dt -SLAM:一个使用 Detectron2 的高度动态场景的语义视觉 SLAM

阿里
Eslamian
电气与计算机工程系
伊斯法罕科技大学，伊朗伊斯法罕
a.eslamian@ec.iut.ac.ir

null
Mohammad Reza
Ahmadzadeh
电气与计算机工程系
伊斯法罕科技大学，伊朗伊斯法罕
Ahmadzadeh@iut.ac.ir

据专家介绍，同步定位和地图绘制(SLAM)是自主机器人系统的内在组成部分。**在过去的几十年里，已经发明和使用了几个性能令人印象深刻的 SLAM 系统。然而，仍然存在未解决的问题，例如如何处理动态情况下的移动物体。经典的 SLAM 系统依赖于静态环境的假设，这在高度动态的情况下是不可行的。近年来，已经提出了几种方法来解决这个问题，但每种方法都有其局限性。本研究结合视觉 SLAM 系统 ORB-SLAM3 和 Detectron2，提出了 Det- SLAM 系统，该系统利用深度信息和语义分割来识别和消除动态点，完成动态情况的语义 SLAM。对公共 TUM 数据集的评估表明，dt -SLAM 比以前的动态 SLAM 系统更具弹性，并且可以降低动态室内场景中相机姿态的估计误差。**

关键词:视觉 SLAM;动态环境;语义分割;Detectron2

我的介绍。

在大多数 SLAM 算法中，室内或密闭区域优先考虑。通常认为这些位置不包括移动物体。虽然这个假设在现实世界中是不切实际的，但有几种方法可以解决这个问题。第一种解决方案使用几何方法忽略移动组件，并基于静态区域创建结构地图。第二种解决方案使用深度神经网络来检测特定的运动物体，并将其排除在后处理算法之外。

此外，随着深度学习的进步，语义信息对于机器人理解其工作环境至关重要;特定的网络可能会实现更显著的语义分割。将这些网络与 SLAM 相结合，可以创建一个语义图，增强机器人的感知能力。第三种选择使用形态学或背景减法器等图像处理技术，以静态场景移除和替换移动元素。这些策略各有优缺点，可以合并起来构建一个与各种情况兼容的新算法。

2。相关的工作

视觉 SLAM 算法一般接受单目、立体或 RGB-D 摄像机输入。在最近的研究中，已经部署了一种基于事件的相机来提高运动模糊性能。这些算法使用各种方法从图像中提取数据，为导航器生成地图。

A. SLAM 架构

假设环境是静止的，现有的算法可以分为两个子组。在间接法中，将每一帧中每个像素的照度与下一帧进行比较;然后，计算相关性，生成地图。根据相关性对应，将其分为密集、半密集和稀疏地图。这种方法对运动模糊更有弹性，尽管它的计算成本很高，而且容易受到像素噪声的影响。DTAM[1]就是这些生成密集地图的算法之一。LSD-SLAM[2]是一种实时技术，可以生成半密集的环境图，但不支持闭环。DSOD[3]是生成稀疏地图的算法的另一个例子，特别是在动态环境中。

在基于特征的方法中，该算法仅从每帧中提取和分析特定的特征点。由于其较低的计算成本，它比直接技术更受欢迎。Mono-SLAM[4]是一种基于特征的算法，使用滤波器跟踪地标并估计位置。然而，大量的地标会导致更高的计算成本。ORB-SLAM2[5]是最成功的基于特征的算法之一，它由三个主要线程组成，用于跟踪和映射，并且是许多后续基于特征的算法的基础。这些算法在动态情况下容易出现许多故障，但尽管进行了修改，但它们可能会在以后的算法中使用。

B.语义 SLAM

DS-SLAM[6]是一种基于 ORB- SLAM2 的鲁棒算法，有两个额外的线程。该算法将语义分割网络与光流方法相结合，在语义上呈现八叉树映射，减少了基于视觉的 SLAM 中动态目标的影响。在分段网中[7]，首先识别已知的动态对象;随后，对其执行移动一致性检查

被分割区域的选定特征点。如果确定关键点的质量是动态的，则在接下来的步骤中忽略整个对象的特征点。

DynaSLAM[8]是一个具有动态对象识别和绘制功能的视觉 SLAM 系统。使用 Mask-RCNN[9]，该系统检测并消除先验的动态项目。此外，多视图几何有助于定位未识别的动态物体。该算法使用先前分割对象的深度图像来绘制所需区域。然后它估计场景的静态组件的地图，这是现实世界应用所需要的。接下来，使用 ORB-SLAM2 特征提取技术提取关键点。最后，计算每个关键点的视差角度。因此，视差超过 30 度的不动物体由于视点差异被认为是动态的。

DP-SLAM[10]的一般性能是在 Dyna-SLAM 基础上发展而来的。对于语义分割，该算法采用 Mask R-CNN。此外，像 DS-SLAM 这样的极外几何方法可以排除错误分类的特征点。由于最高的错误概率发生在分割的边界边缘，因此算法根据之前的帧和贝叶斯定律确定移动点的概率。因此，如果特征点很可能超过阈值，则从进一步处理中丢弃。

YOLO-SLAM[11]基于 ORB-SLAM2，但包含两个额外的线程，语义分割和动态特征筛选。首先，分割线程使用修改版本的 Darknet19-YOLOv3 来选择最知名的动态对象。动态特征筛选线程使用几何深度 RANSAC 将特征点划分为动态和静态子组。该方法计算每个边界框中特征点的深度方差，以区分静态点和动态点。

PSPNet-SLAM[12]是一种实用的算法。该方法与 DS-SLAM 相似，采用语义分割和光流对运动实体进行识别和分类。这种方法与 DS-SLAM 的不同之处在于它使用 PSPNet 而不是分段网进行分割。改进的 RANSAC 模型也用于消除异常值。

最近提出的另一种算法是 Blitz-SLAM[13]，也是基于 ORB-SLAM2。还有两个线程，使用 BlitzNet 的语义分割和一个几何组件。由于该网络的分割不准确，因此采用深度信息来改变分割掩码。在对运动元素与背景进行适当的分割和分离后，评估静止背景的极极条件，并制备点云。

3。系统概述

我们的方法基于 ORB-SLAM3[14]，它在实际环境中具有出色的性能。而且，它是基于 ORB 特性的并发 SLAM 系统的最后一个版本。从遵循三个主要并行线程的输入构建半密集地图。虽然它在高度动态的情况下会失败，但在静态的情况下却能有效地执行。我们在我们的技术中包含了一个语义分割线程来检测和删除帧中的每个动态项。

null 因此，在进一步处理之前消除了移动组件。此外，我们使用图像处理技术修改了部分深度图像输入，这有助于识别以前未检测到的移动物体。这使得该算法比使用几何方法的同类算法运行得更快。我们使用 Detectron2[15]对 ORB-SLAM3 进行预处理;所以它被称为 Det-SLAM。我们的算法主要由两部分组成:

A.语义分割

在这项工作中，我们应用了 Detectron2 的最先进的目标检测来检测动态目标。Detectron2 是由 Facebook 人工智能研究小组(FAIR)代表的最新生成库。它有效地为几个计算机视觉研究项目和 Facebook 应用程序做出了贡献。骨干网、区域建议网(RPN)和盒头是 Detectron2 的三个基本架构组件。骨干网设计利用特征金字塔网络(Feature Pyramid network, FPN)从不同尺度的输入图像中提取特征映射。在 RPN 中，利用置信度分数在不同尺度上提取特征，并在盒头中裁剪项目的感兴趣区域(ROI)。

我们在即时分割模块中使用 COCO 对象检测基线的预训练权重来识别动态对象，例如主要是移动的人类。ResNet 和 FPN 骨干网分别与卷积层和 FCN 头一起用于预测蒙版和盒。这些项目的边界将被确定，该区域将被突出显示，以便进一步处理。Detectron2 利用边界框来定义物体的空间位置，并独立绘制物体的边界。

然而，分割会产生图像边缘，从而产生 ORB 特征;该方法将删除位于项边界内的任何 ORB 特性。相应地，位于该区域内的关键点将被省略。边界框内包含的其他 ORB 功能将是下一阶段的候选功能。

B.深度处理

在接下来的步骤中，准备由边界框确定的 RGB 图像中对应于特定 ROI 的深度图。在这个阶段，考虑图像的边界框内所有区域的深度信息。放置在 ROI 中对象深度范围内的 ORB 特征被排除在进一步处理之外。

消除所有具有近似相同深度的运动物体的 ROI，以去除受运动因素影响的未检测到的可疑物体，即使静态点的一些数据会丢失。

让 $R^{t\ null}$ 表示 t 时刻 ROI 信息的深度范围，表示深度为的像素。目标区域中深度信息的最小值和最大值定义为

$$m_{obj} = \min_{object\ area}\{R_i^t\} \quad (1)$$

$$M_{obj} = \max_{object\ area}\{R_i^t\} \quad (2)$$

ROI 的最大和最小深度值定义为

$$m_{ROI} = \min_{ROI}\{R_i^t\} \quad (3)$$

$$M_{ROI} = \max_{ROI}\{R_i^t\} \quad (4)$$

因此，我们将 ROI 的深度差定义为

$$d = M_{ROI} - m_{ROI} \quad (5)$$

我们定义一个系数$\alpha$to 确定深度差的比值。因此，在深度范围为的 ROI 内的所有像素

$$m_{obj} - \alpha.d \leqslant R_i^t \leqslant M_{obj} + \alpha.d \quad (6)$$

将从进一步的处理中移除。确定图像纹理上αdepends 的值。图像照度对比度越高，αvalue 值越低。由于物体是连续的，所以物体的深度信息是在一个有限的范围内。在实例分割中，物体的边界可能会超过物体，因此某些区域的深度信息可能会在边界上有一些突变。因此，我们只考虑具有最大数量相同深度信息的像素。图 1 显示了 Det-SLAM 的整体视图系统组件。

评估和结果

在本节中，展示了证明 Det-SLAM 有效性的实验结果。使用公共 TUM RGB-D 数据集[16]，我们将我们的技术与同一类别中的方法进行了比较。尽管我们的技术不是实时的，语义分割线程必须在特征提取之前准备好，但实现了最佳的速度/精度平衡。TUM RGB-D 数据集包含许多包含低动态范围和高动态范围场景的序列，这些序列在可比条件下进行评估。我们采用绝对弹道误差(ATE)，将每个时间戳的投影轨迹与地面真实情况进行比较。由于 RANSAC 算法删除了部分输入图像，因此无法将新提取的 ORB 特征与前一帧匹配，因为当前帧中对应的 ORB 特征可能在下一帧中丢失。因此，由于某些帧中关键点的丢失，轨迹已经丢失。为了进一步验证评估结果，我们加入了匹配对的轨迹比例。表 1 显示了剩余的基础真值和输出对的均值、中位数和标准差。此外，表 1 给出了 DS-SLAM、Dyna-SLAM、DP-SLAM、Blitz-SLAM、PSPNet-SLAM、YOLO-SLAM 等视觉动态 SLAM 算法用于评估动态情况下细节轨迹精度的 ATE 值。在上述 SLAM 算法中，有五种已知的具有动态目标的序列。“f/s/static”序列显示两个人做了一会儿手势。然而，他们并没有在房间里移动，我们的算法检测到他们是动态的。“f/w/static”是两个行走的人的同一个场景。此外，“f/w/half”是相同的场景，但相机围绕球体的一半移动。“f/w/xyz”和“f/w/rpy”序列也来自相同的场景，但相机分别沿着滚转-俯仰-偏航的主轴和 xyz 轴移动。最后四个序列代表了高度动态的场景。表 1 证实，对比其他方法，“f/w/static”和“f/s/static”序列的结果进行了优化。这两者的共同之处 null 序列的稳定性，因为我们没有修改 Det-SLAM 算法的跟踪部分。

相对姿势误差(Relative Pose Error, RPE)决定了两个帧序列之间的关系，如表 2 所示。

图 2 描述了每个数据集的 ATE 和 RPE。可以看出，轨迹漂移时间在所有序列中并不显著。所有的研究都是在英特尔酷睿 i7、GeForce GTX 1650 GPU 和 16GB RAM 的计算机上进行的。

结论

本文通过开发 Detectron2，提出了一种新的目标检测和语义分割算法——Det-SLAM。Detectron2 比其他同类模块更精确。在完全连接网络的语义分割中，类标签之间的关系可能会丢失。此外，它可能会忽略小的项目或偶尔识别巨大的对象。Det-SLAM 提高了现有 SLAM 算法的鲁棒性，减少了动态目标对姿态估计的影响。尽管我们的方法在 TUM RGB-D 数据集上取得了成功的性能，但与其他可比较的算法相比，它在某些序列上存在一些缺陷。我们在可比算法中替换了图像处理技术而不是几何约束。因此，计算量被最小化。虽然我们没有将算法的执行时间与其他方法进行比较，但我们的方法可以在标准 PC 桌面上执行。从另一个角度来看，在这个过程中，我们根据条件丢弃了大量的视觉信息，这导致了跟踪帧的错误和一些序列的缺失。所以，可以说该系统不适用于室外环境或摄像机快速运动的情况。

与前面提到的类似方法相比，该算法提供了几个好处。DS-SLAM 中使用的 SegNet 不受对象重叠的影响。这会导致最终结果出现错误。然而，这并没有对所提出的算法构成重大问题。此外，我们使用 ORB-SLAM3 代替 ORB-SLAM2，更有效地优化了关键帧处理。它也有可能在未来使用 IMU 部分。这项工作的未来发展可能包括实时性能，补偿缺失信息的 IMU 数据，以及将前景与输入图像分离的更多图像处理方法。

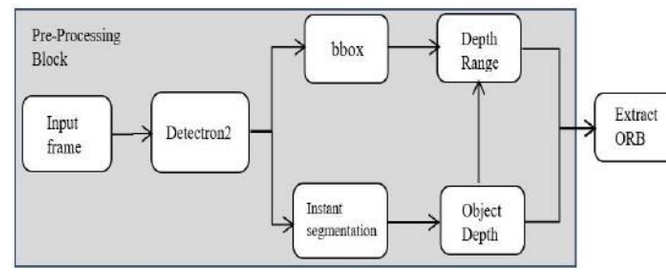


图 1 Det-SLAM 预处理框架

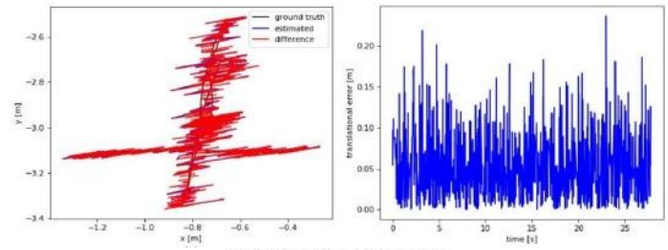
TABEL 1。绝对弹道误差度量结果

sequences		f/w/xyz	f/w/static	f/w/rpy	f/w/half	f/s/static
DS-SLAM	RMSE	0.0247	0.0081	0.4442	0.0303	0.0065
	Mean	0.0186	0.0073	0.3768	0.0258	0.0055
	Median	0.0151	0.0067	0.2835	0.0222	0.0049
	S.D.	0.0161	0.0036	0.2350	0.0159	0.0033
Dyna-SLAM	RMSE	0.0156	0.0068	0.0417	0.0301	0.0063
	Mean	0.0134	0.0059	0.0312	0.0258	0.0055
	Median	0.0118	0.0052	0.0240	0.0218	0.0049
	S.D.	0.0079	0.0034	0.0275	0.0155	0.0031
DP-SLAM	RMSE	**0.0141**	0.0079	0.0356	**0.0254**	0.0059
	Mean	**0.0120**	0.0070	0.0277	**0.0219**	0.0051
	Median	**0.0106**	0.0063	0.0224	**0.0183**	0.0047
	S.D.	0.0073	0.0037	0.0218	0.0129	0.0029
Blitz-SLAM	RMSE	0.0153	0.0102	0.0356	0.0256	-
	Mean	-	-	-	-	-
	Median	-	-	-	-	-
	S.D.	0.0078	0.0052	0.0220	**0.0126**	-
PSPNet-SLAM	RMSE	0.0156	0.0072	**0.0333**	0.0255	0.0058
	Mean	0.0135	0.0064	**0.0261**	0.0222	0.0050
	Median	0.0117	0.0058	**0.0204**	0.0196	0.0044
	S.D.	0.0078	0.0033	**0.0206**	**0.0126**	0.0029
YOLO-SLAM	RMSE	0.0146	0.0073	0.2164	0.0283	0.0066
	Mean	-	-	-	-	-
	Median	-	-	-	-	-
	S.D.	**0.0070**	0.0035	0.1001	0.0138	0.0033
Det-SLAM (Ours)	RMSE	0.0482	**0.0017**	0.0389	0.0925	**0.0036**
	Mean	0.0383	**0.0012**	0.0309	0.0739	**0.0031**
	Median	0.0329	**0.0008**	0.0256	0.0631	**0.0030**
	S.D.	0.0296	**0.0012**	0.0237	0.0557	**0.0017**
	Traj	35.12%	50.06%	48.63%	47.99%	49.66%

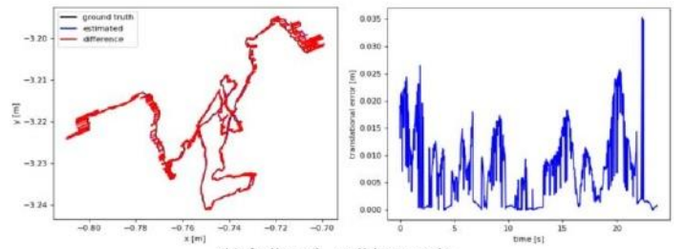
TABEL 2。我们的 Det-SLAM 相对姿势误差度量结果

Sequence	f/w/xyz	f/w/static	f/w/rpy	f/w/half	f/s/static
RMSE	0.0653	0.0100	0.0680	0.01674	0.0223
Mean	0.0517	0.0074	0.0551	0.1310	0.0180
Median	0.0450	0.0065	0.0478	0.1045	0.0141
S.D.	0.0398	0.0067	0.0398	0.1042	0.0132

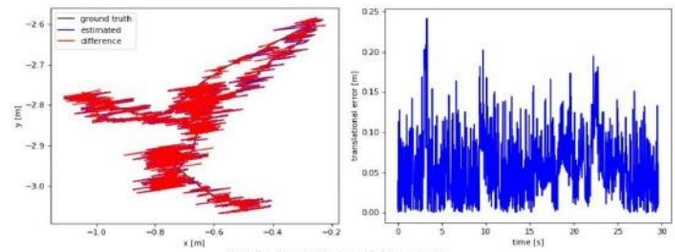
null



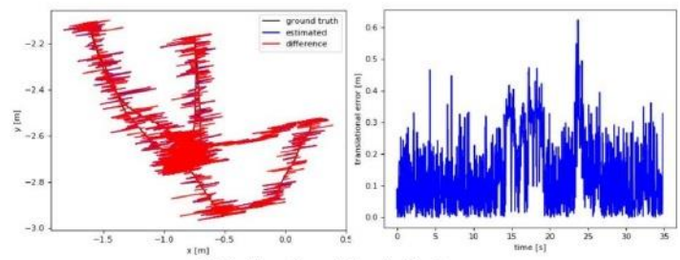
(a) freiburg3_walking_xyz



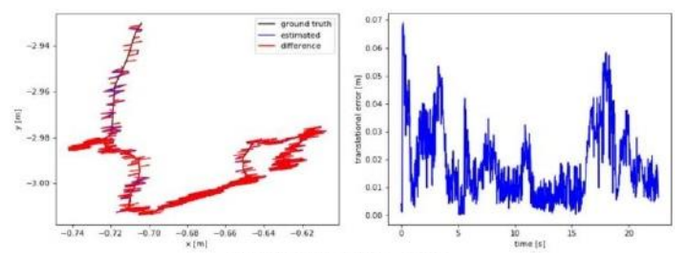
(b) freiburg3_walking_static



(c) freiburg3_walking_rpy



(d) freiburg3_walking_halfsphere



(e) freiburg3_sitting_static

图 2 5 个 TUM RGB-D 序列在我们的 dest - slam 算法中的结果。(左)基于估计和 ground truth 二维轨迹的 ATE。
(右)基于时间尺度和误差距离的 RPE

参考文献。

[1]刘志强，刘志强，A. J. Davison，“DTAM:密集跟踪和实时映射”，2011 年国际计算机视觉会议，2011:IEEE, pp. 2320-2327。

[2]王晓明，王晓明，“LSD-SLAM:大规模直接单目 SLAM”，计算机视觉，2014:Springer, pp. 834-849。

[3]马鹏，朱军，白旸，王超，彭超，“动态环境下的 DSOD: DSO”，IEEE Access, vol. 7, pp. 178300-178309, 2019。

[4]李志刚，李志刚，李志刚，“基于实时单相机的 SLAM”，《IEEE 模式分析与机器智能学报》，vol. 29, no. 10。6, pp. 1052-1067, 2007。

[5]王志刚，Tardós，“基于 rgb-d 相机的开源 slam 系统”，《IEEE 机器人学报》，vol. 33, no. 1。5, pp. 1255-1262, 2017。

[6]余志强，“基于语义的视觉 SLAM:一种面向动态环境的语义 SLAM”，2018 年 IEEE/RSJ 智能机器人与系统国际会议，2018:IEEE, pp. 1168-1174。

[7]张晓明，张晓明，“基于深度卷积的图像分割算法”，《IEEE 图形分析与机器智能学报》，vol. 39, no. 7。12, pp. 2481-2495, 2017。

[8]王晓明，王晓明，王晓明，“基于 gis 的图像识别方法研究”[j]

null 动态场景，”IEEE 机器人与自动化通讯，第 3 卷，第 3 期。4, pp. 4076-4083, 2018。

[9]何凯，王晓明，王晓明，Girshick，“基于 r-cnn 的图像识别”，IEEE 计算机视觉国际会议论文集，2017,pp. 2961-2969。

[10]李安，王超，徐明，陈志，“面向动态环境的 DP-SLAM:基于移动概率的视觉 SLAM”，《信息科学》vol. 56, pp. 128-142, 2021。

[11]吴文，郭亮，高辉，游忠，刘，陈志，“基于几何约束的动态环境下的语义 SLAM 系统”，神经网络计算与应用，vol. 34, no. 11。8, pp. 6011-6026, 2022。

[12]韩红，奚志强，“基于语义分割的动态场景语义 SLAM”，IEEE Access, vol. 8, pp. 43563-43570, 2020。

[13]范勇，张强，唐勇，刘，韩红，“一种基于语义的 SLAM 算法”，《模式识别》，vol. 31, p. 108225, 2022。

[14]王志强，王志强，王志强，Tardós，“一种基于视觉、视觉惯性和多地图碰撞的精确开源库”，《IEEE 机器人学报》，第 37 卷，第 1 期。6, pp. 1874-1890, 2021。

[15]吴彦祖，吴彦祖，吴彦祖。罗，还有 Girshick。“Detectron2。”
https://github.com/facebookresearch/detectron2
(2022)访问。