An Effective Multi-Cue Positioning System for Agricultural Robotics

Marco Imperoli*, Ciro Potena*, Daniele Nardi, Giorgio Grisetti and Alberto Pretto

***Abstract*—The self-localization capability is a crucial component for Unmanned Ground Vehicles (UGV) in farming applications. Approaches based solely on visual cues or on low-cost GPS are easily prone to fail in such scenarios. In this paper, we present a robust and accurate 3D global pose estimation framework, designed to take full advantage of heterogeneous sensory data. By modeling the pose estimation problem as a pose graph optimization, our approach simultaneously mitigates the cumulative drift introduced by motion estimation systems (wheel odometry, visual odometry, . . .), and the noise introduced by raw GPS readings. Along with a suitable motion model, our system also integrates two additional types of constraints: (i) a Digital Elevation Model and (ii) a Markov Random Field assumption. We demonstrate how using these additional cues substantially reduces the error along the altitude axis and, moreover, how this benefit spreads to the other components of the state. We report exhaustive experiments combining several sensor setups, showing accuracy improvements ranging from 37% to 76% with respect to the exclusive use of a GPS sensor. We show that our approach provides accurate results even if the GPS unexpectedly changes positioning mode. The code of our system along with the acquired datasets are released with this paper.**

***Index Terms*—Robotics in Agriculture and Forestry, Localization and Sensor Fusion**

Supplementary Material

The datasets and the project's code are available at:

`http://www.dis.uniroma1.it/~labrococo/fsd`

I. Introduction

IT is commonly believed that the exploitation of autonomous robots in agriculture represents one of the applications with the greatest impact on food security, sustainability, reduction of chemical treatments, and minimization of the human effort. In this context, an accurate global pose estimation system is an essential component for an effective farming robot in order to successfully accomplish several tasks: (i) navigation and path planning; (ii) autonomous ground intervention; (iii) acquisition of relevant semantic information. However, self-localization inside an agricultural environment is a complex task: the scene is rather homogeneous, visually repetitive and often poor of distinguishable reference points.

Manuscript received: February, 24, 2018; Revised April, 20, 2018; Accepted June, 19, 2018. This paper was recommended for publication by Editor C. Stachniss upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the EC under Grant H2020-ICT-644227-Flourish. The Authors are with the Department of Computer, Control, and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy. Email: {imperoli, potena, nardi, grisetti, pretto}@diag.uniroma1.it.

* These two authors contribute equally to the work

Digital Object Identifier (DOI): see top of this page.

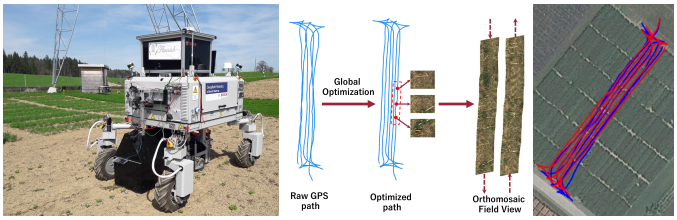

Fig. 1: (Left) The Bosch BoniRob farm robot used in the experiments; (Center) Example of a trajectory (Dataset B, see Sec. IV) optimized by using our system: the optimized pose graph can be then used, for example, to stitch together the images acquired from a downward looking camera; (Right) The obtained trajectory (red solid line) with respect to the trajectory obtained using only the raw GPS readings (blue solid line). Both trajectories have been overimposed on the actual field used during the acquisition campaign.

For this reason, conventional landmark based localization approaches can easily fail. Currently, most systems rely on high-end Real-Time Kinematic Global Positioning Systems (RTK-GPSs) to localize the UGV on the field with high accuracy [1], [2]. Unfortunately, such sensors are typically expensive and, moreover, they require at least one nearby geo-localized ground station to work properly. On the other hand, consumer-grade GPSs[1] usually provide noisy data, thus not guaranteeing enough accuracy and reliability for safe and effective operations. Moreover, a GPS cannot provide the full state estimation of the vehicle, i.e. its attitude, that is an essential information to perform a full 3D reconstruction of the environment. In this paper, we present a robust and accurate 3D global pose estimation system for UGVs (Unmanned Ground Vehicles) designed to address the specific challenges of an agricultural environment. Our system effectively fuses several heterogeneous cues extracted from low-cost, consumer grade sensors, by leveraging the strengths of each sensor and the specific characteristics of the agricultural context. We cast the global localization problem as a pose graph optimization problem (Sec. II): the constraints between consecutive nodes are represented by motion estimations provided by the UGV wheel odometry, local point-cloud registration, and a visual odometry (VO) front-end that provides a full 6D ego-motion estimation with a small cumulative drift[2]. Noisy, but drift-free GPS readings (i.e., the GPS *pose solution*), along with a pitch and roll estimation extracted by using a MEMS Inertial Measurement Units (IMU), are directly integrated as prior nodes. Driven by the fact that both GPS and visual odometry

[1] In this paper, we use GPS as a synonym of the more general acronym GNSS (Global Navigation Satellite System) since almost all GNSSs use at least the GPS system, included the two GNSSs used in our experiments.

[2] In VO open-loop systems, the cumulative drift is unavoidable.

provide poor estimates along the z-axis, i.e. the axis parallel to the gravity vector, we propose to improve the state estimation by introducing two additional altitude constraints:

1) An altitude prior, provided by a Digital Elevation Model (DEM);
2) A smoothness constraint for the altitude of adjacent nodes[3].

Both the newly introduced constraints are justified by the assumption that, in an agricultural field, the altitude varies slowly, i.e. the soil terrain can be approximated by piece-wise smooth surfaces. The smoothness constraints exploit the fact that a farming robot traverses the field by following the crop rows, hence, by using the Markov assumption, the built pose graph can be arranged as a Markov Random Field (MRF). The motion of the UGV is finally constrained using an Ackermann motion model extended to the non-planar motion case. The integration of such constraints not only improves the accuracy of the altitude estimation, but it also positively affects the estimate of the remaining state components, i.e. x and y (see Sec. IV).

The optimization problem (Sec. III) is then iteratively solved by exploiting a graph based optimization framework [3] in a sliding-window (SW) fashion (Sec. III-C), i.e., optimizing the sub-graphs associated to the most recent sensor readings. The SW optimization allows to obtain on-line localization results that approximate the results achievable by an off-line optimization over the whole dataset.

In order to validate our approach (Sec. IV), we used and made publicly available with this paper two novel challenging datasets acquired using a Bosch BoniRob UGV (Fig. 1, left) equipped with, among several others calibrated sensors, two types of low-cost GNSSs: a Precise Point Positioning (PPP) GPS and a consumer-grade RTK-GPS. We report exhaustive experiments with several sensors setups, showing remarkable results: the global localization accuracy has been improved up to 37% and 76%, compared with the raw localization obtained by using only the raw RTK-GPS and PPP-GPS readings, respectively (e.g., Fig. 1). We also show that our approach allows localizing the UGV even though the GPS performances temporarily degrade, e.g. due to a signal loss.

A. Related Work

The problem of global pose estimation for UGVs has been intensively investigated, especially in the context of self driving vehicles and outdoor autonomous robots moving in urban environments. The task is commonly approached by integrating multiple sources of information. Most of the state-of-the-art systems rely on IMU-aided GPS [4], while they differ in the other sensor cues they use in the estimation process. Cameras are used primarily in [5], [6], [7], [8], [9], while LIDARs have been used in [10].

In urban scenarios, the presence of a prior map allows to improve the estimation by constraining the robot motion. [11], [12] use 2D road maps, while [13] propose to use more rich DEMs. The sensors fusion is usually carried out by means of parametric [11] or discrete [10] filtering, pose graph optimization [7], [8], set-membership positioning [13], or hybrid topological/filtering [5].

As stated in the introduction, these approaches cannot be used effectively in agricultural environments, since a prior map is typically not available. In addition, crops exhibit substantially a less stable structure than an urban environment, and their appearance varies substantially over time. Hence, the localization inside an agricultural field, by using a map built on-line, turns out to be extremely difficult since stable features are hard to find. For this reason, most of the available localization methods for farming robots are based on expensive global navigation satellite systems [14], [15], [2]. However, relying on the GPS as the primary localization sensor exposes the system to GPS related issues: potential signal losses, multi-path, and a time-dependent accuracy influenced by the satellite positions.

The main task of an agricultural robot is to follow the crop rows and take some action along the way. To this extent, English *et. al* [16], proposed a vision based crop-row following system. While effective, this system assumes that the crops are clearly visible from the camera of the robot, and this is not true at all growth stages of the plants. Furthermore, the estimate of a crop row tracking tends to accumulate drift along the row direction.

To gain robustness and relax the accuracy requirements on the GPS, it is natural to use the plants as landmarks to build a map using a SLAM algorithm. To this extent, Cheein *et al.* [17] propose to find and to use as landmarks, in a SLAM system, olive tree stems. The stem detection algorithm uses both camera and laser data. Other approaches are based on the detection of specific plant species and thus they address very specific use cases. Jin *et al.* [18] focus on the individual detection of corn plants by using RGB-D data. In [1], the authors propose a MEMS based 3D LIDAR sensor to map an agricultural environment by means of a per-plant detection algorithm. Gai *et al.* [6] proposed an algorithm that follows leaf ridges detected in RGB images to the center. Similar approaches rely on Stem Emerging Points (SEPs) localizations: Mitdiby *et al.* [19] follow sugar beet leaf contours to find the SEPs. In [20] the authors perform machine learning-based SEP localization in an organic carrot field. Kraemer *et al.* [21] proposed an image-based plant localization method that exploits a CNN to learn time-invariant SEPs.

B. Contributions

In this paper, we provide a robust and effective positioning framework targeted for agricultural applications that aims to achieve high level accuracy with low cost GPSs. We integrate in an efficient way, a wide range of heterogeneous sensors into a pose graph by adapting the features of each of them to the specificity of the farming scenario. We exploit domain-specific patterns to introduce further constraints such as a MRF assumption and a DEM that contribute to the improvement of the state estimation. We evaluate our system with extensive experiments that highlight the contribution of each employed cue. We also provide an open-source implementation of our

[3]The term "adjacent" denotes nodes that are temporally or spatially close.

code and two challenging datasets with ground truth, acquired with a Bosch BoniRob farm robot.

II. Multi-Cue Pose Graph

The challenges that must be addressed to design a robust and accurate global pose estimation framework for farming applications are twofold: (i) the agricultural environment appearance, being usually homogeneous and visually repetitive; (ii) The high number of cues that have to be fused together. In this section we describe how we formulate a robust pose estimation procedure able to face both these issues.

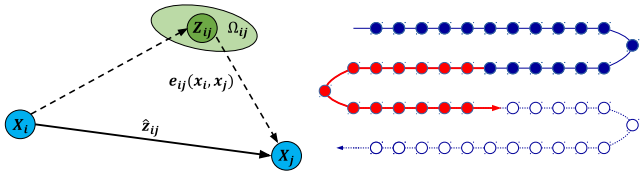

Fig. 2: (Left) Illustration of an edge connecting two nodes x_i and x_j. The error $e_{i,j}$ is computed between the measurement $z_{i,j}$ and the predicted measurement $\hat{z}_{i,j}$. In addition, each edge encodes an information matrix $\Omega_{i,j}$ that represents the uncertainty of the measurement; (Right) Sliding-window sub-graph optimization: nodes that belong to the current sub-graph are painted in red, old nodes no more optimized are painted in blue, while nodes that will be added in the future are painted in white.

The proposed system handles the global pose estimation problem as a pose graph optimization problem. A pose graph is a special case of factor graph[4], where the factors $\langle\cdot\rangle$ are only connected to variables (i.e., nodes) pairs, and variables are only represented by robot poses. For this reason, it is common to represent each factor with an edge. Solving a factor graph means finding a configuration of the nodes for which the likelihood of the actual measurements is maximal. Since we assume that all the involved noises follow a Gaussian distribution, we can solve this problem by employing an iterative least square approach.

We define $X = \{x_0, ..., x_{N-1}\}$ as the vector of graph nodes that represents the robot poses at discrete points in time, where each $x_i = (T_i, R_i)$ is represented by the full 3D pose in terms of a translation vector $T_i = [t_{x,i}\ t_{y,i}\ t_{z,i}]'$ and, using the axis-angle representation, an orientation vector $R_i = [r_{x,i}\ r_{y,i}\ r_{z,i}]'$, both in $\mathbb{R}^3$. This pose is defined with respect to a global reference centered in x_0[5]. We denote with z the sensor measurements that can be related to pairs or single nodes. Let $z_{i,j}$ be a relative motion measurement between nodes x_i and x_j, while z_i be a global pose measurement associated to the node x_i. Additionally, let $\Omega_{i,j}$ and Ω_i represent the information matrices encoding the reliability of such measurements, respectively. From the poses of two nodes x_i and x_j, it is possible to compute the expected relative motion measurement $\hat{z}_{i,j}$ and the expected global measurement $\hat{z}_i$ (see Fig. 2, left). We formulate the errors between those quantities as:

$$e_{i,j} = z_{i,j} - \hat{z}_{i,j},\ \ e_i = z_i - \hat{z}_i, \tag{1}$$

[4]A factor graph is a bipartite graph where nodes encode either variables or measurements, namely the factors.

[5]We transform each global measurement (e.g., GPS measurements) in the reference frame x_0.

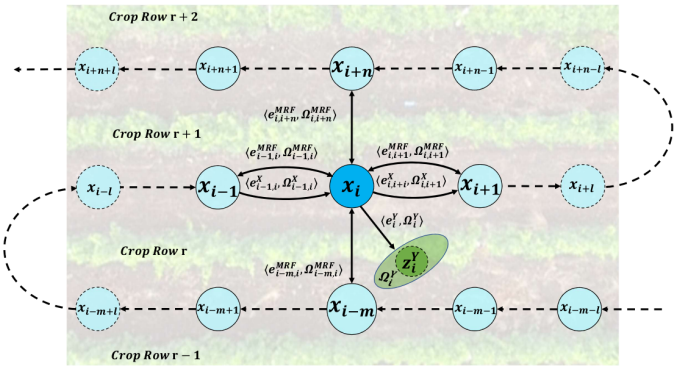

Fig. 3: Overview of the built pose graph. Solid arrows represent graph edges, that encode conditional dependencies between nodes, dotted arrows temporal relationships between nodes. For the sake of clarity, we show here only the edges directly connected with the node x_i, by representing only one instance for each class of edges: (i) the binary non directed MRF constraint $\langle e^{MRF}_{i,i+1}, \Omega^{MRF}_{i,i+1}\rangle$; (ii) the binary directed edge $\langle e^{\mathcal{X}}_{i,i+1}, \Omega^{\mathcal{X}}_{i,i+1}\rangle$ induced from sensor $\mathcal{X} \in \{VO, WO, AMM, LID\}$; (iii) the unary edge $\langle e^{\mathcal{Y}}_i, \Omega^{\mathcal{Y}}_i\rangle$ induced by sensor $\mathcal{Y} \in \{GPS, DEM, IMU\}$. We superimposed the graph on a cultivated field to remark the relationship between the graph structure and the crop rows arrangement.

Thus, for a general sensor $\mathcal{X}$ providing a relative information, we can characterize an edge (i.e., a *binary factor* $\langle e^{\mathcal{X}}_{i,i+1}, \Omega^{\mathcal{X}}_{i,i+1}\rangle$) by the error $e^{\mathcal{X}}_{i,i+1}$ and the information matrix $\Omega^{\mathcal{X}}_{i,j}$ of the measurement, as described in [22]. In other words, an edge represents the relative pose constraint between two nodes (Fig. 2, left). In order to take into account also global pose information, we use unary constraints, namely a measurement that constrains a single node. Hence, for a general sensor $\mathcal{Y}$ providing an absolute information, we define $\langle e^{\mathcal{Y}}_i, \Omega^{\mathcal{Y}}_i\rangle$ as the prior edge (i.e., an *unary factor*) induced by the sensor $\mathcal{Y}$ on node x_i. Fig. 3 depicts a portion of a pose graph highlighting both unary and binary edges. Each edge acts as a directed spring with elasticity inversely proportional to the relative information matrix associated with the measurement that generates the link. Our pose graph is built by adding an edge for each sensor reading, for both relative (e.g., wheel odometry readings) and global (e.g., GPS readings) information. In addition, we propose to integrate other prior information that exploit both the specific target environment and the assumptions we made. In the following, we report the full list of edges exploited in this work, divided between local (relative) and global measurements (we report in brackets the acronyms used in Fig. 3):

Local measurements: Wheel odometry measurements (WO), Visual odometry estimations (VO), Elevation constraints between adjacent nodes (MRF), Ackermann motion model (AMM), Point-clouds local registration (LID).

Global measurements: GPS readings (GPS), Digital Elevation Model data (DEM), IMU readings (IMU).

We define $\langle e^{VO}_{i,i+1}, \Omega^{VO}_{i,i+1}\rangle$ as the relative constraint induced by a visual odometry algorithm, $\langle e^{WO}_{i,i+1}, \Omega^{WO}_{i,i+1}\rangle$ as the relative constraint induced by the wheel odometry, and $\langle e^{LID}_{i,i+1}, \Omega^{LID}_{i,i+1}\rangle$ as the relative constraint obtained by aligning the local point-clouds perceived by the 3D LIDAR sensor.

Often, GPS and visual odometry provide poor estimates of the robot position along the z-axis (i.e, the axis that represents

its *elevation*). In the GPS case, this low accuracy is mainly due to the Dilution of Precision, multipath or atmospheric disturbances, while in the visual odometry this is due to the 3D locations of the tracked points. In a typical agricultural scenario most of the visual features belong to the ground plane. Hence, the small displacement of the features along the z-axis may cause a considerable drift. On the other hand, agricultural fields usually present locally flat ground levels and, moreover, a farming robot usually traverses the field by following the crop rows. Driven by these observations, we enforce the local ground smoothness assumption by introducing an additional type of local constraints that penalizes the distance along the z-coordinate between adjacent nodes. Therefore, the built pose graph can be augmented by a 4-connected MRF [23]: each node is conditionally connected with the previous and the next nodes in the current crop row, and with the spatially closest nodes that belong to the previous and next crop rows, respectively. We refer to this constraint as $\langle e^{MRF}_{i,i+1}, \Omega^{MRF}_{i,i+1} \rangle$ in Fig. 3 (e.g., the set $\{x_{i-1}, x_i, x_{i+1}, x_{i-m}, x_{i+n}\}$). We then add a further type of local constraint based on the Ackermann steering model, that assumes that the robot is moving on a plane. In this work, we relax this assumption to local planar motions between temporal adjacent nodes. Such a motion plane is updated with the attitude estimation of the subsequent node. We integrate this constraint by means of a new type of edge, namely $\langle e^{AMM}_{i,i+1}, \Omega^{AMM}_{i,i+1} \rangle$.

Local constraints are intrinsically affected by a small cumulative drift: to overcome this problem, we integrate in the graph drift-free global measurements as position prior information. In particular, we define a GPS prior z^{GPS}_i and an IMU prior z^{IMU}_i with associated information matrices Ω^{GPS}_i and Ω^{IMU}_i. The IMU is used as a drift-free roll and pitch reference[6], where the drift resulting from the gyroscopes integration is compensated by using the accelerometers data.

Finally, we introduce an additional global measurement by means of an altitude prior, provided by a DEM. A DEM is a special type of Digital Terrain Model that represents the elevation of the terrain at some location, by means of a regularly spaced grid of elevation points [24]. The DEM maps a 2D coordinate to an absolute elevation. Since we assume that the altitude varies slowly, we can use the current position estimate T_i (i.e., the $t_{x,i}$ and $t_{y,i}$ components) to query the DEM for a reliable altitude estimation $z_{DEM,i} = f(t_{x,i}, t_{y,i})$, with associated information matrix Ω^{DEM}_i. The cost function is then assembled as follows:

$$J_i = \sum_{i=1}^{N-1} \Big(\underbrace{\sum_{\mathcal{X}} e^{\mathcal{X}}_{i,i-1} \Omega^{\mathcal{X}}_{i,i-1} e^{\mathcal{X}'}_{i,i-1}}_{Binary\ constraints} + \underbrace{\sum_{\mathcal{Y}} e^{\mathcal{Y}}_{i} \Omega^{\mathcal{Y}}_{i} e^{\mathcal{Y}'}_{i}}_{Unary\ constraints} + \underbrace{\sum_{j \in \mathbb{N}_i} e^{MRF}_{i,j} \Omega^{MRF}_{i,j} e^{MRF'}_{i,j}}_{MRF\ constraint} \Big) \quad (2)$$

[6]We experienced that integrating the full inertial information inside the optimization did not positively affect the state estimation: our intuition is that the slow, often unimodal, motion of our robot makes the IMU biases difficult to estimate and sometimes predominant over the motion components.

where $\mathcal{X}$ and $\mathcal{Y}$ represent respectively the set of binary and unary constraints defined above (see Fig. 3), and $\mathbb{N}_i$ stands for the 4-connected neighborhood of the node x_i.

III. Pose Graph Optimization

In this section, we focus on the solution of the cost function reported in Eq. 2, describing the error computation, the weighting factors assignment procedure and the on-line and off-line versions of the optimization. We finally report some key implementation insights.

A. Error Computation

For each measurement z, given the current graph configuration, we need to provide a prediction $\hat{z}$ in order to compute errors in Eq. 2. $\hat{z}$ represents the expected measurement, given a configuration of the nodes, which are involved in the constraint. Usually, for a binary constraint, this prediction is the relative transformation between the nodes x_i and x_j, while for an unary constraint it is just the full state x_i or a subset of its components. We define $\mathbf{X}_i$ as a general homogeneous transformation matrix related to the full state of the node x_i (e.g., the homogeneous rigid body transformation generated from T_i and R_i) and $\Phi(\cdot)$ as a generic mapping function from $\mathbf{X}_i$ to a vector; now, we can express $\hat{z}_{i,j}$ and $\hat{z}_i$ as:

$$\hat{z}_{i,j} = \Phi(\mathbf{X}_i^{-1} \cdot \mathbf{X}_j), \ \hat{z}_i = \Phi(\mathbf{X}_i) \quad (3)$$

In this work not all the constraints belong to $SE(3)$: indeed, most of used sensors (e.g., WO, IMU) can only observe a portion of the full state encoded in x. Therefore, in the following, we will show how we obtain the expected $\hat{z}$ for each involved cue (for some components, we omit the subscripts i and j by using the relative translations dt and rotations dr between adjacent nodes):

VO and LID: these front-ends provide the full 6D motion: we build $\hat{z}^{VO}$ and $\hat{z}^{LID}$ by computing the relative transformation between the two connected nodes as in Eq. 3;

WO: the robot odometry provides the planar motion by means of a roto-translation $z_{WO} = (dt_x, dt_y, dr_z)$: we build $\hat{z}^{WO}$ as $\Phi(\mathbf{X}_i^{-1} \cdot \mathbf{X}_j)|_{t_x, t_y, r_z}$, the subscripts after $\Phi(\cdot)$ specify that the map to the vector $\hat{z}$ involves only such components;

MRF and DEM: they constrain the altitude of the robot, we obtain the estimated measurements as:

$$\hat{z}^{MRF}_{i,j} = (0, 0, t_{z,i} - t_{z,j}, 0, 0, 0) \quad (4a)$$

$$\hat{z}^{DEM}_{i} = (0, 0, t_{z,i}, 0, 0, 0) \quad (4b)$$

GPS: this sensor only provides the robot position:

$$\hat{z}^{GPS}_i = (T_i, 0_{3\times1}) \quad (5)$$

IMU: from this measurement we actually exploit only the *roll* and *pitch* angles, being the rotation around the z axis provided by the IMU usually affected by not negligible inaccuracies. Therefore, we obtain $\hat{z}^{IMU}_i = \Phi(\mathbf{X}_i)|_{r_{x,i}, r_{y,i}}$;

AMM: we formulate such a constraint by a composition of two transformation matrices. The first one encodes a roto-translation of the robot around the so called *Instantaneous*

Center of Rotation (ICR). We follow the same formulation presented in [25]:

$$\mathbf{X}(\rho, dr_z) = \begin{bmatrix} cos(\frac{dr_z}{2}) & -sin(\frac{dr_z}{2}) & 0 & \rho \cdot cos(\frac{dr_z}{2}) \\ sin(\frac{dr_z}{2}) & cos(\frac{dr_z}{2}) & 0 & \rho \cdot sin(\frac{dr_z}{2}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

where ρ is the norm of the translation along dt_x and dt_y. Additionally, we add a further rotation along those two axes, taking also into account the ground slope, by rotating the ideal plane on which the vehicle steers following the Ackermann motion model:

$$\mathbf{X}(dr_x, dr_y) = \begin{bmatrix} R(dr_x, dr_y) & 0_{1x3} \\ 0_{3x1} & 1 \end{bmatrix} \quad (7)$$

Hence, we obtain $\hat{z}^{AMM}$ as $\Phi(\mathbf{X}(dr_x, dr_y) \cdot \mathbf{X}(\rho, dr_z))$.

B. Dynamic Weight Assignment

The impact of each constraint in the final cost function (Eq. 2) is weighted by its relative information matrix. As a consequence, such information matrices play a crucial role in weighting the measurements, i.e. giving much reliability to a noisy sensor can lead to errors in the optimization phase. We tackle this problem by dynamically assigning the information matrix for each component as follows:

WO: we use as information matrix $\Omega_{i,j}^{WO}$ the inverse of the covariance matrix Σ^{WO} of the robot odometry, scaled by the magnitude of the distance and rotation traveled between the nodes x_i and x_j, as explained in [26];

VO: we use the inverse of the covariance matrix Σ^{VO} provided as output by the visual odometry front-end, weighting the rotational and translational sub-matrices ($\Sigma^{VO,R}$ and $\Sigma^{VO,T}$) with two scalars $\lambda_{VO,R}$ and $\lambda_{VO,T}$, experimentally tuned. Since we do not directly tune the VO system internal parameters, we employ these "VO agnostic" scaling factors that have the analogous effects as injecting a higher sensor noise. In the experiments, we set $\lambda_{VO,R} = 5$ and $\lambda_{VO,T} = 1$;

MRF: we set the information matrix $\Omega_{i,j}^{MRF} = diag(0, 0, w_z^{MRF}, 0, 0, 0)$. The weight $w_z^{MRF} = \lambda_{MRF}/|x_i - x_j|_{t_x,t_y}$ is inversely proportional to the distance in the (x, y) plane between the two nodes, while λ_{MRF} has been experimentally tuned. $\lambda_{MRF} = 0.8$ in the experiments;

GPS: we use as information matrix Ω_i^{GPS}, the inverse of the covariance matrix Σ^{GPS} provided by the GPS sensor;

AMM: we use as information matrix $\Omega_{i,j}^{AMM}$, an identity matrix scaled by the magnitude of the traveled distance between the nodes x_i and x_j, similarly to the wheel odometry constraint. This allows to model the reliability of such a constraint as inversely proportional to the traveled distance;

IMU: we use as information matrix Ω_i^{IMU}, the inverse of the covariance matrix Σ^{IMU} provided by the IMU sensor;

DEM: we set the information matrix $\Omega_i^{DEM} = diag(0, 0, w_z^{DEM}, 0, 0, 0)$, where w_z^{DEM} is empirically tuned. In the experiments we set $w_z^{DEM} = 5$;

LID: we set the information matrix $\Omega_{i,j}^{LID}$ as the inverse of the covariance matrix estimated from the transformation provided by the registration algorithm (e.g., an ICP algorithm), by using the procedure described in [27]. Such an information matrix allows adapting the influence of the point-cloud alignment inside the optimization process, enabling to correctly deal also with the lack of geometrical structure on some dimensions, e.g. in farming scenarios with small plants.

C. Sliding-Window Optimization

A re-optimization of the whole pose graph presented above, every time a new node is added, cannot guarantee the real-time performances required for on-line field operations, especially when the graph contains a large amount of nodes and constraints. We solve this issue by employing a sliding-window approach, namely performing the optimization procedure only on a sub-graph that includes a sequence of recent nodes. Each time a new node associated with the most recent sensor readings is added to the graph, the sub-graph is updated by adding the new node and removing its oldest one, in a SW fashion. The optimization process is performed only on the current sub-graph, while older nodes maintain the state assigned during the last optimization where they were involved. In order to preserve the MRF constraints, the size of the sub-graph is automatically computed so that any adjacent nodes in the previous row are included (see Fig. 2, right). A global optimization of the whole pose graph is then carried out off-line, using as initial guess the node states assigned on-line using the SW approach.

D. Implementation Details

Temporal Synchronization: In the previous sections, we tacitly assumed that all sensor measurements associated with a graph node share the same time stamp. However, in a real context, this is usually not true. In our implementation, we trigger the creation of new nodes every $step_{WO}$ meters ($0.3\ m$ in our experiments), by using the wheel odometry as a distance reference. We associate to each node synchronized estimates of the other sensor readings, obtained by means of linear interpolation over the closest readings of each used sensor. This enables to associate to the same node a set of heterogeneous sensor readings that share the same time stamp.

Visual Odometry Failures: VO systems are usually tuned by default to provide high accuracy at the expense of the robustness. We address this limitation by employing a simple strategy designed to mitigate VO failures. We exploit the local reliability of the WO: when the difference between WO and VO is greater than a given threshold, we assume a failure of the latter. In this case, we reduce the influence of the VO during the pose graph optimization by downscaling its information matrix.

Point-Cloud Registration: Point-clouds acquired by a 3D LIDAR are typically too sparse to perform a robust alignment: thus, we accumulate a number of LIDAR readings into a single point-cloud by using the motion estimations provided by the VO. The point-cloud registration is finally performed using the Iterative Closest Point (ICP) algorithm.

Graph Optimization: We perform both the on-line and off-line pose graph optimizations (Sec. III-C) using the Levenberg-

Marquardt algorithm implemented in the *g2o* graph optimization framework [3].

IV. Experiments

In order to analyze the performance of our system, we collected two datasets[7] with different UGV steering modalities. In *Dataset A* the robot follows 6 adjacent crop rows by constantly maintaining the same global orientation, e.g. half rows have been traversed by moving the robot backward, while in *Dataset B* the robot is constantly moving in the forward direction. Both datasets include data from the same set of sensors: (i) wheel odometry; (ii) VI-Sensor device (stereo camera + IMU) [28]; (iii) Velodyne VLP-16 3D LIDAR; (iv) a low cost U-blox RTK-GPS; (v) an U-blox Precise Point Positioning (PPP) GPS. For a comprehensive description of the UGV farm robot, the sensors setup and the calibration procedure, we refer the interested readers to the on-line supplementary material[8].
In all our experiments, we employ Stereo DSO [29] as VO subsystem and the ICP implementation provided by the PCL library as point-cloud registration front-end. The IMU, the wheel odometry and both the GPSs provide internally filtered outputs (attitude, relative and absolute positions, respectively), along with covariance matrices associated to the outputs in the IMU and GPSs cases. We built the DEM of the inspected field by using the Google Elevation API that provides, for the target field, measurements over a regularly spaced grid with a resolution of 10 meters. We interpolated such measurements to provide a denser information. We acquired a ground truth 3D reference by using a LEICA laser tracker. This sensor tracks a specific target mounted on the top of the robot and provides a position estimation (x, y and z) with millimeter-level accuracy. Both datasets have been acquired by using the Bosch BoniRob farm robot (Fig. 1, left) on a field in Eschikon, Switzerland (Fig. 1, right). In addition to these two datasets, we have created a third dataset (*Dataset C*), where we simulate a sudden RTK-GPS signal loss, e.g. due to a communication loss between the base and the rover stations. In particular, we simulated the accuracy losses by randomly switching for some time to the PPP-GPS readings.

In the following, we report the quantitative results by using the following statistics build upon the localization errors with respect to the ground truth reference: Root Mean Square Error ($RMSE$ in the tables), maximum and mean absolute error (Max and $Mean$), and mean absolute error along each component (err_x, err_y and err_z).

A. Dataset A and Dataset B

This set of experiments shows the effectiveness of the proposed method and the benefits introduced by each cue. We report in Tab. I the results obtained by using different sensor combinations and optimization procedures over *Dataset A* and *Dataset B*. The table is split according to the type of GPS sensor used; the sensor setups that bring the overall best results are highlighted in bold. We also compared our system with the ORB SLAM 2 system [30], a best-in-class Visual SLAM system, with its mapping and loop closures back-ends activated. For a fair comparison, we added the GPS information (PPP and RTK) as a global constraint at each key-frame triggered by ORB SLAM 2.

[7] www.dis.uniroma1.it/~labrococo/fsd
[8] www.dis.uniroma1.it/~labrococo/fsd/ral2018sup.pdf

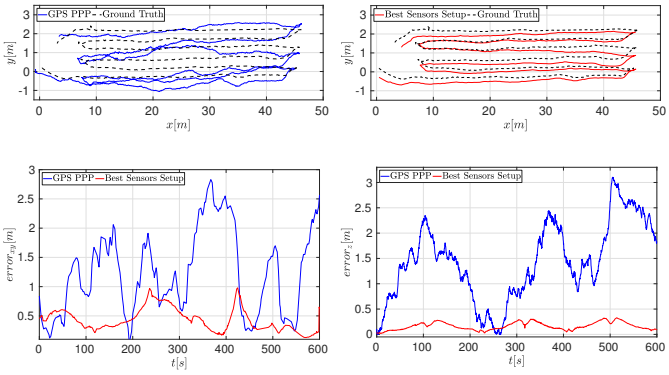

Fig. 4: Dataset A, PPP-GPS: (Top) Qualitative top view comparison between the raw GPS trajectory (left) and the optimized trajectory (right); (Bottom): absolute x, y (left) and z (right) error plots for the same trajectories.

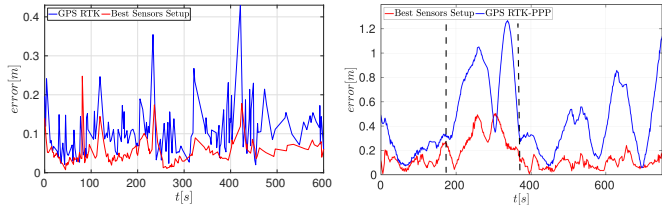

Fig. 5: (Left) Dataset A, RTK-GPS: Absolute error plots for the raw GPS trajectory and the optimized trajectory obtained by using the best sensors configuration (see Tab. I). (Right) Dataset C, absolute error plots for the raw GPS trajectory and the optimized trajectory (see Tab. III). The time interval when the signal loss happens is bounded by the two dashed lines.

A first result is the positive impact of including the new proposed constraints in the optimization: both the ELEV and MRF cues individually integrated lead to noteworthy improvements in the estimation along the z when a noisy GPS is used (PPP-GPS case). Another remarkable result is the decreasing error trend, almost monotonic: the more sensors we introduced in the optimization process, the smaller the resulting $RMSE$ and Max errors are. This behavior occurs in both *Dataset A* and *Dataset B*, and proves how the proposed method properly handles all the available sources of information. Another important outcome is the relative $RMSE$ improvement obtained between the worst and the best set of cues, which is around the 37% for RTK case, and 76% for the PPP case; in both these setups our system outperforms the ORB SLAM 2 system. A noteworthy decrease of the error also happens to the Max error statistic, respectively, 40% and 70%: this fact brings a considerable benefit to agricultural applications, where spikes in the location error might lead to harming crops. For the best performing sensor setup, we also report the results obtained by using the SW, on-line pose graph optimization procedure (Sec. III-C): also in this case the relative improvement is remarkable (32% and 67%, respectively), enabling a safer and more accurate real-time UGV navigation.

TABLE I: Error statistics in *Dataset A* and *Dataset B* by using different sensor setups and constraints for the global, off-line and the sliding-window (SW), on-line pose graph optimization procedures. The results of the ORB SLAM 2 system (OS2 in the table) are reported for both type of GPSs.

									DatasetA					**DatasetB**				
GPS	WO	VO	IMU	AMM	ELEV	LIDAR	MRF	SW	err_x	err_y	err_z	Max	$RMSE$	err_x	err_y	err_z	Max	$RMSE$
PPP									0.349	0.582	1.577	2.959	1.710	0.306	0.501	1.484	2.875	1.621
	✓								0.311	0.520	1.537	2.954	1.630	0.246	0.416	1.424	2.829	1.504
	✓				✓				0.343	0.572	0.475	1.627	1.071	0.241	0.408	0.492	1.782	1.168
	✓	✓							0.239	0.412	0.672	1.628	0.961	0.222	0.362	1.298	2.392	1.211
	✓	✓				✓			0.233	0.422	0.649	1.421	0.863	0.227	0.361	1.292	2.571	1.242
	✓	✓					✓		0.239	0.411	0.528	1.398	0.719	0.221	0.364	1.019	2.362	1.119
	✓	✓		✓		✓			0.224	0.411	0.551	1.375	0.726	0.201	0.397	0.881	2.019	0.951
	✓	✓	✓			✓			0.222	0.389	0.531	1.281	0.729	0.229	0.407	0.652	1.613	0.829
	✓	✓	✓	✓					0.239	0.361	0.523	1.272	0.739	0.231	0.369	0.641	1.461	0.732
	✓	✓	✓	✓		✓			0.224	0.371	0.453	1.124	0.621	0.221	0.362	0.619	1.611	0.734
	✓	✓	✓	✓			✓		0.234	0.360	0.440	1.093	0.564	0.199	0.360	0.475	1.161	0.660
	✓	✓	✓	✓	✓	✓			0.234	0.342	0.311	0.921	0.422	0.198	0.361	0.463	1.121	0.604
	✓	✓	✓	✓	✓		✓		0.211	**0.331**	**0.282**	0.897	0.416	0.182	0.339	0.369	1.198	0.471
	✓	✓	✓	✓	✓	✓	✓		**0.201**	**0.331**	0.289	**0.824**	**0.401**	**0.173**	**0.331**	**0.321**	**1.117**	**0.461**
	✓	✓	✓	✓	✓	✓	✓	✓	0.252	0.419	0.349	0.991	0.549	0.291	0.431	0.459	1.291	0.652
	OS2+GPS								0.234	0.417	0.643	1.534	0.915	0.209	0.401	0.371	2.123	1.047
RTK									0.059	0.051	0.121	0.431	0.128	0.054	0.062	0.091	0.322	0.122
	✓								0.053	**0.042**	0.105	0.431	0.125	0.049	0.058	0.086	0.321	0.119
	✓	✓							0.053	**0.042**	0.054	0.279	0.088	0.047	0.048	0.062	0.192	0.091
	✓	✓	✓						0.048	0.049	0.060	0.306	0.092	0.045	0.046	0.064	0.209	0.091
	✓	✓		✓					0.046	0.047	0.061	0.279	0.090	0.045	**0.045**	0.064	0.211	0.090
	✓	✓		✓		✓			0.046	0.047	0.061	0.278	0.089	0.045	**0.045**	0.062	0.197	0.090
	✓	✓	✓	✓					0.046	0.050	0.056	**0.248**	0.088	0.045	0.046	0.039	0.165	0.075
	✓	✓	✓			✓	✓		0.047	0.049	**0.034**	0.251	0.076	0.045	0.046	0.035	0.154	0.074
	✓	✓	✓	✓	✓	✓	✓		0.051	0.049	0.068	0.312	0.097	0.046	0.048	0.064	0.219	0.095
	✓	✓	✓	✓		✓	✓		**0.045**	0.048	**0.034**	0.260	**0.075**	**0.044**	0.046	**0.034**	**0.151**	**0.073**
	✓	✓	✓	✓		✓	✓	✓	0.053	0.051	0.042	0.272	0.084	0.051	0.051	0.035	0.172	0.084
	OS2+GPS								0.051	0.045	0.059	0.293	0.097	0.051	0.054	0.068	0.231	0.102

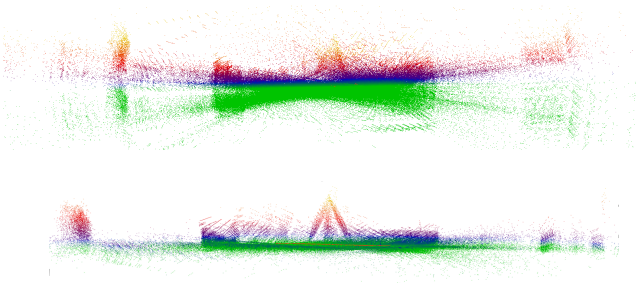

Fig. 6: Comparison between output point-clouds: (top) without IMU and LIDAR and (bottom) with IMU and LIDAR in the optimization.

Fig. 4 (top) depicts a qualitative top view comparison between the raw PPP-GPS trajectory (top-left) and the trajectory (top-right) obtained after the pose graph optimization, using the best sensors configuration in *Dataset A*. The error plots (bottom) show how the introduction of additional sensors and constraints allows to significantly improve the pose estimation. Similar results for *Dataset A* and RTK-GPS are reported in Fig. 5 (left).

For both GPSs, the maximal error reduction happens when all the available cues are used within the optimization procedure except for the low cost RTK-GPS case, where the ELEV constraint worsens the error along the z axis. Actually, the RTK-GPS usually provides an altitude estimate, which is more accurate than the one provided by the interpolated DEM. It is also noteworthy to highlight the propagation of the improvements among state dimensions: the integration of constraints that only act on a part of the state (e.g., IMU, LIDAR, ELEV) also positively affects the remaining state components.

As a further qualitative evaluation, in Fig. 6 we report the global point-cloud obtained by rendering LIDAR scans at each estimated position, with and without the IMU and LIDAR contributions within the optimization procedure: the attitude estimation greatly benefits from these contributions. The runtimes of our system are reported in Tab. II, for both the off-line and on-line, sliding-window cases.

TABLE II: Runtime performance for the global, off-line and the sliding-window (SW), on-line pose graph optimization (Core-i7 2.7 GHz laptop).

	SW	$\#Nodes$	$\#Edges$	$\#Iters$	$time(s)$
Dataset A		786	8259	24	13.493
	✓	98	763	4	0.0989
Dataset B		754	8032	22	12.993
	✓	104	851	5	0.1131

B. Dataset C

This set of experiments is designed to prove the robustness of the proposed system against sudden losses in the GPS sensor accuracy. In Tab. III we report the quantitative results of our system over *Dataset C* by means of $RMSE$ and Max errors. Even in the presence of a RTK-GPS signal loss that lasts for more than one crop row, the best sensors setup leads to a remarkable $RMSE$ of 0.166 m and a relative improvement around the 72%. Moreover, also in *Dataset C* the $RMSE$ and the Max error statistics follow the same decreasing trend shown in Tab. I. In Fig. 5 (right) we compare the absolute error trajectories for the best sensors configuration against the error trajectory obtained by using only the GPS measurements:

TABLE III: Error statistics in *Dataset C* by using different sensor setups and constraints in the optimization procedure.

								DatasetC	
GPS	WO	VO	IMU	AMM	ELEV	LIDAR	MRF	Max	$RMSE$
RTK+PPP								1.313	0.647
	✓							1.291	0.613
	✓	✓						1.259	0.552
	✓	✓		✓		✓		1.171	0.431
	✓	✓	✓			✓		0.882	0.356
	✓	✓		✓	✓	✓		0.551	0.223
	✓	✓	✓				✓	0.655	0.204
	✓	✓	✓	✓			✓	0.521	0.201
	✓	✓	✓	✓	✓	✓	✓	0.534	0.181
	✓	✓	✓	✓		✓	✓	**0.419**	**0.168**

the part where the signal loss occurs is affected by a higher error. Another interesting observation regards the non-constant effects related to the use of the ELEV constraint. As shown in Tab. III, in some cases it allows to decrease the overall error, while in other cases it worsens the estimate. The latter happens when the pose estimation is reliable enough, i.e. when most of the available constraints are already in use. As explained in section IV-A, in such cases the ELEV constraint does not provide any additional information to the optimization procedure, while with a less accurate PPP-GPS its use is certainly desirable.

V. Conclusions

In this paper, we present an effective global pose estimation system for agricultural applications that leverages in a reliable and efficient way an ensemble of cues. We take advantage from the specificity of the scenario by introducing new constraints exploited inside a pose graph realization that aims to enhance the strengths of each integrated information. We report a comprehensive set of experiments that support our claims: the provided localization accuracy is remarkable, the accuracy improvement well scale with the number of integrated cues, the proposed system is able to work effectively with different types of GPS, even in presence of signal degradations. The open-source implementation of our system along with the acquired datasets are made publicly available with this paper.

Acknowledgment

We are grateful to Wolfram Burgard for providing us with the Bosch BoniRob, and to Raghav Khanna and Frank Liebisch to help us in acquiring the datasets.

References

[1] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3D lidar sensor," *Robotics and autonomous systems*, vol. 59, no. 5, pp. 265–273, 2011.

[2] M. Nørremark, H. W. Griepentrog, J. Nielsen, and H. T. Søgaard, "The development and assessment of the accuracy of an autonomous GPS-based system for intra-row mechanical weed control in row crops," *Biosystems Engineering*, vol. 101, no. 4, pp. 396–410, 2008.

[3] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[4] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, Inc., 2008.

[5] D. Schleicher, L. M. Bergasa, M. Ocana, R. Barea, and M. E. Lopez, "Real-time hierarchical outdoor SLAM based on stereovision and GPS fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 440 – 452, 2009.

[6] I. Parra, M. ngel Sotelo, D. F. Llorca, C. Fernndez, A. Llamazares, N. Hernndez, and I. Garca, "Visual odometry and map fusion for GPS navigation assistance," in *IEEE International Symposium on Industrial Electronics (ISIE)*, 2011.

[7] J. Rehder, K. Gupta, S. T. Nuske, and S. Singh, "Global pose estimation with limited GPS and long range visual odometry," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2012.

[8] K. Vishal, C. V. Jawahar, and V. Chari, "Accurate localization by fusing images and GPS signals," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015.

[9] M. Schreiber, H. Knigshof, A.-M. Hellmund, and C. Stiller, "Vehicle localization with tightly coupled GNSS and visual odometry," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[10] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A navigation system for robots operating in crowded urban environments," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.

[11] C. Fouque and P. Bonnifait, "On the use of 2D navigable maps for enhancing ground vehicle localization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[12] M. A. Brubaker, A. Geiger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[13] V. Drevelle and P. Bonnifait, "A set-membership approach for high integrity height-aided satellite positioning," *GPS Solutions*, vol. 15, no. 4, pp. 357–368, October 2011.

[14] A. Stoll and H. D. Kutzbach, "Guidance of a forage harvester with GPS," *Precision Agriculture*, vol. 2, no. 3, pp. 281–291, 2000.

[15] B. Thuilot, C. Cariou, L. Cordesses, and P. Martinet, "Automatic guidance of a farm tractor along curved paths, using a unique CP-DGPS," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001.

[16] A. English, P. Ross, D. Ball, and P. Corke, "Vision based guidance for robot navigation in agriculture," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2014.

[17] F. A. Cheein, G. Steiner, G. P. Paina, and R. Carelli, "Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection," *Computers and Electronics in Agriculture*, vol. 78, pp. 195 – 207, 2011.

[18] J. Jian and T. Lie, "Corn plant sensing using realtime stereo vision," *Journal of Field Robotics*, no. 67, pp. 591–608.

[19] H. S. Midtiby, T. M. Giselsson, and R. N. Jrgensen, "Estimating the plant stem emerging points (PSEPs) of sugar beets at early growth stages," *Biosystems Engineering*, vol. 111, pp. 83 – 90, 2012.

[20] S. Haug, P. Biber, A. Michaels, and J. Ostermann, "Plant stem detection and position estimation using machine vision," in *13th Intl. Conf. on Intelligent Autonomous Systems (Workshops)*, 2014.

[21] F. Kraemer, A. Schaefer, A. Eitel, J. Vertens, and W. Burgard, "From plants to landmarks: Time-invariant plant localization that uses deep pose regression in agricultural fields," *In arXiv:1709.04751*.

[22] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.

[23] A. Blake, P. Kohli, and C. Rother, *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

[24] C. Hirt, *Digital Terrain Models*. Springer International Publishing, 2014, pp. 1–6.

[25] A. Pretto, E. Menegatti, and E. Pagello, "Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[26] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[27] S. M. Prakhya, L. Bingbing, Y. Rui, and W. Lin, "A closed-form estimate of 3D ICP covariance," in *IAPR International Conference on Machine Vision Applications (MVA)*, 2015.

[28] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time SLAM."

[29] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *International Conference on Computer Vision (ICCV)*, 2017.

[30] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.