Towards a Low-Cost Robot Navigation Approach based on a RGB-D Sensor Network

Massimiliano Bertoni
Dept. of Management and Engineering
University of Padova
Vicenza, Italy
massimiliano.bertoni@phd.unipd.it

Stefano Michieletto
Dept. of Management and Engineering
University of Padova
Vicenza, Italy
stefano.michieletto@unipd.it

Giulia Michieletto
Dept. of Management and Engineering
University of Padova
Vicenza, Italy
giulia.michieletto@unipd.it

***Abstract*—Motivated by the recent interest in the modernization of the traditional manufacturing facilities, this work focuses on the development of an automated warehouse where mobile manipulators autonomously navigate in the environment by exploiting additional spatial information gathered by a network of fixed RGB-D sensors. In detail, a navigation approach is proposed wherein both the mobile and fixed nodes composing the system are low-cost devices characterized by limited sensing and computing capabilities. A ROS2-based proof of concept of the outlined scenario is discussed and investigated with special regard to the practical challenges and the possible limitations. The results of some preliminary tests are reported to provide an intuition on the feasibility and potentiality of the designed approach in real-world industrial scenarios.**

***Index Terms*—navigation, sensor network, mobile robot**

I. Introduction

Taking advantage of the IoT paradigm, the innovation spirit permeating the modern manufacturing plants encourages the exploitation of cooperating cyber-physical systems. These are composed of physically and/or virtually interacting heterogeneous devices aiming at optimizing the whole production process [1]. In detail, one of the nine pillars of the Industry 4.0 envisages the use of autonomous robots as a pervasive and ubiquitous technology in the emerging smart factories. Unmanned ground and/or aerial vehicles, thus, turn out to be a powerful resource to accomplish monitoring, inspection, and transporting tasks, acting either as autonomous individual units or cooperative multi-agent groups (see, e.g., [2]–[4]).

In this increasingly automated context, safe and efficient robot navigation becomes a mandatory problem to face. To guarantee high-performance path planning and successful obstacle detection and avoidance, two possible strategies are generally adopted: to equip the autonomous robots with high-level computational capabilities and sensing devices or to exploit external auxiliary sensing and computing systems.

An example of the former strategy is the BigDog, a rough-terrain quadruped robot whose navigation system uses a combination of planar laser scans, stereo vision, and proprioceptive sensing [5]. Similarly, in [6], the indoor localization and navigation of a ground robot is achieved through the use of an onboard Lidar and RGB-D camera. A low-cost autonomous mobile robot system, relying only on a RGB-D camera and a low-end SBC, is proposed in [7]. In particular, a large class of navigation techniques restoring on information-rich sensor systems (e.g., cameras, laser range finders, ultrasound sensors) exploits additional visual markers, as graphical landmarks [8], QR codes [9] or light patterns [10].

A robot navigation strategy based on an auxiliary sensing system is, instead, presented in [11], involving a range sensor network required to cope both with the obstacle detection and trajectory planning for a low-level path tracker robot. Along the same line, different approaches for indoor navigation are described in [12], which rely on the exploitation of a wireless sensor network, avoiding the use of additional sensors onboard. More recently, a novel indoor navigation solution has been presented in [13] based on multiple WiFi signals.

When dealing with high-performing robots in terms of both sensing and computing capabilities, their navigation is facilitated by the rapid and efficient processing of the recorded environmental data. Contrarily, besides potentially communication issues, the approach based on any auxiliary system entails gathering more spatial information, and, thus optimizing the robot path planning also limiting the computational burden.

In light of these facts, this work focuses on the navigation of mobile manipulators within an industrial warehouse equipped with an RGB-D sensor network. As compared to the existing literature, the outlined approach envisages a multi-element scenario wherein both the mobile robot and the auxiliary network are characterized by the same limited sensing capability and comparable modest computational performance. The proposed solution, thus, involves the exploitation of interacting low-cost devices. In this work, we outline an effective proof of concept of the mentioned navigation approach, especially devising a suitable ROS2 architecture to handle the sharing of spatial information gathered by the different devices.

The rest of the paper is organized as follows. The considered industrial application scenario is detailed in Section II. Section III provides an insight into the outlined multi-node network. In Section IV the attention is focused on the designed proof of concept and some preliminary results are reported. In Section V the principal challenging aspects and open design choices affecting the proposed approach are discussed. Finally, some concluding remarks are given in Section VI.

This work was partly supported by the CAR DTG-SEED 2020 project.

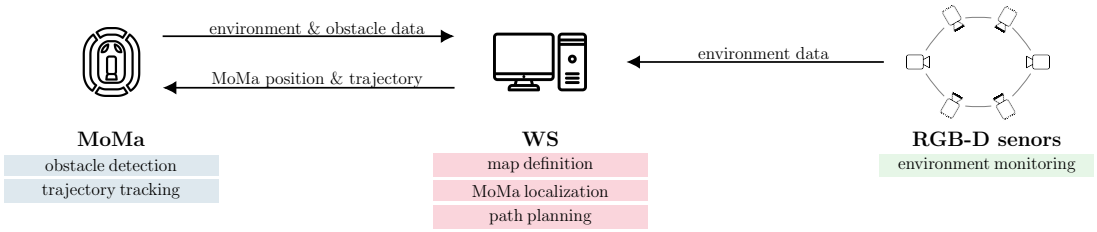

Fig. 1: Schematic representation of the elements (MoMa - Mobile Manipulator, WS - Working Station, RGB-D sensors) involved in the proposed proof of concept, together with the information they are required to exchange (arrows) and the tasks they are charged to accomplish (colored blocks).

II. Application Scenario

In a broad sense, this work focuses on the development of an effective strategy to automatize the management of a manufacturing warehouse within the smart factories context. In this direction, a cutting-edge trend consists in completely demanding the items transportation and pick-and-place operations to a single or a group of autonomous mobile manipulators (MoMas). These are generally constituted of a mobile basis and a manipulator having arbitrary degrees of freedom (dofs), and endowed with a sensing system and a computational unit.

Despite their peculiar features which can lead to different algorithms implementation, the aforementioned MoMas are required *to navigate* in a structured and potentially cluttered environment. In this challenging scenario, facing any robot navigation issue implies coping with

a. the *environment monitoring*, to extract significant spatial information, as, for instance, changes in the warehouse fixed structures configuration;
b. the *obstacle detection*, interpreted as the identification of dynamic intruders and/or temporary environment modifications (e.g., set down packages);
c. the *map definition*, namely the determination of the (time-varying) occupancy grid of the environment conditional on the scene monitoring and obstacles identification;
d. the *robot localization*, i.e., the estimation of the MoMa position with respect to a global fixed frame;
e. the *path planning*, consisting in the computation of the optimal MoMa pathway from a given starting point to a target location according to the current warehouse structure and the presence of obstacles, accounting also for the robot kinematic and dynamic constraints;
f. the *trajectory tracking*, understood as the minimization of the error between the real MoMa position and the desired one output from the path planning at each time instant.

Given these premises, in the rest of the work, we focus the attention on a single MoMa[1] (hereafter, referred also as *mobile node*). The idea is to fulfill all the mentioned tasks mainly resting on the suitable information sharing among the MoMa itself and a network of RGB-D sensors (hereafter, referred also as *fixed nodes*) ad-hoc spread in the environment. In detail, in our experimental setup, the sharing process of the valuable information is assumed to be managed by a working station (WS) that is also required to solve the robot localization and path planning task, thus acting as *central node*. The obstacle detection is, instead, autonomously accomplished by the MoMa, while the map is updated according to the data gathered by the fixed nodes. Figure 1 aims at clarifying the tasks partition among the system components and the information shared between the nodes.

[1]Note that the extension to multiple MoMas acting as a single agent can be performed, for instance, by considering the other robots as dynamic obstacles.

III. Multi-Node System Description

In this section, we provide a complete description of the multi-node system at the core of the proposed proof of concept. In doing this, the attention is focused on highlighting both the high-level properties and the specific technical characteristics of the facilities employed in the preliminary tests.

A. Mobile Manipulator

As already mentioned in Section II, the considered MoMa is an autonomous vehicle composed of four main parts:

- the *actuation unit*, i.e., an unmanned mobile basis;
- the *manipulation unit*, corresponding to an articulated robotic arm acting as the kinematic chain for the end-effector;
- the *sensing system*, usually involving heterogeneous and complementary sensors;
- the *computational unit*, namely a computer devoted to the robot control and decision-making process.

The MoMa adopted in our experimental setup is the LoCoBot (Low-Cost roBot) depicted in Figure 2. This includes a Kobuki base YMR-K101-W1 as actuation unit and a 5 dofs Interobotix WidowX 200 as manipulation unit. The computation unit consists in a mini PC Intel NUC7i5BNH mounting an Intel Core i5-7260U 2.2 GHz (4M cache) processor, 8 Gb DDR4 RAM, and a Wireless-AC 8265 antenna. Finally, the sensing system comprehends a 3-axes gyroscope and some edge detection and bumper sensors, in addition to an Intel RealSense Depth D435 camera including an IR projector and an RGB color sensor. Based on the released data-sheet [14], [15], the used MoMa is also characterized by the features reported in Table I as regards the actuation unit and sensing system. Observe that LoCoBot performance is limited by the low-cost budget: this motivates its adoption in the considered experimental setup.

Fig. 2: LoCoBot
(MoMa - mobile node)

Fig. 3: RPi 4B + Intel D435 camera
(RGB-D sensor - fixed node)

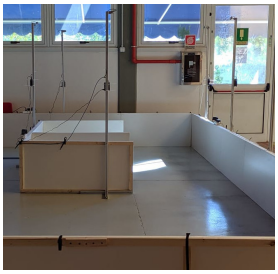

Fig. 4: RGB-D sensor network
in the experimental setup

actuation unit	max linear speed	$70\,\mathrm{cm/s}$
	max angular speed	$180\,^\circ/\mathrm{s}$
sensing system	gyro meas. range	$\pm 250, \pm 500, \pm 2000\,\mathrm{dps}$
	gyro sensibility	$8.75, 17.50, 70\,\mathrm{mdps/dig}$
	min depth dist. at max res.	$28\,\mathrm{cm}$
	depth accuracy	$< 2\%$ at $2\,\mathrm{m}$
	depth FOV	$87\,^\circ \times 58\,^\circ$
	depth output resolution	up to 1280×720
	depth frame rate	up to $90\,\mathrm{fps}$
	RGB frame resolution	1920×1080
	RGB frame rate	$30\,\mathrm{fps}$
	RGB sensor FOV	$6^\circ \times 42^\circ$
	RGB sensor resolution	$2\,\mathrm{MP}$

TABLE I: LoCoBot actuation unit and sensing system features

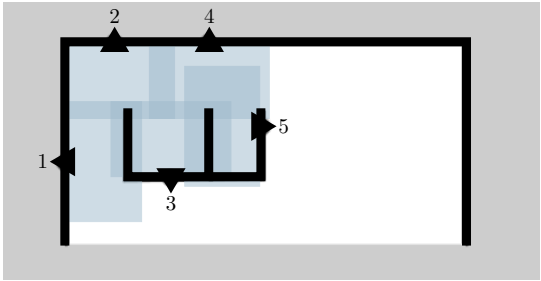

Fig. 5: Experimental setup configuration
(each black triangle represents a RGB-D sensor)

B. RGB-D sensors

The fixed nodes of the considered system consist in RGB-D sensors, i.e., compact devices composed of two parts:

- a *computational unit*, namely a microprocessor;
- a *sensing unit*, subsisting in an RGB camera and an infrared module to recover depth information.

As shown in Figure 3, the computational unit of the RGB-D sensors used in our experimental benchmark consists in a Raspberry Pi 4B held in an aluminum case provided with heat dissipation fans; while the sensing unit is made by an Intel RealSense Depth D435 camera. Note that the employed camera is the same mounted on the LoCoBot to guarantee coherence in terms of system nodes sensing performance and reduce the computational effort of the system. Moreover, the camera is highly configurable providing a large set of resolutions and framerates (II) being versatile for industrial purposes. On the other hand, the choice of using Raspberry Pi 4 is motivated by the fact that this device stands out for its small size ($56 \times 85 \times 15\,\mathrm{mm}$) and low-power consumption ($3.8 - 4.0\,\mathrm{W}$), despite its affordable price ($\leq 100\$$) [16].

Figure 4 illustrates the entire experimental setup; the RGB-D sensors are located on the top of aluminum bars at a height of $2\,\mathrm{m}$ and the whole fixed nodes network is made up of five devices placed as in Figure 5 where it is represented (in black) also the perimeter of the used extruded polystyrene structure covering a $3.75\,\mathrm{m} \times 7.5\,\mathrm{m}$ area. These designed choices ensure that the region covered by each camera is approximate $2.5\,\mathrm{m} \times 1.5\,\mathrm{m}$ (light blue rectangles in Figure 5), hence the entire monitored area measures approximately $10\,\mathrm{mq}$.

C. Working Station

In the considered multi-node system, a WS acts as a central unit managing the data recorded by both the mobile and the fixed nodes (incoming arrows in Figure 1) and providing to the MoMa the information about both its position and the trajectory to follow (outgoing arrow in Figure 1). We emphasize that the aforementioned data streams can generally differ in terms of frequency: the RGB-D sensors can send information at longer time intervals with respect to the MoMa mainly because the data acquired by the mobile node are necessary to its localization in the environment and can be partially computed on-board to reduce the network workload. In addition, the MoMa is also tasked to detect and avoid dynamic obstacles, thus collecting real-time environment information.

In our experimental setup, the WS is an industrial PC equipped with an Intel Core i5-4210M CPU $2.60\,\mathrm{GHz}$ (4M cache) processor, $16\,\mathrm{Gb}$ DDR4 RAM, and Gigabit Ethernet interface. The choice of developing a centralized architecture resting on a WS as central node is motivated by the intent of proposing a low-cost strategy for manufacturing warehouse management which turns out to be also easy to implement in existing facilities using standard industrial infrastructures and protocols. We finally underline that, in our benchmark, all the fixed nodes are connected to the WS by using cabled Ethernet communication links, while the MoMa uses a WiFi connection. This solution aims at minimizing possible problems due to network congestion towards the central node by adopting a dedicated local network composed of an 8 port switch and a WiFi hotspot.

IV. Navigation Approach

This section aim at clarifying both the logical and practical aspects of the proposed proof of concept. In doing this, we report also the results of some preliminary experimental tests.

A. General Overview

The proposed proof of concept rests on the use of ROS2 as middleware to manage both the single system component actions and the interplay among the different nodes. While ROS1 enables a standardized framework for robots, the new version focuses on contemporary state-of-the-art technologies in robotics like distributed processing, real-time control, multi-device applications, poor-quality communication, embedded computing, system scalability, and production deployment. ROS2 represents a challenging design choice since the transition is still in progress: a limited number of packages is available and the documentation is usually poor. Some parts of the software API is still at an early stage and changes between one version and another lead to broken pipelines and minor issues requiring original contributions and time-consuming debugging sessions. However, we decide to use ROS2 to cope with the long term objectives of the project that are not restricted to the design used for this proof of concept.

The main modules composing the designed ROS2 architecture are described in the following.

- *RGB-D camera driver*. The driver provides the interface to the data collected by the RGB-D sensors. Resolution and framerate can be set independently for each camera and also for both the RGB and the Depth components. The ranges for the adopted Intel Realsense D435 cameras are listed in Table II. Intrinsic and extrinsic matrices can be obtained from this package.
- *Robot description*. The module characterizes the robotic platform in terms of physics, volumes, links, and joints. The space occupied by the robot can change depending on the actuators. The package guarantees a consistent transformation between the different points of interest in the robot, like sensor sources of data, hand-eye calibration, and end-effector pose.
- *Base controller*. The package manages the mobile base motion using as input the robot linear and angular velocities. It translates coherently these velocities referred to the robot as a whole to the speed information related to left and right wheels. The package computes also the odometry of the robot by using the integrated IMU.
- *Arm and camera controller*. This controller is in charge of monitoring and moving arm joints and camera pan and tilt. The information about the current state is periodically spread to the Robot Description module to update the position accordingly. In this work, the role of this part is limited to keeping the arm in a resting position where the interference with the environment is minimal. The camera pan and tilt are fixed too, with the sensor looking right in front of the robot, parallel to the ground floor.
- *Laser scan detection*. The module converts the depth images produced by the RGB-D sensors to a coherent 2D laser scan referred to the origin of the map, with a height of 0.5 m. The information coming from the robot is used to generate a local map for localization and obstacle avoidance purposes.
- *Localization*. The package manages the robot localization inside the map generated using the fixed node laser scans as a quasi-static map. The local scan and the odometry of the robot are then coupled together to estimate the robot pose through an Adaptive Monte-Carlo Localizer technique based on a particle filter.
- *Navigation*. The module is composed of three parts: the planner, the controller, and the recovery. The planner aims at computing the shortest path from the current position to a goal based on the occupancy of the map. The controller follows the path generated by the planner by using the local map to avoid previously un-detected obstacles. The recovery covers failures or non-managed situations.
- *Visualization and monitoring*. The package consists of a set of visual tools and monitoring procedures allowing us to set parameters, test system subparts, and easily identify issues when necessary.

	format	resolution	frame rate (fps)
depth	Z [16bit]	1280 × 720	6, 15, 30
		848 × 480	6, 15, 30, 60, 90
		640 × 480	6, 15, 30, 60, 90
		640 × 360	6, 15, 30, 60, 90
		480 × 270	6, 15, 30, 60, 90
		424 × 240	6, 15, 30, 60, 90
RGB	YUY [16bit]	1920 × 1080	6, 15, 30
		1280 × 720	6, 15, 30
		960 × 540	6, 15, 30, 60
		848 × 480	6, 15, 30, 60
		640 × 480	6, 15, 30, 60
		640 × 360	6, 15, 30, 60
		424 × 240	6, 15, 30, 60
		320 × 240	6, 30, 60
		320 × 180	6, 30, 60

TABLE II: Intel Realsense D435 camera specifications

B. Preliminary Tests

To provide a first intuition on the feasibility and validity of the proposed navigation approach, in the following we present the results of some preliminary tests conducted in the experimental scenario depicted in Figure 4. We remark that the intent is not to explore the outlined solution performance, rather to verify its actual practicability in a real-world scenario.

Figure 6 reports the RGB (upper row) and the Depth (lower row) images acquired by the cameras constituting the fixed node network displaced as in Figure 5. For these devices, the framerate is fixed to $15\,\mathrm{fps}$, while the resolution is set to 424×240 for both the RGB and Depth parts. Regarding the MoMa, the resolution and the framerate of the camera mounted on the LoCoBot are both settled to 848×480 at $30\,\mathrm{fps}$. We underline that the mobile platform camera is characterized by higher resolution and framerate to better cope with local changes in the scene, whereas all the fixed sensors work are featured by the same images size and rate to ease the data merging process.

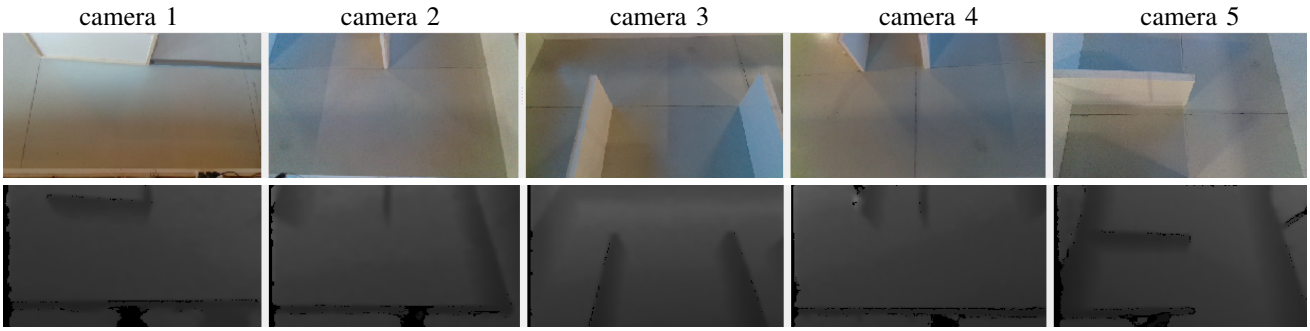

Fig. 6: RGB and Depth images acquired by the cameras composing the fixed node network depicted in Figure 5.

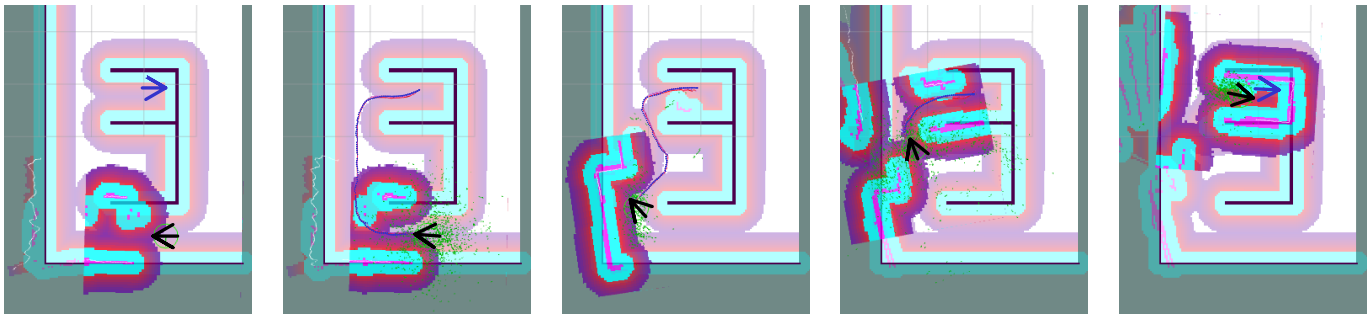

Fig. 7: Results of a preliminary test of LoCoBot navigation conducted in the experimental setup depicted in Figure 4

Figure 7 shows different phases of the MoMa navigation: highlighted colored areas identify the perimeter detected by the mobile robot camera. Note that the covered region measures approximately 1.5m $\times$ 1.5m. The initial and target position of the LoCoBot is displayed in Figure 7(a) in black and blue, respectively, while Figure 7(b) reports the initial instants of the MoMa navigation and the path planned by the WS (blue line). In Figure 7(c), one can observe the presence of an obstacle on the map (a temporarily set down package). This is detected by the robot and its path is accordingly revised. We highlight that the new planned trajectory is characterized by a smooth profile and reveals a conservative behavior as regards the possible maneuvers. In particular, the MoMa plans to overcome the obstacle, while keeping a safe distance from the other (known) elements in the map. In Figure 7(d) the attention is focused on the obstacle avoidance maneuver. We report most critical situation wherein the robot passes by a narrow passages. Finally, Figure 7(e) depicts the task conclusion when the LoCoBot approaches the target location.

V. Discussion

Although the preliminary experimental tests provide encouraging results, the challenging aspects and open design choices affecting the proposed proof of concept are numerous. We aim at discussing some of them in the following.

a) Adopted approach: in the outlined multi-node system, the navigation goal can be faced either in a centralized or a distributed manner. Here, we propose a solution based on a centralized approach, mainly resting on a WS able to handle the most of the tasks itemized in Section II. The advantages derived by this design choice consist in the simple and effort-limited realization of the entire multi-element architecture and also in the high and complete control of all the involved devices. On the other hand, the adoption of a distributed paradigm grants higher network scalability and also robustness in terms of both communication and node failure, at the price of more complicated management of the data. In the distributed scenario, indeed, each RGB-D sensor can possibly share the retrieved information only with a restricted group of nodes in the network and not necessarily submit to a tight periodic schedule. From a practical point of view, coping with the MoMa navigation in a distributed framework implies the online path computation from the LoCoBot itself, and such a path can be determined only in a suboptimal manner since the robot can have only partial information about the environment. This issue can make also the MoMa localization more prone to errors, thus affecting the trajectory tracking accuracy.

b) Communication infrastructure: in our benchmark, the communication between the WS and the RGB-D sensor occurs over Ethernet. This choice is mainly motivated by two factors. The former is the continuous streaming of images going from the fixed nodes to the central one. The Raspberry Pi 4B WiFi module is not very efficient, so selected a cabled connection to grant both speed and reliability. As for the latter, the adopted infrastructure guarantees that the robot is the only device using the WiFi network, therefore preventing possible congestion

in the hotspot. This is crucial to avoid undesired latency in the system. Indeed, adopting a wireless-only approach would reduce the network infrastructure, but the size and frequency of the information interchange should be reduced accordingly. Distributing the computational burden from the WS to the fixed nodes would reduce the network workload and provide alternatives to the current solution.

c) Environmental data: the main source of data representing the environment consists of the RGB-D images.This is basically raw information as collected from the cameras. We underline that different alternatives can be considered, especially with the purpose of balancing the computational effort with the payload in the packages traversing the network to avoid congestion. One solution relies on computing the 2D laser scan directly inside the acquisition node. Another possibility that could work particularly well on fixed nodes focuses on sending differential information including only the parts that are significantly changed with respect to the last update. On the other hand, the 2D map significantly reduces the information available to the MoMa in case of manipulative tasks or complex obstacles. Computing a 3D point cloud can compensate, possibly by integrating into the mobile platform an external Vision Processing Unit (VPU) to manage the additional workload. Balancing image resolution and framerate is fundamental to gathering effective environmental data since they affect all the techniques previously described.

d) Map update: a substantial but released design choice rests on the management of the map update by mean of the fixed nodes. In Section IV, we state that the environment data are periodically gathered from the RGB-D sensors, depending also on the framerate imposed on the cameras. However, map updates can be subject to an obstacle detection or a significant environment change. In this scenario, to suitable handle the RGB-D sensors information it is worth developing an ad-hoc event-based decision-making strategy, which may require increased computational capabilities from the fixed nodes.

e) Robot localization: assuming to deal with WiFi-only communication among the system nodes, the MoMa localization in the environment can be accomplished also performing trilateration based on the received signals power. This localization technique can be exploited in conjunction with the standard visual based ego-pose estimation.

f) Path planning: the path definition can be subject also to further either robot constraints or dynamic environment configuration. For instance, it is suitable to determine the robot desired trajectory by taking into account the shape and volume of the transported item (if any). Along the same line, the path planner can exclude all the regions of the environment that are temporarily unavailable, e.g, the areas occupied by loading and unloading. We observe that the information needed in both the reported examples can be provided by the fixed nodes, which are thus required to fulfill also a surveillance task. More in general, we emphasize that the proposed multi-node system is a flexible framework whose components can accomplish auxiliary and/or other tasks.

VI. Concluding Remarks

In this work, we present a proof of concept for robot navigation in an industrial scenario, exploiting an auxiliary RGB-D sensor network. Main contribution regards the use of low-cost devices interacting among them in a multi-node framework based on a ROS2 architecture. The results of the preliminary tests are encouraging, highlighting the suitability of the designed setup, composed by a LoCoBot as mobile node and a network of RGB-D cameras, paired with Raspberry Pi 4B for the computational unit, as fixed nodes. A working station is adopted as a central unit for a major control of the overall system, however also a distributed paradigm can be quite effortless configured with this design for scalability and robustness purposes. This and other open design choices are discussed in the work, taking into account the challenging aspects for industrial applications, as the communication mean.

We are aware that this work constitutes only a starting point for the development of effective strategies for robot navigation. Particular attention is devoted to the design of a flexible solution, the possibility to extend the network with other MoMas simultaneously acting in the same workspace, and also expandable with an increased number of fixed nodes.

References

[1] B. Chen *et al.*, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017.

[2] A. Ollero *et al.*, "The AEROARMS project: Aerial robots with advanced manipulation capabilities for inspection and maintenance," *Robotics & Automation Magazine*, vol. 25, no. 4, pp. 12–23, 2018.

[3] B. Elaamery *et al.*, "Model predictive control for cooperative transportation with feasibility-aware policy," *Robotics*, vol. 10, no. 3, pp. 84–104, 2021.

[4] N. Lissandrini *et al.*, "Cooperative optimization of UAVs formation visual tracking," *Robotics*, vol. 8, no. 3, p. 52, 2019.

[5] D. Wooden *et al.*, "Autonomous navigation for BigDog," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 4736–4741.

[6] Y. Li and C. Shi, "Localization and navigation for indoor mobile robot based on ROS," in *IEEE Chinese Aut. Congress*, 2018, pp. 1135–1139.

[7] T. Kim, S. Lim, G. Shin, G. Sim, and D. Yun, "Real-time navigation system for a low-cost mobile robot with an rgb-d camera," *arXiv preprint arXiv:2103.03054*, 2021.

[8] W. Yuan, Z. Li, and C.-Y. Su, "RGB-D sensor-based visual SLAM for localization and navigation of indoor mobile robot," in *IEEE Int. Conf. on Advanced Robotics and Mechatronics*, 2016, pp. 82–87.

[9] H. Zhang, C. Zhang, W. Yang, and C.-Y. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," in *IEEE Int. Conf. on Robotics and Biomimetics*, 2015, pp. 2501–2506.

[10] P. Alves, H. Costelha, and C. Neves, "Localization and navigation of a mobile robot in an office-like environment," in *IEEE Int. Conf. on Autonomous Robot Systems*, 2013, pp. 1–6.

[11] H. Li and A. V. Savkin, "An algorithm for safe navigation of mobile robots by a sensor network in dynamic cluttered industrial environments," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 65–82, 2018.

[12] O. M. Elfadil, Y. Alkasim, and E. B. Abbas, "Indoor navigation algorithm for mobile robot using wireless sensor networks," in *IEEE Int. Conf. on Communication, Control, Computing and Electronics Engineering*, 2017, pp. 1–5.

[13] T. T. Khanh, V. Nguyen, X.-Q. Pham, and E.-N. Huh, "Wi-fi indoor positioning and navigation: a cloudlet-based cloud computing approach," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–26, 2020.

[14] "Intel RealSense," https://www.intelrealsense.com, accessed: 2021-10.

[15] "LoCoBot," http://www.locobot.org, accessed: 2021-10.

[16] "Raspberry Pi," https://www.raspberrypi.com, accessed: 2021-10.