



基于视觉的机器人自主定位与障碍物检测方法

丁斗建¹, 赵晓林^{1*}, 王长根², 高关根³, 寇磊³

(1.空军工程大学装备管理与无人机学院, 陕西西安, 710038; 2.中国人民解放军 94639 部队, 江苏南京, 210000;

3.西安飞行自动控制研究所, 陕西西安, 710056)

(通信作者电子邮箱 85327505@qq.com)

摘要: 针对稀疏型同时定位与地图构建 (Simultaneous localization and mapping, SLAM) 算法环境信息丢失导致无法检测障碍物问题, 提出一种基于视觉的机器人自主定位与障碍物检测方法。首先, 利用双目相机得到观测场景的视差图。然后在机器人操作系统 (Robot operating system, ROS) 架构下, 同时运行定位与建图和障碍物检测两个节点。定位与建图节点基于 ORB-SLAM2 完成位姿估计与环境建图。障碍物检测节点在视差图的基础上, 引入深度阈值, 将视差图二值化; 运用轮廓提取算法得到障碍物轮廓信息并计算障碍物凸包面积; 再引入面积阈值, 剔除误检测区域, 从而实时准确的解算出障碍物坐标。最后, 将检测到的障碍物信息插入到环境的稀疏特征地图当中。实验结果表明, 该方法能够在实现机器人自主定位的同时, 快速检测出环境中的障碍物, 检测精度能够保证机器人顺利避障。

关键词: 视觉定位; 障碍物检测; 视觉 SLAM; 机器人操作系统; 立体视觉; 机器人

中图分类号: TP242.6

文献标志码: A

Autonomous positioning and obstacle detection method of robot based on vision

DING Doujian¹, ZHAO Xiaolin^{1*}, WANG Changgen², GAO Guangen³, KOU Lei³

(1.Equipment Management and UAV Engineering College, Air Force Engineering University Xi'an Shanxi 710038, China;

2. Chinese People's Liberation Army 94639 troop Nanjing Jiangsu 210000, China;

3. Xi'an Flight Automatic Control Research Institute, Xi'an Shanxi 710065, China)

Abstract: Aiming at the obstacle detection problem caused by the loss of environmental information in the sparse simultaneous localization and mapping (SLAM) algorithm, an Autonomous positioning and obstacle detection method of Robot based on vision is proposed. Firstly, the parallax map of the observed scene is obtained by binocular camera. Then, under the framework of Robot operating system (ROS), two nodes, namely, localization and obstacle detection, are operated simultaneously. Localization and mapping node based on ORB-SLAM2 complete pose estimation and building environment map. Based on the parallax map, the obstacle detection node introduces the depth threshold to binarize the parallax graph. The contour extraction algorithm is used to obtain the contour information of the obstacle and calculate the convex hull area of the obstacle. Then the area threshold is introduced to eliminate the false detection area, so as to solve the coordinates of obstacles in real time and accurately. Finally, the detected obstacle information is inserted into the sparse feature map of the environment. Experiment results show that this method can quickly detect obstacles in the environment while realizing autonomous localization of the robot, and the detection accuracy can ensure the robot to avoid obstacles smoothly.

Keywords: visual location; obstacle detection; visual simultaneous localization and mapping (VSLAM); robot operating system; stereo vision; robot

收稿日期: 2018-10-31; 修回日期: 2019-01-16; 刊用日期: 2019-01-16。

基金项目: 国家自然科学基金项目 (61503405), 航空科学基金 (20160896007)

作者简介: 丁斗建 (1995-), 男, 江西南昌人, 硕士生, 研究领域为视觉导航、视觉 SLAM,

赵晓林 (1982-), 男, 山东东明人, 博士, 副教授, 研究领域为无人机协同控制, 无人机视觉导航技术。



0.引言

随着机器人应用领域的不断拓展,对其自主性的要求也在不断提升^[1]。要实现机器人自主移动,自主定位技术和自主避障技术是最为关键的两个方面。在机器人自主定位与自主避障研究中,视觉传感器由于具有功耗低、价格便宜、可以获得更加丰富的环境信息等优点^[2],得到了广泛应用^[3-4]。

基于视觉的机器人自主定位研究已有近二十年的历史。Nister 在 2004 年的论文[5]中提出视觉里程计以后,基于视觉里程计的同时定位与地图构建(VSLAM)算法受到广泛的关注。VSLAM 根据所利用的图像信息可分为基于特征的稀疏方法和直接的稠密方法。稠密方法可以建立密集的环境地图,便于后续的自主导航,具有代表性的工作是 LSD-SLAM^[6]。但是,稠密方法计算量很大,难以做到较好的实时性,因此,出于减少计算量提高系统实时性的目的,稀疏型 SLAM 受到了许多研究者的青睐。论文[7]使用 SIFT 特征检测方法,解决了特征匹配中图像旋转、缩放以及光照变化的影响,但 SIFT 特征的提取本身就比较耗时。论文[8]在特征匹配过程中应用了极线约束,提高了匹配的速度与精度。论文[9]提供了一个大场景 VSLAM 框架,采用关键帧优化方法,成为了 VSLAM 的一个特色,但由于缺少回环检测,定位的精度并不高。针对这一问题,论文[10]提出一种基于 ORB 特征的 SLAM 解决方案,通过加入回环检测线程和自动初始化功能,全面提高了系统的性能。但是,以上提到的方法都是通过特征点来建图,得到的只是稀疏的点图,只反映了环境的局部特征信息。因此,只依赖稀疏型 SLAM 算法,能够实现较好的定位,却无法辨别出环境中的障碍物,难以直接用于后续的避障与路径规划等工作中。

为解决上述问题,论文[11]提出一种基于全局观测地图模型的 SLAM 方案,将全局密集信息嵌入到稀疏特征地图中,并将 EKF-SLAM 算法与全局地图观测模型相结合,得到了较为可信的密集地图。论文[12]基于双目立体视觉,提出一种栅格地图构建方法,得到了包含环境几何信息的地图。论文[13]利用迭代滤波以及高斯分布对数据进行稠密化处理,实现了对稀疏性 VSLAM 地图的数据补插。以上方法基本都是致力于对稀疏特征地图进行数据补充,在得到密集地图的同时,也会给系统引入较大的计算量。然而,机器人实现一般的自主移动,并不都需要所有的环境信息,很多情况下只需检测出环境中的障碍物即可满足自主移动要求。

针对上述情况,本文利用立体视觉法检测障碍物来弥补稀疏型 SLAM 环境交互能力弱的缺陷,提出一种基于视觉的

机器人自主定位与障碍物检测方法。首先利用双目相机得到场景视差图,在视差图的基础上,同时运行两个节点。定位与建图节点依赖成熟的视觉 SLAM 算法 ORB-SLAM2^[14]实现相应功能。障碍物检测节点首先对视差图进行三维重建与伪彩色处理,得到特征点的三维信息并将疑似障碍物区域红色显示;然后依据机器人运动特性将视差图二值化,在二值图图像基础上提取障碍物轮廓与位置信息并将障碍物信息插入环境地图。通过实验验证,该方法能够在机器人 SLAM 的同时,快速检测出环境中的障碍物,且检测效果较好。

1.算法设计

1.1 算法整体架构

算法主要由三个模块组成,如图 1 所示,包括视差图获取模块、障碍物检测模块和定位与建图模块。

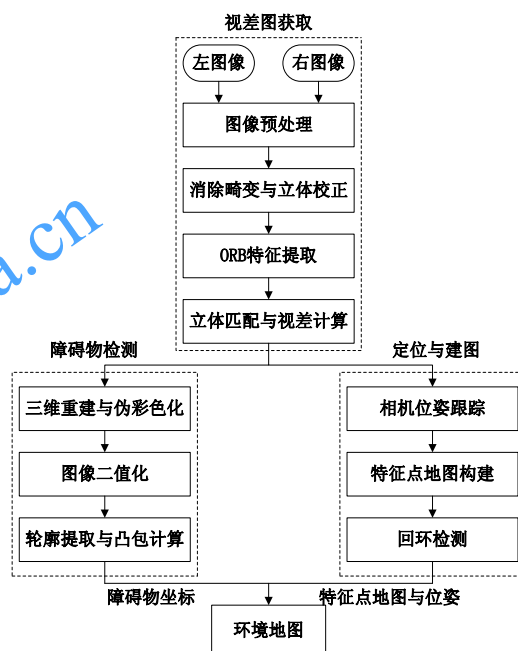


图 1 算法整体架构

Fig. 1 overall architecture of the algorithm

视差图获取模块首先由双目相机采集场景图像,并转换为灰度图像;基于相机标定模型,消除图片畸变并对图片进行立体校正;然后利用 ORB 特征提取算法提取左右图像的特征点信息;最后采用半全局立体匹配算法(SGBM)^[15]计算视差,得到较为精确的致密视差图。得到视差图之后,定位与建图模块依据 ORB-SLAM2 算法,实现相机位姿的跟踪并建立环境的稀疏特征地图。与此同时,障碍物检测模块依据立体视觉法检测出环境中障碍物,并将障碍物信息插入到定位与建图模块所建立的环境地图当中。

算法最终得到一幅具有障碍物信息的环境地图,这一地图包含了障碍物坐标信息、特征点三维信息和相机位姿,在 ORB-SLAM2 基础上实现了环境信息的补充,为机器人与环



境的交互提供依据，同时为后续的路径规划打下基础。

1.2 基于 ORB-SLAM2 的定位与建图算法

定位与建图算法基于 ORB-SLAM2 实现,ORB-SLAM2 功能非常全面，并且具有良好的精度和实时性，是当前稀疏型 SLAM 的代表。因此，本文选择 ORB-SLAM2 的双目部分完成算法的定位与建图模块，其整体的结构框图如图 2 所示，主要包含了位姿跟踪、局部构图和回环检测三个线程。

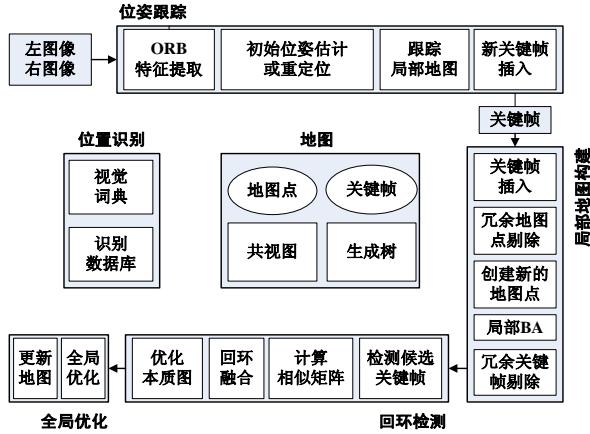


图 2 ORB-SLAM2 系统结构框图

Fig. 2 ORB-SLAM2 system structure diagram

位姿跟踪线程通过寻找当前帧与地图点的匹配，利用光束平差法 (Bundle adjustment, BA) 来最小化重投影误差，从而实现相机位姿的实时估计与优化。其基本原理如式 (1) 所示：

$$\{\mathbf{R}, \mathbf{t}\} = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_{i \in \mathcal{I}} \rho \left\| \mathbf{u}_i - \frac{\mathbf{K}}{s_i} (\mathbf{R} \mathbf{P}_i + \mathbf{t}) \right\|_2^2 \quad (1)$$

其中 $\{\mathbf{R}, \mathbf{t}\}$ 为相机的位姿， ρ 为鲁棒核函数， \mathbf{K} 为相机内参， s_i 为尺度因子， \mathbf{u}_i 为当前帧关键点， \mathbf{P}_i 为与之匹配的地图点。

局部构图线程依据相应准则管理地图点和关键帧并建立局部地图。利用局部 BA 实现优化，如式 (2) 所示：

$$\begin{aligned} & \{\mathbf{P}_i, \mathbf{R}_j, \mathbf{t}_j \mid i \in P_L, j \in K_L\} \\ & = \arg \min_{\mathbf{P}_i, \mathbf{R}_j, \mathbf{t}_j} \sum_{i \in \mathcal{I}} \sum_{k \in K_L \cup K_F} \rho \left\| \mathbf{u}_i - \frac{\mathbf{K}}{s_i} (\mathbf{R}_k \mathbf{P}_i + \mathbf{t}_k) \right\|_2^2 \end{aligned} \quad (2)$$

与位姿跟踪不同，此时关键帧位姿与地图点坐标都为优化变量。 P_L 为关键帧中能观测到的地图点， K_L 为共视关键帧， K_F 为其他关键帧。

回环检测线程通过基于二进制词袋 (Bags of binary words, DBow) [16] 方法实现的位置识别模块来更正累积漂移误差并在新建的线程中执行全局 BA 优化。

1.3 基于立体视觉的障碍物检测算法

双目立体视觉是通过两个不同位置的摄像机采集场景图像，根据两幅不同角度的场景图中的对应点来计算视差。

本文在视差图的基础上，设计了障碍物检测算法。

首先，在视差图基础上，利用三角测量原理将图像转换成三维点云。给定视差 d 、重投影矩阵 \mathbf{Q} 和点坐标 (x, y) ，根据最简单的相似三角形关系，就可以得到对应的三维坐标 $(X/W, Y/W, Z/W)$ ，其转换公式为：

$$\begin{bmatrix} X & Y & Z & W \end{bmatrix}^T = \mathbf{Q} \begin{bmatrix} x & y & d & 1 \end{bmatrix}^T \quad (3)$$

然后按照视差值的不同将视界离散化为一个个平面，从而对图像进行伪彩色处理，如式 (4) 所示：

$$\begin{aligned} R(x, y) &= \begin{cases} 0 & h(x, y) = 0 \\ 255 - h(x, y) & h(x, y) > 0 \end{cases} \\ G(x, y) &= \begin{cases} 0 & h(x, y) = 0 \\ 2 \times h(x, y) & 0 < h(x, y) < 128 \\ 2 \times [255 - h(x, y)] & h(x, y) > 128 \end{cases} \\ B(x, y) &= \begin{cases} 0 & h(x, y) = 0 \\ 255 - h(x, y) & h(x, y) > 0 \end{cases} \end{aligned} \quad (4)$$

其中 $h(x, y)$ 代表经过整形缩放后各个像素点灰度值，范围为 $(0, 255)$ ； $R(x, y)$ 、 $G(x, y)$ 、 $B(x, y)$ 表示各个像素点的 RGB 值。根据机器人运动特性，引入深度阈值 T ，将图像二值化，如式 (5) 所示：

$$g(x, y) = \begin{cases} 0 & f(x, y) < T \\ 1 & f(x, y) \geq T \end{cases} \quad (5)$$

其中， $f(x, y)$ 为图像中各个像素点的深度值， $g(x, y)$ 为二值化后的图像。最后，利用轮廓提取算法 [17] 得到障碍物轮廓信息，并计算障碍物凸包面积 S_i ，引入面积阈值 H ，如果面积小于阈值 H ，则认为是噪声不予以考虑，如式 (6) 所示：

$$P_i = \begin{cases} 0 & S_i < H \\ 1 & S_i \geq H \end{cases} \quad (6)$$

P_i 表示是否输出凸包 i 。算法最终输出的是障碍物中心坐标与轮廓。

2. 基于 ROS 的软件架构设计

本文算法基于 ROS 架构实现，系统的整体流程图如图 3 所示。系统在开始阶段，首先从外部文件中读取摄像头参数并将参数初始化。然后由图像采集节点不断的发布相机采集的场景图像，得到图像信息后，ORB-SLAM2 节点与障碍物检测节点同时运行并订阅图像信息。ORB-SLAM2 节点逐渐完成系统的初始化并针对输入的图像不断执行位姿跟踪、局部构图和回环检测三个线程，从而实现相机位姿的跟踪和环境特征地图构建。障碍物检测节点首先对摄像头参数进行检查，在获得正确的参数的情况下完成图像的立体匹配得到视差图，然后在视差图的基础上依次执行算法的各个步骤，从



而实现环境中障碍物的检测并将障碍物信息发布出去，这一信息将会被 ORB-SLAM2 节点订阅并将其插入到环境地图中。当两个节点同时判断系统关闭时关闭整个系统。

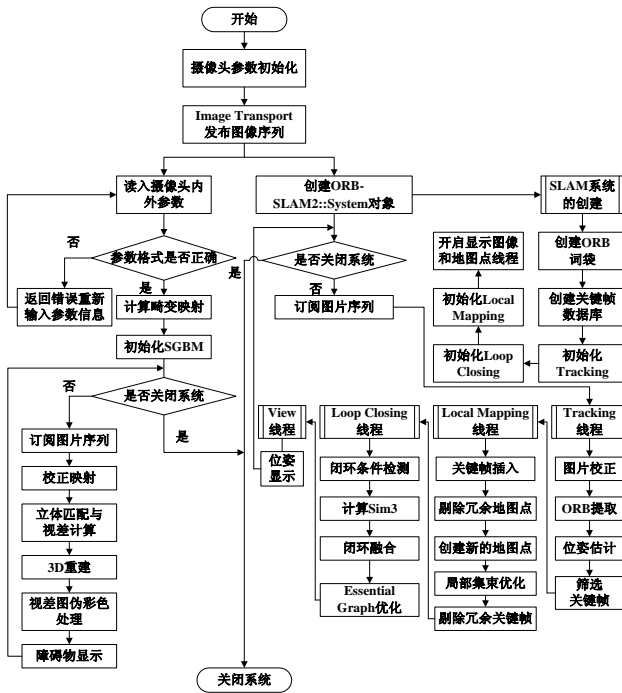


图 3 系统整体流程图

Fig. 3 overall flow chart of the system

系统的节点总体设计如图 4 所示。虚线框外的方框代表的是外部节点，分别是机器人控制节点和传感器之间的坐标转换节点。虚线框内的是本文运行的节点，图像采集节点主要完成图像的采集和分发；障碍物检测节点订阅来自图像采集节点的图像信息并实现障碍物检测和发布障碍物的信息；ORB_SLAM2 节点订阅来自图像采集节点的图像信息处理后，实时发布相机的位姿信息，并订阅来自障碍物检测节点的障碍物信息插入环境地图。

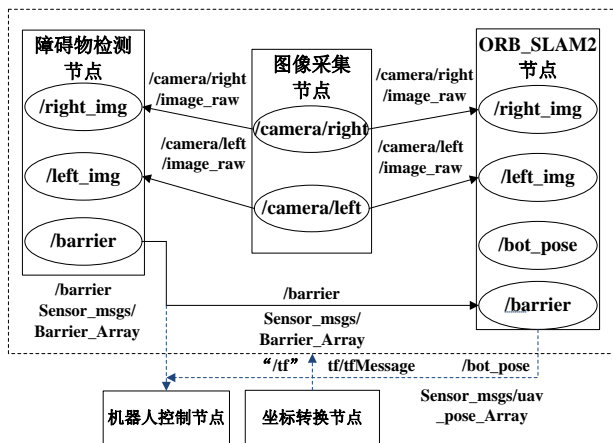


图 4 系统运行节点图

Fig. 4 node diagram of the system running

3. 实验验证与分析

3.1 数据集测试

针对系统的定位性能，本文利用著名的双目数据集 KITTI^[18]数据集对系统进行测试，实验设备为一台配备 Intel core i7 处理器的笔记本电脑，并装有 64 位的 Ubuntu16.04 操作系统。得到的各数据子集的定位均方根误差对比如表 1 所示：

表 1 均方根误差对比（单位：米）

Tab. 1 comparison of root-mean-square errors (unit: meters)

数据子集	LSD-SLAM	ORB-SLAM	本文算法
03	1.21	0.62	0.62
04	0.23	0.23	0.23
05	1.51	0.81	0.82
06	1.32	0.81	0.82
07	0.50	0.51	0.51
08	3.92	3.61	3.62

从表中数据可以看出，在大多数数据子集的测试中，本文算法的定位精度优于稠密方法 LSD-SLAM，且和 ORB-SLAM2 的定位精度基本一致，这是由于本文的定位与建图算法是基于 ORB-SLAM2 的双目模块实现的，因此能够保证较高的定位精度。

3.2 室内实物实验

3.2.1 实验硬件选型

实验硬件系统由两个同一型号的 USB 摄像机和图像处理终端共同组成，并将其安装在轮式机器人 Arduino 4WD 上进行实验。为了实现左右图像序列的获取，本文选择两个内参数相近的 JD-202USB 摄像机，如图 5(a)所示，其感光元件为 CMOS，最大分辨率为 1280×720，帧数率为 30 帧每秒。为了提高图像处理能力，本文选择了香蕉派 BPI-M3 作为图像处理终端，如图 5(b)所示。香蕉派 BPI-M3 是一个拥有 8 核 1.8GHz 处理器和 2GB LPDDR3 内存的超级单板电脑，它可以运行 Android, Ubuntu 等操作系统，具有计算处理速度快、外设接口丰富、体积小小巧轻便易挂载和开源的社区等特点，满足本文的算法设计需求。



(a)双目相机

(b)BPI-M3

图 5 实验硬件系统

Fig. 5 experimental hardware system

3.2.2 实验方案设计

实验数据来源于实验室运动场景，运动场景的平面图以及设计的运动轨迹如图 6 所示。机器人在室内按预定轨迹运



动一周,运动的途中,在场景图标标注的三个障碍物点设定三种不同的障碍物环境,以验证算法在不同环境下的障碍物检测效果。障碍物1为简单环境,只设置一个纹理清晰的静态障碍物;障碍物2为复杂环境,是在简单环境的基础上添加了多个不同的静态障碍物;障碍物3为动态环境,是指将静态障碍物换成运动的履带小车。

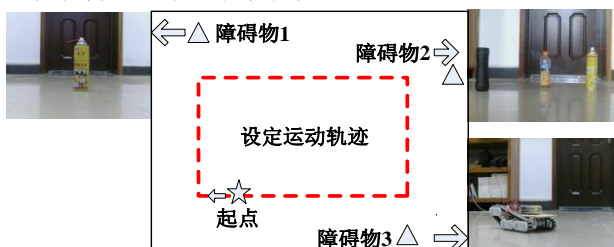


图6 运动场景平面图

Fig. 6 plan of moving scene

简单环境和复杂环境实验过程中,我们分别在机器人距离障碍物0.3m、0.8m、1.5m、2m、2.5m、3m的地方,对相应场景进行障碍物检测,分别记录系统的检测距离,单帧处理时间和误检率。动态障碍物环境下,履带车在机器人前以不同的速度进行运动,以此来检验系统对于运动目标的检测能力。深度阈值取对应的场景距离分别为1.8m、2.5m、3.2m的三个值,在不同阈值下进行多次实验;面积阈值取为100个像素点,将凸包面积小于100个像素点的区域视为误检测区域,予以剔除。

3.2.3 实验结果及分析

自主定位方面,最终生成的关键帧轨迹与环境特征点地图如图7所示:

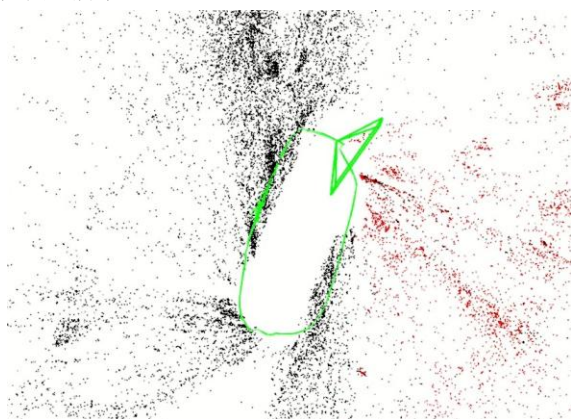


图7 关键帧轨迹与环境特征点地图

Fig. 7 key frame track and environment feature point map

对于实物实验的自主定位性能,本文只针对实验结果作简要分析。精度方面,形成闭环路径前,系统生成的路径会出现局部的偏移,闭环路径形成后,系统开始进行全局优化,对轨迹图和位姿进行调整。将生成的路径与设计的路径比较,可以发现整体生成图与设计的路线拟合的比较好,没有出现较大的偏差。实时性方面,系统启动以后,每秒处理帧数在

14~18之间,可以满足机器人一般运动状态下的定位。稳定性方面,实验过程中,有时会出现跟踪丢失的情况,但是由于采用了关键帧存取,实现了系统快速的重定位能力,可以保证稳定的定位输出。

障碍物检测方面,三种环境下不同时刻的障碍物检测结果如图8-10所示,图中从左到右分别为机器人在距离障碍物2.0m、1.5m和0.8m处的实验结果。在每幅子图中,从上到下依次为左图像,右图像,伪彩色图像和障碍物提取图像。简单环境和复杂环境的测试距离分别由远到近进行,动态障碍物的运动分为沿相机坐标系的X轴和Z轴方向运动。

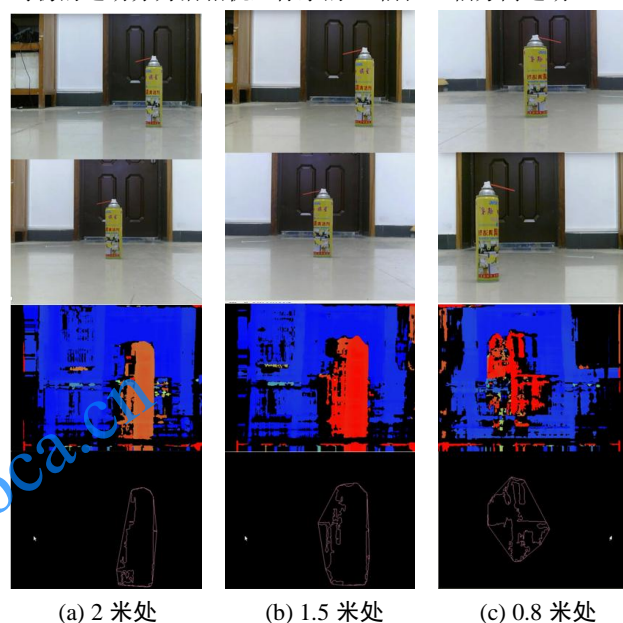


图8 简单障碍物环境测试结果

Fig. 8 test results of simple obstacle environment

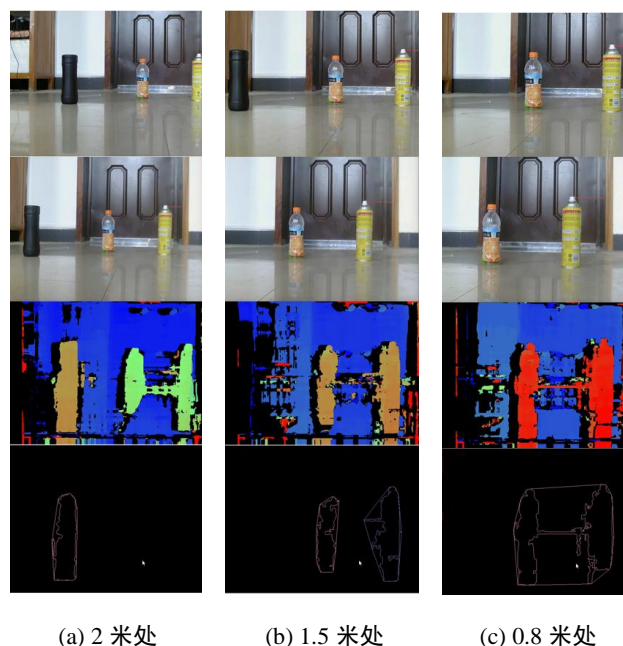


图9 复杂障碍物环境测试结果

Fig. 9 test results of complex obstacle environment

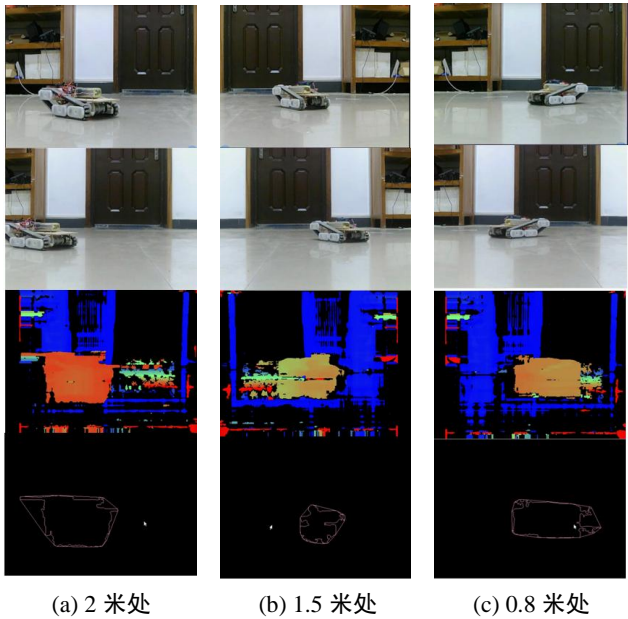


图 10 动态障碍物环境测试结果

Fig. 10 test results of dynamic obstacle environment

分别用 A、B、C 表示简单障碍物环境、复杂障碍物环境和动态障碍物环境下的测试结果，实验结果主要从三个方面分析：

(1) 障碍物检测精度

障碍物检测精度主要从误检率与检测距离误差两个方面进行评价，通过记录不同环境下的误检测和对比算法输出的障碍物检测距离与实际距离的数值，来评价整个系统的精度。误检率定义为：

$$\text{误检率} = \frac{\text{误检测数}}{\text{总检测数}} \times 100\%$$

所谓误检测，是指物体位于深度阈值所对应的场景距离之外，而被检测为障碍物，或者指物体位于深度阈值所对应的场景距离之内，而不被检测为障碍物。本文在不同设定阈值下进行多次试验，得到了一个统计结果，如图 11 所示：

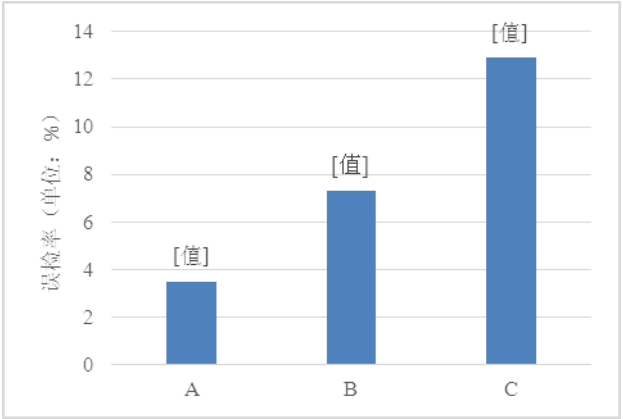


图 11 三种环境下的误检率

Fig. 11 error detection rates under three environments

从图中可以看出，当障碍物数量较少时，系统的误检率比较低，仅为 3.5%，检测效果较好；当障碍物数量较多时，

误检率升高至 7.3%；动态障碍物环境下，误检率最高，为 12.9%。由此可以得到，障碍物之间的相互干扰，会影响障碍物的检测效果，且易出现被检测成为一个大的障碍物的情况；同时运动过程也会造成匹配过程的准确度降低，更容易出现障碍物丢失。在各个检测位置上，算法输出的障碍物中心与机器人的距离和实际距离的误差如表 2 所示：

表 2 障碍物检测的距离误差（单位：米）

Tab. 2 distance error of obstacle detection (unit: meters)

实际距离	简单	复杂	动态
0.3	0.02	0.05	0.08
0.8	0.03	0.06	0.13
1.5	0.04	0.06	0.14
2.0	0.05	0.08	0.16
2.5	0.07	0.09	0.17
3.0	0.09	0.13	0.17
误差均值	0.05	0.08	0.14

从实验数据可知，简单环境下，障碍物检测的距离误差在 5cm 左右，复杂环境下，误差上升到 8cm 左右，两种环境下，检测精度都较高，在控制机器人规避障碍物时，通过设定一定的余量，即可实现机器人的顺利避障。而动态环境下误差最大，达到近 14cm，检测效果不太理想。同时，本文还发现，当障碍物沿 Y 轴运动时，系统的持续跟踪检测能力较强；当障碍物沿 Z 轴运动时，障碍物容易丢失，提取的障碍物与实际大小相差较大。

(2) 障碍物检测实时性

障碍物检测的实时性通过系统处理每帧图像的时间来体现，具体时间如图 12 所示。

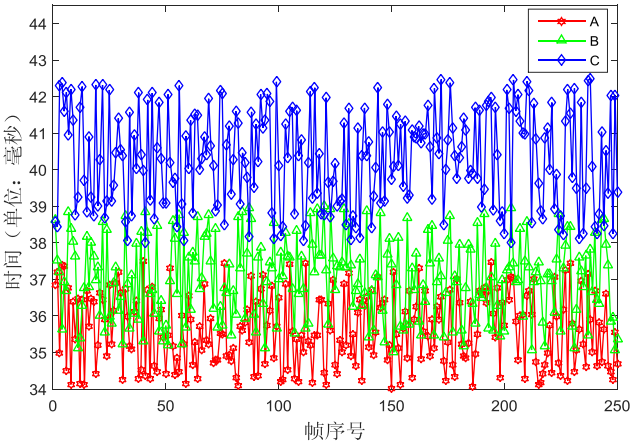


图 12 每帧处理时间

Fig. 12 processing time per frame

从图中可以看出，简单环境下，系统处理每帧图像的时间稳定在 35ms 附近，复杂情况下上升至 37ms 左右，动态环境下系统处理每帧图像耗费的时间最长，为 40ms 左右。总的来说，三种环境下，系统处理图像的速度都相对较快，实



时性较好,基本能够满足机器人一般运动下顺利避障的要求。

(3) 障碍物检测范围

障碍物检测范围是通过改变机器人与障碍物的距离来测试,相应的结果如图 13 所示,从图中可以看出,在三个环境中,简单环境的检测范围最大,最小检测距离为 0.3m,最大检测距离为 2.9m;复杂环境其次,最小检测距离为 0.6m,最大检测距离为 2.8m;动态环境下的最小检测距离为 0.7m,最大检测距离为 2.3m。由此可以得到,三种环境下,障碍物检测范围都较宽,且物体运动对测量范围的影响较大。

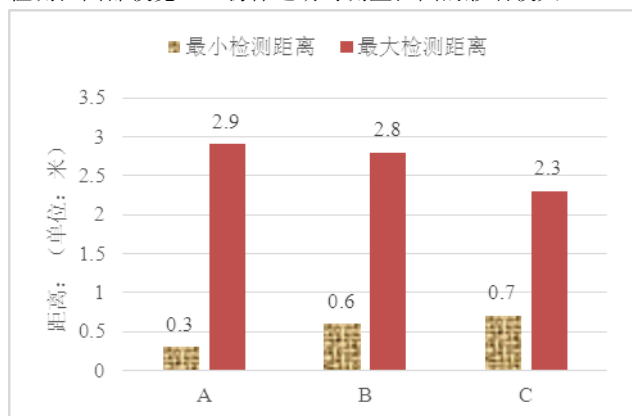


图 13 障碍物检测范围

Fig. 13 obstacle detection range

综上所述,算法能够在实现自主定位的同时,检测出环境中的障碍物,且在三种设定环境下,障碍物检测效果都较好,基本能够满足机器人一般运动下自主避障要求。其中,简单环境下算法检测效果最好,复杂环境次之,动态障碍物对算法检测效果影响较大。

4. 结语

本文主要研究了基于视觉的机器人自主定位和障碍物检测问题。利用典型稀疏性 SLAM 算法 ORB-SLAM2 实现了算法的自主定位功能。针对 ORB-SLAM2 存在的环境信息丢失的问题,在 ROS 架构下,为其引入了基于深度的障碍物检测算法,解决了其不能检测障碍物的问题。在基于立体视觉的障碍物检测算法中,进一步引入深度与面积阈值,提高算法的检测精度。算法在搭建的实验平台上得到了验证,结果表明该系统是有效可行的。如何在具有障碍物信息的环境地图中进行路径规划是未来需要进一步研究的工作。

参考文献

[1] 杨洋,童东兵,陈巧玉.面向未知地图的六足机器人路径规划算法[J]. 计算机应用,2018,38(06):1809-1813. (YANG Y, TONG D B, CHEN Q Y. Six-legged robot path planning algorithm for unknown map [J]. Journal of Computer Applications, 2018, 38(06):1809-1813.)

[2] Naya B, Sabrina H, Stefan W, etc. Navigation and Vision System of a Mobile Robot [C]. 2018 19th International Conference on Research

and Education in Mechatronics (REM), 2018, 99-104.

[3] 胡章芳,鲍合章,陈旭,范霆铠,赵立明.基于改进闭环检测算法的视觉同时定位与地图构建[J].计算机应用,2018,38(03):873-878. (HU Z F, BAO H Z, CHEN X, FAN T K, ZHAO L M. Visual simultaneous location and mapping based on improved closed-loop detection algorithm [J]. Journal of Computer Applications, 2018, 38(03):873-878.)

[4] 张永,杨浩.基于优化视觉词袋模型的图像分类方法[J].计算机应用,2017,37(08):2244-2247. (ZHANG Y, YANG H. Image classification method based on optimized bag-of-visual words model [J]. Journal of Computer Applications, 2017, 37(08):2244-2247.)

[5] Nister D, Naroditsky O, and Bergen J. Visual odometry [J]. Computer Vision and Pattern Recognition, 2004, pp. 652-659.

[6] Engel J, Schops T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM [C]. //13th European Conference on Computer Vision. Berlin, Germany: Springer, 2014: 834-849.

[7] Sim R, Elinas P, Griffin M. Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters [C]. The 3rd Canadian Conference on Computer and Robot Vision (CRV'06), 2006. 21-21.

[8] Menglong Yang, Yiguang Liu, Zhisheng You. The Euclidean embedding learning based on convolutional neural network for stereo matching [J]. Neurocomputing, 2017, 267:195-200.

[9] Georg K, David M. Parallel Tracking and Mapping for Small AR Workspaces [C]. 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, 225-234.

[10] Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System [J]. IEEE Transactions on Robotics, 2015, 32(5):1147-1163.

[11] 周武,赵春霞,沈亚强,张棉好.基于全局观测地图模型的 SLAM 研究[J].机器人,2010,32(05):647-654. (ZHOU W, ZHAO C X, SHEN Y Q, ZHANG M H. SLAM research based on global observation map model [J]. Robot, 2010, 32(05):647-654.)

[12] 王轩,叶平,贾庆轩.双目立体视觉栅格地图构建方法[J].软件,2012,33(11):233-236. (WANG X, YE P, JIA Q X. Stereo vision location and grid map building method [J]. Software, 2012, 33(11):233-236.)

[13] 陈炜楠,朱蕾,张宏,林旭滨,管盼生.稀疏视觉 SLAM 对平面激光雷达传感的稠密化模拟[J].机器人,2018,40(03):273-281. (CHEN W N, ZHU L, ZHANG H, LIN X B, GUAN Y S. Planar LiDAR densified simulation from sparse visual SLAM [J]. Robot, 2018, 40(03):273-281.)

[14] Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardos. ORB-SLAM2: An Open-source SLAM System for Monocular, Stereo and RGB-D Cameras [J]. IEEE Transactions on Robotics, 2016, 33(5):1255-1262.

[15] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information [C]. 2005 IEEE



- Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005. 807-814.
- [16] Galvez-Lopez D and Tardos JD. Bags of binary words for fast place recognition in image sequences [J]. IEEE Transactions on Robotics, 2012, 28(5):1188-1197.
- [17] Suzuki. Topological structural analysis of digitized binary images by border following [J]. Computer Vision, Graphics, and Image Processing, 1985, 30(1): 32-46.
- [18] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset [J]. The International Journal of Robotics Research, 2013, 32(11):1231-1237.

This work is partially supported by the National Natural Science Foundation of China (61503405), Aeronautical science foundation (20160896007).

DING Doujian, born in 1995, M. S. candidate. His research interests include visual navigation, visual SLAM.

ZHAO Xiaolin, born in 1982, Ph. D. , Associate professor. His research interests include uav collaborative control, uav visual navigation technology.