表征动态物体矩形框在世界坐标系的坐标，如图 1 所示。假设关键帧中有 n 帧普通帧，第 j 帧对应的动态物体矩形框在世界坐标系的坐标 P_{ij} 可通过相邻关键帧的矩形框坐标计算：

$$P_{ij} = \frac{j}{n-1} (P'_i - P_i) + P_i \quad (14)$$

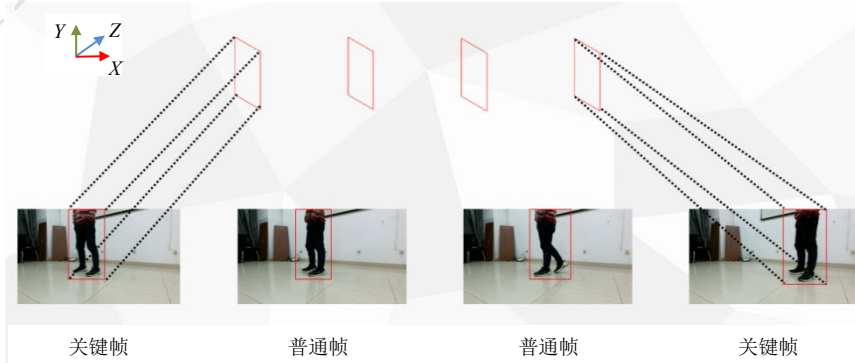


图 1 目标检测效果重投影图

将其重投影到每一个普通帧上，用投影四边形框替代普通帧进行目标检测，减少系统进行目标检测的开销，使系统满足实时性要求。

$$p_{ij} = (T_{c_k}^w T_c^r)^{-1} \cdot \pi^{-1}(P) \quad (15)$$

式中， p_{ij} 为第 j 帧普通帧上动态物体检测的矩形框像素坐标。

2.3 利用几何一致性判别动态物体

利用 YOLO v3 完成目标检测后，需判别检测的物体是否为动态物体。首先，利用图像金字塔提取图像特征点，并与最邻近的关键帧进行特征点匹配；然后，利用随机采样一致性选取内点计算基础矩阵 F ，基础矩阵 F 反映同一物体点在 2 个成像平面投影点之间的约束关系；当物体不动时，约束关系不会发生改变。假设两帧之间存在同一物体上的投影点 P_1 和 P_2 的坐标分别为

$$p_1 = [u_1, v_1, 1]^T \quad (16)$$

$$p_2 = [u_2, v_2, 1]^T \quad (17)$$

理想情况下，投影点和基础矩阵满足对极约束：

$$e = p_2^T F p_1 = 0 \quad (18)$$

由于重投影误差，即使物体静态的情况下， e 仍不为 0。设定一个误差值 ε ，当 $e > \varepsilon$ ，则认为该特征点对为动态特征点对。当目标检测的 1 个目标中 50% 的点为动态特征点，就认为被检测的物体为动态物体。

3 实验结果与分析

实验利用 Turtlebot3 机器人，配置为 Intel E3 CPU，P2000 GPU 和 32 GB 内存。本文 SLAM 系统搭建在 ROS 平台，分为 Tracking, Local Mapping, Loop Closing, Detecting 线程。在 DRE_SLAM 团队开源数据集上进

行测试。数据集分为 ST, LD 和 HD 3 类，分别代表环境中物体运动为静止、少量物体运动和大量物体运动。数据集提供 RGBD 相机和编码器数据，并提供 groundtruth 值可对比系统运行结果和实际运动值的误差。

本系统和 DS_SLAM 在 HD, LD 和 ST 3 个数据集下相机位姿误差的均方根、均值和中值的对比表如表 1 所示。

表 1 本系统与 DS_SLAM 轨迹误差对比

数据集	DS_SLAM			OURS		
	均方根	均值	中值	均方根	均值	中值
HD	0.1704	0.1526	0.1380	0.0211	0.0195	0.0198
LD	0.1156	0.1059	0.1017	0.0256	0.0233	0.0213
ST	0.1570	0.1366	0.1323	0.0227	0.0187	0.0154

由表 1 可看出：本系统在 3 个数据集下都具有更优的轨迹精度。

本系统运行 3 个数据集平均每个线程耗费的时间如表 2 所示。

表 2 本系统各线程耗时

线程	运行频率	耗时/ms
Tracking	每帧	38.27
Detecting	关键帧	55.13
Local Mapping	关键帧	140.07
Loop Closure	触发回环	243.31

由表 2 可知：目标检测线程在新关键帧产生时工作，仅耗时 55 ms，确保了系统实时性。普通帧个数是关键帧的 20 倍左右，没有经过多视图几何优化的目标检测性能将降低近 20 倍，同时目标检测网络的效果可以覆盖到每一个普通帧，在关键帧频率较高情况下，有较好效果。普通帧目标检测效果图如图 2 所示。

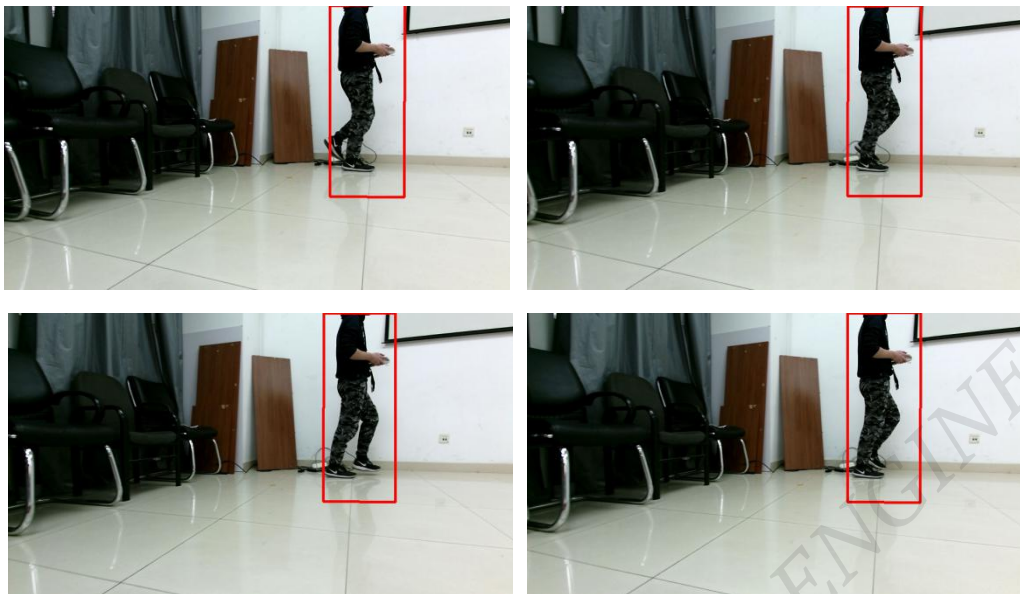


图2 普通帧目标检测效果图

由图2可看出：普通帧中的动态物体基本可被识别，运动幅度较大的物体出现识别不全的情况。

4 结论

本文对动态场景下移动机器人定位问题进行讨论，利用多传感器融合解决了移动机器人在动态场景鲁棒性降低的问题，给出编码器和相机运动模型，分析2种传感器模型误差的来源，并利用非线性优化最小化误差的方式实现了传感器融合。本文的SLAM系统融合深度学习中标检测网络，进一步排除动态物体对帧间匹配和三维建图的干扰；同时利用多视图几何法，将目标检测的效果从关键帧投影到普通帧中，缩减了目标检测线程的开销。目前系统还存在缺陷，之后的研究工作将从2方面进行优化：1）解决编码器传感器在打滑情况下，数据出现失真的问题；2）将关键帧目标检测效果重投影到普通帧后，提高目标检测的精度。

参考文献

[1] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: toward the

robust-perception age[J]. IEEE Transactions on Robotics, 2016,32(6):1309-1332.

[2] FUENTES-PACHECO J, RUIZ-ASCENCIO J, RENDÓN-MANCHA J M. Visual simultaneous localization and mapping: a survey[J]. Artificial Intelligence Review, 2015,43(1):55-81.

[3] Tong Q, Li P, Shen S. VINS-mono: a robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2017(99):1-17.

[4] Campos C, Elvira R, JGG Rodríguez, et al. ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map SLAM[J]. Under review,2020.

[5] HUANG J, YANG S, ZHAO Z, et al. ClusterSLAM: a SLAM backend for simultaneous rigid body clustering and motion estimation[C]// ICCV 2019, 2019.

[6] Newcombe Richard A, Shahram Izadi, Otmar Hilliges, et al. KinectFusion: real-time dense surface mapping and tracking[C]. IEEE International Symposium on Mixed & Augmented Reality IEEE, Basel, Switzerland, 2011.

[7] SCONA R, JAIMEZ M, PETILLOT Y R, et al. StaticFusion: background reconstruction for dense RGB-D SLAM in dynamic environments[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018: 3849-3856.

(下转第48页)