基于地标树地图 Z_∞算法和全向摄像机的高效导航

摘要　基于地图的导航是任何移动机器人的关键任务。在许多平台上，这个问题是通过应用基于度量网格图的同步定位和映射（SLAM）来解决的。这种解决方案在资源充足、工作空间有限的机器人上很好地工作。在无边界工作空间中操作的负载有限的平台通常没有足够的资源来保持度量世界的表示。然而，许多应用要求机器人能够在不同的操作区域之间自主导航。在这项工作中，地标树地图（lt-map）是一个资源高效的拓扑地图概念，首次应用于配备全方位摄像机的移动机器人平台。它使机器人能够有效地使在线获取的地图适应可用的内存。在地图采集和导航过程中，运动由-算法估计。这两种方法都基于相似的概念，从而产生了互惠互利的结果。基于 LT 地图的高效导航策略使机器人能够可靠地跟踪先前记录的路径。在室内和室外环境下，对移动机器人进行了评价。实验证明了该方法的可行性，并表明对地图进行修剪只会使其轨迹平滑，这是预期和期望的行为。

一、动机

导航是一项非常复杂的任务，通常需要移动机器人的大部分资源。一般来说，机器人平台在探索过程中会构建度量图，并将自己定位在这些图中。文献[1]中存在大量的方法来解释这种所谓的同时定位和映射（SLAM）概念。它们在组织地图和执行本地化的方式上有所不同，但一般来说，它们共同依赖于环境的度量表示。虽然这种度量图简化了许多计算步骤，例如精确规划新的轨迹，但它需要大量的内存来表示环境。

许多应用需要资源有限的机器人来覆盖远距离，这些机器人连接不同的工作空间：一个微型飞行器（MAV），从救援队开始，朝特定方向飞行，以检测需要帮助的人；或者一个外国星球上的漫游者，在分析后一段时间内必须返回基站。收集的探针。这种应用不需要对环境进行完整的度量表示，而是需要在运行和资源效率高的位置绘制本地度量图。

连接这些工作区并确保机器人能够在它们之间可靠切换的路线图。自适应树数据结构，如四叉树、八叉树或 K-D 树[2]，如果空间使用不平衡，或重新平衡的地图维护成本高，则会产生计算开销。因此，它们也不足以代表这种伸展的通道。

与这些方法相比，生物系统，如昆虫、动物或人类，似乎并不依赖于公制地图来导航[3]。一般来说，它们甚至不具备精确的公制导航感官。他们的行为表明，拓扑导航基于所谓的认知地图，在那里视觉感知通常起着最重要的作用。受这些见解的启发，并以类似的健壮性和效率为目标，我们介绍了 Landmark Tree Map（LT Map），并在模拟中说明了它的功能[4]。它代表了一个拓扑图，它依赖于测角仪的测量，而没有可用的公制距离信息。该算法设计为具有高度可扩展性，并动态适应所提供的内存。第三节详细讨论了标志树的层次结构及其优点。

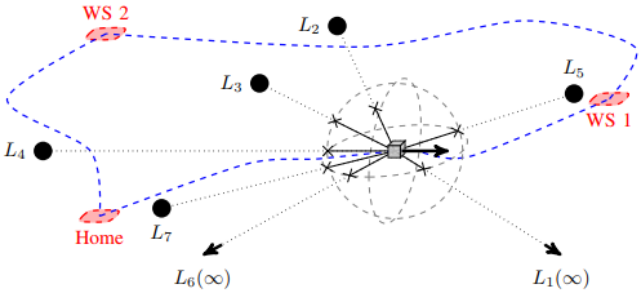


图 1 这幅图描述了我们在这项工作中解决的问题。以灰色盒子为代表的移动机器人需要在不同的工作区之间移动，红色区域表示。对于这些区域，可能需要高分辨率的度量图，但某些移动机器人平台上的有限资源可能会妨碍完整操作空间的完整度量表示。我们提出了一个路线图导航，它能有效地适应可用的内存，并允许移动机器人在不同的工作空间之间移动，仅仅基于指向地标的周围球体上的十字表示的方位测量 L

在这项工作中，我们实现了一个基于移动平台的 LT 地图导航。自主导航仅基于全向摄像头的测量，并能够在工作区之间进行有效的切换，如图 1 所示。使用算法计算运动，该算法允许有效和稳健的估计[5]。由于相似的数据聚类到近距离的地标中，因此 LT 地图和-算法相互受益。我们证明，简单的控制策略足以实现路径跟踪所需的行为。我们还将讨论处理传感器噪声测量所需的修改，以及在内存不足时地图修剪的效果。

本文的其余部分结构如下。首先，我们在第三节和第四节中描述了 LT 地图和概念。这些方法的组合、它们在折反射照相机测量中的应用以及所实施的导航策略在第五节中进行了说明。最后，我们根据室内和室外实验的结果评估了第六节中的方法。

二、相关工作

在许多移动平台上，实现了传统的度量 SLAM 算法，允许直接更新地图和轨迹规划。例如，PTAM 算法[6]可实时实现单眼相机的并行跟踪和映射。它是为增强现实工作空间而设计的，并创建了一个相当密集的环境地图。为了在资源有限、能够覆盖长距离的系统上使用这种方法，地图可以作为滚动地图来实现[7]。这样就可以提供精确的局部定位，但一旦超出地图的限制，全局姿态的固有漂移可能会阻止机器人找到其原点。对于许多应用程序来说，返回到某个位置至关重要，因此，在内存有限的情况下，应优先选择不同位置的全局连接，而不是本地信息。

拓扑图比基于网格的度量图具有更好的空间比例，但是缺少的几何信息也存在一些缺点，例如，它阻止了快捷方式的计算。因此，大多数拓扑方法仍然包含并使用度量信息[8]。另一种流行的方法是使用子映射，其中几个小的度量映射通过拓扑图[9]，[10]连接。这种混合方法结合了两种范例的优点。在我们的工作中，我们关注如何在没有任何度量信息的情况下连接这些子地图。我们仅仅依赖于连接不同工作空间的路径的纯方位测量。路径被存储为拓扑路线图，它提供的导航灵活性不如基于网格的表示法，但它提供的开销最小，因此是存储所需信息的最有效方法。

文献中有几种方法使用全向摄像机进行拓扑导航。温特斯等人引入一个拓扑图，它在关键位置触发一个可视化的服务机制，如狭窄通道[11]。Lui 和 Jarvis 展示了一个装有两个全向摄像头的移动机器人[12]。该系统基于未压缩图像的 Haar 小波特征建立了一个度量拓扑图。公制姿态是由一个三自由度（DOF）视觉里程测量模块来计算的，该模块使用来自摄像头的立体通信，在 GPU 上进行处理。在[13]中，Murillo 等人提出一种拓扑和度量局部化的结合，其中使用一组大描述符的金字塔核来检测粗糙位置，然后使用一维三焦点张量来计算精确的度量局部化估计。像在这些方法中一样，拓扑图一般用于环路闭合检测、粗略定位或触发某些行为。这些位置由从图像中得到的特征信号定义，如直方图[14]、线性 PCA[15]或 HAAR 小波系数[16]。在其他实现中，将提取标志，并使用一组特征描述符来定义特定位置，如单词包模型[17]。然而，这些方法都不允许在内存不足的情况下轻松地修剪信息。识别地图中的信息对于本地化来说是多余的或不重要的，这在计算上是昂贵的，并且需要对整个地图进行繁琐的评估。因此，通常情况下，如果这些信息可用，则会丢弃整个节点（位置），或者在度量空间中对地图进行同等细化。

我们在这项工作中采用的机器人控制概念与[18]相似，但我们没有估计基本矩阵，而是利用基于鲁棒的姿态估计来实现一个有效但可靠的速度控制范式。

三、地标树图

lt-map 算法的设计目的是使路线图能够有效地在线扩展到系统上的可用内存[4]。路线的非公制表示法只需要测角仪的测量值，而不需要任何距离信息。这些地标以树形结构存储，提供全局和局部地标的层次结构。这是通过用测量方位角扩展每个地标描述符来实现的。该角度必须补偿机器人的旋转运动，例如使用罗盘，以便仅包含平移信息。这样，遥远的地貌，

如地平线上的地标，即使经过长的轨迹，也包含相同的扩展描述符。由于运动视差的存在，靠近传感器的局部地标已经以较小的平移运动改变了它们的视角。纯旋转运动不会更改描述符角度元组，因此不会影响映射。在下文中，我们将这些元组表示为界标，其中=（dis）是描述符向量，而 ϕ 是包含方位角和高程的向量，在补偿旋转运动后测量界标所依据的方位角和高程。一个视图框架被定义为从某个位置感知的一组标志，因此描述了该位置。
在生成地图的过程中，即首次遍历路径时，将为每个新测量计算地标。每当两个相应的标志之间的加权平均角度超过指定的阈值时，将由当前的一组标志定义新的视框。两组地标之间的相似性度量是用一个伪 Huber 成本计算的。这样

$$\delta_t = \frac{1}{N}\sum_{i=1}^{N} 2b^2 \left(\sqrt{1 + \left(\frac{1 - \boldsymbol{l}_i^T \boldsymbol{l}_i'}{b} \right)^2} - 1 \right), \quad (1)$$

其中，land lde 注意指向不同位置相同地标的单位向量。单位向量可以很容易地从 ϕ 计算出来。此成本函数包含一个控制参数，用于四次加权较小的误差，但与坡度 2b 线性加权较大的误差。与前一个视图框架相比，在最新的分支处，只有新的标志添加为新的叶。因此，持久的标志位于上节点，靠近根，并且包含翻译不变的信息，我们在下面将其表示为全局信息。所谓的局部的、依赖于翻译的信息，是由较低的节点上的地标表示的，一直到树叶。从全局信息到局部信息的逐步转换可以满足环境的要求，并由不同的层隐式实现。在这一点上，我们要强调的是，在实践中，地标不仅是根据它们的距离来排列的，而且是根据它们的稳定性来排列的。靠近根部的节点只包含真正稳定的标志，而较不稳定的节点（无法连续跟踪）则位于靠近树叶的位置。树中从根到叶的每个完整分支表示一个视图框架。这种树的结构如图 2 所示。有关更详细的解释，请参阅[4]

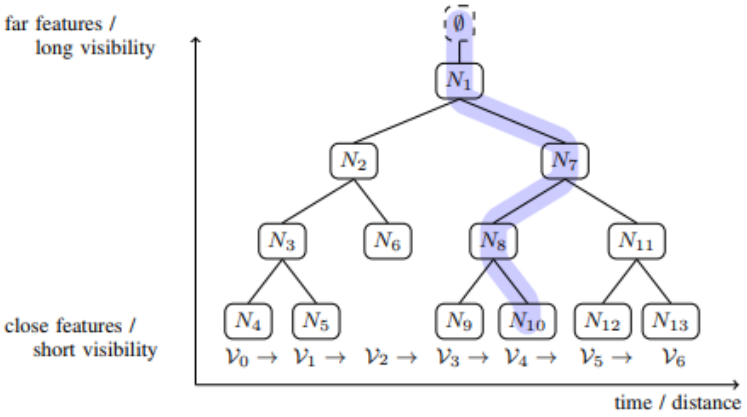


图 2 地标树的两个维度表示移动距离（时间）和地标到机器人的距离。树的每个分支都是一个视图框（如的蓝色突出显示）。沿着分支的节点包含在特定位置获取的标志。

四、算法 $Z\infty$

LT 地图需要旋转对齐的地标来构建地图。该算法设计用于远程导航，因此也适用于室外场景。该算法是一种有效的基于特征的运动估计技术，在这种环境中也有很大的好处，而且只依赖测角仪的测量[5]，[19]。它使用 ransac[20]框架将远距离的、平移不变的地标与局部的、平移可变的地标分开，并拒绝不适合普通刚性运动模型的动态对象上的特征。这样，旋转估计与平移估计分开，平移估计允许对两个运动分量应用有效的闭合形式解，如[5]所述。然而，这里提出的翻译估计是基于投影相机。为了根据全方位测量估计平移，我们将当前地标集的以下成本函数最小化：

$$\mathrm{E}(\boldsymbol{t}) \;=\; \sum_{i=0}^{N} w_i \cdot \left(\left(\boldsymbol{l}_i' \times \boldsymbol{l}_i \right)^T \boldsymbol{t} \right)^2 - \lambda \left(\|\boldsymbol{t}\|^2 - 1 \right) , \quad (2)$$

其中 land lde 记录旋转补偿方向向量到参考地标和当前测量。通过拉格朗日乘子强制转换 t 具有单位长度，并且每个地标向量可以通过一个因子加权， 如果应用不同的跟踪精度。更换后 M=×我们得到

$$\begin{aligned} \mathrm{E}(\boldsymbol{t}) \;&=\; \sum_{i=0}^{N} w_i \left(\boldsymbol{m}_i^T \boldsymbol{t} \right)^2 - \lambda \left(\|\boldsymbol{t}\|^2 - 1 \right) \\ &=\; \sum_{i=0}^{N} w_i \boldsymbol{t}^T \boldsymbol{m}_i \boldsymbol{m}_i^T \boldsymbol{t} - \lambda \left(\|\boldsymbol{t}\|^2 - 1 \right) \\ &=\; \boldsymbol{t}^T \left(\sum_{i=0}^{N} w_i \boldsymbol{m}_i \boldsymbol{m}_i^T \right) \boldsymbol{t} - \lambda \left(\|\boldsymbol{t}\|^2 - 1 \right) \quad (3) \end{aligned}$$

替换 m 并设置的导数 e（）到零导致特征向量问题，由定义 *T*

$$\begin{aligned} 0 \;&=\; 2\boldsymbol{M}\boldsymbol{t} - 2\lambda \boldsymbol{t} \\ \lambda \boldsymbol{t} \;&=\; \boldsymbol{M}\boldsymbol{t} \, . \qquad (4) \end{aligned}$$

因此，到符号为止，t 是对应于矩阵 m 的最小特征值的特征向量，可以通过奇异值分解（SVD）计算，从而

$$\tilde{\boldsymbol{t}} \;=\; \boldsymbol{V}_{\boldsymbol{M}_{:,3}} \quad \text{with} \quad \mathrm{SVD}\left(\boldsymbol{M} \right) = \boldsymbol{U}_{\boldsymbol{M}} \boldsymbol{\Sigma}_{\boldsymbol{M}} \boldsymbol{V}_{\boldsymbol{M}}^T , \quad (5)$$

而 vm 中的下标是指矩阵的最后一列。

$$\boldsymbol{t} = \begin{cases} -\tilde{\boldsymbol{t}} & \text{if } \sum_{i=0}^{N} \left(\boldsymbol{l}_i' - \boldsymbol{l}_i \right)^T \tilde{\boldsymbol{t}} < 0 \\ \tilde{\boldsymbol{t}} & \text{else} \end{cases} . \qquad (6)$$

为了确保正确的旋转估计，我们为 ransac 选择的一组界标添加了另一个约束条件，作为最佳的内部集。我们拒绝所有相邻两个地标之间的最大角度大于 180 的集合。这就防止了平移运动被误认为是旋转。我们还评估了使用视觉罗盘进行旋转估计。然而，众所周知，在不平衡的环境中，视觉罗盘只能获得较差的结果[21]。当一个方向上的物体比另一个方向上的物体近得多时，视觉罗盘会失效，并且在基于模型的-算法旋转估计中表现得更好。

然而，为了应用-算法，需要识别远平移不变特征。一方面，通过提取存储在树的上部节点中的地标，LT 地图可以方便地访问这些地标。这为有效的旋转估计提供了一组初始标志，并根据 RANSAC 的要求减少了随机采样步骤的数量。另一方面，由-算法返回的一组平移不变特征可以用来寻找所有视框上节点的对应关系。由此产生的匹配允许识别无法跟踪某些远距离标志的循环闭包或视框，因此，这些标志丢失了。如果检测到这些间隙，可以通过将相应的标志移动到树的较高级别的公共节点来填充。最后，两种算法，即 LT 地图和算法，将地标分为远距地标和局部地标。lt map 逐步执行此操作，而-algorithm 提供二进制拆分。

五、使用全向摄像头的基于 LT-MAP 的导航

首先，我们想激发我们对传感器的选择。所提出的导航框架中使用的所有算法都是基于测角仪的测量。我们致力于开发一种能在资源有限的平台上运行的高效算法。为此，传感器也应小巧轻便，功耗低。相机符合这些标准，可以实现大孔径角度。增加视场，同时保持像素分辨率不变，可以看到精度在 100[19]左右达到最佳。尽管如此，我们还是选择使用反射照相机，如[22]所示，因为大视场显著提高了鲁棒性。尤其是在路线图导航算法中，路径在两个

方向上都是横穿的，因此不仅要向前看，还要向后看，以使其他传感器的方向保持在行驶方向上，这一点至关重要。此外，全向摄像机可以最大限度地降低缺少用于旋转估计的远距特征的风险。这样也可以在狭窄的通道和墙壁前面操作。

我们在全方位图像上评估了不同的全球特征追踪器，asift[23]、sift[24]、surf[25]和 brick[26]。虽然筛选、冲浪和轻快之间没有太大的差异，但最可靠的跟踪是使用 asift。然而，它需要最多的处理时间，因此，我们选择使用快速，因为它的高效率。实验还表明，如果全向图像在处理前被解压为全景图像，那么所有跟踪器的性能都会得到改善。这使得我们可以使用 U-BRICK（算法的旋转变体）来提高性能。

A.路线图学习

如果一个机器人第一次探索一个环境，路径将被学习为 It-map。机器人的起点和初始方向定义了原点。用-算法估计出代表纯平移运动的光流矢量，并根据式 1 的引入，确定是否需要添加新的视框。如果触发了新的视框，则当前帧的功能将与上一个图像的功能相匹配，以便在将其添加为新的视框之前拒绝不稳定的标志。

在 LT 图中，树结构的一个优点是，删除本地或不稳定信息的成本很低。一旦达到内存限制，树的最低级别将被简单地截断并释放内存。因此，机器人不必事先知道它将获得的路径的长度。它可以简单地从一个很小的值开始，并在需要时立即对树进行修剪。这样，地图以非度量方式动态地将其空间分辨率适应可用内存。在完成地图的情况下，不会丢弃第一个和最后一个分支节点中的地标，因为我们希望有更高的精度来找到目标或开始位置。其想法是，在路径导航过程中，精确地保持在轨道上的精度并不是那么重要。但是，我们希望在找到目标位置时具有较高的准确性，因为它可能是理想工作空间中的关键位置。此外，如果在当前工作区内无法准确知道起始位置，则起始点应提供尽可能多的信息，以便可靠地钩住轨迹。因此，第一个和最后一个分支中节点的标志不会被丢弃，而是向上移动到父节点中，如图 3 所示。如果在采集过程中修剪地图，也会修剪最后一个分支，因为它不需要更高的精度。修剪后，我们检查所有树枝是否包含足够的地标，以便可靠的归位，否则这些树枝的新叶会被删除，树会被压缩。

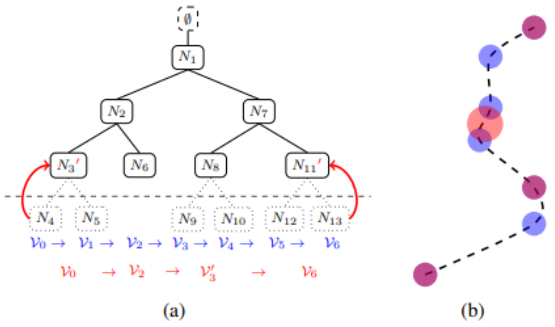


图 3 如果必须修剪树以释放一些空间，则会删除最低级别的节点，如左图所示。如果外部分支表示路径的端点，则将标志移动到父节点中，如红色箭头所示，以便和. 右图绘制了相应的示例路径，图框位置表示为圆。可用的局部信息越少，圆圈就越大，它描述了一个框架的标题区域。蓝色圆圈对应于原始视图框架，红色圆圈对应于修剪过的树。

B.路线图

由一个 LT 映射定义的路由可以简单地从一个图幅移动到另一个图幅，从而从开始到目标图幅依次提取每个分支的标志。这些地标集用作姿态估计的参考[1]，通过再次应用-算法计算。因此，Landmark 集合中的异常值由 ransac 框架过滤。产生的平移矢量 t 被用作运动控制的输入。为了抑制错误的平移估计，我们对最近获取的向量 t 的方位角和仰角应用了中值滤波器。正确设置 360°切口的中值旋转计算的圆形问题解决如下：

$$\hat{\boldsymbol{\phi}} = \operatorname{med}\left(\left\{\operatorname{mod}\left(\boldsymbol{\phi}_k - \bar{\boldsymbol{\phi}} + \pi, 2\pi\right)\right\}_{k=0}^{K}\right) + \bar{\boldsymbol{\phi}} - \pi \quad (7)$$

带，其中 ϕ=（φa，φe）t 由矢量 t 的方位角和仰角组成，中位数和模函数按元素应用。最后，二维航向矢量的中值可以通过

$$\hat{\boldsymbol{t}} = \left(\cos\left(\hat{\phi}_a\right),\ \sin\left(\hat{\phi}_a\right),\ \sin\left(\hat{\phi}_e\right)\right)^T . \quad (8)$$

假设更多的测量结果产生更高的置信度，我们通过中值滤波器中的元素数量 N 和用于估计的标志物数量来缩放向量 t 从而

$$\boldsymbol{v} = \frac{K}{K_{\max}} \min\left(\frac{N}{N_{\max}}, 1\right) \hat{\boldsymbol{t}}, \quad (9)$$

其中表示足够进行可靠运动估计的指定标志数量，是根据传感器、特征跟踪器和环境选择的最大滤波器尺寸的经验值。通过简单的 PID 控制器，利用矢量 V 控制机器人的速度。它被解释为比例速度，乘以一个恒定的最大速度。因此，如果机器人对方向估计有信心，它会不断地调整方向以跟随路径，移动速度更快，如果不确定性增加，移动速度也会变慢。这种行为允许获得更多的测量结果，例如，在环境条件较差的情况下。

在两种情况下会触发切换到下一个视图框。根据等式 1，平移流矢量足够小，这意味着机器人靠近参考视框，或者估计的方向矢量变化超过 90°，这意味着机器人刚刚通过视框，想要向后移动。如果连续帧之间的方向变化超过 90，则第二个约束可能存在问题。机器人在切换到第一个视框后，会立即在其背面找到第二个视框，这将触发另一个视框开关。因此，在切换视框后，需要一些时间才能转到新的方向。这可以很容易地通过在每个视框开关后短时间超时或等待直到方向匹配来实现。

六、实验

作为我们实验的平台，我们使用 Pioneer-3DX 机器人，如图 4 所示。该平台配备了不同的地面真值传感器和标记。在我们的室内实验中，我们依靠最先进的跟踪系统 1，在室外，我们使用测速仪[2]的测量值，它提供了最精确的测量值。导航命令通过串行接口发送到机器人电机控制板的速度接口。作为中间件，对于不同模块之间的通信，我们使用 ROS[3]。所有的处理都是在板上完成的，可以通过 WiFi 进行监控。为了在紧急情况下或手动操作时停止机器人，可以通过游戏板进行控制。勘探轨迹采用人工控制。

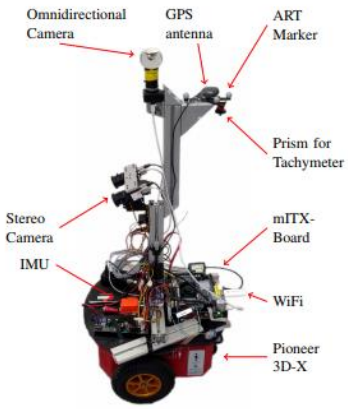
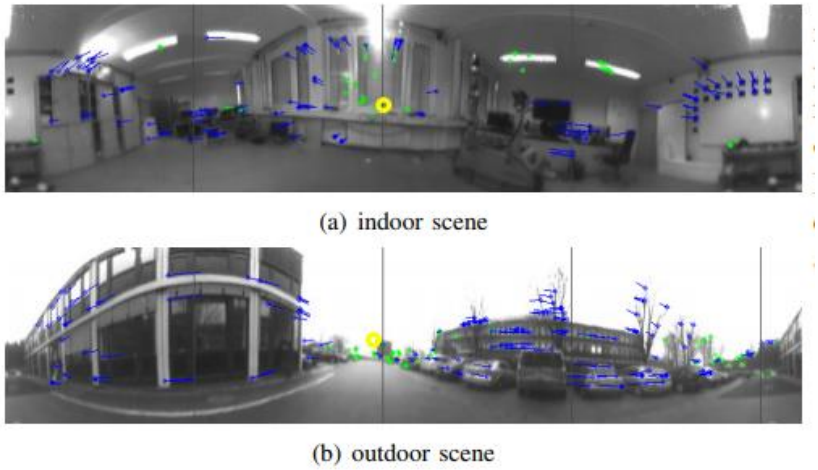


图 4 在我们的实验中，我们使用了先锋 3DX 作为移动平台。机器人装备了 GPS 天线、测速仪的参考棱镜和地面真实性的艺术标记。通过融合 IMU 测量值和立体里程计（如[27]所述）计算出进一步的参考值，而半全局匹配（SGM）立体处理运行在 Xilinx Spartan-6 FPGA 上。使用 Kontron KTQM67/MITX 嵌入式主板作为处理平台，配备 Intel Core i7 处理器、8GB DDR3 和 SSD 存储设备。所有的处理都是在船上完成的，无线网络用来监控主机上的数据。

我们在 5Hz 下采集和处理图像，分辨率为 474×474 像素，也可以在更小的平台上实现。然后将图像解压成两个维度分辨率均为 0.5/px 的全景图，得到 720×172 像素的可用图像。我们不断调整快速探测器，在室内场景中找到约 400 个地标，在室外实验中找到 500 个，得出了约 200-300 个有效匹配。未经压缩的全向图像被扩展为包含一个小的重叠，它补偿了 U-Brick 的描述符大小，并防止了特征在边界处丢失。图 5 显示了由折反射照相机获取的两个示例图像。



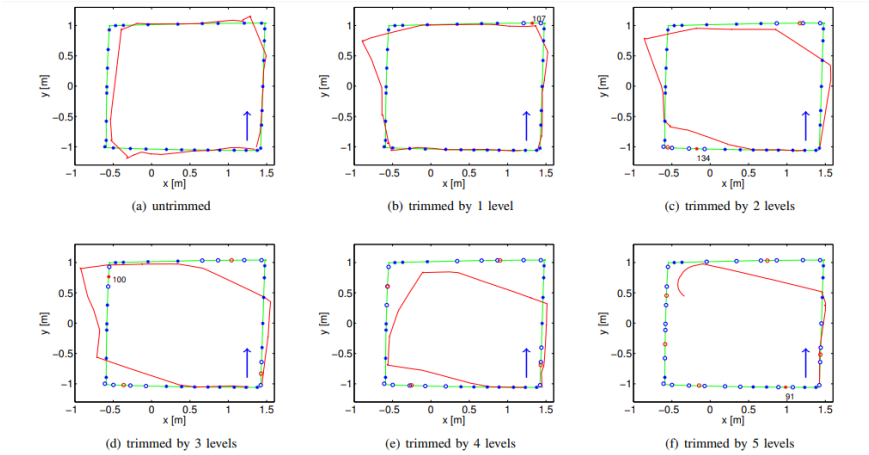
(a) indoor scene

(b) outdoor scene

图 5 这些截图显示了两个示例，其中室内和室外场景中的平移光流向量（蓝色）、用于旋转估计的标志（绿色）和估计的平移方向（黄色）。灰色线以 90 步表示角度。

A.室内评价

我们在实验室对所提出的导航策略进行了首次评估，并对机器人进行了控制



(a) untrimmed (b) trimmed by 1 level (c) trimmed by 2 levels

(d) trimmed by 3 levels (e) trimmed by 4 levels (f) trimmed by 5 levels

图 6 这些图显示了完整的和修剪过的树木的室内实验的轨迹。绿线表示映射轨迹，蓝色星号表示获取视框的位置，红色轨迹表示机器人跟随的路径。蓝色箭头表示行驶方向和起点。蓝色圆圈是修剪过的图幅，红色标记表示剩余的图幅分支，而它们的位置被选作修剪过的图幅的重心。作为标记，我们使用一个红色的星号，如果足够的地标仍然可以找到，和一个红色的圆圈否则。作为这个决定的起点，我们在实验中选择了 50 个地标。红色星号旁边的数字表示剩余地标的数量。

用于构建 LT 地图，获取 34 个视图框。在这个实验中，我们简化了控制模式，即机器人评估图像，转换到新的方向，然后直线移动 15 厘米，然后再次停止获取新的图像。这样，控制器就不会对结果进行平滑处理，从而可以收集到对实际运动估计的更多洞察。图 6 显示了不同修剪级别的路径跟踪性能。该树被逐步修剪一级，由此产生的轨迹证明，对于每一个修剪步骤，精度降低。机器人开始切割边缘，但即使只有八分之四的树水平，也能可靠地找到目标。只剩下三层，机器人在一段时间后就会迷路。它找不到足够的功能来可靠地估计导航方向并启动搜索模式[4]，如图 6（f）所示。

表 I 显示了生成的 LT 映射的一些统计信息，如视框数、元素总数和被修剪的树占用的内存百分比。几乎有一半的内存使用率，可以可靠地达到目标。此外，通过计算机器人假定到达视框的位置与视框采集的真实位置之间的平均误差，评估了路径跟踪的性能。
在最后一列中，我们将每个测量位置的平均距离与映射轨迹上最近的点进行比较。这两个值都随着修剪级别的数量而增加，这证明由于本地信息较少，精度降低。在这一点上，我们要强调的是，该算法并不是为了高精度地再现一条路径，而是为了可靠地达到仅使用有限内存的目标。

TABLE I

MAPPING STATISTICS FOR THE INDOOR SCENARIO

Height (trim)	View-frames	Land-marks	Memory usage [%]	Viewframe error [m]	Path error [m]
8 (0)	34	10171	100	0.057	0.055
7 (-1)	33	9732	95.7	0.052	0.057
6 (-2)	28	8322	81.8	0.133	0.099
5 (-3)	23	7193	70.7	0.145	0.108
4 (-4)	18	5832	57.3	0.208	0.1127
3 (-5)	11	3551	34.9	-	0.1547

B.室外评价
在我们的室外实验中，除了第 V-B 节所述的速度控制器的应用以及使用测速仪作为地面实况而非艺术跟踪系统之外，我们使用了与室内实验相同的设置。机器人被手动沿着一条记录为“LT 地图”的路径操纵，之后，机器人被命令使用三个不同修剪的地图返回原点。图 7 显示了路径的轨迹，通过修剪树，路径变得更平滑。一旦我们修剪了五个层次，机器人就无法找到足够的地标来可靠地计算运动方向，因此开始执行搜索模式。

TABLE II

MAPPING STATISTICS FOR THE OUTDOOR SCENARIO

Height (trim)	View-frames	Land-marks	Memory usage [%]	Viewframe error [m]	Path error [m]
9 (0)	24	8493	100	0.777	0.551
7 (-2)	23	7862	92.6	0.615	0.499
5 (-4)	18	6229	73.3	0.872	0.408

在表二中，我们再次列出了获得的 LT 图的一些统计数据和获得的精度。有趣的是，通过修剪树，路径精度略有提高，这可以解释为速度控制器的配置相当保守，因此机器人对运动变化的反应相当缓慢。

与室内实验相比，我们获得的视窗较少，这可以用地标的平均距离较大来解释。因此，相似性度量以较大的间隔触发视框，从而自动适应场景中可用的信息。δ_T

图 8 显示了图 7（c）所示的运行时间内计算的方位角方向。角度用机器人框架表示，控制器直接使用。正如人们所看到的，当机器人转向新的方向时，到下一个视框的角度在切换后收敛到零。如果角度超过±90，机器人假定已经通过了视框，并切换到下一个。视框开关的另一个标准是相似性度量是否低于指定的阈值，如绿色星号所示。正如第 V-B 节所激励的那样，5 秒的超时确保机器人可以在角度标准触发下一个开关之前对准新的视框。这由黑色水平线中的虚线间隙表示。根据机器人的旋转速度选择超时，以允许 180°旋。

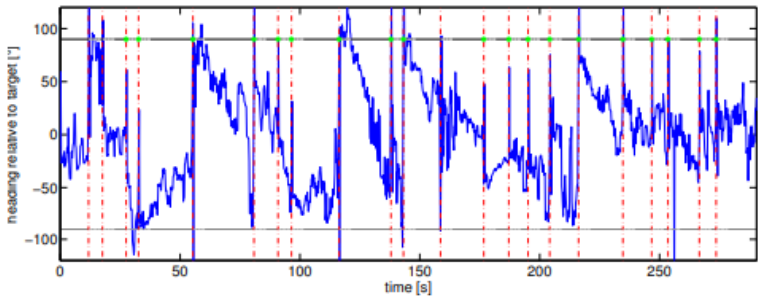


图 8 该图显示了图 7（c）所示轨迹的计算方位角（蓝线）。红色竖线表示视框开关，黑色横线表示视框切换超时。在由两条闭合虚线表示的时间段内，任何由角度约束触发的视框切换都将被阻止。

七.结论与未来工作

在这项工作中，我们提出了一种高效和可扩展的路线图方法，它可以应用于资源有限的平台，以连接远程工作空间。它允许在运行时有效地将映射的分辨率调整到所提供的内存。不需要度量信息-显示的导航仅依赖于摄像机的方位测量。将-算法应用于全方位测量，对运动进行了估计。由于基本概念相似，这两种方法都受益于彼此。Z 无穷大

在我们的实验中，我们已经证明了导航算法可以在小型机器人系统上实现，只需要一个低帧速率的低分辨率全向摄像头。通过修剪 LT 映射树以释放一些内存，本地信息将被丢弃。这使得轨迹更加平滑，与局部障碍物检测相结合，完全满足设计要求。通过直接评估光流的碰撞时间，可以很容易地实现这种避障[5]。

作为下一步，我们将引入一个度量，它允许我们确定我们可以修剪树的高度，并且仍然有足够高的概率成功地沿着路径走。我们还希望将该算法应用于 MAV，这使得我们能够在三维空间中验证该方法。此外，我们认为目前这种方法的瓶颈是特征跟踪器，我们希望改进它，使其能够在更长的距离内跟踪地标。