第 49 卷 第 6 期/2022 年 3 月/中国激光 研究论文

中国激光

基于动态特征剔除的图像与点云融合的机器人位姿估计方法

张磊[1,2]，徐孝彬[1,2,3,4*]，曹晨飞[1,2]，何佳[1,2]，冉莹莹[1,2]，谭治英[1,2]，骆敏舟[1,2]

[1]河海大学机电工程学院，江苏 常州 213022；

[2]河海大学江苏省特种机器人技术重点实验室，江苏 常州 213022；

[3]南京航空航天大学机电学院，江苏 南京 210016；

[4]常州常工电子科技股份有限公司，江苏 常州 213001

摘要 针对动态物体影响传感器进行机器人位姿估计的问题，本文提出了一种基于动态特征剔除点云与图像融合的位姿估计方法。首先，YOLOv4 和 PointRCNN 分别被用于识别图像和点云中的潜在运动目标并提取候选框。其次，在视觉定位方面，双目视觉与稀疏光流被用于路标点的构建与追踪，并根据候选框剔除动态特征点，随后构建重投影误差函数，通过基于 RANSAC 剔除的非线性优化方法求解相机位姿；在激光定位方面，提取前后帧的直线与平面特征点，并根据候选框进行筛选，基于特征点到直线或平面的距离构建误差函数，进而求解激光雷达位姿。为使系统不再局限于单一传感器的使用环境限制，通过自适应加权方法，有效融合了两种位姿结果。最后，通过 KITTI 数据集和动态场景采集的数据进行定量实验对比，验证了剔除动态特征后的位姿估计的精确性以及融合算法的有效性。

关键词 机器视觉；机器人定位；自适应融合；激光雷达；图像；深度学习

中图分类号 TP391 **文献标志码** A **doi**：10.3788/CJL202249.0610001

1 引 言

随着科技的发展，面向实际工程的机器人技术得到了迅速发展，并在人们的日常生活得到了广泛应用。机器人相关技术在工业生产[1]、测绘[2]、灾难救援[3]、军事[4]、家庭生活[5]等领域得到了一定发展，其中的机器人定位技术是机器人导航的关键一环，定位的准确性直接影响机器人导航的准确性[6-7]。全球定位系统(GPS)和惯性传感器(IMU)组合导航是常用的定位方式[8]，然而，当机器人在野外、隧道、地下等环境时，GPS 信号常会因为被遮挡而发生丢失。单一的激光雷达或视觉传感器定位往往受限于特定的使用环境，因此，多传感器融合的定位方式具有重要的应用价值。

在激光定位方面，Kohlbrecher 等[9]提出了能够实现地面搜救的 Hector SLAM 算法。Hector SLAM 算法通过双线性插值法将不同点的概率值分解到不同的栅格中，并结合高斯牛顿迭代算法来获取前后帧的匹配结果。Menna 等[10]通过对激光雷达扫描点云进行法线估计和曲率计算实现了点云分割，并对地面、墙、楼梯和可通行障碍物这四类物体进行了标记，同时通过迭代最近点(ICP)算法实现了机器人定位。Li 等[11]提出了矿用机器人定位算法，他们采用正态分布变换算法对帧间进行匹配，创新性地提出将地面与墙面也作为约束来实现位姿图优化的方法，以此实现机器人定位，并将实验结果与 LeGO-LOAM(Lightweight and Ground-Optimized Lidar Odometry and Mapping)算法[12]的结果进行了对比，论证了所提方法具有更高的精度；但是，在实际工作条件下很难找到平整路面与墙体特征。

收稿日期：2021-06-09；**修回日期**：2021-07-14；**录用日期**：2021-08-10

基金项目：国家自然科学基金(51805146)、江苏省重点研发计划(BE2020082-1)、中央高校基本科研业务费专项资金资助项目(B200202221)、江苏省博士后科研资助计划(2020Z138)

通信作者：*xxbtc@hhu.edu.cn

最开始的单目视觉定位系统是由 Davison 提出的纯视觉 SLAM 系统[13]，计算帧间位姿变换时，该系统将图像扭曲为卷积神经网络(CNN)匹配和扩展卡尔曼滤波(EKF)[14]。但传统的滤波器利用的信息十分有限，往往不能进一步提高全局位姿精度，因此，基于非线性优化的光束平差(BA)算法框架逐渐占据了主导地位。Zhang 等[15]和邹斌等[16]基于视觉 SLAM 框架[15]通过尺度不变特征变换(SIFT)算法提取特征，有效提高了定位的鲁棒性和稳定性。Li 等[17]通过选取关键帧建立了单目相机半稠密地图，同时对关键帧进行语义分割，建立了室内外三维(3D)语义地图。ORB-SLAM[18-19]系列是基于特征匹配的视觉定位系统的代表，该系统使用 ORB 特征快速匹配角点并创建局部和全局地图，根据相机成像模型创建重投影误差函数，使用高斯牛顿迭代对位姿和地图同时进行优化。SVO[20]将特征点法和直接法相结合，通过三角测量实现深度估计，同时分别使用最小化灰度误差和最小化重投影误差对传感器位姿和路标点坐标进行优化。

然而，单一的激光定位与图像定位都有一定局限性：激光雷达在直线和平面特征缺乏的环境下难以找到特征点对，而图像定位在提取特征点时易受环境光干扰，从而降低了定位精度。尤其是出现多个运动目标时，多目标的检测与识别会影响定位过程中特征的提取。

在激光雷达和图像融合定位[21]方面，Chen 等[22]通过识别楼梯和坡道进行状态切换，并采用最小二乘算法将视觉 SLAM 算法和 Hector-slam 算法进行融合。但是在非结构环境下，角点、直线等特征点云的数量稀少，2D 雷达不能通过提取足够多的信息实现定位。Zhang 等[23]基于点云曲率提取直线和平面上的特征点，通过点到直线和平面的距离构造误差函数，进而求解雷达位姿。Kubelka 等[24]针对有雾、烟等恶劣条件的地面搜索救援，结合机器人运动学显式建模和基于机器学习的数据驱动方法，实现了较准确的定位，使得移动机器人能够穿越室内外各类障碍物。Wang 等[25]将图像与深度图融合，基于改进的迭代最近邻点算法(ICP 算法)与 2D/3D 小波变换原理，实现了机器人的定位。上述研究表明，大多数机器人单一传感器定位研究集中在特征提取的改进上，激光雷达与图像融合研究主要集中在弥补单一传感器的信息缺失上。实际上，机器人运动在非结构化环境中，如何在运动过程中获取准确的特征信息是难点。

针对动态物体定位精度不高的问题，本文首先采用 YOLOv4[26]和 PointRCNN[27]识别图像和点云中的目标物体并提取候选框；之后，采用光流法跟踪前后帧角点，并根据候选框剔除动态特征，构造重投影误差函数，同时进行基于 RANSAC 的非线性优化，得到最佳位姿；接着，提取点云中的直线和平面特征点，并根据候选框进行筛选，同时利用点到线和面的距离构造误差函数，实现前后帧位姿解算；最后，基于特征点数量对两者的位姿进行动态融合。

2 整体框架

为降低动态物体对位姿估计的影响，本文将基于深度学习的物体检测网络引入到算法框架中，剔除目标物体的特征点，避免错误的信息被引入到误差函数中。同时，基于特征点数量对视觉与激光融合的位姿估计进行自适应动态融合。算法的整体框架如图 1 所示。

位姿估计算法分两个线程同时进行，分别对图像和点云数据进行目标特征剔除，并分别对各自的传感器位姿进行估计，最后对两者输出的位姿结果进行融合。接下来分别阐述两种位姿估计算法及融合方法。

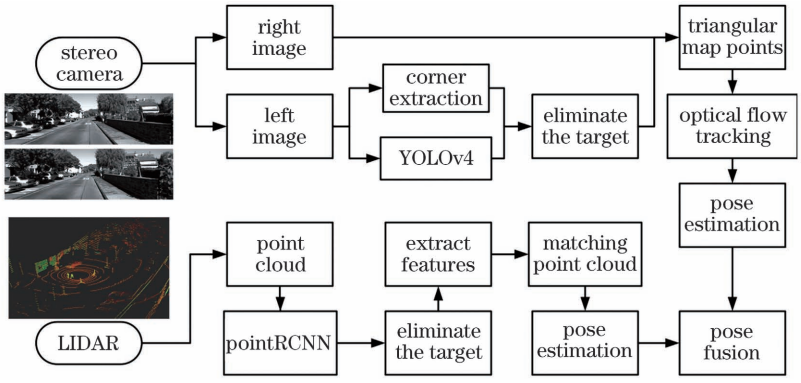


图 1 位姿估计整体框架

Fig. 1 Overall framework of pose estimation

3　位姿估计

采用双目相机和多线激光雷达作为传感器。双目相机可以更好地还原环境深度，避免尺度漂移；多线激光雷达可以计算传感器的六自由度位姿，方便后续的融合。

3.1　视觉位姿估计

视觉位姿估计采用双目相机作为视觉传感器可以更好地还原真实尺度。提取 GFTT(good feature to track)角点作为三维重建和前后帧特征匹配的特征点。首先，提取上一帧左目图像的 GFTT 角点集合 P_{GL}，根据左右目相机的极线约束，通过一维搜索找到右目图像对应的特征点集合 P_{GR}，二者形成特征点对。根据三角测量原理以及双目相机内外参数，对特征点对进行三维重建，得到对应的路标点。

采用稀疏光流法追踪左目图像前后帧的特征点坐标。假设上一帧图像 I_1 中的特征点(m,n)周围的光度不变，构造光度误差函数；设该特征点在当前帧图像 I_2 中的坐标为$(m+x,n+y)$，则误差函数可以表示为

$$\arg\min_{x,y}\| I_1(m,n)-I_2(m+x,n+y)\|。\tag{1}$$

对误差函数进行线性化，即

$$f(x+\Delta x,y+\Delta y)=$$
$$f(\boldsymbol{t}+\Delta\boldsymbol{t})\approx f(\boldsymbol{t})+\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{t})\Delta\boldsymbol{t},\tag{2}$$

式中：$\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{t})$为$(m+x,n+y)$在图像 I_2 中的像素梯度，$\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{t})=-\left[\dfrac{\Delta I_2}{\Delta(m+x)},\dfrac{\Delta I_2}{\Delta(n+y)}\right]$；$\boldsymbol{t}$ 表示像素坐标(x,y)。使用雅克比矩阵 $\boldsymbol{J}$ 迭代求解，得到当前帧最佳的特征点位置。假设成功追踪到 n 个路标点，对于已知的 n 个路标点 P_i 以及它们在当前帧的投影 p_i，需要计算当前帧在全局坐标系下的位姿，设位姿对应的李群为 $\boldsymbol{T}$。设某个空间点 P_i 的坐标为(X_i,Y_i,Z_i)，它在当前帧左图的投影像素 p_i 的坐标向量为$[u_i,v_i]^{\mathrm{T}}$。由世界坐标系与相机坐标系之间的转换关系可以构建出重投影误差函数为

$$e=\arg\min_{\boldsymbol{T}}\sum_{i=1}^{n}\left\|\boldsymbol{p}_i-\frac{1}{s_i}\boldsymbol{K}\boldsymbol{T}\boldsymbol{P}_i\right\|_2^2,\tag{3}$$

式中：s_i 为路标点的深度 Z_i；$\boldsymbol{K}$ 是左目相机的内参矩阵；$\boldsymbol{P}_i$ 为空间点 P_i 的三维坐标向量；$\boldsymbol{p}_i$ 为 p_i 的坐标向量。误差函数由所有路标点在当前帧图像中的重投影误差组成。车辆、行人等动态物体在上一帧传感器坐标系下的位置并不是固定不变的，因此其在图像中的投影并不会出现在预估的位置上。如果将这些动态物体对应的路标点同样加入到误差函数中，就会对最后的迭代优化结果造成较大影响。

为了消除动态物体对位姿估计的影响，在提取图像特征点的同时，将图像传入 YOLOv4 深度神经网络，并输出得到潜在动态目标物体的候选框。设待判断的特征点的坐标为(u_i,v_i)，候选框的范围为$[u_{\min},u_{\max}]$和$[v_{\min},v_{\max}]$，则对每一个特征点可做出如下判断：

$$\begin{cases}u_{\min}-w/2<u_i<u_{\max}+w/2\\ v_{\min}-w/2<v_i<v_{\max}+w/2\end{cases}。\tag{4}$$

(4)式中的 w 是稀疏光流追踪过程中光度误差感知范围的边长，单位是 pixel。光流追踪具有一定范围的感受野，因此必须适当扩大剔除的范围，避免较近的背景点也发生错误追踪，从而最大限度地排除动态物体对误差函数的影响。

得到剔除物体后的路标点集合 P_{last} 以及对应的当前帧的左目图像的特征点 $p_{current}$ 后，使用基于 RANSAC 改进的高斯牛顿法对相机位姿进行迭代求解，算法流程如表 1 所示。EPNP(Efficient Perspective-n-Point)[28] 被用于初始位姿的确定。

表 1　基于 RANSAC 的非线性优化算法

Table 1　Nonlinear optimization algorithm based on RANSAC

Input: P_{last}; $p_{current}$; Camera internal reference $\boldsymbol{K}$; Jacobian matrix of error function for pose $\boldsymbol{J}$
Output: Camera pose of the current frame $\boldsymbol{T}$
Use EPNP to initially solve the camera pose $\boldsymbol{T}_0$
$\boldsymbol{T}=\boldsymbol{T}_0$
Sum of reprojection errors $e_{total}=0$
for $i=1\rightarrow$The maximum number of iterations n **do**
　for $P_j\in P_{last}$　$p_j\in p_{current}$ **do**
　　Reprojection error $e_j=|\boldsymbol{p}_j-\boldsymbol{P}_j\boldsymbol{K}\boldsymbol{T}/s_i|$
　　if $e_j>$Error threshold t_0 **then**
　　　record $\boldsymbol{P}_j$ and $\boldsymbol{p}_j$
　　　continue
　　end if
　　$e_{total}=e_{total}+e_j$
　　Calculate the Hessian matrix $\boldsymbol{H}=\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}}$，以及 $\boldsymbol{g}=-\boldsymbol{J}e_j$
　　Calculate the iteration value $\Delta\boldsymbol{T}=\boldsymbol{H}^{-1}\boldsymbol{g}$
　　$\boldsymbol{T}=\Delta\boldsymbol{T}\boldsymbol{T}$
　end for
　Eliminate the recorded $\boldsymbol{P}_j$ and $\boldsymbol{p}_j$
end for

将 EPNP 求解的相机位姿作为高斯牛顿迭代的初始位姿，可以使收敛速度更快且不易陷入局部最小值。每次迭代后，将发生异常投影的路标点和特征点全部剔除，以保证收敛速度和结果的准确性。

3.2 激光位姿估计

激光位姿估计与基于视觉的位姿估计面临着相同的问题。无论是基于特征的空间点最近距离迭代，还是基于点线面特征的位姿估计，动态物体点在世界坐标系下的位置是不固定的。即使迭代过程中得到了正确的位姿，位姿也会因动态点在全局坐标系中位置的变化而发生变化，导致此时的误差函数并不接近 0，从而向错误的结果进一步迭代。

为消除动态点云对激光位姿估计的影响，需要对激光雷达的每一帧点云进行预处理。采用 PointRCNN 网络对潜在的动态物体点云进行精细分割，并得到对应的 3D 候选框。根据候选框的范围剔除点云，降低位姿估计时点云错误配准的概率。以行驶的汽车为例，点云剔除的过程如下：

设被处理后的第 k 次和第 $k+1$ 次扫描的两帧点云分别为 P_k 和 P_{k+1}，i 为 P_k 中的一个点，$i\in P_k$；设 S 为激光雷达在同一扫描中返回的与 i 连续的点集，S 包含 i 前后的各一半点，即 i 在 S 的中间，且 S 中两个点之间间隔相同的角度。定义一个参数 α 来评估局部曲面的平滑度，根据 c 选取每一帧的直线和平面点[23]。参数 α 的表达式为

$$\alpha=\left\|\sum_{j\in S,j\neq i}\left(X_{(k,i)}^{L}-X_{(k,j)}^{L}\right)\right\|^{2},\tag{5}$$

式中：X^L 表示雷达坐标系下的坐标。

激光定位方法用于估计激光雷达在一次扫描时间范围内的位姿变化量，该方法需要依靠两帧点云之间的匹配来实现。设第 k 次扫描的时间段为 $t_k\sim t_{k+1}$，扫描过程中感知到点云 P_k，在下一帧开始时，P_k 被投影到 t_{k+1} 处与第 $k+1$ 帧形成参照。在第 $k+1$ 次扫描期间，将上一帧点云 P_k 重新投影到这一帧，用 $\bar{P}_k$ 表示。

为了实现两帧点云间的匹配进而构建误差方程，根据两帧点云特征点之间的对应关系，可以列出特征点到它对应线或面的距离的表达式。E_{k+1} 和 H_{k+1} 分别表示第 $k+1$ 次扫描时点云中的边缘点和平面点，因为扫描持续时间内激光雷达也在运动，所以进一步设 $\bar{E}_{k+1}$ 和 $\bar{H}_{k+1}$ 是 E_{k+1} 和 H_{k+1} 重新投影到第 $k+1$ 次扫描起始时刻的复制点集，则 $\bar{E}_{k+1}$ 对应 $k+1$ 次扫描点云过程中提取的边缘点集合。对于 $\bar{E}_{k+1}$ 中的一个边缘点 i，假设已经在上一帧的投影点集合 $\bar{P}_k$ 中找到了与之对应的线 (j,l)，则点 i 到线 (j,l) 的距离可以表示为[23]

$$d_{\varepsilon}=\frac{\left|\left(\bar{\boldsymbol{X}}_{(k+1,i)}^{L}-\bar{\boldsymbol{X}}_{(k,j)}^{L}\right)\times\left(\bar{\boldsymbol{X}}_{(k+1,i)}^{L}-\bar{\boldsymbol{X}}_{(k,l)}^{L}\right)\right|}{\left|\bar{\boldsymbol{X}}_{(k,j)}^{L}-\bar{\boldsymbol{X}}_{(k,l)}^{L}\right|},\tag{6}$$

式中：$\bar{\boldsymbol{X}}_{(k+1,i)}^{L}$、$\bar{\boldsymbol{X}}_{(k,j)}^{L}$、$\bar{\boldsymbol{X}}_{(k,l)}^{L}$ 分别是激光雷达坐标系下点 i、j、l 的坐标。$\bar{H}_{k+1}$ 对应 $k+1$ 次扫描点云过程中提取的平面点集合。对于 $\bar{H}_{k+1}$ 中的一个平面点 e'，假设已经在 $\bar{P}_k$ 中找到了与之对应的平面片 (a,b,c)，则点 e' 到平面 (a,b,c) 的距离可以表示为[23]

$$d_{\varepsilon}=\frac{\left|\left(\bar{\boldsymbol{X}}_{(k+1,e)}^{L}-\bar{\boldsymbol{X}}_{(k,a)}^{L}\right)\cdot\left[\left(\bar{\boldsymbol{X}}_{(k,a)}^{L}-\bar{\boldsymbol{X}}_{(k,b)}^{L}\right)\times\left(\bar{\boldsymbol{X}}_{(k,a)}^{L}-\bar{\boldsymbol{X}}_{(k,c)}^{L}\right)\right]\right|}{\left|\left(\bar{\boldsymbol{X}}_{(k,a)}^{L}-\bar{\boldsymbol{X}}_{(k,b)}^{L}\right)\times\left(\bar{\boldsymbol{X}}_{(k,a)}^{L}-\bar{\boldsymbol{X}}_{(k,c)}^{L}\right)\right|}。\tag{7}$$

假设激光雷达在扫描过程中以恒定的角速度和线速度运动，设第 k 次扫描时间为 $t_k\sim t_{k+1}$，设 $\boldsymbol{T}_{k+1}^{L}$ 为第 k 次扫描过程中激光雷达的位姿变换矩阵，$\boldsymbol{T}_{k+1}^{L}=[t_x,t_y,t_z,\theta_x,\theta_y,\theta_z]^{\mathrm{T}}$，其中，$t_x$、$t_y$、$t_z$ 分别对应雷达坐标系中 x、y、z 轴正方向的平移量，θ_x、θ_y、θ_z 是遵循右手法则的旋转角。

在 $\bar{P}_{k+1}$ 中选定一点 i，设 t_i 是它的时间戳，$\boldsymbol{T}_{(k+1,i)}^{L}$ 是 $[t_{k+1},t_i]$ 之间的姿态变换矩阵[23]，因为 t_i 介于 t_{k+1} 与 t 之间，所以 $\boldsymbol{T}_{(k+1,i)}^{L}$ 可以由 $\boldsymbol{T}_{k+1}^{L}$ 通过线性插值计算得到，即

$$\boldsymbol{T}_{(k+1,i)}^{L}=\frac{t_i-t_{k+1}}{t-t_{k+1}}\boldsymbol{T}_{k+1}^{L}。\tag{8}$$

为了求解激光雷达的运动，需要在 E_{k+1} 和 $\bar{E}_{k+1}$ 或者 H_{k+1} 和 $\bar{H}_{k+1}$ 之间建立一个几何关系。设 $\boldsymbol{T}_{(k+1,i)}^{L}$ 的平移向量为 $\boldsymbol{t}_{(k+1,i)}^{L}$，利用(7)式所示的转换可以推导出

$$\boldsymbol{X}_{(k+1,i)}^{L}=\boldsymbol{T}_{(k+1,i)}^{L}\bar{\boldsymbol{X}}_{(k+1,i)}^{L}。\tag{9}$$

计算误差函数 d 关于 $\boldsymbol{T}_{k+1}^{L}$ 的雅可比矩阵，并表示为 $\boldsymbol{J}$，其中 $\boldsymbol{J}=\partial d/\partial\boldsymbol{T}_{k+1}^{L}$；然后，采用 Levenberg-Marquardt 算法进行非线性迭代求解，使 d 趋于零，即

$$\boldsymbol{T}_{k+1}^{L}\leftarrow\boldsymbol{T}_{k+1}^{L}-\left[\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}+\lambda\,\mathrm{diag}(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})\right]^{-1}\boldsymbol{J}^{\mathrm{T}}d,\tag{10}$$

式中：λ 为拉格朗日乘子。

3.3 位姿融合

视觉里程计和激光里程计各有优劣：视觉位姿估计在室内外场景中均可使用，但是对光照条件的依赖程度较高；激光位姿估计多用于室内场景，不受光照的影响，但是在非结构特征场景下往往不能实现点云的有效配准。综合两者的优势可以提高位姿估计的鲁棒性，同时使位姿估计可用于各种环境，不再受光照条件的限制。

设由图像和点云得到的当前帧相对于上一帧的位姿分别为 $\boldsymbol{T}_{\mathrm{C}}$ 和 $\boldsymbol{T}_{\mathrm{L}}$，以左目相机的相机坐标系作为机器人自身的参考坐标系，同时设激光雷达与左目相机之间的外参矩阵为 $\Delta\boldsymbol{T}_{\mathrm{LC}}$（用于将位姿估计统一到相同的参考坐标系中），则 $\boldsymbol{T}'_{\mathrm{L}}$ 在相机坐标系中可以表示为

$$\boldsymbol{T}'_{\mathrm{L}}=\Delta\boldsymbol{T}_{\mathrm{LC}}\boldsymbol{T}_{\mathrm{L}}\Delta\boldsymbol{T}_{\mathrm{LC}}^{-1}。\tag{11}$$

设位姿 $\boldsymbol{T}'_{\mathrm{L}}$ 对应的平移量和欧拉角为向量 $\boldsymbol{w}_{\mathrm{L}}=(x_{\mathrm{L}}\ y_{\mathrm{L}}\ z_{\mathrm{L}}\ p_{\mathrm{L}}\ y_{\mathrm{L}}\ r_{\mathrm{L}})$，位姿 $\boldsymbol{T}_{\mathrm{C}}$ 对应的平移量和欧拉角为向量 $\boldsymbol{w}_{\mathrm{C}}=(x_{\mathrm{C}}\ y_{\mathrm{C}}\ z_{\mathrm{C}}\ p_{\mathrm{C}}\ y_{\mathrm{C}}\ r_{\mathrm{C}})$，其中 p 代表俯仰角，y 代表航偏角，r 代表翻滚角，x、y、z 分别代表各个轴的位移分量。设本次视觉位姿估计中提取到的有效特征点数量为 N_{C}，设激光位姿估计中提取到的有效特征点数量为 N_{L}。定义融合后的位姿 $\boldsymbol{w}_{\mathrm{F}}=(p_{\mathrm{F}}\ y_{\mathrm{F}}\ r_{\mathrm{F}}\ x_{\mathrm{F}}\ y_{\mathrm{F}}\ z_{\mathrm{F}})$ 为

$$\boldsymbol{w}_{\mathrm{F}}=\frac{(N_{\mathrm{C}}\boldsymbol{w}_{\mathrm{C}}+N_{\mathrm{L}}\boldsymbol{w}_{\mathrm{L}})}{(N_{\mathrm{C}}+N_{\mathrm{L}})}。\tag{12}$$

根据两种方法分别使用到的特征点数量来动态调整权重，从而实现位姿的融合。

4 实验结果

为了验证动态物体对位姿估计的影响，以及所提融合位姿估计算法的有效性，本文使用公开的 KITTI 数据集以及动态场景下的实验数据进行对比说明。

4.1 公开数据集上的对比实验

为了进一步验证动态物体的剔除对位姿估计的影响，分别评估视觉定位、激光雷达定位及其融合定位的结果。为了对位姿精度进行量化处理，需要采集动态场景下的图像和点云数据，并且需要保证相机雷达之间的精确标定以及传感器实时的真实位姿。因此，选择自动驾驶 KITTI 的 Odometry 数据集，截取环境中存在移动车辆的部分连续图像和点云作为实验数据，截取内容为 05 和 08 序列的部分片段。KITTI 数据已经将图像和点云的时间戳对齐，从而保证了融合是可以定量评价的。在本实验中，深度神经网络只需要对车辆进行识别，以方便后续的剔除工作。数据处理用到的处理器为 Intel i7-10875H(2.1 GHz)和 RTX2060(6 GB)，运行内存为 16 GB。选取的动态实验场景如图 2 所示。

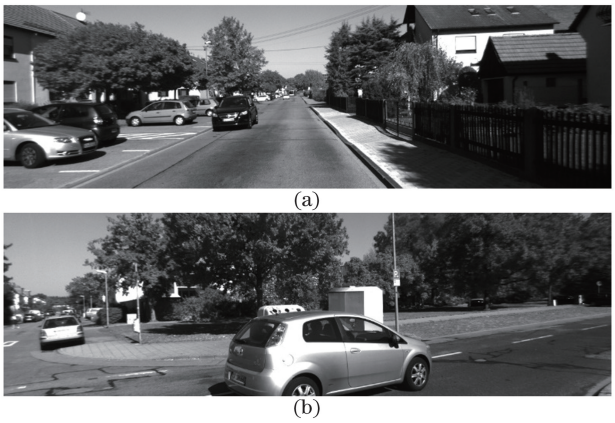


图 2 动态实验场景。(a) 05 序列实验场景；(b) 08 序列实验场景

Fig. 2 Dynamic experimental scenes. (a) 05 sequence experiment scene; (b) 08 sequence experiment scene

图 2(a)中的黑色轿车与背景的移动方向相同但速度更快，图 2(b)中的银色轿车在视野中横向穿过，08 序列场景的动态特征数量比 05 序列场景的更多，相对速度也更大。

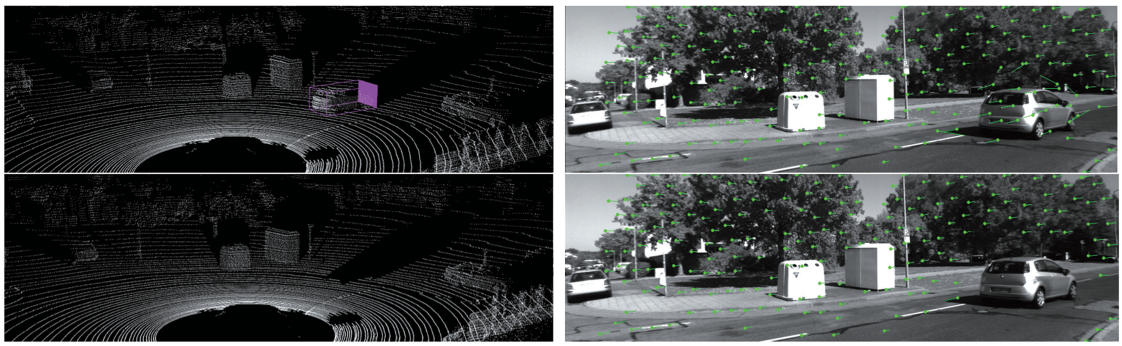
实验选取相对位姿误差(RPE)的均值和标准差作为动态场景下位姿估计的性能指标，对位姿平移、旋转的 6 个分量分别进行评价。设定间隔为 0.1 s 的相邻两帧之间的相对位姿与真实值的误差为相对位姿误差。设算法估计位姿为 $\boldsymbol{E}_i$，真实位姿为 $\boldsymbol{G}_i$，则相对位姿误差 $\boldsymbol{O}_i$ 为

$$\boldsymbol{O}_i=(\boldsymbol{E}_i^{-1}\boldsymbol{E}_{i+1})^{-1}(\boldsymbol{G}_i^{-1}\boldsymbol{G}_{i+1})。\tag{13}$$

4.1.1 动态场景算法对比

首先，根据动态目标物体候选框的范围分别剔除动态点云与图像中的动态特征点，剔除后的视觉效果如图 3 所示。其中绿色点代表提取的角点，绿色线段表示追踪到的光流轨迹。可以很明显地看出，车辆对应的光流方向与背景点有明显差异。

分别验证剔除动态物体对视觉和激光雷达位姿估计的影响。实验中选取的光流法感知范围为 21×21，选取的重投影误差阈值 $t_0=10$ pixel。本文在不同场景下分别对比了常用的非线性优化 BA 模型[29]、LOAM[23]三维激光 SLAM 算法与本文所提视觉与激光位姿估计算法的效果，统计结果如表 2 和表 3 所示。其中视觉位姿算法(Visual)和激光位姿算法(LIDAR)均为本文提出的位姿估计算法。



(a) (b)

图 3 剔除动态特征。(a)剔除动态点云;(b)剔除动态特征点

Fig. 3 Elimination of dynamic features. (a) Elimination of dynamic point clouds; (b) elimination of dynamic feature points

表 2 不同场景下对比算法的误差均值

Table 2 Mean error of comparison algorithms in different scenarios

Sequence	Algorithm	Pitch angle /(°)	Yaw angle /(°)	Roll angle /(°)	x /m	y /m	z /m
05	BA	0.0669	0.0391	**0.0363**	0.0197	0.0104	**0.0136**
	Visual	**0.0302**	**0.0273**	0.0380	**0.0126**	**0.0087**	0.0192
	LOAM	0.0376	**0.0138**	**0.0377**	0.0160	**0.0091**	**0.0093**
	LIDAR	**0.0374**	0.0148	0.0383	**0.0160**	0.0094	0.0107
08	BA	0.0451	0.0359	0.0478	0.0318	0.0196	0.0225
	Visual	**0.0106**	**0.0248**	**0.0135**	**0.0096**	**0.0031**	**0.0059**
	LOAM	0.0052	0.0097	0.0049	0.0055	0.0016	**0.0060**
	LIDAR	**0.0049**	**0.0080**	**0.0046**	**0.0052**	**0.0015**	0.0073

表 3 不同场景下对比算法的误差标准差

Table 3 Error standard deviation of comparison algorithms in different scenarios

Sequence	Algorithm	Pitch angle /(°)	Yaw angle /(°)	Roll angle /(°)	x /m	y /m	z /m
05	BA	0.0930	0.0263	0.0321	**0.0099**	0.0080	**0.0103**
	Visual	**0.0392**	**0.0216**	**0.0295**	0.0112	**0.0069**	0.0131
	LOAM	**0.0322**	**0.0075**	0.0221	**0.0043**	0.0074	**0.0054**
	LIDAR	0.0330	0.0083	**0.0215**	0.0044	**0.0074**	0.0057
08	BA	0.0431	0.0305	0.0446	0.0350	0.0166	0.0306
	Visual	**0.0129**	**0.0154**	**0.0115**	**0.0095**	**0.0014**	**0.0053**
	LOAM	**0.0046**	**0.0054**	0.0029	0.0038	0.0010	0.0044
	LIDAR	0.0047	0.0071	**0.0027**	**0.0036**	**0.0009**	**0.0041**

对比不同算法的位姿结果可以发现,剔除动态特征后,位姿的 6 个分量的误差在大多数情况下均有所降低。对各分量计算矢量和及其误差均值,本文所提视觉位姿算法在两个场景下的综合误差相比 BA 算法平均降低了 0.0300°和 0.0167 m,激光位姿算法的位移误差相比 LOAM 算法降低了 0.0010 m,但角度误差升高了 0.0016°。同时,与激光位姿估计相比,视觉位姿估计的精度提升得更加明显。由于三维激光雷达可以感知 360°的环境,而采用的数据集场景中仅有少量移动物体,移动特征占总特征的比例较小,因此受影响的程度较小。

接下来比较不同算法结果的标准差。本文所提视觉位姿算法的角度标准差和位移标准差相比 BA 算法分别降低了 0.0458°和 0.0181 m,激光位姿算法的位移标准差相比 LOAM 算法降低了 0.0002 m,但角度标准差却升高了 0.0009°。可知,本文所提视觉位姿算法极大地提高了结果的准确度。对比 05 序列和 08 序列的激光位姿算法的结果可知,环境中的动态特征越多,改进效果越明显。

4.1.2 位姿融合实验

融合两种传感器的位姿结果可以使整个系统应用在更多场景中,对环境的适应性更强。融合算法与 BA、LOAM、ORBSLAM2 算法的误差平均值对比如表 4 所示,不同场景下位移和角度的矢量和误差如图 4 所示。

表 4 不同算法的误差均值

Table 4 Mean error of different algorithms

Sequence	Algorithm	Pitch angle /(°)	Yaw angle /(°)	Roll angle /(°)	x /m	y /m	z /m
05	BA[29]	0.0669	0.0391	0.0363	0.0197	0.0104	0.0136
	ORBSLAM2[19]	**0.0307**	0.0114	0.0283	0.0099	0.0223	0.0243
	LOAM[23]	0.0376	0.0138	0.0377	0.0160	0.0091	0.0093
	Fusion	0.0332	**0.0081**	**0.0221**	**0.0046**	**0.0070**	**0.0057**
08	BA[29]	0.0451	0.0359	0.0478	0.0318	0.0196	0.0225
	ORBSLAM2[19]	0.0067	0.0177	0.0076	**0.0046**	0.0023	0.0068
	LOAM[23]	**0.0052**	0.0097	**0.0049**	0.0055	**0.0016**	0.0060
	Fusion	0.0060	**0.0086**	0.0064	0.0050	0.0018	**0.0050**

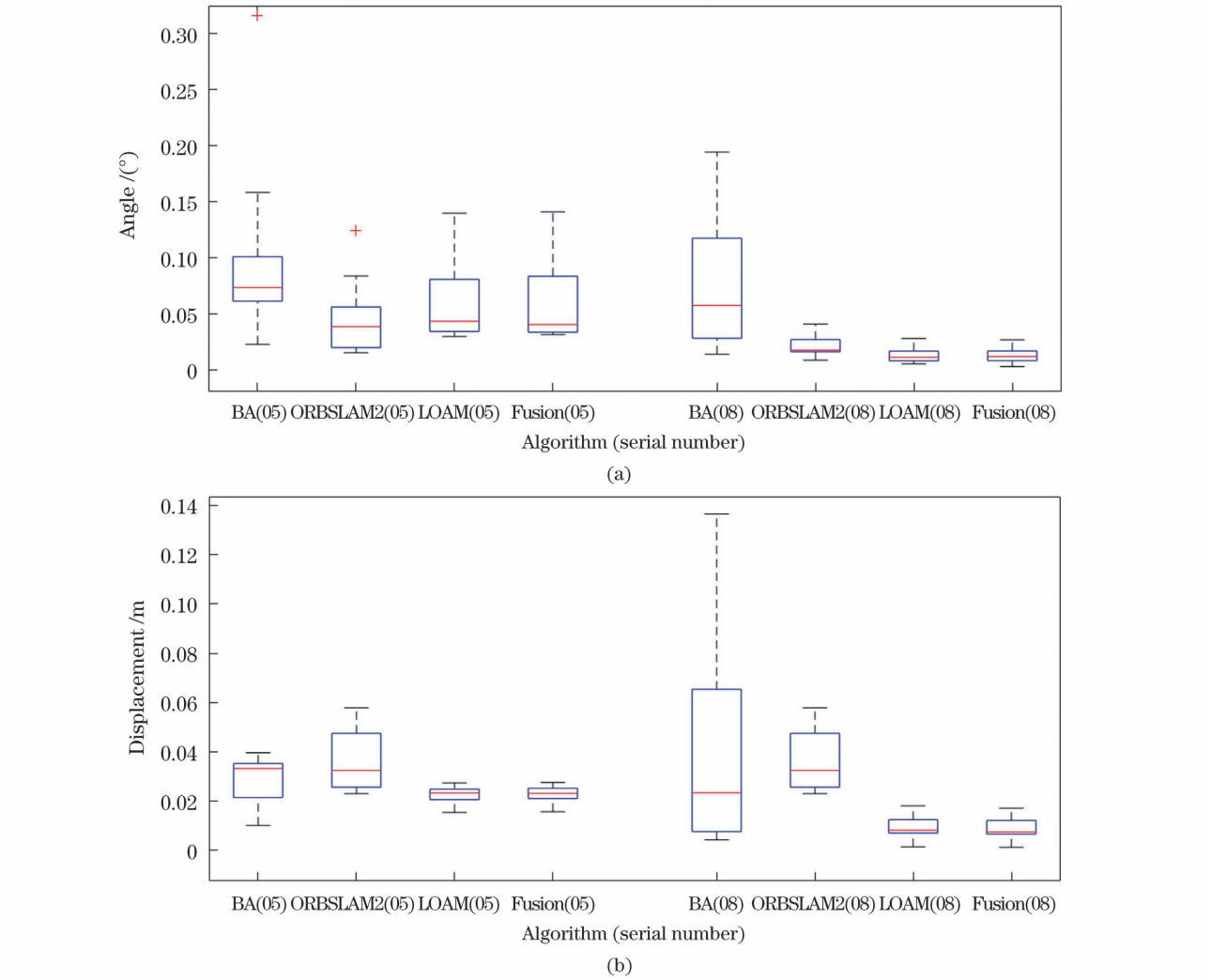


图 4 位姿误差箱形图。(a)角度误差箱形图;(b)位移误差箱形图

Fig. 4 Box plots of pose error. (a) Angular error box plot; (b) displacement error box plot

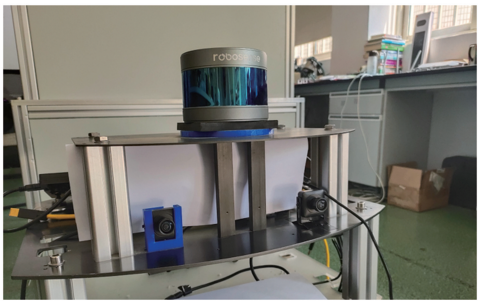
在图 4 的箱形图中,矩形内的横线表示误差中位数,矩形的上下边沿表示上下四分位数,矩形外的上下短线表示误差的最大值和最小值,离散的十字点表示异常值。结合表 3 和图 4 可以看出:本文所提融合算法在 05 序列中的位移误差与 BA 视觉算法、LOAM 算法相比平均降低了 0.0105 m 和 0.0010 m;与 ORBSLAM2 算法相比,本文所提融合算法的位移误差在 05 序列和 08 序列中分别降低了 0.0081 m 和 0.0026 m。实验结果表明,融合得到的位姿结果在保证误差稳定的同时,也使精度进一步提高。当环境不适合某个传感器进行位姿估计时,另一个传感器仍可以提供准确的结果,因此系统

更不易发生位姿丢失。

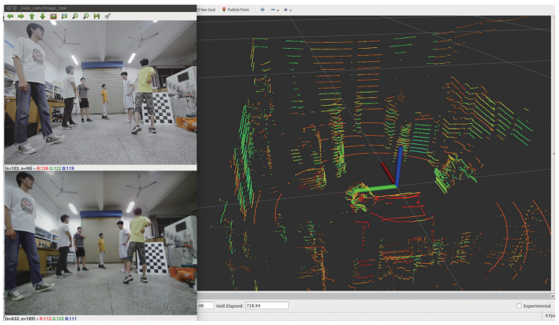
4.2　室内动态场景对比实验

公开的里程计数据集大多是在静态环境下采集的，极少有多动态目标的实验场景。鉴于此，本文搭建了由两个平行放置的相机和激光雷达组成了的实验平台，并模拟了室内动态环境，即：6 个行人在相机与激光雷达的视野范围内移动。由于无法得到这种复杂环境下机器人实际运动的准确位姿，因此实验中通过固定平台来探究动态场景对静止机器人位姿估计的影响。

实验平台和实验场景如图 5 所示，其中图 5(a)是实验平台，图 5(b)是机器人操作系统(ROS)采集数据的可视化界面。两个相机均为 640×480 的彩色免驱 USB 相机，激光雷达采用的是 RoboSense 的 16 线扫描式 3D 激光雷达，数据采集均在 Ubuntu16.04 系统环境中进行。使用 Autoware[30] 自动驾驶开源平台工具对双目相机的相关内参以及相机与激光雷达之间的外参进行标定，并在位姿估计前对图像进行双目校正。数据处理所用的处理器为 Intel i7-10875H(2.1 GHz)和 RTX2060(6 GB)，运行内存 16 GB。



(a)



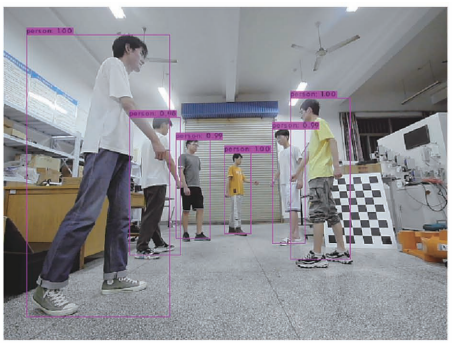
(b)

图 5　实验平台和实验场景。(a)实验平台；(b)实验场景

Fig. 5　Experimental platform and experimental scene. (a) Experimental platform; (b) experimental scene

采用深度学习网络对图像和点云进行预处理，得到目标行人的候选框。YOLOv4 和 PointRCNN 输出结果的可视化效果如图 6 所示。

(a)



(b)

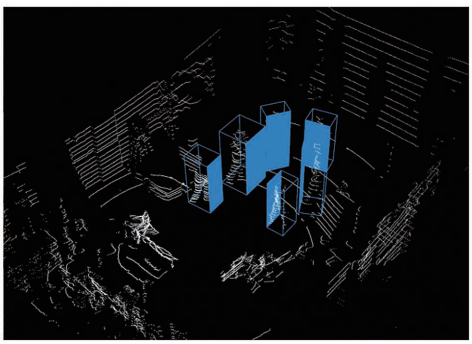


图 6　深度学习提取候选框。(a)图像候选框；(b)点云候选框

Fig. 6　Candidate frames extraction by deep learning. (a) Image candidate frame; (b) point cloud candidate frame

在室内动态场景下，对 BA、LOAM、ORBSLAM2 与本文所提视觉位姿估计、激光位姿估计以及融合位姿估计算法进行对比，对比结果如表 5 所示。

表 5　室内动态场景下相对位姿误差实验结果对比

Table 5　Comparison of experimental results of relative pose errors in indoor dynamic scenes

Algorithm	Pitch angle /(°)	Yaw angle /(°)	Roll angle /(°)	x /m	y /m	z /m
BA	0.0255	1.1481	0.0134	0.0123	0.0016	0.1018
ORBSLAM2	0.0323	0.0901	0.0416	0.0058	0.0038	0.0033
Visual	0.0084	0.3568	**0.0106**	0.0086	0.0026	0.0134
LOAM	0.0779	0.0953	0.2583	0.0071	0.0098	**0.0019**
LIDAR	0.0048	0.0198	0.1041	0.0058	0.0024	0.0052
Fusion	**0.0035**	**0.0178**	0.0926	**0.0044**	**0.0024**	0.0060

实验数据表明，剔除动态特征后的视觉与激光位姿估计结果均优于同类型算法，其中，视觉位姿估计受动态特征点的影响最为明显。融合算法得到的角度和位移的平均相对误差分别为 0.0944°和 0.0078 m，相比 LOAM 算法降低了 0.1918°和 0.0045 m。ORBSLAM2 算法的总位移误差相对较小，为 0.0076 m，角度偏差较大，为 0.1044°。ORBSLAM2 使用 ORB 特征描述子匹配角点，因此动态特征对它的影响较小。但是，这是在可见光环境下进行的测试，如果在夜晚或者光照条件不足的环境下，相机将不能捕获有效图像，会导致 ORBSLAM2 结果失真；相反，此时的融合算法仍会输出激光位姿估计结果，且与 LOAM 相比仍有更高的精确度。

4.3 运行时间对比

本文进一步统计了各算法的运行时间。以 KITTI 数据集 08 序列动态场景片段的运行时间为例，4 种算法的运行时间箱形图如图 7 所示，算法的平均运行时间如表 6 所示。LOAM 的平均运行时间最短为 0.0242 s，融合算法的运行时间最长，为 0.2974 s。融合算法的运算时间包括深度神经网络检测、位姿估计和位姿融合三部分，三个部分顺序执行，因此运行时间相对较长。ORBSLAM2 和 LOAM 是实时定位与建图算法的代表，本文提出的融合算法主要是通过离线状态下的位姿估计来辅助建立高精度三维地图，目的是提高精度而不是保证算法在线运行的实时性。

表 6 算法的平均运行时间

Table 6 Average running time of different algorithms

Algorithm	Average running time /s
BA	0.0317
ORBSLAM2	0.1462
LOAM	**0.0242**
Fusion	0.2974

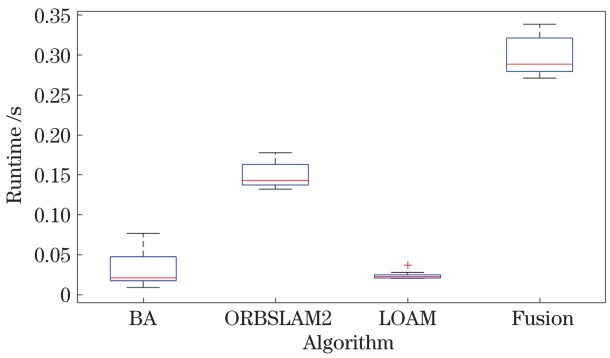


图 7 运行时间箱形图

Fig. 7 Running time box plot

5 结 论

本文提出了一种基于动态特征剔除图像与点云融合的机器人位姿估计算法。运用深度学习的方法提取图像和点云中目标物体的候选框，并将提取的候选框运用到后续的数据处理与特征优化中，彻底避免了动态特征错误匹配所引起的误差函数异常，进而消除了其对位姿估计的影响。同时，本文基于特征点数量对位姿进行了动态加权融合。最后，本文使用公开的 KITTI 数据集以及搭建的实验平台在动态场景下采集的实验数据，对比了所提算法与 BA、LOAM 和 ORBSLAM2 三个主流算法的位姿估计精度。实验结果表明，剔除动态特征后的融合位姿估计结果具有更高的精度。同时，顺序执行的处理逻辑可以保证离线状态下系统不受运行时间的影响，能正确处理每一帧数据。

参 考 文 献

[1] Liu L, Ma G Q, Gao Y, et al. Flexible measurement technology of complex curved surface three-dimensional shape robot based on iGPS[J]. Chinese Journal of Lasers, 2019, 46(3): 0304006.
刘丽, 马国庆, 高艺, 等. 基于 iGPS 的复杂曲面三维形貌机器人柔性测量技术[J]. 中国激光, 2019, 46(3): 0304006.

[2] Hu S X, Chen C P, Zhang A W. Application of SLAM in vehicle-borne mobile mapping system[J]. Chinese Journal of Lasers, 2012, 39(11): 1108012.
胡少兴, 陈春朋, 张爱武. 同步定位及地图创建算法在车载移动测绘系统中的应用[J]. 中国激光, 2012, 39(11): 1108012.

[3] Zeng S P, Zhou G B, Li W W, et al. Summary of key technologies of rescue robot for public safety[J]. Robot Technique and Application, 2019(2): 20-25.
曾世藩, 周广兵, 李文威, 等. 面向公共安全的救援机器人关键技术综述[J]. 机器人技术与应用, 2019(2): 20-25.

[4] Cao S H, Zhang C X, Wang G Z, et al. Development situation and military application of autonomous underwater vehicle[J]. Ship Engineering, 2019, 41(2): 79-84, 89.
曹少华, 张春晓, 王广洲, 等. 智能水下机器人的发展现状及在军事上的应用[J]. 船舶工程, 2019, 41(2): 79-84, 89.

[5] Taniguchi A, Isobe S, El Hafi L, et al. Autonomous planning based on spatial concepts to tidy up home environments with service robots[J]. Advanced Robotics, 2021, 35(8): 471-489.

[6] Jung J, Oh T, Myung H. Magnetic field constraints and sequence-based matching for indoor pose graph SLAM[J]. Robotics and Autonomous Systems, 2015, 70: 92-105.

[7] Carlone L, Dellaert F. Duality-based verification techniques for 2D SLAM[C]// 2015 IEEE International Conference on Robotics and Automation (ICRA), May 26-30, 2015, Seattle, WA, USA. New York: IEEE Press, 2015: 4589-4596.

[8] Shen K, Wang M L, Fu M Y, et al. Observability analysis and adaptive information fusion for integrated navigation of unmanned ground vehicles[J]. IEEE Transactions on Industrial Electronics, 2020, 67(9): 7659-7668.

[9] Kohlbrecher S, von Stryk O, Meyer J, et al. A flexible and scalable SLAM system with full 3D motion estimation[C]// 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, November 1-5, 2011, Kyoto, Japan. New York: IEEE Press, 2011: 155-160.

[10] Menna M, Gianni M, Ferri F, et al. Real-time autonomous 3D navigation for tracked vehicles in rescue environments[C]// 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 14-18, 2014, Chicago, IL, USA. New York: IEEE Press, 2014: 696-702.

[11] Li M G, Zhu H, You S Z, et al. Efficient laser-based 3D SLAM for coal mine rescue robots[J]. IEEE Access, 2019, 7: 14124-14138.

[12] Shan T X, Englot B. LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 1-5, 2018, Madrid, Spain. New York: IEEE Press, 2018: 4758-4765.

[13] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6): 1052-1067.

[14] Huang G P, Mourikis A I, Roumeliotis S I. Analysis and improvement of the consistency of extended Kalman filter based SLAM[C]// 2008 IEEE International Conference on Robotics and Automation, May 19-23, 2008, Pasadena, CA, USA. New York: IEEE Press, 2008: 473-479.

[15] Zhang Z, Nejat G. Intelligent sensing systems for rescue robots: landmark identification and three-dimensional mapping of unknown cluttered urban search and rescue environments[J]. Advanced Robotics, 2009, 23(9): 1179-1198.

[16] Zou B, Lin S Y, Yin Z S. Semantic mapping based on YOLOv3 and visual SLAM[J]. Laser & Optoelectronics Progress, 2020, 57(20): 201012.
邹斌, 林思阳, 尹智帅. 基于 YOLOv3 和视觉 SLAM 的语义地图构建[J]. 激光与光电子学进展, 2020, 57(20): 201012.

[17] Li X P, Ao H X, Belaroussi R, et al. Fast semi-dense 3D semantic mapping with monocular visual SLAM[C]//2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), October 16-19, 2017, Yokohama, Japan. New York: IEEE Press, 2017: 385-390.

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

[19] 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.

[20] Forster C, Pizzoli M, Scaramuzza D. SVO: fast semi-direct monocular visual odometry[C]// 2014 IEEE International Conference on Robotics and Automation (ICRA), May 31-June 7, 2014, Hong Kong, China. New York: IEEE Press, 2014: 15-22.

[21] Ma Z G, Zhao Y G, Liu C Y, et al. Survey of SLAM with laser-camera fusion sensor[J]. Computer Measurement & Control, 2019, 27(3): 1-6.
马争光, 赵永国, 刘成业, 等. 激光和视觉融合 SLAM 方法研究综述[J]. 计算机测量与控制, 2019, 27(3): 1-6.

[22] Chen X, Zhang H, Lu H M, et al. Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue[C]// 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), October 11-13, 2017, Shanghai, China. New York: IEEE Press, 2017: 41-47.

[23] Zhang J, Singh S. Low-drift and real-time lidar odometry and mapping[J]. Autonomous Robots, 2017, 41(2): 401-416.

[24] Kubelka V, Reinstein M, Svoboda T. Tracked robot odometry for obstacle traversal in sensory deprived environment[J]. IEEE/ASME Transactions on Mechatronics, 2019, 24(6): 2745-2755.

[25] Wang H L, Zhang C J, Song Y, et al. Three-dimensional reconstruction based on visual SLAM of mobile robot in search and rescue disaster scenarios[J]. Robotica, 2020, 38(2): 350-373.

[26] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. (2020-04-23)[2021-04-28]. https://arxiv.org/

abs/2004.10934.

[27] Shi S S, Wang X G, Li H S. PointRCNN: 3D object proposal generation and detection from point cloud [C] // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 770-779.

[28] Lepetit V, Moreno-Noguer F, Fua P. EPnP: an accurate O(n) solution to the PnP problem [J]. International Journal of Computer Vision, 2008, 81(2): 155-166.

[29] Triggs B, McLauchlan P F, Hartley R I, et al. Bundle adjustment: a modern synthesis [M] // Triggs B, Zisserman A, Szeliski R. Vision algorithms: theory and practice. Lecture notes in computer science. Heidelberg: Springer, 2000, 1883: 298-372.

[30] Kato S, Takeuchi E, Ishiguro Y, et al. An open approach to autonomous vehicles [J]. IEEE Micro, 2015, 35(6): 60-68.

Robot Pose Estimation Method Based on Image and Point Cloud Fusion with Dynamic Feature Elimination

Zhang Lei[1,2], Xu Xiaobin[1,2,3,4*], Cao Chenfei[1,2], He Jia[1,2], Ran Yngying[1,2], Tan Zhiying[1,2], Luo Minzhou[1,2]

[1] *College of Mechanical & Electrical Engineering, Hohai University, Changzhou, Jiangsu* 213022, *China*;

[2] *Jiangsu Key Laboratory of Special Robot Technology, Hohai University, Changzhou, Jiangsu* 213022, *China*;

[3] *College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu* 210016, *China*;

[4] *Changzhou Changgong Electronic Technology Co., Ltd., Changzhou, Jiangsu* 213001, *China*

Abstract

Objective Robot positioning is an important component of both robot navigation and SLAM technology. During robot positioning, LIDAR and cameras are often used to collect environmental data. Through calculations, structural or texture features in the environment are obtained and, as a result, the robot's pose is indirectly determined. However, dynamic objects in the actual environment will have a great impact on the accuracy of pose estimation. The position of dynamic features in the global coordinate system changes, affecting the robot's relative pose estimation. GPS-IMU combined navigation is another popular positioning method. When the robot locates in the field, tunnel, underground, or other environments, GPS signals are often blocked, resulting in signal loss. However, the positioning of a single LIDAR or vision sensor is often limited by a specific use environment. For example, the camera cannot be used in low-light conditions. Therefore, the multisensor fusion positioning method has a greater application value. This paper proposes a pose estimation algorithm based on deep learning and adaptive fusion of pose for LIDAR and camera pose estimation in a dynamic environment.

Methods This article suggests a pose estimation algorithm for LIDAR and stereo camera pose estimation in a dynamic environment that is based on deep learning and adaptive pose fusion. YOLOv4 is used to extract candidate frames of potential moving objects in the image. Then, the optical flow method is used to track the corner points of the front and rear frames and eliminate dynamic features based on the candidate frame. The reprojection error function is developed from triangulated map and feature points. Nonlinear optimization using RANSAC is used to find the best pose. PointRCNN is used in LIDAR pose estimation to extract candidate frames from a point cloud of potential moving objects. Meanwhile, the linear and planar feature points in the point cloud are extracted and screened according to the candidate frame. The point-to-line and point-to-surface distances are used to construct an error function that calculates the poses of the preceding and following frames. Finally, the pose estimation results of the two are dynamically weighted and fused based on the number of feature points of the image and point cloud.

Results and Discussions The public KITTI data set and the experimental data collected by the experimental platform we built in dynamic scenarios are compared to validate the effect of dynamic objects on pose estimation and the effectiveness of the fusion pose estimation algorithm proposed in this paper. First, while comparing, it is

discovered that after excluding the dynamic features, the errors of the six components of the pose are reduced in most cases. The comprehensive error of the visual pose algorithm in this paper in the two scenes is reduced by 0.0300° and 0.0167 m on average. The displacement error of the LIDAR pose algorithm is reduced by 0.0010 m; however, the angle error is increased by 0.0016°. Simultaneously, the accuracy of visual pose estimation is more obvious than LIDAR pose estimation (Table 1). The fusion result is compared with the average error of BA, LOAM, and ORBSLAM2. The results of the fusion algorithm used in this paper produce fewer errors in the 05 sequence than the BA vision algorithm and LOAM. Compared with BA and LOAM algorithms, our fusion algorithm's displacement error is reduced by 0.0105 m and 0.0010 m, respectively. The displacement errors of our algorithm in scenes 05 and 08 are reduced by 0.0081 m and 0.0026 m compared with the results of ORBSLAM2 algorithm (Table 3). Second, this paper constructs an experimental platform comprising two parallel stereo cameras and LIDAR that simulates the indoor dynamic environment. Six pedestrians move within the field of view of the camera and LIDAR. Experimental results show that after removing dynamic features, the vision and LIDAR pose estimation results outperform the same type of algorithms. The average relative error of the angle and displacement of the fusion result is 0.0944° and 0.0078 m, which is 0.1918° and 0.0045 m greater than the accuracy of the LOAM algorithm. The accuracy of the algorithm is increased by 0.0100° compared to ORBSLAM2 (Table 4).

Conclusions This paper presents a robot pose estimation algorithm that is based on the fusion of dynamic feature elimination images and point clouds. The method of deep learning is used to extract the candidate frame of the target object from an image and point cloud, which is then used for data processing and feature optimization. It completely avoids the error function abnormality caused by incorrect matching of dynamic features and eliminates its effect on the pose estimation. Simultaneously, this paper performs a dynamic weighted fusion of the pose based on the number of feature points. Finally, this paper uses the public KITTI data set and the experimental data collected by the experimental platform construct-in dynamic scenarios to compare the pose estimation accuracy of the three mainstream algorithms of BA, LOAM, and ORBSLAM2. Experiments show that removing dynamic features improves the accuracy of pose estimation to varying degrees. The posture result after fusion is more stable. Furthermore, the sequential processing logic ensures that the system is unaffected by the running time in the offline state to correctly process each frame of data.

Key words machine vision; robot positioning; adaptive fusion; LIDAR; image; deep learning