
2D SLAM 技术在机器人操作系统中的应用评价

Joao Machado Santos, David Portugal and Rui P. Rocha

摘要: 本文对机器人操作系统(ROS)中几种基于激光的二维同步定位和地图构建(SLAM)技术进行了研究。所有的方法都在2D模拟和实际实验中进行了评估和比较。为了得出测试技术性能的结论,在相同条件下收集实验结果,并应用基于K-最近邻概念的广义性能度量。此外,还检查了每种技术的 CPU 负载。

这项工作提出了对每个解决方案的劣势和优势的见解,这样的分析对于根据预期最终应用的特性而决定采用哪种解决方案至关重要。

1. 前言

同时定位和地图构建(SLAM)是机器人学中最广泛的研究课题之一。它有助于在未知环境中建立和更新地图,同时机器人保留有关其位置的信息。

本体传感器在估计移动机器人的运动、绘制高维度的环境地图、确定在不同时间点采取的传感器测量值是否对应着世界上同一对象的问题,以及世界随时间变化的事实等问题时,都会受到累积误差的影响,这代表了SLAM^[1]中最大的挑战。

机器人操作系统(ROS)是当今最流行的机器人框架。它提供了一组工具,库和驱动程序,以帮助开发具有硬件抽象的机器人应用程序^[2]。ROS使研究人员能够快速,轻松地进行模拟和真实世界的实验。

在这项工作中分析的所有五种SLAM技术都可以在ROS中获得,并且已经通过Stage在2D模拟和基于Arduino的定制机器人中完成了测试。本文所介绍的研究是我们最终目标的第一步,即提出一种用于城市搜索和救援(USAR)场景的SLAM技术,其环境通常含有烟雾和尘埃颗粒。因此,有必要研究最流行和最常用的方法,这项工作将作为我们后来技术的指导,同样对SLAM和ROS感兴趣的研究人员也是一种指导。

这项工作得到了肖邦研究项目(PTDC EEA-CRO 119000 2010)、博士资助(SFRH BD 64426 2009)和系统与机器人研究所(EST项目)的支持,所有这些项目均由葡萄牙科学机构 Fundacao para a Ciencia e a Tecnologia资助。J.M. Santos, D. Portugal 和 R.P. Rocha 就职于联合国大学系统与机器人研究所,位于葡萄牙科英布拉 3030-290 号,邮编: {jsantos,davidbsp, [rprocha](mailto:rprocha@isr.uc.pt)}@isr.uc.pt。

2. 相关工作

目前,所有公认的机器人建图算法都有一个共同特征:它们依赖于概率。应用概率的优点是对测量噪声的鲁棒性以及在测量和估计过程中正式表示不确定性的能力。大多数用来解决建图问题的概率模型依赖于贝叶斯规则^[1]。

卡尔曼滤波器(KF)是贝叶斯滤波器最受欢迎的实现之一^[1]。KF有两个不同的阶段:预测和更新。预测阶段估计来自先前迭代的状态空间(先验),而在更新阶段中,状态估计与传感器提供的观测相结合,更新阶段的结果称为后验。扩展卡尔曼滤波器(EKF)是基于KF的先验知识,解决了机器人位姿模型的非线性问题。在[4]中对基于EKF的非线性二维SLAM问题的解的收敛性和不一致性问题进行了检验。

粒子滤波器(PF)是贝叶斯滤波器的另一种应用。其后验概率由一组加权粒子表示,每个粒子都被赋予一个重要因子。假设下一个状态仅依赖于当前的状态,即马尔可夫假设^[5]。PFs具有通过多模态分布表示不确定性,并处理非高斯噪声的优点。Montemerlo等^[6]提出了一种名为FastSLAM的新方法。它利用修改的PF来估计后验。之后,每个粒子都具有K次卡尔曼滤波器,用于估计K个标志位置。结果表明,执行该算法的计算工作量低于EKF方法。此外,即使使用少量粒子,该方法用于处理大量的标志,结果仍然合适。此外,在[7]中提出了一种基于PF的方法。这项工作在第3-B节中有更详细的讨论。

同样重要的是基于图形的SLAM算法，因为它们弥补了PF和EKF技术的一些弱点^[9]。在这些SLAM算法中，提取的数据用于构建图。该图由节点和弧组成。图中的每个弧表示连续位姿之间的约束，其可以是运动事件或测量事件。为了获得地图，所有约束被线性化并且获得稀疏矩阵，表示稀疏图。这种类型的算法首先由Lu和 Miliotis^[8]提出。在他们的工作中，通过扫描匹配过程恢复了姿态约束。但是，由于使用的优化过程在还原过程中删除了映射变量，算法在大型场景中的适用性是不切实际的。此外，Carlone等人^[10]也开发了基于图的SLAM方法，在第3-E节中讨论。

在过去几年中，SLAM方法的数量有所增加，比较不同方法的必要性也大大增加。目视检查得到的标测图无法进行正确比较。需要精确地评估结果要求一个更准确的方法——定量尺度。例如，在[12]中，开发了用于比较 SLAM 算法的度量，其中结果不使用参考进行评价，而是通过在数据采集过程中考虑机器人的姿势进行评价。这一事实允许在具有不同输出的算法之间进行比较。此外，提出的方法是独立于移动机器人的传感器配置，但它需要在数据集应用之前手动编辑数据集。

所有公认的 SLAM 评估方法都依赖于社区可用的标准数据集。然而，这些与 ROS 框架并不兼容。相反，在这项工作中，一项在ROS可用的以主要激光为基础的二维 SLAM 算法的研究被提出。所有被测试的技术都使用占用率网格作为最终输出，并使用地图相似性度量进行分析。重点放在地图质量上，而不是位姿估计误差，因为地图输出受定位问题的影响很大。主要目标是提供在 ROS 中实现的所有五种算法的优点和缺点的概述，并提供简单、准确的定量比较，从而为 ROS 用户定义选择最适合其要求的算法的一般指南。

3. 2D SALM算法

在本节中，对五种SLAM技术进行了进行了简要描述，即：HectorSLAM, Gmapping, KartoSLAM, CoreSLAM和LagoSLAM。

A. HectorSLAM

HectorSLAM结合了基于强健扫描匹配的2D SLAM系统和惯性系统3D导航技术的使用。

利用现代激光雷达的高更新率和地距离测量噪声，作者将重点放在对机器人运动的实时估计。不使用里程计信息，这使得这种方法有可能在像四旋翼无人机这样的空中机器人和在不平坦地形中运行的地面机器人中实现。另一方面，当只有低速率扫描可用时，他可能有问题，并且里程计估计值相当准确时，他不发挥作用。目前为止，2D姿态估计基于光束端点与所获得的映射的对准的优化。将端点投影到实际地图中，并估计占用概率。使用高斯-牛顿方程解决扫描匹配问题，该方程可找到地图上激光束最适合的刚性变换。此外，为了避免陷入局部极小值，采用了多分辨率地图表示。最后，基于 EKF 对导航滤波器进行三维状态估计，然而，只有当存在惯性测量单元（IMU）时才需要这样做，例如空中机器人。因此，该方式将不会在本工作中使用。

B. Gmapping

Gmapping是一种基于激光的SLAM算法，如[7]所述。此外，它是在机器人领域全球使用最广泛的SLAM包。该算法由Grisetti等人提出，是一种Rao-Blackwellized PF SLAM方法。PF系列算法通常需要大量粒子才能获得良好的结果，这增加了其计算复杂性。此外，与 PF 重新采样过程相关的损耗问题，降低了算法精度。这是因为重要性权重的粒子可能变得微不足道。因此，这意味着消除正确假设的概率很小。

自适应重采样技术已经在[7]中发展，它最大限度地减少了粒子耗竭问题，这个过程只在需要时才进行。作者还提出了一种来计算精确分布的方法，该方法不仅考虑机器人平台的移动，而且还考虑最近的观察。这减少了 PF 预测步骤中机器人位置的不确定性。结果显示，扫描匹配过程中由于不确定性较低，所需的颗粒数量减少。在我们的实验中，Gmapping使用的粒子数量为30。

C. KartoSLAM

KartoSLAM是 SRI Intentional sKarto Robotics 开发的一种基于图的SLAM方法，通过使用高度优化和非迭代的稀疏线性系统 Cholesky 矩阵分解作为其求解器为 ROS 进行了扩展^[13]。基于图的SLAM算法使用图优化进行构图。在这种情况下，每个节点代表机器人沿着其轨迹的姿势和一组传感器的测量值。它们通过弧连接，弧代表连续姿势之间的运动。对于每个新节点，通过找到与弧的约束一致的节点的空间配置来计算地图。在可用于ROS的KartoSLAM版本中，稀疏姿势调整（SPA）负责扫描匹配和回环检测程序。地标数量越多，所需的内存量就越多。但是，在维护大规模环境的地图时，基于图形的SLAM算法通常比其他方法更有效。在KartoSLAM的特定情况下，它非常有效，因为它只维护姿势图。

D. CoreSLAM

CoreSLAM^[5]是原始200行代码tinySLAM算法的ROS包装器，其创建目的是简单易懂，性能损失最小。该算法分为两个步骤：距离计算和地图更新。第一步，对于每次传入的扫描，它利用非常简单的 PF 算法计算距离。PF 将来自 LRF 的每次扫描与标测图匹配，过滤器的每个颗粒代表机器人的可能姿势，并具有相关的重量，这取决于之前的迭代。选择最优假设后，剔除权重较低的粒子，生成新的粒子。在更新步骤中，在图谱中绘制与接收扫描相对应的线。然而，当探测到障碍物时，tinySLAM 只画出一个点，在障碍物周围画出一组可调整的点。

E. LagoSLAM

基于图的 SLAM 算法的基础是最小化非线性非凸代价函数^[10]。更确切地说，在每次迭代中，初始问题的局部凸近似被解决，以便更新图配置。重复该过程，直至达到代价函数的局部最小值。然而，这个优化过程高度依赖于初始猜测来收敛。Carlone 等人^[10]开发了一种称为 LagoSLAM（图优化的线性逼近）的新方法，其中优化过程不需要初始猜测。此外，该技术可用于任何标准优化器。事实上，ROS 中可用的算法可以在三种不同的优化器之间进行选择：基于树的网络优化器（TORO）^[7]、g2o^[16]和 LAGO^[10]。在进行的实验中，使用了 LAGO 优化器。假设图中每个节点的相对位置和相对方位都是独立的，作者等价于求解了一个非凸代价函数的方程组。为此，提出了一套基于图论的方法，通过线性定向和线性位置估计，获得非线性系统的一阶近似。

4. 结果和讨论

所描述的所有 5 种 SLAM 技术均采用 2D 模拟和真实实验进行了测试。模拟在Stage^[8]中进行，Stage是集成在ROS中的逼真的2D机器人模拟器。此外，还在现实世界场景中使用物理机器人进行了测试，以研究这些 SLAM 功能包在缺乏完美模拟条件下的行为。尽管 SLAM 算法在无噪声的里程计和距离传感信息等阶段模拟中具有良好的条件，但在测量不确定的情况下，SLAM 算法可能无法得到理想的结果。在所有实验中，机器人都是遥控操作的。请注意，ROS 提供的抽象层允许将相同的代码用于模拟和实际实验。HectorSLAM 需要具有高更新率的 LRF。实验中使用的 Hokuyo URG-04LX-UG01 LRF 的更新速率为 10 Hz，Stage中使用相似的最大更新速率。为了解决这个问题，机器人采用低角速度和低线速度驱动。在执行的测试中，每种方法的输出（如前所述）是各自生成的 2D 占用比栅格地图。

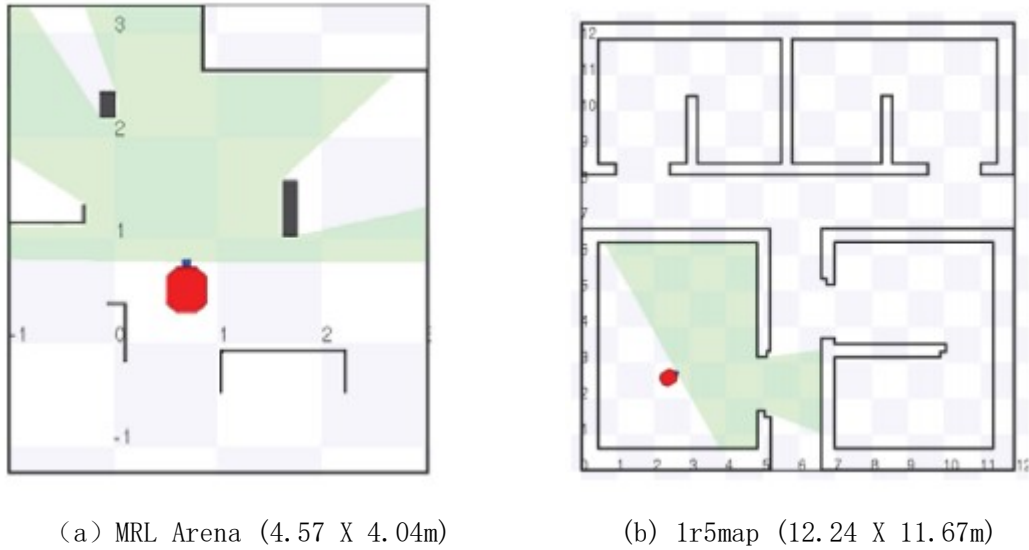


图1: 模拟实验中使用的地图

表1: 在MRL竞技场中每种算法的误差估计 (模拟实验)

Simulation Experiments				
HectorSLAM	Gmapping	KartoSLAM	CoreSLAM	LagoSLAM
0.4563	0.4200	0.5509	11.8393	1.4646

为了评价得到的地图的质量, 对生成的地图和地面实况之间的误差进行了分析。使用基于 k -nearest 概念的性能指标。为此, 使用基于强度的图像配准工具, 计算得到的地面实况和图谱之间的最佳拟合对齐 (见图 2)。

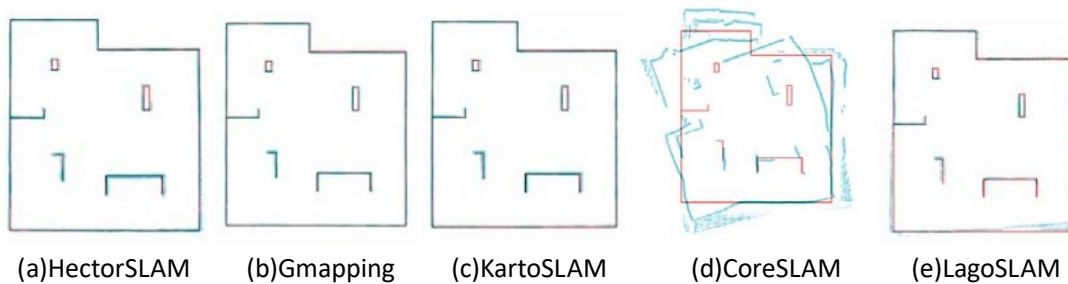


图2: 在 MRL Arena 环境中通过模拟获得的地图。红色代表地面实况, 蓝色代表最终地图。

该过程工作如下: 每个算法的最终构图是二值化的。二值化地图只包含场景的边界和障碍物。然后, 使用图像处理工具箱中的一组 **Matlab** 函数将二值化映射与相应的地面实况狭义对齐。由于地面实况图和生成的地图都是对齐的, 因此使用 **knnsearch** 确定从地面实况图的每个占用单元到结果地图中最近的单元的距离, **knnsearch** 计算 k -nearest 临近单元 (在这里 $k = 1$)。然后将得到的所有距离之和除以地面实况图中所占单元的数量。这个误差度量提供了距离的标准化度量 (以单元格为单位), 只要地面实况地图可用的情况下, 就可以在任何通用占用网格中应用。

A. 仿真实验

Stage 模拟使用两种不同的地图进行: MRL Arena 和 1r5map, 如图 1 所示。特别集中在前者, 因为 MRL Arena 同时应用于模拟和真实实验。1r5map 地图能够在更大的场景中分析 SLAM 技术的行为, 并且每平方英尺的功能更少 (参见图 1b)。这对于分析每种方法对地标的依赖性尤为重要。在模拟实验中, 距离传感器的模型定义就像现实世界实验中使用的传感器: Hokuyo URG-04LX-UG01, 其最大测量距离约为 5.6 米。使用键盘^[9]执行遥控操作。机器人的所有感知和运动数据 (LFR、里程表、速度指令等) 被先记录, 然后为每个算

法实验。因此，所有 SLAM 包均在相同的精确条件下进行测试。正是由于 rosbag 工具^[10]的存在，这才可能发生。

对于每种算法，最终标测图的分辨率设置为 0.01 米/像素。为了缓解低扫描率，将 HectorSLAM 中使用的子图数量定义为5。由于每个子图具有其先前子图的一半分辨率，因此扫描匹配过程更准确，即在较低分辨率图中扫描匹配性能更高。在所有实验中，都使用默认参数。例如，前面提到的Gmapping 算法的粒子数是 30。

分析MRL地图的模拟结果，根据表I和图2，Gmapping和HectorSLAM生成的地图具有最低的误差和相似的误差。另一方面，KartoSLAM出现稍大的错误，而CoreSLAM的结果呈现出最高的错误值。Gmapping 是一个极度优化的 PF 算法，具有改进的重采样过程，这保证了所得地图的质量。同时，HectorSLAM 的扫描匹配过程显示了其有效性。然而，必须注意的是，为补偿来自 LFR 的速率更新，给机器人的低速命令，对实验结果产生了一定影响。由于 KartoSLAM 和 LagoSLAM 都是基于图的 SLAM 方法，比较它们之间的误差是有趣的。两人都成功绘制了Arena 地图。然而，LagoSLAM 获得的误差最大（CoreSLAM除外）。这种现象可以通过KartoSLAM采用的SPA求解器方法具有令人印象深刻的性能来解释。然而，用 LagoSLAM 得到的地图的质量仍然是合适的。

为了比较不同场景中的所有 SLAM 方法，还使用 1r5map 进行了一系列模拟。这些结果见表2和图3。1r5map 是一个相对较大的 map，具有少量的标志地标。在这种情况下，HectorSLAM 获得了比 Gmapping 更高的误差值。原因之一是 HectorSLAM 在很大程度上依赖于连续测量之间的扫描匹配，由于这些模拟实验中使用的传感器的特性，无法观察到 HectorSLAM 的全部潜力。此外，由于标志建筑数量减少，由于扫描匹配过程未提供足够的信息，因此误差不断增加。还有就是它不使用里程表信息，当通过固定宽度的长廊时会出现一些问题。总之，在本试验中使用 HectorSLAM 获得的较差结果不足为奇。更多的是，Gmapping 算法呈现了例外的结果，它揭示了 PF 方法的精确性。KartoSLAM 获得了第二低的误差值，揭示了基于图的 SLAM 方法的稳健性。再者，LagoSLAM 获得了比 KartoSLAM 更高的误差值，CoreSLAM 是性能最差的算法。误差值是通过地面真值中的点与图谱中最近点之间的欧氏距离获得的，由于图谱尺寸较大，在 1r5map 中获得的误差大于在其他实验中获得误差，这在 CoreSLAM 的情况下尤其明显。

表2 1r5map中每种算法的误差估计

Simulation Experiments				
HectorSLAM	Gmapping	KartoSLAM	CoreSLAM	LagoSLAM
7.4581	5.3670	5.4380	171.5218	9.3041

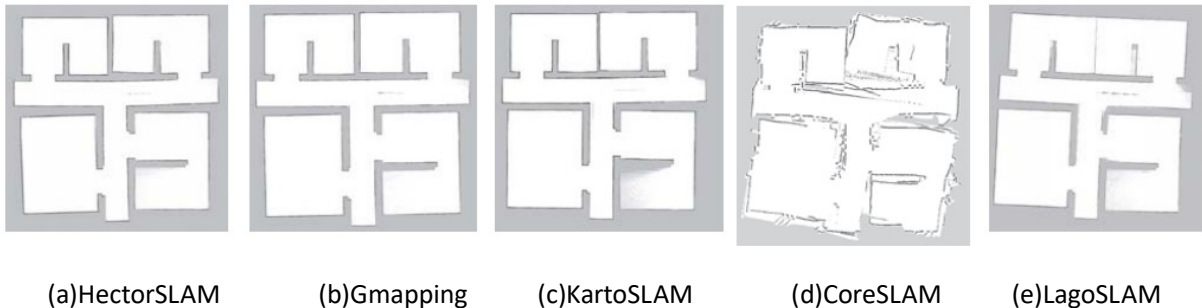


图 3 在 1r5map 环境中通过模拟获得的栅格地图

B. 真实环境实验

在现实世界实验中，使用配有Hokuyo URG-04LXUG01 和 Asus eeePC 1025C传感器的 Stingbot^[11]机器人，运行 Ubuntu 11.10 和 ROS Fuerte，进行了三次不同轨迹和初始位置的运行。同样，记录了之前的所有数据，随后对每个算法进行了实验。采用了仿真实验所采

用的算法参数，在真实的 MRL Arena场所中进行测试。

图 4 显示了所有五种技术都能够成功地构建地图，每种算法得到的误差如表3所示。可以看出，一般而言，所有技术产生的结果都比模拟中的结果要差。由于在绘制真实世界 MRL Arena 时，激光扫描数据上的机器人位置存在估计误差和噪声，产生了这种轻微的性能影响。对误差的分析可以提供有关算法性能的更准确的信息。

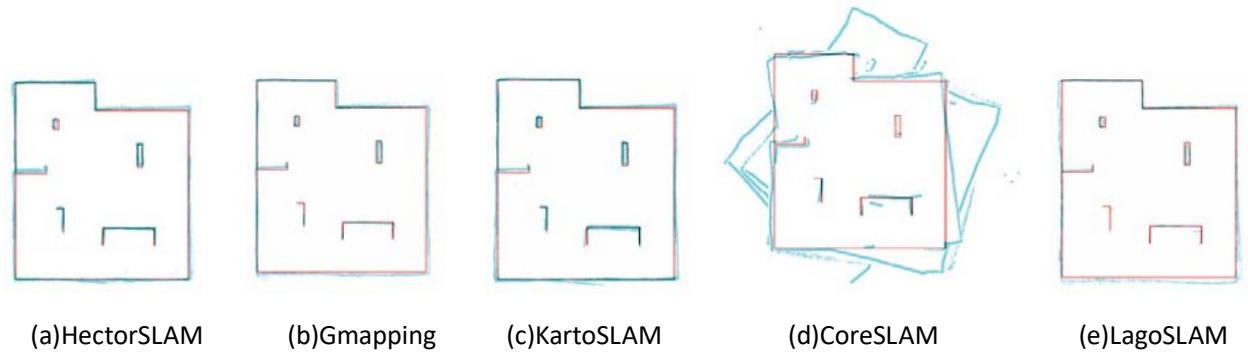


图4 真实世界中的性能分析。红色代表地面实况，蓝色代表最终地图

表3 MRL Arena中每个算法的误差估计（真实实验）

Real World Experiments				
HectorSLAM	Gmapping	KartoSLAM	CoreSLAM	LagoSLAM
1.1972	2.1716	1.0318	14.75333	3.0264
0.5094	0.6945	0.3742	7.9463	0.8181
1.0656	1.6354	0.9080	7.5824	2.5236

尽管虚拟环境和现实环境之间存在差异，但从两种设置中提取的结果都遵循一些一般趋势，特别是对于HectorSLAM，Gmapping和LagoSLAM。根据[15]的作者，CoreSLAM 可以在几种不同的环境中具有良好的性能；然而，从我们的实验中得到的结果并不支持这种说法。

在 KartoSLAM 算法中，真实世界实验中得到的误差并不比模拟中的误差大多少。事实上，一般来说，KartoSLAM 是真实世界中表现最好的技术，受噪音的影响比其他方法小。这可以解释，这不仅是由于 KartoSLAM 中使用的 SPA 求解器的性能所致，而且还因为它是一种完整的 SLAM 方法，即使用整个路径和地图获得地图，而不仅仅是最新的地图和姿态。在所有实验中，CoreSLAM 的较差结果显示其环闭合程序很少收敛，这在显示真实世界实验和所有详细结果的视频中很明显。

除进行误差分析外，还评价了使用每种技术对计算负载。图 5 和表 4 比较了配有 Intel Core i7-3630QM 和 8 GB RAM 的笔记本电脑在运行每种算法时的 CPU 负载。

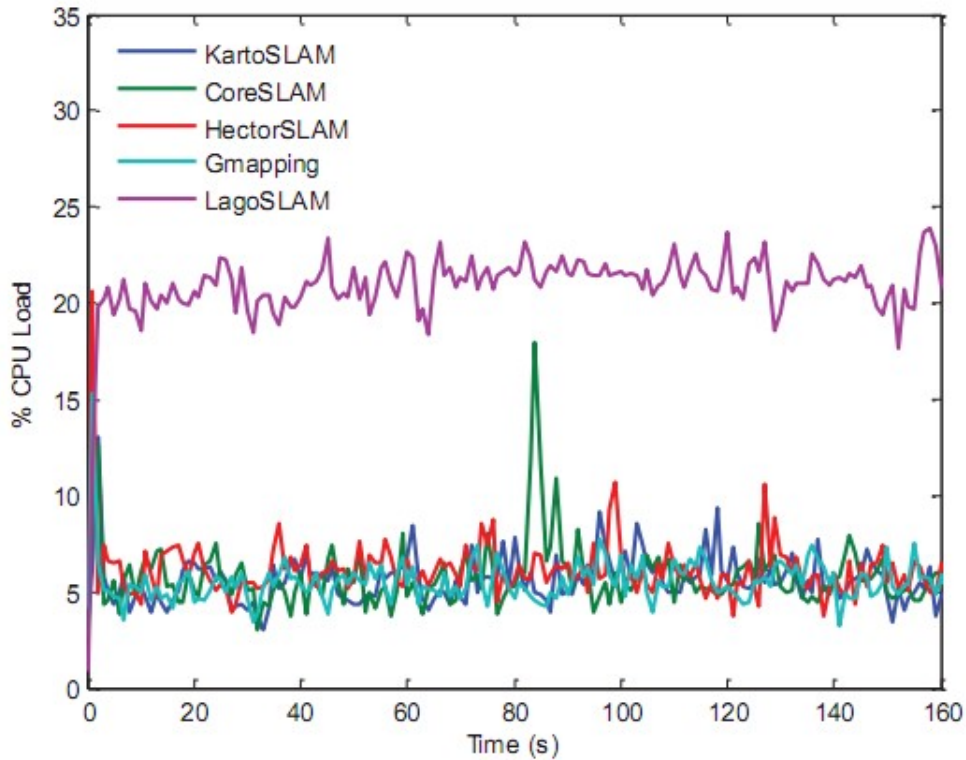


图5 使用真实数据集的每个 SLAM 方法的 CPU 负载的变化

表4 2D SLAM 方法的 CPU 负载 (%)：平均值 (\bar{x})、中位数 ($\sim x$) 和标准差 (σ)

	HectorSLAM	Gmapping	KartoSLAM	CoreSLAM	LagoSLAM
\bar{x}	6.1107	7.0873	5.4077	5.5213	21.0839
$\sim x$	5.9250	5.5800	5.3000	5.4400	21.2250
σ	1.993	4.4287	1.3018	1.6311	2.1684

仔细看结果，LagoSLAM 的 CPU 使用率最高。而且，得到的值与其他四种算法相差相当远。这可以通过为达到给定图形配置的最小代价函数而开发的过程来解释，如第3-E节所述。实验期间其他四种方法所需的资源相似，如表4所示。该CPU分析显示，在现场实验中，所分析的所有五种算法在所需资源方面都非常有效，并且可以在线采用，以绘制通用2D场景。

C. 讨论

根据我们的实验，一些观点可以被保留。一方面，HectorSLAM 仅依赖扫描匹配，不使用里程计，根据机器人和环境特性，这可能是一种优势或者时一种劣势。另一方面，理想情况下，应该使用特定硬件进行测试，例如高频率 LFR。

Gmapping 在所有实验中都表现出了它的稳健性，在每个实验中，它的错误和 CPU 负载总是很低。它结合了扫描匹配信息和里程表信息，以尽量减少粒子的数量。

KartoSLAM 和 LagoSLAM 都是基于图的 SLAM 方法，但它们的结果明显不同。KartoSLAM 的 CPU 负载较低且地图精确表达，而 LagoSLAM 生成了误差和 CPU 负载都较大的地图。造成这种差异的原因与两种技术不同的图形配置和图形优化过程有关。

最后，CoreSLAM 取得了不太令人印象深刻的结果，可能表明其环路闭合机制缺乏收敛性。CoreSLAM使用简单的 PF，它需要更多的粒子，但与每个粒子相关的计算能力较低。根据[15]，CoreSLAM使用一个非常简单的PF来匹配LFR读数和地图，这可能导致错误的位置估计。此外，最初工作的重点是提供一种简单的 SLAM 技术，能够在长廊内导航而不会丢失其位置，而不是环路闭合系统。

5.结论

在这项工作中，五个在ROS中可用的代表性的二维 SLAM 算法进行了测试，通过模拟和现实世界的实验，对每种解决办法的弱点和长处进行了讨论。提供了可用于ROS的每种 2D SLAM技术的准确概述，以根据一个人的要求阐明方法的选择。

在未来的工作中，我们打算在 ROS 中开发一种 2D SLAM 技术，用于低能见度的室内场景，例如烟雾。基于本文中观察到的性能，这种新技术将可能适应 Gmapping 或 KartoSLAM，并将其扩展到 LRF 以外的更多传感输入信息，例如声纳、IMU 和/或粉尘传感器。

参考文献

- [1] S. Thrun., W. Burgard, D. Fox., Probabilistic Robotics, MIT Press, 2005.
- [2] M. Quigley et al., ROS: an open-source Robot Operating System. In IEEE International Conference on Robotics and Automation(ICRA), Workshop on Open Source Software, 2009.
- [3] A. Araújo, D. Portugal, M. Couceiro and R. P. Rocha. Integrating Arduino-based Educational Mobile Robots in ROS, In Int. Conf. on Autonomous Robot Systems, Lisbon, Portugal, April 25-29, 2013.
- [4] S. Huang, G. Dissanayake. Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM, In IEEE Trans. on Robotics, 2(5), Oct. 2007.
- [5] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo Localization for Mobile Robots, In Artificial Intelligence, 128, 2001.
- [6] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, In AAAI National Conference on Artificial Intelligence, 2002.
- [7] G. Grisetti, C. Stachniss, W. Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters, In Trans. On Robotics , 23(1), Feb. 2007.
- [8] F. Lu, E. Milios, Globally Consistent Range Scan Alignment for Environment Mapping, In Autonomous Robots, 1997.
- [9] S. Thrun, M. Montemerlo. The GraphSLAM Algorithm With Applications to Large-Scale Mapping of Urban Structures, In Proc. of the Int. Journal on Robotics Research, 2005.
- [10] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A linear approximation for graph-based simultaneous localization and mapping, In Proc. of the Int. Conf. Robotics: Science and Systems, 2011.
- [11] S. Kohlbrecher, J. Meyer, O. Von Stryk, U. Klingauf. A Flexible and Scalable SLAM System with Full 3D Motion Estimation, In the Int. Symp. on Safety, Security and Rescue Robotics (SSRR), Nov. 2011.
- [12] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, A. Kleiner. On Measuring the Accuracy of SLAM Algorithms, Autonomous Robots, 27(4), Nov. 2009.
- [13] R. Vincent, B. Limketkai, M. Eriksen. Comparison of indoor robot localization techniques in the absence of GPS, In Proc. of SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XV of Defense, Security, and Sensing Symposium, April 2010.
- [14] K. Konolige, G. Grisetti, R. Kümmerle, B. Limketkai, R. Vincent, Efficient Sparse Pose Adjustment for 2D Mapping, In Proc. of Int. Conf. on Intelligent Robots and Systems (IROS), Oct. 2010.
- [15] B. Steux, O. El Hamzaoui. tinySLAM: A SLAM algorithm in less than 200 lines C-language program, In Proc. of the Int. Conf. on Control Automation Robotics & Vision (ICARCV), Dec. 2010.
- [16] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard: g2o: A General Framework for Graph Optimization, IEEE International Conference on Robotics and Automation (ICRA), 2011.