

Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif,
Davide Scaramuzza, José Neira, Ian D. Reid, John J. Leonard

Abstract—**Simultaneous Localization and Mapping (SLAM)** consists in the concurrent construction of a representation of the environment (the *map*), and the estimation of the state of the robot moving within it. The SLAM community has made astonishing progress over the last 30 years, enabling large-scale real-world applications, and witnessing a steady transition of this technology to industry. We survey the current state of SLAM. We start by presenting what is now the *de-facto* standard formulation for SLAM. We then review related work, covering a broad set of topics including robustness and scalability in long-term mapping, metric and semantic representations for mapping, theoretical performance guarantees, active SLAM and exploration, and other new frontiers. The paper serves as a tutorial for the non-expert reader. It is also a position paper: by looking at the published research with a critical eye, we delineate open challenges and new research issues, that still deserve careful scientific investigation. The paper also contains the authors' take on two questions that often animate discussions during robotics conferences: *do robots need SLAM? Is SLAM solved?*

I. INTRODUCTION

SLAM consists in the simultaneous estimation of the state of the robot and the map of the environment. In simple instances, the robot state is described by its pose (position and orientation), although other quantities may be included in the state, such as robot velocity, sensor biases, and calibration parameters. The map, on the other hand, is a representation of aspects of interest (e.g., position of landmarks, obstacles) describing the environment in which the robot operates.

The need to build a map of the environment is twofold. First, the map is often required to support other tasks; for instance, a map can inform path planning or provide an intuitive visualization for a human operator. Second, the map allows limiting the error committed in estimating the state of the robot. In the absence of a map, dead-reckoning would quickly drift over time; on the other hand, given a map, the robot can “reset” its localization error by re-visiting known

C. Cadena is with the Autonomous Systems Lab, ETH Zurich, Switzerland.
E-mail: cesarc@eth.ch

L. Carlone is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, USA. E-mail: lcarlone@mit.edu

H. Carrillo is with the Escuela de Ciencias Exactas e Ingeniería, Universidad Sergio Arboleda, Colombia. E-mail: henry.carrillo@usa.edu.co

Y. Latif and I.D. Reid are with the Australian Center for Robotic Vision, University of Adelaide, Australia. E-mail: yasir.latif@adelaide.edu.au, ian.reid@adelaide.edu.au

J. Neira is with the Dept. Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain. E-mail: jneira@unizar.es

D. Scaramuzza is with the Robotics and Perception Group, University of Zurich, Switzerland. E-mail: sdavide@ifi.uzh.ch

J.J. Leonard is with Marine Robotics Group, the Massachusetts Institute of Technology, USA. E-mail: jleonard@mit.edu

areas (the so called *loop closure*). Therefore, SLAM finds applications in all scenarios in which a prior map is not available and needs to be built.

In some robotics applications a map of the environment is known *a priori*. For instance, a robot operating on a factory floor can be provided with a manually built map of artificial beacons placed in the environment. Another example is the case in which the robot has access to GPS measurements (the GPS satellites can be considered as moving beacons at known locations). The popularity of the SLAM problem is connected with the emergence of indoor applications of mobile robotics. Indoor operation rules out the use of GPS to bound the localization error; furthermore, SLAM provides an appealing alternative to user-built maps, showing that robot operation is possible in absence of an *ad-hoc* localization infrastructure.

A thorough historical review of the first 20 years of the SLAM problem is discussed by Durrant-Whyte and Bailey in [13, 86]. The surveys [13, 86] mainly cover what we call the *classical age* (1986-2004); the classical age saw the introduction of the main probabilistic formulations for SLAM, including approaches based on Extended Kalman Filters, Rao-Blackwellised Particle Filters, and maximum likelihood estimation; moreover, it delineated the first challenges connected to efficiency and robust data association. The subsequent period is what we call the *algorithmic-analysis age* (2004-2015), and is partially covered by Dissanayake *et al.* in [84]. The algorithmic analysis period saw the study of fundamental properties of SLAM, including observability, convergence, and consistency. In this period, the key role of sparsity towards efficient SLAM solvers was also understood, and the main open-source SLAM libraries were developed.

We review the main SLAM surveys to date in Table I, observing that most recent surveys only cover specific aspects or sub-fields of SLAM. The popularity of SLAM in the last 30 years is not surprising if one thinks about the manifold aspects that SLAM involves. At the lower level (called the *front-end* in Section II) SLAM naturally intersects other research fields such as computer vision and signal processing; at the higher level (that we later call the *back-end*), SLAM is an appealing mix of geometry, graph theory, optimization, and probabilistic estimation. Finally, a SLAM expert has to deal with practical aspects ranging from sensor modeling to system integration.

The present paper gives a broad overview of the current state of SLAM, and offers the perspective of part of the community on the open problems and future directions for the SLAM research. The paper summarizes the outcome of the workshop “The Problem of Mobile Sensors: Setting future

TABLE I: Surveying the surveys

Year	Topic	Reference
2006	Probabilistic approaches and data association	Durrant-Whyte and Bailey [13, 86]
2008	Filtering approaches	Aulinas <i>et al.</i> [12]
2008	Visual SLAM	Neira <i>et al.</i> (special issue) [200]
2011	SLAM back-end	Grisetti <i>et al.</i> [117]
2011	Observability, Consistency, Convergence	Dissanayake <i>et al.</i> [84]
2012	Visual Odometry	Scaramuzza and Fraundofer [102, 248]
2016	Multi robot SLAM	Saeedi <i>et al.</i> [244]
2016	Visual Place recognition	Lowry <i>et al.</i> [177]

goals and indicators of progress for SLAM” [35], held during the *Robotics: Science and System* (RSS) conference (Rome, July 2015). While we often mention topological mapping, our main focus is on metric and semantic SLAM, and we refer the reader to the recent survey of Lowry *et al.* [177] for a more comprehensive coverage of vision-based topological SLAM and place recognition. Before delving into the paper, we provide our take on two questions that often animate discussions during robotics conferences.

Do autonomous robots really need SLAM? Answering this question requires understanding what makes SLAM unique. SLAM aims at building a globally consistent representation of the environment, leveraging both ego-motion measurements and loop closures. The keyword here is “loop closure”: if we sacrifice loop closures, SLAM reduces to odometry. In early applications, odometry was obtained by integrating wheel encoders. The pose estimate obtained from wheel odometry quickly drifts, making the estimate unusable after few meters; this was one of the main thrusts behind the development of SLAM: the observation of external landmarks is useful to reduce the trajectory drift and possibly correct it. However, more recent odometry algorithms are based on visual and inertial information, and have very small drift ($< 0.5\%$ of the trajectory length). Hence the question becomes legitimate: do we really need SLAM? Our answer is three-fold.

First of all, we observe that the SLAM research done over the last decade was the one producing the visual-inertial odometry algorithms that currently represent the state of the art; in this sense visual-inertial navigation (VIN) *is* SLAM: VIN can be considered a *reduced* SLAM system, in which the loop closure (or place recognition) module is disabled. More in general, SLAM provided an excuse to study sensor fusion under more challenging setups (i.e., no GPS, low quality sensors) than previously considered in other literatures (e.g., inertial navigation in aerospace engineering).

The second answer regards loop closures. A robot performing odometry and neglecting loop closures interprets the world as an “infinite corridor” (Fig. 1a) in which the robot keeps exploring new areas indefinitely. A loop closure event informs the robot that this “corridor” keeps intersecting itself (Fig. 1b). The advantage of loop closure now becomes clear: by finding

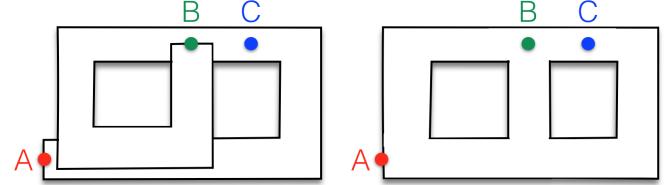


Fig. 1: Left: map built from odometry. The map is homotopic to a long corridor that goes from the starting position A to the final position B. Points that are close in reality (e.g., B and C) may be arbitrarily far in the odometric map. Right: map build from SLAM. By leveraging loop closures, SLAM estimates the actual topology of the environment, and “discovers” shortcuts in the map

loop closures, the robot understands the real topology of the environment, and is able to find shortcuts between locations (e.g., point B and C in the map). Therefore, if getting the right topology of the environment is one of the merits of SLAM, why not simply drop the metric information and just do place recognition? The answer is simple: the metric information makes place recognition much simpler and more robust; the metric reconstruction informs the robot about loop closure opportunities and allows discarding spurious loop closures. Therefore, while SLAM might be redundant in principle (an oracle place recognition module would suffice for topological mapping), SLAM offers a natural defense against wrong data association and perceptual aliasing, where similarly looking scenes, corresponding to distinct locations in the environment, would deceive place recognition. In this sense, the SLAM map provides a way to predict and validate future measurements: we believe that this mechanism is key to robust operation.

The third answer is that SLAM is needed since many applications implicitly or explicitly do require a globally consistent map. For instance, in many military and civil applications the goal of the robot is to explore an environment and report a map to the human operator. Another example is the case in which the robot has to perform structural inspection (of a building, bridge, etc.); also in this case a globally consistent 3D reconstruction is a requirement for successful operation.

One can identify tasks for which different flavors of SLAM formulations are more suitable than others. Therefore, when a roboticist has to design a SLAM system, he/she is faced with multiple design choices. For instance, a topological map can be used to analyze reachability of a given place, but it is not suitable for motion planning and low-level control; a locally-consistent metric map is well-suited for obstacle avoidance and local interactions with the environment, but it may sacrifice accuracy; a globally-consistent metric map allows the robot to perform global path planning, but it may be computationally demanding to compute and maintain. A more general way to choose the most appropriate SLAM system is to think about SLAM as a mechanism to compute a sufficient statistic that summarizes all past observations of the robot, and in this sense, which information to retain in this compressed representation is deeply task-dependent.

Is SLAM solved? This is another question that is often asked within the robotics community, c.f. [106]. The difficulty of answering this question lies in the question itself: SLAM is such a broad topic that the question is well posed only for a given robot/environment/performance combination. In

particular, one can evaluate the maturity of the SLAM problem once the following aspects are specified:

- *robot*: type of motion (e.g., dynamics, maximum speed), available sensors (e.g., resolution, sampling rate), available computational resources;
- *environment*: planar or three-dimensional, presence of natural or artificial landmarks, amount of dynamic elements, amount of symmetry and risk of perceptual aliasing. Note that many of these aspects actually depend on the sensor-environment pair: for instance, two rooms may look identical for a 2D laser scanner (perceptual aliasing), while a camera may discern them from appearance cues;
- *performance requirements*: desired accuracy in the estimation of the state of the robot, accuracy and type of representation of the environment (e.g., landmark-based or dense), success rate (percentage of tests in which the accuracy bounds are met), estimation latency, maximum operation time, maximum size of the mapped area.

For instance, mapping a 2D indoor environment with a robot equipped with wheel encoders and a laser scanner, with sufficient accuracy ($< 10\text{cm}$) and sufficient robustness (say, low failure rate), can be considered largely solved. Similarly, vision-based SLAM with slowly-moving robots (e.g., Mars rovers, domestic robots), and visual-inertial odometry can be considered mature research fields. On the other hand, other robot/environment/performance combinations still deserve a large amount of fundamental research. Current SLAM algorithms can be easily induced to fail when either the motion of the robot or the environment are too challenging (e.g., fast robot dynamics, highly dynamic environments); similarly, SLAM algorithms are often unable to face strict performance requirements, e.g., high rate estimation for fast closed-loop control. This survey will provide a comprehensive overview of these open problems, among others.

We conclude this section with some broader considerations about the future of SLAM. We argue that we are entering in a third era for SLAM, the *robust-perception age*, which is characterized by the following key requirements:

- 1) *robust performance*: the SLAM system operates with low failure rate for an extended period of time in arbitrary environments; the system includes fail-safe mechanisms and has self-tuning capabilities¹ in that it can adapt the selection of the system parameters to the scenario.
- 2) *high-level understanding*: the SLAM system goes beyond basic geometry reconstruction to obtain a high-level understanding of the environment (e.g., semantic, affordances, high-level geometry, physics);
- 3) *resource awareness*: the SLAM system is tailored to the available sensing and computational resources, and provides means to adjust the computation load depending on the available resources;
- 4) *task-driven inference*: the SLAM system produces adaptive map representations, whose complexity can change depending on the task that the robot has to perform.

¹The SLAM community has been largely affected by the “curse of manual tuning”, in that satisfactory operation is enabled by expert tuning of the system parameters (e.g., stopping conditions, thresholds for outlier rejection).

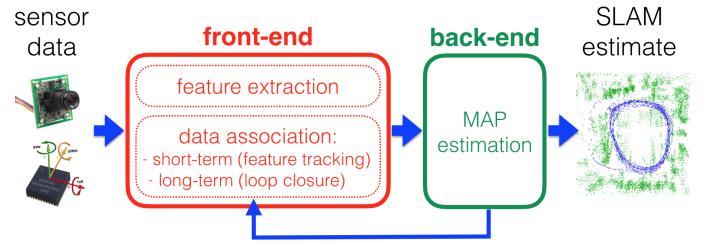


Fig. 2: Front-end and back-end in a typical SLAM system.

Paper organization. The paper starts by presenting a standard formulation and “architecture” for SLAM (Section II). Section III tackles robustness in life-long SLAM. Section IV deals with scalability. Section V discusses how to represent the geometry of the environment. Section VI extends the question of the environment representation to the modeling of semantic information. Section VII provides an overview of the current accomplishments on the theoretical aspects of SLAM. Section VIII broadens the discussion and reviews the active SLAM problem in which decision making is used to improve the quality of the SLAM results. Section IX provides an overview of recent trends in SLAM, including the use of unconventional sensors and new formulations. Section X provides final remarks. Throughout the paper, we provide many pointers to related work outside the robotics community. Despite its unique traits, SLAM is related to problems addressed in computer vision, computer graphics, and control theory, and cross-fertilization among these fields is a necessary condition to enable fast progress.

For the non-expert reader, we recommend to read Durrant-Whyte and Bailey’s SLAM tutorials [13, 86] before delving in this position paper. The more experienced researcher can jump directly to the section of interest, where he/she will find a self-contained overview of the state of the art and open problems.

II. ANATOMY OF A MODERN SLAM SYSTEM

A SLAM system can be conceptually divided into two parts: the *front-end* and the *back-end*. The front-end abstracts sensor data into models that are amenable for estimation, while the back-end performs inference on the abstracted data produced by the front-end. This architecture is summarized in Fig. 2. We review both components, starting from the back-end.

Maximum-a-posteriori (MAP) estimation and the SLAM back-end. The current *de-facto* standard formulation of SLAM has its origins in the seminal paper of Lu and Milios [178], followed by the work of Gutmann and Konolige [120]. Since then, numerous approaches have improved the efficiency and robustness of the optimization underlying the problem [80, 97, 118, 146, 147, 216, 271]. All these approaches formulate SLAM as a maximum-a-posteriori estimation problem, and often use the formalism of *factor graphs* [163] to reason about the interdependence among variables.

Assume that we want to estimate an unknown variable \mathcal{X} ; as mentioned before, in SLAM the variable \mathcal{X} typically includes the trajectory of the robot (as a discrete set of poses) and the position of landmarks in the environment. We are given

a set of measurements $Z = \{z_k : k = 1, \dots, m\}$ such that each measurement can be expressed as a function of \mathcal{X} , i.e., $z_k = h_k(\mathcal{X}_k) + \epsilon_k$, where $\mathcal{X}_k \subseteq \mathcal{X}$ is a subset of the variables, $h_k(\cdot)$ is a known function (the *measurement* or *observation* model), and ϵ_k is random measurement noise.

In MAP estimation, we estimate \mathcal{X} by computing the assignment of variables \mathcal{X}^* that attains the maximum of the posterior $\mathbb{P}(\mathcal{X}|Z)$ (the *belief* over \mathcal{X} given the measurements):

$$\mathcal{X}^* \doteq \operatorname{argmax}_{\mathcal{X}} \mathbb{P}(\mathcal{X}|Z) = \operatorname{argmax}_{\mathcal{X}} \mathbb{P}(Z|\mathcal{X})\mathbb{P}(\mathcal{X}) \quad (1)$$

where the equality follows from the Bayes theorem. In (1), $\mathbb{P}(Z|\mathcal{X})$ is the likelihood of the measurements Z given the assignment \mathcal{X} , and $\mathbb{P}(\mathcal{X})$ is a prior probability over \mathcal{X} . The prior probability includes any prior knowledge about \mathcal{X} ; in case no prior knowledge is available, $\mathbb{P}(\mathcal{X})$ becomes a constant (uniform distribution) which is inconsequential and can be dropped from the optimization. In that case MAP estimation reduces to *maximum likelihood estimation*.

Assuming that the measurements $Z = \{z_k : k = 1, \dots, m\}$ are independent (i.e., the noise terms affecting the measurements are not correlated), problem (1) factorizes into:

$$\mathcal{X}^* = \operatorname{argmax}_{\mathcal{X}} \mathbb{P}(\mathcal{X}) \prod_{k=1}^m \mathbb{P}(z_k|\mathcal{X}) = \mathbb{P}(\mathcal{X}) \prod_{k=1}^m \mathbb{P}(z_k|\mathcal{X}_k) \quad (2)$$

where, on the right-hand-side, we noticed that z_k only depends on the subset of variables in \mathcal{X}_k .

Problem (2) can be interpreted in terms of inference over a factors graph [163]. The variables correspond to nodes in the factor graph. The terms $\mathbb{P}(z_k|\mathcal{X}_k)$ and the prior $\mathbb{P}(\mathcal{X})$ are called *factors*, and they encode probabilistic constraints over a subset of nodes. A factor graph is a graphical model that encodes the dependence between the k -th factor (and its measurement z_k) and the corresponding variables \mathcal{X}_k . A first advantage of the factor graph interpretation is that it enables an insightful visualization of the problem. Fig. 3 shows an example of factor graph underlying a simple SLAM problem (more details in Example 1 below). The figure shows the variables, namely, the robot poses, the landmarks positions, and the camera calibration parameters, and the factors imposing constraints among these variables. A second advantage is generality: a factor graph can model complex inference problems with heterogeneous variables and factors, and arbitrary interconnections; the connectivity of the factor graph in turns influences the sparsity of the resulting SLAM problem as discussed below.

In order to write (2) in a more explicit form, assume that the measurement noise ϵ_k is a zero-mean Gaussian noise with information matrix Ω_k . Then, the measurement likelihood in (2) becomes:

$$\mathbb{P}(z_k|\mathcal{X}_k) \propto \exp\left(-\frac{1}{2}\|h_k(\mathcal{X}_k) - z_k\|_{\Omega_k}^2\right) \quad (3)$$

where we use the notation $\|e\|_{\Omega}^2 = e^T \Omega e$. Similarly, assume that the prior is Normally distributed:

$$\mathbb{P}(\mathcal{X}) \propto \exp\left(-\frac{1}{2}\|h_0(\mathcal{X}) - z_0\|_{\Omega_0}^2\right) \quad (4)$$

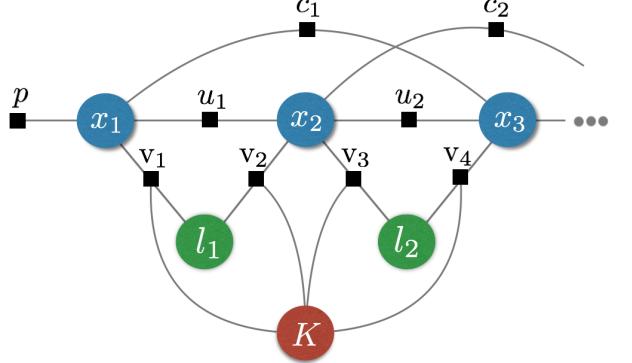


Fig. 3: **SLAM as a factor graph**: Blue circles denote robot poses at consecutive time steps (x_1, x_2, \dots), green circles denote landmark positions (ℓ_1, ℓ_2, \dots), red circle denotes the variable associated with the intrinsic calibration parameters (K). Factors are shown as black dots: the label “u” marks factors corresponding to odometry constraints, “m” marks factors corresponding to camera observations, “c” denotes loop closures, and “p” denotes prior factors.

for some given function $h_0(\cdot)$, prior mean z_0 , and information matrix Ω_0 . Since maximizing the posterior is the same as minimizing the *negative log-posterior*, the MAP estimate in (2) becomes:

$$\begin{aligned} \mathcal{X}^* &= \operatorname{argmin}_{\mathcal{X}} -\log \left(\mathbb{P}(\mathcal{X}) \prod_{k=1}^m \mathbb{P}(z_k|\mathcal{X}_k) \right) = \\ &= \operatorname{argmin}_{\mathcal{X}} \sum_{k=0}^m \frac{1}{2} \|h_k(\mathcal{X}_k) - z_k\|_{\Omega_k}^2 \end{aligned} \quad (5)$$

which is a nonlinear least squares problem, as in most problems of interest in robotics, $h_k(\cdot)$ is a nonlinear function. Note that the formulation (5) follows from the assumption of Normally distributed noise. Other assumptions for the noise distribution lead to different cost functions; for instance, if the noise follows a Laplace distribution, the squared ℓ_2 -norm in (5) is replaced by the ℓ_1 -norm. To increase resilience to outliers, it is also common to substitute the squared ℓ_2 -norm in (5) with robust loss functions (e.g., Huber or Tukey loss) [133].

The computer vision expert may notice a resemblance between problem (5) and *bundle adjustment* (BA) in Structure from Motion [275]; both (5) and BA indeed stem from a maximum-a-posteriori formulation. However, two key features make SLAM unique. First, the factors in (5) are not constrained to model projective geometry as in BA, but include a broad variety of sensor models, e.g., inertial sensors, wheel encoders, GPS, to mention a few. Second, problem (5) is designed to be solved *incrementally*: new measurements are made available at each time step as the robot moves.

The minimization problem (5) is commonly solved via successive linearizations, e.g., the Gauss-Newton (GN) or the Levenberg-Marquardt methods (more modern approaches, based on convex relaxations and Lagrangian duality are reviewer in Section VII). The Gauss-Newton method proceeds iteratively, starting from a given initial guess $\hat{\mathcal{X}}$. At each iteration, GN approximates the minimization (5) as

$$\delta_{\mathcal{X}}^* = \operatorname{argmin}_{\delta_{\mathcal{X}}} \frac{1}{2} \sum_{k=0}^m \|A_k \delta_{\mathcal{X}} - b_k\|_{\Omega_k}^2 = \operatorname{argmin}_{\delta_{\mathcal{X}}} \frac{1}{2} \|\mathbf{A} \delta_{\mathcal{X}} - \mathbf{b}\|_{\Omega}^2 \quad (6)$$

where $\delta_{\mathcal{X}}$ is a small “correction” w.r.t. the linearization point $\hat{\mathcal{X}}$, $A_k \doteq \frac{\partial h_k(\mathcal{X})}{\partial \mathcal{X}}$ is the Jacobian of the measurement function $h_k(\cdot)$ with respect to \mathcal{X} , $b_k \doteq z_k - h(\hat{\mathcal{X}})$ is the *residual error* at $\hat{\mathcal{X}}$; on the right-hand-side of (6), A (resp. b) is obtained by stacking A_k (resp. b_k); Ω_k is a block diagonal matrix including the measurement information matrices Ω_k as diagonal blocks.

The optimal correction $\delta_{\mathcal{X}}^*$ which minimizes (6) can be computed in closed form as:

$$\delta_{\mathcal{X}}^* = (A^\top \Omega A)^{-1} A^\top \Omega b \quad (7)$$

and, at each iteration, the linearization point is updated via $\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}} + \delta_{\mathcal{X}}^*$. The matrix $(A^\top \Omega A)$ is usually referred to as the *Hessian*. So far we assumed that \mathcal{X} belongs to a vector space (for this reason the sum $\hat{\mathcal{X}} + \delta_{\mathcal{X}}^*$ is well defined). When \mathcal{X} includes variables belonging to a smooth manifold (e.g., rotations), the structure of the GN method remains unaltered, but the usual sum (e.g., $\hat{\mathcal{X}} + \delta_{\mathcal{X}}^*$) is substituted with a suitable mapping, called *retraction* [1]. In the robotics literature, the retraction is often denoted with the operator \oplus and maps the “small correction” $\delta_{\mathcal{X}}^*$ defined in the tangent space of the manifold at $\hat{\mathcal{X}}$, to an element of the manifold, i.e., the linearization point is updated as $\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}} \oplus \delta_{\mathcal{X}}^*$.

The key insight behind modern SLAM solvers, is that the Jacobian matrix A appearing in (7) is sparse and its sparsity structure is dictated by the topology of the underlying factor graph. This enables the use of fast linear solvers to compute $\delta_{\mathcal{X}}^*$ [146, 147, 164, 229]. Moreover, it allows designing *incremental* (or *online*) solvers, which update the estimate of \mathcal{X} as new observations are acquired [146, 147, 229]. Current SLAM libraries (e.g., GTSAM [83], g2o [164], Ceres [3], iSAM [146], and SLAM++ [229]) are able to solve problems with tens of thousands of variables in few seconds.

The SLAM formulation described so far is commonly referred to as *maximum-a-posteriori* estimation, *factor graph optimization*, *graph-SLAM*, *full smoothing*, or *smoothing and mapping* (SAM). A popular particularization of this framework is *pose graph optimization*, in which the variables to be estimated are poses sampled along the trajectory of the robot, and each factor imposes a constraint on a pair of poses.

MAP estimation has been proven to be more accurate and efficient than original approaches for SLAM, based on nonlinear filtering. We refer the reader to the survey [13, 86] for an overview on filtering approaches, and to [264] for a comparison between filtering and smoothing approaches. However, we remark that some SLAM systems based on EKF have also been demonstrated to attain state-of-the-art performance. Excellent examples of EKF-based SLAM systems include the Multi-State Constraint Kalman Filter of Mourikis and Roumeliotis [193], and the VIN systems of Kottas *et al.* [160] and Hesch *et al.* [126]. Not surprisingly, the performance mismatch between filtering and MAP estimation gets smaller when the linearization point for EKF is accurate (as it happens in visual-inertial navigation problems), when using sliding-window filters, and when potential sources of inconsistency in EKF are taken care of [126, 130, 160].

As discussed in the next section, MAP estimation is usually performed on a pre-processed version of the sensor data. In this regard, it is often referred to as the SLAM *back-end*.

Sensor-dependent SLAM front-end. In practical robotics applications, it might be hard to write directly the sensor measurements as an analytic function of the state, as required in MAP estimation. For instance, if the raw sensor data is an image, it might be hard to express the intensity of each pixel as a function of the SLAM state; the same difficulty arises with simpler sensors (e.g., a laser with a single beam). In both cases the issue is connected with the fact that we are not able to design a sufficiently general, yet tractable representation of the environment; even in presence of such general representation, it would be hard to write an analytic function that connects the measurements to the parameters of such representation.

For this reason, before the SLAM back-end, it is common to have a module, the *front-end*, that extracts relevant features from the sensor data. For instance, in vision-based SLAM, the front-end extracts the pixel location of few distinguishable points in the environment; pixel observations of these points are now easy to model within the back-end (see Example 1). The front-end is also in charge of associating each measurement to a specific landmark (say, 3D point) in the environment: this is the so called *data association*. More abstractly, the data association module associates each measurement z_k with a subset of unknown variables \mathcal{X}_k such that $z_k = h_k(\mathcal{X}_k) + \epsilon_k$. Finally, the front-end might also provide an initial guess for the variables in the nonlinear optimization (5).

A pictorial representation of a typical SLAM system is given in Fig. 2. The data association module in the front-end includes a short-term data association block and a long-term one. Short-term data association is responsible for associating corresponding features in consecutive sensor measurements; for instance, short-term data association would track the fact that 2 pixel measurements in consecutive frames are picturing the same 3D point. On the other hand, long-term data association (or loop closure) is in charge of associating new measurements to older landmarks. We remark that the back-end usually feeds feedback information to the front-end, e.g., to support loop closure detection and validation.

The pre-processing that happens in the front-end is sensor dependent, since the notion of *feature* changes depending on the input data stream we consider. In the colored box below we provide an example of SLAM for a robot equipped with a monocular camera and a 3D laser scanner.

III. LONG-TERM AUTONOMY I: ROBUSTNESS

A SLAM system might be fragile in many aspects: failure can be algorithmic² or hardware-related. The former class includes failure modes induced by limitation of the existing SLAM algorithms (e.g., difficulty to handle extremely dynamic environments). The latter includes failures due to sensor or actuator degradation. Explicitly addressing these failure modes is crucial for long-term operation, where one can no longer take simplified assumptions on the structure of the environment (e.g., mostly static) or fully rely on on-board sensors. In this section we review the main challenges

²We omit the (large) class of software-related failures, while the non-expert reader must be aware that integration and testing are key aspects of SLAM and robotics in general.

Example 1: SLAM with camera and laser. We consider the SLAM problem of Fig. 3, describing a robot equipped with an uncalibrated monocular camera and a 3D laser scanner. The SLAM state at time t is $\mathcal{X} = \{x_1, \dots, x_t, l_1, \dots, l_n, K\}$, where x_i is the 3D pose of the robot at time i , l_j is the 3D position of the j -th landmark, and K is the unknown camera calibration.

The laser front-end feeds laser scans to the iterative-closest-point (ICP) algorithm to compute the relative motion (odometry) between two consecutive laser scans. Using the odometry measurements, the back-end defines *odometry* factors, denoted with “u” in Fig. 3. Each odometry factor connects consecutive poses, and the corresponding measurement z_i^u is described by the measurement model:

$$z_i^u = h_i^u(x_i, x_{i+1}) \oplus \epsilon_i^u \quad (8)$$

where $h_i^u(\cdot)$ is a function computing the relative pose between x_i and x_{i+1} , and ϵ_i^u is a zero-mean Gaussian noise with information matrix Ω_i^u . In (8), the standard sum is replaced with the retraction \oplus which maps a vector to an element of $\text{SE}(3)$, the manifold of 3D poses. Loop closing factors, denoted with “c” in Fig. 3, generalize the odometry factors, in that they measure the relative pose between non-sequential robot poses:

$$z_{ij}^c = h_{ij}^c(x_i, x_j) \oplus \epsilon_{ij}^c \quad (9)$$

where $h_{ij}^c(\cdot)$ computes the relative pose between x_i and x_j , and the noise ϵ_{ij}^c has information matrix Ω_{ij}^c .

The visual front-end extracts visual features from each camera image, acquired at time i ; then, the data association module matches each feature observation to a specific 3D landmark l_j . From this information, the back-end defines *vision* factors, denoted with “v” in Fig. 3. Each vision factor includes a measurement z_{ij}^v and its measurement model is

$$z_{ij}^v = h_{ij}^v(x_i, l_j, K) + \epsilon_{ij}^v \quad (10)$$

where z_{ij}^v is the pixel measurement of the projection of landmark l_j onto the image plane at time i . The function $h_{ij}^v(\cdot)$ is a standard perspective projection.

Finally, the prior factor “p” encodes our prior knowledge of the initial pose of the robot:

$$z^p = x_1 \oplus \epsilon_{ij}^v \quad (11)$$

where z^p is the prior mean and ϵ_{ij}^v has information matrix Ω^p . Since the on-board sensors in our example cannot measure any external reference frame, we conventionally set the prior mean z^p to be the identity transformation.^a

Given all these factors, the SLAM back-end estimates \mathcal{X} by minimizing the following nonlinear least-squares problem:

$$\begin{aligned} \mathcal{X}_k^* = \min_{\mathcal{X}_k} & \|z^p \ominus x_1\|_{\Omega^p}^2 + \sum_{i \in \mathcal{O}} \|z_{ij}^c \ominus h_{ij}^c(x_i, x_j)\|_{\Omega_i^u}^2 + \\ & \sum_{ij \in \mathcal{L}} \|z_{ij}^c \ominus h_{ij}^c(x_i, x_j)\|_{\Omega_{ij}^c}^2 + \sum_{ij \in \mathcal{P}} \|z_{ij}^v - h_{ij}^v(x_i, l_j, K)\|_{\Omega_{ij}^v}^2 \end{aligned}$$

where \ominus can be roughly understood as a “difference” on the manifold $\text{SE}(3)$, and \mathcal{O} , \mathcal{L} and \mathcal{P} identify the set of odometry, loop closure, and vision factors, respectively. Note how the front-end performs an “abstraction” of the sensor data into relative pose measurements in the factor graph.

^aThe prior is useful to make the MAP estimate unique: in absence of a global reference frame, the MAP estimator would admit an infinite number of solutions, which are equal up to roto-translation.

to algorithmic robustness. We then discuss open problems, including robustness against hardware-related failures.

One of the main sources of algorithmic failures is data association. As mentioned in Section II data association matches each measurement to the portion of the state the measurement refers to. For instance, in feature-based visual SLAM, it associates each visual feature to a specific landmark. Perceptual aliasing, the phenomenon for which different sensory inputs lead to the same sensor signature, makes this problem particularly hard. In such a case, data association establishes wrong measurement-state matches (false positives), which in turns result in wrong estimates from the back-end.

The situation is made worse by the presence of unmodeled dynamics in the environment including both short-term and seasonal changes, which might deceive short-term and long-term data association. A fairly common assumption in current SLAM approaches is that the world remains unchanged as the robot moves through it (in other words, landmarks are static). This *static world assumption* holds true in a single mapping run in small scale scenarios, as long as there are no *short term dynamics* (e.g., people and objects moving around). When mapping over longer time scales and in large environments, change is inevitable. The variations over day and night, seasonal changes such as foliage, or change in the structure of the environment such as old building giving way to new constructions, all affect the performance of a SLAM systems. For instance, systems relying on the repeatability of visual features fail in drastic illumination changes between day and night, while methods leveraging the geometry of the environment might fail when old structures disappear.

Another aspect of robustness is that of doing SLAM in harsh environments such as underwater [19, 89, 92, 151]. The challenge in this case is the limited visibility, the constantly changing conditions, and the impossibility of using conventional sensors (e.g., laser range finder).

Brief Survey. Robustness issues connected to wrong data association can be addressed in the front-end and/or in the back-end of a SLAM system. Traditionally, the front-end has been entrusted to establishing correct data association. Short-term data association is the easiest one to tackle: if the sampling rate of the sensor is relatively fast, compared to the dynamics of the robot, tracking features that correspond to the same 3D landmark is easy. For instance, if we want to track a 3D point across consecutive images and assuming that the framerate is sufficiently high, standard approaches based on descriptor matching and optical flow [248] ensure reliable tracking. Intuitively, at high framerate, the viewpoint of the sensor (camera, laser) does not change significantly, hence the features at time $t+1$ remain close to the ones observed at time t .³ Long-term data association in the front-end is more challenging and involves loop closure *detection* and *validation*. For loop closure detection at the front-end, a brute-force approach which detects features in the current measurement (e.g., image) and tries to match them against all previously detected features quickly becomes impractical. Bag-of-words [256]

³In hindsight, the fact that short-term data association is much easier and more reliable than the long-term one is what makes (visual, inertial) odometry simpler than SLAM.

avoid this intractability by quantizing the feature space and allowing more efficient search. Bag-of-words can be arranged into hierarchical vocabulary trees [208] that enable efficient lookup in large-scale datasets. Bag-of-words-based techniques such as [71, 72, 108] have shown reliable performance on the task of single session loop closure detection. However, these approaches are not capable of handling severe illumination variations as visual words can no longer be matched. This has led to develop new methods that explicitly account for such variations by matching sequences [192], gathering different visual appearances into a unified representation [64], or using spatial as well as appearance information [127]. A detailed survey on visual place recognition can be found in Lowry *et al.* [177]. Feature-based methods have also been used to detect loop closures in laser-based SLAM front-ends; for instance, Tipaldi *et al.* [273] propose FLIRT features for 2D laser scans.

Loop closure validation, instead, consists in additional geometric verification steps to ascertain the quality of the loop closure. In vision-based applications, RANSAC is commonly used for geometric verification and outlier rejection, see [248] and the references therein. In laser-based approaches, one can validate a loop closure by checking how well the current laser scan matches the existing map (i.e., how small is the residual error resulting from scan matching).

Despite the progress made to robustify loop closure detection at the front-end, in presence of perceptual aliasing, it is unavoidable that wrong loop closures are fed to the back-end. Wrong loop closures can severely corrupt the quality of the MAP estimate [268]. In order to deal with this issue, a recent line of research [2, 49, 167, 214, 267] proposes techniques to make the SLAM back-end resilient against spurious measurements. These methods reason on the validity of loop closure constraints by looking at the residual error induced by the constraints during optimization. Other methods, instead, attempt to detect outliers *a priori*, that is, before any optimization takes place, by identifying incorrect loop closures that are not supported by the odometry [215, 243].

In dynamic environments, the challenge is twofold. First, the SLAM system has to detect, discard, or track changes. While mainstream approaches attempt to discard the dynamic portion of the scene [199], some works include dynamic elements as part of the model [241, 282, 283]. The second challenge consists in the fact that the SLAM system has to model permanent or semi-permanent changes, and understand how and when to update the map accordingly. Current SLAM systems that deal with dynamics either maintain multiple (time-dependent) maps of the same location [79, 159], or have a single representation parameterized by some time-varying parameter [161, 247].

Open Problems. In this section we review open problems and novel research questions arising in life-long SLAM.

Failsafe SLAM and recovery: Despite the progress made on the SLAM back-end, current SLAM solvers are still vulnerable in presence of outliers. This is mainly due to the fact that virtually all robust SLAM techniques are based on iterative optimization of nonconvex costs. This has two consequences: first, the outlier rejection outcome depends on the quality of the initial guess fed to the optimization; second, the system is

inherently fragile: the inclusion of a single outlier degrades the quality of the estimate, which in turn degrades the capability of discerning outliers later on. An ideal SLAM solution should be *fail-safe* and *failure-aware*, i.e., the system needs to be aware of imminent failure (e.g., due to outliers or degeneracies) and provide recovery mechanisms that can re-establish proper operation. None of the existing SLAM approaches provides these capabilities.

Robustness to HW failure: While addressing hardware failures might appear outside the scope of SLAM, these failures impact the SLAM system, and the latter can play a key role in detecting and mitigating sensor and locomotion failures. If the accuracy of a sensor degrades due to malfunctioning or aging, the quality of the sensor measurements (e.g., noise, bias) does not match the noise model used in the back-end (c.f. eq. (3)), leading to poor estimates. This naturally poses different research questions: how can we detect degraded sensor operation? how can we adjust sensor noise statistics (covariances, biases) accordingly?

Metric Relocalization: While appearance-based, as opposed to feature-based, methods are able to close loops between a day and a night sequences [64, 192], or between different seasons [202] the resulting loop closure is topological in nature. For metric relocalization (i.e., estimating the relative pose with respect to the previously built map), feature-based approaches are still the norm [108] and cannot be extended to such circumstances. While vision has become the sensor of choice for many application and loop closing has become a sensor signature matching problem [71, 72, 108, 273], other sensors as well as information inherent to the SLAM problem might be exploited. For instance, Brubaker *et al.* [32] use trajectory matching to address the shortcomings of cameras. Additionally mapping with one sensor modality and localizing in the same map with a different sensor modality can be a useful addition. The work of Wolcott *et al.* [297] and Forster *et al.* [99] is a step in this direction. Another problem is localizing from significantly different viewpoints with heterogeneous sensing. Forster *et al.* [99] study vision-based localization in a lidar-based map. Majdik *et al.* [182] study how to localize a drone in a cadastral 3D model textured from Google Street View images, Behzadin *et al.* [20] show how to localize in hand-drawn maps using a laser scanner, and Winterhalter *et al.* [296] carry out RGB-D localization given 2D floor plans.

Time varying and deformable maps: Mainstream SLAM methods have been developed with the rigid and static world assumption in mind; however, the real world is non rigid both due to dynamics as well as the inherent deformability of objects. An ideal SLAM solution should be able to reason about dynamics in the environment including non-rigidity, work over long time period generating “all terrain” maps and be able to do so in real time. In the computer vision community, there have been several attempts since the 80s to recover shape from non-rigid objects but with restrictive applicability, for example Pentland *et al.* [222, 223] require prior knowledge of some mechanical properties of the objects, Bregler *et al.* in [31] and [274] requires a restricted deformation of the object and show examples in human faces. Recent results in non-rigid SfM such as [5, 6, 114] are less restrictive

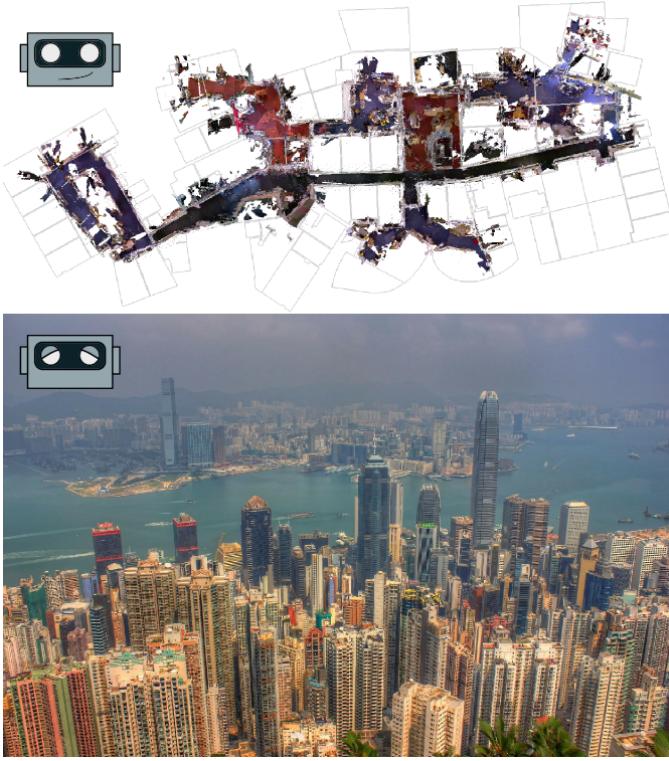


Fig. 4: Mapping large-scale real-world environments poses formidable challenges to the existing SLAM algorithms. Perceptual aliasing and dynamics of the environment (moving objects, appearance and geometry changes across seasons) call for algorithms that are inherently robust to these phenomena, as well as fail detection and recovery techniques. The extended scale of the environment calls for estimation algorithms that can be easily parallelized, as well as better map representation that store information in a compact and actionable way.

but still work in small scenarios. In the SLAM community, Newcombe *et al.* [204] have addressed the non-rigid case for small-scale reconstruction. However, addressing the problem of non-rigid maps at a large scale is still largely unexplored.

Automatic parameter tuning: SLAM systems (in particular, the data association modules) require extensive parameter tuning in order to work correctly for a given scenario. These parameters include thresholds that control feature matching, RANSAC parameters, and criteria to decide when to add new factors to the graph or when to trigger a loop closing algorithm to search for matches. If SLAM has to work “out of the box” in arbitrary scenarios, methods for automatic tuning of the involved parameters need to be considered.

IV. LONG-TERM AUTONOMY II: SCALABILITY

While modern SLAM algorithms have been successfully demonstrated in indoor building-scale environments (Fig. 4, top), in many application endeavors, robots must operate for an extended period of time over larger areas (Fig. 4, bottom). These applications include ocean exploration for environmental monitoring, 24/7 cleaning robots in our cities, or large-scale precision agriculture. For such applications the size of the factor graph underlying SLAM can grow unbounded, due to the continuous exploration of new places and the increasing time of operation. In practice, the computational time and

memory footprint are bounded by the resources of the robot. Therefore, it is important to design SLAM methods whose computational and memory complexity remains bounded.

In the worst-case, successive linearization methods based on *direct* linear solvers imply a memory consumption which grows quadratically in the number of variables in (6). When using iterative linear solvers (e.g., the conjugate gradient [82]) the memory consumption grows linearly in the number of variables. The situation is further complicated by the fact that, when re-visiting a place multiple times, factor graph optimization become less efficient as nodes and edges are continuously added to the same spatial region, compromising the sparsity structure of the graph.

In this section we review some of the current approaches to control, or at least reduce, the growth of the size of the problem and discuss open challenges.

Brief Survey. We focus on two ways to reduce the complexity of factor graph optimization: (i) *sparsification methods*, which trade-off information loss for memory and computational efficiency, and (ii) *out-of-core and multi-robot methods*, which split the computation among many robots/processors.

Node and edge sparsification. This family of methods addresses scalability by reducing the number of nodes *added* to the graph, or by *pruning* less “informative” nodes and factors. Ila *et al.* [135] use an information-theoretic approach to add only non-redundant nodes and highly-informative measurements to the graph. Johannsson *et al.* [141], when possible, avoid adding new nodes to the graph by inducing new constraints between existing nodes, such that the number of variables grows only with size of the explored space and not with the mapping duration. Kretzschmar *et al.* [162] propose an information-based criterion for determining which nodes to marginalize in pose graph optimization. Carlevaris-Bianco and Eustice [37], and Mazuran *et al.* [186] introduce the *Generic Linear Constraint* (GLC) factors and the *Nonlinear Graph Sparsification* (NGS), respectively. These methods operate on the Markov blanket of a marginalized node and compute a sparse approximation of the blanket. Huang *et al.* [129] sparsify the Hessian matrix by solving an ℓ_1 -regularized minimization problem.

Out-of-core (parallel) SLAM: Parallel *out-of-core* algorithms for SLAM split the computation (and memory) load of factor graph optimization among multiple processors. The key idea is to divide the factor graph in different subgraphs and optimize the overall graph by alternating local optimization of each subgraph, with a global refinement. The corresponding approaches are often referred to as *submapping algorithms*. Ni *et al.* [205] present an exact submapping approach for factor graph optimization, and propose to cache the factorization of the submaps to speed-up computation. Ni and Dellaert [206] organize submaps in a binary tree structure and use nested dissection to minimize the dependence between two subtrees. Zhao *et al.* [304] present an approximation strategy for large scale SLAM by solving a sequence of submaps and merging them in a divide and conquer manner using linear least squares. Frese *et al.* [104] propose a multi-level relaxation approach. Grisetti *et al.* [116] propose a hierarchy of submaps: whenever an observation is acquired, the highest level of the hierarchy

is modified and only the areas which are substantially affected are changed at lower levels. Suger *et al.* [266] present an approximate SLAM approach based on hierarchical decomposition. Choudhary *et al.* [62] parallelize computation via the Alternating Direction Method of Multipliers. Klein and Murray [155] approximately decouple localization and mapping in two threads that run in parallel. Sibley *et al.* [253] reduces the computational cost by optimizing a relative parametrization over a local neighbourhood. Strasdat *et al.* [263] take a two-stage approach and optimize first a local pose-features graph and then a pose-pose graph as done by [253]. Williams *et al.* [295] split factor graph optimization in a fast “filtering” tread and a slower “smoothing” tread, which are periodically synchronized.

Distributed multi robot SLAM: One way of mapping a large-scale environment is to deploy multiple robots doing SLAM, and divide the scenario in smaller areas, each one mapped by a different robot. This approach has two main variants: the *centralized* one, where robots build submaps and transfer the local information to a central station that performs inference [7, 14, 85, 98, 152, 169, 187, 238], and the *decentralized* one, where there is no central data fusion and the agents leverage local communication to reach consensus on a common estimate. Nerurkar *et al.* [201] propose an algorithm for cooperative localization based on distributed conjugate gradient. Aragues *et al.* [9] use a distributed Jacobi approach to estimate a set of 2D poses [10]. Araguez *et al.* [11] investigate consensus-based approaches for map merging. Knuth and Barooah [157] estimate 3D poses using distributed gradient descent. Cunningham *et al.* [73, 74] use Gaussian elimination, and develop an approach, called DDF-SAM, in which each robot exchanges a Gaussian marginal over the *separators* (i.e., the variables shared by multiple robots). A recent survey on multi-robot SLAM approaches can be found in [244].

While Gaussian elimination has become a popular approach it has two major shortcomings. First, the marginals to be exchanged among the robots are dense, and the communication cost is quadratic in the number of separators. This motivated the use of sparsification techniques to reduce the communication cost [220]. The second reason is that Gaussian elimination is performed on a linearized version of the problem, hence approaches as DDF-SAM [74] require good linearization points and complex bookkeeping to ensure consistency of the linearization points across the robots. An alternative approach to Gaussian elimination is the Gauss-Seidel approach of Choudhary *et al.* [63], which implies a communication burden which is linear in the number of separators.

Open Problems. Despite the amount of work to reduce complexity of factor graph optimization, the literature has large gaps on other aspects related to long-term operation.

Map maintenance: A fairly unexplored question is how to store the map during long-term operation. Even when memory is not a tight constraint, e.g. data is stored on the cloud, raw representations as point clouds or volumetric maps (see also Section V) are wasteful in terms of memory; similarly, storing feature descriptors for vision-based SLAM quickly becomes cumbersome. Some initial solutions have been recently proposed for localization against a compressed known map [180]

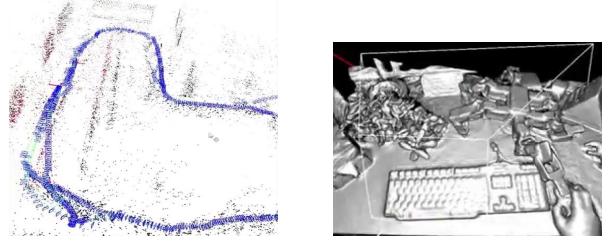


Fig. 5: Left: feature-based map of a room produced by ORB-SLAM [198]. Right: dense map of a desktop produced by DTAM [203].

and, for memory-efficient dense reconstruction [156].

Robust distributed mapping: While approaches for outlier rejection have been proposed in the single robot case, the literature on multi robot SLAM completely overlooks the problem of outliers. Dealing with spurious measurements is particularly challenging for two reasons. First, the robots might not share a common reference frame, making harder to detect and reject wrong loop closures. A first work that tackles this issue is [136]. Second, in the distributed setup, the robots have to detect outliers from very partial and local information; this setup appears completely unexplored.

Learning, forgetting, remembering: A related open question for long-term mapping is how often to update the information contained in the map and how to decide when this information becomes outdated and can be discarded. When is it fine to forget? What can be forgotten and what is essential to maintain? While this is clearly task-dependent, no grounded answer to these questions has been proposed in the literature.

Resource-constrained platforms: Another relatively unexplored issue is how to adapt existing SLAM algorithms to the case in which the robotic platforms has *severe* computational constraints. This problem is of great importance when the size of the platform is scaled down, e.g., mobile phones, micro aerial vehicles, or robotic insects [298]. Many SLAM algorithms are too expensive to run on these platforms, and it would be desirable to have algorithms in which one can tune a “knob” which allows to gently trade-off accuracy for computational cost. Similar issues arise in the multi-robot setting: how can we guarantee reliable operation for multi robot teams when facing tight bandwidth constraints and communication dropout? The “version control” approach of Cielewski *et al.* [65] is a first study in this direction.

V. REPRESENTATION I: METRIC REASONING

This section discusses how to model geometry in SLAM. More formally, a *metric representation* (or metric map) is a symbolic structure that encodes the geometry of the environment. We claim that understanding how to choose a suitable metric representation for SLAM (and extending the set of representations currently used in robotics) will impact many research areas, including long-term navigation, physical interaction with the environment, and human-robot interaction.

Geometric modeling appears much simpler in the 2D case, with only two predominant paradigms: *landmark-based maps* and *occupancy grid maps*. The former models the environment as a sparse set of landmarks, the latter discretizes the environment in cells and assigns a probability of occupation to each

cell. The problem of standardization of these representations in the 2D case has been tackled by the *IEEE RAS Map Data Representation Working Group*, which recently released a standard for 2D maps in robotics [119]; the standard defines the two main metric representations for planar environments (plus topological maps) so to facilitate data exchange, benchmarking, and technology transfer.

The question of 3D geometry modeling is more delicate, and the understanding of how to efficiently model 3D geometry during mapping is in its infancy. In this section we review metric map representations, taking a broad perspective across robotics, computer vision, computer aided design (CAD), and computer graphics. Our taxonomy draws inspiration from [96, 237, 251], and includes pointers to more recent works.

Landmark-based sparse representations. Most SLAM methods represent the scene as a set of *sparse* 3D landmarks corresponding to discriminative features in the environment (e.g., lines, corners) [198]; one example is shown in Fig. 5(left). These are commonly referred to as *landmark-based* or *feature-based* representations, and have been widespread in mobile robotics since early work on localization and mapping [170], and in computer vision in the context of *Structure from Motion* [4, 275]. A common assumption underlying these representations is that the landmarks are distinguishable, i.e., sensor data measure some geometric aspect of the landmark, but also provide a *descriptor* which establishes a (possibly uncertain) data association between each measurement and the corresponding landmark. While a large body of work focuses on the estimation of point features, the robotics literature includes extension to more complex geometric landmarks, including lines, segments, or arcs [179].

Low-level raw dense representations. Contrarily to landmark-based representations, dense representations attempt to provide high-resolution models of the 3D geometry; these models are more suitable for obstacles avoidance and path planning in robotics, or for visualization and rendering, see Fig. 5(right). Among dense models, *raw representations* describe the 3D geometry by means of a large unstructured set of points (i.e., *point clouds*) or polygons (i.e., *polygon soup* [252]). Point clouds have been widely used in robotics, in conjunction with stereo and RGB-D cameras, as well as 3D laser scanners [209]. These representations have recently gained popularity in monocular SLAM, in conjunction with the use of *direct methods* [138, 203, 228], which estimate the trajectory of the robot and a 3D model directly from the intensity values of all the image pixels. Slightly more complex representations are *surfels maps*, which encode the geometry as a set of disks [294]. While these representations are visually pleasant, they are usually cumbersome as they require storing a large amount of data. Moreover, they give a low-level description of the geometry, neglecting, for instance, the topology of the obstacles.

Boundary and spatial-partitioning dense representations. These representations go beyond unstructured set of low-level primitives (e.g., points) and attempt to explicitly represent surfaces (or *boundaries*) and volumes. Boundary representations (b-reps) define 3D objects in terms of their surface boundary. Particularly simple boundary representations

are plane-based models, which have been used for mapping by Castle *et al.* [57] and Kaess [145, 179]. More general b-reps include *curve-based representations* (e.g., tensor product of NURBS or B-splines), *surface mesh models* (connected sets of polygons), and *implicit surface representations*. The latter specify the surface of a solid as the zero crossing of a function defined on \mathbb{R}^3 [139]; examples of functions include *radial-basis functions* [51], *signed-distance function* [75], and *truncated signed-distance function* (TSDF) [302]. TSDF are currently a popular representation for vision-based SLAM in robotics. Mesh models have been also used in [293, 294].

Spatial-partitioning representations define 3D objects as a collection of contiguous non-intersecting primitives. The most popular spatial-partitioning representation is the so called *spatial-occupancy enumeration*, which decomposes the 3D space into identical cubes (*voxels*), arranged in a regular 3D grid. More efficient partitioning schemes include *octree*, *Polygonal Map octree*, and *Binary Space-Partitioning tree* [96, §12.6]. In robotics, octree representations have been used for 3D mapping [91], while commonly used *occupancy grid maps* [87] can be considered as probabilistic variants of spatial-partitioning representations. In 3D environments without hanging obstacles, 2.5D elevation maps have been also used [29]. Before moving to higher-level representations, let us better understand how sparse (feature-based) representations (and algorithms) compare to dense ones in visual SLAM.

Which one is best: feature-based or dense, direct methods? The answer depends on the task and on the performance (i.e., time, accuracy, failure rate, etc.). The disadvantage of feature-based approaches is the dependence of feature type, the reliance on numerous detection and matching thresholds, the necessity for robust estimation techniques to deal with incorrect correspondences, and the fact that most feature detectors are optimized for speed rather than precision. Dense, direct methods exploit all the information in the image, even from areas where gradients are small; thus, they can outperform feature-based methods in scenes with poor texture, defocus, and motion blur [203, 228]. However, they require high computing power (GPUs) for real-time performance. Furthermore, how to jointly estimate dense structure and motion is still an open problem (currently they can be only be estimated subsequently to one another). To avoid the caveats of feature-based methods there are two alternatives. *Semi-dense* methods overcome the high-computation requirement of dense method by exploiting only pixels with strong gradients (i.e., edges) [88, 101]; *semi-direct* methods instead leverage both sparse features (such as corners or edges) and direct methods [100, 101, 140] and are proven to be the most efficient [101]; additionally, because they rely on sparse features, they allow joint estimation of structure and motion.

High-level object-based representations. While point clouds and boundary representations are currently dominating the landscape of dense mapping, we envision that higher-level representations, including objects and solid shapes, will play a key role in the future of SLAM. Early techniques to include object-based reasoning in SLAM are “SLAM++” from Salas-Moreno *et al.* [245], the work from Civera *et al.* [66], and Dame *et al.* [77]. Solid representations explicitly encode the

fact that real objects are three-dimensional rather than 1D (i.e., points), or 2D (surfaces). Modeling objects as solid shapes allows associating physical notions, such as volume and mass, to each object, which is definitely important for robots which have to interact with the world. Luckily, existing literature from CAD and computer graphics paved the way towards these developments. Few examples of solid representations are:

- *Parameterized Primitive Instancing*: relies on the definition of *families* of objects (e.g., cylinder, sphere). For each family, one defines a set of parameters (e.g., radius, height), that uniquely identifies a member (or *instance*) of the family. To the best of our knowledge, no SLAM method is currently able to produce such a representation.
- *Sweep representations*: define a solid as the sweep of a 2D or 3D object along a trajectory through space. Typical sweeps representations include translation sweep (or *extrusion*) and rotation sweep. For instance, a cylinder can be represented as a translation sweep of a circle along an axis that is orthogonal to the plane of the circle. Sweeps of 2D cross-sections are known as *generalized cylinders* in computer vision [25], and they have been used in robotic grasping [225]. This representation has not been used in a SLAM context.
- *Constructive solid geometry*: defines complex solids by means of boolean operations between primitives [237]. An object is stored as a tree in which the leaves are the primitives and the edges represent operations. This representation has not appeared in a SLAM context.

We conclude this review by mentioning that other types of representations exist, including feature-based models in CAD [250], dictionary-based representations [242, 303], affordance-based models [154], generative and procedural models [189], and scene graphs [142]. In particular, dictionary-based representations, which define a solid as a combination of atoms in a dictionary, have been considered in robotics and computer vision, with dictionary learned from data [242, 303] or based on existing repositories of object models [166, 175].

Open Problems. The following problems regarding metric representation for SLAM deserve a large amount of fundamental research, and are still vastly unexplored.

High-level, expressive representations in SLAM: While most of the robotics community is currently focusing on point clouds or TSDF to model 3D geometry, these representations have two main drawbacks. First, they are somehow wasteful. For instance, both representations would use many parameters (i.e., points, voxels) to encode even simple environments, as an empty room. Second, these representations do not provide any high-level understanding of the 3D geometry. For instance, consider the case in which the robot has to figure out if it is moving in a room or in a corridor. A point cloud does not provide readily usable information about the type of environment (i.e., room VS corridor). On the other hand, more sophisticated models (e.g., parameterized primitive instancing) would provide easy ways to discern the two scenes (e.g., by looking at the parameters defining the primitive). Therefore, the use of higher-level representations in SLAM carries three promises. First, using more compact representations would provide a natural tool for map compression for large-scale

mapping. Second, high-level representations would provide a higher-level description of objects geometry which is a desirable feature to facilitate data association, place recognition, and semantic understanding. Such high-level description of the 3D geometry (e.g., reasoning in terms of closed solid shapes) would also impact human-robot interaction (the robot can reason in terms of geometric shapes as humans do) and allow including physical properties (e.g., weight, dynamics) as part of the inference/mapping process. Finally, using rich 3D representations would enable interactions with existing standards for construction and management of modern buildings, including CityGML [211] and IndoorGML [212]. No SLAM techniques can currently build higher-level representations, beyond point clouds, mesh models, surfels models, and TSDFs.

Optimal Representations: While there is a relatively large body of literature on different representations for 3D geometry, few works focused on understanding which criteria should guide the choice of a specific representation. Intuitively, in simple indoor environments one should prefer parametrized primitives since few parameters can sufficiently describe the 3D geometry; on the other hand, in complex outdoor environments, one might prefer mesh models. Therefore, how should we compare different representations and how should we choose the “optimal” representation?

Requicha [237] identifies few basic properties of solid representations that allow comparing different representation. Among these properties we find: *domain* (the set of real objects that can be represented), *conciseness* (the “size” of a representation for storage and transmission), *ease of creation* (in robotics this is the “inference” time required for the construction of the representation), and *efficacy in the context of the application* (this depends on the tasks for which the representation is used). Therefore, the “optimal” representation is the one that enables performing a given task, while being concise and easy to create. Soatto and Chiuso [259] define the optimal representation as a minimal sufficient statistics to perform a given task, and its maximal invariant to nuisance factors. Finding a general yet tractable framework to choose the best representation for a task remains an open problem.

Automatic, Adaptive Representations: Traditionally, the choice of a representation has been entrusted to the roboticist designing the system, but this has two main drawbacks. First, the design of a suitable representation is a time-consuming task that requires an expert. Second, it does not allow any flexibility: once the system is designed, the representation of choice cannot be changed; ideally, we would like a robot to use more or less complex representations depending on the task and the complexity of the environment. The automatic design of optimal representations will have a large impact on long-term navigation.

VI. REPRESENTATION II: SEMANTIC REASONING

Semantic mapping consists in associating semantic concepts to geometric entities in robot’s surroundings. Recently, the limitations of purely geometric maps have been recognized and this spawned a significant and ongoing body of work in semantic mapping of environments, among others to enhance

robot's autonomy and robustness, facilitate more complex tasks (e.g. avoid muddy-road while driving), move from path-planning to task-planning, and enable advanced human-robot interaction [16, 36, 190, 194, 218, 234, 246, 299]. These observations led to a large number of approaches which vary in the numbers and types of semantic concepts and means of associating them with different parts of the environments. As an example, Pronobis and Jensfelt [233] label different rooms, while Pillai and Leonard [226] segment several known objects, in the map. With the exception of few approaches, semantic parsing at the basic level was formulated as a classification problem, where simple mapping between the sensory data and semantic concepts has been considered.

Semantic vs. topological SLAM. As mentioned in Section I, topological mapping drops the metric information and only leverages place recognition to build a graph in which the nodes represent distinguishable “places”, while edges denote reachability among places. We note that *topological* mapping is radically different from *semantic* mapping. While the former requires recognizing a previously seen place (disregarding whether that place is a kitchen, a corridor, etc.), the latter is interested in classifying the place according to semantic labels. A comprehensive survey on vision-based topological SLAM is presented in Lowry *et al.* [177], and some of its challenges are discussed in Section III. In the rest of this section we focus on semantic mapping.

Semantic SLAM: Structure and detail of concepts. The unlimited number of, and relationships among, concepts for humans opens a more philosophical and task-driven decision about the level and organization of the semantic concepts. The detail and organization depend on the context of what, and where, the robot is supposed to perform a task, and they impact the complexity of the problem at different stages. A semantic representation is built by defining the following aspects:

- *Level/Detail of semantic concepts:* For a given robotic task, e.g. “going from room A to room B”, coarse categories (rooms, corridor, doors) would suffice for a successful performance, while for other tasks, e.g. “pick up a tea cup”, finer categories (table, tea cup, glass) are needed.
- *Organization of semantic concepts:* The semantic concepts are not exclusive. Even more, a single entity can have an unlimited number of properties or concepts. A chair can be “movable” and “sittable”; a dinner table can be “movable” and “unsittable”. While the chair and the table are pieces of furniture, they share the movable property but with different usability. Flat or hierarchical organizations, sharing or not some properties, have to be designed to handle this multiplicity of concepts.

Brief Survey. There are three main ways to attack semantic mapping, and assign semantic concepts to data.

- *SLAM helps Semantics:* The first robotic researchers working on semantic mapping started by the straightforward approach of segmenting the metric map built by a classical SLAM system into semantic concepts. An early work was that of Mozos *et al.* [194], which builds a geometric map using a 2D laser scan and then fuses the classified semantic places from each robot pose through an associative Markov

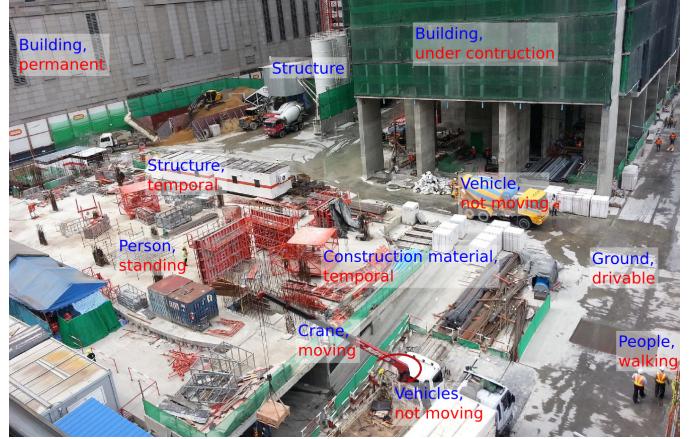


Fig. 6: Semantic understanding allows humans to predict changes in the environment at different time scales. For instance, in the construction site shown in the figure, we account for the motion of the crane and expect the crane-truck at the bottom of the figure not to move in the immediate future, while at the same time we can predict the semblance of the site which will allow us to localize ourselves even after the construction finishes. This is possible thanks to the fact that we reason on the functional properties and interrelationships of the objects in the environment. Enhancing our robots with similar capabilities is an open problem for semantic SLAM.

network in an offline manner. An online semantic mapping system was later proposed by Pronobis *et al.* [233, 234], who combine three layers of reasoning (sensory, categorical and place) to build a semantic map of the environment using laser and camera sensors. More recently, Cadena *et al.* [36] uses the motion estimation, and interconnect a coarse semantic segmentation with different object detectors to outperform the individual systems. Pillai and Leonard [226] take a monocular SLAM system to boost the performance in the task of object recognition in videos.

- *Semantics helps SLAM:* Soon after the first semantic maps came out, another trend started by taking advantage of known semantic classes or objects. The idea is that if we can recognize them in a map then we can take our prior knowledge about their geometry to improve the estimation of that map. First attempts were done in small scale by Castle *et al.* [57] and by Civera *et al.* [66] with a monocular SLAM with sparse features and by Dame *et al.* [76] with a dense map representation. Taking advantage of RGB-D sensors, Salas-Moreno *et al.* [246] propose a SLAM system based entirely on known objects and the floor plane.
- *Joint SLAM and Semantics inference:* Researchers with expertise in both computer vision and robotics realized that they could perform monocular SLAM and map segmentation within a joint formulation. The online system of Flint *et al.* [95] presents a model that leverages the Manhattan world assumption to segment the map in the main planes in indoor scenes. Bao *et al.* [16] proposed one of the first approaches to jointly estimate camera parameters, scene points and object labels using both geometric and semantic attributes in the scene. In their work, the authors demonstrate the improved object recognition performance and robustness, at the cost of a run-time of 20 minutes per image-pair, and the limited number of object categories makes the approach impractical for on-line robot operation. In the same line, Häne *et al.* [123] solve a more specialized class-dependent

optimization problem in outdoors scenarios. Although still offline, Kundu *et al.* [165] reduce the complexity of the problem by a late fusion of the semantic segmentation and the metric map, a similar idea that was proposed earlier by Sengupta *et al.* [249] using stereo cameras. It should be noted that [165] and [249] focus only on the mapping part and they do not refine the early computed poses in this late stage. Recently a promising online system was proposed by Vineet *et al.* [280] using stereo cameras and a dense map representation.

Open Problems. The problem of including semantic information in SLAM is in its infancy, and, contrarily to metric SLAM, it still lacks a cohesive formulation. Fig. 6 shows a construction site as a simple example where we can find the challenges discussed below.

Semantic mapping is much more than a categorization problem: The semantic concepts are evolving to more specialized information such as affordances and actionability⁴ of the entities in the map and the possible interactions among different active agents in the environment. How to represent these properties, and interrelationships, are questions to answer for high level human-robot interaction.

Ignorance, awareness, and adaptation: Given some prior knowledge, the robot should be able to reason about new concepts and their semantic representations, that is, it should be able to discover new objects or classes in the environment, learning new properties as result of active interaction with other robots and humans, and adapting the representations to slow and abrupt changes in the environment over time. For example, suppose that a wheeled-robot needs to classify whether a terrain is drivable or not, to inform its navigation system. If the robot finds some mud on a road, that was previously classified as drivable, the robot should learn a new class depending on the grade of difficulty of crossing the muddy region, or adjust its classifier if another vehicle stuck on the mud is perceived.

*Semantic-based reasoning*⁵: As humans, the semantic representations allow us to compress and speed-up reasoning about the environment, while assessing accurate metric representations take us some effort. Currently, this is not the case for robots. Robots can handle (colored) metric representation but they do not truly exploit the semantic concepts. Our robots are currently unable to effectively, and efficiently localize and continuously map using the semantic concepts (categories, relationships and properties) in the environment. For instance, when detecting a car, a robot should infer the presence of a planar ground under the car (even if occluded) and when the car moves the map update should only refine the hallucinated ground with the new sensor readings. Even more, the same update should change the global pose of the car as a whole in a single and efficient operation as opposed to update, for

⁴The term *affordances* refers to the set of possible actions on a given object/environment by a given agent [110]. While the term *actionability* includes the expected utility of these actions.

⁵Reasoning in the sense of localization and mapping. This is only a sub-area of the vast area of *Knowledge Representation and Reasoning* in the field of Artificial Intelligence that deals with solving complex problems, like having a dialogue in natural language or inferring a person's mood.

instance, every single voxel.

VII. NEW THEORETICAL TOOLS FOR SLAM

This section discusses recent progress towards establishing performance guarantees for SLAM algorithms, and elucidates on open problems. The theoretical analysis is important for three main reasons. First, SLAM algorithms and implementations are often tested in few problem instances and it is hard to understand how the corresponding results generalize to new instances. Second, theoretical results shed light on the intrinsic properties of the problem, revealing aspects that may be counterintuitive during empirical evaluation. Third, a true understanding of the structure of the problem allows pushing the algorithmic boundaries, enabling to extend the set of real-world SLAM instances that can be solved.

Early theoretical analysis of SLAM algorithms were based on the use of EKF; we refer the reader to [84, 286] for a comprehensive discussion. Here we focus on factor graph optimization approaches. Besides the practical advantages (accuracy, efficiency), factor graph optimization provides an elegant framework which is more amenable analysis.

In absence of priors, MAP estimation reduces to maximum-likelihood estimation. Consequently, in absence of priors, SLAM inherits all the properties of maximum likelihood estimators: the estimator in (5) is consistent, asymptotically Gaussian, asymptotically efficient, and invariant to transformations in the Euclidean space [188, Theorems 11-1,2]. Some of these properties are lost in presence of priors (e.g., the estimator is no longer invariant [188, page 193]).

In this context we are more interested in *algorithmic properties*: does a given algorithm converge to the MAP estimate? How can we improve or check convergence? What is the breakdown point in presence of spurious measurements?

Brief Survey. Most SLAM algorithms are based on iterative nonlinear optimization [81, 115, 134, 146, 147, 213]. SLAM is a nonconvex problem and iterative optimization can only guarantee local convergence. When an algorithm converges to a local minimum⁶ it usually returns an estimate that is completely wrong and unsuitable for navigation (Fig. 7). State-of-the-art iterative solvers fail to converge to a global minimum of the cost for relatively small noise levels [39, 47].

Failure to converge in iterative methods triggered efforts towards a deeper understanding of the SLAM problem. Huang and collaborators [131, 284] pioneered this effort, with initial works discussing the nature of the nonconvexity in SLAM. Huang *et al.* [132] discuss the number of minima in small pose graph optimization problems. Knuth and Barooah [158] investigate the growth of the error in absence of loop closures. Carlone [38] provides estimates of the basin of convergence for the Gauss-Newton method. Carlone and Censi [39] show that rotation estimation can be solved in closed form in 2D and show that the corresponding estimate is unique. The recent use of alternative maximum likelihood formulations (e.g., assuming Von Mises noise on rotations [40, 240]) enabled even stronger results. Carlone and Dellaert [45, 46] show that under

⁶We use the term “local minimum” to denote a stationary point of the cost which does not attain the globally optimal objective.

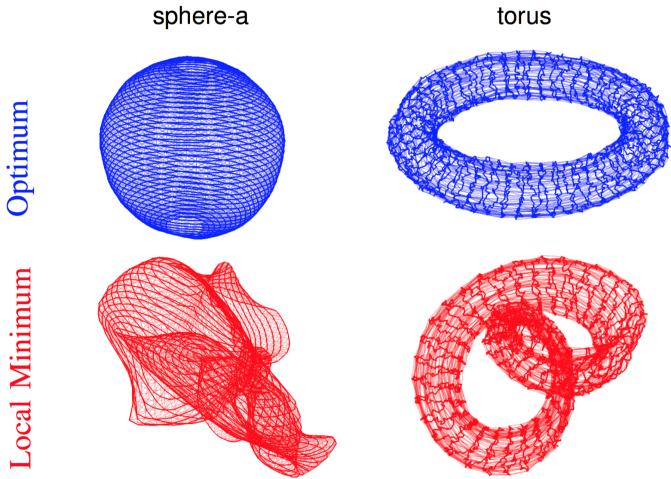


Fig. 7: The back-bone of most SLAM algorithms is the estimation of robot trajectory, which is computed via non-convex optimization. A globally optimal solution corresponds to a correct map reconstruction. However, if the estimate does not converge to the global optimum, the map reconstruction is largely incorrect, hindering robot operation. Recent theoretical tools are enabling detection of wrong convergence episodes, and are opening avenues for fail detection and recovery techniques.

certain conditions (strong duality) that is often encountered in practice, the maximum likelihood estimate is unique and pose graph optimization can be solved globally, via (convex) semidefinite programming (SDP).

As mentioned earlier, the theoretical analysis is sometimes the first step towards the design of better algorithms. Besides the dual SDP approach of [45, 46], other authors proposed convex relaxation to avoid convergence to local minima. These contributions include the work of Liu *et al.* [176] and Rosen *et al.* [240]. Another successful strategy to improve convergence consists in computing a suitably *initialization* for iterative nonlinear optimization. In this regard, the idea of solving for the rotations first and to use the resulting estimate to bootstrap nonlinear iteration has been demonstrated to be very effective in practice [27, 39, 42, 43, 47]. Khosoussi *et al.* [149] leverage the (approximate) separability between translation and rotation to speed up optimization.

Recent theoretical results on the use of Lagrangian duality in SLAM also enabled the design of *verification techniques*: given a SLAM estimate these techniques are able to judge whether such estimate is optimal or not. Being able to ascertain the quality of a given SLAM solution is crucial to design failure detection and recovery strategies for safety-critical applications. The literature on verification techniques for SLAM is very recent: current approaches [45, 46] are able to perform verification by solving a sparse linear system and are guaranteed to provide a correct answer as long as strong duality holds (more on this point later).

We note that these results, proposed in a robotics context, provide a useful complement to related work in other communities, including localization in multi agent systems [61, 224, 227, 276, 285], structure from motion in computer vision [103, 113, 124, 184], and cryo-electron microscopy [254, 255].

Open Problems. Despite the unprecedented progress of the last years, several theoretical questions remain open.



Fig. 8: A simile to the active SLAM problem is faced in strategy video games, where an agent or a group of agents need to carefully plan a strategy with limited resources to conquer an unknown territory. In active SLAM there could be one or multiple robots which will need to decide which actions to take in order to autonomously map and localize in an unknown environment. The limited resources in active SLAM are, among others, the computational capabilities of the robot (CPU), the memory size available for environmental representation (RAM), the time constraints given but high level task (Time), robot's skills in mapping and localization (Skills) and energy remaining in the battery (Battery).

Generality, guarantees, verification: The first question regards the generality of the available results. Most results on guaranteed global solutions and verification techniques have been proposed in the context of pose graph optimization. Can these results be generalized to arbitrary factor graphs?

Weak or Strong duality? The works [45, 46] show that when strong duality holds SLAM can be solved globally; moreover, they provide empirical evidence that strong duality holds in most problem instances encountered in practical applications. The outstanding problem consists in establishing *a-priori* conditions under which strong duality holds. We would like to answer the question “given a set of sensors (and the corresponding measurement noise statistics) and a factor graph structure, does strong duality hold?”. The capability to answer this question would define the domain of applications in which we can compute (or verify) global solutions to SLAM. This theoretical investigation would also provide fundamental insights in sensor design and active SLAM (Section VIII).

Resilience to outliers: The third question regards estimation in presence of spurious measurements. While recent results provide strong guarantees for pose graph optimization, no result of this kind is available in presence of outliers. Despite the work on robust SLAM (Section III) and new modeling tools for the non-Gaussian noise case [239], the design of global techniques that are resilient to outliers and the design of verification techniques that can certify the correctness of a given estimate in presence of outliers remain open.

VIII. ACTIVE SLAM

So far we described SLAM as an estimation problem that is carried out passively by the robot, i.e. the robot performs SLAM *given* the sensor data, but without acting deliberately to collect it. In this section we discuss how to leverage robot’s motion to improve the mapping and localization results.

The problem of controlling robot’s motion in order to minimize the uncertainty of its map representation and localization

is usually named *active SLAM*⁷. This definition stems from the well-known Bajcsy's active perception [15] and Thrun's robotic exploration [272, Ch. 17] paradigms.

Brief Survey. The first proposal and implementation of an active SLAM algorithm can be traced back to Feder [93, 94] while the name was coined in [171]. However, active SLAM has its roots in ideas from artificial intelligence and robotic exploration that can be traced back to the early eighties (c.f. [18]). Thrun in [270] and [269] concluded that solving the *exploration-exploitation* dilemma, i.e., finding a balance between visiting new places (exploration) and reducing the uncertainty by re-visiting known areas (exploitation), provides a more efficient alternative with respect to random exploration or pure exploitation.

Active SLAM is a decision making problem and there are several general frameworks for decision making that can be used as backbone for exploration-exploitation decisions. One of these frameworks is the Theory of Optimal Experimental Design (TOED) [221] which, applied to active SLAM [52, 53], allows selecting future robot action based on the predicted map uncertainty. Information theoretic [181, 236] approaches have been also applied to active SLAM [56, 261]; in this case decision making is usually guided by the notion of *information gain*. Control theoretic approaches for active SLAM include the use of Model Predictive Control [171, 172]. A different body of works formulates active SLAM under the formalism of Partially Observable Markov Decision Process (POMDP) [144] which in general is known to be computationally intractable; approximate but tractable solutions for active SLAM include Bayesian Optimization [185] or efficient Gaussian beliefs propagation [219], among others.

A popular framework for active SLAM consists in selecting the best future action among a finite set of alternatives. This family of active SLAM algorithms proceeds in three main steps [26, 44, 183]: 1) The robot identifies possible locations to explore or exploit, i.e. vantage locations, in its current estimate of the map; 2) The robot computes the utility of visiting each vantage point and selects the action with the highest utility; and 3) The robot carries out the selected action and decides if it is necessary to continue or to terminate the task. In the following, we discuss each point in details.

Selecting vantage points: Ideally, a robot executing an active SLAM algorithm should evaluate every possible action in the robot and map space, but the computational complexity of the evaluation grows exponentially with the search space which proves to be computationally intractable in real applications [33, 185]. In practice, a small subset of locations in the map is selected, using techniques such as frontier-based exploration [148, 300]. Recent works [279] and [137] have proposed approaches for continuous-space planning under uncertainty that can be used for active SLAM; currently these approaches can only guarantee convergence to locally optimal policies. Another recent continuous-domain avenue for active SLAM algorithms is the use of potential fields. Some examples are [278] which use convolution techniques to compute entropy

⁷Other names have been used in the literature, including *Simultaneous Planning Localization and Mapping* (SPLAM) [172], or Autonomous, Active or Integrated Exploration [183].

and select the robot's actions, and [143] which resorts to the solution of a boundary value problem.

Computing the utility of an action: Ideally, to compute the utility of a given action the robot should reason about the evolution of the posterior over the robot pose and the map, taking into account future (controllable) actions and future (unknown) measurements. If such posterior were known, an information-theoretic function, as the information gain, could be used to rank the different actions [28, 262]. However, computing this joint probability analytically is, in general, computationally intractable [44, 90, 262]. In practice, one resorts to approximations. Initial work considered the uncertainty of the map and the robot to be independent [277] or conditionally independent [262]. Most of these approaches define the utility as a linear combination of metrics that quantify robot and map uncertainties [26, 28, 41, 150]. One drawback of this approach is that the scale of the numerical values of the two uncertainties is not comparable, i.e. the map uncertainty is often orders of magnitude larger than the robot one, so manual tuning is required to correct it. Approaches to tackle this issue have been proposed for particle-filter-based SLAM [26, 44], and for pose graph optimization [56].

The Theory of Optimal Experimental Design (TOED) [221] can also be used to account for the utility of performing an action. In the TOED, every action is considered as a stochastic design, and the comparison among designs is done using their associated covariance matrices via the so-called optimality criteria, e.g. *A-opt*, *D-opt* and *E-opt*. A study about the usage of optimality criteria in active SLAM can be found in [53, 55].

Executing actions or terminating exploration: While executing an action is usually an easy task, using well-established techniques from motion planning, the decision on whether or not the exploration task is complete, is currently an open challenge that we discuss in the following paragraph.

Open Problems. Several problems still need to be addressed, for active SLAM to have impact in real applications.

Fast and accurate predictions of future states: In active SLAM each action of the robot should contribute to reduce the uncertainty in the map and improve the localization accuracy; for this purpose, the robot must be able to forecast the effect of future actions on the map and robots localization. The forecast should be fast to attain any time constraint and precise to support the decision process. In the SLAM community it is well known the importance of closing loops for reducing uncertainty and for obtaining higher localization and mapping accuracy [105]. Nonetheless, efficient methods for forecasting the occurrence and the effect of a loop closing is yet to be devised for active SLAM. Moreover, predicting the effects of future actions is still a computational expensive task [137, 185]. Recent approaches to forecasting future robots states can be found in the machine learning literature, and involve the use of spectral techniques [260] and deep learning [281].

Enough is enough: When do you stop doing active SLAM? Active SLAM is a computationally expensive task: therefore a natural question is when we can stop doing active SLAM and focus resources on other tasks. Balancing active SLAM decisions and exogenous tasks is critical, since in most real-world

tasks, active SLAM is only a means to achieve an intended goal. Additionally, having a stopping criteria is a necessity because at some point it is provable that more information would lead not only to a diminishing return effect but also, in case of contradictory information, to an unrecoverable state (e.g. several wrong loop closures). Uncertainty metrics from TOED, which are task oriented, seem promising as stopping criteria, compared to information-theoretic metrics which are difficult to compare across systems [8, 54, 125].

Performance guarantees: Another important avenue is to look for mathematical guarantees for active SLAM and for near-optimal policies. Since solving the problem exactly is intractable, it is desirable to have approximation algorithms with clear performance bounds. Examples of this kind of effort is the use of submodularity [112] in the related field of active sensors placement.

IX. NEW SENSORS FOR SLAM

Besides the development of new algorithms, progress in SLAM (and mobile robotics in general) has often been triggered by the availability of novel sensors. For instance, the introduction of 2D laser range finders enabled the creation of very robust SLAM systems, while 3D lidars have been a main thrust behind recent applications, such as autonomous cars. In the last five years, a large amount of research has been devoted to vision sensors, with successful applications in augmented reality and vision-based navigation. Standard vision sensors will be a key technology in the short-medium term, especially for micro aerial vehicles, whose payload and power limitations prevent the use of bulky and power-hungry sensors like laser range finders. Table II reports a list of SLAM datasets and the corresponding list of sensors. Note that the list of benchmarking problems is usually restricted to a small set of sensors, while some of the datasets (i.e., the ones used to exercise the back-end) are agnostic to the sensor type. At this point, a question that naturally arises is: what will be the next sensor technology to drive future long-term SLAM research? Which types of algorithms will be needed to make best use of the new sensors?

We argue that one of the reasons we do not have yet autonomous robots with the same agility and maneuverability of insects or animals, such as cheetah and birds, is the high latency and temporal discretization of standard computer-vision sensing architectures. The latency is the time we need to capture an image plus the time to process it, while the temporal discretization is the time we need to wait until the next frame arrives (cf. Fig. 9a). The latency of an average computer vision pipeline is typically in the range of 50-200 ms, while the temporal discretization ranges between 5 and 100 ms, depending on the frame rate and exposure time of the camera.⁸ High latency and temporal discretization put a hard bound on the maximum achievable agility of a robotic platform. As an example, consider that the constant time of the actuators of a medium-size drone are in the order of 30

⁸There exist high-speed cameras with frame rates up to 1Mhz; however they are bulky ($> \text{kg}$), power hungry ($> 100 \text{ W}$), and need a lot of external illumination to work.

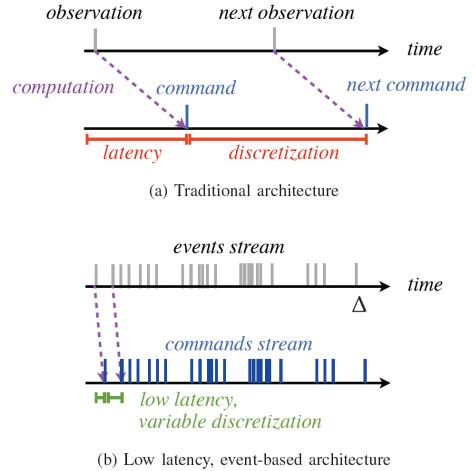


Fig. 9: Image courtesy of Censi & Scaramuzza [60]. To improve the agility of autonomous flying robots, we need to improve the total latency of the sensing pipeline. Using a device like a DVS we can theoretically obtain a sensing pipeline which has microsecond latency.

ms and can quickly fall below one millisecond if we shrink the size of the drone and its propellers/wings. Therefore, to achieve agile motion, we need faster sensors and low-latency processing. We argue that *event-based cameras* are a perfect candidate to substitute current vision sensors.

Event-based Cameras. Event cameras, such as the Dynamic Vision Sensor (DVS) [174] or the Asynchronous Time-based Image Sensor (ATIS) [231] are recently changing the landscape of vision-based perception. Contrarily to standard frame-based cameras, which send entire images at fixed frame rates, an event camera only sends the local pixel-level changes caused by movement in a scene at the time they occur. Notably, an event is triggered if the relative change of intensity perceived by a pixel is larger than a certain threshold.⁹ Thus, the output is a sequence of asynchronous events at *microsecond time* resolution (cf. Fig. 9b). More specifically, each event is a tuple $[t_k, (x_k, y_k), p_k]$, where t_k is the timestamp of the event, (x_k, y_k) are the coordinates of the pixel that triggered the event, and the binary value $p_k \in \{+1, -1\}$ is the event polarity, which is the sign of the brightness change.

An event-based camera has five key advantages compared to standard frame-based cameras: a temporal latency of 1 microsecond, a measurement rate of up to 1MHz, a dynamic range of up to 140 dB (vs 60-70 dB of standard cameras), a power consumption of 20 mW (vs 1.5 W of standard cameras), and very low bandwidth data transfer (for average motions, $\sim \text{kilobytes/s}$ vs $\sim \text{Megabytes/s}$ of standard cameras). These properties will enable the design of a new class of algorithms to track a baseball in the moonlight and develop autonomous robots with the same agility of a fly, localizing and mapping in challenging lighting and high-speed conditions. However, all these advantages come at a cost: since the output is composed by a sequence of asynchronous events, traditional frame-based computer-vision algorithms are not applicable.

⁹More specifically, in a DVS an event is triggered if the log-intensity difference is larger than 10-20%. The human eye is sensitive to log-intensity changes of 1%.

This requires a *paradigm shift* from the traditional computer vision approaches developed over the last fifty years.

Brief Survey. Since the first event camera became commercially available only in 2008 [174], the literature is still small and accounts for less than 100 papers on the algorithmic use of these sensors. Event-based adaptations of feature detection and tracking [67], LED tracking [60], epipolar geometry [21], image reconstruction, motion estimation, optical flow [17, 22, 69, 153], 3D reconstruction [50] iterative closest points [207] have already been proposed.

Recently, event-based visual odometry [58], localization [195, 196, 290], and SLAM [291, 292] algorithms have also been presented. The design goal of such algorithms is that each incoming event can asynchronously change the estimated state of the system, thus, preserving the event-based nature of the sensor and allowing the design of highly-reactive systems [68, 197]. Adaptations of the continuous-time trajectory-estimation methods [23, 107] are very suitable for event cameras [195]: since a single event does not carry enough information for state estimation and because a DVS has an average data rate of 100,000 events a second, it can become intractable to estimate the DVS pose at the discrete times of all events due to the rapidly growing size of the state vector needed to represent all such poses. Using a continuous-time framework, the DVS pose trajectory can be approximated by a smooth curve in the space of rigid-body motions using basis functions, such as cubic splines, and optimized according to the observed events [195].

Open Problems. Despite the promise of event cameras, their use does not come without practical and algorithmic challenges.

Modeling, Sensitivity, and Resolution: Event cameras have a complicated analog circuitry, with nonlinearities and biases that can change the sensitivity of the pixels, and other dynamic properties, which unfortunately make the events highly susceptible to noise. A full characterization of the sensor noise and sensor non idealities is still missing. Thus, we are still far from solving localization and mapping with event sensors.

Since events are triggered by brightness changes, events occur in correspondence of edges in the scenes. However, most proposed event pipelines have been developed for scenes with strong gradients and can fail in presence of smooth edges.

While the temporal resolution is very high ($1\mu\text{s}$), the spatial resolution is still low (128×128 pixels), an issue that is being addressed by current efforts [30]. Additionally, newly developed event sensors overcome some of the original limitations: an ATIS sensor sends the magnitude of the pixel-level brightness; a DAVIS sensor [30] can output both frames and events (this is made possible by embedding a standard frame-based sensor and a DVS into the same pixel array).

Event-based or standard camera? Which sensor is best for SLAM? The performance of a given algorithm-sensor pair for SLAM depends on the sensor and algorithm parameters, and on the environment [258]. A complete treatment of how to choose algorithms and sensors to achieve the best performance has not yet been found. However, a preliminary study by Censi et al. [59], has shown that the performance for a given task also depends on the power available for sensing. It also suggests

that the optimal sensing architecture may have multiple sensors that might be instantaneously switched on and off according to the required performance level.

X. CONCLUSION

As we enter in the *robust-perception age*, the SLAM community needs to address a new and broad set of challenges. The design of fail-safe SLAM systems, the development of self-tuning algorithms, the creation of advanced metric and semantic representations for the environment, and the implementation of task-aware and resource-aware SLAM systems are, among others, formidable challenges, that are made even more interesting by the recent availability of new theoretical tools and novel sensors. SLAM still constitutes an indispensable backbone for most robotics applications and despite the amazing progress over the past decades, existing SLAM systems are far from providing insightful, actionable, and compact models of the environment, comparable to the ones effortlessly created and used by humans.

REFERENCES

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2007.
- [2] Prabhakar Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 62–69. IEEE, 2013.
- [3] S. Agarwal, K. Mierle, and Others. Ceres solver. URL <https://code.google.com/p/ceres-solver/>.
- [4] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *European Conf. on Computer Vision (ECCV)*, pages 29–42, 2010.
- [5] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. Sequential non-rigid structure from motion using physical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(5):979–994, 2016.
- [6] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. Real-time 3d reconstruction of non-rigid shapes with a single moving camera. *Computer Vision and Image Understanding (CVIU)*, 2016, to appear.
- [7] L. Andersson and J. Nygård. C-SAM : Multi-robot SLAM using square root information smoothing. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2008.
- [8] E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers. A theoretical look at information-driven sensor management criteria. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8, July 2011.
- [9] R. Aragues, L. Carlone, G. Calafiore, and C. Sagües. Multi-agent localization from noisy relative pose measurements. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 364–369, 2011.

TABLE II: Datasets and Sensors for SLAM.

IMU (Inertial Measurement Unit), GPS (Global Positioning System), LABELS (Human-annotated labels), 2D/3D LIDAR (Light Detection And Ranging), LOC (Localization ground truth), MAP (map ground truth), RGB (color images), RGBD (color & depth images), Stereo (Bi/trinocular stereo images).

Topic	References	Data									
		IMU	GPS	Labels	LIDAR2D	LIDAR3D	LOC	MAP	RGB	RGBD	Stereo
Front-ends and back-ends in small/medium scale scenarios	[265]	✓					✓			✓	
	[289]						✓	✓		✓	
	[210]	✓	✓		✓		✓	✓			
	[257]		✓		✓		✓				✓
	[128, 217]	✓	✓	✓	✓	✓	✓		✓		
	[121, 287, 288]	✓	✓								✓
	[109]	✓	✓	✓	✓	✓	✓		✓		✓
	[34]	✓					✓	✓	✓		✓
	[230]	✓	✓			✓				✓	✓
Back-ends in small/medium scale	[48, 168]			✓			✓				
Front-ends and back-ends in synthetic scenarios	[122]						✓	✓		✓	
Loop-closures	[111, 235]		✓							✓	
	[191]			✓						✓	
	[232]			✓	✓		✓		✓		
Long-term SLAM	[24, 78]			✓	✓				✓		
	[232]			✓	✓		✓		✓		
Semantic SLAM	[70]		✓	✓			✓		✓		
	[232, 301]			✓	✓		✓		✓		
Multi-robot SLAM	[173]			✓			✓				

- [10] R. Aragues, L. Carbone, G. Calafiore, and C. Sagües. Distributed centroid estimation from noisy relative measurements. *Systems & Control Letters*, 61(7):773–779, 2012.
- [11] R. Aragues, J. Cortes, and C. Sagües. Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 2012.
- [12] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó. The slam problem: A survey. In *Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pages 363–371. IOS Press, 2008. ISBN 978-1-58603-925-7.
- [13] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.
- [14] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte. Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [15] R. Bajcsy. Active Perception. *Proceedings of the IEEE*, 76(8):966–1005, August 1988. ISSN 0018-9219. doi: 10.1109/5.5968.
- [16] S.Y. Bao, M. Bagra, Y.W. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2703 –2710, june 2012. doi: 10.1109/CVPR.2012.6247992.
- [17] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2016.
- [18] A.G. Barto and R.S. Sutton. Goal seeking components for adaptive intelligence: An initial assessment. Technical Report Technical Report AFWAL-TR-81-1070, Air Force Wright Aeronautical Laboratories. Wright-Patterson Air Force Base, 01 1981.
- [19] C. Beall, B.J. Lawrence, V. Iila, and F. Dellaert. 3D reconstruction of underwater structures. In *IEEE/RSJ*

- Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [20] Bahram Behzadian, Pratik Agarwal, Wolfram Burgard, and Gian Diego Tipaldi. Monte carlo localization in hand-drawn maps. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4291–4296. IEEE, 2015.
- [21] Ryad Benosman, Sio-Hoi Ieng, Paul Rogister, and Christoph Posch. Asynchronous event-based Hebbian epipolar geometry. *IEEE Trans. Neural Netw.*, 22(11): 1723–1734, 2011.
- [22] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012.
- [23] Charles Bibby and Ian D. Reid. A hybrid SLAM representation for dynamic marine environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.
- [24] P. Biber and T. Duckett. Experimental analysis of sample-based maps for long-term slam. *The International Journal of Robotics Research (IJRR)*, 28(1):20–33, 2009.
- [25] T.O. Binford. Visual perception by computer. In *IEEE Conf. on Systems and Controls*, 1971.
- [26] J.L. Blanco, J.A. Fernández-Madrigal, and J. Gonzalez. A Novel Measure of Uncertainty for Mobile Robot SLAM with RaoBlackwellized Particle Filters. *The International Journal of Robotics Research (IJRR)*, 27(1): 73–89, January 2008. doi: 10.1177/0278364907082610.
- [27] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2D lidar maps. *Robotics and Autonomous Systems*, 57(12):1211–1224, 2009.
- [28] F. Bourgault, A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information Based Adaptive Robotic Exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 540–545, 2002. doi: 10.1109/IRDS.2002.1041446.
- [29] C. Brand, M.J. Schuster, H. Hirschmuller, and M Suppa. Stereo-vision based obstacle mapping for indoor/outdoor slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [30] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbrück. A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor. *IEEE J. of Solid-State Circuits*, 49(10):2333–2341, 2014. ISSN 0018-9200. doi: 10.1109/JSSC.2014.2342715.
- [31] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696 vol.2, 2000. doi: 10.1109/CVPR.2000.854941.
- [32] Marcus Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, 2013.
- [33] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider. Coordinated Multi-robot Exploration. *IEEE Transactions on Robotics (TRO)*, 21(3):376–386, June 2005. ISSN 1552-3098. doi: 10.1109/TRO.2004.839232.
- [34] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. doi: 10.1177/0278364915620033.
- [35] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, J. Neira, I.D. Reid, and JJ. Leonard. Robotics: Science and systems (rss), workshop “the problem of mobile sensors: Setting future goals and indicators of progress for slam”. URL <http://ylatif.github.io/movingsensors/>.
- [36] C. Cadena, A. Dick, and I. Reid. A fast, modular scene understanding system using context-aware object detection. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, Seattle, WA, USA, May 2015.
- [37] N. Carlevaris-Bianco and R.M. Eustice. Generic factor-based node marginalization and edge sparsification for pose-graph slam. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5748–5755, 2013.
- [38] L. Carlone. Convergence analysis of pose graph optimization via Gauss-Newton methods. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 965–972, 2013.
- [39] L. Carlone and A. Censi. From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization. *IEEE Trans. Robotics*, 30(2):475–492, 2014.
- [40] L. Carlone and F. Dellaert. Duality-based verification techniques for 2D SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4589–4596, 2015.
- [41] L. Carlone, J. Du, M. Kaouk, B. Bona, and M. Indri. An Application of Kullback-Leibler Divergence to Active SLAM and Exploration with Particle Filters. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 287–293, October 2010. doi: 10.1109/IROS.2010.5652164.
- [42] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A linear approximation for graph-based simultaneous localization and mapping. In *Robotics: Science and Systems (RSS)*, 2011.
- [43] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A fast and accurate approximation for planar pose graph optimization. *Intl. J. of Robotics Research*, 33(7):965–987, 2014.
- [44] L. Carlone, Jingjing Du, M. Kaouk, B. Bona, and M. Indri. Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence. *Journal of Intelligent & Robotic Systems*, 75(2):291–311, October 2014. ISSN 0921-0296. doi: 10.1007/s10846-013-9981-9.
- [45] L. Carlone, G. Calafiore, C. Tommolillo, and F. Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Trans. on Robotics*, accepted, 2015.

- [46] L. Carlone, D. Rosen, G. Calafiole, J.J. Leonard, and F. Dellaert. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015.
- [47] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [48] Luca Carlone. Pose graph optimization datasets. <http://www.lucacarlone.com/index.php/resources/datasets>, Jun 2016. MIT.
- [49] Luca Carlone, Andrea Censi, and Frank Dellaert. Selecting good measurements via ℓ_1 relaxation: a convex approach for robust estimation over graphs. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [50] J. Carneiro, S.-H. Ieng, C. Posch, and R. Benosman. Event-based 3D reconstruction from neuromorphic retinas. *Neural Networks*, 45:27–38, 2013.
- [51] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH*, pages 67–76, 2001.
- [52] H. Carrillo, Y. Latif, J. Neira, and J. A. Castellanos. Fast Minimum Uncertainty Search on a Graph Map Representation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2504–2511, Vilamoura, Portugal, October 2012. doi: 10.1109/IROS.2012.6385927.
- [53] H. Carrillo, I. Reid, and J. A. Castellanos. On the Comparison of Uncertainty Criteria for Active SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2080–2087, St. Paul, MN, USA, May 2012. doi: 10.1109/ICRA.2012.6224890.
- [54] H. Carrillo, O. Birbach, H. Taubig, B. Bauml, U. Frese, and J. A. Castellanos. On Task-oriented Criteria for Configurations Selection in Robot Calibration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3653–3659, Karlsruhe, Germany, May 2013. doi: 10.1109/ICRA.2013.6631090.
- [55] H. Carrillo, Y. Latif, M. L. Rodríguez, J. Neira, and J. A. Castellanos. On the Monotonicity of Optimality Criteria during Exploration in Active SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2014.
- [56] H. Carrillo, P. Dames, K. Kumar, and J. A. Castellanos. Autonomous Robotic Exploration Using Occupancy Grid Maps and Graph SLAM Based on Shannon and Rényi Entropy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.
- [57] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 4102–4107, 2007.
- [58] A. Censi and D. Scaramuzza. Low-latency event-based visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.
- [59] A. Censi, E. Mueller, E. Frazzoli, and S. Soatto. A power-performance approach to comparing sensor families, with application to comparing neuromorphic to traditional vision sensors. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.
- [60] Andrea Censi, Jonas Strubel, Chistian Brandli, Tobi Delbruck, and Davide Scaramuzza. Low-latency localization by active LED markers tracking using a dynamic vision sensor. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.
- [61] A. Chiuso, G. Picci, and S. Soatto. Wide-sense estimation on the special orthogonal group. *Commun. Inf. Syst.*, 8:185–200, 2008.
- [62] S. Choudhary, L. Carlone, H.I. Christensen, and F. Dellaert. Exactly sparse memory efficient slam using the multi-block alternating direction method of multipliers. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.
- [63] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H.I. Christensen, and F. Dellaert. Distributed trajectory estimation with privacy and communication constraints: a two-stage distributed gauss-seidel approach. In *Intl. Conf. on Robotics and Automation*, submitted, 2015.
- [64] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.
- [65] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Siegwart. Map api - scalable decentralized map building for robots. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6241–6247, May 2015. doi: 10.1109/ICRA.2015.7140075.
- [66] J. Civera, D. Galvez-Lopez, L. Riazuelo, J.D. Tardos, and J. M M Montiel. Towards semantic SLAM using a monocular camera. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277–1284, 2011. doi: 10.1109/IROS.2011.6094648.
- [67] Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. Asynchronous event-based corner detection and matching. *Neural Networks*, 66:91–106, 2015. ISSN 0893-6080. doi: <http://dx.doi.org/10.1016/j.neunet.2015.02.013>. URL <http://www.sciencedirect.com/science/article/pii/S0893608015000477>.
- [68] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R.J. Douglas, and T. Delbruck. A pencil balancing robot using a pair of AER dynamic vision sensors. In *Int. Conf. on Circuits and Systems (ISCAS)*, 2009.
- [69] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger. Interacting maps for fast visual interpretation. In *Int. Joint Conf. on Neural Networks (IJCNN)*, pages 770–776, 2011.
- [70] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson,

- Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [71] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Intl. J. of Robotics Research*, 27(6):647–665, June 2008.
- [72] M. Cummins and P. Newman. Highly scalable appearance-only slam - fab-map 2.0. In *Robotics: Science and Systems (RSS)*, 2009.
- [73] A. Cunningham, M. Paluri, and F. Dellaert. DDF-SAM: Fully distributed slam using constrained factor graphs. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [74] A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.
- [75] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, 1996.
- [76] A. Dame, V.A. Prisacariu, C.Y. Ren, and I. Reid. Dense reconstruction using 3d object shape priors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1288–1295. IEEE, 2013.
- [77] Amaury Dame, Victor A. Prisacariu, Carl Y. Ren, and Ian Reid. Dense reconstruction using 3d object shape priors. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
- [78] F. Dayoub, G. Cielniak, and T. Duckett. Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robotics and Autonomous Systems*, 59(5):285–295, 2011.
- [79] F. Dayoub, G. Cielniak, and T. Duckett. Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robotics and Autonomous Systems*, 59(5):285–295, 2011.
- [80] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *Int. J. Robotics Research*, 25(12), December 2006.
- [81] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.
- [82] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C.E. Thorpe. Subgraph-preconditioned conjugate gradient for large scale slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [83] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012.
- [84] G. Dissanayake, S. Huang, Z. Wang, and R. Ranasinghe. A review of recent developments in simultaneous localization and mapping. In *2011 6th International Conference on Industrial and Information Systems*, pages 477–482, Aug 2011. doi: 10.1109/ICIINFS.2011.6038117.
- [85] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert. Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [86] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.
- [87] A. Elfes. Occupancy grids: A probabilistic framework for robot perception and navigation. *Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.
- [88] J. Engel, J. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Eur. Conf. on Computer Vision (ECCV)*, 2014.
- [89] R.M. Eustice, H. Singh, J.J. Leonard, and M.R. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *Intl. J. of Robotics Research*, 25(12):1223–1242, Dec 2006.
- [90] N. Fairfield and D. Wettergreen. Active SLAM and Loop Prediction with the Segmented Map Using Simplified Models. In A. Howard, K. Iagnemma, and A. Kelly, editors, *Field and Service Robotics*, volume 62 of *Springer Tracts in Advanced Robotics*, pages 173–182. Springer, 2010. ISBN 978-3-642-13407-4. doi: 10.1007/978-3-642-13408-1_16.
- [91] N. Fairfield, G. Kantor, and D. Wettergreen. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *J. of Field Robotics*, 24:3–21, 2007.
- [92] Nathaniel Fairfield, George Kantor, and David Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24(1-2):03–21, 2007.
- [93] H. J. S. Feder. *Simultaneous Stochastic Mapping and Localization*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [94] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive Mobile Robot Navigation and Mapping. *The International Journal of Robotics Research (IJRR)*, 18(7):650–668, July 1999. doi: 10.1177/02783649922066484.
- [95] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2228–2235, 2011.
- [96] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1992.
- [97] J. Folkesson and HI Christensen. Graphical SLAM for outdoor applications. *Journal of Field Robotics*, 23(1): 51–70, 2006.
- [98] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative monocular SLAM with multiple micro aerial vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3962–3970, 2013. URL <http://dx.doi.org/10.1109/IROS.2013.6696923>.
- [99] C. Forster, M. Pizzoli, and D. Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3971–3978, 2013. URL <http://dx.doi.org/10.1109/IROS.2013>.

- 6696924.
- [100] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014. doi: 10.1109/ICRA.2014.6906584.
 - [101] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo 2.0: Semi-direct visual odometry for monocular and multi-camera systems. *IEEE Transactions on Robotics*, accepted, January 2016.
 - [102] F. Fraundorfer and D. Scaramuzza. Visual odometry. Part II: Matching, robustness, optimization, and applications. *IEEE Robotics Automation Magazine*, 2012. doi: 10.1109/MRA.2012.2182810.
 - [103] J. Fredriksson and C. Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In *Asian Conf. on Computer Vision (ACCV)*, 2012.
 - [104] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Trans. Robotics*, 21(2):196–207, April 2005.
 - [105] Udo Frese. A discussion of simultaneous localization and mapping. *Autonomous Robots*, 20(1):25–42, 2006. ISSN 1573-7527. doi: 10.1007/s10514-006-5735-x.
 - [106] Udo Frese. Interview: Is slam solved? *KI - Künstliche Intelligenz*, 24(3):255–257, 2010. ISSN 1610-1987. doi: 10.1007/s13218-010-0047-x. URL <http://dx.doi.org/10.1007/s13218-010-0047-x>.
 - [107] Paul Furgale, Timothy D. Barfoot, and Gabe Sibley. Continuous-time batch estimation using temporal basis functions. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.
 - [108] D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2197158.
 - [109] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
 - [110] James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.
 - [111] Arren Glover, Will Maddern, Michael Milford, and Gordon Wyeth. FAB-MAP + RatSLAM: Appearance-based SLAM for Multiple Times of Day. In *ICRA*, Anchorage, USA, 2010.
 - [112] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research (JAIR)*, 42:427–486, 2011.
 - [113] V. M. Govindu. Combining two-view constraints for motion estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 218–225, 2001.
 - [114] G. Grasa, E. Bernal, S. Casado, I. Gil, and J. M. M. Montiel. Visual slam for handheld monocular endoscope. *IEEE Transactions on Medical Imaging*, 33(1):135–146, Jan 2014. ISSN 0278-0062. doi: 10.1109/TMI.2013.2282997.
 - [115] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. *Trans. on Intelligent Transportation systems*, 10(3): 428–439, 2009.
 - [116] G. Grisetti, R. Kuemmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010.
 - [117] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, winter 2010. ISSN 1939-1390. doi: 10.1109/MITS.2010.939925.
 - [118] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard. A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent. In *Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.
 - [119] IEEE RAS Map Data Representation Working Group. Ieee standard for robot map data representation for navigation, sponsor: IEEE robotics and automation society. URL <http://standards.ieee.org/findstds/standard/1873-2015.html/>.
 - [120] J. S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation CIRA*, pages 318–325, 1999.
 - [121] Roberto Guzman, Jean-Bernard Hayet, and Reinhard Klette. Towards ubiquitous autonomous driving: The ccsad dataset. In *Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015.
 - [122] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.
 - [123] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.
 - [124] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013.
 - [125] A. O. Hero and D. Cochran. Sensor management: Past, present, and future. *IEEE Sensors Journal*, 11(12):3064–3075, Dec 2011. ISSN 1530-437X. doi: 10.1109/JSEN.2011.2167964.
 - [126] J.A. Hesch, D.G. Kottas, S.L. Bowman, and S.I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *Intl. J. of Robotics Research*, 33(1):182–201, 2014.
 - [127] Kin Leong Ho and Paul Newman. Loop closure detection in slam by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9): 740–749, 2006.
 - [128] Albert S. Huang, Matthew Antone, Edwin Olson, Luke Fletcher, David Moore, Seth Teller, and John Leonard. A high-rate, heterogeneous data set from the darpa urban challenge. *The International Journal of Robotics Research*, 29(13):1595–1601, 2010. doi:

- 10.1177/0278364910384295.
- [129] G. Huang, M. Kaess, and J. J. Leonard. Consistent sparsification for graph optimization. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 150–157, Sept 2013. doi: 10.1109/ECMR.2013.6698835.
- [130] G.P. Huang, A.I. Mourikis, and S.I. Roumeliotis. An observability-constrained sliding window filter for SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 65–72, 2011.
- [131] S. Huang, Y. Lai, U. Frese, and G. Dissanayake. How far is SLAM from a linear least squares problem? In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3011–3016, 2010.
- [132] S. Huang, H. Wang, U. Frese, and G. Dissanayake. On the number of local minima to the point feature based SLAM problem. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2074–2079, 2012.
- [133] Peter J Huber. Robust statistics. *Wiley Series in Probability and Mathematical Statistics, New York: Wiley*,—c1981, 1, 1981.
- [134] S. Viorela Ila, Lukáš Polok, Pavel Smrž, Marek Šolony, and Pavel Zemčík. Incremental block cholesky factorization for nonlinear least squares in robotics. In *In proceedings of The Robotics: Science and Systems 2013 Conference*, pages 1–8. MIT Press, 2013. ISBN 978-981-07-3937-9.
- [135] V. Ila, J.M. Porta, and J. Andrade-Cetto. Information-based compact pose slam. *Robotics, IEEE Transactions on*, 26(1):78 –93, feb. 2010. ISSN 1552-3098. doi: 10.1109/TRO.2009.2034435.
- [136] V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert. Incremental distributed inference from arbitrary poses and unknown data association: Using collaborating robots to establish a common reference. *IEEE Control Systems Magazine (CSM), Special Issue on Distributed Control and Estimation for Robotic Vehicle Networks*, 36(2):41–74, 2016.
- [137] Vadim Indelman, Luca Carlone, and Frank Dellaert. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7):849–882, 2015. doi: 10.1177/0278364914561102.
- [138] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Intl. Conf. on Computer Vision (ICCV)*, pages 626–633, 1999.
- [139] editor J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann, 1997.
- [140] H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6): 377–394, 2003.
- [141] H. Johannsson, M. Kaess, M. Fallon, and J.J. Leonard. Temporally scalable visual slam using a reduced pose graph. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 54–61. IEEE, 2013.
- [142] J. Johnson, R. Krishna, M. Stark, J. Li, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [143] V.A.M. Jorge, R. Maffei, G.S. Franco, J. Daltronzo, M. Giambastiani, M. Kolberg, and E. Prestes. Ouroboros: Using potential field in unexplored regions to close loops. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2125–2131, May 2015. doi: 10.1109/ICRA.2015.7139479.
- [144] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(12):99 – 134, 1998. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(98)00023-X.
- [145] M. Kaess. Simultaneous localization and mapping with infinite planes. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [146] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Trans. on Robotics, TRO*, 24(6):1365–1378, Dec 2008.
- [147] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012. doi: 10.1177/0278364911430419.
- [148] Matan Keidar and Gal A. Kaminka. Efficient frontier detection for robot exploration. *The International Journal of Robotics Research*, 33(2):215–236, 2014. doi: 10.1177/0278364913494911.
- [149] K. Khosoussi, S. Huang, and G. Dissanayake. Exploiting the separable structure of SLAM. In *Robotics: Science and Systems (RSS)*, 2015.
- [150] A. Kim and R. M. Eustice. Perception-driven Navigation: Active visual SLAM for Robotic Area Coverage. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3196–3203, May 2013. doi: 10.1109/ICRA.2013.6631022.
- [151] Ayoung Kim and Ryan M Eustice. Real-time visual slam for autonomous underwater hull inspection using visual saliency. *Robotics, IEEE Transactions on*, 29(3): 719–733, 2013.
- [152] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller. Multiple relative pose graphs for robust cooperative mapping. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, pages 3185–3192, Anchorage, Alaska, May 2010.
- [153] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison. Simultaneous mosaicing and tracking with an event camera. In *British Machine Vision Conf. (BMVC)*, 2014.
- [154] V.G. Kim, S. Chaudhuri, L. Guibas, and T. Funkhouser. Shape2Pose: Human-centric shape analysis. In *SIGGRAPH*, 2014.
- [155] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR ’07*, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 978-1-4244-1749-0. doi: http://dx.doi.

- org/10.1109/ISMAR.2007.4538852. URL <http://dx.doi.org/10.1109/ISMAR.2007.4538852>.
- [156] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. doi: 10.15607/RSS.2015.XI.040.
- [157] J. Knuth and P. Barooah. Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.
- [158] J. Knuth and P. Barooah. Error growth in position estimation from noisy relative pose measurements. *Robotics and Autonomous Systems*, 61(3):229–224, 2013.
- [159] K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based Maps. *The International Journal of Robotics Research*, 29(8):941–957, 2010. doi: 10.1177/0278364910370376.
- [160] Dimitrios G. Kottas, Joel A. Hesch, Sean L. Bowman, and Stergios I. Roumeliotis. On the consistency of vision-aided inertial navigation. In *Intl. Sym. on Experimental Robotics (ISER)*, 2012.
- [161] T. Krajník, J. P. Fentanes, O. M. Mozos, T. Duckett, J. Ekekrantz, and M. Hanheide. Long-term topological localisation for service robots in dynamic environments using spectral maps. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4537–4542, Sept 2014. doi: 10.1109/IROS.2014.6943205.
- [162] H. Kretzschmar, C. Stachniss, and G. Grisetti. Efficient information-theoretic graph pruning for graph-based slam with laser range finders. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 865 –871, sept. 2011. doi: 10.1109/IROS.2011.6094414.
- [163] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.
- [164] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.
- [165] A. Kundu, Y. Li, F. Dellaert, F. Li, and J.M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *Computer Vision ECCV 2014*, pages 703–718, 2014. doi: 10.1007/978-3-319-10599-4_45.
- [166] K. Lai and D. Fox. Object recognition in 3D point clouds using web data and domain adaptation. *Intl. J. of Robotics Research*, 29(8):1019–1037, 2010.
- [167] Y. Latif, C. Cadena, and J. Neira. Robust Loop Closing Over Time. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.
- [168] Yasir Latif. Dataset for robust slam backend evaluation. <https://github.com/ylatif/dataset-RobustSLAM>, Jun 2016. University of Zaragoza.
- [169] M.T. Lazaro, L.M. Paz, P. Pinies, J.A. Castellanos, and G. Grisetti. Multi-robot SLAM using condensed measurements. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1069–1076, 2011.
- [170] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *Intl. J. of Robotics Research*, 11(4):286–289, 1992.
- [171] C. Leung, S. Huang, and G. Dissanayake. Active SLAM using Model Predictive Control and Attractor Based Exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5031, October 2006. doi: 10.1109/IROS.2006.282530.
- [172] C. Leung, S. Huang, N. Kwok, and G. Dissanayake. Planning Under Uncertainty using Model Predictive Control for Information Gathering. *Robotics and Autonomous Systems (RAS)*, 54(11):898–910, November 2006. ISSN 0921-8890. doi: 10.1016/j.robot.2006.05.008.
- [173] Keith YK Leung, Yoni Halpern, Timothy D Barfoot, and Hugh HT Liu. The utias multi-robot cooperative localization and mapping dataset. *The International Journal of Robotics Research*, 30(8):969–974, 2011. doi: 10.1177/0278364911398404.
- [174] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbrück. A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE J. of Solid-State Circuits*, 43(2):566–576, 2008.
- [175] J.J. Lim, H. Pirsiavash, and A. Torralba. Parsing IKEA objects: Fine pose estimation. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2992–2999, 2013.
- [176] M. Liu, S. Huang, G. Dissanayake, and H. Wang. A convex optimization based approach for pose SLAM problems. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1898–1903, 2012.
- [177] S. Lowry, N. Snderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, Feb 2016. ISSN 1552-3098. doi: 10.1109/TRO.2015.2496823.
- [178] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
- [179] Y. Lu and D. Song. Visual navigation using heterogeneous landmarks and unsupervised geometric constraints. *IEEE Trans. Robotics*, 31(3):736–749, 2015.
- [180] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. doi: 10.15607/RSS.2015.XI.037.
- [181] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN 0521642981.
- [182] András L Majdik, Damiano Verda, Yves Albers-Schoenberg, and Davide Scaramuzza. Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles. *Journal of Field Robotics*,

- 32(7):1015–1039, 2015.
- [183] A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte. An Experiment in Integrated Exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 534–539, September 2002. doi: 10.1109/IRDS.2002.1041445.
- [184] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.
- [185] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet. A Bayesian Exploration-exploitation Approach for Optimal Online Sensing and Planning with a Visually Guided Mobile Robot. *Autonomous Robots (AR)*, 27(2):93–103, August 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9130-2.
- [186] M. Mazuran, W. Burgard, and G.D. Tipaldi. Nonlinear factor recovery for long-term slam. *The International Journal of Robotics Research*, 35(1-3):50–72, 2016. doi: 10.1177/0278364915581629.
- [187] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J.J. Leonard. Real-time 6-DOF multi-session visual SLAM over large scale environments. *Robotics and Autonomous Systems*, 61(10):1144–1158, October 2013. ISSN 0921-8890. doi: 10.1016/j.robot.2012.08.008.
- [188] J. Mendel. *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Prentice Hall, Signal Processing Serier, 1995.
- [189] P. Merrell and D. Manocha. Model synthesis: A general procedural modeling algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):715–728, 2010.
- [190] B. Micusik, J. Košeká, and G. Singh. Semantic parsing of street scenes from video. *The International Journal of Robotics Research*, 31(4):484–497, 2012. doi: 10.1177/0278364912437903. URL <http://ijr.sagepub.com/content/31/4/484.abstract>.
- [191] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649, May 2012. doi: 10.1109/ICRA.2012.6224623.
- [192] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.
- [193] A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3565–3572, April 2007.
- [194] O.M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 55(5):391–402, 2007.
- [195] E. Mueggler, G. Gallego, and D. Scaramuzza. Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems (RSS)*, 2015.
- [196] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-DOF pose tracking for high-speed maneuvres. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [197] Erich Mueller, Andrea Censi, and Emilio Frazzoli. Low-latency heading feedback control with neuromorphic vision sensors using efficient approximated incremental inference. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 992–999. IEEE, 2015.
- [198] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015.
- [199] J. Neira and J.D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robot. Automat.*, 17(6):890–897, December 2001.
- [200] José Neira, Andrew J Davison, and John J Leonard. Guest editorial special issue on visual slam. *Robotics, IEEE Transactions on*, 24(5):929–931, 2008.
- [201] E.D. Nerurkar, S.I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1402–1409, May 2009.
- [202] Peer Neubert, Niko Sunderhauf, and Peter Protzel. Appearance change prediction for long-term navigation across seasons. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 198–203. IEEE, 2013.
- [203] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: Dense tracking and mapping in real-time. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2320–2327, Barcelona, Spain, Nov 2011.
- [204] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015.
- [205] K. Ni, D. Steedly, and F. Dellaert. Tectonic SAM: Exact; out-of-core; submap-based SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Rome; Italy, April 2007. URL <http://www.cc.gatech.edu/~dellaert/pubs/Ni07icra.pdf>.
- [206] Kai Ni and Frank Dellaert. Multi-level submap based slam using nested dissection. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010. URL <http://frank.dellaert.com/pubs/Ni10iros.pdf>.
- [207] Zhenjiang Ni, Aude Bolopion, Joel Agnus, Ryad Benosman, and Stephane Regnier. Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics. *IEEE Trans. Robotics*, 28:1081–1089, 2012.
- [208] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [209] Andreas Nüchter. *3D robotic mapping: the simultaneous localization and mapping problem with six degrees*

- of freedom*, volume 52. Springer, 2009.
- [210] Philipp Oettershagen, Thomas Stastny, Thomas Mantel, Amir Melzer, Konrad Rudin, Pascal Gohl, Gabriel Agamennoni, Kostas Alexis, and Roland Siegwart. *Long-Endurance Sensing and Mapping Using a Hand-Launchable Solar-Powered UAV*, pages 441–454. Springer International Publishing, Cham, 2016. ISBN 978-3-319-27702-8. doi: 10.1007/978-3-319-27702-8_29.
- [211] Open Geospatial Consortium (OGC). Citygml 2.0.0, 2012.
- [212] Open Geospatial Consortium (OGC). OGC IndoorGML standard, 2014.
- [213] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2262–2269, May 2006.
- [214] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.
- [215] Edwin Olson, Matthew R Walter, Seth J Teller, and John J Leonard. Single-cluster spectral graph partitioning for robotics applications.
- [216] Edwin Olson, John Leonard, and Seth Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2262–2269, 2006.
- [217] Gaurav Pandey, James R McBride, and Ryan M Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011. doi: 10.1177/0278364911400640.
- [218] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz. Semantic object maps for robotic housework - representation, acquisition and use. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4644–4651, Oct 2012. doi: 10.1109/IROS.2012.6385603.
- [219] Sachin Patil, Gregory Kahn, Michael Laskey, John Schulman, Ken Goldberg, and Pieter Abbeel. *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*, chapter Scaling up Gaussian Belief Space Planning Through Covariance-Free Trajectory Optimization and Automatic Differentiation, pages 515–533. Springer International Publishing, 2015. ISBN 978-3-319-16595-0. doi: 10.1007/978-3-319-16595-0_30.
- [220] L. Paull, G. Huang, M. Seto, and J. Leonard. Communication-constrained multi-AUV cooperative SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [221] A. Pázman. *Foundations of Optimum Experimental Design*. Mathematics and its Applications. Springer, Berlin, Germany, 1986. ISBN 9789027718655.
- [222] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):730–742, Jul 1991. ISSN 0162-8828. doi: 10.1109/34.85661.
- [223] A. Pentland and J. Williams. Good vibrations: Modal dynamics for graphics and animation. *SIGGRAPH Comput. Graph.*, 23(3):207–214, July 1989. ISSN 0097-8930. doi: 10.1145/74334.74355. URL <http://doi.acm.org/10.1145/74334.74355>.
- [224] J. R. Peters, D. Borra, B. Paden, and F. Bullo. Sensor network localization on the group of 3D displacements. *SIAM Journal on Control and Optimization*, 2015.
- [225] C.J. Phillips, M. Lecce, C. Davis, and K. Daniilidis. Grasping surfaces of revolution: Simultaneous pose and shape recovery from two views. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1352–1359, 2015.
- [226] S. Pillai and J. Leonard. Monocular slam supported object recognition. In *Proceedings of Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.
- [227] G. Piovan, I. Shames, B. Fidan, F. Bullo, and B. Anderson. On frame and orientation localization for relative sensing networks. *Automatica*, 49(1):206–213, 2013.
- [228] M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2609–2616, 2014. URL <http://dx.doi.org/10.1109/ICRA.2014.6907233>.
- [229] L. Polok, V. Ila, M. Solony, P. Smrz, and P. Zemcik. Incremental block cholesky factorization for nonlinear least squares in robotics. In *Robotics: Science and Systems (RSS)*, 2013.
- [230] François Pomerleau, M. Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, December 2012.
- [231] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. An asynchronous time-based image sensor. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2130–2133. IEEE, 2008.
- [232] A. Pronobis and B. Caputo. Cold: The cosy localization database. *The International Journal of Robotics Research*, 28(5):588–594, 2009. doi: 10.1177/0278364909103912.
- [233] A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3515–3522. IEEE, 2012.
- [234] A. Pronobis, O.M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 2009.
- [235] Rawseeds. Rawseeds. <http://www.rawseeds.org/home/>, Jun 2016. Rawseeds.
- [236] A. Rényi. On Measures Of Entropy And Information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1960.
- [237] A.G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys (CSUR)*, 12(4):437–464, 1980.
- [238] L. Riazuelo, J. Civera, and JMM Montiel. C2TAM:

- A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4):401–413, 2014.
- [239] David M Rosen, Michael Kaess, and John J Leonard. Robust incremental online inference over sparse factor graphs: Beyond the gaussian case. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1025–1032. IEEE, 2013.
- [240] D.M. Rosen, C. DuHadway, and J.J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [241] D.M. Rosen, J. Mason, and J.J. Leonard. Towards lifelong feature-based mapping in semi-static environments. In *Robotics: Science and Systems (RSS), workshop: The Problem of Mobile Sensors: Setting future goals and indicators of progress for SLAM*, 2015.
- [242] M. Ruhnke, L. Bo, D. Fox, and W. Burgard. Hierarchical sparse coded surface models. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6238–6243, 2014.
- [243] Deon Sabatta, Davide Scaramuzza, and Roland Siegwart. Improved appearance-based matching in similar and dynamic environments using a vocabulary tree. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1008–1013. IEEE, 2010.
- [244] S. Saeedi, M. Trentini, M. Seto, and H. Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.
- [245] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.
- [246] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [247] J. M. Santos, T. Krajnik, J. Pulido Fentanes, and T. Duckett. Lifelong information-driven exploration to complete and refine 4d spatio-temporal maps. *IEEE Robotics and Automation Letters*, PP(99):1–1, 2016. ISSN 2377-3766. doi: 10.1109/LRA.2016.2516594.
- [248] D. Scaramuzza and F. Fraundorfer. Visual odometry: Part i the first 30 years and fundamentals. 2011.
- [249] S. Sengupta and P. Sturges. Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order mrf. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1874–1879, May 2015. doi: 10.1109/ICRA.2015.7139442.
- [250] J.J Shah and M. Mäntylä. *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. Wiley, New York, 1995.
- [251] V. Shapiro. Solid modeling. In G.E. Farin and M.-S. Kim J. Hoschek, editors, *Handbook of Computer Aided Geometric Design*. Elsevier, 2002.
- [252] C. Shen, J.F. O'Brien, and J.R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *Proceedings of ACM SIGGRAPH 2004*, pages 896–904. ACM Press, 2004. URL <http://graphics.cs.berkeley.edu/papers/Shen-IAI-2004-08/>.
- [253] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems (RSS)*, Seattle, USA, June 2009.
- [254] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Appl. Comput. Harmon. Anal.*, 30:20–36, 2010.
- [255] A. Singer and Y. Shkolnisky. Three-dimensional structure determination from common lines in Cryo-EM by eigenvectors and semidefinite programming. *SIAM J. Imaging Sciences*, pages 543–572, 2011.
- [256] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Intl. Conf. on Computer Vision (ICCV)*, 2003.
- [257] Mike Smith, Ian Baldwin, Winston Churchill, Rohan Paul, and Paul Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599, 2009. doi: 10.1177/0278364909103911.
- [258] S. Soatto. Steps Towards a Theory of Visual Information: Active Perception, Signal-to-Symbol Conversion and the Interplay Between Sensing and Control. *ArXiv e-prints*, 2011.
- [259] S. Soatto and A. Chiuso. Visual scene representations: sufficiency, minimality, invariance and deep approximation. In *ICLR Workshop, ArXiv version: 1411.7676*, 2014.
- [260] L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola. Hilbert space embeddings of hidden Markov models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-210)*, 2010.
- [261] C. Stachniss. *Robotic Mapping and Exploration*, volume 55. Springer, Berlin, Germany, 2009. ISBN 9783642010965.
- [262] C. Stachniss, G. Grisetti, and W. Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In *Proceedings of Robotics: Science and Systems Conference (RSS)*, Cambridge, MA, USA, June 2005.
- [263] H. Strasdat, A.J. Davison, J.M.M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.
- [264] H. Strasdat, J.M.M. Montiel, and A.J. Davison. Visual slam: Why filter? *Image and Vision Computing*, 30(2): 65 – 77, 2012. ISSN 0262-8856. doi: <http://dx.doi.org/10.1016/j.imavis.2012.02.009>.
- [265] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgbd slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
- [266] B. Suger, G.D. Tipaldi, L. Spinello, and W. Burgard. An Approach to Solving Large-Scale SLAM Problems with a Small Memory Footprint. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

- [267] N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2012.
- [268] N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1254–1261. IEEE, 2012.
- [269] S. Thrun. Exploration in Active Learning. In *Handbook of Brain Science and Neural Networks*, pages 381–384. MIT Press, Boston, MA, USA, 1995.
- [270] S. Thrun and K. Möller. Active Exploration in Dynamic Environments. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 531–538. Morgan-Kaufmann, 1992.
- [271] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *Intl. J. of Robotics Research*, 25(5/6):403–430, 2005.
- [272] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Boston, MA, USA, 2005. ISBN 0262201623.
- [273] Gian Diego Tipaldi and Kai O Arras. Flirt-interest regions for 2d range data. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3616–3622. IEEE, 2010.
- [274] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, May 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.70752.
- [275] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 298–372. Springer Verlag, 2000.
- [276] R. Tron, B. Afsari, and R. Vidal. Intrinsic consensus on SO(3) with almost global convergence. In *IEEE Conf. on Decision and Control*, 2012.
- [277] R. Valencia, J. Vallv(é), G. Dissanayake, and J. Andrade-Cetto. Active Pose SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1885–1891, October 2012. doi: 10.1109/IROS.2012.6385637.
- [278] Joan Vallvé and Juan Andrade-Cetto. Potential information fields for mobile robot exploration. *Robotics and Autonomous Systems*, 69:68 – 79, 2015. ISSN 0921-8890. doi: <http://dx.doi.org/10.1016/j.robot.2014.08.009>.
- [279] J. van den Berg, S. Patil, and R. Alterovitz. Motion Planning Under Uncertainty Using Iterative Local Optimization in Belief Space. *The International Journal of Robotics Research (IJRR)*, 31(11):1263–1278, September 2012. doi: 10.1177/0278364912456319.
- [280] V. Vineet, O. Miksik, M. Lidegaard, M. Niessner, S. Golodetz, V.A. Prisacariu, O. Kahler, D.W. Murray, S. Izadi, P. Peerez, and P.H.S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 75–82, May 2015. doi: 10.1109/ICRA.2015.7138983.
- [281] Niklas Wahlstrm, Thomas B Schn, and Marc Peter Deisenroth. Learning deep dynamical models from image pixels. *IFAC-PapersOnLine*, 48(28):1059 – 1064, 2015. ISSN 2405-8963. doi: <http://dx.doi.org/10.1016/j.ifacol.2015.12.271>. 17th {IFAC} Symposium on System Identification {SYSID} 2015Beijing, China, 1921 October 2015.
- [282] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1871–1878, Oct 2012. doi: 10.1109/IROS.2012.6385561.
- [283] C.C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *Intl. J. of Robotics Research*, 26(9):889–916, 2007.
- [284] H. Wang, G. Hu, S. Huang, and G. Dissanayake. On the structure of nonlinearities in pose graph SLAM. In *Robotics: Science and Systems (RSS)*, 2012.
- [285] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 30, 2013.
- [286] Z. Wang, S. Huang, and G. Dissanayake. *Simultaneous Localization and Mapping: Exactly Sparse Information Filters*. World Scientific, 2011.
- [287] Michael Warren, D. McKinnon, H. He, and Ben Upcroft. Unaided stereo vision based pose estimation. In Gordon Wyeth and Ben Upcroft, editors, *Australasian Conference on Robotics and Automation*, Brisbane, 2010. Australian Robotics and Automation Association. URL <http://eprints.qut.edu.au/39881/>.
- [288] Michael Warren, David McKinnon, Hu He, Arren Glover, Michael Shiel, and Ben Upcroft. Large Scale Monocular Vision-only Mapping from a Fixed-Wing suAS. In *International Conference on Field and Service Robotics*, Matsushima, Japan, 2012.
- [289] O. Wasenmüller, M. Meyer, and D. Stricker. Corbs: Comprehensive rgb-d benchmark for slam using kinect v2. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–7, March 2016. doi: 10.1109/WACV.2016.7477636.
- [290] D. Weikersdorfer and J. Conradt. Event-based particle filtering for robot self-localization. In *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2012.
- [291] D. Weikersdorfer, R. Hoffmann, and J. Conradt. Simultaneous localization and mapping for event-based vision systems. In *Int. Conf. on Computer Vision Systems (ICVS)*, 2013.
- [292] David Weikersdorfer, David B Adrian, Daniel Cremers, and Jorg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 359–364. IEEE, 2014.
- [293] T. Whelan, J. B. McDonald, M. Kaess, M. F. Fallon,

- H. Johannsson, and J. J. Leonard. Kintinuous: Spatially extended Kinect-Fusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, July 2012.
- [294] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.
- [295] S. Williams, V. Indelman, M. Kaess, R. Roberts, J.J. Leonard, and F. Dellaert. Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing. *The International Journal of Robotics Research*, 33(12):1544–1568, 2014. doi: 10.1177/0278364914531056.
- [296] Wera Winterhalter, Freya Fleckenstein, Bastian Steder, Luciano Spinello, and Wolfram Burgard. Accurate indoor localization for rgb-d smartphones and tablets given 2d floor plans. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3138–3143. IEEE, 2015.
- [297] Ryan W. Wolcott and Ryan M. Eustice. Visual localization within LIDAR maps for automated urban driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, Chicago, IL, USA, September 2014.
- [298] R. Wood. RoboBees project. Harvard University, 2015. URL <http://robobees.seas.harvard.edu/>.
- [299] C. Wu, I. Lenz, and A. Saxena. Hierarchical semantic labeling for task-relevant rgb-d perception. In *Robotics: Science and Systems (RSS)*, 2014.
- [300] B. Yamauchi. A Frontier-based Approach for Autonomous Exploration. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 146–151. IEEE, July 1997. doi: 10.1109/CIRA.1997.613851.
- [301] Shao-Wen Yang, Chieh-Chih Wang, and Charles Thorpe. The annotated laser data set for navigation in urban areas. *The International Journal of Robotics Research*, 30(9):1095–1099, 2011. doi: 10.1177/0278364910389840.
- [302] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *Intl. Conf. on Computer Vision (ICCV)*, 2007.
- [303] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. Metaxas, and X. Zhou. Towards robust and effective shape modeling: Sparse shape composition. *Medical Image Analysis*, 16(1):265–277, 2012.
- [304] L. Zhao, S. Huang, and G. Dissanayake. Linear SLAM: A linear solution to the feature-based and pose graph SLAM based on submap joining. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.