



Dilshan Godaliyadda, Deepak Poddar, Soyeb Nagori, Do-Kyoung Kwon, and Pramod Swami

## 摘要

无论驾驶辅助或导航任务有多简单，准确映射车辆或机器人环境的能力都是极为重要的。运动恢复结构算法 (SFM) 是满足这一需求的最被广泛使用的算法之一，因为它只用一个摄像头传感器就能实现精确映射。本应用报告描述了 TDA4VM 器件上支持占位栅格 (OG) 映射的 SFM 算法的硬件加速实施。TDA4VM 片上系统 (SOC) 是 Jacinto™ 7 系列 SoC 中的第一款商用器件，该系列 SoC 是专为汽车和类似机器人应用从零设计的。此系列 SoC 包含两个主要变体，分别是为 ADAS 和机器人中的类似认知任务而设计的 TDA4x 系列，以及为云连接网关系统而设计的 DRA8x 系列。TDA4x 系列是为认知和相关分析而设计的，所以配备了 SFM 等算法可以无缝映射的硬件加速器和通用处理内核。本应用报告将介绍关于 SFM 的这种映射。

## 内容

1 引言.....	2
2 什么是运动恢复结构？.....	2
3 占位栅格映射简介.....	3
4 从点云到 OG 地图.....	3
5 算法流程：基于 SFM 的 OG 映射.....	4
6 算法流程：TDA4VM 上基于 SFM 的 OG 映射.....	4
7 TDA4VM 上的第一个实施示例.....	6
8 TDA4VM 上的第二个实施示例.....	7
9 参考文献.....	7

## 商标

Jacinto™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

## 1 引言

无论是车道辅助或盲点检测等简单的任务，还是自主导航等更复杂的任务，了解车辆或机器人的周围环境对于任务的成功和安全都至关重要。车辆和机器人通过将雷达、激光雷达和摄像头等传感器捕捉的数据转换为车辆决策引擎可使用的格式来感知环境。基于激光雷达 (LiDAR) 的地图通常是最准确的，但对于大多数车辆或机器人来说成本过高。因此，基于雷达和摄像头的解决方案通常使用范围更广。

**SFM** 算法是使用最广泛的基于摄像头的映射算法之一。**SFM** 算法本身输出一个点云 (从周围的物体提取的一组点)，然后某类映射算法可以使用该点云。本文所述的应用将点云输送给占位栅格映射算法以生成周围环境的地图。

在汽车和机器人应用中，接收传感器数据、将数据转换为可用格式以及根据感知环境规定动作的操作通常在嵌入式平台上执行。德州仪器 (TI) 的 **Jacinto 7 TDA4x** 系列高性能 **SoC** 专为满足汽车、工业和机器人市场的各种算法需求而从零设计。运动恢复结构 (**SFM**) 算法就是该器件的设计所依据的一种算法。因此，该算法的关键计算块无缝地映射至 **TDA4VM** 器件上的硬件加速器或通用处理内核。本文描述了基于 **SFM** 的 **OG** 映射算法、**TDA4VM** 器件以及该算法如何映射到该器件以实现环境的高保真实时地图，然后展示了该器件上的一些实施示例和相应的输出。

## 2 什么是运动恢复结构？

在计算机视觉中，物体相对于车辆的位置是通过由两个摄像头拍摄的图像确定的，这两个摄像头各自安装在完全不同的位置，均朝向目标物体。其中重要的是提取和匹配两幅图像中该物体的关键点，然后使用一种叫做三角测量的过程来解析构成物体的点的位置。在计算机视觉领域中，使用两个摄像头区分点在空间中的位置的过程叫做立体视觉或立体深度估算，两幅图像中所有对应项生成的点集叫做点云。尽管立体视觉在汽车和机器人领域中得到广泛应用，但满足图像处理要求方面的系统成本很高，因为它需要两个高精度摄像头以相对较高的频率捕捉图像。

相比之下，运动恢复结构 (**SFM**) 算法可以在运动中通过单个摄像头生成点云。顾名思义，**SFM** 中有一个摄像头由于运动而在两个连续时间实例下处于两个不同的位置，考虑到帧中的对象在两个时间实例之间没有移动而且我们知道摄像头的相对运动，所以这就相当于拥有两个处在不同位置的摄像头。因此，只用一个摄像头就可以使用立体视觉理论有效地生成点云。

**SFM** 算法有两种主要类型，分别基于传统计算机视觉和深度学习。尽管这两类都可以在 **TDA4VM** 上执行，但本文的重点在于前者，即基于传统计算机视觉技术的算法。然后，需要利用由 **SFM** 算法生成的点云来生成周围环境的地图，在此处所述的应用中，映射任务使用了 **2D OG** 映射方法。

### 3 占位栅格映射简介

占位栅格映射方法广泛用于表示车辆或机器人周围环境，因为它们可用于从泊车、障碍物识别到路缘检测等各种 ADAS 应用，并且可以高效存储。OG 地图是用二维 (2D) 或三维 (3D) 栅格表示的车辆周围环境。地图中的每个栅格单元都有一个对应的状态，例如 *占用*、*空闲* 或 *未知*，根据从雷达、摄像头或激光雷达等传感器接收到的信息计算得出。图 3-1 中显示了 OG 地图的简单说明性示例。

OG 地图可以以车辆或机器人为中心，也可以任意参照系为中心。前者叫做 *以自我为中心的 OG 地图*，后者叫做 *以世界为中心的 OG 地图*。根据是否使用前一帧的信息，OG 地图进一步划分为累积或瞬时地图。

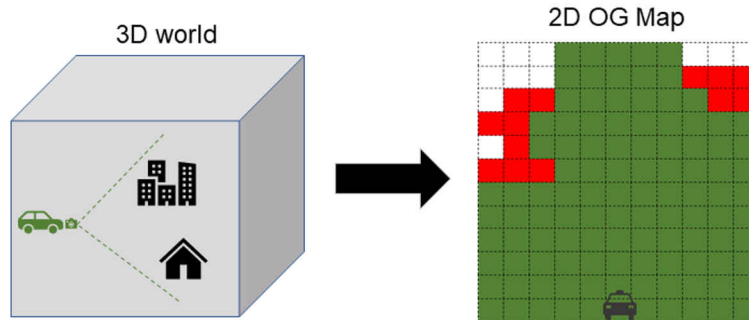


图 3-1. 左：世界图示；右：OG 地图图示  
(绿色单元表示为空，红色单元表示被占用，白色单元表示未知)

本应用报告描述了 *以世界为中心的 2D OG 地图*，它是使用由单眼摄像头采集的测量值所生成的点云构建的。此 OG 地图有三种状态 - *占用*、*空闲* 或 *未知*。本应用报告重点探讨了瞬时 OG 映射算法。然后，本文的后半部分将简要描述 TDA4VM 器件随附的软件开发工具包 (SDK) 中包含的一个累积 OG 映射应用。下一节将描述如何以点云为输入，根据点云和车辆/机器人的位置生成 OG 地图。

### 4 从点云到 OG 地图

若要创建以世界为中心的 OG 地图，首先使用来自惯性导航系统 (INS) 的车辆位置数据和传感器校准数据，将 SFM 算法生成的点云中的每个点放置在以世界为中心的坐标系中。传感器校准数据包括不同传感器彼此间的相对位置。然后，每个点分类为源自地面 (*地面点*) 或源自物体 (*物点*)。接下来，使用以世界为中心的点云如下确定 OG 地图中每个单元的状态 -- 如果在一个单元中发现预定数量的 *物点*，则视为 *占用*；如果在一个单元中发现预定数量的 *地面点*，则视为 *空闲*；如果两个条件都不符合，则视为 *未知*。

## 5 算法流程：基于 SFM 的 OG 映射

图 5-1 中显示了本应用报告中描述的基于 SFM 的 OG 映射算法的算法流程。

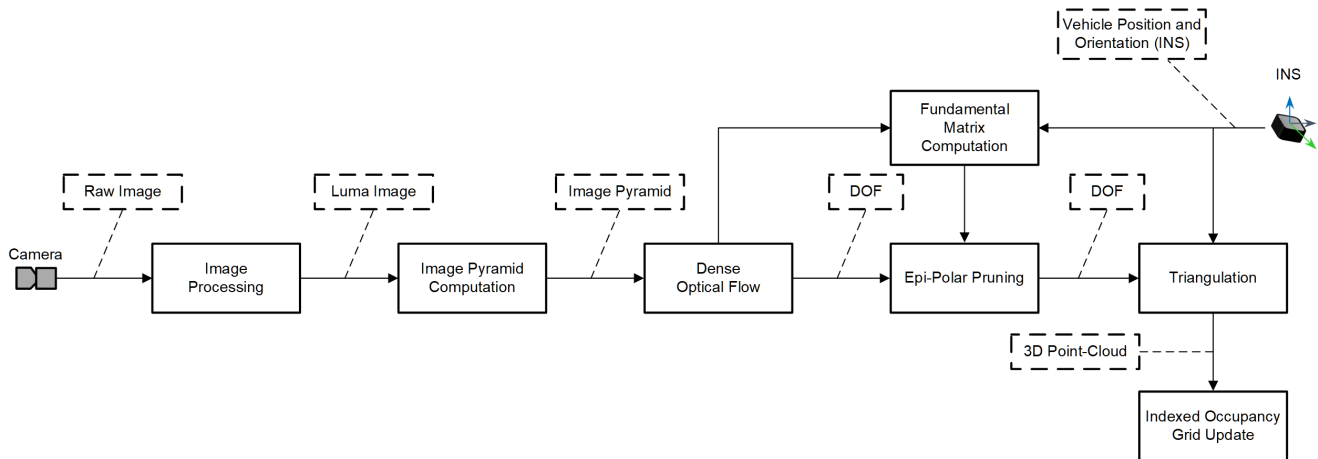


图 5-1. 基于 SFM 的占位栅格映射算法流程

请务必注意，密集光流 (DOF) 块、图像锥体形成和图像处理等子任务需要在硬件加速器 (HWA) 上执行，因为没有硬件加速，这些处理密集型任务就无法实时执行。

## 6 算法流程：TDA4VM 上基于 SFM 的 OG 映射

TDA4VM SoC 是围绕 ADAS 和机器人必需执行的 SFM 和其他常见计算机视觉任务而设计的，毫无疑问，它非常适用于 SFM 算法。图 6-1 显示了 TDA4VM 器件，图 6-2 显示了图 5-1 中映射到 TDA4VM 的算法流程。此处提到的视觉预处理加速器 (VPAC) 包含一些模块，可加速不同的图像预处理子任务，如色调映射、噪声滤波、镜头失真校正等。深度和运动处理加速器 (DMPAC) 包含两个模块，分别是加速立体深度估算的立体视差引擎 (SDE) 和加速 DOF 的密集光流引擎 (DOF 引擎)。更多有关 TDA4VM 器件的信息，请参阅 <https://www.ti.com.cn/product/cn/TDA4VM>。

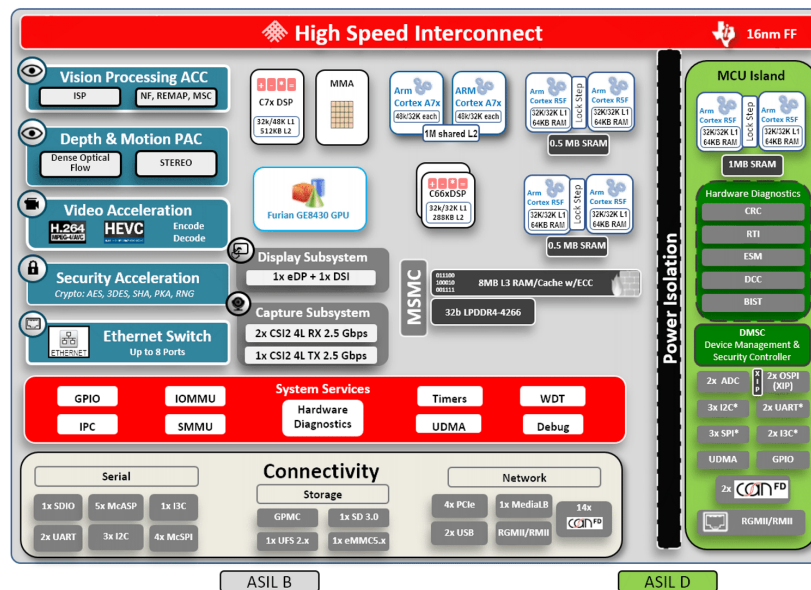


图 6-1. TDA4VM 方框图

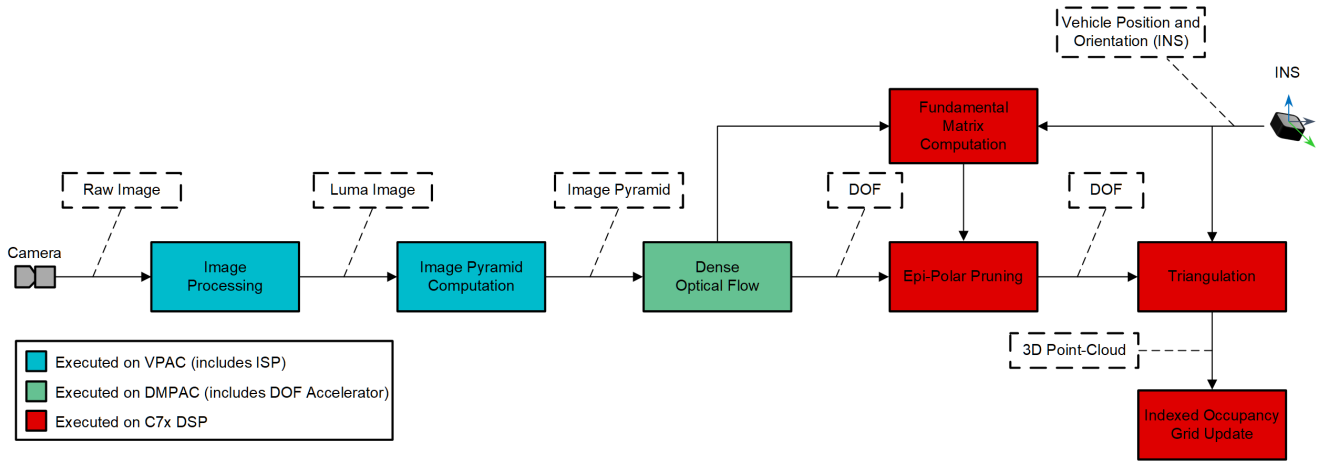


图 6-2. TDA4VM 上基于 SFM 的占位栅格映射算法流程

表 6-1 显示了 SFM OG 映射算法地图的子任务在列表中对应的 TDA4VM SoC 中的不同处理器。

表 6-1. SFM OG 映射算法，子任务到 TDA4VM 的映射

步骤	TI 处理器器件 IP
原始图像捕获	图像传感器
图像发送到处理器	CSI-2 端口
图像处理成 Luma 图像	视觉预处理加速器 (VPAC)
图像锥体形成	视觉预处理加速器 (VPAC)
密集光流计算	深度和运动处理加速器 (DMPAC)
极线修剪	C7x DSP
基础矩阵计算	C7x DSP
三角测量	C7x DSP
索引 OG 地图更新	C7x DSP

如果不通过专用 HWA 加速，其中某些步骤的计算量会大得惊人。例如，算法中的 DOF 部分。对于 2MP 摄像头输入，C7x DSP 消耗量为每秒 2000 兆周 (MCPS) 以生成 2000 个关键点。而 DOF HWA 仅消耗 394 个 MCPS。表 6-2 汇总了结果。

表 6-2. 2MP 摄像头输入 C7x DSP 与 DOF HWA 的 DOF 计算比较

TI 处理器器件 IP	每秒兆周 (MCPS)
DMPAC 上的 DOF HWA	394
C7x DSP	2000

此外，此算法中的所有子任务都可以使用 DSP 和 HWA 来执行，因此 TDA4VM 器件中的通用 ARM 内核和深度学习加速器可以预留给其他应用。

接下来介绍 SFM OG 映射的两个实施示例。这两个应用都包含在 TDA4x 器件附带的 SDK 中。第一个适用于道路上行驶的车辆，因此使用由安装在车辆前方的摄像头提供的输入数据。第二个适用于自动泊车，因此使用由安装在车辆侧面的摄像头提供的输入数据。

## 7 TDA4VM 上的第一个实施示例

此应用使用保存的图像和 INS 数据运行，因此跳过了上图中摄像头传感器到 VPAC 分支的步骤。有关如何运行应用的文档以及所有必要的库可在 SDK 中找到，并且在节 9 中列出。如需应用的特定文档，请点击[此处](#)。

图 7-1 显示了此应用的输出样例。图表上方和下方分别显示了密集光流的输出和稀疏光流 (SOF) 的输出。此图清晰地说明了为什么 OG 映射等应用优选 DOF 而非 SOF，因为在稀疏点云中无法检测来向汽车。实验采用了由 Carla Simulator 0.9.9.4 [1] 生成的模拟数据，其中车辆前方安装了虚拟摄像头。这里的空闲单元标记为绿色，占用单元标记为红色，未知单元标记为黄色。此示例以不到 60% 的 C7x DSP 利用率、30fps 的帧率使用 0.5MP 摄像头数据生成 20,000 个 3D 关键点。虽然在此应用中跳过了 ISP 部分，但 ISP 有能力以 600MP/s 的速度处理数据，因此能接受 30fps 帧率的 0.5MP 单目摄像头。用户可以点击[此处](#)申请一个数据表，其中包含了与不同内核利用率相关的精确信息。



图 7-1. 含 DOF [上] 和 SOF [下] 的 SFM OG 映射应用输出示例

客户可以使用 SDK 中的示例 (包括 SFM OG 映射示例) 以缩短开发时间。如果客户更愿意使用自己的流水线或修改的流水线，可以使用这些示例作为参考或起点。

## 8 TDA4VM 上的第二个实施示例

这个基于 SFM 的 OG 映射应用位于 SDK 的 PTK 演示部分下。如需此算法的文档，请点击[此处](#)。这个特殊的应用构建了一个以世界为中心累积而成的 2D OG 地图，而不是由上述应用创建的瞬时地图。此算法使用的链条几乎与上述链条相同，只有几个微小的差异。图 8-1 显示了此应用的输出样片。这里的摄像头安装在车辆侧面，也使用在 TI 内部停车场中捕捉的已保存数据。这里的空闲单元标记为绿色，占用单元标记为红色，未知单元标记为黑色。

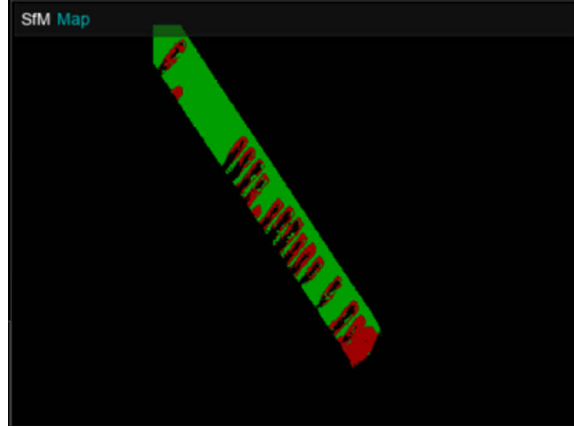


图 8-1. SFM OG 映射应用的输出示例

## 9 参考文献

1. Dosovitskiy, Alexey, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An open urban driving simulator." In Conference on robot learning, pp. 1-16. PMLR, 2017.

表 9-1. 重要链接

参考文献名称	链接
TDA4VM 产品信息：	<a href="https://www.ti.com.cn/product/cn/TDA4VM">https://www.ti.com.cn/product/cn/TDA4VM</a>
评估板购买/申请	常见板： <a href="https://www.ti.com.cn/tool/cn/J721EXCPXEVM">https://www.ti.com.cn/tool/cn/J721EXCPXEVM</a> TDA4VM 处理器 SOM： <a href="https://www.ti.com.cn/tool/cn/J721EXSOMXEVM">https://www.ti.com.cn/tool/cn/J721EXSOMXEVM</a>
下载 SDK	<a href="https://www.ti.com.cn/tool/cn/PROCESSOR-SDK-DRA8X-TDA4X">https://www.ti.com.cn/tool/cn/PROCESSOR-SDK-DRA8X-TDA4X</a>
SDK 文档	<a href="https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/latest/exports/docs/psdk_rtos/docs/user_guide/index.html">https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/latest/exports/docs/psdk_rtos/docs/user_guide/index.html</a>
Jacinto 7 视频培训系列	<a href="https://training.ti.com/jacinto7-platform">https://training.ti.com/jacinto7-platform</a>
E2E 处理器论坛的技术支持：	<a href="https://e2e.ti.com/support/processors/f/791">https://e2e.ti.com/support/processors/f/791</a>
PTK API 指南 (第二个应用中使用的库)	<a href="https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/latest/exports/docs/perception/docs/ptk_api_guide/index.html">https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/latest/exports/docs/perception/docs/ptk_api_guide/index.html</a>

## 重要声明和免责声明

TI 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安全、安保或其他要求。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 (<https://www.ti.com/legal/termsofsale.html>) 或 [ti.com](https://www.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。

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

Copyright © 2021, 德州仪器 (TI) 公司



## 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 © 2022，德州仪器 (TI) 公司