Accepted manuscript. Link to published version:
https://doi.org/10.1016/j.compag.2022.107419

An RGB-D multi-view perspective for autonomous agricultural robots

Fabio Vulpi[a,b], Roberto Marani[a], Antonio Petitti[a], Giulio Reina[b], Annalisa Milella[a]

[a] *Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA), Institute of National Research Council of Italy (CNR), via Amendola 122 D-O, Bari, 70126, Bari, Italy*
[b] *Department of Mechanics, Mathematics and Management, Polytechnic of Bari, Via Orabona 4, Bari, 70125, Bari, Italy*

Abstract

Automated in-field data gathering is essential for crop monitoring and management and for precision farming treatments. To this end, consumer-grade digital cameras have been shown to offer a flexible and affordable sensing solution. This paper describes the integration and development of a cost-effective multi-view RGB-D device for sensing and modelling of agricultural environments. The system features three RGB-D sensors, arranged to cover a horizontal field of view of about 130 deg in front of the vehicle, and a suite of localization sensors consisting of a tracking camera, an RTK-GPS sensor and an IMU device. The system is intended to be mounted on-board an agricultural vehicle to provide multi-channel information of the surveyed scene including color, infrared and depth images, which are then combined with localization data to build a multi-view 3D geo-referenced map of the traversed crop. The experimental demonstrator of the multi-sensor system is presented along with the steps for the integration of the different sensor data into a unique multi-view map. Results of field experiments conducted in a commercial vineyard are included, as well, showing the effectiveness of the proposed system. The resulting map could be useful for precision agriculture applications, including crop health monitoring, and to support autonomous driving.

Keywords: Agricultural robotics, advanced perception, RGB-D sensing, multi-view imaging, environment mapping, precision farming

1. Introduction

Persistent and timely crop monitoring is crucial for the development of sustainable food production systems. While remote sensing from satellites and aircrafts has been successfully used for some decades to allow for rapidly mapping and characterize wide areas through acquisition of multispectral imagery and 3D data, they generally lack the spatio-temporal resolution needed for precision agriculture or phenotyping tasks. In crops with smaller extension, Unmanned Aerial Vehicles (UAVs) have been effectively adopted for remote crop survey with higher spatial and temporal resolution compared to satellite and airborne devices. In high-density crops, however, collecting information on biophysical properties at plant or leaf/fruit level via aerial sensing may be still infeasible. As a result, ground-based sensing through Unmanned Ground Vehicles (UGVs) has been proposed for in-field close-range applications [1].

In this respect, imaging sensors embedded on ground vehicles have recently attracted much attention as an effective solution to collect high resolution proximal data and provide information on plant characteristics at centimetre or sub-centimetre scale. At the same time, they can be used for vehicle guidance automation and scene perception and understanding in general, thus contributing to all aspects of crop monitoring automation, from data gathering up to high-level data processing and interpretation.

Among imaging sensors, consumer-grade RGB-D cameras, i.e., sensors that embed color and depth sensing into a unique device, have emerged in the last years in mobile robotics applications, to provide the vehicle with a real-time representation of both appearance and 3D structure of the environment. One shortcoming of typical consumer-grade color-depth sensors is their limited field-of-view, which makes them unsuitable for applications that need continuous survey of extensive areas, like in agricultural settings.

In this work, a multi-view RGB-D camera system is proposed to enhance the perception ability of an unmanned agricultural robot. The system features three RGB-D cameras arranged to cover a horizontal field of view of about 130 deg. The cameras are integrated with localization sensors, including a stereo-based tracking camera, an RTK-GPS and an IMU, which provide accurate vehicle position information for image geo-referencing based on Extended Information Filter (EIF). All the sensors are physically integrated in a custom-built sensor box, which is designed to be self-contained from both a computational and energy point of view and independent from the particular vehicle architecture. The system is intended to generate accurate maps to facilitate operations on a narrow scale with a smaller environment footprint. To this end, a point cloud assembler that uses EIF-based pose estimates and the known relative poses between sensors is developed to reconstruct a geo-referenced multi-view 3-D map of the traversed environment, which could provide a useful input to precision farming technologies and crop monitoring tasks.

An additional use of the map is that it can be incorporated into a high-fidelity simulator that can support the development and pre-testing of algorithms for autonomous driving. In this respect, the map can be converted into a mesh representation using the Ball Pivoting Algorithm (BPA) and imported into a simulation environment under Gazebo [2].

The rest of the paper is structured as follows. First, related work is presented in Section 2. The hardware and software design of the multi-view sensing device is reported in Section 3. Section 4 describes the multi-view 3D mapping approach. The simulator is detailed in Section 5. Experimental results obtained in real agricultural settings are presented in Section 6. Finally, conclusions are drawn in Section 7.

2. Related Work

The availability of up-to-date and accurate data is an essential pre-requisite for precision farming tasks, such as variable rate application of fertilizers/pesticides, identification of infected plants or invasive species, and controlled traffic farming. While satellite and airborne technologies have been in use for some decades to effectively provide multi-spectral and 3D information in wide agricultural and forestry areas, these platforms generally lack the resolution needed to observe stems, leaves or fruits. Satellite images typically have pixel resolution of hundreds of meters and airborne sensing may provide resolution of a few meters, whereas monitoring orchards or vineyards requires observations at a smaller scale. Information update frequency is also limited, varying from hours to several days. In crops with smaller extension, UAVs equipped with RGB, multispectral or LiDAR sensors, have been adopted to overcome these bottlenecks, allowing for efficient crop survey at user–defined spatio-temporal resolutions to assess vegetation vigor or for canopy characterization [3], [4]. However, in high density crops, using aerial data can still be ineffective for precise measurement at leaf/fruit level, e.g., for health status assessment and yield estimation.

As an alternative or complementary approach, proximal sensing from ground-based or manually deployed devices can be performed. Proximal sensors range from RGB cameras to high-resolution hyperspectral imaging, infrared (IR) thermal cameras, and 2D/3D LiDARs. Applications include fruit detection and counting [5], up to plant phenotyping [6], health status assessment and growth monitoring [7], [8]. While these methods were proved to be effective and accurate for detailed information extraction, they are often constrained to structured environments, such as greenhouses and specific acquisition conditions, such as controlled illumination or pre-defined positioning of the sensing devices, or they require the adoption of expensive high-resolution sensors [9], which limits their practical implementation.
In order to address these issues, crop monitoring by agricultural ground robots has been proposed as a step forward to automated proximal measurement and characterization of high-value crops and soils [10], [11]. While much work has been done in the context of ground robots for harvesting and picking operations, the use of UGVs for in-field crop monitoring and assessment has been proposed more recently. UGVs can carry a number of sensing devices, thus potentially providing an efficient means to gather multi-modal information at a narrow scale. At the same time, they can be equipped with manipulators and actuators to perform targeted actions, such as selective spraying or fertilizing, with relatively high operating times.
Although UGVs offer enough payload to transport a number of bulky sensors, keeping low complexity and costs is a major requirement for in-field implementation. In this respect, visual sensors mounted on ground robots have been shown to provide an efficient and affordable solution in a wide range of agricultural applications, including plant and fruit detection, fruit grading, ripeness detection, yield prediction, plant and fruit health protection and disease detection. In addition, visual sensors provide a rich source of information to support autonomous navigation functions such as localization, obstacle detection and situation awareness in general [1], [12].
Among visual sensors, portable consumer-grade RGB-D cameras, like Microsoft Kinect, have been receiving growing attention, as an effective means to recover in real-time 3D textured models of plants and extract plant and fruit features [13], although the application of this sensor remains mostly limited to indoor contexts. A novel family of highly portable, consumer depth cameras has been introduced by Intel in 2015 (R200 and D4xx, Santa Clara, CA, USA). These cameras are similar to the Kinect sensor in scope and cost, but use a different working principle based on IR stereo, which makes them

more suitable for outdoor conditions. In addition, their output include RGB information, infrared images and 3D depth data, thus covering a wide range of information about the scene. The potential of these sensors for agricultural applications has been investigated in recent works [14], [15].
Following this research trend, this work explores the potential of a multi-view RGB-D system for geo-referenced image acquisition and mapping of a high-value crop, like a vineyard. The device is built following a modular approach and can be mounted on any agricultural vehicle to provide ground-based 3D reconstruction of the traversed crop rows. Data acquisition and processing can be carried out during vehicle operations, in a non-invasive and completely automatic way, while requiring low investment and maintenance costs.
One specific aspect addressed is accurate vehicle localization. Localization of the UGV is essential for correct merging of point cloud streams and thus for the construction of geo-referenced 3-D maps. In [16], the UGV pose estimation problem is formulated as a pose graph optimization to mitigate sensor drift and significantly improve state estimation accuracy using a Digital Elevation Model (DEM) and a Markov Random Field (MRF) assumption. Authors in [17] proposed a Simultaneous Localization And Mapping (SLAM) method for generating the map of an agricultural environment and simulated it on Gazebo and Robot Operating System (ROS) for the case of an apple farm, showing good results in fruit mapping. A well-established solution to the localization problem to fuse information from multiple sensors is Kalman filtering. In this work, we use the information form of the Kalman Filter as data fusion strategy for heterogeneous sensors. The reason is related to the high reliability of such algorithm, as confirmed by recent research (e.g.,[18], [19]). Other alternatives have been investigated in the literature, including, for example, particle filtering, which however has proven to be less accurate for localization purposes ([20], [21]).

3. Multi-view Sensing Device

This section describes the development of a multi-sensor box for close range sensing and modelling of agricultural environments. The sensor suite is intended to be mounted on board an agricultural robot and is designed to be self-contained, both from a computational and energy point of view, and independent from the particular vehicle architecture.

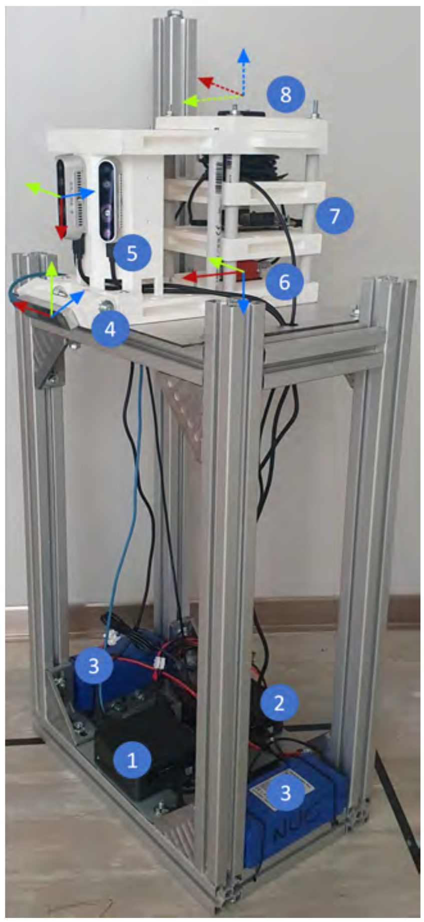
3.1. Hardware Design

The sensor suite is shown in Figure 1 (a). It consists of two sensor arrays, namely a *Perception Sensors* array and a *Navigation Sensors* array. The perception sensors include three Intel RealSense D435 RGB-D cameras arranged to cover a wide horizontal field of view of about 130 deg in front of the vehicle, which extends up to about 145 deg when considering infrared depth information only. The mounting case allows one to alternatively place up to two cameras in lateral configuration, e.g., to keep the image plane parallel to a crop row for tasks such as row following and/or monitoring. A closeup of the multi-camera system is shown in Figure 1 (b). The navigation sensors comprise one Intel RealSense tracking camera T265, one X-Sense IMU MTi-300 and two U-Blox GPS Zed-F9P providing RTK-GPS data in rover-base configuration. All sensors are integrated in a 3-D printed PLA box (see Figure 1 (c)), which was designed following a modular approach, so that it can be assembled in multiple ways according to the specific needs of the test field. The described sensor suite can be fixed to the vehicle through a metal frame, built with aluminium bars and plates and designed to be stable and of adjustable height. Two Intel NUC7i7DNHE computers are used for data gathering. The PCs, powered by lithium batteries, are fixed at the bottom of the metal frame. Overall, the proposed sensor box provides a flexible and self-contained data gathering device with a cost of about 6.5k €(i.e., 27% for the two processing units, 45% for the IMU, 14% for the cameras, 7% for the GPS, and 7% for the batteries).

3.2. Acquisition Software Design

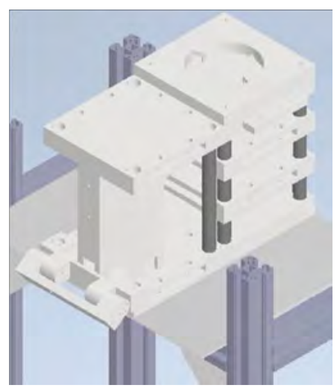
The data gathering pipeline of the sensor suite is shown in Figure 2. The sensor box provides two processing units, one running Ubuntu and the other running Windows 11. The NUC Ubuntu is devoted to gathering positional measurements produced by the GPS sensor and the IMU sensor. Data acquisition is made through ROS drivers using a dedicated ROS node for each sensor. Then, all the acquired data are stored in ROS bags.

For image acquisition and storage, a software package, named SensorBox, was developed using the Intel RealSense SDK 2.0 (v. 2.49), running on NUC Windows. The software architecture of the whole package is divided into two executables: *MultiBagReader* and *MultiBagWriter*. The scheme of the first executable (*MultiBagWriter*) is shown in Figure 3. It works in a producer-consumer logic, where the Intel RealSense cameras connected to the processing unit are first opened to produce the data which is then consumed,



(a)

(b)



(c)

Figure 1: (a) Demonstrator of the UGV's sensor box: (1)-(2) Intel NUC Windows PCs; (3) Batteries, (4) T265 Camera, (5) D435 Cameras, (6) X-Sense MTI-300 IMU, (7) U-blox ZED F9P board, (8) Sensor Box GPS antenna. (b) Closeup of the multi-camera system. (c) CAD model of the sensor frame.

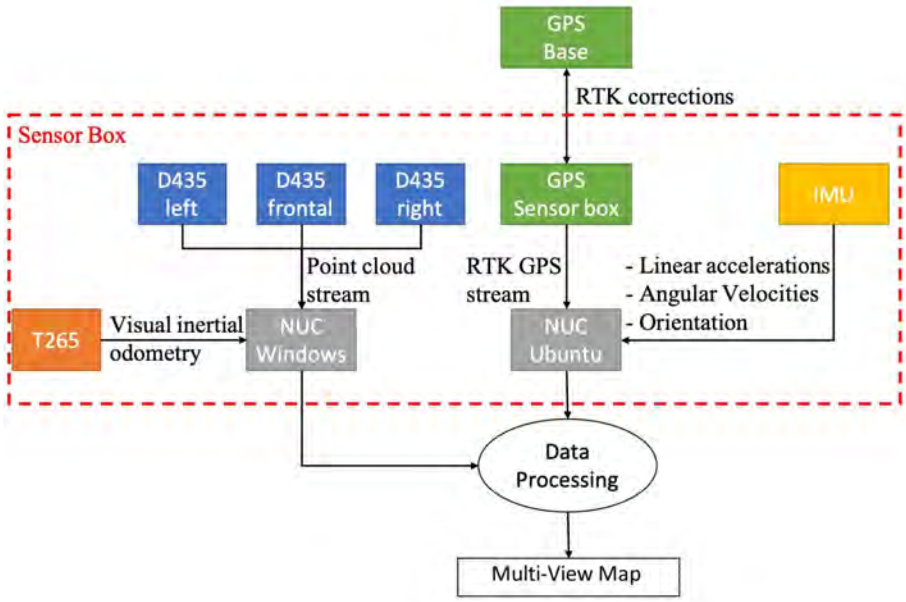


Figure 2: Sensor box data flow.

i.e. displayed, to show the acquired field of view and/or the computed visual odometry. Then, when the user starts the acquisition, the data are encapsulated in several ROS bags, stored on the local hard disk of the NUC. In this way, each camera produces a bag file at the maximum achievable rate (up to 30 fps), without any further processing to prevent frame drops. It is worth noticing that each camera works in a free run mode and, thus, their frames are not temporally synchronized, i.e. acquired exactly at the same time instant. The second executable (*MultiBagReader*) opens the bags, divides the RealSense pipelines to have single streams in each pipeline, and then reports all the acquired frames to a global temporal reference, thus performing software synchronization. The software features a user interface for both writing and reading modules, as shown in Figure 4. The open-source code of the software is available on GitHub (https://github.com/ispstiima/SensorBox).

3.3. Sensor Synchronization and Calibration

The association of heterogeneous data requires temporal and spatial calibration. For time synchronization, a timestamp-based approach was adopted, whereby each sensor observation was marked with a timestamp. In addition,

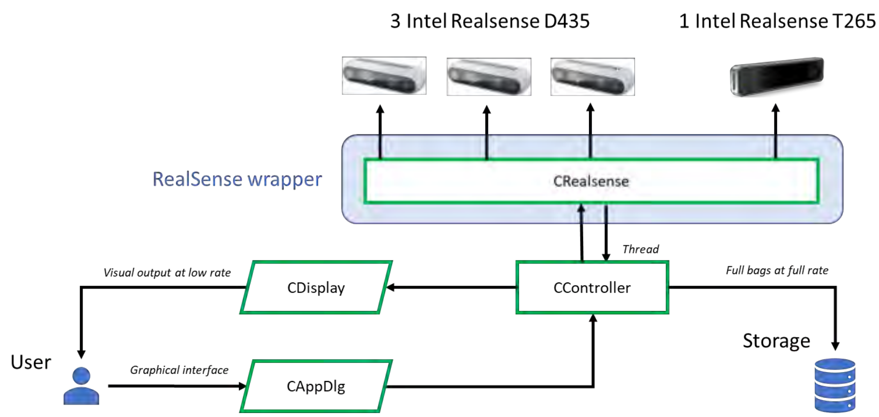


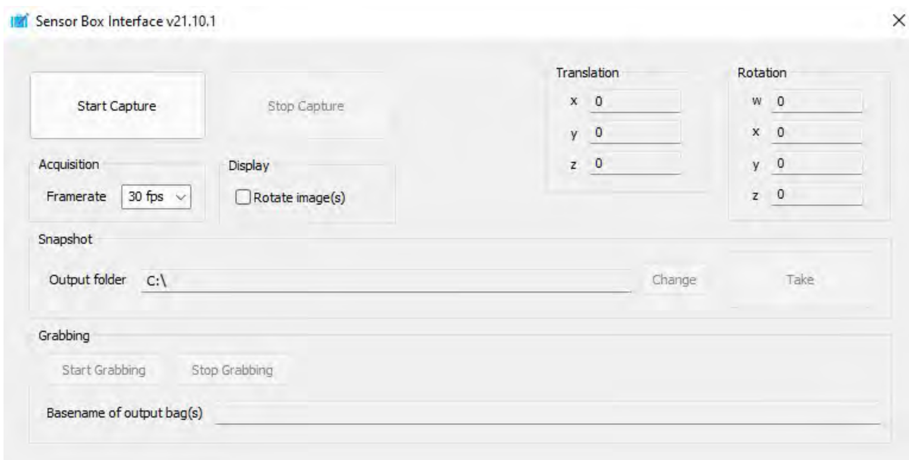
Figure 3: Schematic of the image acquisition software.

to register all sensor data with respect to a common reference frame, spatial calibration was performed to estimate the relative position and orientation of the sensors with respect to each other. Spatial calibration was performed by construction, considering that all the sensors are located in the sensor box at fixed positions. This proved to be sufficiently accurate for the purpose of this work, although optimization strategies, such as the one proposed by the authors in [22], can be also adopted to further improve the registration accuracy.

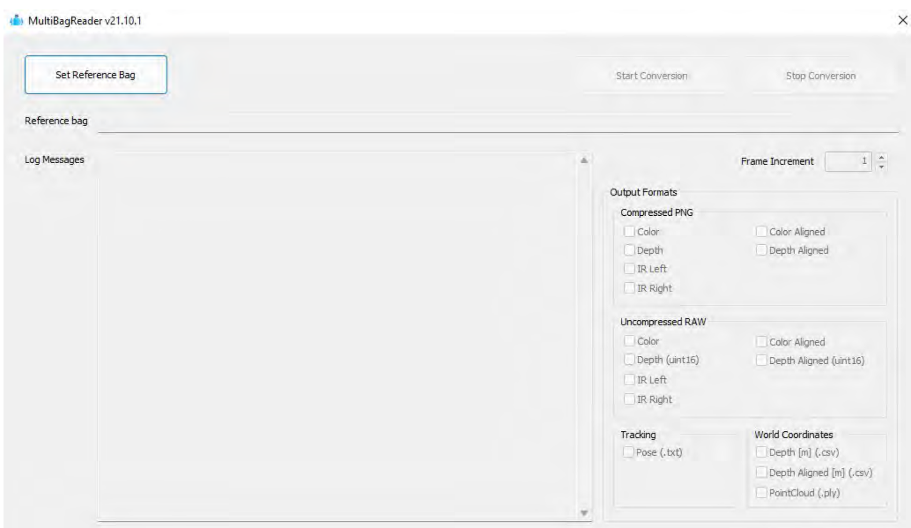
4. Multi-view 3D Mapping

The data acquired by the sensor suite are processed to build a multi-view map of the traversed environment, following multiple stages. First, data from GPS, IMU and T265 sensors are fused by an EIF to generate pose estimates. Successively, the point clouds obtained by each of the three RGB-D cameras are assembled into a unique map, using the EIF pose estimates and the known relative poses between the sensors. The map can be then converted into a 3D mesh representation for efficient storage and inspection, as well as, for import in a dedicated simulation environment, as will be described later in Section 5. In more detail, with reference to Figure 1, let us introduce the reference frames denoted with the following subscripts:

- sb: Sensor Box frame (Figure 1, a-6)
- w: East-North-Up world frame (Figure 1, a-8)



(a)



(b)

Figure 4: User interface of the multi-view camera system: (a) interface for data acquisition and storage (MultiBagWriter); (b) interface for reading stored image databases (MultiBagReader).

- *cam*: Camera Frame (Figure 1, a-5)
- s: Reference frame for the s-th sensor (RTK-GPS, T265, IMU).

Furthermore, quantities of interest are presented here to facilitate the understanding of the mapping reconstruction process:

- $p_{B-A}(t) \in \mathbb{R}^{1,3}$: position vector of reference frame A at time t expressed in reference frame B
- $\boldsymbol{q}_{B-A}(t) \in \mathbb{C}^{1,4}$: quaternion orientation of reference frame A at time t expressed in reference frame B. Note that only quaternions are denoted in bold.

For the reader's convenience, basic concepts on quaternion analysis appear in the Appendix. The interested reader is referred to the literature (e.g., [23]) for more details.

The mapping algorithm proceeds according to the following three steps:

1. *Pose estimation*: position and orientation of the sensor box can be estimated from different sensors (RTK-GPS, T265, IMU) and fused within an EIF. In general, the sensor s can provide information about its position and orientation expressed either in the world frame or with respect to its initial pose. In the former case, Equations (1) give the sensor box orientation $\mathbf{q}_{w-sb}$ and position p_{w-sb} in the world frame, knowing $\mathbf{q}_{s-sb}$ and p_{sb-s}

$$\begin{aligned} \mathbf{q}_{w-sb}(t) &= \mathbf{q}_{w-s}(t)\mathbf{q}_{s-sb} \\ [p_{w-sb}(t), 0] &= [p_{w-s}(t), 0] - \mathbf{q}_{w-sb}(t)[p_{sb-s}, 0] \end{aligned} \tag{1}$$

If the sensor provides $p_{s_0-s}(t)$ and $\mathbf{q}_{s_0-s}(t)$ with respect to its initial condition s_0, Equations (2) compute p_{w-sb} and $\mathbf{q}_{w-sb}$

$$\begin{aligned} \mathbf{q}_{w-sb}(t) &= \mathbf{q}_{w-sb_0}\mathbf{q}_{sb_0-s}(t)\mathbf{q}_{s-sb} \\ [p_{w-sb}(t), 0] &= \mathbf{q}_{w-sb_0}\mathbf{q}_{sb-s}[p_{s_0-s}(t), 0]\mathbf{q}_{s-sb}\mathbf{q}_{sb_0-w} - \\ &\quad + \mathbf{q}_{w-sb}(t)[p_{sb-s}, 0] + \\ &\quad + [p_{w-sb_0}, 0] \end{aligned} \tag{2}$$

Apart from $\mathbf{q}_{sb-s} = \mathbf{q}^{-1}_{s-sb}$, initial position and orientation of the sensor box in the world frame (p_{w-sb_0} and $\mathbf{q}_{w-sb_0}$) are needed to compute p_{w-sb}

and $\mathbf{q}_{w-sb}$. In Equations (2) notation $[p, 0]$ indicates the quaternion with zero real part corresponding to position vector p.

Please note that in both Equations (1) and (2) quaternion multiplication is omitted for ease of notation.

The predictive model used to implement the EIF is expressed by Equations (3)

$$\begin{aligned}
\tilde{p}_{t+1|t} &= \tilde{p}_{t|t} + \tilde{v}_{t|t} \cdot \Delta t \\
[\tilde{v}_{t+1|t}, 0] &= [\tilde{v}_{t|t}, 0] + (\tilde{\mathbf{q}}_{t|t}[a_{sb}, 0]\tilde{\mathbf{q}}_{t|t}^{-1} - [g, 0]) \cdot \Delta t \\
\tilde{\mathbf{q}}_{t+1|t} &= \|(\tilde{\mathbf{q}}_{t|t} + \frac{\Delta t}{2} \cdot \tilde{\mathbf{q}}_{t|t}[\omega_{sb}, 0])\|
\end{aligned} \tag{3}$$

where position $\tilde{p}$, velocity $\tilde{v}$, and quaternion $\tilde{\mathbf{q}}$ describe the state of the sensor box over time expressed in the w frame. Subscript $t+1|t$ denotes predicted value and $t|t$ indicates posterior value corrected with measurements. Expression $\|\mathbf{q}\|$ represents quaternion normalization. Angular velocity ω_{sb} and linear acceleration a_{sb} are measurements provided by the IMU and expressed in the sb frame, and g denotes gravity acceleration in the world frame.

2. *Point cloud assembly* The pose estimates are then used to transform and assemble the point clouds provided by each RGB-D camera in its own frame. Denoting with $P_{cam} \in \mathbb{R}^{3,1}$ a generic point of the point cloud in the camera frame cam and with $T_{cam} \in \mathbb{R}^{4,4}$ the homogeneous transformation matrix from frame cam to frame sb, Equation (4) expresses P_{cam} in the sb frame as P_{sb}.

$$[{P_{sb}}^T, 1]^T = T_{cam}[{P_{cam}}^T, 1]^T \tag{4}$$

The homogeneous transformation matrix T_{sb} from the sensor box frame to the inertial frame can be obtained by Equation (5) using the pose estimated by the EIF

$$T_{sb} = \begin{bmatrix} rotm(\tilde{\mathbf{q}}_{t|t}) & \tilde{p}_{t|t}^T \\ [0,0,0] & 1 \end{bmatrix} \tag{5}$$

where $rotm(\mathbf{q}) \in \mathbb{R}^{3,3}$ returns the rotation matrix uniquely assigned to quaternion $\mathbf{q}$. Finally, 3D points can be transformed from the sb to the w frame using Equation (6)

$$[{P_w}^T, 1]^T = T_{sb}[{P_{sb}}^T, 1]^T \tag{6}$$

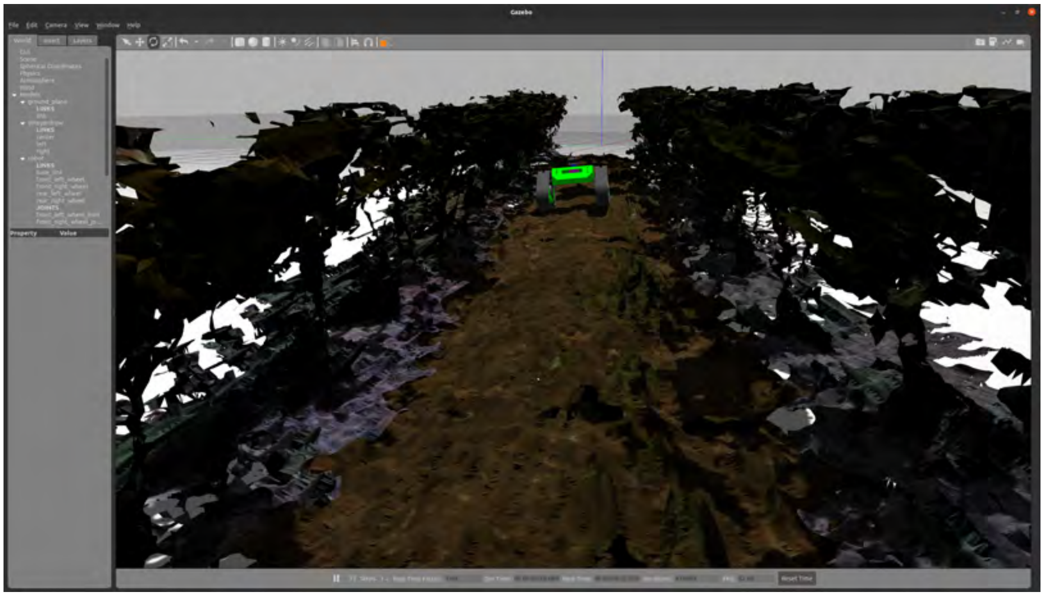


Figure 5: Gazebo simulation environment composed of a vineyard row and a mobile robot.

3. *Map merging.* Assembled point clouds gathered by the three cameras and transformed in the world frame are then merged together and downsampled (voxel grid size= 0.01 m).

5. Simulation Environment

The possibility to simulate robots in outdoor environments under different conditions is of paramount importance in agricultural robotics. In this regard, the most famous simulator framework used by the robotic community is Gazebo [2]. Gazebo is an open-source simulator born in 2002 for robotics. After over 15 years of development, Gazebo is undergoing a significant upgrade and modernization. Today, Gazebo is part of Ignition, which is a collection of open-source software libraries designed by Open Robotics to simplify the development of high-performance applications. Creating Gazebo models means creating SDF (Simulation Description Format) files to define SDF Model Objects. The primary element composing an SDF file is the link. A link contains the physical properties of one body of the model. Each link may have many collision and visual elements. Usually, these two elements define the 3D mesh file describing the 3D surface of the object. Gazebo requires that mesh files be formatted as STL, Collada, or OBJ, with Collada

and OBJ preferred formats.
In this work, the point cloud map, obtained as described Section 4, is modeled using a meshing algorithm, which allows one to generate a mesh-based representation for map import in the Gazebo simulation environment, according to the following steps:

- **Downsampling**: to downsample the obtained dense map is a not mandatory task that enables to speed up the mesh reconstruction process and improves the outcome of the whole process. To accomplish this task, the samples are generated according to a Poisson-disk distribution [24];
- **Normals computation**: the knowledge of the normals is necessary to reconstruct the surface of the elements composing the map. Normals are computed on the basis of the 10 closest points;
- **Surface reconstruction**: this step regards the reconstruction of the surface starting from the set of points and normals. In this case, the Ball Pivoting algorithm is used [25] to compute a triangle mesh. It is based on the principle that three points form a triangle if a ball of a user-specified radius touches them without containing any other point;
- **Texture mapping**: the texture mapping is build by triangle-by-triangle parametrization;
- **Color transfer**: this step concerns the process of projecting a 2D image to a 3D model's surface for texture mapping, the so called *UV mapping*. Once a UV map is available, the color can be transferred to the reconstructed surface;
- **Save the mesh**: finally, the mesh is ready to be exported in a suitable format.

Thus, thanks to the created mesh files, it is possible to develop a Gazebo SDF model object describing the reconstructed vineyard row. As an example, Figure 5 showcases a mobile robot crossing a vineyard row developed by following the procedure described above.

Figure 6: Robotic platform used for in-field testing equipped with the multi-sensor box.

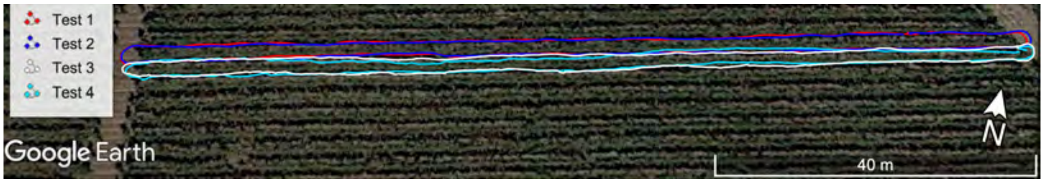


Figure 7: Google Earth view of the four paths estimated by EIF in a commercial vineyard, San Donaci, Apulia Region, Italy (40°27'16.2"N 17°54'30.6"E).

6. In-field Testing

The multi-sensor system is mounted and integrated on a tracked robot developed at the Politecnico of Bari, and it is tested in field conditions, as shown in Figure 6. Dedicated tests are performed in a commercial vineyard in San Donaci, Apulia region, Italy. Specifically, the robot is guided to follow closed-loop trajectories around different crop rows while gathering the sensor data. The data were then processed offline to recover the 6DoF path and the 3D map of the environment.
In this section, first, the localization performance of the proposed system is analyzed in terms of accuracy and repeatability. Then, the mapping results are discussed.

6.1. Localization performance

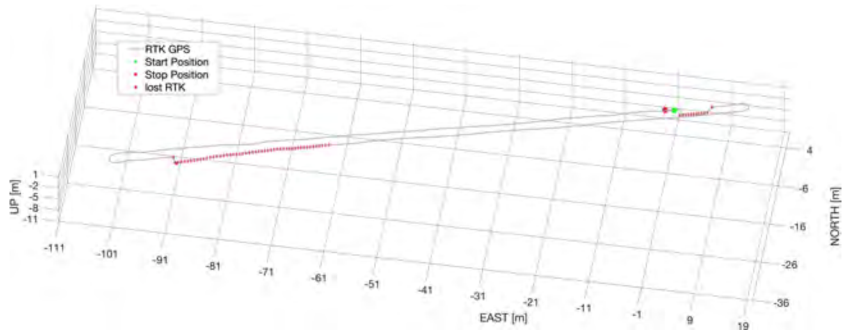
Four closed-loop runs are considered, performed along two different crop rows of about 120 m length, referred to as Test 1 to Test 4 in the following. They belong to two field campaigns carried out in September and October 2021, respectively, during different times of the day. Three localization sources are compared, namely RTK-GPS only, T265 only and EIF. A projection in Google Earth view of the trajectories reconstructed by EIF for the four paths is shown in Figure 7. Numerical results for all runs are collected in Table 1, showing the discrepancy in the East-North-Up (w) frame between the starting and ending points of the trajectory, expressed in terms of 3D Euclidean distance (D), 2D Euclidean distance in the motion plane (D_{EN}) and altitude distance (D_U), and the standard deviation of altitude measurements (σ_U) along the entire path.

The robot path as estimated by each localization source is reported in Figure 8 for Test 1. In this test, pose estimates using only RTK-GPS (Figure 8 (a)) are consistent as long as RTK correction is available. The starting and ending points are close to each other and the altitude estimate is stable

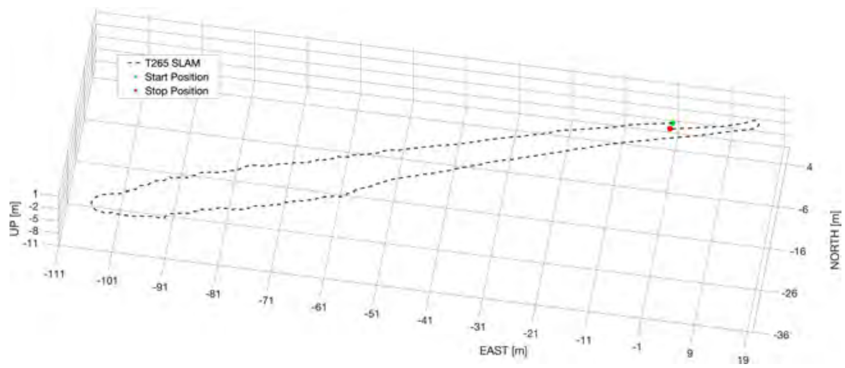
Table 1: Comparison of different localization sources along four robot paths (Test 1 to 4): discrepancy in the East-North-Up frame between the starting and ending points of the trajectory expressed in terms of 3D Euclidean distance (D), 2D Euclidean distance in the motion plane (D_{EN}), altitude distance (D_U), and standard deviation of altitude measurements (σ_U) along the entire path.

Test	*Source*	$D[m]$	$D_{EN}[m]$	$D_U[m]$	$\sigma_U[m]$
1	RTK-GPS	1.501	1.511	0.026	0.451
	T265	1.276	1.275	0.046	2.425
	EIF	1.514	1.513	0.032	0.228
2	RTK-GPS	0.579	0.579	0.013	0.452
	T265	5.909	4.921	3.270	2.199
	EIF	0.579	0.579	0.011	0.209
3	RTK-GPS	1.852	1.517	1.063	0.370
	T265	7.264	5.811	4.359	1.440
	EIF	1.833	1.484	1.077	0.355
4	RTK-GPS	1.290	1.073	0.717	0.670
	T265	5.341	4.760	2.424	1.497
	EIF	0.739	0.356	0.647	0.428

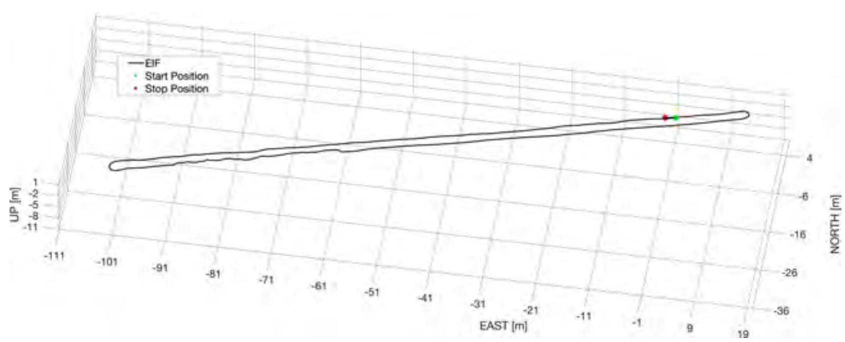
except when the connection to the base GPS is lost and the RTK correction is missing (red diamonds in Figure 8 (a)). Figure 8 (b) shows the path as reconstructed by the T265 proprietary visual-inertial SLAM algorithm in its own frame and successively transformed in the world frame. Compared to RTK-GPS path, the distance between starting and ending points estimated by the T265 camera is 15.6% smaller in terms of East-North coordinates (D_{EN}) but 77% larger in terms of altitude (D_U). Low accuracy of vertical displacement estimates leads to large standard deviation of altitude measurements (σ_U), which for T265 is of 2.43 m, about 5 times larger than the one obtained by RTK-GPS. Figure 8(c) shows the path reconstructed by the EIF. The EIF uses linear acceleration and angular velocity measures to make state predictions using the model described by equations (3), and corrects its predictions using measures of RTK-GPS (world position and velocity), IMU (world orientation) and T265 (relative position and orientation). The EIF estimates a difference between starting and ending points 0.18% larger than the RTK-GPS in terms of East-North position and 23% larger in terms of vertical displacement. However, the standard deviation of altitude measures



(a)



(b)



(c)

Figure 8: Localization results for Test 1: (a) from RTK-GPS only; (b) from T265 camera only; (c) after EIF fusion of RTK-GPS, IMU and T265 measurements. In (a), red diamonds are overlaid in two different zones without RTK coverage due to connection loss.

is 0.22 m for the EIF, i.e., 49% smaller than the one provided by RTK-GPS, suggesting an overall improvement in position estimate when fusing measurements with the EIF. This improvement is due to the fact that the RTK corrections are not available when connection is lost with the base GPS, whereas the EIF adjusts the position estimates for these instants using predictions with IMU data and short-term measures of the T265 camera when the covariance of position and velocity provided by the GPS grows.
When considering a second run along the same row (Test 2), similar results are obtained (see Figure 9 and the corresponding row in Table 1) in terms of σ_U which attests to 0.45 m for RTK-GPS, 2.20 m for T265 camera and 0.21 m for EIF. The discrepancy between starting and ending points (D) for T265 is higher than the one obtained from RTK-GPS indicating that visual inertial odometry should be only used for short-term displacement estimation. Again, the use of EIF allows for a reduction of σ_U of 53% with respect to GPS and of 90% with respect to T265, while preserving loop closure accuracy.
The localization results for a path along a different crop row (Test 3) is reported in Figure 10. In this case, the T265 results in a substantially degraded estimation, and EIF mainly relies on GPS leading to σ_U of 0.35 m. On the contrary, Figure 11 refers to an example where the quality of GPS signal is poor in several parts of the trajectory (Test 4). Again, EIF is able to compensate for the GPS outages mainly relying on T265 information showing better performance for all the metrics.

6.2. Mapping

For each geo-referenced position, the corresponding multi-view data can be recovered. As an example, Figure 12 shows the robot path (Test 1) overlaid over Google Earth view with three pinpointed positions, whereas the corresponding multi-view output is displayed in Figure 13.

Point clouds are collected and assembled in the w frame using estimates of both absolute position and orientation of the sensor box. In Figure 14, the EIF observer output is used to merge point clouds collected by the frontal camera. Figure 15(a) shows, instead, about 20 m of merged point clouds from all cameras using 6DoF odometry provided by the EIF. This map can be processed to extract high-level information about the crop, such as vegetation indexes and morphological information. As an example, Figures 15(b) and (c) show the map of Figure 15 (a) augmented with Green-Red Vegetation Index (GRVI) and crop elevation information, respectively.

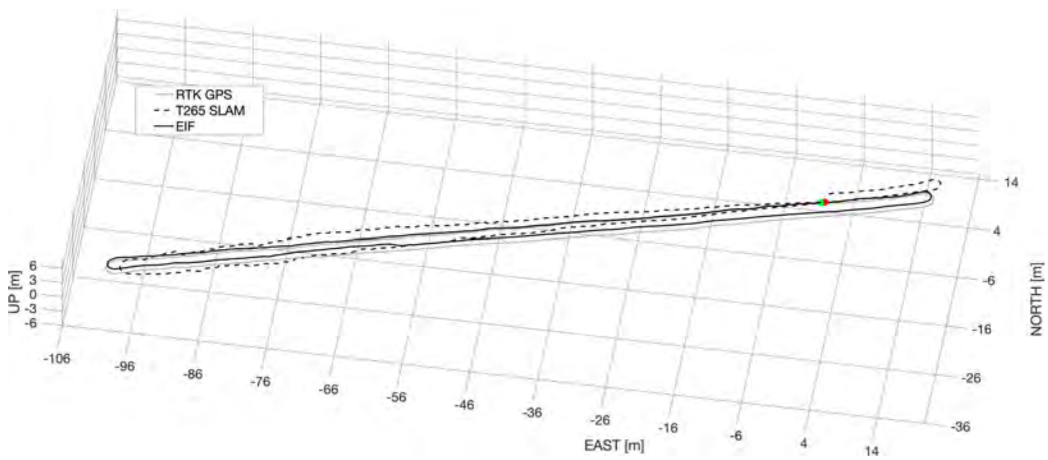


Figure 9: Localization results for a second run along the same path of Figure 8 (Test 2) from RTK-GPS only (solid grey line), T265 only (dashed black line) and EIF (solid black line). Start and stop positions for EIF trajectory are denoted by green and red dot, respectively.

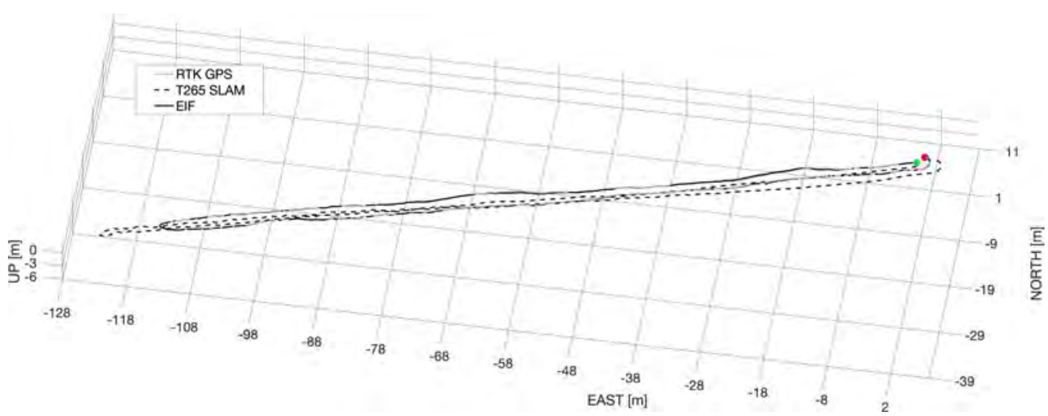


Figure 10: Localization results for Test 3 from RTK-GPS only (solid grey line), T265 only (dashed black line) and EIF (solid black line). Start and stop positions for EIF trajectory are denoted by green and red dot, respectively. In this test, the T265 estimate is substantially degraded and the EIF mainly relies on GPS.

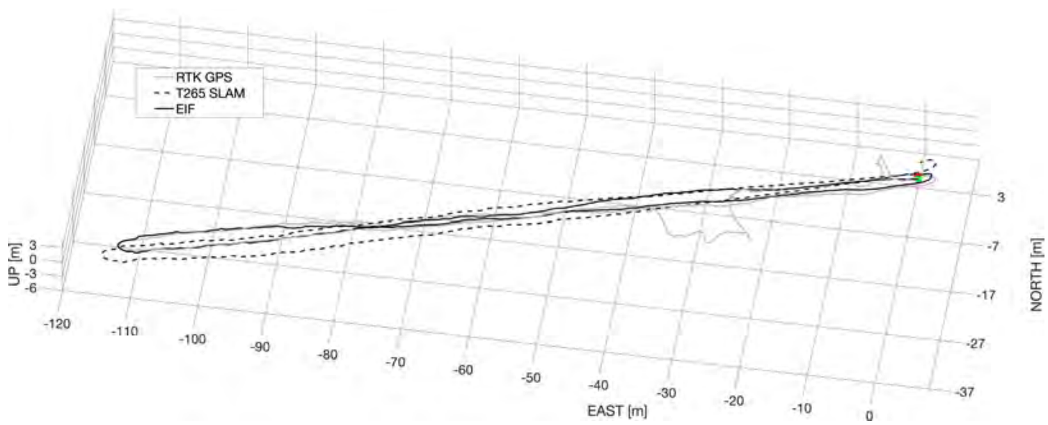


Figure 11: Localization results for Test 4 from RTK-GPS only (solid grey line), T265 only (dashed black line) and EIF (solid black line). Start and stop positions for EIF trajectory are denoted by green and red dot, respectively. In this test, EIF is able to compensate poor GPS signal quality based on T265 information.

Figure 12: EIF-derived path overlaid on Google Earth view for Test 1. Three successive positions of the robot are pinpointed. For these positions, the corresponding visual data are shown in Figure 13.

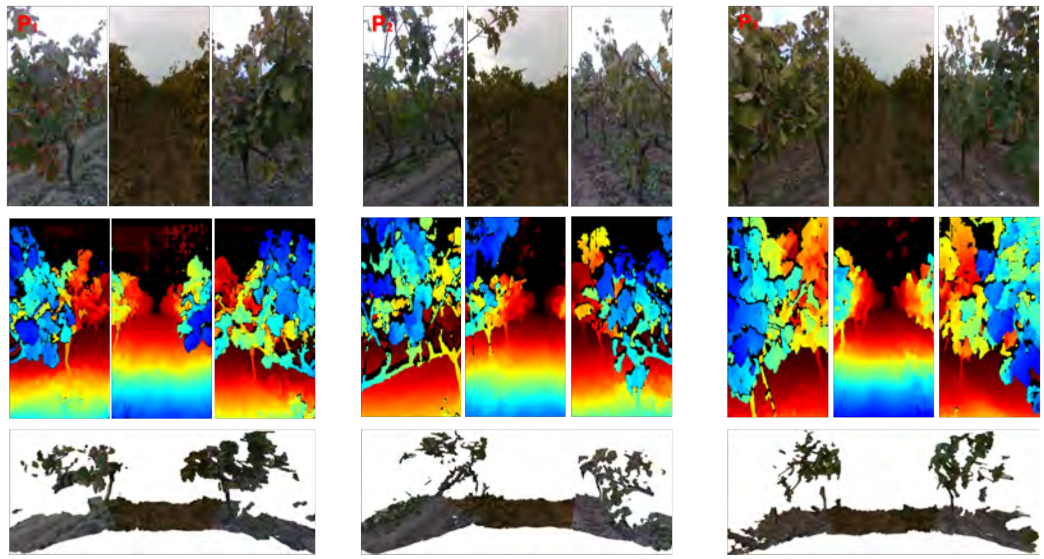
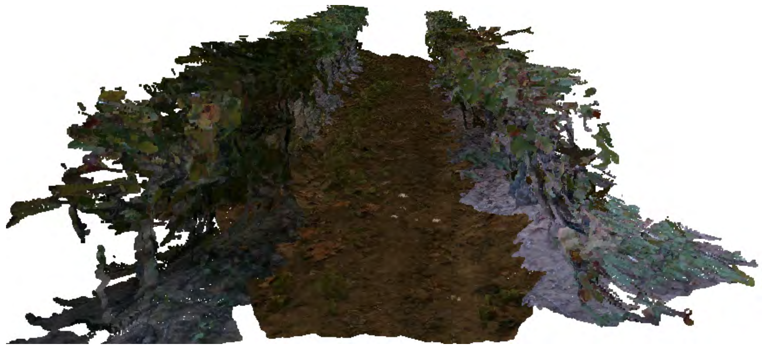


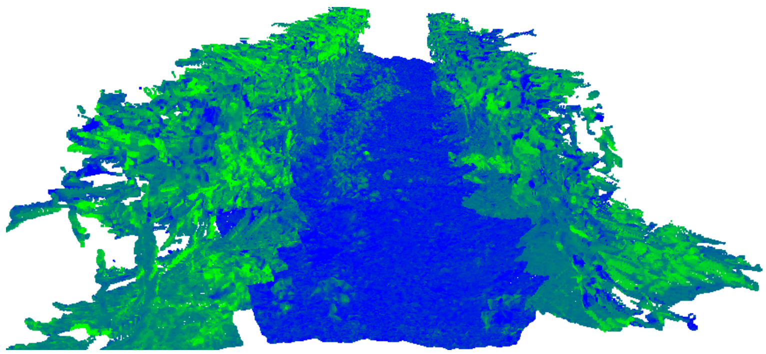
Figure 13: Output of the multi-view camera system for three robot locations along the path (Test 1): (first row) color images, (second row) depth images obtained from IR stereo reconstruction and (third row) multi-view 3D point cloud.

Figure 14: Mapping results (Test 1): upper view of the terrain map reconstructed by the central camera. The robot trajectory estimated by the sensor fusion approach is also overlaid.

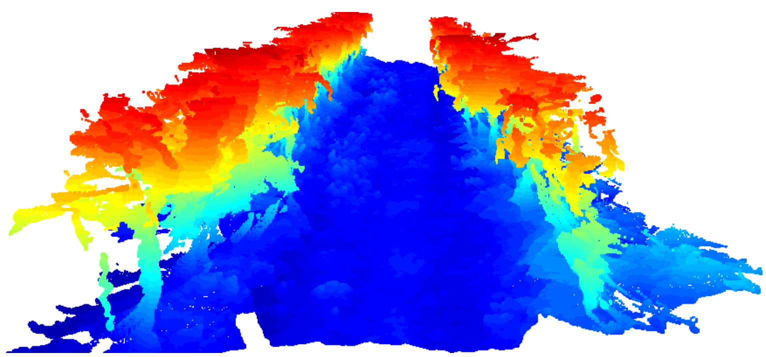
Accurate localization information is essential to assemble subsequent point clouds acquired by the D435 cameras and build the environment map. This can be clearly seen in Figure 16, where two different 6DoF localization sources are compared. In detail, Figure 16(a) shows a group of point clouds badly assembled with synced data of GPS for position and IMU for orientation when RTK correction are missing. Figure 16(b) is obtained using the EIF for the same time span, clearly showing the improvement in point cloud assembling.



(a)

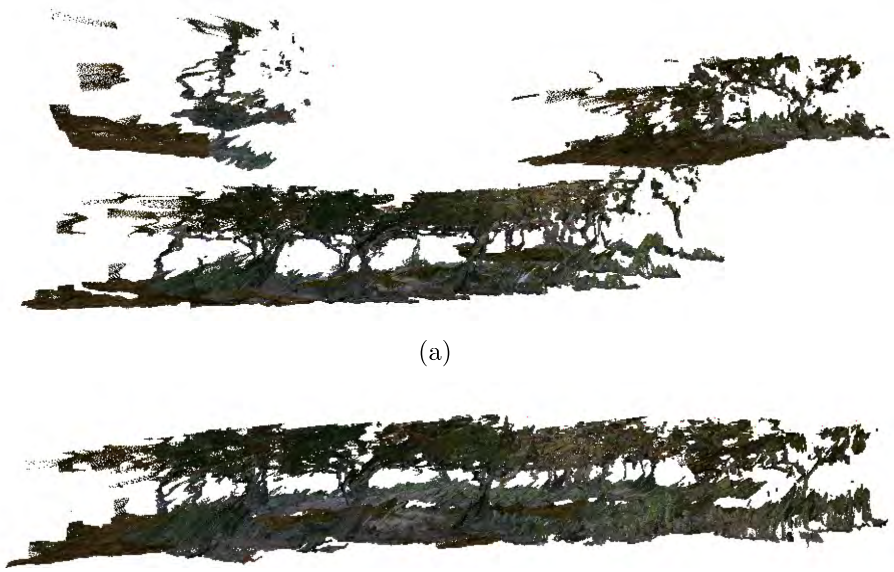


(b)



(c)

Figure 15: Closeup of the multi-view map for Test 1 (first 20 m): (a) RGB, (b) GRVI and (c) elevation map. In (b), green points refer to vegetation, whereas blue points correspond to non-vegetated parts. Lighter green denotes higher GRVI values. In (c), a jet colormap is used to represent point height with respect to ground.



(a)

(b)

Figure 16: Mapping results (Test 1): closeup of loop closure before (a) and after (b) EIF correction.

7. Conclusions

In this paper, the development, implementation and testing of a multi-view RGB-D sensing device is presented. The system is intended to be mounted on an agricultural ground robot for in-field proximal monitoring of high-value crops. A multi-view mapping approach to combine information from multiple visual and localization sensors and produce a high-resolution 3D reconstruction of agricultural environments is described. It is based on an EIF algorithm to fuse information from RTK-GPS, IMU and visual-inertial SLAM to obtain an accurate estimation of the vehicle position in the field. Then, on the basis of localization data, subsequent point clouds reconstructed by the RGB-D sensors during robot motion can be assembled to generate a high-resolution map of the surveyed environment. Results of dedicated tests performed in a commercial vineyard are presented, showing the effectiveness of the proposed system for in-field data gathering in an automatic and non-invasive way.
Future work will include the processing of the maps using supervised or un-supervised classification methods to generate semantic representations of the

environment, which can be used to improve vehicle autonomy and safety. Research will focus on the integration of output maps into Farm Management Information Systems (FMIS) to enable map-based control of agricultural applications and machinery. In this respect, future efforts will be devoted to address the real-time challenge by using the multi-view maps for online navigation of autonomous agricultural vehicles. Furthermore, methods for identification and mapping of anomalies, such as weeds, as well as the extraction of geometric measurements, such as plant volume/height estimates, will be integrated to enable precision farming practices. This would also improve the cost-benefit ratio of the sensor suite.

Acknowledgments

This work was funded by the following research programs: European Union's Horizon 2020 research and innovation programme under grant agreement "ATLAS - Agricultural Interoperability and Analysis System" (No. 857125); ERA-NET ICT-AGRI-FOOD COFUND "ANTONIO-multimodal sensing for individual plANT phenOtyping in agriculture robotics" (No. 41946); "E-crops - Technologies for Digital and Sustainable Agriculture" funded by the Italian Ministry of University and Research (MUR) under the PON Agrifood Program (No. ARS01_01136); CNR DIITET project "DIT.AD022.180 Transizione industriale e resilienza delle Società post-Covid19 (FOE 2020)", sub task activity "Agro-Sensing". The authors are grateful to Cantina San Donaci agricultural farm for hosting experimental tests. The administrative and technical support by Michele Attolico and Giuseppe Bono is also gratefully acknowledged.

Disclaimer: The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the EASME nor the European Commission is responsible for any use that may be made of the information contained therein.

Author contributions

Fabio Vulpi: Conceptualization, Methodology, Software, Validation, Writing-Original draft; **Roberto Marani**: Software, Writing-Original draft;

Antonio Petitti: Software, Writing-Original draft; **Giulio Reina** and **Annalisa Milella**: Conceptualization, Methodology, Software, Validation, Writing-Original draft, Writing- Reviewing and Editing, Supervision.

Appendix

Quaternion representations are convenient for composition of rotations and coordinate transformations. The unit quaternion $\mathbf{q} = [q_x, q_y, q_z, q_w]$, is uniquely mapped to a rotation matrix R and describes the transformation between two reference frames as a rotation of a certain angle θ around the direction vector $\vec{n}$ following Equation A.1.

$$\mathbf{q} = [n_x sin\frac{\theta}{2}, n_y sin\frac{\theta}{2}, n_z sin\frac{\theta}{2}, cos\frac{\theta}{2}] \tag{A.1}$$

Denoting with s_1 and $\vec{v}_1$ respectively the scalar and vector part of quaternion $\mathbf{q}_1 = [\vec{v}_1, s_1] = [v_{1x}, v_{1y}, v_{1z}, s_1]$, and with s_2,$\vec{v}_2$ the scalar and vector part of quaternion $\mathbf{q}_2$, the operation of quaternion product can be expressed as

$$\mathbf{q}_2\mathbf{q}_1 = [s_2\vec{v}_1 + s_1\vec{v}_2 + \vec{v}_2 \times \vec{v}_1, s_2 s_1 - \vec{v}_2 \cdot \vec{v}_1] \tag{A.2}$$

where quaternion product symbol has been omitted for readability, whereas $\vec{v}_2 \cdot \vec{v}_1$ denotes dot product between vectors $\vec{v}_1$ and $\vec{v}_2$ and finally $\vec{v}_2 \times \vec{v}_1$ denotes cross product between the two vectors. Quaternion product is a non-commutative operation and returns a quaternion that represents the orientation obtained after the sequence of transformations $\mathbf{q}_1$ and then $\mathbf{q}_2$. The norm of a quaternion $\mathbf{q}$ is denoted as $\|\mathbf{q}\|^2 = q_x^2 + q_y^2 + q_z^2 + q_w^2$. The conjugate of quaternion $\mathbf{q} = [\vec{v}, s]$ is represented as $\mathbf{q}^* = [-\vec{v}, s]$, while its inverse $\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\sqrt{\|\mathbf{q}\|^2}}$. For a unit quaternion we have $\|\mathbf{q}\|^2 = 1$ so its conjugate coincides with its inverse. All quaternions describing orientation in 3-D space are unit quaternions. The normalized quaternion denoted as $\|\mathbf{q}\| = \frac{\mathbf{q}}{\sqrt{\|\mathbf{q}\|^2}}$ has a unit norm and each of its components are divided by $\sqrt{\|\mathbf{q}\|^2}$. Let us denote with $\mathbf{q}_{B-A}$ the quaternion describing orientation of frame A with respect to frame B written in frame B, its inverse is $\mathbf{q}_{B-A}^{-1} = \mathbf{q}_{A-B}$. Then, composition of rotations can be obtained in a convenient form as

$$\mathbf{q}_{C-A} = \mathbf{q}_{C-B}\mathbf{q}_{B-A} \tag{A.3}$$

Consider the position vector $\vec{p}_A$ in frame A, then its projection in reference frame B can be obtained as

$$[\vec{p}_B, 0] = \mathbf{q}_{B-A}[\vec{p}_A, 0]\mathbf{q}_{A-B} \tag{A.4}$$

Finally, denoting with $\vec{\omega}_B(t)$ the angular velocity of moving frame B in its reference frame A, the derivative of the quaternion $\mathbf{q}_{A-B}(t)$ expressed in the inertial frame A can be computed as

$$\frac{d\mathbf{q}_{A-B}(t)}{dt} = \frac{1}{2}\mathbf{q}_{A-B}(t)[\vec{\omega}_B(t), 0] \tag{A.5}$$

References

[1] G. Reina, A. Milella, R. Rouveure, M. Nielsen, R. Worst, M. R. Blas, Ambient awareness for agricultural robotic vehicles, Biosystems Engineering 146 (2016) 114–132, special Issue: Advances in Robotic Agriculture for Crops. doi:https://doi.org/10.1016/j.biosystemseng.2015.12.010.

[2] Gazebo simulator, `http://gazebosim.org/`, accessed: 2021-12-06.

[3] A. Matese, P. Toscano, S. F. Di Gennaro, L. Genesio, F. P. Vaccari, J. Primicerio, C. Belli, A. Zaldei, R. Bianconi, B. Gioli, Intercomparison of UAV, aircraft and satellite remote sensing platforms for precision viticulture, Remote Sensing 7 (3) (2015) 2971–2990.

[4] L. Comba, A. Biglia, D. Ricauda Aimonino, P. Gay, Unsupervised detection of vineyards by 3d point-cloud uav photogrammetry for precision agriculture, Computers and Electronics in Agriculture 155 (2018) 84–95.

[5] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, K. Lewis, Sensors and systems for fruit detection and localization: A review, Computers and Electronics in Agriculture 116 (2015) 8–19. doi:https://doi.org/10.1016/j.compag.2015.05.021.

[6] Z. Li, R. Guo, M. Li, Y. Chen, G. Li, A review of computer vision technologies for plant phenotyping, Computers and Electronics in Agriculture 176 (2020) 105672. doi:https://doi.org/10.1016/j.compag.2020.105672.

[7] J. Ma, K. Du, L. Zhang, F. Zheng, J. Chu, Z. yu Sun, A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing, Comput. Electron. Agric. 142 (2017) 110–117.

[8] P. Jiang, Y. Chen, B. Liu, D. He, C. Liang, Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks, IEEE Access 7 (2019) 59069–59080. doi:10.1109/ACCESS.2019.2914929.

[9] J. Mack, C. Lenz, J. Teutrine, V. Steinhage, High-precision 3d detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning, Computers and Electronics in Agriculture 135 (2017) 300–311.

[10] A. Milella, R. Marani, A. Petitti, G. Reina, In-field high throughput grapevine phenotyping with a consumer-grade depth camera, Computers and Electronics in Agriculture 156 (2019) 293–306.

[11] A. Milella, G. Reina, M. Nielsen, A multi-sensor robotic platform for ground mapping and estimation beyond the visible spectrum, Precision Agric 20 (2019).

[12] A. Milella, G. Reina, 3D reconstruction and classification of natural environments by an autonomous vehicle using multi-baseline stereo, Intel Serv Robotics 7 (2014) 79–92.

[13] Y. Chéné, D. Rousseau, P. Lucidarme, J. Bertheloot, V. Caffier, P. Morel, É. Belin, F. Chapeau-Blondeau, On the use of depth camera for 3D phenotyping of entire plants, Computers and Electronics in Agriculture 82 (2012) 122–127.

[14] R. P. Devanna, A. Milella, R. Marani, S. P. Garofalo, G. A. Vivaldi, S. Pascuzzi, R. Galati, G. Reina, In-field automatic identification of pomegranates using a farmer robot, Sensors 22 (15) (2022).

[15] I. C. Condotta, T. M. Brown-Brandl, S. K. Pitla, J. P. Stinn, K. O. Silva-Miranda, Evaluation of low-cost depth cameras for agricultural applications, Computers and Electronics in Agriculture 173 (2020) 105394.

[16] M. Imperoli, C. Potena, D. Nardi, G. Grisetti, A. Pretto, An effective multi-cue positioning system for agricultural robotics, IEEE Robotics and Automation Letters 3 (4) (2018) 3685–3692. doi:10.1109/LRA.2018.2855052.

[17] N. Habibie, A. M. Nugraha, A. Z. Anshori, M. A. Ma'sum, W. Jatmiko, Fruit mapping mobile robot on simulated agricultural area in Gazebo simulator using simultaneous localization and mapping (SLAM), in: 2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS), 2017, pp. 1–7. doi:10.1109/MHS.2017.8305235.

[18] J. A. Hage, S. Mafrica, M. E. B. E. Najjar, F. Ruffier, Informational framework for minimalistic visual odometry on outdoor robot, IEEE Transactions on Instrumentation and Measurement 68 (8) (2019) 2988–2995.

[19] W. Xu, Y. Cai, D. He, J. Lin, F. Zhang, Fast-lio2: Fast direct lidar-inertial odometry, IEEE Transactions on Robotics (2022).

[20] I. Ullah, X. Su, J. Zhu, X. Zhang, D. Choi, Z. Hou, Evaluation of localization by extended Kalman filter, unscented Kalman filter, and particle filter-based techniques, Wireless Communications and Mobile Computing 2020 (2020) 1–15.

[21] G. Reina, A. Leanza, G. Mantriota, Model-based observers for vehicle dynamics and tyre force prediction, Vehicle System Dynamics (2021) 1–26.

[22] A. Petitti, F. Vulpi, R. Marani, A. Milella, A self-calibration approach for multi-view RGB-D sensing, in: E. Stella (Ed.), Multimodal Sensing and Artificial Intelligence: Technologies and Applications II, Vol. 11785, International Society for Optics and Photonics, SPIE, 2021, pp. 50 – 55. doi:10.1117/12.2595165.

[23] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, Principles of Robot Motion, The MIT Press, Cambridge, MA, USA, 2004.

[24] M. Corsini, P. Cignoni, R. Scopigno, Efficient and flexible sampling with blue noise properties of triangular meshes, IEEE Transac-

tions on Visualization and Computer Graphics 18 (6) (2012) 914–924. doi:10.1109/TVCG.2012.34.

[25] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The ball-pivoting algorithm for surface reconstruction, IEEE Transactions on Visualization and Computer Graphics 5 (4) (1999) 349–359. doi:10.1109/2945.817351.