PART TWO
SLAM IN PRACTICE

II

Prelude

Ayoung Kim, Timothy Barfoot, Luca Carlone,
Frank Dellaert, and Daniel Cremers

Part I laid the foundations by introducing the basic language of factor graphs and their role as a SLAM back-end. Building on this groundwork, Part II focuses on the characteristics and integration of various sensor modalities used in SLAM, which directly impacts the SLAM front-end. Together, these two parts provide a cohesive understanding of the SLAM pipeline: the back-end, discussed in Part I, addresses the underlying optimization problem, while the front-end, covered in this part, tackles sensor-specific tasks such as preprocessing, time synchronization, noise filtering, and measurement modeling.

This part explores the widely adopted sensors in SLAM, organizing the chapters based on the common categorization of SLAM algorithms by their primary sensor type. The sensors discussed in Part II include RGB cameras and LiDARs, along with emerging modalities such as event cameras and radars. IMUs are emphasized for their critical role in SLAM, both as standalone sensors and in combination with other modalities. The discussion also explores how robot kinematics can be integrated as measurements in the SLAM system.

II.1 Structure of SLAM Framework

Most SLAM front-ends rely on two key modules: odometry and loop closure. While these modules are essential, priors can be occasionally incorporated alongside them. The *odometry* module imposes measurement constraints between consecutive nodes in a graph, typically arranged in sequential time order. By chaining together these odometry estimations, a trajectory can be inferred, though it will inevitably accumulate drift over time. The accumulated drift highlights the importance of the *loop-closure* module as a key component in the SLAM system, as it mitigates this drift by recognizing previously visited scenes and establishing connections to historical nodes. Incorporating loop-closure detection and correction is the essence of SLAM. Beyond these two key modules, a unary factor may also be included as a prior, depending on the sensor type. These unary factors are often used to incorporate additional sensor data or priors, enhancing the overall SLAM system's

accuracy and robustness (*e.g.,* GPS and depth). Although not the central element, these factors serve as a valuable supporting component.

II.1.1 Odometry

The odometry module estimates the relative transformation between two consecutive nodes in a graph. This transformation can vary in complexity depending on the scenario. The most common case involves two nodes corresponding to consecutive sensor frames, where a relative 6-DOF binary factor is inferred between them. In this context, the comparison of sensor measurements is performed by computing pixel-to-pixel, point-to-point, or feature-to-feature loss. Many visual and LiDAR SLAM systems compute odometry in this manner.

At times, the odometry computation involves more detailed procedures. When kinematic or dynamic information is available, such as from radar or leg encoders and contact sensors, the odometry computation becomes less relative. For instance, in radar, both range and radial velocity are available. By using the radial velocity to infer ego-velocity, a 6-DOF factor can be integrated between two frames. Leg odometry can also be enhanced by incorporating data from encoders or contact sensors, along with robot kinematics, during the odometry computation. The most widely used odometry method is inertial odometry. Inertial measurements involve more complex computations between two nodes, often implemented as a pre-integration factor. This pre-integration can serve as odometry but is frequently combined with other sensors to form a more comprehensive pre-integration factor.

Its naming varies based on the primary sensor used, such as visual odometry, LiDAR odometry, or leg odometry. In this part, each chapter provides a detailed exploration of the odometry module tailored to specific sensor modalities.

II.1.2 Loop-closure

The loop-closure module entails two problems. The first problem is *place recognition*, which involves using information-retrieval methods to identify a candidate loop closure in a topological manner. In this book, we refer to place recognition as the loop-closure candidate-detection problem. It is the retrieval task that involves identifying a query's nearest index from the database using retrieval or matching algorithms. We will also use the terms *query* and *database*, which are commonly used in the information-retrieval literature, but also commonly used in place recognition. The query is the current sensor measurement and the database includes a collection of places in the form of raw sensor measurements or descriptors.

Loop-closure detection is commonly performed using a variety of extrovert sensors, such as RGB cameras, event cameras, LiDARs, and radars. Early work focused on creating highly descriptive and compact place (or scene) descriptors. For

instance, visual place recognition applied information retrieval techniques, introducing visual words to implement a bag-of-words model. Primarily, binary bag-of-words (DBoW) has been widely adopted in many visual SLAM applications. Similarly, for range sensors, compact descriptors for range measurements, such as those used in Scan Context, have been developed. These hand-crafted descriptors are now transitioning to learning-based approaches, including cross-modal place recognition, which enables place recognition from different sensor modalities.

When detecting a loop-closure, both discernibility and scalability are crucial. Place recognition must accurately distinguish between similar-looking but distinct locations, avoiding perceptual aliasing. To achieve this, it is essential to develop effective descriptors or train networks that ensure strong discernibility. In addition, during long-term SLAM operations over large areas, both the map and the query database will grow. Consequently, loop-closure detection must be performed efficiently, even as the map expands.

Once a candidate is found, *re-localization* or *relative pose estimation* aims to estimate a fine registration between the candidate match and the current data, which is needed for the SLAM back-end optimization. Once the proposal loop-closure candidate is secured, the follow-up registration encompasses estimating a relative transformation (either full 6-DOF or partial) between the query node and the candidate node, resulting in a binary factor.

II.1.3 Priors and Unary Factors

Some sensors provide measurements as priors rather than relative measurements between frames. This is the case for sensors like GPS measurements in terrain navigation, depth sensors in underwater environments, or radar providing instantaneous velocity. These priors can be incorporated into the SLAM system as unary factors, constraining a node with details and enhancing the accuracy of the overall system. Unfortunately, their modeling will not be discussed in detail in the following chapters. For further insights, readers are referred to dedicated resources on GNSS [151] or UWB applications.

II.2 Sensors in a Factor Graph

From the factor-graph SLAM perspective, each sensor modality produces a factor, acting as a building block for the entire SLAM system. An example factor graph in Figure II.1 illustrates a possible integration case. The factor-graph framework, introduced in Part I, serves as a generic optimization back-end, where each sensor can contribute to the graph either independently or through integration.

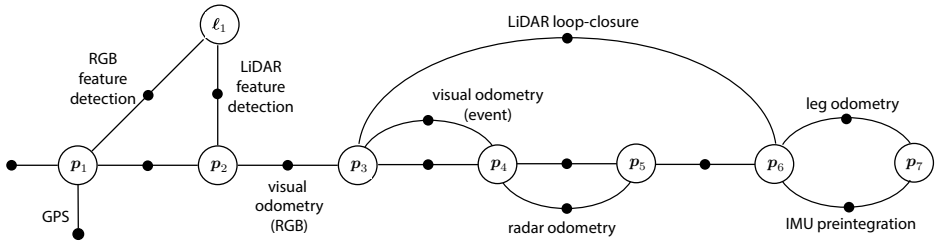

Figure II.1 Sample factor-graph consists of all sensor modalities introduced in this part.

II.2.1 Selecting the Right Sensor for Your Application

Let's briefly overview the extrovert perceptual sensors commonly used in SLAM systems, highlighting their advantages and limitations.

RGB Camera: RGB cameras are among the most widely adopted sensors in SLAM applications due to their affordability, ability to capture semantic information, and their close resemblance to the human visual system. These cameras capture rich visual context, including semantic information, which serves as a powerful cue for various visual SLAM tasks.

However, they do have notable limitations. One significant drawback is their reliance on adequate lighting conditions; without sufficient illumination, the quality of captured images degrades, which can hinder performance in low-light environments. Additionally, RGB cameras are sensitive to glare, reflections, and shadows, and their performance is significantly limited in extreme environments, such as those with fog or smoke.

LiDAR: LiDAR directly measures distance without the need for triangulation, enabling it to accurately capture 2D/3D geometry from range measurements. Indeed, range sensors, such as 2D LiDAR and sonar, are foundational sensing modalities in the history of SLAM. The point cloud data generated by LiDAR has been widely adopted for dense 2D/3D mapping in both indoor and outdoor environments, making it especially valuable for large-scale applications in construction sites, urban areas, and forests.

However, LiDAR's main drawback is its high cost, which can be a burden in terms of both price, memory, power consumption, and computation. The point cloud data it generates is often extensive, requiring substantial computational power and memory for processing and storage. These challenges should be carefully considered when using LiDAR in budget-sensitive, real-time systems (*e.g.,* drones). Furthermore, despite their robustness to the illumination conditions, their performance can be impaired in adverse weather conditions, such as heavy rain, fog, or snow.

Radar: Radar is less affected by environmental conditions, making it a reliable sensor for deployment in extreme environments. It can provide velocity information

through Doppler measurements, adding an additional layer of functionality. While the localization and mapping accuracy of radar is often lower than that of LiDAR, it remains one of the few robust sensors capable of completing SLAM in extreme environments where LiDAR and cameras may be limited. However, radar data tends to be noisy and extremely sparse, which can complicate its interpretation. The signal processing required for radar data is also computationally intensive, and the sensor measurements are not as intuitive as those from optical sensors, making them more challenging to interpret and integrate into systems.

Event camera: Event cameras are unique sensors that offer extremely high temporal resolution (in the order of microseconds) while consuming very low power. By transmitting data only for pixels that change, they are highly energy-efficient, capturing fast-moving objects without motion blur. Their asynchronous nature also leads to efficient data processing, generating smaller data streams compared to conventional frame-based cameras that operate at the same sampling rate. These features make event cameras ideal for real-time, high-speed applications. Additionally, their ability to handle both very bright and dark conditions with high dynamic range (HDR) makes them more effective than frame-based cameras in challenging lighting environments. Despite their advantages, their main limitations are their noise, and lack of fine texture details and absolute intensity output.

II.2.2 Sensor Fusion

In SLAM literature, the common practice is to combine multiple sensors in a complementary manner. The most favored sensor in this integration would be inertial measurements. Incorporating an Inertial Measurement Unit (IMU) with other sensors significantly enhances the performance of many SLAM systems. IMUs provide valuable information about motion, velocity, and orientation, which improves the accuracy and robustness of systems like VINS, LIO, and RIO. These systems leverage the pre-integration of IMU data to maintain continuous tracking and correct drift over time. While IMU-only systems have seen some development, such as efforts in IMU-based odometry, they remain limited in terms of accuracy and reliability. When combined with other sensors like LiDAR or cameras, however, IMUs strengthen the overall SLAM system, compensating for the weaknesses of individual sensors and enabling more precise and reliable mapping and localization, especially in dynamic or challenging environments.

Incorporating the kinematics of the robot platform is also crucial. Many SLAM systems in robotics are designed to work with specific platforms, such as UGVs, drones, or legged robots. By formulating the kinematics of these platforms and integrating them with other sensor modalities, the performance and robustness of the system can be significantly enhanced.

We also need to consider the heterogeneous aspect during sensor fusion. Camera and LiDAR combination would be one of the popular SLAM system. This is

because semantic information from camera and direct range measurement from LiDAR can effectively enhance each sensor's limitation. Camera-radar fusion follows a similar strategy, complementing each other to mutually enhance performance and address limitations. The velocity measurements from radar can be used to detect dynamic objects instantaneously, while the semantic information from the RGB camera adds valuable context. Similarly, the combination of LiDAR and radar is beneficial, as radar can improve odometry and facilitate dynamic object removal, while the dense point clouds from LiDAR contribute to higher-quality mapping and submap matching.

Lastly, there are often heterogeneous characteristics even among sensors of the same type. RGB cameras, for instance, can vary significantly depending on factors such as lens type and shutter mechanism. When combining among LiDARs, the beam pattern, point cloud density, and field of view (FOV) can vary significantly between different LiDAR models, and this discrepancy must be addressed when integrating LiDAR with other sensors. This is also true for radar, where the two main types—spinning radar and SoC radar—differ significantly in terms of data type, measurement techniques, and their applications.

II.2.3 Calibration and Synchronization of Sensors

Sensor calibration can be categorized into two types: intrinsic and extrinsic. Intrinsic calibration focuses on determining the model parameters specific to a sensor, such as the focal length and distortion coefficients for a camera or the intensity calibration for LiDAR. In the case of a camera using a pinhole camera model, intrinsic calibration solves for parameters like focal length and distortion coefficients. Intrinsic calibration begins with a sensor model and solves for the parameters that define that model.

On the other hand, extrinsic calibration involves finding the transformation between multiple sensors, aligning their coordinate systems, and ensuring accurate data fusion. This process includes establishing correspondences and using them in an optimization problem to solve for the relative transformation between two sensors. The core challenge arises when establishing correspondences across different modalities. For example, to match a pixel from an RGB camera to a 3D point from a LiDAR, one needs to solve the multi-modal registration problem. Due to differences in data formats and underlying physics, comparing data from two sensors is often difficult, much like comparing apples to oranges.

A widely adopted solution is to use a target to carefully capture data from both sensors in order to solve the registration problem. However, this calibration is often limited to the target and may suffer from drift over time. To address this, targetless calibration methods have been developed, which leverage the surroundings as calibration features. Alternatively, calibration can be updated adaptively in real-time by treating it as an optimization parameter.

When integrating multiple sensors into a SLAM system, time synchronization must be carefully managed. Each sensor operates at its own sampling rate, meaning data from different sensors arrives at different intervals. In robotics, the movement of the platform further exacerbates this issue, as the motion induces discrepancies between sensor data streams. Proper synchronization ensures that all sensor data is aligned in time, enabling accurate fusion and minimizing errors in mapping and localization. Of course, strict hardware synchronization is not always feasible, and interpolation—potentially using faster sampling sensors like the IMU—may be necessary to bridge the gaps between sensor data streams.

II.3 Evaluation

SLAM evaluation is typically conducted from multiple perspectives, often at the level of individual modules such as odometry, place recognition, and mapping. As expected, real-time performance is crucial for odometry, whereas place recognition and mapping can tolerate slower processing rates. More computationally intensive tasks, such as map maintenance and updates, are often deferred to post-processing.

The most common SLAM evaluation metric is trajectory accuracy, typically measured by comparing the estimated path to a ground truth trajectory. However, generating ground truth requires expensive sensors and often extensive post-processing. While this trajectory-to-trajectory comparison sounds straightforward, it's not always feasible—especially in environments like indoors, where RTK GPS is ineffective. In such cases, a few surveyed points using QR markers or artificial targets can serve as an alternative. Furthermore, SLAM performance can be analyzed from multiple perspectives by focusing on different quantitative metrics [381, 1278, 413] during trajectory evaluation.

For place recognition, standard classification metrics such as the precision-recall curve, AUC, and F1 score are commonly used. In the context of robot re-localization, however, not only the accuracy metrics but also the distribution [576] of candidate matches plays a critical role in assessing performance.

Lastly, map evaluation is often the most challenging aspect of SLAM, not only due to the large spatial scale but also because of the difficulty in obtaining accurate ground truth. Static LiDAR systems, such as terrestrial laser scanners (TLS), are commonly used to create high-fidelity reference maps for comparison. In addition to global accuracy, evaluations should also consider fine-grained structural details and the memory efficiency of the map representation.

II.4 How to Read this Part?

The organization of this part is structured around different sensor modalities. Each chapter follows a similar structure, focusing on odometry, place recognition, and

SLAM details for each sensor. It is recommended to start with the Visual SLAM chapter to grasp the basic definitions of each SLAM module. After that, the chapters can be read in any order.

Following the discussion on RGB cameras SLAM in Chapter 7, two range sensors are introduced: LiDAR SLAM in Chapter 8 and radar SLAM in Chapter 9. Moving beyond conventional sensors, we delve into event camera SLAM in Chapter 10. The IMU, covered in Chapter 11, serves as an inertial odometry sensor but demonstrates greater potential when combined with other modalities through preintegration techniques. Additionally, this part explores how odometry can be modeled for legged robot systems in Chapter 12.