Perception System Design for Low-Cost Commercial Ground Robots: Sensor Configurations, Calibration, Localization and Mapping

Yiming Chen, Mingming Zhang, Dongsheng Hong, Chengcheng Deng, and Mingyang Li

***Abstract*— For commercially successful ground robots, high degree of autonomy, low manufacturing and maintenance cost, as well as minimized deployment limitations in different environments are essential attributes. To deliver an 'anywhere deployable' product, it is impractical to rely on one single sensor or one single piece of algorithm to overcome all related challenges. Instead, the entire robotic system should be dedicated designed, including the choices of sensors, processors, algorithm integration for various functionality, and so on.**

This paper presents our design of perception system for commercial ground robots, which is able to operate in most common environments. The designed system is equipped with low-cost sensors and processors. The first key contribution of this paper is the design of the robotic sensory system, which includes a monocular camera, a 2D laser range finder (LRF), wheel encoders, and an inertial measurement unit (IMU). Our sensory system can be built at a cost of as low as *$100*. Furthermore, the selected sensors provide complementary characteristics for perception of both robot ego-motion and its surrounding environments, which are the prerequisites for 'anywhere' deployment. The second key contribution of this paper is that a complete set of technologies is proposed based on our sensor systems, including sensor calibration (factory calibration and online calibration), localization (environmental exploring and re-localization), as well as mapping. The proposed methodology includes both efficient engineering implementation and theoretical novelty for high performance systems. Experimental results from our robotic testing platform and off-the-shelf commercial robots are presented. These results demonstrate that the proposed system can be deployed in various environmental conditions without performance compromise.

(a) Hotel service robot

(b) Hospital assistant robot

(c) Restaurant waiter robot

(d) Campus logistic robot.

Fig. 1: Representative commercial robots based on the proposed low-cost perception system design. (a) a hotel service robot, (b) a hospital assistant robot, (c) a restaurant waiter robot, and finally (d) a campus logistic robot for delivering packages. Note that, the first three types of robots are already commercially released and available on the market, while the last one is a robotic testing platform.

I. Introduction

The robotics community has been under fast development in recent 30 years, in both academia and industry. While plenty of research progress has been made, the real-world robotic applications are still quite limited. Most deployed robots are either in special, highly controlled environment (e.g., robotic arms in factory assembly lines, or automatic guided vehicles in large-scale warehouses) or single-functional miniature ones (e.g, robotic vacuum cleaners, or smart drones). Other types of robotic systems are either still under academical research or only being operated in a small number of 'trial' locations, including self-driving vehicles, service robots (used in hotels, shopping centers, etc.), package delivering robots, emergency rescuing robots, and so on. There are two main factors that slow down wide deployment of robotic systems: manufacturing costs and technology maturity. On one hand, today the cost of building a robot is still high, which limits its affordability. On the other hand, the technology itself is not mature enough, to allow robots to autonomously operate in various environments without human intervention.

In this paper, we tackle this problem by proposing a *product-ready* robotic perception system design, including both sensor selection and configuration, as well as algorithms and implementations, for robust and long-term autonomy. In particular, our low-cost system exploits a monocular camera (stereo setup will be helpful but not mandatory), a short-range 2D laser range finder, wheel encoders, and an inertial measurement unit. In contrast to the 'simplest' design perspective, which is to enable autonomous robots with minimum number of sensors [1][2][3], our sensory setup is relatively complicated and not highly theoretically challenging. However, from redundant sensing and practical deployment perspective, the four sensors in our setup offer complementary characteristics which enable robust robotic operation with extremely rare theoretical degenerate cases and low failure rates. The detailed discussion of sensor setup can be found in Section III, and we also emphasize that our system is of low cost, in terms of both manufacture

The authors are with Alibaba A.I. Labs, Hangzhou, China. {yimingchen, mingmingzhang, hds.hds, chengsheng.dcc, mingyangli}@alibaba-inc.com.

and computation. In fact, the full sensor setup of our design can be built at just around $300 (even as low as $100 for mass production), and our proposed online algorithms only requires one core of a modern ARM processor.

Another key contribution of this paper is the detailed algorithm design and efficient implementation that enables mobile robots to move autonomously for 'anywhere deployment'. In particular, we propose

- A single-step in-factory batch calibration algorithm, which is able to calibrate sensors' intrinsic and extrinsic parameters efficiently and accurately;
- A localization algorithm via tightly-coupled sensor fusion, for performing area exploration, re-localization, and sensor online calibration;
- A mapping algorithm, to generate accurate hybrid maps for enabling long-term deployment for commercial robots.

It is also important to point out that, instead of focusing on 'mathematical novelty' in algorithm design, this paper works on 'system design' to drive low-cost commercial robots into reality.

II. Related Work

Related work can be grouped into two categories based on robotic sensor system and algorithm design: the ones that rely on minimum number of sensors (one or two) [1][2][3][4][5] and the ones use multiple sensors similarly to ours with their own limitations [6][7][8][9].

A. Systems with minimum number (1-2) of sensors

Among various sensors used in robotic systems, camera (RGB camera or/and depth camera) and laser range finders (single-beam or multi-beam) are the most popular ones, and a variety of autonomy algorithms are designed by using them as the only or major sensors. When used alone, each of these sensors can support full autonomy of robots in proper environments. For example, camera provides dense perception of surrounding environment at typically 3-30Hz, whose sensory information can be used for conducting 3D SLAM [2][10]. Camera's advantages in size, cost, and power lend itself easy integration into different types of robots, even miniature ones [2]. Depth cameras can also be used as the main sensor of a robot, providing localization, mapping, and object avoidance capabilities [5][11]. Compared to depth cameras, LRF typically has larger field of views (FOV) and longer sensing distance, and thus also widely used for building commercial robots [1][4].

Moreover, due to the recent development of MEMS technology, cameras are often integrated with IMUs. This allows a robot to have better scale estimation capability and also better system stability under aggressive motion or in challenging environments [3][12][13]. Similarly, LRF sensors can also be aided by IMUs, for attaining improved performance [14]. Researchers also work on integrating IMU with wheel encoders to generate accurate dead-reckoning estimates [15], while the lack of the exteroceptive sensors makes long-term pose drift inevitable.

B. Systems with redundant multi-sensor system

Nowadays one of the most challenging robotics research project is autonomous driving. Typically, autonomous driving cars are equipped with accurate but expensive GNSS-INS system, tens of radar sensors, multiple RGB cameras, and multiple multi-beam LRFs [6]. To make similar level autonomy more affordable and feasible to smaller robots, systems that rely on cameras and 3D-sensing LRFs are developed, in which the expensive GNSS-INS system can be removed [7][16]. In these systems, the 3D-sensing LRFs are either built by multi-beams LRFs which are also of high cost, or by spinning 2D LRFs which also have their own limitations for commercial robot deployment (e.g., price and life-cycle of spinning rotor, challenges in robotic industrial and structural design, and so on). To develop ground robots, wheel encoders are low-cost but efficient sensors. However, most systems that utilize wheel encoders focus only on environments with planar surfaces [9][8]. Infrared and ultrasonic sensors are also widely used in robotic systems. Although they can be used for localization and mapping, due to the low precision and resolution, they are mainly used for obstacle avoidance [17].

This paper presents our robotic perception system design with a camera, a single-beam LRF, an IMU, and wheel encoders. Such a system is of low cost and can guarantee robust performance in most commonly seen environments ('anywhere deployment'). These are the objectives that can *not* be simultaneously achieved by *any* of the above mentioned systems.

III. Sensor Configuration and Discussion

Robotic sensors can be grouped into two types: proprioceptive sensors (measuring robot's own motion) and exteroceptive sensors (perceiving the surrounding environments). To exploit the complimentary sensory capabilities, a robot is typically equipped with both proprioceptive and exteroceptive sensors.

1) Proprioceptive Sensors: The proposed design integrates both an IMU and wheel encoders, due to their complementary properties. An IMU measures angular velocity and specific force (gravity affected local linear acceleration) of a moving frame at high frame-rates ($\geq$100 Hz), and its measurements can be used to characterize robot motion in a 3D space. Although IMU is widely used in robotic applications, there are limitations due to the nature of the sensor, even fused with other sensors. There are a couple of degenerate cases that can result in motion estimation failure. For example, the robot is stationary, traveling in constant circular or linear velocity, and so on [3][18]. Additionally, since an IMU can *not* obtain linear velocity estimates directly, when a robot is navigating in challenging environments, the localization estimates from an iterative estimator might fall into local minimum values consistently, and thus lead to low estimation performance or even divergence [19]. However, these challenges can all be overcome by integrating wheel encoders, which provide velocity estimate directly. On the other hand, wheel encoders can only characterize motion of a robot on a 2D plane. These complementary properties

make the IMU and wheel encoders a perfect proprioceptive sensor pair and furthermore they are both of low cost. Section VII-A and Table I show that, when an IMU and wheel encoders fused together, the motion estimation is improved significantly.

TABLE I: Proprioceptive sensor performance evaluation: Localization errors of using an IMU only, wheel encoders (WE) only, and two sensors integrated (IMU+WE).

	IMU	WE	WE+IMU
INDOOR SCENE I (2D)			
position err. (m)	7.8e+4	7.498	3.149
rotation err. (deg.)	7.2e+3	123.384	50.409
INDOOR SCENE II (2D)			
position err. (m)	1.2e+3	17.922	0.716
rotation err. (deg.)	3.0e+3	212.227	0.684
OUTDOOR SCENE I (3D)			
position err. (m)	3.5e+5	138.491	124.345
rotation err. (deg.)	6.3e+3	132.094	83.708
OUTDOOR SCENE II (3D)			
position err. (m)	2.7e+5	215.198	100.776
rotation err. (deg.)	1.9e+3	195.101	67.431

2) Exteroceptive Sensors: A short-range low-cost 2D LRF and a monocular camera are integrated by design.

Vision based robotic systems can work successfully when their captured images contain enough static, distinguishable information (e.g., sparse feature points [12], semi-dense point clouds [10], CNN features [20], and so on). While this is true for most cases, there are still many environments not conforming this assumption, especially inside of a building and across buildings (due to dark environments, feature-less scenes, a large number of moving object, short-time and long-term light condition changes, etc.).

On the other hand, a LRF has large FOV ($200-360$ degree) and is robust to light conditions and feature richness in environment. The weakness of low-cost LRF is the short sensing range (making it incapable for outdoor navigation), inability to capture 3D information, and large noises. The first two problems can be well compensated by cameras, and the third one can be improved by probabilistic estimation based sensor fusion. Fig. 2 shows representative real scenarios which require both camera and LRF equipped.

(a) (b)

Fig. 2: Representative scenarios when perception system will fail by only using a LRF (a) or a camera (b). (a) In front of building on campus when all structures are far away, and (b) feature-less room in a building.

3) Complete Design: It is well known that proprioceptive and exteroceptive sensors provide complementary information, and thus the integration of both enhances robotic perception system performance. As a result, we design our robots with sensors mentioned above together, i.e. camera, LRF, IMU, wheel encoders. Another crucial factor that needs to be well considered for building a robotic perception system is sensor-to-sensor rigid connection and accurate time synchronization. Although it is possible to design algorithms to estimate time-varying spatial-temporal calibration parameters online [12], it may impair the system performance and introduce the deployment risks. Thus, we put this as a strong requirement to manufacturers.

IV. Notation

In this paper, we use $\mathcal{C}$, $\mathcal{L}$, $\mathcal{I}$, $\mathcal{O}$ to represent reference frames of camera, LRF, IMU, and wheel encoders (WE) respectively. The origin of frame $\mathbf{B}$ expressed in frame $\mathbf{A}$ is denoted as ${}^{\mathbf{A}}\mathbf{p}_{\mathbf{B}} \in \mathbb{R}^3$. The rotation matrix from frame $\mathbf{B}$ to frame $\mathbf{A}$ is denoted as ${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{R} \in \mathbf{SO}(3)$, and ${}^{\mathbf{A}}_{\mathbf{B}}\bar{\mathbf{q}}$ is the corresponding unit quaternion vector. We denote ${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{t}$, the complete $\mathbf{SE}(3)$ transformation between the two frames, with ${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{t} = \left[{}^{\mathbf{A}}\mathbf{p}_{\mathbf{B}}^{\intercal}, {}^{\mathbf{A}}_{\mathbf{B}}\bar{\mathbf{q}}^{\intercal}\right]^{\intercal}$. Additionally, the homogeneous transformation matrix ${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{T}$ is defined as

$$ {}^{\mathbf{A}}_{\mathbf{B}}\mathbf{T} = \begin{bmatrix} {}^{\mathbf{A}}_{\mathbf{B}}\mathbf{R} & {}^{\mathbf{A}}\mathbf{p}_{\mathbf{B}} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^4 \tag{1} $$

whose multiplication operation with a translation vector $\mathbf{p} \in \mathbb{R}^3$ is ${}^{\mathbf{A}}_{\mathbf{B}}\mathbf{T} \cdot \mathbf{p} = {}^{\mathbf{A}}_{\mathbf{B}}\mathbf{R}\,\mathbf{p} + {}^{\mathbf{A}}\mathbf{p}_{\mathbf{B}}$. It is also assumed that a robot moves with respect to a global reference frame $\mathbf{G}$, in which gravitational acceleration ${}^{\mathbf{G}}\mathbf{g} \in \mathbb{R}^3$ is well known. the linear velocity, linear acceleration and rotational velocity of frame $\mathbf{B}$ with respect to $\mathbf{A}$, expressed in $\mathbf{A}$ are denoted as ${}^{\mathbf{A}}\mathbf{v}_{\mathbf{B}}$, ${}^{\mathbf{A}}\mathbf{a}_{\mathbf{B}}$, and ${}^{\mathbf{A}}\boldsymbol{\omega}_{\mathbf{B}} \in \mathbb{R}^3$, respectively. Finally, $\mathbf{a}^{\intercal}$ to represent transpose of the variable $\mathbf{a}$.

V. Sensor System Calibration

In this section, we propose our approach for in-factory sensor calibration. The existing calibration approaches are typically performed within pairs of sensors, e.g., camera to LRF [21], IMU to camera [12], camera to WE [22], etc. A uniform, tightly-coupled algorithm is proposed for fast and accurate in-factory calibration. Implementation details and observability analysis are presented with respect to the calibration environment shown in Fig. 3.

A. Problem Definition

Suppose $\boldsymbol{\theta} = [\boldsymbol{\xi}^{\intercal}, \boldsymbol{\psi}^{\intercal}]^{\intercal}$ is the sensor parameter to calibrate. In particular, $\boldsymbol{\xi}$ and $\boldsymbol{\psi}$ represent sensor intrinsic and extrinsic parameters

$$ \boldsymbol{\xi} = [\boldsymbol{\xi}_{\mathbf{O}}^{\intercal}, \boldsymbol{\xi}_{\mathbf{I}}^{\intercal}, \boldsymbol{\xi}_{\mathbf{C}}^{\intercal}]^{\intercal}, \quad \boldsymbol{\psi} = \left[{}^{\mathbf{O}}_{\mathbf{L}}\mathbf{t}^{\intercal}, {}^{\mathbf{O}}_{\mathbf{C}}\mathbf{t}^{\intercal}, {}^{\mathbf{O}}_{\mathbf{I}}\mathbf{t}^{\intercal}\right]^{\intercal} \tag{2} $$

where $\boldsymbol{\xi}_{\mathbf{O}}$, $\boldsymbol{\xi}_{\mathbf{I}}$, $\boldsymbol{\xi}_{\mathbf{C}}$ denote intrinsics of WE, IMU, and camera respectively[1]. Note that our algorithm is generic, and can be applied to different sensor models, e.g., camera

[1]Typically, intrinsics of LRFs are well calibrated by manufacturers, and we here ignore the corresponding terms.

Fig. 3: Calibration 'Pyramid' for in-factory calibration, with April tags [23] on each surface to provide visual constraints and metric scale. Our design makes it easy for a robot to observe multiple surfaces simultaneously, to increase the total number of measurements. Note that each surface is not perpendicular to ground planes such that the roll and pitch of LRF is identifiable. This calibration pyramid is hollow so portable and of low cost.

models [24][25], wheel models [26][27], etc. The calibration problem is formulated as

$$\arg\max_{\boldsymbol{\xi},\boldsymbol{\psi},\boldsymbol{\gamma}} \mathcal{P}\left(\boldsymbol{\xi},\boldsymbol{\psi},\boldsymbol{\gamma}|\mathbf{z}_m\right) \tag{3}$$

where $\mathbf{z}_m$ and $\boldsymbol{\gamma}$ represent the sensor measurements and other latent variables of interests, e.g., poses and structural information.

In our system, the measurement $\mathbf{z}_m$ contains:

$$\mathbf{z}_m := \{\mathbf{c}_m, \mathbf{l}_m, \mathbf{o}_m, \boldsymbol{\omega}_m, \mathbf{a}_m\} \tag{4}$$

where $\mathbf{c}_m, \mathbf{l}_m, \mathbf{o}_m, \boldsymbol{\omega}_m, \mathbf{a}_m$ denote camera, LRF, WE, gyroscope and accelerometer measurements respectively.

During calibration, the robot moves (see Section V-C for details) around the calibration pyramid, and sensor measurements are collected. By defining 'keyframe' as the frame when images are available, $\boldsymbol{\gamma}$ is modeled as

$$\boldsymbol{\gamma} = \left[{}^{\mathbf{G}}_{\mathbf{A}_1}\mathbf{t}^\intercal, {}^{\mathbf{A}_1}_{\mathbf{A}_2}\mathbf{t}^\intercal, \cdots, {}^{\mathbf{A}_{K-1}}_{\mathbf{A}_K}\mathbf{t}^\intercal, {}^{\mathbf{G}}_{\mathbf{C}_1}\mathbf{t}^\intercal, \cdots, {}^{\mathbf{G}}_{\mathbf{C}_M}\mathbf{t}^\intercal\right]^\intercal \tag{5}$$

where M and K denote the number of keyframes and number of calibration surfaces, and $\mathbf{A}_i$ represents the i-th surface on calibration pyramid.

B. Optimization Formulation for Calibration

The proposed calibration method is optimization based and minimize costs formulated from sensor measurements. In particular, the camera cost function h_c is defined as:

$$\mathbf{h}_c^{i,j} = \mathbf{z}_m^{i,j} - \boldsymbol{\pi}\left(\mathbf{u}_{i,j}, \boldsymbol{\xi}_{\mathbf{C}}\right), \text{with}\, \mathbf{u}_{i,j} = \begin{bmatrix} \frac{{}^{\mathbf{C}_i}x_{f_j}}{{}^{\mathbf{C}_i}z_{f_j}} & \frac{{}^{\mathbf{C}_i}y_{f_j}}{{}^{\mathbf{C}_i}z_{f_j}} \end{bmatrix}^\intercal$$

where function $\pi(\cdot)$ is the camera observation function, which projects a 3D target point onto homogeneous image plane and then apply camera intrinsic model with parameters $\boldsymbol{\xi}_{\mathbf{C}}$. Note that the target board index k is omitted for simpler presentation. To compute $\mathbf{u}_{i,j}$, ${}^{\mathbf{C}_i}\mathbf{p}_{f_j}$ is given by

$${}^{\mathbf{C}_i}\mathbf{p}_{f_j} = \begin{bmatrix} {}^{\mathbf{C}_i}x_{f_j} \\ {}^{\mathbf{C}_i}y_{f_j} \\ {}^{\mathbf{C}_i}z_{f_j} \end{bmatrix} = {}^{\mathbf{C}}_{\mathbf{O}}\mathbf{T}{}^{\mathbf{O}_i}_{\mathbf{G}}\mathbf{T}\,{}^{\mathbf{G}}\mathbf{p}_{f_j}, {}^{\mathbf{G}}\mathbf{p}_{f_j} = {}^{\mathbf{G}}_{\mathbf{A}}\mathbf{T}\,{}^{\mathbf{A}}\mathbf{p}_{f_j} \tag{6}$$

where ${}^{\mathbf{A}}\mathbf{p}_{f_j}$ is known based on target pattern detection [23].

On the other hand, the LRF cost function is modeled by

$$\mathbf{h}_l^{i,j} = \mathbf{e}_3^\intercal {}^{\mathbf{G}}_{\mathbf{A}}\mathbf{T}^{-1}{}^{\mathbf{G}}_{\mathbf{O}_i}\mathbf{T}\,{}^{\mathbf{O}}_{\mathbf{L}}\mathbf{T}\,{}^{\mathbf{L}}\mathbf{p}_{f_j}, {}^{\mathbf{L}}\mathbf{p}_{f_j} = \begin{bmatrix} r_j\cos(\alpha_j) \\ r_j\sin(\alpha_j) \\ 0 \end{bmatrix} \tag{7}$$

where r_j and α_j denote a LRF range measurement and the corresponding bearing angle, respectively. Minimizing the cost derived from $h_l^{i,j}$ is to align the corresponding LRF scanning planes (inferred implicitly from the LRF measurement) with the calibration pyramid surfaces. The IMU and WE pose propagation functions $g_{\mathrm{i}}^{i,j}$ and $g_{\mathrm{o}}^{i,j}$ are formulated similarly to that of [3] and [8], respectively.

With $\boldsymbol{\xi}, \boldsymbol{\psi}, \boldsymbol{\theta}, \boldsymbol{\gamma}$ and cost functions defined above, the log-likelihood of eqn. (3) can be computed and turned into the following cost

$$\begin{aligned} c(\boldsymbol{\xi},\boldsymbol{\psi},\boldsymbol{\gamma}) &= c_c + c_l + c_i + c_o \\ &= \sum_{i=1}^{M}\left(\sum_{k=1}^{K}\left(\sum_{j=1}^{N_c^{i,k}}||\mathbf{h}_{\mathrm{c}}^{i,j,k}||^2_{\mathbf{Q}_c} + \sum_{j=1}^{N_l^{i,k}}||\mathbf{h}_{\mathrm{l}}^{i,j,k}||^2_{\mathbf{Q}_l}\right)\right) \\ &+ \sum_{i=1}^{M-1}\left(||{}^{\mathbf{G}}_{\mathbf{C}_k}\mathbf{t} - \sum_{j=1}^{N_i^i}\mathbf{g}_{\mathrm{i}}^{i,j}||^2_{\mathbf{Q}_i} + ||{}^{\mathbf{G}}_{\mathbf{C}_k}\mathbf{t} - \sum_{j=1}^{N_o^i}\mathbf{g}_{\mathrm{o}}^{i,j}||^2_{\mathbf{Q}_o}\right) \end{aligned} \tag{8}$$

where c_c, c_l, c_i and c_o are costs for each sensor, $N_c^{i,k}$ and $N_l^{i,k}$ are the number of camera measurements and the number of LRF points, corresponding to keyframe i and calibration target k, N_i^i and N_o^i are the numbers of IMU and WE measurements between frames i and $i+1$, h_c and h_l are camera and LRF cost functions, $\mathbf{g}_i$ and $\mathbf{g}_o$ are pose prediction functions using IMU and WE measurements respectively, $\mathbf{Q}_c, \mathbf{Q}_l, \mathbf{Q}_i$ and $\mathbf{Q}_o$ are the corresponding linearized noise covariance matrices. Note that $\|\mathbf{a}\|^2_{\mathbf{Q}} := \mathbf{a}^\intercal\mathbf{Q}^{-1}\mathbf{a}$.

C. Identifiability, Initialization, and Optimization

To ensure the calibration process to work correctly, the related theoretical issues and implementation details should also be well considered, including identifiability, initialization, and numerical optimization.

1) Identifiability: In [21][28] it is shown that camera to LRF extrinsics are identifiable, using calibration setup similar to ours. For WE extrinsics, if a robot only moves on 2D planar surfaces, the projection of ${}^{\mathbf{O}}\mathbf{p}_{\mathbf{C}}$ along the vertical direction is unobservable [29]. Additionally, under planar motion, the projection of ${}^{\mathbf{I}}\mathbf{p}_{\mathbf{C}}$ along the vertical direction is also unobservable [12]. However, if a robot moves on multiple planes during calibration process, the unobservability problem will disappear [29][12]. On the other hand, for the latent variable $\boldsymbol{\gamma}$, there are also four elements unobservable, corresponding to global position and yaw [12].

2) Initialization: Since we focus on robot 'design', instead of activating a 'black-box' , initial estimates of sensor calibration parameters, $\boldsymbol{\xi}$ and $\boldsymbol{\psi}$, can be given in advance[2]. On the other hand, to initialize $\boldsymbol{\gamma}$, we first apply PnP [30]

[2]Camera intrinsics are provided by camera manufacturer in our project. It can also be done from numerical initialization using target observations.

algorithm for each image with respect to corresponding target boards, to obtain ${}^{\mathbf{A}_i}_{\mathbf{C}_\ell}\mathbf{t}$. Based on the observation of multiple targets in a set of images, target to target transformation can be initialized with ${}^{\mathbf{A}_i}_{\mathbf{A}_j}\mathbf{T} = {}^{\mathbf{A}_i}_{\mathbf{C}_\ell}\mathbf{T}\,{}^{\mathbf{C}_\ell}_{\mathbf{A}_j}\mathbf{T}$. Additionally, based on the observability of global frames, ${}^{\mathbf{G}}_{\mathbf{C}_1}\mathbf{T}$ is initialized by computing roll and pitch angles from IMU measurements and IMU to WE extrinsics, and setting yaw and global positions to be zero. Subsequently, other variables in $\boldsymbol{\gamma}$ can be fully initialized.

3) Optimization: To solve eqn. (8), nonlinear iterative optimization approach (e.g., Levenberg-Marquardt) is used with initialization method described in Section V-C.2. Our system can be deployed to two types of robot: 1) robot that is only designed to move in 2D planar scenes (e.g, Fig. 1(c)), and 2) robot that can move in complicated 3D environment (e.g., Fig. 1(d)). Limited by structural design, 2D robot might not be able to move up or down slopes, even during calibration. So we choose to *fix* unobservable parameters (see Sec. V-C.1) in the calibration process. This 'unobservable-parameter-fix' method will not lead to compromised performance, since these parameters are also not of importance in the real-time 2D deployment (see Sec VII-B and Table. III for validation). If a robot can move in 3D, we drive the robot on multiple planar slopes during calibration to ensure full calibration.

VI. Localization and Mapping

A. Localization Formulation

Localization algorithms are typically formulated under either filter-based approaches [8][12] or optimization based ones [4][13][2]. Comparison of two families of algorithms are widely discussed recently [31], and it is not fair to draw an easy conclusion that one outperforms the other completely. The decision depends on use cases, software implementation and tuning, and so on. In our system, the calibration and mapping algorithms (see Sec. V and VI-E) rely on optimization framework, which is more suitable for processing data in batch, especially when computation resource is not the bottleneck. Thus, it is natural for us to select optimization based localization framework, such that a large part of the cost functions used in calibration and mapping can be re-used in localization. By this way, the calibration, localization and mapping are within an unified framework, which saves algorithm /software development and maintenance effort.

In this section a localization map is assumed available. Cost functions can be formulated based on both spatial information recorded in the map and constraints *not* associated to the map (a.k.a. 'local' in literature). With these two types of costs being defined, the proposed algorithm can operate freely both in either mapped area or environment under exploration (by ignoring the first type of costs).

Similar to recent state-of-the-art localization approaches [4][9][14][8][32], a sliding-window estimator is introduced for localization, whose state vector at time-stamp k is

$$\mathbf{x}_k = \begin{bmatrix} \boldsymbol{\xi}_{I_k}^\intercal & \boldsymbol{\psi}^\intercal & \boldsymbol{\beta}_k^\intercal & {}^{\mathbf{G}}_{\mathbf{C}_{k-s+1}}\mathbf{t}^\intercal & \cdots & {}^{\mathbf{G}}_{\mathbf{C}_k}\mathbf{t}^\intercal & {}^{\mathbf{G}}\mathbf{v}_{\mathbf{I}_k}^\intercal \end{bmatrix}^\intercal \tag{9}$$

where $\boldsymbol{\xi}_{I_k}$ and $\boldsymbol{\psi}$ are IMU intrinsics (i.e., biases) and sensor extrinsics (see eqn. (2)) to estimate online, s represents the size of sliding window, and ${}^{\mathbf{G}}\mathbf{v}_{\mathbf{I}_k}$ is the velocity of IMU corresponding to the latest frame. Moreover, as this paper is dedicated to ground robot, extra motion constraints can be enforced for performance improvement. Specifically, the local ground surfaces are approximated by a differentiable two-dimensional manifolds, $\boldsymbol{\beta}_k$ represents the corresponding parameters. The detailed manifold analysis and formulation can be found in our previous work [32].

B. Statistical Optimization

To illustrate the proposed algorithm, we explain the details at timestamp $k+1$ given the results at k. Similar to other sliding-window based optimization methods [13][32], at timestamp k our system has an estimate of state $\mathbf{x}_k$ and prior $\mathbf{x}_p$, as well as the corresponding prior information matrix $\mathbf{A}_p$ and vector $\mathbf{b}_p$.

When new sensor measurements arrive, we rely on IMU and WE measurements to propagate the latest pose [32], and once enough spatial displacement reached we place a new keyframe by augmenting the state vector

$$\mathbf{x}_{k+1}^{aug} = \begin{bmatrix} \mathbf{x}_k^\intercal & \boldsymbol{\xi}_{I_{k+1}}^\intercal & \boldsymbol{\beta}_{k+1}^\intercal & {}^{\mathbf{G}}_{\mathbf{C}_{k+1}}\mathbf{t}^\intercal & {}^{\mathbf{G}}\mathbf{v}_{\mathbf{I}_{k+1}}^\intercal & \mathbf{f}^\intercal \end{bmatrix}^\intercal \tag{10}$$

where $\boldsymbol{\xi}_{I_{k+1}}$ and $\boldsymbol{\beta}_{k+1}$ are latest IMU intrinsics and local manifold parameters, ${}^{\mathbf{G}}_{\mathbf{C}_{k+1}}\mathbf{t}$ and ${}^{\mathbf{G}}\mathbf{v}_{\mathbf{I}_{k+1}}$ are computed by pose prediction, $\mathbf{f}$ represents 3D point features observed by keyframes in $\mathbf{x}_{k+1}^{aug}$, whose corresponding measurements have not been processed yet [12]. Note that $\boldsymbol{\beta}_{k+1}$ conform random walk processes. In this paper, traditional sparse feature extraction and matching methods are applied to process images, followed by RANSAC algorithms for outlier rejection [13][32]. The augmented state, $\mathbf{x}_{k+1}^{aug}$ in eqn. (10) can also be re-written as

$$\mathbf{x}_{k+1}^{aug} = \begin{bmatrix} \mathbf{x}_{k+1}^\intercal & \mathbf{x}_m^\intercal \end{bmatrix}^\intercal, \mathbf{x}_m = \begin{bmatrix} \boldsymbol{\xi}_{I_k}^\intercal & \boldsymbol{\beta}_k^\intercal & {}^{\mathbf{G}}_{\mathbf{C}_{k-s}}\mathbf{t}^\intercal & {}^{\mathbf{G}}\mathbf{v}_{\mathbf{I}_k}^\intercal & \mathbf{f}^\intercal \end{bmatrix}^\intercal. \tag{11}$$

Therefore, at timestamp $k+1$, to perform pose optimization, non-linear iterative optimization method first optimizes $\mathbf{x}_{k+1}^{aug}$ based on the sensor measurements and prior terms. Subsequently, we marginalize $\mathbf{x}_m$ and update the prior terms (estimates, information matrix and vector) to keep the computational cost bounded. Mathematical details if marginalization can be found in [13] to see the mathematical details.

C. Cost Function

Now, the remaining problem is the cost function design for estimating $\mathbf{x}_{k+1}^{aug}$. By modifying calibration optimization (see. eqn. 8), the localization cost function is defined as follows

$$c(\mathbf{x}_{k+1}^{aug}) = c_i + c_o + c_c^\dagger + c_c^\ddagger + c_l^\dagger + c_m + c_s \tag{12}$$

where c_i and c_o are relative pose constraints derived from IMU and WE measurements (identical to that of eqn. 8). $c_c^\dagger$ is the camera cost corresponding to 3D features that are modeled in map and detected by loop closure detection (see

Section VI-D), $c_c^{\ddagger}$ are the ones with respect to *unknown* features $\mathbf{f}$, in current environment, $c_l^{\dagger}$ is the LRF function with respect to map, c_m is the marginalized cost and c_s is the manifold cost. In particular, $c_c^{\dagger}$ and $c_c^{\ddagger}$ are similarly to c_c in eqn. (8) and eqn. (6) with small difference that target frame $\mathbf{A}$ is removed, and ${}^{\mathbf{G}}\mathbf{p}_f$ is well known in $c_c^{\dagger}$ but unknown in $c_c^{\ddagger}$. The LRF cost function $c_l^{\dagger}$ is an Euclidean signed distance function (ESDF) cost, from our previous work [4]

$$c_l^{\dagger} = \sum_{j=1,\cdots,N_l} ||\mathcal{D}\left({}^{\mathbf{G}}_{\mathbf{O}}\mathbf{T} \cdot {}^{\mathbf{O}}_{\mathbf{L}}\mathbf{T} \cdot {}^{\mathbf{L}}\mathbf{p}_{f_j}\right)||^2_{\mathbf{Q}_l} \tag{13}$$

where ${}^{\mathbf{L}}\mathbf{p}_{f_j}$ is defined in eqn. (7), and $\mathcal{D}(\mathbf{m})$ represents the ESDF function at the map location $\mathbf{m}$ [4]. Our preference of dense parameterization than sparse parametric features (e.g., blobs or lines) is due to two reasons. First, dense approach exploits more measurement information and thus leads to higher theoretical accuracy, but only requires limited computational resources even on low-end ARM processors. Secondly, dense parameterization allows more flexible environment representation, which can easily model environments that lack certain types of sparse features. In fact, this is inspired by [5], where cost function is of both sparse RGB camera and dense depth camera costs. The manifold cost c_s is explained in details in our previous work [32], and the details are omitted here.

D. Loop Closure Detection

Computing $c_c^{\ddagger}$ in eqn. 12 requires loop closure detection to establish association of current image measurement to 3D features in pre-built map. Loop closure detection is widely applied in visual SLAM problems (e.g., DoW2 and feature reprojection in [2]). The weakness of this standard process is that, once a wrong loop closure detection is applied, the localization results will be severely affected and cannot be reverted. With the aid of LRF, scan-to-map ICP verification after image based detection to validate the loop closure. This can reject most false-positive results.

E. Mapping

Since we focus on commercial field and service robots, mapping is typically conducted *offline* to allow annotating points of interests (POIs) in maps, before executing real-time tasks. Our high-level mapping strategy is inspired by [33] which first build a statistically optimal sparse map and subsequently compute the dense information. The detailed procedure is as follows:

1) Localization in unknown environments: The first step of mapping is to obtain a high-quality open-loop trajectory estimate, when exploring *unknown environments*. This is equivalent to the localization methods described in Section VI-B and VI-C, by removing cost functions that are related to map, i.e., $c_c^{\dagger}$ and $c_l^{\dagger}$.

2) Pose graph optimization: From the open-loop trajectory, keyframe-to-keyframe relative pose estimate as well as the corresponding uncertainties can be derived to form a pose graph. Also, to correct long term drift, vision-based loop closure constraints between keyframes are applied [2]. The relative pose computed by ICP can be also added into the pose graph optimization as measurement constraints to enhance accuracy and stability.

3) Bundle adjustment (BA): Once pose graph optimization is finalized, we proceed to perform bundle adjustment optimization for refining all keyframes, used features, and manifold parameters in localization by a single non-linear optimizer. In BA, the features that are re-detected by loop closure are merged as the same features, pose estimates are all initialized by pose graph optimization results, and the cost function contains measurements from IMU, WE, and camera, similarly to eqn. 12.

4) Probabilistic ESDF mapping: Once BA is complete, we consider the keyframe poses to be optimally known and fixed in following operations. Then, based on LRF measurements we compute the ESDF dense map, which is a grid based map with each grid denoting its euclidean distance to the nearest obstacle. With known keyframe poses, this can be computed by probabilistic ray-casting process [4]. ESDF is introduced due to its advantages used for localization [4].

VII. Experiments

Experimental results are based on the product-ready commercial robots shown in Fig. 1. The specification of sensors are listed in Table II.

TABLE II: Sensor Suite.

Sensor	Model	Frequency (Hz)
Camera	MYNT (monocular)	10
IMU	Bosch BMI088	100
LRF	LeiShen N30101S	10
WE	Clearpath Jackal	30

A. Sensor Fusion of IMU and Wheel Encoders

The first experiment was conducted to validate our proposal of using IMU and WE together. The dead reckoning performance was evaluated by using 1) IMU-only, 2) WE-only and 3) IMU-fused-WE. The IMU-fused-WE algorithm was implemented by using only IMU and WE measurements for the localization algorithm presented in Section VI. Table I presents the results, by repeating the same tests for two indoor and two outdoor datasets. Results from the 'full' localization specified in Section VI were used as evaluation baseline, as all exteroceptive and proprioceptive sensory information was fused. The 'approximate' final translation and orientation drift of each run was compared against that from the baseline. The results demonstrate that the deployed IMU-fused-WE outperforms other alternatives by a large margin in both indoor and outdoor scenarios.

B. Calibration

In this section, we show results of the proposed calibration algorithm. The first experiment is to demonstrate that, for robots moving on 2D surfaces, not calibrating unobservable parameters (e.g., vertical axis between WE and camera) is a valid practice, as mentioned in Section V. Specifically, we

performed localization for a 2D robot for around 200 meters in an indoor environment, by manually adding errors from -20% to 20% in ${}^{\mathbf{O}}z_{\mathbf{C}}$. Results in Table. III show that this causes negligible accuracy changes.

TABLE III: Sensitivity analysis of unobservable calibration parameters (UCP) on 2D robot: Localization errors when manually adding $X\%$ errors to UCP.

	-20%	**-10%**	**-5%**	**0%**	**5%**	**10%**	**20%**
INDOOR 2D							
pos. err. (%)	0.323	0.321	0.319	0.322	0.323	0.324	0.322
rot. err. (%)	0.133	0.129	0.130	0.129	0.124	0.123	0.127

Additionally, another experiment is conducted to show that the proposed tightly-coupled multi-sensory calibration works better than pair-wise sensor calibration. Since the installed MYNT vision device contains a stereo pair[3], the stereo baseline can be considered as 'ground truth'. Sensor calibration was performed using the proposed algorithm and [22], with the left and the right camera involved separately. Subsequently, the camera-to-camera transformation was computed and compared to the ground truth. Table IV shows the results for 7 calibration datasets, in which the proposed algorithm always attained better accuracy.

TABLE IV: Compare proposed calibration alg. against [22]: Reconstructed errors in stereo baseline.

Data Number	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Proposed Alg.							
baseline. err. (mm)	0.8	0.6	1.0	0.7	1.3	1.4	0.7
Alg. [22]							
baseline. err. (mm)	3.1	2.7	3.9	3.9	4.8	3.2	3.5

C. Localization And Mapping

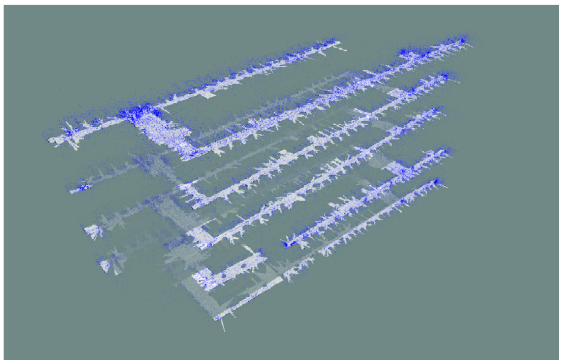

Fig. 4: Hybrid map of a five-stories building. For visualization purpose, we draw the estimated camera point clouds (blue), and the ESDF-converted occupancy grids (write).

This section demonstrates the experiment results on the performance of the proposed localization and mapping methods, for both indoor and outdoor environments. The indoor tests were conducted inside a five-stories building, and the outdoor tests were performed between buildings on our company campus. With the proposed mapping method, the representative localization maps are shown in Fig. 4 and 5. The ground truth used for indoor experiments is from a April tag [23] based method which can guarantee 5mm accuracy. The ground truth used for outdoor experiments is from a u-blox F9P RTK GNSS receiver aided by a CORS station within 5km. This outdoor ground truth can guarantee 5cm accuracy. In the experiments, operating areas were first mapped by our proposed mapping algorithm. Totally 26 indoor and 15 outdoor datasets were post-processed to get localization results. For indoor, the average localization error is around 5cm and for outdoor is around 35cm. Fig. V and VI show the error statistics from 12 randomly selected datasets for each scenario. For more comparison of our localization method with state-of-the-art algorithms, see [32].

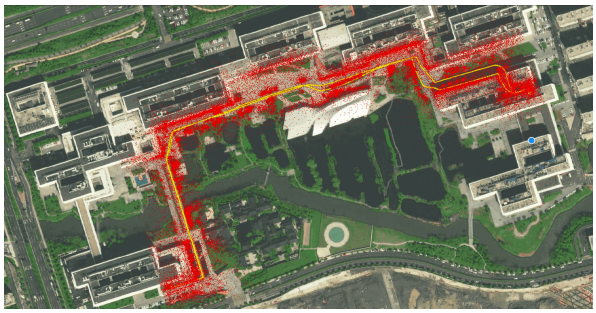

Fig. 5: Localization map between two buildings at a company's campus. For visualization purpose, we draw estimated camera point clouds (red), and robot's trajectory during mapping (yellow).

Experiments were also conducted with fully autonomous robotic systems based on our perception system. The robot with our perception system can navigate autonomously via pre-defined paths with planning and control modules. During Feb. 14-24, 2019, a total number of 84 field tests was conducted with 16.7km total distance traveled outdoor on Alibaba Xixi campus (see Fig. 5). During the tests, human intervention was only performed *once* when localization was lost (which is due to our immature software implementation). The above comparison shows the effectiveness of the proposed method. On the other hand, when [8] was implemented to perform the same tasks, human's intervention was needed even multiple times in a single day.

Additionally, when running on Nvidia Jetson TX2, the proposed localization algorithm only occupies one ARM core (equivalent to A73) and is able to achieve an average 35 msec per optimization operation, while the average keyframes frequency is less than 5Hz. This shows that the proposed method can easily run in real-time on low-cost processors.

VIII. Conclusion

In this paper, a low-cost perception system design is proposed for product-ready commercial robots, including sensors choices and configurations, and algorithmic methods.

[3]Note that only one camera is used in our perception system and all the experiments

TABLE V: Indoor Localization Error against Ground Truth.

Data Number	1	2	3	4	5	6	7	8	9	10	11	12
Mean (m)	0.0220	0.0284	0.0331	0.0249	0.0166	0.0264	0.0984	0.1188	0.0412	0.0632	0.1065	0.0591
Max (m)	0.0566	0.0881	0.0633	0.0474	0.0518	0.0439	0.2413	0.2069	0.0707	0.1217	0.1866	0.1250

TABLE VI: Outdoor Localization Error against Ground Truth.

Data Number	1	2	3	4	5	6	7	8	9	10	11	12
Mean (m)	1.0216	0.3094	0.4219	0.4109	0.3954	0.3771	0.2929	0.2156	0.2120	0.2947	0.2959	0.3179
Max (m)	2.7232	0.7567	0.9884	0.9135	0.9062	0.9361	0.7634	0.8630	0.8276	1.2216	1.1537	1.1669

Extensive experiments show that the proposed system can be deployed with accurate and robust performance.

ACKNOWLEDGEMENT

Authors appreciate the great help from our outstanding colleagues, L. Xu, C. Lin, Y. Zhao, L. Cui, B. Chen, B. Ji, D. Zhu, S. Pang, G. Li, Y. Zheng, M. Zhang, W. Liu, J. Dang, H. Chou and Z. Liu.

REFERENCES

[1] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1271–1278, 2016.

[2] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[3] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[4] Mingming Zhang, Yiming Chen, and Mingyang Li. SDF-loc: Signed distance field based 2d relocalization and map update in dynamic environments. In *American Control Conference*, 2019.

[5] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4), 2017.

[6] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *IEEE Intelligent Vehicles Symposium*, 2011.

[7] Ji Zhang and Sanjiv Singh. Laser–visual–inertial odometry and mapping with high robustness and low drift. *Journal of Field Robotics*, 35(8):1242–1264, 2018.

[8] Kejian J Wu, Chao X Guo, Georgios Georgiou, and Stergios I Roumeliotis. VINS on wheels. In *IEEE International Conference on Robotics and Automation*, pages 5155–5162, 2017.

[9] Meixiang Quan, Songhao Piao, Minglang Tan, and Shi-Sheng Huang. Tightly-coupled monocular visual-odometric SLAM using wheels and a mems gyroscope. *arXiv preprint arXiv:1804.04854*, 2018.

[10] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*, pages 834–849, 2014.

[11] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, pages 2320–2327, 2011.

[12] Mingyang Li, Hongsheng Yu, Xing Zheng, and Anastasios I Mourikis. High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation. In *IEEE International conference on Robotics and Automation*, pages 409–416, May 2014.

[13] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[14] Patrick Geneva, Kevin Eckenhoff, Yulin Yang, and Guoquan Huang. LIPS: Lidar-inertial 3d plane slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 123–130, 2018.

[15] Martin Brossard and Silvere Bonnabel. Learning wheel odometry and imu errors for localization. *hal-01874593*, 2018.

[16] Yan Lu, Joseph Lee, Shu-Hao Yeh, Hsin-Min Cheng, Baifan Chen, and Dezhen Song. Sharing heterogeneous spatial knowledge: Map fusion between asynchronous monocular vision and lidar or other prior inputs. In *The International Symposium on Robotics Research*, 2017.

[17] S Adarsh, S Mohamed Kaleemuddin, Dinesh Bose, and KI Ramachandran. Performance comparison of infrared and ultrasonic sensors for obstacles of different materials in vehicle/robot navigation applications. In *IOP Conference Series: Materials Science and Engineering*, volume 149, 2016.

[18] Dimitrios G Kottas, Kejian J Wu, and Stergios I Roumeliotis. Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3172–3179, 2013.

[19] Mingyang Li, Joel Hesch, and Zachary Moratto. Real-time visual-inertial motion tracking fault detection, November 23 2017. US Patent App. 15/596,016.

[20] Nate Merrill and Guoquan Huang. Lightweight unsupervised deep loop closure. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[21] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE.

[22] Chao X Guo, Faraz M Mirzaei, and Stergios I Roumeliotis. An analytical least-squares solution to the odometer-camera extrinsic calibration problem. In *IEEE International Conference on Robotics and Automation*, 2012.

[23] Andrew Richardson, Johannes Strom, and Edwin Olson. Aprilcal: Assisted and repeatable camera calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[24] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.

[25] Frederic Devernay and Olivier Faugeras. Straight lines have to be straight. *Machine vision and applications*, 13(1):14–24, 2001.

[26] Andrea Censi, Antonio Franchi, Luca Marchionni, and Giuseppe Oriolo. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Transactions on Robotics*, 29(2):475–492, 2013.

[27] Xingxing Zuo, Mingming Zhang, Yiming Chen, Yong Liu, Guoquan Huang, and Mingyang Li. Visual-inertial localization for skid-steering robots with kinematic constraints. In *The International Symposium on Robotics Research*, 2019.

[28] Wenbo Dong and Volkan Isler. A novel method for the extrinsic calibration of a 2d laser rangefinder and a camera. *IEEE Sensors Journal*, 18(10):4200–4211, 2018.

[29] Agostino Martinelli. State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration. *IEEE Transactions on Robotics*, 27(2):239–255, 2011.

[30] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155, Jul 2008.

[31] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual slam: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.

[32] Mingming Zhang, Yiming Chen, and Mingyang Li. Vision-aided localization for ground vehicle. *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2019.

[33] Raúl Mur-Artal and Juan D Tardós. Probabilistic semi-dense mapping from highly accurate feature-based monocular slam. In *Robotics: Science and System*, 2015.