



环境识别

使用 SLAM 算法进行环境识别

在实现自动驾驶的过程中，随时了解车辆在其环境中的位置显得至关重要。但是在没有详细地图或卫星导航的情况下，如何确定车辆的位置呢？为此，SLAM 算法提供了智能解决方案。

了解车辆在其环境中的确切位置是自动驾驶的重要前提条件。最常见的解决方案是使用精确的地图和卫星或惯性导航系统（惯性测量单元 (IMU)）以及航位推算导航（使

用运动方向和速度进行位置识别）。但是，在可预见的未来，还只有一些成片拼凑的详细地图网络可供使用，而且卫星辅助定位的精度通常也不够高。为此，同步定位与地图构建 (SLAM) 算法提供了相应的解决方案。

这些算法使用环境传感器（例如激光雷达传感器）的数据生成整个车辆环境的三维地图，并在该环境中定位车辆。当前环境中可用的数据越多（例如将粗略地图数据与采集的三维数据进行比较和对应），算法的精度就越高。



在 RTMaps 中使用 SLAM 算法

来自 Dibotics 的增强型 LiDAR 3D SLAM 算法可用于多传感器开发环境 RTMaps 的预配置组件库（实时多传感器应用，请参阅信息框简介：Intempora 的 RTMaps 工具）。

RTMaps 集成在 dSPACE 工具链中。SLAM 算法是一种完全基于传感器数据的感知和定位算法，它们既不需要传感器融合，也不需要测距（基于推进系统的位置估计，例如车轮旋转）。仅仅基于激光雷达传感器的数据，SLAM 算法便可以通过组合大量数据集实时生成三维模型。然后，可以从不同的角度查看三维模型，并且还能对被检测对象进行分类（图 1）。同时，车辆可在此环境中进行定位。之后，环境模型将成为自动驾驶应用后续开发步骤的基础，例如，其可用于轨迹规划和运动控制。在 RTMaps 中，SLAM 库的一个简单应用示例是其可回放驾驶测试期间采集的传感器数据。基于模块的开发通常简单而直观。首先，通过拖放操作将所需组件从 RTMaps 组件库添加到图表中。然后通过对话框对组件进行参数化，并在最后一步中根据需要需要通过数据链路进行连接。最后，只需单击按钮即可执行和测试应用。为此，需要通过回放模块回放保存到存储介质的原始数据，并将其传输到 SLAM 和分段算法。算法输出将通过三维可

视化模块以图形方式进行显示。在鼠标操作期间，可以更改三维模型的观测视角。由于 RTMaps 具有多线程功能和高效的内存管理，即使是复杂的算法也可以被高效运行。

在车辆中开发自动驾驶功能

为了在车辆中快速执行原型开发和算法测试，应用系统必须连接到真实的环境传感器。只需将 RTMaps 图中的回放模块替换为用于采集传感器数据的组件，即可完成这一连接。开发环境提供各种接口，用于连接摄像头、雷达、激光雷达和车辆总线。它还设置了时间戳，按照与时间相关的方式采集、处理和回放传感器数据。对于自动驾驶功能的开发，我们通常使用特定算法处理所生成的三维环境模型的信息，例如用于情况分析或轨迹规划。这些算法可以作为自定义组件（基于 C++、Python 或 Simulink® 代码）集成到 RTMaps 中。在集成自定义算法时，开发人员可以通过使用 Windows® 和 Linux 中的 SDK Wizards 以及大量代码示例获得支持。此外，可以直接与车载开发平台 dSPACE MicroAutoBox 上执行的控制算法进行所需的数据交换，例如用于运动控制。通过专用组件库能够在 dSPACE 工具链中无缝集成 RTMaps，并确保通过 dSPACE 实时平台实现时间同步的数据传输。 >>

RTMaps 可以在所有重要步骤中帮助开发人员创建自动驾驶算法，支持开发人员在嵌入式平台上进行开发。

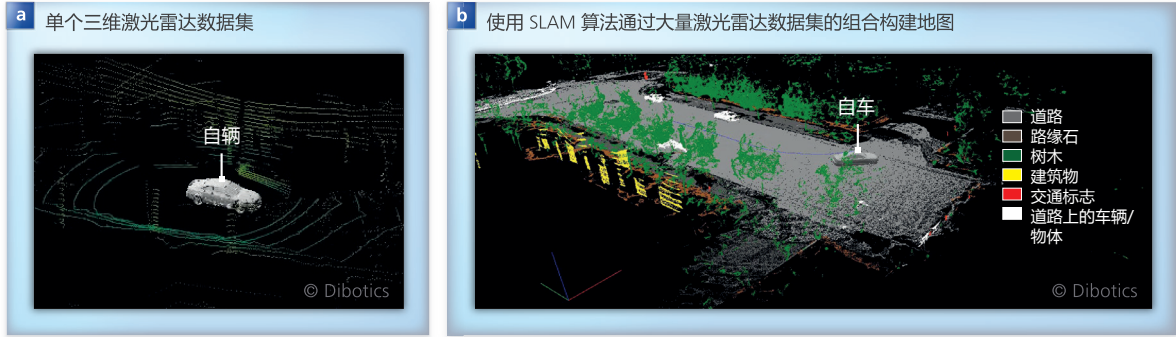


图 1：将未经处理的单个激光雷达点云数据 (a) 与由大量激光雷达数据集 (b) 构建的三维地图进行比较，证明了 SLAM 算法的有效性。

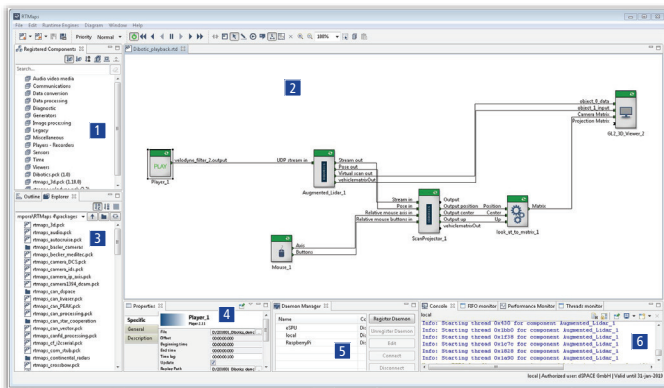
连续工作 – 从 PC 到嵌入式平台

开发平台的功能及性能（例如计算能力或硬件加速）能够影响复杂算法的运行行为。因此，开发人员往往希望尽可能在早期阶段在目标平台上进行开发。无论他们选择哪一种平台：在 RTMaps 的便捷工具支持下，他们都能在 PC 上熟悉的操作环境中工

作。利用 Remote Studio Connector（图 3）可以经由 TCP/IP 直接连接目标平台以处理和执行 RTMaps 框图，而目标平台本身则无需连接键盘、鼠标或显示器。这样的开发特性使工作环境更加便捷，因为开发人员一直在适合所选系统的组件库中工作，还可以直接从所选平台的数据系统加载和

保存应用。此外，通过参数化选项还可以在目标平台启动时立即在运行环境中加载和启动所需的应用。通过向 RTMaps Studio 添加嵌入式平台支持，自动驾驶功能的原型开发达到了最佳水平，因为开发人员对嵌入式平台的 PC 环境十分熟悉，进而可以简单、方便地进行开发工作。■

图 2：RTMaps 的用户界面。模块化开发环境可以轻松处理各种多传感器应用。为此，RTMaps 提供了丰富全面的组件库。



- 1 组件库 – 显示在 RTMaps 环境中注册的/可用的组件模型。这些组件按功能分类（传感器、查看器、图像处理等）。
- 2 RTMaps 框图 – 用于创建应用程序框图。本例中所使用的元件是用于回放激光雷达数据的模块、用于 SLAM 和分段算法的 Dibotic 组件、用于设置观测视角的鼠标控制元件以及三维可视化元件。
- 3 资源管理器和概览 – 资源管理器：提供了计算机本地目录视图，通常显示的目录带有可在 RTMaps 中注册的程序包。概览：此视图显示了整个框图，能够进行概览和快速导航。
- 4 属性 – 用户可以为当前图表设置不同的参数。
- 5 Daemon Manager – 用户可以连接到嵌入式平台，并且可以打开、处理和执行远程图表。
- 6 控制台 – 提供有关各种主题的信息，例如命令、组件消息、警告和错误。

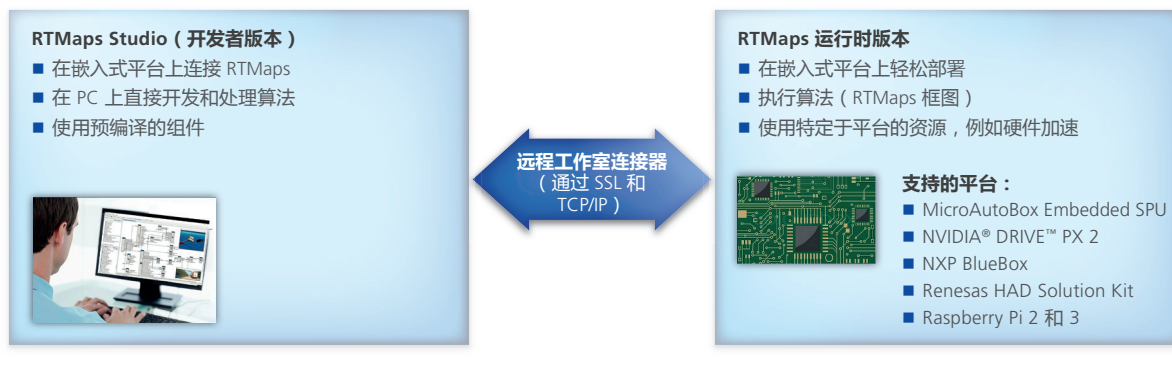
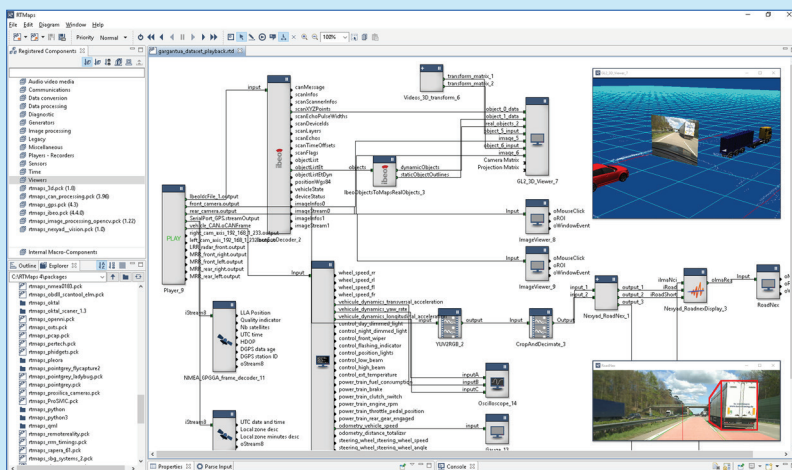


图 3 : 通过将 RTMaps 与 Remote Studio Connector 结合使用, 开发人员可以访问各种嵌入式平台上的嵌入式原型环境, 并且可以在熟悉的 PC 工作环境中工作。

简介: Intempora 的 RTMaps 工具



ARM 的平台上提供独特的多线程功能, 并进行高效的内存管理。

dSPACE 工具链中的无缝集成

dSPACE 专门为实时系统和基于 PC 的仿真平台 VEOS 提供了一个 Simulink 环境下的接口模块库, 用于将 RTMaps 集成至 dSPACE 工具链中。ControlDesk 也可以与 RTMaps 轻松连接, 以实现对所开发应用的参数化操作。

新增: 直接在嵌入式平台上进行原型开发

通过将 RTMaps 与 Remote Studio Connector 相结合, 现可以更快地在嵌入式平台上进行更快捷的原型开发, 从而实现高度自动化的驾驶。无需为嵌入式平台连接鼠标、键盘以及显示器, 用户可以基于熟悉的 RTMaps 开发环境, 在远程 PC 上方便直接地对目标平台进行开发工作。

面向多传感器应用的强大开发环境

Intempora RTMaps (dSPACE 自 2016 年以来一直是此产品的经销商) 是一款模块化的软件开发和运行时环境, 用于复杂的多传感器应用, 在自动驾驶、机器人和航空航天工业等特定领域发挥了重要作用。开发人员通过使用全面的组件库可以准确地采集、同步和处理来

自各种传感器 (摄像头、雷达和激光雷达) 的数据。用户特定的组件可以使用 C++、Python 或 Simulink 代码进行开发, 而且易于集成。一些更为复杂的算法, 如: OpenCV, 来自 Dobotics 的增强型激光雷达三维 SLAM 算法, 以及 NVIDIA 的 DriveWorks 都已经是现成的组件库资源。RTMaps 可以在基于 PC 和