

## Technical White Paper

## 自主移动机器人：借助 AM69A 实现智能安全导航



Do-Kyong Kwon

## 摘要

移动机器人的自主导航需要人工智能 (AI) 和传统计算机视觉的高性能计算资源，以便利用多个传感器执行各种任务。本文介绍了移动机器人自主导航的主要任务和相关技术。本文还表明，AM69A 处理器是 AM6xA 可扩展嵌入式处理器系列中的高端器件，针对自主导航而精心设计，具有出色的性能和开发便利性。

## 内容

1 引言.....	2
2 定位和建图.....	3
2.1 同步定位与建图.....	3
2.2 图形 SLAM.....	4
2.3 定位.....	5
3 周围环境感知.....	6
4 规划路线.....	7
5 总结.....	8

## 插图清单

图 1-1. 自主导航的三项主要任务.....	2
图 1-2. AM69A 简化方框图.....	3
图 2-1. 机器人姿态和观测结果图.....	4
图 2-2. 图形 SLAM 流程.....	4
图 2-3. 使用预构建地图的定位过程.....	5
图 4-1. AM69A 自主导航系统的示例数据流.....	7

## 表格清单

表 2-1. 图形 SLAM 中的各项技术.....	6
----------------------------	---

## 1 引言

自主移动机器人 (AMR) 有助于提高制造、仓储、物流等领域的生产力和运营效率。例如，AMR 可在仓库和物流中心搬运包裹，用真空吸尘器清洁地板，以及在餐厅供应食品和饮料。早期 AMR 通常在人类无法进入的工作区内运行，并在地面的车道和 [AprilTag](#) 引导下沿着预定义路径导航。因此，早期 AMR 不需要大量的传感器和严格的功能安全特性。这种遵循预定义路径的机器人也称为自动导航车 (AGV)。与之不同的是，最近的 AMR 配备了先进的传感器，可在与人类共用的工作区内运行，并在环境中自由而安全地导航，从而在指定位置执行分配的任务，尽可能减少了人工干预。

如图 1-1 所示，AMR 要实现安全自主导航，需要完成三项主要任务：定位、感知和规划。首先，移动机器人必须知道自己在工作区中的位置。精确定位是实现自主导航的最低要求。定位后，移动机器人必须感知移动物体（包括人类和其他运行中的机器人）的动态环境。接下来，机器人必须规划一条通往目的地的路径，以相应地控制自己，避免出现导致安全问题的情况。本文讨论了这些任务的实现原理和固有挑战，同时重点介绍了定位和建图。

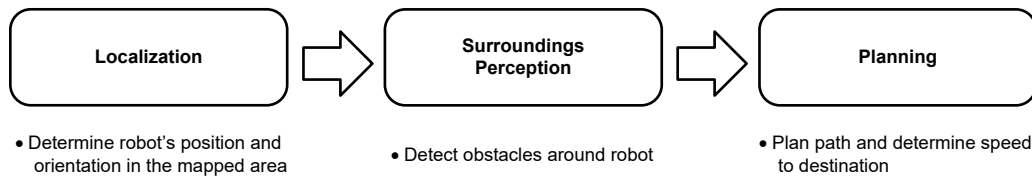


图 1-1. 自主导航的三项主要任务

[AM69A 处理器](#) 是一款异构微处理器，专为采用传统分析和 AI 算法的高性能计算应用而构建。关键元件包括八核 Arm® Cortex® A72、视觉处理加速器 (VPAC)、具有矩阵乘法加速器 (MMA) 的四核 C7x 数字信号处理器 (DSP)、图形处理单元 (GPU)、视频编解码器、隔离式 MCU 岛等。VPAC 具有多个加速器，包括视觉成像子系统 (VISS)，即图像信号处理器 (ISP)、镜头失真校正 (LDC) 和多标量 (MSC)。图 1-2 所示为 AM69A 简化方框图。更多详细信息，请参阅 [AM69x 处理器，器件修订版本 1.0](#) 数据表。[在智能摄像头应用中使用 AM69A 实现高级 AI 视觉处理](#) 技术白皮书中介绍了 AM69A 的多摄像头 AI 用例。该白皮书解释了 AM69A 为何是同时运行所有三项任务来实现自主导航的理想处理器。

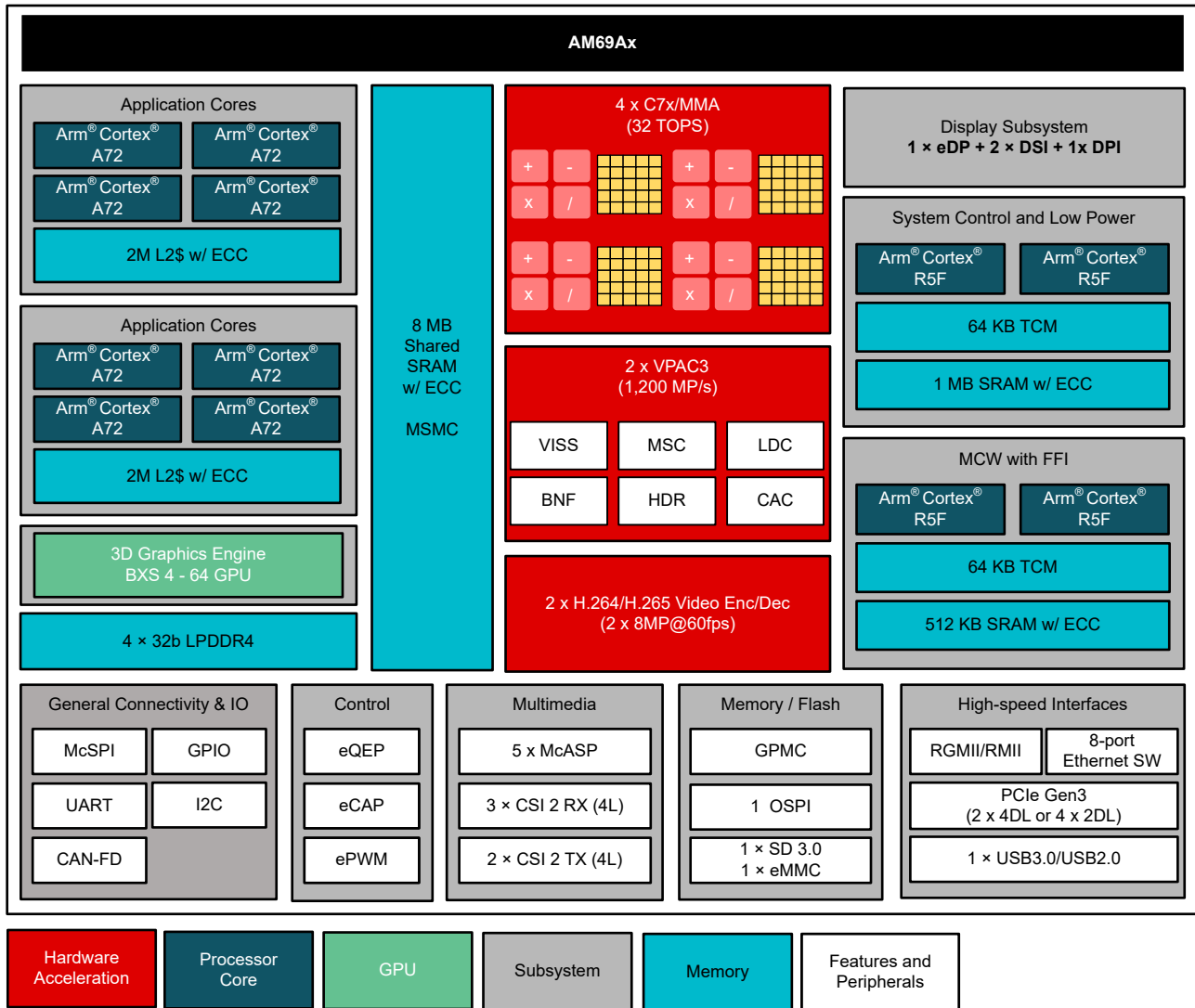


图 1-2. AM69A 简化方框图

## 2 定位和建图

通过在地面上绘制车道和 **AprilTag**，可以轻松实现机器人沿固定路径导航的过程。例如，通过定义机器人在检测和识别每个标签时必须执行的操作，可以对机器人进行编程，使其从一个标签行进到另一个标签（例如前进、右转并移动、提起包装并前进等）。在这些情况下，只要能够按指示精确控制机器人，机器人就不需要进行定位。

定位是移动机器人在动态导航时计算其在地图区域中的位置和方向的过程。大多数情况下，地图也由移动机器人预先构建，并以机器人可重复用于定位的格式保存。移动机器人探索未知环境，并在构建地图的同时更新其位置。因此，该建图过程称为“同步定位与建图”（SLAM）。

### 2.1 同步定位与建图

根据所使用的技术，SLAM 算法分为三类：基于滤波器的 SLAM、图形 SLAM 和基于深度学习 (DL) 的 SLAM。基于滤波器的 SLAM 将该问题视为状态估计问题。状态由机器人的姿态和地图组成，由滤波器根据机器人探索过程中的测量值进行迭代更新。基于 DL 的 SLAM 通过用 DL 网络替换整个端到端过程来解决问题。只有图形 SLAM 中的子任务可以用 DL 网络替代。然而，将此类算法归类为图形 SLAM 是合理的。图形 SLAM 是目前较为先进的算法。基于端到端 DL 的 SLAM 最近确实取得了可喜的成果，但还不成熟，而基于滤波器的 SLAM 总体上比图形 SLAM 的表现更差。因此，本文的讨论仅限于图形 SLAM。SLAM 算法可以根据主传感器进一步分类，例如视觉

SLAM、激光雷达 (LiDAR) SLAM 等。惯性测量单元 (IMU) 或惯性导航系统 (INS) 通常与主传感器一起使用，以提高姿态估计的准确度。

## 2.2 图形 SLAM

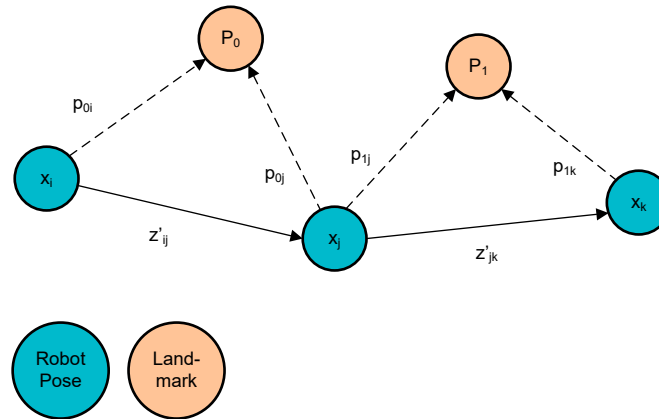


图 2-1. 机器人姿态和观测结果图

图形 SLAM 将 SLAM 问题视为图形结构的非线性优化，其中节点表示机器人的姿态和不同时间的观测结果，边表示姿态之间的约束条件。图 2-1 展示了一个玩具示例，说明了如何借助机器人移动时的姿态和观测结果来构建图形。 $x_i$ 、 $x_j$  和  $x_k$  是机器人的姿态， $P_0$  和  $P_1$  是摄像头观测到的地标、物体或点。此处假设通过  $x_i$  和  $x_j$  观测  $P_0$ ，通过  $x_j$  和  $x_k$  观测  $P_1$ 。 $p_{nt}$  (其中  $n=0, 1, t=i, j, k$ ) 定义为通过  $x_t$  观测的  $P_n$  的位置。图形 SLAM 的目标是确定机器人姿态之间的变换，即  $z'_{ij}$  和  $z'_{jk}$ ，使由不同姿态产生的相同观测结果尽可能地重叠。因此，需要确定  $x_t$  (其中  $t=i, j, k$ )，使  $(p_{0i} - p_{0j})^2 + (p_{1j} - p_{1k})^2$  最小。

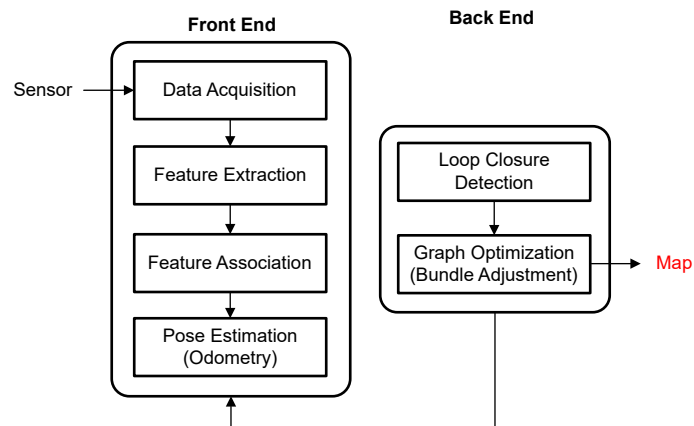


图 2-2. 图形 SLAM 流程

图 2-2 显示了图形 SLAM 流程，主要由两部分组成，即前端和后端。前端处理输入传感器数据，以估计移动机器人的姿态以及机器人周围的明显特征。前端处理包括以下步骤：

- 数据采集：从摄像头和激光雷达等传感器连续捕获数据。捕获的传感器数据在此处进行预处理。例如，在数据采集期间执行 ISP、镜头失真校正、立体图像校正和激光雷达点云运动补偿等操作。在固定时间间隔内捕获的数据构成一帧。
- 特征提取：从每一帧中提取独特的描述性特征。这些特征可以是图像中的关键点 (视觉 SLAM)，也可以是几何特征 (例如边缘、平面以及扫描的 2D 或 3D 点本身) (激光雷达 SLAM)。
- 特征关联：移动机器人将从不同帧中提取的特征进行关联，以确定哪些特征与环境中的相同特征相对应。
- 姿态估计：根据相关特征，移动机器人估计其位置和方向在两帧之间的变化，从而估计移动机器人的新姿态。估计完姿态，就可以使用相关特征来更新地图。

在前端估计的移动机器人姿态是有误差的，并且误差会随着移动机器人的移动而累积，从而导致巨大的漂移误差。后端负责细化估计的姿态并更新地图。此细化包括以下步骤：

- 闭环检测：此过程可确定以前是否访问过当前位置。要通过图形优化减少漂移误差，确定访问过的位置至关重要。
- 图形优化：图形保留在图形 SLAM 中，如图 2-1 所示。目标是更新先前估计的姿态，使从一个姿态观测到的结果（如特征）与从其他姿态观测到的相同结果尽可能重叠。牛顿法、高斯牛顿法和列文伯格-马夸尔特 (LM) 优化等非线性优化技术均用于此用途。

一般来说，在视觉 SLAM 中，创建的地图是一组提取的特征及其位置和描述符；在激光雷达 SLAM 中，创建的地图是一组几何特征或点本身。在许多用例中，地图在经过后处理后还会以占据栅格地图的形式保存，以便于障碍物检测和路径规划。

### 2.3 定位

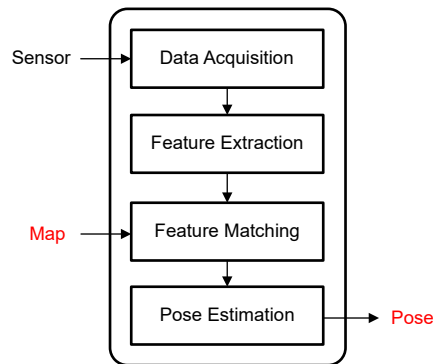


图 2-3. 使用预构建地图的定位过程

图 2-3 显示了地图环境中的定位过程。该过程类似于图 2-2 中的 SLAM 前端。唯一的区别是，一旦从帧中提取特征，就会从地图中搜索相应的特征，而不是从其他帧中搜索相应的特征。找到匹配的特征后，就可以通过多点透视成像 (PnP)、迭代最近点 (ICP) 等算法计算出移动机器人的姿态。

表 2-1 总结了图形 SLAM 和定位的每个步骤中广泛使用的技术。

表 2-1. 图形 SLAM 中的各项技术

	视觉 SLAM	激光雷达 SLAM
特征提取	<ul style="list-style-type: none"> <li>特征检测和描述符：                             <ul style="list-style-type: none"> <li>SIFT、SURF、KAZE、AKAZE、ORB、BRISK</li> <li>基于卷积神经网络 (CNN)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>2D 或 3D 点本身</li> <li>几何特征                             <ul style="list-style-type: none"> <li>边沿</li> <li>平面</li> </ul> </li> </ul>
特征关联	<ul style="list-style-type: none"> <li>特征描述符之间的最小欧几里得距离或汉明距离</li> </ul>	<ul style="list-style-type: none"> <li>2D 或 3D 点、边和平面之间的最小欧几里得距离</li> </ul>
姿态估计	<ul style="list-style-type: none"> <li>直接线性变换 (DLT)</li> <li>LM 优化</li> <li>PnP</li> </ul>	<ul style="list-style-type: none"> <li>扫描匹配                             <ul style="list-style-type: none"> <li>占据栅格匹配</li> <li>ICP</li> <li>正态分布变换 (NDT)</li> </ul> </li> <li>LDT、LM 优化</li> </ul>
闭环检测	<ul style="list-style-type: none"> <li>词袋模型 (BoW)</li> <li>基于外观的全局图像描述符</li> <li>基于 CNN</li> </ul>	<ul style="list-style-type: none"> <li>扫描匹配</li> <li>分段匹配</li> <li>基于 CNN</li> </ul>
图形优化 (捆绑包调整)	<ul style="list-style-type: none"> <li>利用矩阵稀疏性</li> <li>下面是牛顿法、高斯牛顿法、LM 技术</li> </ul>	<ul style="list-style-type: none"> <li>利用矩阵稀疏性</li> <li>下面是牛顿法、高斯牛顿法、LM 技术</li> </ul>

AM69A 嵌入式处理器是 SLAM 和定位的理想选择。八核 A72 为复杂的 SLAM 和定位算法提供了足够的计算能力。可以对 AM69A 快速实现许多开源算法并进行基准测试。此外，图 2-2 和图 2-3 中的功能块（例如特征提取、特征匹配和姿态估计）可以转移到硬件加速器 (HWA) 和 C7x DSP 以提高性能。内部研究表明，通过将立体整流和特征提取转移到 LDC、MSC 和 DSP，带有立体摄像机的 ORB SLAM 的吞吐量提高了 2 到 3 倍。

### 3 周围环境感知

移动机器人必须理解周围环境的动态变化，以实现安全导航。必须尽快检测并避开阻碍移动机器人行进的动力障碍物。移动机器人也可以在定位过程中检测障碍物，因为地图中不存在障碍物的特征。但并不总是能够成功识别这些特征。有时，属于障碍物的特征被错误地匹配到地图中的特征。因此，识别并排除属于障碍物的特征对于可靠定位非常重要。

这就是 AI 和传感器与毫米波 (mmWave) 雷达等其他传感器模式融合的意义所在。AI 和传感器融合可帮助移动机器人准确感知动态物体，从而有助于更安全、更智能地进行导航。基于视觉的深度学习网络能够检测障碍物，对其进行分类，并测量它们与移动机器人的距离。TI 毫米波雷达的独特之处在于，雷达提供障碍物的距离、速度和到达角信息，机器人可以利用这些信息更好地导航而不会发生碰撞。

AM69A 处理器具有四个运行频率为 1GHz 的 512 位 C7x DSP，每个 C7x DSP 都与支持每周期 4K 8 位固定乘法累加 (MAC) 的四个 MMA 的每一个紧密耦合。四个 MMA 提供密集的 32 万亿次运算/秒 (TOPS)，可同时支持具有多个传感器的各种深度学习网络。此外，借助适用于 AM69A 的 Processor SDK Linux，可在 AM69A 上更轻松、更快速地开发 AI 应用。该软件开发套件 (SDK) 实现了多个开源组件之间的相互作用，并且还支持深度学习运行时（例如，在基础 Linux® 组件基础上构建的 TFLite、ONNX 和 TVM）以及用于远程内核和 HWA 的固件包。TI 将其原始训练框架中的 100 多个模型 (PyTorch、TensorFlow 和 MXNet) 转换并导出为适用于 C7xMMA 架构的格式，并将其托管在 Edge AI Model Zoo 中。TI 还提供 Edge AI Studio 工具集，旨在加快在 TI 嵌入式处理器（包括 AM69A）上开发边缘 AI 应用。利用 Edge AI Studio，可以构建、评估和部署深度学习模型。TI E2E™ 论坛文章 [如何简化嵌入式边缘 AI 应用开发](#) 更详细地介绍了旨在助开发一臂之力的 TI 免费工具和软件。



## 4 规划路线

有了定位和周围环境感知功能，移动机器人就拥有确定路径并安全导航至目的地所需的所有信息。有多种路径规划算法可供使用。其中，占据栅格地图上的路径规划简单而有效。此外，由于是在更新占据栅格地图期间检测到障碍物，因此占据栅格地图广泛用于障碍物检测和路径规划。通过识别已占据和未占据的栅格，移动机器人可以发现一条高效且安全的通往目的地的路径。

图 4-1 是 AM69A 上自主导航系统的数据流示例，使用的是安装在移动机器人上的图像传感器。VPAC3 VISS 对原始图像进行处理并去马赛克，然后 LDC 消除图像失真。为了感知周围环境，MSC 重新调整未失真图像的大小，并将其转换为 RGB 格式，供 DL 网络使用。通过 MMA 加速 DL 推理，从图像传感器检测物体并确定其姿态。为了进行定位，从未失真图像中提取特征点，然后从地图中搜索相应的特征点，以确定移动机器人在地图上的姿态。C7x DSP 可以优化该定位过程。许多特征提取算法使用金字塔图像，而 MSC 可以加速金字塔图像的生成。进行路径规划时将考虑到移动机器人的姿态和检测到的物体。由于也可以使用移动机器人的姿态将检测到的物体投影到地图上，因此会识别移动机器人周围的自由空间，从而确定通往目的地的路径和移动机器人的控制命令。该示例数据流可通过多个图像传感器进行扩展，并在传感器之间进行适当的外部校准。

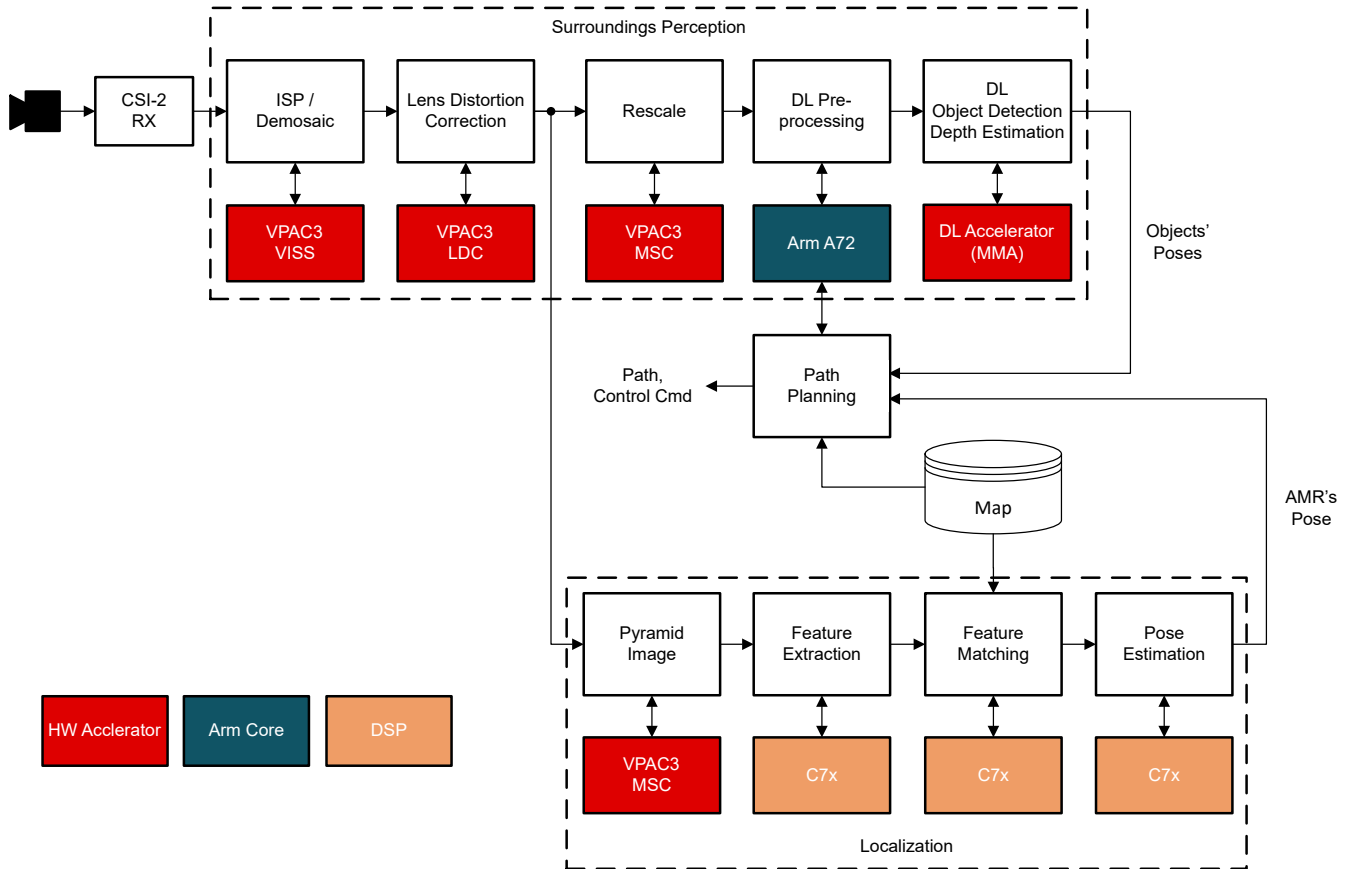


图 4-1. AM69A 自主导航系统的示例数据流

采用 2D 激光雷达和 IMU 的自主导航系统是在 [Robotics SDK](#) 基础上实现，并配有 [SCUTTLE 机器人](#)。对于该系统，使用 2D 激光雷达 SLAM 离线构建占据栅格地图，并使用相同的 2D 激光雷达传感器将该地图用于实时导航。对于自主导航，将在每次激光雷达扫描时执行图 1-1 中的三项任务（即定位、周围环境感知和路径规划），并且控制命令将转换为脉宽调制 (PWM) 信号以控制电机。[工程页面](#)中描述了这一数据流，并且可以在 [AM69A 处理器](#)上复制该系统。

## 5 总结

本白皮书提供了有关建图、定位、动态环境感知和路径规划的技术概述，这些都是移动机器人自主导航的核心部分。添加 AI 使机器人更智能、更安全，从而可在家庭自动化、工厂自动化、仓储和物流等更广泛的应用中提高生产力和效率。[AM69A 处理器](#)有助于为自主移动机器人构建精心设计的平台，提供出色的计算能力、AI 性能以及易于使用的软件开发工具和支持。



## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司