

# 视觉SLAM算法综述（2021）

本篇是对《A Comprehensive Survey of Visual SLAM Algorithms》（DOI: 10.3390/robotics11010024）的个人翻译，供学习使用。

作者：Atomic Energy and Alternative Energies Commission

时间：

- 2021年12月12日（投稿）
- 2022年02月07日（接收）
- 2022年02月10日（发表）

- 摘要
- 一、简介
- 二、基于视觉的 SLAM 概念
  - 2.1. 纯视觉 SLAM
    - 2.1.1 特征法
    - 2.1.2 直接法
  - 2.2. 视觉惯性SLAM
  - 2.3. RGB-D SLAM
- 三、视觉SLAM算法
  - 3.1. 纯视觉 SLAM
    - 3.1.1 MonoSLAM (2007)
    - 3.1.2 并行跟踪和建图(PTAM, Parallel Tracking and Mapping) (2007)
    - 3.1.3 稠密跟踪和建图 (DTAM, Dense Tracking and Mapping) (2011)
    - 3.1.4 半直接视觉里程计 (SVO, Semi-Direct Visual Odometry, 2014)
    - 3.1.5 大规模直接单目SLAM (LSD-SLAM, Large-Scale Direct Monocular SLAM, 2014)
    - 3.1.6 ORB-SLAM 2.0 (2017)
    - 3.1.7 CNN-SLAM (2017)
    - 3.1.8 直接稀疏测距法 (DSO, Direct Sparse Odometry, 2018)
    - 3.1.9 小结
  - 3.2 视觉惯性SLAM (VI-SLAM, Visual-Inertial SLAM)
    - 3.2.1 多状态约束卡尔曼滤波器 (MSCKF, Multi-State Constraint Kalman Filter, 2007)
    - 3.2.2. 基于开放关键帧的视觉-惯性SLAM (OKVIS, Open Keyframe-Based Visual-Inertial SLAM (2014))
    - 3.2.3. Robust Visual Inertial Odometry (2015)

- 3.2.4. 视觉-惯性ORB-SLAM (VIORB, Visual Inertial ORB-SLAM, 2017)
  - 3.2.5. 单目视觉惯性系统 (VINS-Mono, Monocular Visual-Inertial System, 2018)
  - 3.2.6. 视觉惯性直接稀疏测距 (VI-DSO, Visual-Inertial Direct Sparse Odometry, 2018)
  - 3.2.7. ORB-SLAM3 (2020)
  - 3.2.8 小结
  - 3.3. RGB-D SLAM
  - 3.3.1. KinectFusion (2011)
    - 3.3.2. SLAM++ (2013)
    - 3.3.3. Dense Visual Odometry (2013)
    - 3.3.4. RGBDSLAMv2 (2014)
    - 3.3.5. General Comments
- 四、主要问题与未来方向
    - 4.1. 基于深度学习的算法
    - 4.2. 基于语义的算法
    - 4.3. 动态SLAM算法
  - 五、数据集和标定
  - 六、结论
  - References

## 摘要

“同时定位和建图”(SLAM)技术可以在未知环境中同时创建地图，并估计传感器的位姿，因此被广泛研究。其中视觉SLAM技术相对于其他基于传感器的SLAM技术，拥有成本更低、体积更小的优势，由此在该领域发挥着重要作用。

文献系统显示实现视觉SLAM系统有着多种不同路径和方法。在各种已发表文章中，该领域的初学者可能会在发现识别和分析主流算法以及根据他的项目限制选择最合适的算法上存在一定问题。

我们的主要工作如下：

1. 介绍三种主流的视觉SLAM实现路径：**纯视觉、视觉惯性和RGB-D**，通过图表和流程图对每种路径的主要算法进行了回顾，并给出每种技术的优缺点。
2. 提出六个标准来简化SLAM算法的分析工作，并同时考虑了软件和硬件条件。
3. 提出视觉SLAM领域的一些主要问题和未来方向，并对一些现有的基准数据集进行概述。

我们的目的是让那些发起SLAM项目的人能够很好地理解SLAM技术的主要元素和特征。

关键词：嵌入式SLAM；评估标准；RGB-D SLAM；视觉惯性SLAM；视觉SLAM；3D重建

## 一、简介

同步定位和建图 (SLAM) 技术由 **Smith** 在 1986 年<sup>[1]</sup> 首次提出，应用范围广泛，尤其是在**增强现实 (AR)**<sup>[2-4]</sup>和**机器人技术**<sup>[5-7]</sup>领域。**SLAM**过程旨在建图未知环境，同时通过传感器提供的信号在该环境中定位传感器系统。在机器人技术中，地图的构建是一项至关重要的任务，因为它允许对地标进行可视化，从而促进环境的可视化。此外，它可以帮助机器人进行状态估计，重新定位，并在重新访问已访问区域时减少估计误差<sup>[2]</sup>。

地图构建还有另外两个任务：**定位**和**路径规划**。根据 Stachniss<sup>[3]</sup>，建图问题可以描述成从机器人的视角检查三个问题：**世界是什么样的？我在哪里？以及如何到达给定位置？**

1. 建图任务阐明了第一个问题，该任务通过搜索以构建环境模型。为此，它需要观察到的地标位置（南川注：由这些地标组成地图），即由定位任务提供的第二个问题的答案。
2. 定位任务通过搜索以确定机器人的位姿，即其方向和位置，从而在地图上定位机器人。
3. 根据前两个任务，路径规划解决了最后一个问题，并试图估计机器人到达给定位置的轨迹。它依赖于定位任务提供的当前机器人的位姿，以及建图任务提供的环境特征。

**SLAM 是一种集成了建图和定位任务的解决方案。**

基于视觉的 SLAM 算法被认为特别有吸引力，因为它们的传感器配置简单、尺寸小型化、成本低。因此，文献中提出了许多基于视觉的技术，这使得根据项目限制选择最合适的技术变得困难。基于视觉的方法可以分为三个主要类别：纯视觉 SLAM、视觉惯性 (VI) SLAM 和 RGB-D SLAM。

1. 纯视觉SLAM是指仅基于单目或立体相机提供的 2D 图像的 SLAM 技术。由于视觉输入有限<sup>[4]</sup>，它们呈现出更高的技术难度。
2. 视觉惯性SLAM是指在纯视觉的基础之上添加惯性测量单元 (IMU) 来提高视觉 SLAM 算法的鲁棒性，惯性测量单元 (IMU)具有小型化和低成本的特点，同时实现了高精度，这是许多需要轻量级应用的重要方面设计，例如 自动赛车<sup>[ 11]</sup>。
3. RGB-D SLAM是在视觉的 SLAM 系统之上结合深度传感器以处理深度信息。

为了获得对 SLAM 问题的总体概述和介绍， Durrant-White 和 Bailey<sup>[12,13]</sup>的工作提出了一个从问题描述到环境表示的 SLAM 教程。此外， Cadena 等人<sup>[2:1]</sup>分析了SLAM的主要开放问题和未来前景。在视觉基础技术的回顾与调查上， Yousif等人<sup>[5]</sup>和富恩特斯-帕切科等人<sup>[6]</sup>概述了纯视觉 SLAM 技术和基本算法中使用的主要概念。优素福等人<sup>[5:1]</sup>还简要描述了基于 RGB-D 的 SLAM 问题。竹富等人<sup>[4:1]</sup>和 Covolan等人<sup>[7]</sup>概述了基于视觉的 SLAM 方法中使用的主要概念，重点关注纯视觉和基于 RGBD 的方法并描述了主要算法。 Servieres等人最近发表的文章<sup>[8]</sup>提出了主要基于视觉的 SLAM 算法的分类，并进行了历史研究。

桂等人<sup>[9]</sup>和陈等人<sup>[10]</sup>介绍了视觉惯性 SLAM 和视觉惯性里程计方法的主要概念和算法，考虑了基于滤波和基于优化的观点。在<sup>[9:1]</sup> 中，介绍了截至 2015 年的技术，而在<sup>[10:1]</sup>中，还包括了截至 2018 年的

算法。视觉惯性导航的主要概念和技术的概述也可以在 [11] 中找到。关于 RGB-D方法，Chen等人 [12] 从 RGB-D 建模中使用的主要概念提出了一个全局视角。Zhang等人最近的一项调查 [13] 概述了主要概念并描述了主流的基于 RGB-D 的 SLAM 算法。可以看出，文献中有一些关于视觉SLAM技术的评论和调查；尽管如此，它们中的大多数仅限于三种主要方法中的一种或两种，并且没有详细介绍算法。因此，有必要对这三种方法和基本算法进行回顾，以帮助研究人员和学生开始他们的视觉 SLAM 技术工作，以获得对这一大领域的概述。

因此，本文回顾了最具代表性的视觉SLAM技术，并概述了每种方法的主要优缺点。本文提出了三个主要贡献：

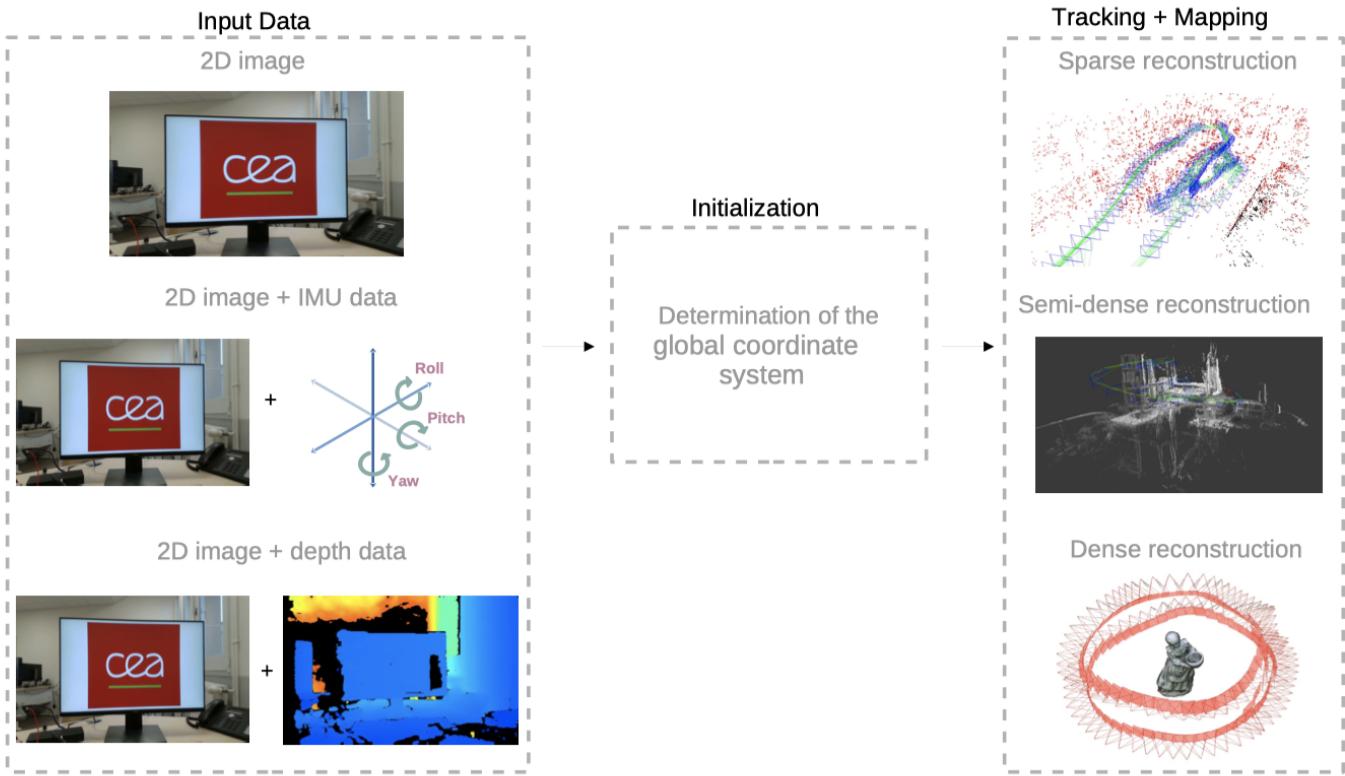
1. 通过图表和流程图的构建来解释最具代表性的基于视觉的 SLAM 算法。这种方法将对读者有所帮助，因为它在启动项目时提供了 SLAM 技术的概述，并允许读者第一次接触基于视觉的SLAM 算法。
2. 据我们所知，这是第一篇介绍三种主要基于视觉的方法的评论文章，对每种方法进行了单独的分析，并对这些方法进行了一般分析。
3. 针对开始研究 SLAM 算法的读者，我们提出了在不同技术和实现中要遵守的六个主要标准，方便根据个人的应用具体问题具体分析，包括软件条件（例如回环检测）和硬件条件（例如 SoC 实现上的 SLAM）。

本文组织如下：第2节介绍了三种选定方法的主要基本概念；第3节介绍了我们为评估 SLAM 算法和现有路径里最具代表性的 SLAM 技术而建立的六个标准；第4节介绍了视觉 SLAM 社区最近面临的一些主要问题，并指出了处理这些问题的未来方向；第5节概述了一些最重要的公开可用的基准数据集；最后，我们的结论在第6节中给出。

## 二、基于视觉的 SLAM 概念

本节介绍与基于视觉的 SLAM 和里程计算法相关的概念，以及本文所涵盖的基于视觉的方法的主要特征。

视觉SLAM在传感器系统中使用一个或多个摄像头，接收2D图像作为信息源。一般来说，基于视觉的 SLAM 算法分为三个主要线程：初始化、跟踪和建图<sup>[4:2]</sup>。图1显示了基于视觉的 SLAM 方法中通常存在的三个主要部分的总体视图。



**Figure 1.** General components of a visual-based SLAM. The depth and inertial data may be added to the 2D visual input to generate a sparse map (generated with the ORB-SLAM3 algorithm [22] in the MH\_01 sequence [23]), semi-dense map (obtained with the LSD-SLAM [24] in the dataset provided by the authors), and a dense reconstruction (Reprinted from [25]).

(深度和惯性数据可以添加到2D视觉输入，以生成稀疏地图重建、半稠密地图重建以及和稠密地图重建)

如图所示，在视觉 SLAM 系统中，输入可以是 2D 图像、或者 2D 图像加上 IMU 数据，也可以是 2D 图像加上深度数据，具体取决于所使用的方法，即纯视觉（第2.1 节）、视觉惯性（第2.2 节）或基于 RGB-D（第2.3 节）。初始化决定了全局坐标并构建初始地图，为接下来的两个主要任务服务：跟踪和建图。跟踪过程负责传感器位姿的连续估计，通常该算法在当前帧和地图之间建立 2D-3D 对应关系，构成了一个称为透视n点（perspective-n-points）的问题。有几种方法可以解决这个问题，**EPnP**是最具代表性的解决方案之一[14]。建图过程负责在相机移动时计算和扩展 3D 结构。深度数据计算根据所采用的算法而有所不同（第3节分别针对每种算法提供详细解释）。最后，根据实施的技术，建图过程将导致稀疏、半稠密或稠密的 3D 重建。

尽管在这篇文章中，我们把这些概念统归于 SLAM 方法论范畴，但我们分别考虑了视觉 SLAM 和视觉里程计 (VO, Visual-Odometry) 技术，因为它们密切相关。VO 算法同样也可以用于估计机器人的位置，它们把摄像头作为信息来源。**视觉SLAM和VO的主要区别在于是否考虑估计轨迹和地图的全局一致性**[5:2]。虽然 VO 仅执行局部优化，但视觉 SLAM 算法还采用回环检测（参见第3节），能够校正机器人轨迹末端累积的漂移。

## 2.1. 纯视觉 SLAM

纯视觉 SLAM 系统基于 2D 图像处理。在从多个角度获取图像后，系统执行初始化过程以决定全局坐标系并重建初始地图。在依赖于滤波算法的基于特征的算法中，第一步初始化地图点时具有高不确定性，但稍后可能会收敛到它们的实际位置。之后便是跟踪，它试图估计相机的位姿。同时，在更多未知场景被观察到后，建图过程会将新的点加入到3D重建。

纯视觉 SLAM 系统可以使用单目或立体相机。考虑到传感器的小尺寸（所有提出的方法中最小的）、价格低、易于校准和降低功耗<sup>[15]</sup>，基于单目相机的 SLAM 是一个经过充分探索的领域。尽管有这些优点，基于单眼的系统在系统初始化方面面临更大的挑战，因为至少需要两个不同的视图来确定初始深度、位姿估计以及与漂移和尺度估计相关的问题。最后一个问题可以通过立体相机来弥补，立体相机的主要优势在于仅在一帧中呈现立体视图。然而，传感器的尺寸比简单的单目相机更显著。此外，它需要对每一帧进行更多的处理，因为在立体匹配阶段需要进行图像校正。

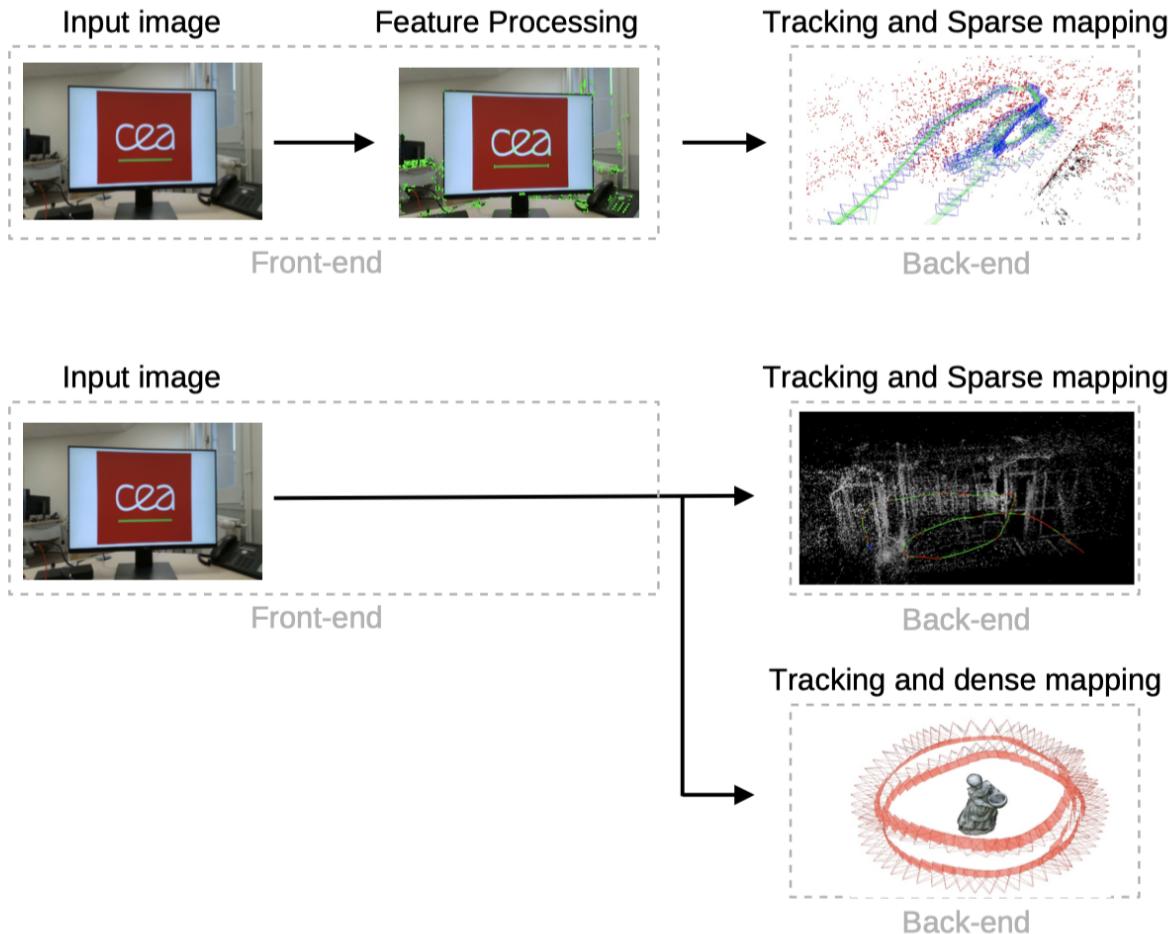
纯视觉SLAM可以分成两种方法：特征的法、直接法。

### 2.1.1 特征法

基于特征的 SLAM 算法考虑一定数量的兴趣点，称为关键点(KeyPoints)。它们可以在多张图像中被检测到，并通过比较它们的描述符进行匹配；这个过程提供了相机位姿估计信息。描述符数据和关键点位置组成特征，即算法用于处理跟踪和建图的数据。由于基于特征的方法不使用所有的帧信息，它们适用于嵌入式实现。然而，特征提取在无纹理环境中可能会失败<sup>[16]</sup>，并且它会生成稀疏地图，提供的信息比稠密地图少。

### 2.1.2 直接法

与基于特征的方法相比，直接法使用无预处理的传感器数据，考虑像素的强度，并最大限度地减少光度误差。有许多基于这种方法的不同算法，根据所选择的技术，重建可能是稠密的、半稠密的或稀疏的。重建密度是算法实时操作的一个重要约束，因为对于稠密和半稠密重建的结构和相机位置的联合优化比稀疏重建的计算成本更高<sup>[17]</sup>。图2显示了基于特征（间接）和直接法的主要区别，根据它们的前端和后端，即分别负责传感器数据抽象和抽象数据解释的的算法部分。



**Figure 2.** General differences between feature-based and direct methods. Top: main steps followed by the feature-based methods, resulting in a sparse reconstruction (map generated with the ORB-SLAM3 algorithm [22] in the MH\_01/EuRoC sequence [23]). Bottom: main steps followed by a direct method, that may result in a sparse (generated from the reconstruction of *sequence\_02*/TUM MonoVO [30] with the DSO algorithm [31]) or dense reconstruction (Reprinted from [25]), according to the chosen technique.

(特征法往往导致生成稀疏地图，直接法一般生成半稠密或稠密地图)

## 2.2. 视觉惯性SLAM

视觉惯性SLAM (VI, Visual-Inertial) 方法结合惯性测量来估计结构和传感器位姿。

惯性数据通过使用惯性测量单元 (IMU) 提供，该单元由陀螺仪、加速度计和另外的磁力计设备组成。通过这种方式，IMU 能够提供与沿 x、y、z 轴的角速率（陀螺仪）和加速度（加速度计）有关的信息，此外，还可以提供设备周围的磁场（磁力计）。

虽然添加 IMU 可能会增加环境的信息丰富度并提供更高的准确度，但也增加了算法的复杂度，尤其是在初始化步骤期间，因为除了相机位姿的初始估计外，算法还必须估计IMU位姿。**VI-SLAM**算法可以根据相机和IMU数据融合的类型来划分，可以松耦合也可以紧耦合。松散耦合方法不合并 IMU 状态来估

计完整姿势，相反，IMU 数据用于估计传感器位置的方向和变化<sup>[10:2]</sup>。而紧耦合方法将相机和 IMU 数据融合为运动方程，继而进行状态估计。

南川：我的理解是松耦合只估计传感器位姿，作为SLAM算法的输入；而紧耦合则综合传感器位姿与VO，两者以某种设定的程序互相影响。具体还得去看看第18号引文。

此外，VI-SLAM 算法根据其后端方法呈现不同的实现，可以是基于滤波或基于优化的。**基于滤波的方法的前端依赖于特征提取，而基于优化的方法（也称为基于关键帧的方法）依赖于全局优化**，这提高了系统的准确性，但也增加了算法的计算成本。

## 2.3. RGB-D SLAM

基于 RGB-D 数据的 SLAM 系统随着 **2010 年微软 Kinect 的问世**而开始受到更多关注。RGB-D 传感器由单目 RGB 摄像头和深度传感器组成，使 SLAM 系统能够以可行的精度直接获取深度信息，通过低成本硬件实时完成。由于 RGB-D 设备直接向 SLAM 系统提供深度图，因此基于这种方法的 SLAM 总体框架与已经提出的其他框架不同。

大多数基于RGB-D的系统利用迭代最近点（ICP）算法来定位传感器，融合深度图以获得整个结构的重建。RGB-D 系统具有无需任何预处理步骤即可提供彩色图像数据和稠密深度图等优点，因此降低了 SLAM 初始化的复杂性<sup>[4:3]</sup>。尽管如此，这种方法最适合室内环境，并且需要大内存和功耗<sup>[18]</sup>。

## 三、视觉SLAM算法

第2节中每种提出的路径都包括几种算法，因此很难根据自己的项目限制选择最合适的 SLAM 或 VO 算法。因此，我们提出了每种方法中最具代表性的算法，这些算法是根据文献反馈选择的，以完成对每种方法的简要回顾，并基于六个选定的标准进行系统分析，这些标准通常被视为 SLAM 项目的限制因素。

除了提出的标准外，还需要对场景和应用进行表征，因为某些场景可能会呈现特定的属性，这些属性可能暗示特定的评估标准，例如<sup>[19]</sup> 中提出的分析。作者考虑了自动驾驶应用的特性，这意味着一组特定的标准，例如所需的精度、可扩展性、动态性等。因此，考虑到 SLAM 系统的一般方法，我们建立了六个影响系统尺寸、精度的标准，和硬件实现。它们是：**算法类型、地图密度、全局优化、回环检测、可用性和嵌入式实现**：

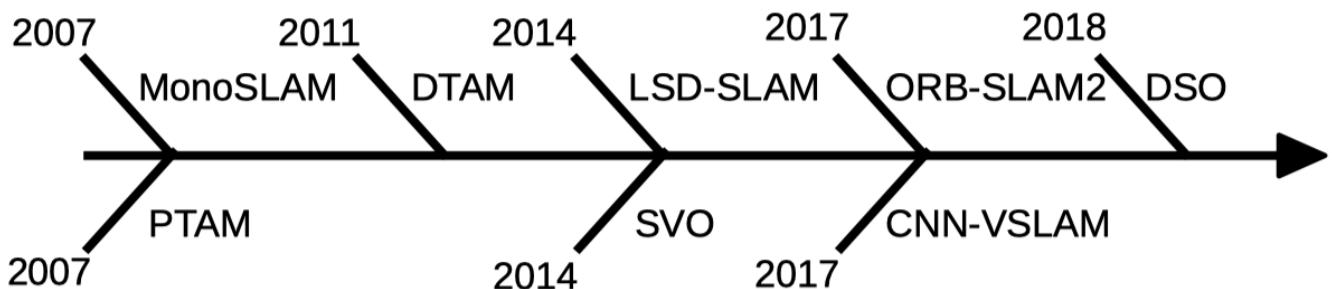
- **算法类型**：该标准表明算法采用的方法。对于纯视觉算法，我们将它们分为特征法、直接法与混合法。对于视觉惯性算法来说，它们必须是基于滤波或基于优化的方法。而 RGB-D 方法可以根据它们的跟踪方法进行划分，可以是直接的、混合的或基于特征的。
- **地图密度**：一般来说，稠密重建比稀疏重建需要更多的计算资源，对内存使用和计算成本有影响。另一方面，它提供了更详细和准确的重建，这可能是 SLAM 项目中的关键因素。

- 全局优化：SLAM 算法可能包括全局地图优化，这是指在考虑整个结构的一致性的情况下，搜索以补偿相机移动引入的累积误差的技术。
- 回环检测：回环检测是指 SLAM 算法识别之前检测到的图像以估计和校正传感器移动期间累积的漂移。
- 可用性：一些 SLAM 算法是由其作者提供并开源，或者由第三方提供它们的实现，以方便使用和复现。（原句“have their implementations made available by third parties, facilitating their usage and reproduction”，不确定自己翻译的对不对）
- 嵌入式实现：嵌入式 SLAM 实现是一个新兴领域，用于多种应用，尤其是机器人和汽车领域。该标准取决于每个算法的硬件约束和特殊性，因为其算法架构必须在能耗、内存和处理器占用率之间权衡。我们收集了在某些平台（比如微控制器和 FPGA 板等）完全嵌入式 SLAM 系统的主流文章。

在下文中，我们展示了选定的 SLAM 算法，根据其公开年份，这些算法被分别认为是三种路径内最具代表性的。

### 3.1. 纯视觉 SLAM

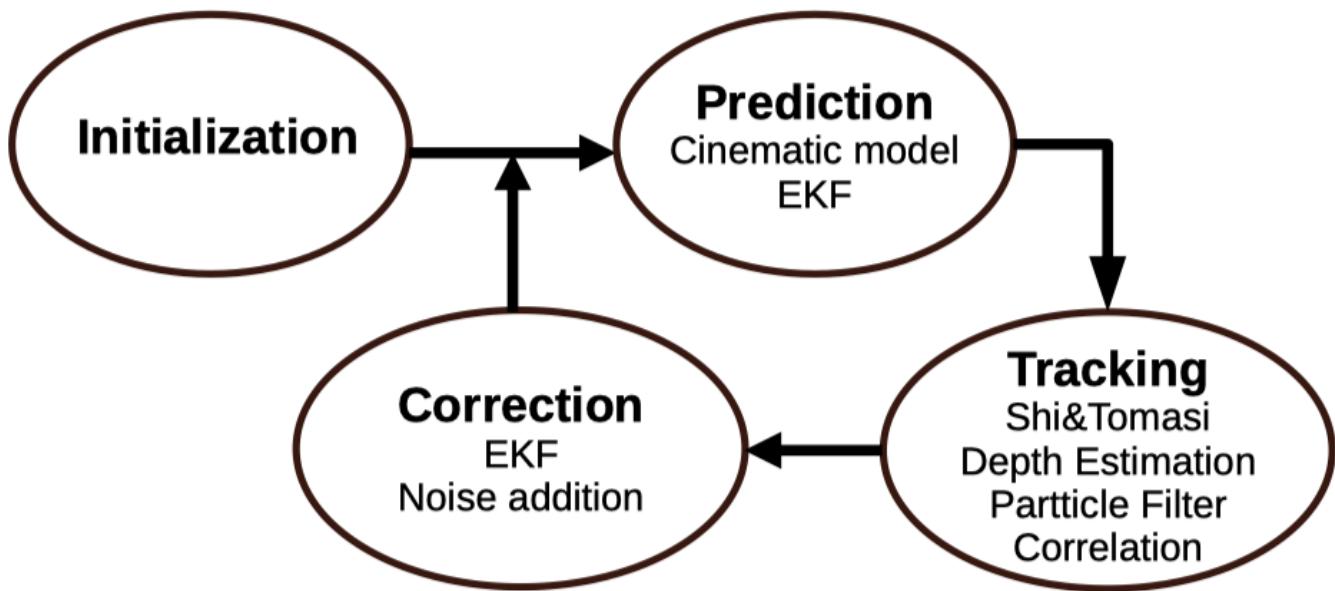
所选的纯视觉 SLAM 算法如图3所示，并在以下小节中进行解释。



**Figure 3.** Timeline representing the most representative visual-only SLAM algorithms.

#### 3.1.1 MonoSLAM (2007)

第一个单目 SLAM 算法是 MonoSLAM，由 David Son 等人<sup>[15:1]</sup>在 2007 年提出。算法的第一步包括系统的初始化。然后，它考虑恒速运动模型更新状态向量，其中使用扩展卡尔曼滤波器 (EKF) 实时估计相机运动和环境结构。该算法由图4 表示。



**Figure 4.** Diagram representing the MonoSLAM algorithm.

MonoSLAM 实时运行并由作者提供。此外，由于 MonoSLAM 是基于 EKF 的，这是一个已经被广泛讨论的主题，因此在文献中可以找到几个基于该算法的嵌入式实现。在[34,35] 中， Vincke 等人基于 MonoSLAM 算法，结合多个传感器和多处理器架构来评估其实现。在 [20] 中，作者使用 ARM + DSP + GPU 架构（OMAP3530 架构）来实现定位、重建和特征检测。他们将此架构与用于数据预处理和机器人控制的协处理器 ATMega168 相结合。在 [21] 中，他们将架构基于使用 OMAP4430 架构提供的多 CPU + GPU 的组合。作者在单核和双核 ARM 架构中实现了算法的不同任务，并比较了它们的性能。此外，他们使用 OMAP4430 提供的 ARM 和 NEON 处理器并行化了匹配和初始化任务。

**MonoSLAM** 的初始化步骤需要一个已知的目标，这并不总是可以获得的。此外，算法的复杂性随着环境的大小成比例地增加。该算法既不采用全局优化技术，也不采用回环检测检测。最后，它只重建地标地图，这对于需要更准确重建的应用程序来说可能是一个缺点。

### 3.1.2 并行跟踪和建图(PTAM, Parallel Tracking and Mapping) (2007)

另一个先驱算法是并行跟踪和建图 (PTAM) [22] 算法。

**PTAM** 是第一个将 **Tracking** 和 **Mapping** 分为两个不同的线程算法，并将关键帧的概念应用于建图线程。首先，建图线程执行地图初始化。当相机移动的时候，新的关键帧被添加到系统中，初始地图被扩展。两个连续关键帧之间的三角化计算新点的深度信息。跟踪线程计算相机姿势，并且对于每个新帧，它估计一个初始姿势以执行地图点在图像上的投影。**PTAM** 使用对应关系(**correspondences**)来计算通过最小化重投影误差 (**reprojection error**) 的相机位姿。

图5表示其执行的步骤。

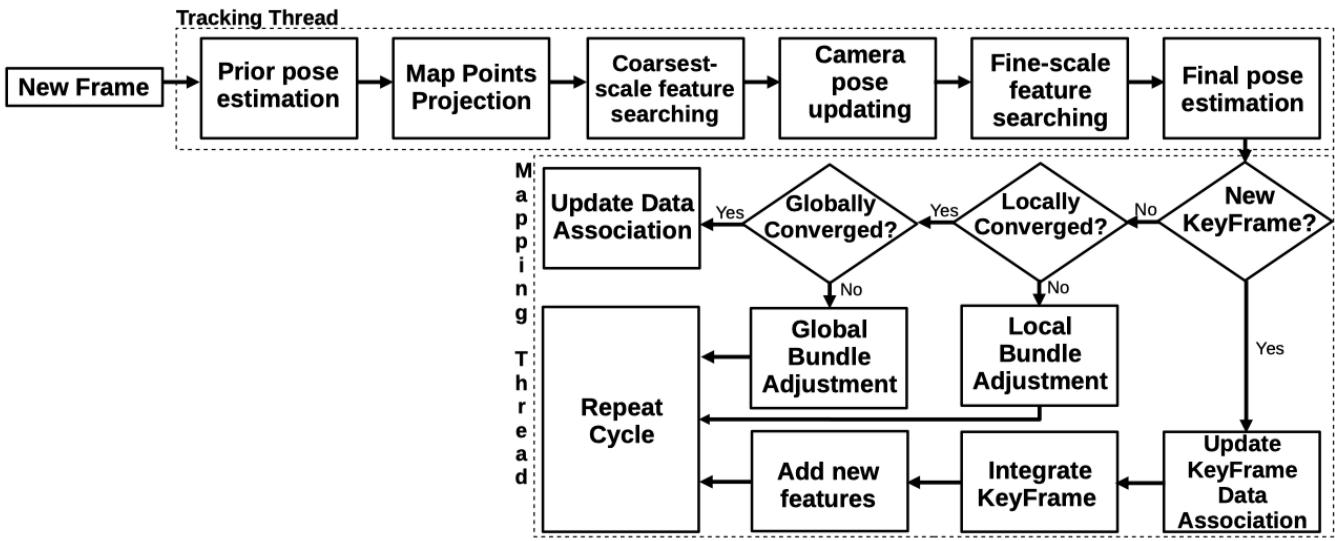


Figure 5. Diagram representing the PTAM algorithm.

PTAM 允许地图持有大量特征，并执行全局优化。尽管有这些优点，但由于捆绑调整 (BA, Bundle Adjustment) 步骤，PTAM 算法呈现出很高的复杂性。此外，它不计入回环检测，并且生成的地图更适合识别地标。此外，它需要用户交互建立初始关键帧，呈现出不可忽略的力量消耗，这使得它不适合低成本的嵌入式系统<sup>[23]</sup>。

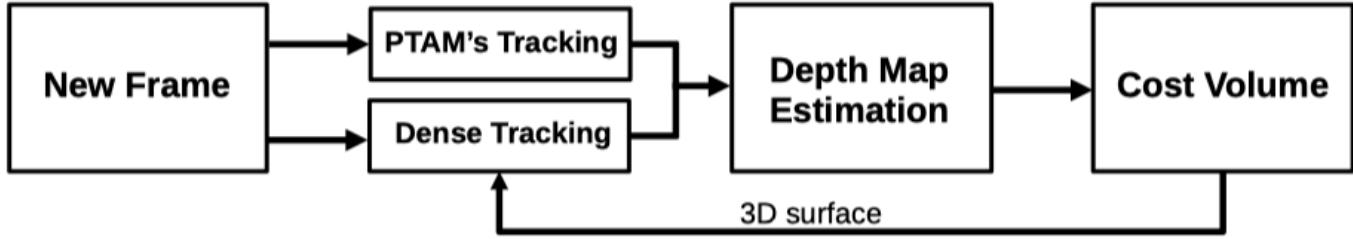
### 3.1.3 稠密跟踪和建图 (DTAM, Dense Tracking and Mapping) (2011)

Newcombe 等人提出的稠密跟踪和建图 (DTAM)<sup>[24]</sup>，是文献中第一个完全直接的方法。该算法分为两个主要部分：稠密建图和稠密跟踪。

(下面一句好难翻哦，google直接摆烂，找了deepl，感觉翻的不错，原句是：“The first stage searches to estimate the depth values by defining data cost volume representing the average photometric error of multiple frames computed for the inverse depth of the current frame.”)

第一阶段通过定义数据成本量来搜索估计深度值，数据成本量代表了为当前帧的反深度计算的多个帧的平均光度误差。最小化光度误差的逆深度被选择用于重建。在稠密跟踪阶段，DTAM通过对齐在虚拟摄像机中投射的密集模型的图像和当前帧来估计运动参数。

图6显示了 DTAM 算法的一般视图。

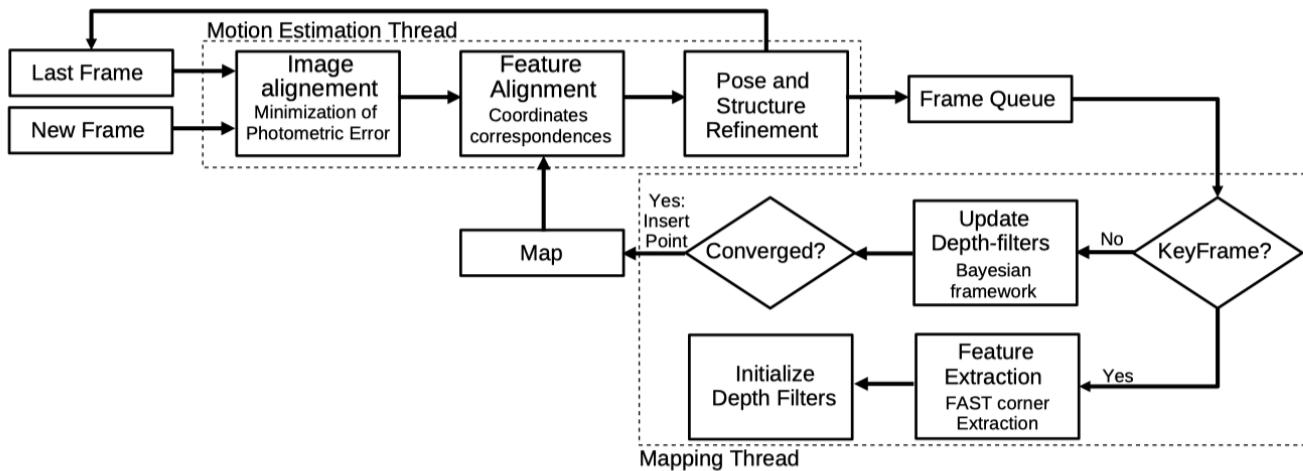


**Figure 6.** Diagram representing the DTAM algorithm.

该算法提供了准确而详细的重建，但这密度重建的水平会影响存储和处理数据的计算成本。因此，为了实现实时操作，该算法需要最先进的GPU [4:4]。[25]中的作者采用了不同 iPhone 的 CPU + GPU 架构模型来实现基于 DTAM 的全稠密算法。他们将 CPU 用于跟踪任务，GPU 用于深度估计和帧融合。DTAM 不实施回环检测技术或全局优化。

### 3.1.4 半直接视觉里程计 (SVO, Semi-Direct Visual Odometry, 2014)

半直接视觉测绘 (SVO) 算法 [26] 结合了特征法和直接法的优点。该算法分为两条主线：运动估计和映射。第一条线搜索以估计传感器的运动参数，包括最小化光度误差。映射线程是基于概率深度仪的，它搜索以估计每个二维特征的最佳深度值。当该算法达到低不确定性时，它就会在重建中插入三维点，如图7所示。



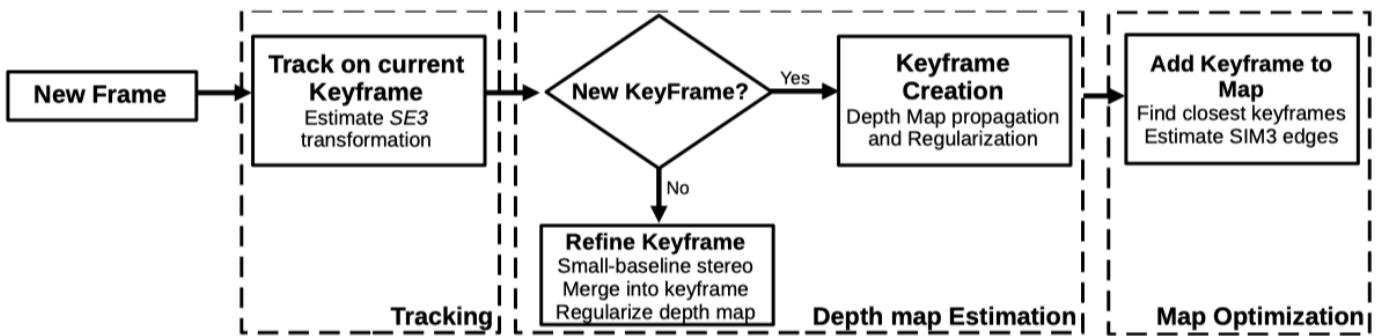
**Figure 7.** Diagram representing the SVO algorithm. Adapted from [40].

SVO 允许直接像素对应，也支持概率映射方法。此外，该算法能够以高帧率运行，因为它不需要为每一帧提取特征 [27]，这使得它能够在低成本的嵌入式系统中运行，如 [26:1] 所考虑的嵌入式平台，包括 Odroid-U2。然而，由于短期的数据关联，SVO 的精度有限 [28]。SVO 并没有实现全局优化技术或回环检测。作者已经提出了一个 SVO 的扩展版本，即 SVO 2.0 [29]，其中该算法能够处理立体数据和 IMU 信息。

### 3.1.5 大规模直接单目 SLAM (LSD-SLAM, Large-Scale Direct Monocular SLAM, 2014)

大规模直接单目SLAM (LSD-SLAM) [30] 基于直接法进行半稠密的重建。该算法由三个主要步骤组成：跟踪、深度图估计和地图优化。第一步是最小化测光误差以估计传感器的姿态。接下来，LSD-SLAM在深度图估计步骤中进行关键帧选择。如果它在算法中增加了一个新的关键帧，它就初始化其深度图；否则，它就通过执行几个小基线的立体比较来重新确定当前关键帧的深度图。最后，在地图优化步骤中，LSD-SLAM将新的关键帧纳入地图中，并通过应用姿势图优化算法对其进行优化。图8说明了这个过程。

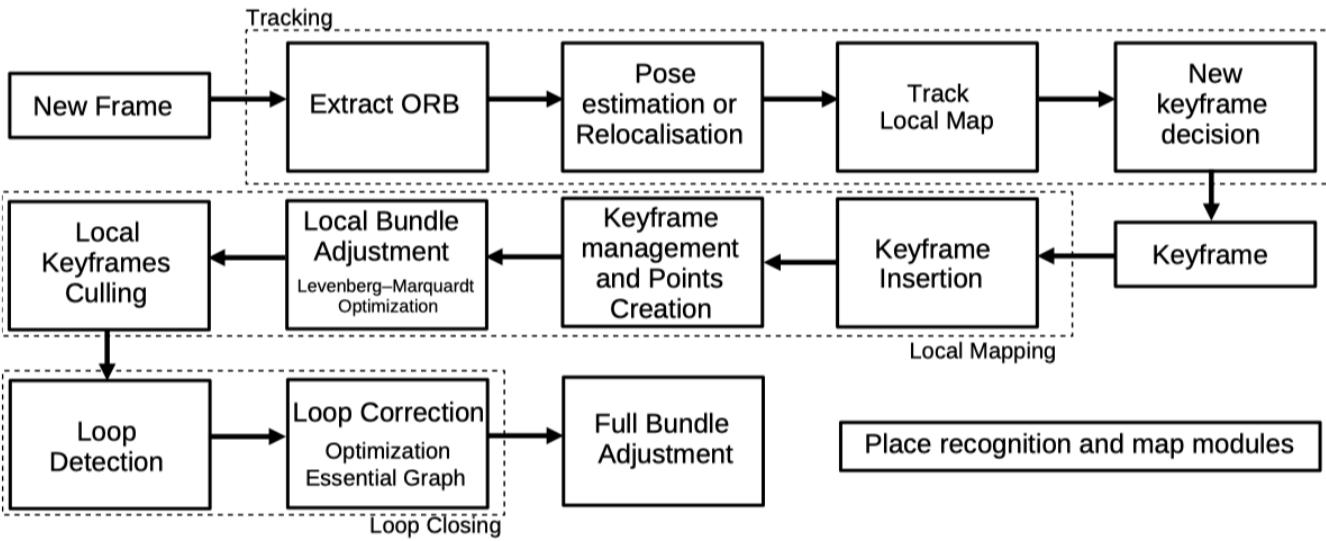
这项技术允许实时构建大规模地图，并采用全局优化和回环检测。此外，通过将无需特征提取（直接法的特点）与半稠密重建相结合，该方法提高了其效率，使嵌入式实现成为可能。Boikos和Christos-Savvas在[29,43]中使用CPU + FPGA架构来实现LSD-SLAM算法。在[17:1]中，作者在FPGA上实现了两个加速器，以执行跟踪线程中更昂贵的任务；也就是雅各布计算以及残差和权重计算。ARM CPU被用来实现该算法的其他任务。在[31]中，作者在FPGA上实现了直接跟踪线程，而CPU则负责内存、硬件控制和参数设置。LSD-SLAM地图估计基本上是基于姿势图的优化[28:1]，该算法取得的精度低于其他算法，如PTAM和ORB-SLAM[27:1]。



**Figure 8.** Diagram representing the LSD-SLAM algorithm. Adapted from [24].

### 3.1.6 ORB-SLAM 2.0 (2017)

ORB-SLAM2算法[32]起源于ORB-SLAM[27:2]，被认为是基于特征的算法中最先进的算法。它在三个平行线程中工作：跟踪、建图和回环检测。第一个线程通过寻找特征对应关系和最小化重投影误差来定位传感器。本地映射线程负责地图管理操作。最后一个线程，即回环检测，负责检测新的循环并纠正循环中的漂移误差。在处理完这三个线程后，该算法还考虑了整个结构，并通过执行全束调整（full Bundle Adjustment）来估计运动一致性。图9表示构成该算法的线程。

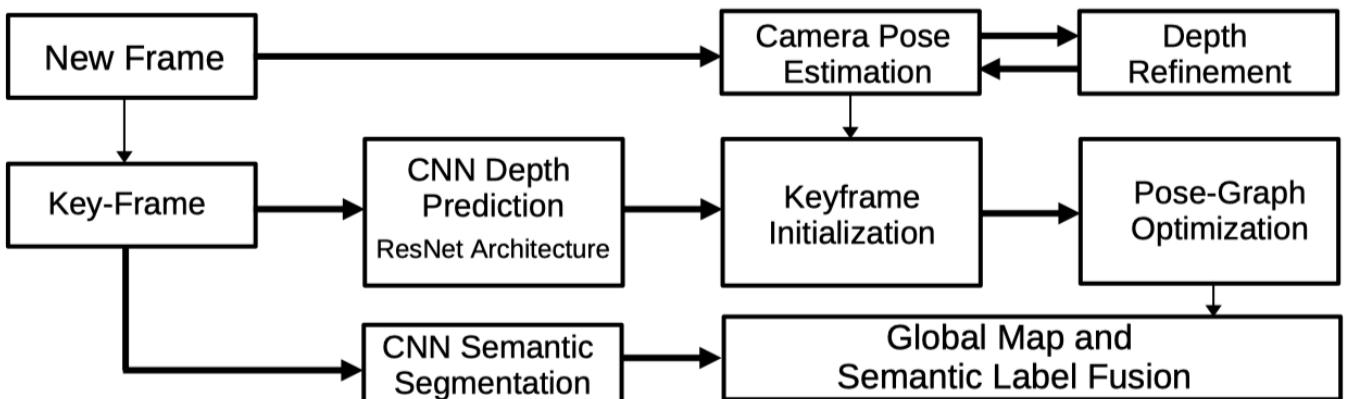


**Figure 9.** Diagram representing the ORB-SLAM 2.0 algorithm. Adapted from [41].

ORB-SLAM2考虑了单眼、立体和RGB-D方法，并实施了全局优化和回环检测技术。尽管如此，如果系统不能识别高相似度的帧，跟踪失败的情况可能会导致丢失状态[33]。此外，这种方法需要在处理图像时以相同的帧速率获取图像，这使得嵌入式平台的实时操作变得很困难[34]。尽管在文献中可以找到一些嵌入式的实现方法，但这是一个事实。Yu等人[35]使用CPU来运行ORB-SLAM算法，Abouzahir等人[34:1]在不同的基于CPU和GPU的平台上实现该算法，并评估了平台上每个线程的性能。

### 3.1.7 CNN-SLAM (2017)

CNN-SLAM[36]是最早提出基于卷积神经网络（CNN）的实时SLAM系统的作品之一。该算法可分为两个不同的管道：一个应用于每个输入帧，另一个应用于每个关键帧。第一个管道负责通过最小化当前帧和最近的关键帧之间的测光误差来进行摄像机的姿态估计。同时，对于每个关键帧，深度由CNN来预测。此外，该算法还预测了每一帧的语义分割。在这些处理步骤之后，该算法进行了位姿图的优化，以获得全局优化的位姿估计，如图10所示。



**Figure 10.** Diagram representing the CNN-SLAM algorithm. Adapted from [48].

这种算法不受绝对规模的限制，因为它使用深度预判来进行规模估计<sup>[36:1]</sup>。此外，它还包括全局优化和回环检测。作者需要采用CPU+GPU架构来实时运行该算法。

### 3.1.8 直接稀疏测距法 (DSO, Direct Sparse Odometry, 2018)

直接稀疏测距 (DSO) 算法<sup>[37]</sup>结合了直接法和稀疏重建。DSO算法考虑了一个最近的帧的窗口。它通过应用优化关键帧窗口和逆深度图的局部BA来进行连续优化。该算法将图像分为若干块，并选择最高强度的点。DSO在优化过程中考虑了曝光时间和镜头失真，以增加算法的鲁棒性。最初，该算法不包括全局优化或回环检测，但Xiang等人<sup>[38]</sup>提出了DSO算法的扩展，包括回环检测和姿势图优化。图11表示了DSO的主要步骤。

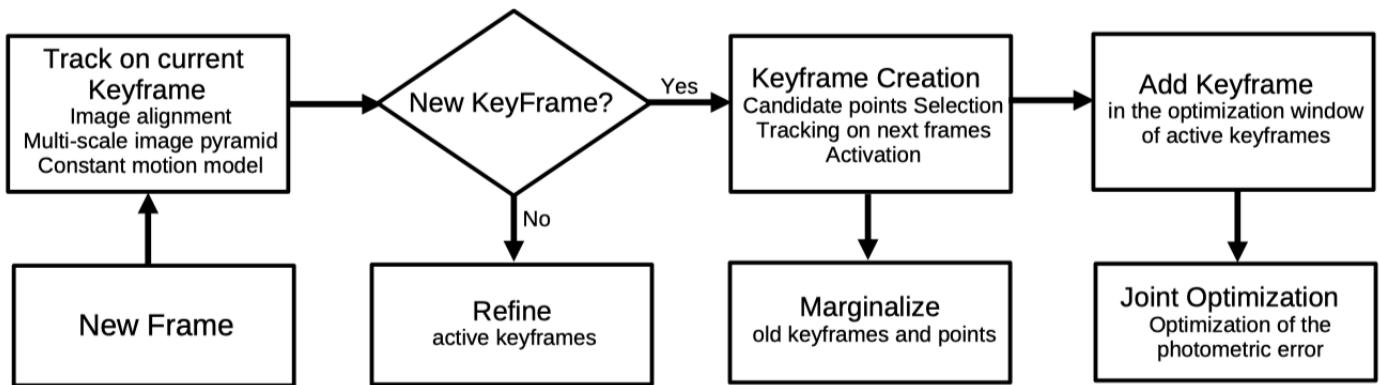


Figure 11. Diagram representing the DSO algorithm.

### 3.1.9 小结

在本节中，我们介绍了主要的基于纯视觉的SLAM算法。表1总结了所介绍的纯视觉SLAM算法的主要特点和分析标准。

Table 1. Main aspects related to the visual-only SLAM approaches.

Method	Type	Map Density	Global Optim. *	Loop Closure	Embed. Implem. **	Availability
MonoSLAM	Feature-based	Sparse	No	No	[34,35]	[50]
PTAM	Feature-based	Sparse	Yes	No	[51]	[52]
DTAM	Direct	Dense	No	No	[39]	[53]
SVO	Hybrid	Sparse	No	No	[40]	[54]
LSD	Direct	Semi-dense	Yes	Yes	[29,43]	[55]
ORB-SLAM	Feature-based	Sparse	Yes	Yes	[46,47]	[56]
CNN-SLAM	Direct	Semi-dense	Yes	Yes	-	[57]
DSO	Direct	Sparse	No	No	-	[58]

\* Global Optimization. \*\* Embedded Implementation.

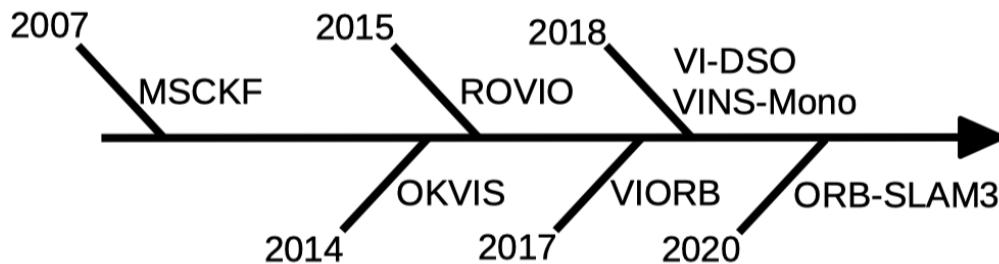
每种方法的主要优点和缺点都已经分别阐述。从总体上看，基于视觉的SLAM算法可以被认为是一个经过充分探索的领域，因为大多数算法是由作者提供的，这也对文献中发现的嵌入式SLAM实现产生了影响。

表1中列出的嵌入式实施方案考虑了完整的SLAM算法的实施和不对最初提出的技术进行基本修改的工作。然而，有可能在文献中找到一些基于所提出的算法的基本概念的嵌入式实现方法。例如，MonoSLAM原则已被用于开发和实现其他几个SLAM的SoC实现，如Piat等人最近提出的异构架构<sup>[39]</sup>。

此外，可以注意到基于CNN的SLAM算法的不断发展。除了所提出的CNN-SLAM，文献中还发现了其他算法，如CNN-SVO<sup>[16:1]</sup>算法，该算法使用深度预测来初始化深度滤波器。自2019年推出人工智能加速器赛灵思深度学习处理器单元（Xilinx Deep-learning Processor Unit）<sup>[40]</sup>以来，基于CNN的SLAM算法的硬件实现的发展一直在增长。这种硬件已经使基于CNN算法的嵌入式实现取得了进展：其中一个例子是在<sup>[41]</sup>中提出的工作，它使用FPGA平台来执行基于CNN的特征提取器。

## 3.2 视觉惯性SLAM (VI-SLAM, Visual-Inertial SLAM)

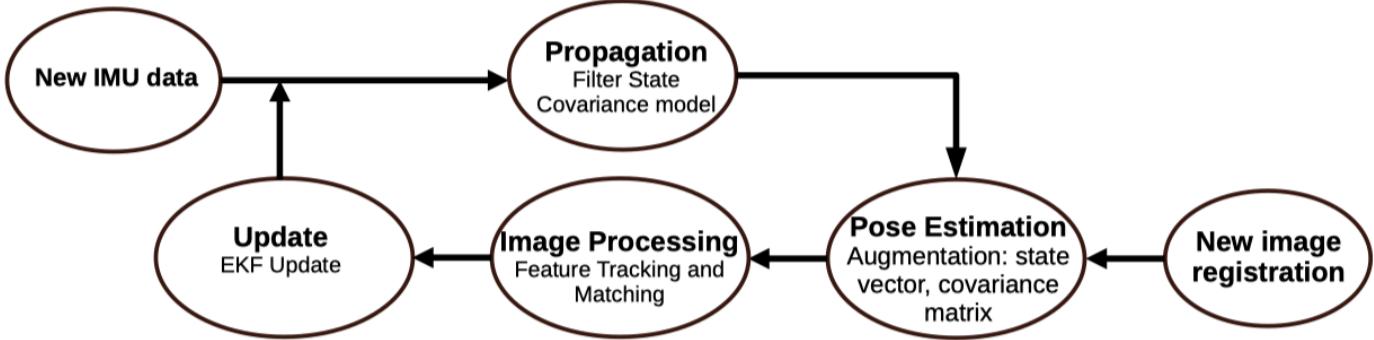
图12展示了代表所选视觉-惯性算法的时间线，并在以下各小节中对这些算法进行了解释。



**Figure 12.** Timeline representing the most representative visual-inertial SLAM algorithms.

### 3.2.1 多状态约束卡尔曼滤波器 (MSCKF, Multi-State Constraint Kalman Filter, 2007)

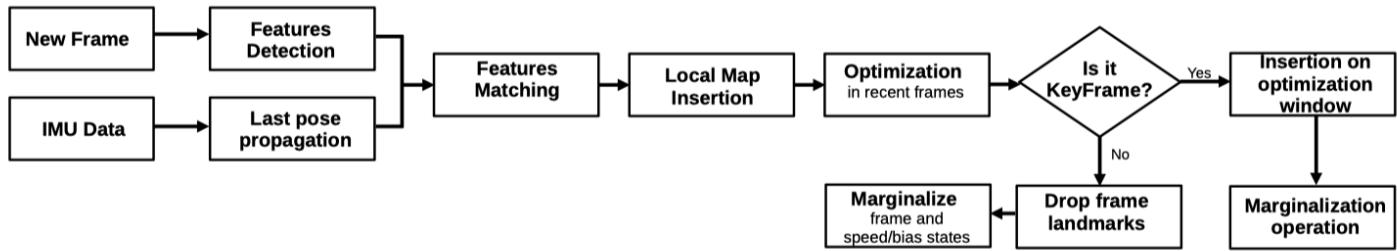
多状态约束卡尔曼滤波器 (MSCKF) <sup>[42]</sup>可以使用单眼和立体相机来实现<sup>[43]</sup>。该算法的管道包括三个主要步骤：传播、图像注册和更新。在第一步中，MSCKF考虑了连续时间IMU模型的离散化，以获得滤波器的状态和协方差的传播。然后，图像注册在每次记录新的图像时进行状态增强。这一估计被添加到状态和协方差矩阵中，以启动图像处理模块（特征提取）。最后，该算法进行了滤波器的更新。图13表示该算法MSCKF被认为是文献中最快的基于滤波器的方法之一<sup>[44]</sup>，这是其计算成本低的结果<sup>[43:1]</sup>，这使得该算法适用于嵌入式实现。Delmerico和Scaramuzza<sup>[45]</sup>使用基于CPU架构的不同硬件平台来实现视觉-惯性SLAM算法作者在三种不同的嵌入式板上实现了该算法—Intel NUC、Up Board和ODROID。然而，该算法进行的雅各布计算可能会导致不一致和准确性的损失<sup>[46]</sup>。



**Figure 13.** Diagram representing the MSCKF algorithm.

### 3.2.2. 基于开放关键帧的视觉-惯性SLAM (OKVIS, Open Keyframe-Based Visual-Inertial SLAM (2014))

基于开放关键帧的视觉-惯性SLAM (OKVIS) [47]是一种基于优化的方法。它将IMU数据和重投影结合到一个目标函数中，使算法能够共同优化加权重投影误差和IMU的时间误差。该算法建立了一个局部地图，然后根据关键点的匹配区域来选择后续的关键帧。该算法的描述如图14所示与其他算法相比。

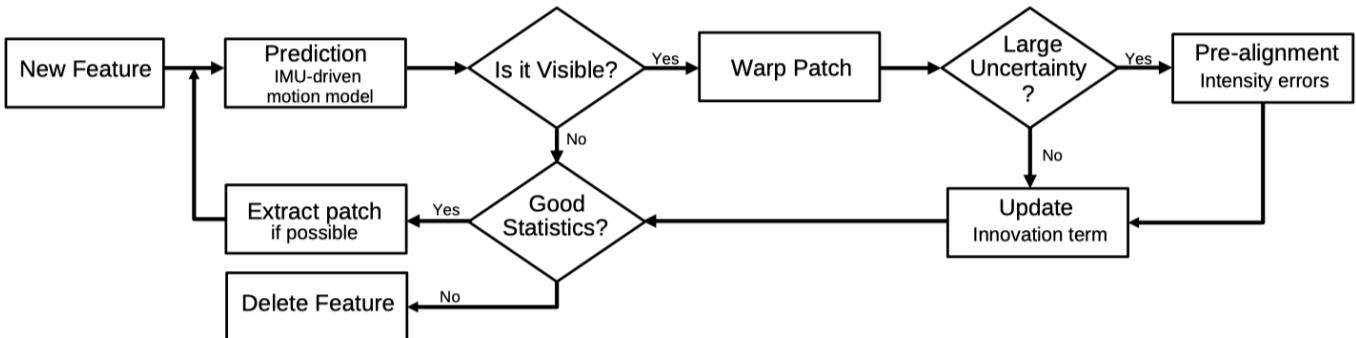


**Figure 14.** Diagram representing the OKVIS algorithm.

OKVIS算法的内存占用率较低（这将在下面的小节中解释），如VINS-Mono、VIORB和ROVIO<sup>[10:3]</sup>，这使得其嵌入式实现成为可能。已经提到，Delmerico和Scaramuzza的工作<sup>[45:1]</sup>使用不同的CPU平台来实现OKVIS算法。然而，为了在Up Board和ODROID中实现实时性能，作者需要减少关键点的数量、关键帧窗口和与IMU相关的帧。Nikolic等人<sup>[48]</sup>使用FPGA-CPU架构来评估OKVIS算法的性能作者利用FPGA上的逻辑块来实现图像处理技术，并加速了关键点的检测过程然而，事实证明，该算法的准确性不如其他算法<sup>[10:4]</sup>。

### 3.2.3. Robust Visual Inertial Odometry (2015)

Robust Visual Inertial Odometry (ROVIO) 算法<sup>[49]</sup>是另一种基于滤波器的方法，它采用EKF方法，与其他基于滤波器的方法类似，它使用IMU数据进行状态传播，并使用摄像机数据进行滤波器更新。然而，除了进行特征提取外，ROVIO还对特征周围的多级斑块进行提取，如图15所示。预测和更新步骤使用这些补丁来获得创新项，即计算框架和多级补丁投射到框架中的误差。

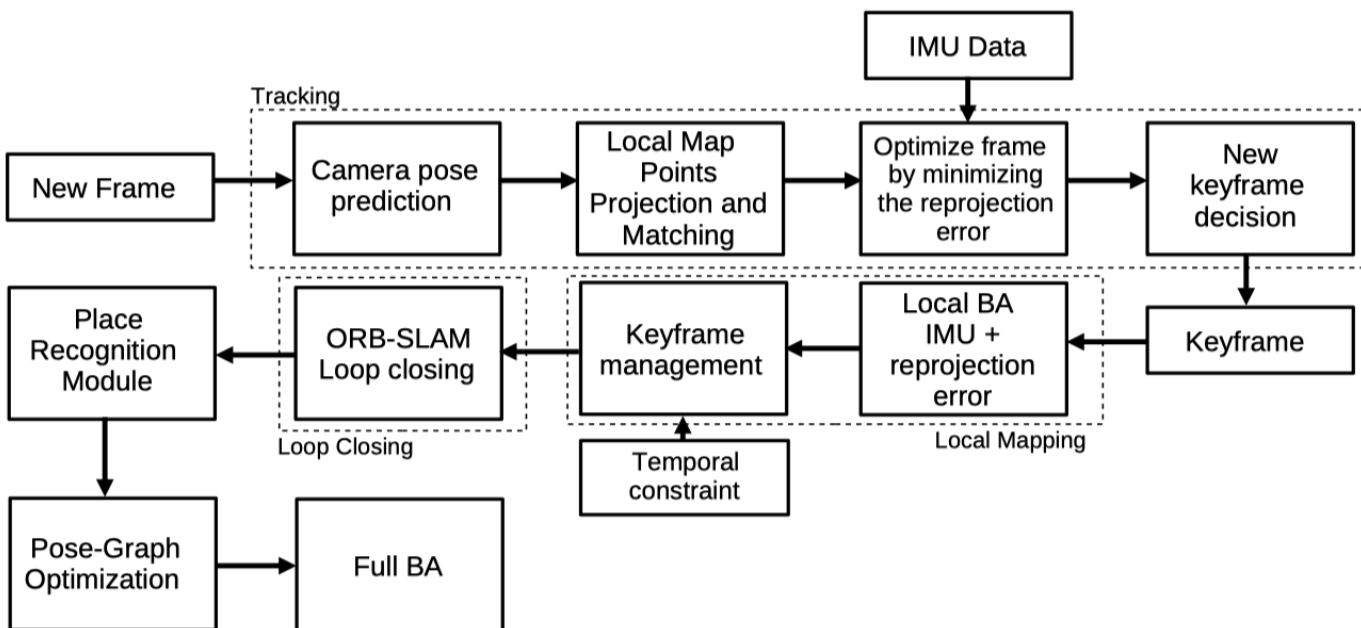


**Figure 15.** Diagram representing the feature handling performed by the ROVIO algorithm. Adapted from [69].

ROVIO算法在低资源利用率下实现了良好的准确性和稳健性[18,65]，适合于嵌入式实现<sup>[45:2]</sup>。然而，事实证明，该算法对每一帧的处理时间更为敏感<sup>[45:3]</sup>，并且不如其他算法准确，例如VI-DSO<sup>[50]</sup>。

### 3.2.4. 视觉-惯性ORB-SLAM (VIORB, Visual Inertial ORB-SLAM, 2017)

视觉-惯性ORB-SLAM (VIORB) 算法<sup>[51]</sup>是基于上文描述的ORB-SLAM算法<sup>[32:1]</sup>。因此，该系统也有三个主要线程：跟踪、建图和回环检测。在VIORB中，跟踪线程估计传感器的位姿、速度和IMU偏差。此外，该线程对匹配点的重投影误差和IMU误差数据进行联合优化。建图线程考虑到IMU的操作，采用了不同的剔除策略。最后，回环检测线程实现了一个位置识别模块，以识别之前识别过的关键帧。此外，该算法还进行了优化，以最小化累积误差。图16试图说明ORB-SLAM算法（见图9）和其视觉惯性版本之间的主要区别。

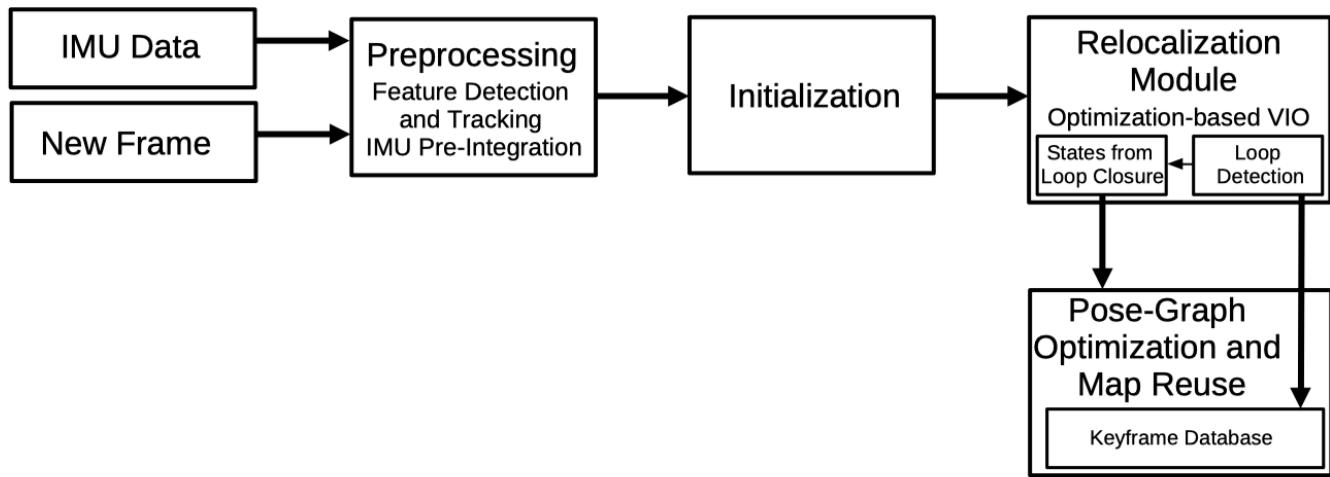


**Figure 16.** Diagram representing the VIORB algorithm.

VIORB算法是第一个采用地图重用的视觉惯性方法，它具有高性能的精度[64,70,72]和内存使用[10:5]。尽管如此，IMU的初始化需要10到15秒[51:1]，而且没有发现嵌入式的实现方式。在[28:2]中，作者提出了ORB-SLAM3算法，该算法基于ORB-SLAM2和VIORB算法与其前身VIORB相比，该系统的初始化时间有所减少。

### 3.2.5. 单目视觉惯性系统 (VINS-Mono, Monocular Visual-Inertial System, 2018)

单目视觉惯性系统 (VINS-Mono) [52]是一个单目视觉惯性状态估计器。它从一个负责特征提取和跟踪的测量过程开始，并对各帧之间的IMU数据进行预整合。然后，该算法执行一个初始化过程，为一个非线性优化过程提供初始值，使视觉和惯性误差最小。VINS还实现了一个重新定位和一个姿势图形优化模块，将IMU测量值和特征观测值合并起来。图17说明了VINS-Mono算法的情况。



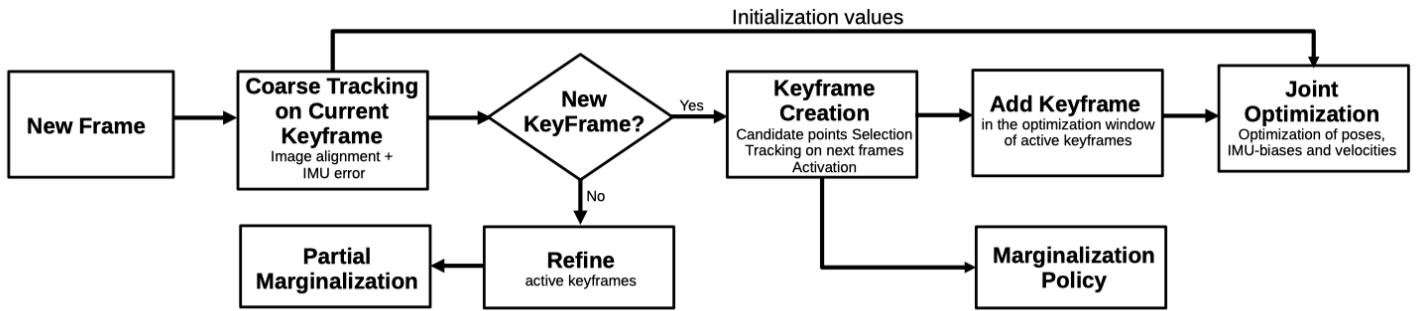
**Figure 17.** Diagram representing the VINS-Mono algorithm.

该算法还可以考虑采用双目和立体的方法[53]。与其他算法相比，VINS-Mono已经显示出很高的精确度。然而，与ROVIO、VIORB和OKVIS等算法相比，它的内存使用率最高[10:6]。尽管如此，由于在优化过程中只考虑了最新IMU状态下的位姿和速度，该算法仍然证明了其在嵌入式实现中的适用性[52:1]。

南川注：看了一下github，显然已经有基于VINS-Mono的扩展版本VINS-fusion了，mono只支持单目，fusion还支持双目和rgb-d，见：<https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>

### 3.2.6. 视觉惯性直接稀疏测距 (VI-DSO, Visual-Inertial Direct Sparse Odometry, 2018)

视觉惯性直接稀疏测距 (VI-DSO) 算法[50:1]是基于已经提出的DSO算法[37:1]。该算法搜索最小化一个能量函数，该函数结合了测光误差和惯性误差，建立在一个非线性动态模型之上。图18显示了VI-DSO算法的概况，展示了它与DSO技术的主要区别。



**Figure 18.** Diagram representing the VI-DSO algorithm.

VI-DSO是DSO的扩展，它考虑了惯性信息，因此比原来的DSO和其他算法，如ROVIO<sup>[50:2]</sup>，具有更好的准确性和稳健性。然而，初始化程序依赖于捆绑调整(Bundle Adjustment)，这使得初始化很慢<sup>[28:3]</sup>。该算法不进行全局优化和回环检测，而且在文献中没有发现嵌入式的实现。

### 3.2.7. ORB-SLAM3 (2020)

已经提到的ORB-SLAM3算法<sup>[54]</sup>是一种结合ORB-SLAM和VIORB算法的技术。与其前身一样，该算法分为三个主要线程：跟踪、建图，以及回环检测和地图合并（而非仅仅是回环检测）。

此外，ORB-SLAM3还维护一个名为Atlas的多地图表示法，它维护一个由跟踪线程使用的主动地图，以及用于重新定位和地点识别的非主动地图。前两个线程遵循与VIORB相同的原则，而地图合并被添加到最后一个线程中。回环检测和地图合并线程使用Atlas中的所有地图来识别共同部分，并根据重叠区域的位置，执行循环校正或合并地图并改变活动地图。

ORB-SLAM3的另一个重要方面涉及到所提出的初始化技术，该技术依赖于单独应用于视觉和惯性估计的Maximum-a-Posteriori算法，随后进行联合优化。

该算法可用于单目、立体和RGB-D相机，并实现了全局优化和回环检测技术。然而，作者在<sup>[55]</sup>中展示了ORB-SLAM3在线性能的显著错误结果。在<sup>[56]</sup>中，该算法获得了良好的性能，但未能处理所有的序列，并且在户外序列中获得了不准确的估计。

### 3.2.8 小结

本节介绍了七种主要的视觉-惯性SLAM算法，并对每种算法进行了单独分析。表2总结了所提出的视觉-惯性SLAM算法的主要特点和分析标准。

**Table 2.** Main aspects related to the visual-inertial SLAM approaches. All approaches present tightly coupled sensor fusion.

Method	Type	Map Density	Global Optim. *	Loop Closure	Embed. Implem. **	Availability
MSCKF	Filtering-based	Sparse	No	No	[65]	[78,79]
OKVIS	Optimization-based	Sparse	No	No	[65,68]	[80]
ROVIO	Filtering-based	Sparse	No	No	[65]	[81]
VINS	Optimization-based	Sparse	Yes	Yes	[65,74]	[82]
VIORB	Optimization-based	Sparse	Yes	Yes	-	-
VI-DSO	Optimization-based	Sparse	No	No	-	[83]
ORB-SLAM3	Optimization-based	Sparse	Yes	Yes	-	[84]

\* Global Optimization. \*\* Embedded Implementation.

在一般分析中，在基于视觉的**SLAM**算法中加入**IMU**的主要目的是增加系统的稳健性，这一点已经被证明是正确的[2,22,70]。

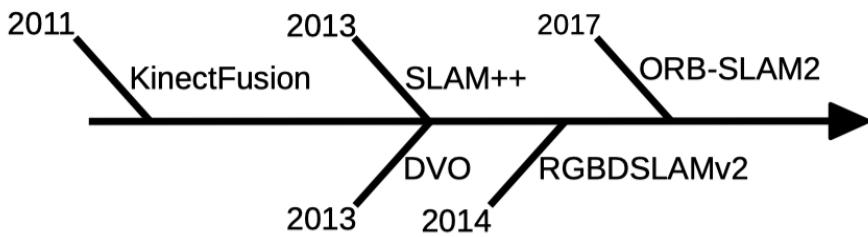
我们观察到这些开源算法得到了更多的文献反馈，这直接推动了后续的嵌入式实现。不过与纯视觉不同的是，我们没有找到VIORB算法的嵌入式版本，因为原文没有提供开源版本；而最近的版本，即开源的ORB-SLAM3，是在2020年发表的[28:4]。至于DSO算法的惯性版本，作者没有提供一个开源的实现；但是，可以找到第三方的实现[57]，尽管它需要优化。

基于视觉-惯性的SLAM方法是一个不断发展的领域，最近有几篇文章已经发表，将IMU技术与大量的传感器相结合[85-87]。基于对纯视觉SLAM技术的研究，我们可以找到几篇能有效率提高基于VI的SLAM算法的初始化步骤的性能的文章[75,88,89]。

### 3.3. RGB-D SLAM

南川注：由于项目不需要，因此本节跳过翻译

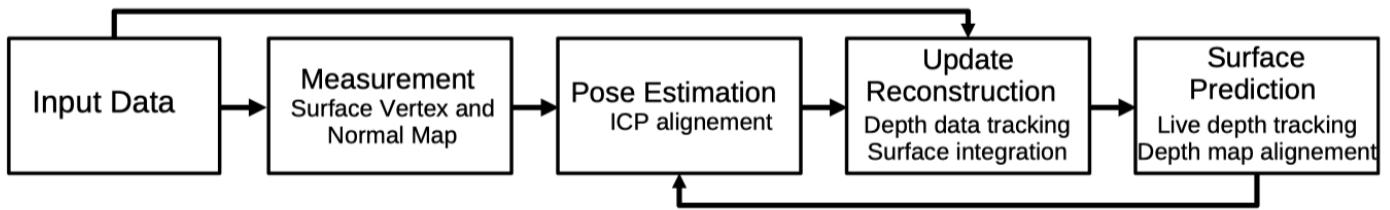
The most representative SLAM algorithms based on RGB-D sensors, i.e., considering RGB images and depth information directly, are presented in Figure 19, according to their published years, and explained in the following subsections.



**Figure 19.** Timeline representing the most representative RGB-D-based SLAM algorithms.

#### 3.3.1. KinectFusion (2011)

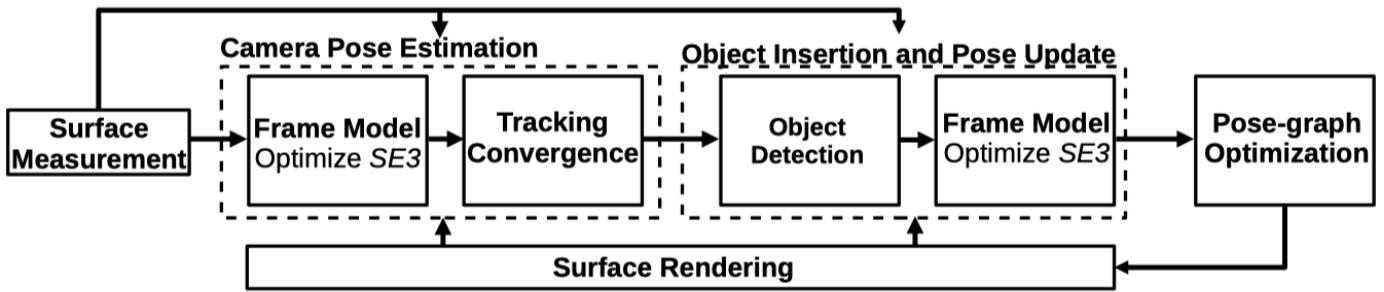
The KinectFusion algorithm [58] was the first algorithm based on an RGB-D sensor to operate in real-time. The algorithm includes four main steps: the measurement, pose estimation, reconstruction update, and surface prediction. In the first step, the RGB image and depth data are used to generate a vertex and a normal map. In the pose estimation step, the algorithm applies the ICP alignment between the current surface and the predicted one (provided by the previous step). Then, the reconstruction update step integrates the new depth frame into the 3D reconstruction, which is raycasted into the new estimated frame to obtain a new dense surface prediction. The KinectFusion algorithm is capable of good mapping in maximum medium-sized rooms [58:1]. However, it accumulates drift errors, since it does not perform loop closing [59]. Nardi et al., in [60], propose an implementation for the KinectFusion and test it in different CPU- and GPU-based platforms. Bodin et al. [61] use the framework proposed by [60:1] to implement the KinectFusion in two different CPU and GPU platforms. An overview of the steps performed by the algorithm is shown in Figure 20.



**Figure 20.** Diagram representing the KinectFusion algorithm. Adapted from [90].

### 3.3.2. SLAM++ (2013)

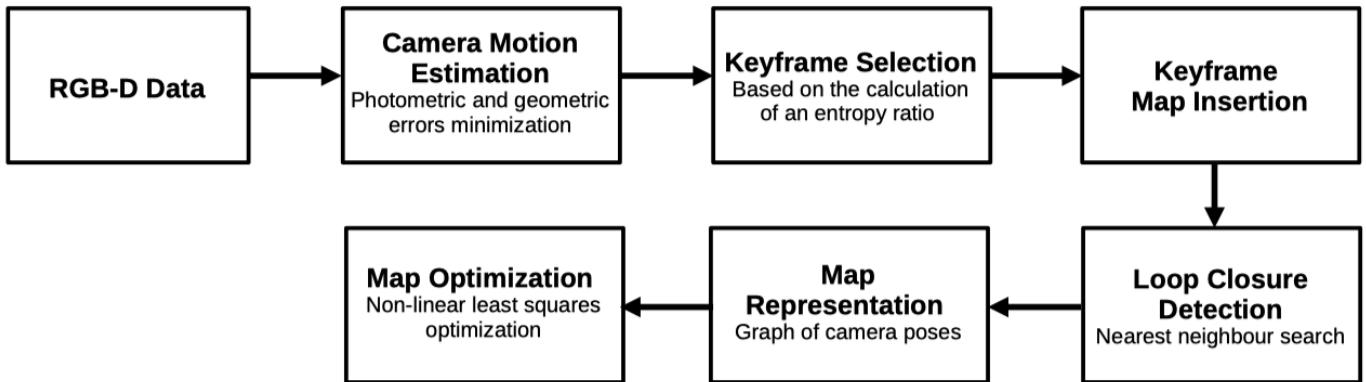
The SLAM++ algorithm [62] is an object-oriented SLAM algorithm that takes advantage of previously known scenes containing repeated objects and structures, such as a classroom. After the system initialization, SLAM++ operates in four steps: camera pose estimation, object insertion, and pose update, pose-graph optimization, and surface rendering. The first step estimates the current camera pose by applying the ICP algorithm, considering dense multi-object prediction in the current SLAM graph. Next, the algorithm searches to identify objects in the current frame using the database information. The third step inserts the considered objects in the SLAM graph by performing a pose-graph optimization operation. Finally, the algorithm renders the objects in the graph, as shown in Figure 21. SLAM++ performs loop closure detection and, by considering the object's repeatability, it increases its efficiency and scene description. Nevertheless, the algorithm is most suitable for already known scenes.



**Figure 21.** Diagram representing the SLAM++ algorithm. Adapted from [94].

### 3.3.3. Dense Visual Odometry (2013)

The dense visual odometry SLAM (DVO-SLAM) algorithm, proposed by Kerl et al. [63], is a keyframe-based technique. It minimizes the photometric error between the keyframes to acquire the depth values and pixels coordinates, as well as camera motion. The algorithm calculates, for each input frame, an entropy value that is compared to a threshold value. The same principle is used for loop detection, although it uses a different threshold value. The map is represented by a SLAM graph where the vertex has camera poses, and edges are the transformations between keyframes. This algorithm is robust to textureless scenes and performs loop closure detection. The map representation relies on a representation of the keyframes, and the algorithm does not perform an explicit map reconstruction. Figure 22 shows an overview of the DVO algorithm.

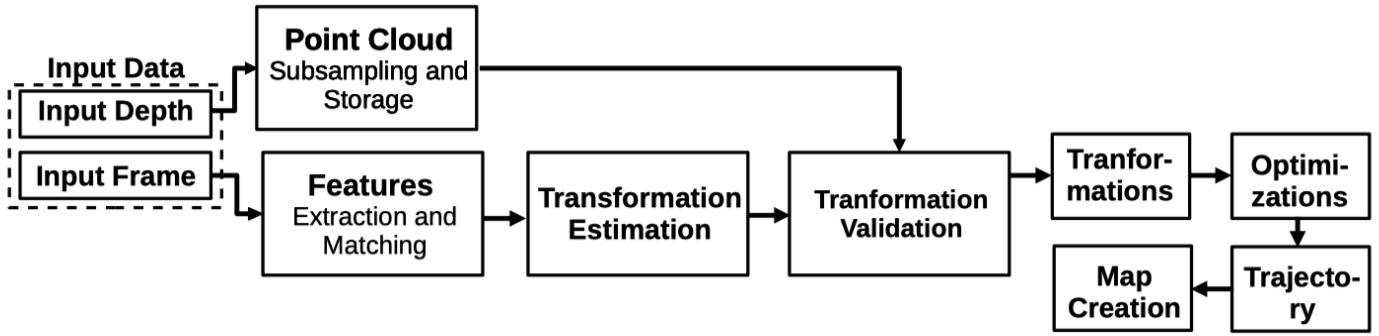


**Figure 22.** Diagram representing the DVO algorithm.

### 3.3.4. RGBDSLAMv2 (2014)

The RGBDSLAMv2 [64] is one of the most popular RGB-D-based algorithms and relies on feature extraction. It performs the RANSAC algorithm to estimate the transformation between the matched features and the ICP algorithm to obtain pose estimation. Finally, the system executes a global optimization and loop closure to eliminate the accumulated error. In addition, this method proposes using an environment measurement model (EMM) to validate the transformations obtained between the frames. The algorithm is based on SIFT features, which degrades its real-time performance.

RGBDSLAMv2 presents a high computation consumption and requires a slow movement by the sensor for its correct operation [59:1]. Figure 23 represents the algorithm.



**Figure 23.** Diagram representing the RGBDSLAMv2 algorithm. Adapted from [96].

### 3.3.5. General Comments

Section 3.3 individually presented the most representative RGB-D-based techniques. Table 3 summarizes the main characteristics and analyzed criteria for the presented algorithms.

**Table 3.** Main aspects related to the RGB-D-based SLAM approaches.

Method	Tracking Method	Map Density	Loop Closure	Embed. Implement. *	Availability
KinectFusion	Direct	Dense	No	[92,93]	[97]
SLAM++	Hybrid	Dense	Yes	-	-
RGBDSLAMv2	Feature-based	Dense	Yes	-	[98]
DVO	Direct	Dense	Yes	-	[99]
ORB-SLAM 2.0	Feature-based	Dense	Yes	-	[56]

\* Embedded Implementation.

RGB-D-based SLAM algorithms represent an alternative solution to the visual-only and visual-inertial SLAM. In general, they construct dense maps, enabling them to represent the environment in greater detail. In addition, it is a more robust approach regarding low- texture environments thanks to the depth sensor. Concerning embedded implementations, it is possible to find, in the literature, several solutions searching to accelerate parts of the RGB-D-based algorithms that usually require more computation load, such as the ICP algorithm. Beshaw et al. [65] and Williams et al. [66] propose different architectures to accelerate the ICP algorithm, and Gautier et al. [67] implemented the ICP and the volumetric integration algorithms in a heterogeneous architecture. Recent publications have focused on developing robust RGB-D SLAM algorithms considering dynamic environments conditions [103–105].

## 四、主要问题与未来方向

尽管SLAM领域已被广泛研究多年，但仍有几个开放的问题。目前，SLAM和测距算法的技术水平越来越多地寻求加强算法的鲁棒性，优化计算资源的使用，并在地图表示中发展环境的理解<sup>[2:2]</sup>。关于鲁棒性，SLAM和测距技术仍然存在一些破坏算法鲁棒性的主要问题<sup>[2:3]</sup>。其中之一是跟踪失败<sup>[68]</sup>；面对一些挑战或长期场景，算法可能仍然无法识别和关联当前接收图像中的特征，导致姿势估计不准确。这可能会影响到循环闭合技术<sup>[69]</sup>和重新定位<sup>[8,108]</sup>。作为这个问题的解决方案，作者一直在探索新的方法来处理SLAM问题。最近的工作提出结合深度学习和光谱技术<sup>[109,110]</sup>来提高系统的鲁棒性；第4.1节讨论了一些基于深度学习的算法的主要例子。

另一个降低SLAM算法鲁棒性的主要问题是假设静态场景，而现实世界呈现出动态环境；这可能导致跟踪<sup>[70]</sup>和重建<sup>[71]</sup>失败。处理动态场景可能被认为是一个挑战，因为它要求算法检测动态物体，避免跟踪该物体，并将其从地图中排除<sup>[72]</sup>。正如第3.3.5节中提到的，已经发表了一些作品，提出了对这一核心问题的解决方案；第4.3节将讨论更多有代表性的例子。

除了鲁棒性，最近的SLAM算法还试图考虑计算资源的使用<sup>[2:4]</sup>。目前这个话题导致了地图存储的内存使用的开放问题<sup>[2:5]</sup>。在长期操作中存储地图可能会大大增加内存的使用量，这可能会对内存有限的系统操作产生影响，例如。g.嵌入SLAM。然而，在文献中已经可以找到为这一主题提出解决方案的作品。一个例子是Opdenbosch等人的工作。<sup>[73]</sup>提出了一种高效的地图压缩方法，并证明了其能够在不丢失相关信息的情况下大幅减少地图的数据和大小。除了地图存储之外，影响资源使用的另一个主要问题是地图的稀疏性。密集和半密集地图提供了更详细的环境表示，但这一特点对资源的使用有一定的影响。已经证明，与半密集和密集的地图相比，稀疏地图的功耗更低-Wan等人。<sup>[74]</sup>因此，尽管它们提供的细节较少，但它们可能更适合于嵌入式实现。

目前，SLAM算法还试图在执行的重建中发展我们对环境的理解<sup>[2:6]</sup>。除了获得几何信息外，这些算法还通过识别环境中的物体来获得有关环境的信息，比如说。基于语义的SLAM是一个不断发展的SLAM类别，它能够实现这种更好的环境抽象化。语义SLAM是SLAM的一个趋势性话题，第四节中讨论了一些主要的例子。2.接下来，我们将简要讨论一些我们认为具有代表性的视觉-SLAM和视觉-odometry领域的未来方向的相关文章。

## 4.1. 基于深度学习的算法

UnDeepVO<sup>[75]</sup>是一种融合了深度学习概念的杰出算法。这种单眼视觉测量算法可以通过一个深度神经网络进行姿势和深度估计。作者用无监督学习的方式利用立体图像训练UnDeepVO；此外，他们在训练的损失函数中同时考虑了空间和时间的密集信息。事实证明，这种方法比其他单眼方法，如ORB-SLAM（无环形闭合）更加准确和稳健。

最近，同一研究小组提出了DeepSLAM<sup>[76]</sup>。该系统考虑了一个使用无监督学习训练的跟踪网和映射网，并在损失函数中考虑了空间和时间几何。该算法还包含一个Loop-Net来进行循环检测。DeepSLAM呈现出比其他单眼算法，如ORB-SLAM更好的性能，以及比ORB-SLAM和LSD-SLAM更好的稳健性。

另一个基于深度学习的相关算法是DF-SLAM<sup>[77]</sup>。DF-SLAM遵循一个与ORB-SLAM类似的框架，但不使用手工制作的特征，在第2节中解释。1.1，它使用TFeat网络描述的深层局部特征。作者提供了几个比较DF-SLAM和ORB-SLAM2的结果；对于大多数序列，提议的算法获得了更好的性能。最近，有可能在文献中找到一些概述[119-121]，涉及基于深度学习的算法应用于深度估计和SLAM的主要概念方向。更多使用深度学习技术的方法将在第4节讨论。3作为动态SLAM算法的一个解决方案。

## 4.2. 基于语义的算法

将语义信息纳入视觉-SLAM问题是一个不断发展的领域，近年来已经引起了更多的关注。这一领域的一项重要的最新研究在<sup>[78]</sup>中提出。作者提出了一种新的数据关联方法，该方法结合了对象检测器的信息，提出了一种解决方案，可以在因子图解决方案中同时表示数据关联和地标类。与其他包含语义数据关联技术的解决方案相比，该方法呈现出较少的错误。更多包含语义数据的方法将在第4节讨论。3作为动态SLAM算法的一个解决方案。随着这一领域的发展，也有必要建立验证基于语义的算法的方法。作者在<sup>[79]</sup>中介绍了一个新的综合生成的基准数据集，除了传统的轨迹的地面真相外，还包含语义标签、关于场景组成的信息、地面真相三维模型和物体的姿势。此外，他们还提出了可以评估基于语义的算法性能的评价指标。

## 4.3. 动态SLAM算法

对考虑动态环境的SLAM算法的研究对于提高算法在更多现实情况下的鲁棒性至关重要。首先，在<sup>[80]</sup>中，然后在<sup>[81]</sup>中，Sun等人提出了一种运动消除技术来处理RGB-D方法中的环境动态。提出了一种运动消除技术，以处理RGB-D方法中环境的动态性问题。在<sup>[81:1]</sup>中，移除算法可分为两部分；首先，它识别移动物体，并利用图像中物体造成的误差更新前景模型。然后，它进行了前景分割。该算法获得了更好的性能，特别是在高动态环境下，比一些最先进的技术，如DVO，有更好的表现。

Xiao等人提出的动态SLAM是一种对动态场景具有鲁棒性的基本算法。<sup>[82]</sup>；这种方法同时包含了深度学习和语义技术。该系统采用CNN在语义层面上检测动态物体；它将动态和静态特征分开，将动态物体视为异常值。此外，他们还提出了一种补偿算法，以提高检测精度和基于特征的框架。追踪线程结合了语义数据，抛弃或保留了特征。与其他方法如LSD-SLAM、SVO和PTAM相比，动态SLAM呈现出更高的准确性；与ORB-SLAM2相比，动态SLAM的稳健性更好。

DynaSLAM II<sup>[83]</sup>是另一种相关方法，它结合了语义分割来追踪动态物体。该算法基于ORB-SLAM2，在每一个新的帧中进行语义分割和特征提取。该算法不对动态物体进行假设，并进行动态和静态特征的数据关联。静态特征被用来估计摄像机的初始位置，然后对轨迹、边界框和三维点进行优化。

DynaSLAMII显示出与其他最先进的算法，如ORB-SLAM2相当的性能。

# 五、数据集和标定

在文献中的所有SLAM算法中，必须对它们进行公平的比较，以确定哪一种算法在某些情况下有更好的表现。文献中提出了几个具有不同特征的基准数据集，以探索SLAM的能力和稳健性。在这里，我们提出了公开可用的基准数据集，用于评估所提出的SLAM算法的原始文章。

TUM的RGB-D数据集<sup>[84]</sup>由几个图像序列组成，包含在室内环境中用微软Kinetic在两个不同的平台上记录的彩色和深度图像：机器人和手持。该系统与一个运动捕捉系统同步，以提供地面真实。此外，作者提出了两个指标来评估轨迹的局部准确性和全局一致性；它们分别是相对姿势误差和绝对轨迹误差。

KITTI数据集<sup>[85]</sup>包含由彩色和灰度立体相机记录的户外序列。KITTI还展示了来自三维激光扫描仪的数据和由INS/GPS提供的地面实况。该传感器系统是同步的，并安装在一辆汽车上。此外，作者还提供了用于动态物体分类的tracklets和用于评估机器人任务的基准，如视觉测距和SLAM。

另一个主要的基准数据集是ICL-NUIM<sup>[86]</sup>。该数据集专注于RGB-D算法，并通过八个合成的室内场景为评估三维重建提供数据。一个手持式RGB-D相机生成序列，而地面实况包括一个三维表面模型和由SLAM算法估计的轨迹<sup>[87]</sup>。EuRoC基准数据集<sup>[88]</sup>被广泛用于评估纯视觉和视觉-惯性SLAM和测距算法。这些数据是由一个微型飞行器（MAV）在两个室内环境中收集的，它提供了11个立体图像和IMU数据的序列。地面真相是由全站仪和运动捕捉系统获得的。

一个常用于评估单眼系统的数据集是TUM MonoVO<sup>[89]</sup>。它包含了由两个手持式非立体单眼相机提供的几个光度校准的室内和室外序列。由于场景的多样性，作者没有提供姿势的基本事实，但他们执行了在同一位置开始和结束的大序列，允许评估循环漂移。最后，为视觉-惯性系统评估提供的数据集是TUM VI数据集<sup>[90]</sup>。它提供了由一个与IMU同步的立体相机拍摄的几个室内和室外序列。该传感器系统是手持式的，和TUM MonoVO一样，不可能为整个序列建立地面真实。然而，他们通过运动捕捉系统为系统的开始和结束提供地面真实。

表4总结了本工作中提出的主要基准数据集特征。

**Table 4.** Main aspects related to the presented benchmark datasets.

Dataset	Year	Env. *	Platform	Sensor System	Ground-truth	Availability
TUM RGB-D KITTI	2012	Indoor	Robot/Handheld Car	RGB-D camera	Motion capture	[133]
	2013	Outdoor		Stereo-cameras 3D laser scanner	INS/GPS	[134]
ICL-NUIM	2014	Indoor	Handheld	RGB-D camera	3D surface model SLAM estimation	[135]
EuRoC	2016	Indoor	MAV	Stereo-cameras IMU	Total Station Motion capture	[136]
	2016	Indoor/Outdoor		Non-stereo cameras	-	[137]
TUM MonoVO TUM VI	2018	Indoor/Outdoor	Handheld	Stereo-camera IMU	Motion capture (partially)	[138]

\* Environment: indoor or outdoor.

## 六、结论

基于视觉的SLAM技术代表了一个广泛的研究领域，这要归功于其廉价和小型传感器系统所提供的稳健性和准确性。文献中提出了许多不同的视觉SLAM算法，使研究人员在评估其优点和缺点时没有标准，难以选择。

在本文中，我们介绍了主要的基于视觉的SLAM方法，并对每种方法中最典型的一组技术进行了简要描述和系统分析。为了指导在所有算法中的选择，我们提出了六个标准，这些标准是几个SLAM项目的限制因素：算法类型、重建地图的密度、全局优化和回环检测的存在、其可用性和已经进行的嵌入式实现。研究人员可以根据他们的应用考虑每个标准，并从本文中获得初步分析。

此外，我们还提出了一些主要问题，提出了该领域的未来发展方向，并讨论了用于视觉-SLAM和测距算法评估的主要基准数据集。关于未来的工作，我们将把所提出的标准分析应用于核退役的情况。最佳的SLAM算法应在考虑到该环境和应用所具有的各种特征和规格后选择。

**Author Contributions:** Conceptualization, A.M.B., M.M., Y.M., G.C. and F.C.; methodology, A.M.B., M.M. and Y.M.; formal analysis, A.M.B., M.M., Y.M. and F.C.; investigation, A.M.B.; writing—original draft preparation, A.M.B.; writing—review and editing, M.M., Y.M., G.C. and F.C.; supervision, M.M., Y.M., G.C. and F.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding. **Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Smith, R.; Cheeseman, P. On the Representation and Estimation of Spatial Uncertainty. *Int. J. Robot. Res.* 1987, 5, 56–68. [CrossRef:<http://doi.org/10.1177/027836498600500404>] ↩
2. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* 2016, 32, 1309–1332. [CrossRef:<http://dx.doi.org/10.1109/TRO.2016.2624754>] ↩ ↩ ↩ ↩ ↩ ↩ ↩ ↩
3. Stachniss, C. *Robotic Mapping and Exploration*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 55. ↩
4. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* 2017, 9, 1–11. [CrossRef:<http://dx.doi.org/10.1186/s41074-017-0027-2>] ↩ ↩ ↩ ↩ ↩

5. Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An Overview to Visual Odometry and Visual SLAM: Applications to MobileRobotics. *Intell. Ind. Syst.* 2015, 1, 289–311.  
[CrossRef:<http://dx.doi.org/10.1007/s40903-015-0032-7>] ↪ ↪ ↪
6. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J.M. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* 2015, 43, 55–81.  
[CrossRef:<http://dx.doi.org/10.1007/s10462-012-9365-8>] ↪
7. Covolan, J.P.; Sementille, A.; Sanches, S. A mapping of visual SLAM algorithms and their applications in augmented reality. In Proceedings of the 2020 22nd Symposium on Virtual and Augmented Reality (SVR), Porto de Galinhas, Brazil, 7–10 November2020.  
[CrossRef:<http://dx.doi.org/10.1109/SVR51698.2020.00019>] ↪
8. Servières, M.; Renaudin, V.; Dupuis, A.; Antigny, N. Visual and Visual-Inertial SLAM: State of the Art, Classification, andExperimental Benchmarking. *J. Sensors* 2021, 2021, 2054828.  
[CrossRef:<http://dx.doi.org/10.1155/2021/2054828>] ↪
9. Gui, J.; Gu, D.; Wang, S.; Hu, H. A review of visual inertial odometry from filtering and optimisation perspectives. *Adv. Robot.* 2015, 29, 1–13.  
[CrossRef:<http://dx.doi.org/10.1080/01691864.2015.1057616>] ↪ ↪
10. Chen, C.; Zhu, H.; Li, M.; You, S. A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based andOptimization-Based Perspectives. *Robotics* 2018, 7, 45.  
[CrossRef:<http://dx.doi.org/10.3390/robotics7030045>] ↪ ↪ ↪ ↪ ↪ ↪ ↪ ↪
11. Huang, G. Visual-Inertial Navigation: A Concise Review. In Proceedings of the 2019 International Conference on Robotics andAutomation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9572–9582. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2019.8793604>] ↪
12. Chen, K.; Lai, Y.; Hu, S. 3D indoor scene modeling from RGB-D data: A survey. *Comput. Vis. Media* 2015, 1, 267–278. [CrossRef:<http://dx.doi.org/10.1007/s41095-015-0029-x>] ↪
13. Zhang, S.; Zheng, L.; Tao, W. Survey and Evaluation of RGB-D SLAM. *IEEE Access* 2021, 9, 21367–21387. [CrossRef:<http://dx.doi.org/10.1109/ACCESS.2021.3053188>] ↪
14. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.* 2008, 81, 155. [CrossRef:<http://dx.doi.org/10.1007/s11263-008-0152-6>] ↪
15. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. Pattern Anal. Mach.Intell.* 2007, 29, 1052–1067.  
[CrossRef:<http://dx.doi.org/10.1109/TPAMI.2007.1049>]  
[PubMed:<http://www.ncbi.nlm.nih.gov/pubmed/17431302>] ↪ ↪
16. Loo, S.Y.; Amiri, A.; Mashohor, S.; Tang, S.; Zhang, H. CNN-SVO: Improving the Mapping in Semi-Direct Visual OdometryUsing Single-Image Depth Prediction. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA),Montreal, QC, Canada, 20–24 May 2019. ↪ ↪
17. Boikos, K.; Bouganis, C.S. Semi-dense SLAM on an FPGA SoC. In Proceedings of the 2016 26th International Conference on FieldProgrammable Logic and Applications (FPL), Lausanne,

- Switzerland, 29 August–2 September 2016; pp. 1–4.  
[CrossRef:<http://dx.doi.org/10.1109/FPL.2016.7577365>] ↪ ↪
18. Canovas, B.; Rombaut, M.; Nègre, A.; Pellerin, D.; Olympieff, S. Speed and Memory Efficient Dense RGB-D SLAM in DynamicScenes. In Proceedings of the IROS 2020—IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 25–29 October 2020; pp. 4996–5001. [CrossRef:<http://dx.doi.org/10.1109/IROS45743.2020.9341542>] ↪
19. Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Trans. Intell. Veh.* 2017, 2, 194–220.  
[CrossRef:<http://dx.doi.org/10.1109/TIV.2017.2749181>] ↪
20. Vincke, B.; Elouardi, A.; Lambert, A. Design and evaluation of an embedded system based SLAM applications. In Proceedings of the 2010 IEEE/SICE International Symposium on System Integration, Sendai, Japan, 21–22 December 2010; pp. 224–229.  
[CrossRef:<http://dx.doi.org/10.1109/SII.2010.5708329>] ↪
21. Vincke, B.; Elouardi, A.; Lambert, A.; Merigot, A. Efficient implementation of EKF-SLAM on a multi-core embedded system. In Proceedings of the IECON 2012—38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 25–28 October 2012; pp. 3049–3054.  
[CrossRef:<http://dx.doi.org/10.1109/IECON.2012.6389411>] ↪
22. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.  
[CrossRef:<http://dx.doi.org/10.1109/ISMAR.2007.4538852>] ↪
23. Serrata, A.A.J.; Yang, S.; Li, R. An intelligible implementation of FastSLAM2.0 on a low-power embedded architecture. *EURASIP J. Embed. Syst.* 2017, 2017, 27. ↪
24. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327. ↪
25. Ondruška, P.; Kohli, P.; Izadi, S. MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on MobilePhones. *IEEE Trans. Vis. Comput. Graph.* 2015, 21, 1251–1258.  
[CrossRef:<http://dx.doi.org/10.1109/TVCG.2015.2459902>] ↪
26. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.  
[CrossRef:<http://dx.doi.org/10.1109/ICRA.2014.6906584>] ↪ ↪
27. Mur-Artal, R.; Montiel, J.; Tardos, J. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* 2015, 31, 1147–1163.  
[CrossRef:<http://dx.doi.org/10.1109/TRO.2015.2463671>] ↪ ↪ ↪
28. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; M. Montiel, J.M.; D. Tardós, J. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.*

- 2021, 37, 1874–1890. [CrossRef:<http://dx.doi.org/10.1109/TRO.2021.3075644>] ↵ ↵ ↵ ↵ ↵
29. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* 2017, 33, 249–265. [CrossRef:<http://dx.doi.org/10.1109/TRO.2016.2623335>] ↵
30. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Computer Vision–ECCV 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 834–849. ↵
31. Boikos, K.; Bouganis, C.S. A high-performance system-on-chip architecture for direct tracking for SLAM. In Proceedings of the 2017 27th International Conference on Field Programmable Logic and Applications (FPL), Gent, Belgium, 4–6 September 2017; pp. 1–7. [CrossRef:<http://dx.doi.org/10.23919/FPL.2017.8056831>] ↵
32. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* 2017, 33, 1255–1262. [CrossRef:<http://dx.doi.org/10.1109/TRO.2017.2705103>] ↵ ↵
33. Zhan, Z.; Jian, W.; Li, Y.; Yue, Y. A SLAM Map Restoration Algorithm Based on Submaps and an Undirected Connected Graph. *IEEE Access* 2021, 9, 12657–12674. [CrossRef:<http://dx.doi.org/10.1109/ACCESS.2021.3049864>] ↵
34. Abouzahir, M.; Elouardi, A.; Latif, R.; Bouaziz, S.; Tajer, A. Embedding SLAM algorithms: Has it come of age? *Robot. Auton. Syst.* 2018, 100, 14–26. [CrossRef:<http://dx.doi.org/10.1016/j.robot.2017.10.019>] ↵ ↵
35. Yu, J.; Gao, F.; Cao, J.; Yu, C.; Zhang, Z.; Huang, Z.; Wang, Y.; Yang, H. CNN-based Monocular Decentralized SLAM on embeddedFPGA. In Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), New Orleans, LA, USA, 18–22 May 2020; pp. 66–73. [CrossRef:<http://dx.doi.org/10.1109/IPDPSW50202.2020.00019>] ↵
36. Tateno, K.; Tombari, F.; Laina, I.; Navab, N. CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6565–6574. [CrossRef:<http://dx.doi.org/10.1109/CVPR.2017.695>] ↵ ↵
37. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* 2018, 40, 611–625. [CrossRef:<http://dx.doi.org/10.1109/TPAMI.2017.2658577>] [PubMed:<http://www.ncbi.nlm.nih.gov/pubmed/28422651>] ↵ ↵
38. Gao, X.; Wang, R.; Demmel, N.; Cremers, D. LDSO: Direct Sparse Odometry with Loop Closure. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018. ↵
39. Piat, J.; Fillatreau, P.; Tortei, D.; Brenot, F.; Devy, M. HW/SW co-design of a visual SLAM application. *J.-Real-Time Image Process.* 2018. [CrossRef:<http://dx.doi.org/10.1007/s11554-018-0836-2>] ↵

40. DPU for Convolutional Neural Network. Available online: <https://www.xilinx.com/products/intellectual-property/dpu.html#overview> (accessed on 21 January 2022). ↵
41. Xu, Z.; Yu, J.; Yu, C.; Shen, H.; Wang, Y.; Yang, H. CNN-based Feature-point Extraction for Real-time Visual SLAM on EmbeddedFPGA. In Proceedings of the 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom ComputingMachines (FCCM), Fayetteville, AR, USA, 3–6 May 2020; pp. 33–37. ↵
42. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings ofthe 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572. ↵
43. Sun, K.; Mohta, K.; Pfrommer, B.; Watterson, M.; Liu, S.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V. Robust Stereo Visual InertialOdometry for Fast Autonomous Flight. *IEEE Robot. Autom. Lett.* **2018**, *3*, 965–972. [CrossRef:<http://dx.doi.org/10.1109/LRA.2018.2793349>] ↵ ↵
44. Li, S.P.; Zhang, T.; Gao, X.; Wang, D.; Xian, Y. Semi-direct monocular visual and visual-inertial SLAM with loop closure detection. *Robot. Auton. Syst.* **2019**, *112*, 201–210. [CrossRef:<http://dx.doi.org/10.1016/j.robot.2018.11.009>] ↵
45. Delmerico, J.; Scaramuzza, D. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May2018; pp. 2502–2509. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2018.8460664>] ↵ ↵ ↵ ↵
46. Li, M.; Mourikis, A.I. Improving the accuracy of EKF-based visual-inertial odometry. In Proceedings of the 2012 IEEE InternationalConference on Robotics and Automation, Saint Paul, MI, USA, 14–18 May 2012; pp. 828–835. ↵
47. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-Based Visual-Inertial Odometry Using NonlinearOptimization. *Int. J. Robot. Res.* **2014**, *34*, 314–334. [CrossRef:<http://dx.doi.org/10.1177/0278364914554813>] ↵
48. Nikolic, J.; Rehder, J.; Burri, M.; Gohl, P.; Leutenegger, S.; Furgale, P.T.; Siegwart, R. A synchronized visual-inertial sensor systemwith FPGA pre-processing for accurate real-time SLAM. In Proceedings of the 2014 IEEE International Conference on Roboticsand Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 431–437. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2014.6906892>] ↵
49. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust visual inertial odometry using a direct EKF-based approach. InProceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany,28 September–3 October 2015; pp. 298–304. [CrossRef:<http://dx.doi.org/10.1109/IROS.2015.7353389>] ↵
50. Von Stumberg, L.; Usenko, V.; Cremers, D. Direct Sparse Visual-Inertial Odometry Using Dynamic Marginalization. InProceedings of the 2018 IEEE International Conference on Robotics and

- Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2510–2517.  
[CrossRef:<http://dx.doi.org/10.1109/ICRA.2018.8462905>] ↪ ↪ ↪
51. Mur-Artal, R.; Tardós, J.D. Visual-Inertial Monocular SLAM With Map Reuse. *IEEE Robot. Autom. Lett.* 2017, 2, 796–803. [CrossRef:<http://dx.doi.org/10.1109/LRA.2017.2653359>] ↪ ↪
52. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* 2018, 34, 1004–1020.  
[CrossRef:<http://dx.doi.org/10.1109/TRO.2018.2853729>] ↪ ↪
53. Paul, M.K.; Wu, K.; Hesch, J.A.; Nerurkar, E.D.; Roumeliotis, S.I. A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 165–172. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2017.7989022>] ↪
54. Campos, C.; Montiel, J.M.; Tardós, J.D. Inertial-Only Optimization for Visual-Inertial Initialization. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 51–57.  
[CrossRef:<http://dx.doi.org/10.1109/ICRA40945.2020.9197334>] ↪
55. Seiskari, O.; Rantalankila, P.; Kannala, J.; Ylilammi, J.; Rahtu, E.; Solin, A. HybVIO: Pushing the Limits of Real-Time Visual-Inertial Odometry. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2022; pp. 701–710.  
↪
56. Merzlyakov, A.; Macenski, S. A Comparison of Modern General-Purpose Visual SLAM Approaches. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 9190–9197.  
[CrossRef:<http://dx.doi.org/10.1109/IROS51168.2021.9636615>] ↪
57. VI-Stereo-DSO. Available online: <https://github.com/RonaldSun/VI-Stereo-DSO> (accessed on 21 January 2022). ↪
58. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136. ↪ ↪
59. Jin, Q.; Liu, Y.; Man, Y.; Li, F. Visual SLAM with RGB-D Cameras. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 4072–4077.  
[CrossRef:<http://dx.doi.org/10.23919/ChiCC.2019.8865270>] ↪ ↪
60. Nardi, L.; Bodin, B.; Zia, M.Z.; Mawer, J.; Nisbet, A.; Kelly, P.H.J.; Davison, A.J.; Luján, M.; O’Boyle, M.F.P.; Riley, G.D.; et al. Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5783–5790. ↪ ↪
61. Bodin, B.; Nardi, L.; Zia, M.Z.; Wagstaff, H.; Shenoy, G.S.; Emani, M.; Mawer, J.; Kotselidis, C.; Nisbet, A.; Lujan, M.; et al. Integrating algorithmic parameters into benchmarking and design

- space exploration in 3D scene understanding. In Proceedings of the 2016 International Conference on Parallel Architecture and Compilation Techniques (PACT), Haifa, Israel, 11–15 September 2016; pp. 57–69. [CrossRef:<http://dx.doi.org/10.1145/2967938.2967963>] ↩
62. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.; Davison, A.J. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1352–1359. [CrossRef:<http://dx.doi.org/10.1109/CVPR.2013.178>] ↩
63. Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2100–2106. ↩
64. Endres, F.; Hess, J.; Sturm, J.; Cremers, D.; Burgard, W. 3-D Mapping With an RGB-D Camera. *IEEE Trans. Robot.* 2014, 30, 177–187. [CrossRef:<http://dx.doi.org/10.1109/TRO.2013.2279412>] ↩
65. Belshaw, M.S.; Greenspan, M.A. A high speed iterative closest point tracker on an FPGA platform. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1449–1456. [CrossRef:<http://dx.doi.org/10.1109/ICCVW.2009.5457442>] ↩
66. Williams, B. Evaluation of a SoC for Real-time 3D SLAM. Doctoral Dissertation, Iowa State University, Ames, IA, USA, 2017. ↩
67. Gautier, Q.; Shearer, A.; Matai, J.; Richmond, D.; Meng, P.; Kastner, R. Real-time 3D reconstruction for FPGAs: A case study for evaluating the performance, area, and programmability trade-offs of the Altera OpenCL SDK. In Proceedings of the 2014 International Conference on Field-Programmable Technology (FPT), Shanghai, China, 10–12 December 2014; pp. 326–329. [CrossRef:<http://dx.doi.org/10.1109/FPT.2014.7082810>] ↩
68. Deng, X.; Zhang, Z.; Sintov, A.; Huang, J.; Bretl, T. Feature-constrained Active Visual SLAM for Mobile Robot Navigation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7233–7238. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2018.8460721>] ↩
69. Jaenal, A.; Zuñiga-Nöel, D.; Gomez-Ojeda, R.; Gonzalez-Jimenez, J. Improving Visual SLAM in Car-Navigated Urban Environments with Appearance Maps. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 4679–4685. [CrossRef:<http://dx.doi.org/10.1109/IROS45743.2020.9341451>] ↩
70. Scona, R.; Jaimez, M.; Petillot, Y.R.; Fallon, M.; Cremers, D. StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3849–3856. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2018.8460681>] ↩

71. Soares, J.C.V.; Gattass, M.; Meggiolaro, M.A. Visual SLAM in Human Populated Environments: Exploring the Trade-off between Accuracy and Speed of YOLO and Mask R-CNN. In Proceedings of the 2019 19th International Conference on Advanced Robotics(ICAR), Horizonte, Brazil, 2–6 December 2019; pp. 135–140. [CrossRef:<http://dx.doi.org/10.1109/ICAR46387.2019.8981617>] ↪
72. Soares, J.C.V.; Gattass, M.; Meggiolaro, M.A. Crowd-SLAM: Visual SLAM Towards Crowded Environments using ObjectDetection. *J. Intell. Robot. Syst.* 2021, 102, 50. [CrossRef:<http://dx.doi.org/10.1007/s10846-021-01414-1>] ↪
73. Van Opdenbosch, D.; Aykut, T.; Alt, N.; Steinbach, E. Efficient Map Compression for Collaborative Visual SLAM. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018;pp. 992–1000. [CrossRef:<http://dx.doi.org/10.1109/WACV.2018.00114>] ↪
74. Wan, Z.; Yu, B.; Li, T.; Tang, J.; Wang, Y.; Raychowdhury, A.; Liu, S. A Survey of FPGA-Based Robotic Computing. *IEEE CircuitsSyst. Mag.* 2021, 21, 48–74. [CrossRef:<http://dx.doi.org/10.1109/MCAS.2021.3071609>] ↪
75. Li, R.; Wang, S.; Long, Z.; Gu, D. UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018;pp. 7286–7291. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2018.8461251>] ↪
76. Li, R.; Wang, S.; Gu, D. DeepSLAM: A Robust Monocular SLAM System With Unsupervised Deep Learning. *IEEE Trans. Ind.Electron.* 2021, 68, 3577–3587. [CrossRef:<http://dx.doi.org/10.1109/TIE.2020.2982096>] ↪
77. Kang, R.; Shi, J.; Li, X.; Liu, Y.; Liu, X. DF-SLAM: A Deep-Learning Enhanced Visual SLAM System based on Deep Local Features. *arXiv* 2019, arXiv:1901.07223 ↪
78. Doherty, K.; Fourie, D.; Leonard, J. Multimodal Semantic SLAM with Probabilistic Data Association. In Proceedings of the 2019International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2419–2425. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2019.8794244>] ↪
79. Cao, Y.; Hu, L.; Kneip, L. Representations and Benchmarking of Modern Visual SLAM Systems. *Sensors* 2020, 20, 2572. [CrossRef:<http://dx.doi.org/10.3390/s20092572>] [PubMed:<http://www.ncbi.nlm.nih.gov/pubmed/32366018>] ↪
80. Sun, Y.; Liu, M.; Meng, M.Q.H. Improving RGB-D SLAM in dynamic environments: A motion removal approach. *Robot. Auton.Syst.* 2017, 89, 110–122. [CrossRef:<http://dx.doi.org/10.1016/j.robot.2016.11.012>] ↪
81. Sun, Y.; Liu, M.; Meng, M.Q.H. Motion removal for reliable RGB-D SLAM in dynamic environments. *Robot. Auton. Syst.* 2018, 108, 115–128. [CrossRef:<http://dx.doi.org/10.1016/j.robot.2018.07.002>] ↪ ↪
82. Xiao, L.; Wang, J.; Qiu, X.; Rong, Z.; Zou, X. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robot. Auton. Syst.*

- 2019, 117, 1–16. [CrossRef:<http://dx.doi.org/10.1016/j.robot.2019.03.012>] ↵
83. Bescos, B.; Campos, C.; Tardós, J.D.; Neira, J. DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM. *IEEE Robot. Autom. Lett.* 2021, 6, 5191–5198. [CrossRef:<http://dx.doi.org/10.1109/LRA.2021.3068640>] ↵
84. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 573–580. [CrossRef:<http://dx.doi.org/10.1109/IROS.2012.6385773>] ↵
85. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* 2013, 32, 1231–1237. [CrossRef:<http://dx.doi.org/10.1177/0278364913491297>] ↵
86. Handa, A.; Whelan, T.; McDonald, J.; Davison, A.J. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1524–1531. [CrossRef:<http://dx.doi.org/10.1109/ICRA.2014.6907054>] ↵
87. Whelan, T.; Kaess, M.; Johannsson, H.; Fallon, M.; Leonard, J.J.; McDonald, J. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *Int. J. Robot. Res.* 2015, 34, 598–626. [CrossRef:<http://dx.doi.org/10.1177/0278364914551008>] ↵
88. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* 2016, 35, 1157–1163. [CrossRef:<http://dx.doi.org/10.1177/0278364915620033>] ↵
89. Engel, J.; Usenko, V.; Cremers, D. A Photometrically Calibrated Benchmark For Monocular Visual Odometry. *arXiv* 2016, arXiv:1607.02555. ↵
90. Schubert, D.; Goll, T.; Demmel, N.; Usenko, V.; Stückler, J.; Cremers, D. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1680–1687. [CrossRef:<http://dx.doi.org/10.1109/IROS.2018.8593419>] ↵