Sensor data fusion for smart AMRs in human-shared industrial workspaces

Sensor data fusion for smart AMRs in human-shared industrial workspaces

Marina Indri, Fiorella Sibona, Pangcheng David Cen Cheng
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{marina.indri, fiorella.sibona, pangcheng.cencheng}@polito.it

***Abstract*—A growing presence of mobile agents is envisaged in the smart factories scenario of the next future. The safe motion of traditional Automated Guided Vehicles in human-shared workspaces can be achieved thanks to the support of a fleet of Autonomous Mobile Robots, acting as a net of meta-sensors, able to detect the human presence and share the information. This paper proposes a preliminary working implementation of one meta-sensor module, exploiting the synergistic use of different sensors through an overall affordable and accessible sensor data fusion algorithm. Experimental results in a laboratory environment confirm the validity of the approach.**

***Index Terms*—Mobile robots, human recognition, obstacle avoidance, sensor data fusion.**

I. Introduction and state of the art

In recent years, mobile robots have been widely employed in many different fields, since they can be adapted to a vast range of applications. In the industrial context, mobile robots can be classified as Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs). The AGVs are mobile platforms that perform repetitive tasks, such as material transportation, following a pre-defined path within industrial environments [1]. Human-accessible workspaces are traditionally separated from the AGV operational space due to safety reasons, since AGVs do not have a decision mechanism based on artificial intelligence, and most of them require a particular infrastructure setup [2]. On the other hand, an AMR is able to sense its surroundings in order to create a model of the environment and locate itself in it, leading to the capability of operating in an unknown or partially known environment. Usually, an AMR is equipped with an heterogeneous set of sensors whose output data streams are processed by complex control systems [3]. It is well known that merging the information coming from different sensors improves the efficiency and robustness of the measurements, but on the other hand it increases the complexity of the hardware and software required for merging and processing the information deriving from different sources [4].

Depending on the application, one may combine different types of sensors in order to cover a wider range of measurements or duplicate the data to avoid false positives. For example, AMRs working in industrial applications, where a safe behavior between machinery and human operators must be ensured, are commonly equipped with a combination of vision and distance sensors, e.g., cameras with Radio Detection And Ranging (RADAR) systems [5] or cameras with Laser Imaging Detection and Ranging (LIDAR) sensors [6], [7].

The vision sensor is used for recognition of objects or particular geometric patterns, but is influenced by environmental conditions, e.g., the ambient lighting. The distance sensor provides high accuracy on measurements, even though its performance may be affected by (i) the reflection properties of the object to be detected, in the case of LIDARs, and (ii) by external radio wave frequencies, when using RADARs. Despite the limitations of each sensor type, by combining them it is possible to associate a detected object with its corresponding distance in the robot coordinate system.

A possible method for performing sensor fusion is represented by the *machine learning* approach. A commonly used Neural Network (NN) for object classification in an image or video frame is the *You Only Look Once* (YOLO) [8] real-time object detection system. YOLO applies a Convolutional Neural Network (CNN) to the full image, dividing it into regions and performing the bounding boxes prediction and relative probabilities computation for each region. The bounding boxes that have high confidence scores are kept as final predictions, resulting in detections. In [9], the authors use the data set coming from RGB-D cameras and compare the performance of several CNNs in order to robustly detect and localize a person. The work in [10] proposes a person detection algorithm based on the linear Support Vector Machines (SVM) learning process that extracts laser features and Histograms of Oriented Gradients (HOG) features from image data. Other approaches, e.g., in [11], combine a Kalman filter with the Global Nearest-Neighbour method, in order to predict and resolve the data association problem for people tracking. Furthermore, a considerable body of literature treats the LIDAR and vision data fusion as an extrinsic calibration problem: the two sensors' coordinate systems are put in relation through a rigid body transformation, so to align the data derived from both sensors [12]. For the purpose of computing this transformation, usually an external object is required, such as a checkerboard pattern [13], [14] or a trirectangular trihedron [15], to match the correspondences between the two sensors and obtain a mapping that transforms the points from the laser to the camera coordinate system, and thereafter to the image plane. Although most of the researchers use stereo cameras and 3D LIDARs to model the environment (since they can give a detailed representation of the surroundings), these devices are expensive, and most of the time it is possible to overcome this problem by using transformation matrices when using a 2D LIDAR and an entry-level camera [16].

The research activity has been partially supported by the HuManS – Human-centered Manufacturing Systems project, funded by Regione Piemonte within the MIUR-POR FESR 2014/2020 funding program.

If we consider the problem from a higher level of analysis, a mobile robot reaction to dynamically-changing surroundings has been addressed in several ways. In [17], the mobile platforms have the capability of overtaking unforeseen obstacles appearing on (or near) the pre-computed path, exploiting local deviations fostered by a centralized data fusion system, which takes in on-board sensing along with infrastructure-based environment perception system measurements. In [18], the mobile robot navigation in unknown dynamic environments implements a pedestrian-like behavior: thanks to the human neural system, a pedestrian can estimate the time to collision with obstacles and changes its direction accordingly, while optimizing path length and safety distance. The navigation approach is similarly based on motion model predictability of obstacles.

The scenario envisaged for the smart factories of the next future implies a significant presence of mobile agents, both having some level of autonomy or not, in open spaces shared with humans. This paper addresses, in particular, the system described in [19], where AMRs act as *meta-sensors*, i.e., as mobile entities included in a wider concept of sensor system, having the aim of supporting a net of traditional AGVs in order to increase their consciousness about their surroundings and to be compliant with collaborative operations between humans and robots. The contribution of this paper represents a preliminary working implementation of one meta-sensor module: it exploits camera-laser data sensor fusion techniques to ensure a safe behavior when specific objects are detected, and shares relevant data with other robotic systems. In particular, the attention is devoted to human detection, which has to be guaranteed in a safe way through a proper SW/HW architecture including both *safe* and *not safe* devices, from the industrial point of view.

Unlike other commonly adopted methods, where the sensor system is trained at the beginning to combine the data from different sources, we take advantage of a pre-trained NN for human identification. The outputs of the NN are the elements of the image already classified and labeled, which are subsequently used for the sensor fusion algorithm, avoiding the creation of a further dataset.

The aim of the authors is then to provide an affordable solution, which takes advantage of sensor fusion to integrate state-of-the-art object detection algorithms taking data from low-cost vision sensors (that may not be intrinsically safe, when used on their own) to complement standard safety compliant sensors (e.g., safety-rated laser scanning systems). Moreover, with respect to the most recent concept that positions AMRs as an evolution of traditional AGVs, here the former support the latter ones during their daily tasks, enabling the possibility of bringing back to light pre-existing obsolete systems. Furthermore, non-infrastructural sensors allow for a more flexible set-up and scalability with respect to other solutions, which associates centralized systems with infrastructural monitoring.

The paper is organized as follows: Section II presents the proposed solution, providing a high level description within the working scenario. Then in Section III, the adopted implementation is briefly illustrated. Section IV presents the obtained results when testing the sensor data fusion algorithm to detect and safely avoid humans. Finally, Section V draws some conclusions and open issues.

II. Sensor data fusion for a meta-sensor AMR

This work describes a first implementation of the entity called meta-sensor, which plays a fundamental role within the system whose specifications are set in [19]. In order to give a background and motivation to the features that have been built up for this entity, a brief description of the working scenario is provided hereafter. The system is thought for ideally any flexible production line, composed of traditional AGVs, workstations, cobots, in spaces shared with human operators. Three macro-elements can be identified: (i) a meta-sensor AMR fleet, (ii) the Sensors Synergy Center (SSC) and (iii) the AGV Coordination Center Interface.

The work presented here covers points (i) and (ii): note that the developed structure/model for the AMR meta-sensor entity can be replicated for other elements of the fleet. When going through the solution description, notice that our AMR is a feature-enhancer entity more than a mere evolution of the classical AGV; it is a part of the AGV net, the "brain" behind the system synergy, leveraging sensor data fusion to improve the AGV fleet awareness about the environment's dynamical changes. In particular, in our case we consider the human operator as the target of interest to be detected and advertised to all agents moving within the system.

A. Solution overview within the meta-sensor system

Having fixed the AMR role in the system, we can identify the behaviour and capabilities we would like the AMR element and SSC to have. In order to achieve smart navigation in a human-shared workspace, we need to gather informative data from the surroundings. With this aim, but with the purpose of keeping cost low, too, we decided to equip a mobile robot with a monocular camera and a 2D laser range finder in order to perform data association. So to perform a correct mapping of information, the transformation between the laser and the camera must be computed (extrinsic calibration).

The artificial intelligence needed for spotting any human obstacles is entrusted to the vision part of the sensor system (human-obstacle detection).

Once a human obstacle is detected and its absolute position identified, we want the AMR to share this information with other agents and define a reaction rule in the presence of this particular kind of obstacle (human-obstacle avoidance).

1) Extrinsic calibration: In order to merge the information coming from a monocular camera and a laser range finder, an extrinsic calibration method is required to transform the laser points in the camera reference frame and project them onto the image plane. For that, first we determined the internal and external parameters of the camera, and then computed the rigid body transformation between the laser and the camera coordinate systems.

- *Camera Calibration*. The camera calibration consists in estimating a relationship between the information of the camera coordinate system and the image frame, along with the relative pose of the camera with respect to the world reference frame [20]. Assuming the pinhole model of the camera, the transformation of the 3D points $\mathbf{C_p} = [X, Y, Z, 1]^T$ to the 2D points $\mathbf{c_p} = [u, v, 1]^T$ is defined as:

$$\mathbf{c_p} \sim \mathbf{K} \cdot \left[\mathbf{R} \,\vdots\, \mathbf{t}\right] \cdot \mathbf{C_p} \tag{1}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the so-called camera intrinsic matrix, $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^{3\times1}$ are the extrinsic parameters of the camera, which relate the world 3D information to the camera coordinate system.
The intrinsic matrix $\mathbf{K}$ is a projective transformation of the 3D points from the camera coordinates into the 2D image coordinates and is defined as:

$$\mathbf{K} = \begin{bmatrix} \alpha_1 & s & c_x \\ 0 & \alpha_2 & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where α_1 and α_2 are the focal lengths in pixel units, c_x and c_y the coordinates of the principal point of the image in pixel units, and s the skew coefficient between the axis of the image.
Since the ideal pinhole camera model does not have a lens and the image from a real camera may present some deformation, we have performed the correction by estimating the radial and tangential distortion coefficients [21].

- *Camera-Laser calibration*. Once the camera is calibrated, we applied a laser to camera extrinsic calibration algorithm based on [14], taking into account the physical characteristics of the sensors, so to estimate the relative pose of the camera with respect to the laser range finder. Let's consider a point $\mathbf{C_p} \in \mathbb{R}^{4\times1}$ in the camera coordinate system, located in $\mathbf{L_p} \in \mathbb{R}^{4\times1}$ in the laser reference frame. The rigid transformation between the two coordinate systems can be expressed as:

$$\mathbf{L_p} = \left[\begin{array}{c|c} & \\ \mathbf{\Phi} & \mathbf{\Delta} \\ & \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \cdot \mathbf{C_p} \tag{3}$$

where $\mathbf{\Phi} \in \mathbb{R}^{3\times3}$ is the rotation matrix of the camera with respect to the laser range finder and $\mathbf{\Delta} \in \mathbb{R}^{3\times1}$ the relative translation vector.

Hereafter, we enter the core description of the centralized SSC features integrating the meta-sensors net.

2) Human-obstacle detection: In general, the aim of object identification is to determine the presence, in some given image or frame, of any instance of objects categorized into classes. The recognized objects spatial locations within the image reference frame are returned, e.g., via bounding boxes. Object detection performances have been boosted up with the introduction of deep learning techniques, which let a machine automatically learn feature representations from data. With deep learning, in general, patterns are classified using statistical techniques based on sample data and processing it with multi-layered neural networks [22], [23]. The CNNs are among the most popular architectures for deep-learning: they are designed to receive multiple arrays data as input, e.g., a three-channel (RGB) image array structure. Many methods handle detection as a classification problem, i.e., object proposals are produced and fed to a classifier. However, some other methods formulate detection as a regression problem, having spatially separated bounding boxes and associated class probabilities as output [24]. Most of the recent methods are region-based, i.e., they perform a selective research to obtain region proposals, despite this kind of approach often represents a speed bottleneck.

Hence, regression-based methods getting rid of the region proposal step have represented a suitable choice for our purposes, since we need to identify a specific object class with some index of confidence (i.e., the output class probabilities for the detected objects), and locate these objects in a suitable spatial representation in a reasonable time. Figure 1 represents the adopted human detection process at high level.

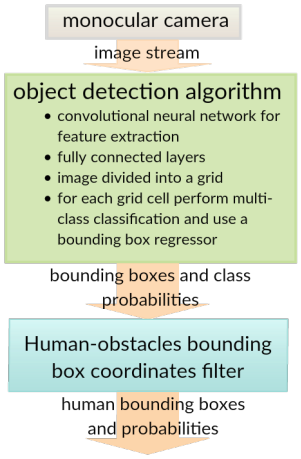

Fig. 1. Human obstacle detection process.

3) Human-obstacle avoidance: For what concerns a mobile platform reaction to a human-obstacle (advertised by other agents on the shared map or directly sensed by the considered robot), the authors have decided to apply a more conservative approach in terms of safety distance separating the moving agent and the detected obstacle. By extracting the human location expressed in the shared map reference frame, we get target positions that can be enclosed in *virtual cages*, i.e., areas considered not accessible by the network of mobile robots, characterized by a greater safety radius value with respect to other obstacles. Note that, since relevant information is shared between the Sensors Synergy Center and the AGV Coordination Center, the presence of humans is made available to both mobile robot fleets (AMRs and AGVs). Thus, the rules for obstacle avoidance applied in this case are not different from the ones adopted when encountering generic obstacles.

III. Solution Implementation

A. Hardware setup

In order to test the sensor data fusion algorithm for human detection and avoidance, we have employed a Pioneer 3DX mobile robot [25] equipped with a SICK LMS200 laser range finder [26] with 10-meter range and scanning angle of 180°, an entry level IP camera (ONVIF [27] standalone unit, accessible via its IP address). As a processing unit for receiving data and controlling the robot, we used a Raspberry Pi [28] 3 Model B mounting an ARM Cortex-A53 (x4 core) CPU (1.2 GHz) and 1-GB RAM; the code that required a higher computational effort has been run on a desktop PC with a Intel Core i7-7700 CPU and a dedicated GTX1060/6GB GPU. The monocamera has been placed above the laser range finder and its orientation

set such that the image plane intersects the laser plane as shown in Figure 2.

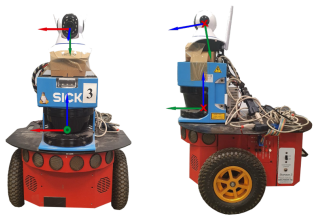

Fig. 2. Representation of the sensors reference frames.

B. Software implementation

The steps taken for the solution implementation are described below.

1) Laser-Camera transformation computation:

- **Intrinsic Calibration with MATLAB**: The intrinsic calibration of the camera was performed using the MATLAB Computer Vision Toolbox, which requires a set of 20 images taken when a checkerboard is placed at different orientations, assuming that all inclination angles are kept below 45° with respect to the camera plane [29]. The camera provides an image resolution of 1280 x 720 pixels, and so the following intrinsic matrix is obtained:

$$\mathbf{K} = \begin{bmatrix} 1.3046 \cdot 10^3 & 0 & 727.7219 \\ 0 & 1.3064 \cdot 10^3 & 241.5278 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Extrinsic calibration and laser point projection**: The extrinsic calibration was computed by collecting simultaneously data from both the vision and range finder sensors, i.e., 20 images of the checkerboard in several orientations and their corresponding laser readings. Taking into account the characteristics of the sensors, we applied a modified version of the extrinsic calibration algorithm proposed by Zhang and Pless in [14], obtaining the following transformation matrices:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9848 & 0.1736 \\ 0 & -0.1736 & 0.9848 \end{bmatrix}, \mathbf{\Delta} = \begin{bmatrix} -0.0215 \\ -0.2065 \\ 0.0304 \end{bmatrix}$$

With the rotation matrix $\mathbf{\Phi}$ and translation vector $\mathbf{\Delta}$, it is possible to compute the laser points in the camera reference frame, and project them in the image plane. Indeed, by defining the matrix $\mathbf{H} \in \mathbb{R}^{3\times4}$ as:

$$\mathbf{H}(i,j) = \mathbf{\Phi}^{-1}(i,j), \text{ for } i = 1,...,3 \text{ and } j = 1,...,3$$
$$\mathbf{H}(i,4) = -\mathbf{\Delta}(i), \text{ for } i = 1,...,3$$

and combining (2) and (3) the vector $\mathbf{c_p}$ is obtained as:

$$\mathbf{c_p} = \mathbf{K} \cdot \mathbf{H} \cdot \mathbf{L_p} \tag{4}$$

Note that the world 3D reference frame is coincident with the camera coordinates one, and so $\mathbf{R}$ is the identity matrix and $\mathbf{t}$ is a null vector. In order to have a correct representation of the points in the image, the vector $\mathbf{c_p}$ must be normalized with respect to the third component, leading to a vector in the form $\hat{\mathbf{c}}_\mathbf{p} = [u, v]^T$.

As a result, we obtained a projection of the laser points in the image plane, as shown in Figure 3, which is quite reasonable, since the mapped points are coherent with the hit surfaces.

Fig. 3. Laser points projected in the image plane.

2) Relevant bounding box information extraction: Since the image processing requires a set of different tools, we have decided to group all the related software within a Docker [30] container, which leverages the GPU computational capabilities and removes the burden of installing ad-hoc tools, fostering portability and scalability.

As a human detection system we have chosen YOLO (which satisfies our requirements, as specified in Section II-A2), that processes the IP camera stream, previously interpreted as a generic webcam output using the *gstreamer* [31] tool.
The YOLO code that generates the bounding boxes on screen has been modified, so to save their coordinates information in a text file. This file is used as a source for a ROS [32] node that reads it and wraps into ROS topic messages only the information we are interested in (i.e., the lines identified by a "person" label). All communications between the container and the other ROS distributed nodes happen through the host network. Notice that simply reading the stream output by a low-cost IP plug & play camera lightens up the already computationally heavy image processing step.

3) Mapping laser data to the image plane: The information published by the laser range finder is processed within a specific ROS node and projected on the image plane, exploiting the estimated calibration parameters. This same node is also in charge of comparing such information with the messages published on the topic related to the human-obstacles bounding boxes coordinates: data are synchronously gathered from topics, by exploiting the *ApproximateTimeSynchronizer* class that is included in the `message_filters` ROS package. If a correspondence among pixel coordinates is found, the relative points of interest are translated to be expressed in the global map reference frame, using the computed rigid transformation between the sensors and the robot model reference frames (all made available through the `tf` ROS publishing system).

4) Detected humans as virtual obstacles: In order to implement the desired safe reaction to human obstacles, we have taken advantage of the Time Elastic Bands (TEB) local planner [33] which creates three elastic bands based on different criteria and chooses the shortest one, taking into account dynamic obstacles and vehicle constraints [34]; it also allows to define custom virtual obstacles by specifying their location and shape. This local planner can be integrated with the global planner provided by the ROS navigation package. In this way, all points falling into each person bounding box are published in the map as circular obstacles with a user-defined radius: the detected human horizontal extension influences the obstacle shape. Thus, a human presence adds a constraint for the re-

planning process of the robot, ensuring safety between humans and robots.

Figure 4 summarizes the whole process: the YOLO C++ software has been edited in order to select all the "person" labeled objects and append them to a *.txt* file, whose content is fed to a ROS node for its translation into ROS messages. Another node is in charge of filtering synchronized data to identify laser points falling into the pixel ranges corresponding to humans. Finally, virtual obstacles are published in correspondence of the human coordinates, with a radius that is added to the *rviz* inflation one.

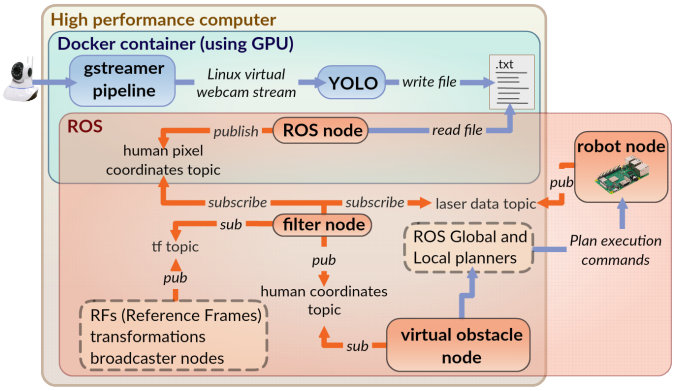

Fig. 4. Sketch of the process workflow.

IV. Experimental Results

In this section we present the results obtained by employing the proposed sensor data fusion algorithm to enable a safe mobile robot reaction in the presence of humans.

Furthermore, with the aim of demonstrating the achieved results, a sample working scenario is considered in the video available in [35].

The main features of our algorithm are highlighted in some salient points of the video that can be summarized as follows:

- In Figure 5, we can see that a human is detected and its location (perceived by the laser and inflated by ROS) is reinforced by the publication of a set of virtual obstacles (red squared markers). The path computed by the local planner conservatively stays out of the inflation radius imposed by ROS, since the obstacle position is computed not only as detected by the laser range finder, but as the result of the latter and the YOLO processed information.
- Figure 6 illustrates how the virtual obstacle publication depends on the size of the bounding box detected during the image processing stage.
- Our algorithm is able to identify, and consequently publish, two human obstacles simultaneously: this influences the re-planning process for the mobile robot, as can be seen in Figure 7.

Note that, since the overall execution time depends on (i) the chosen planner, (ii) the adopted algorithms, and (iii) on the computational capability of the computer executing the sensor data fusion code, the reaction of the robot might fall within the standard definition of soft real-time processes or not.

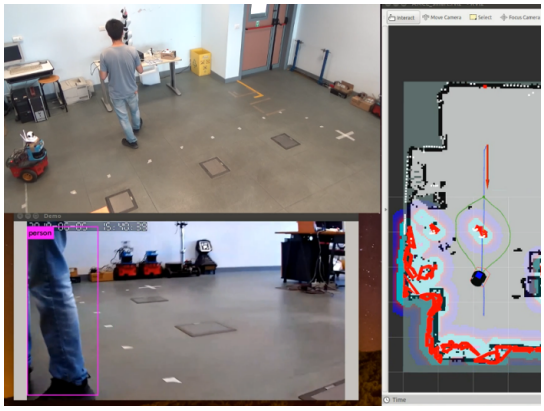

Fig. 5. Human detection and relative virtual obstacle publication at 02:37.

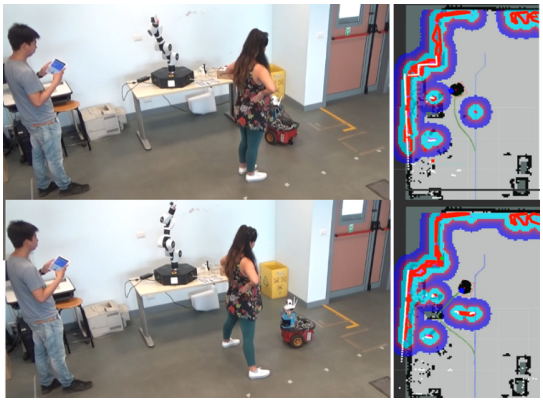

Fig. 6. The local planned path depends on the virtual obstacle, which covers the whole detected person bounding box.

V. Conclusions and Future Works

In this paper we presented an overall affordable and accessible sensor data fusion algorithm for mobile robots to ensure safety in an industrial environment shared with human operators. The main contribution of the paper is to show how a safe human detection can be achieved by the synergistic use of different sensors (even low cost ones) within an overall

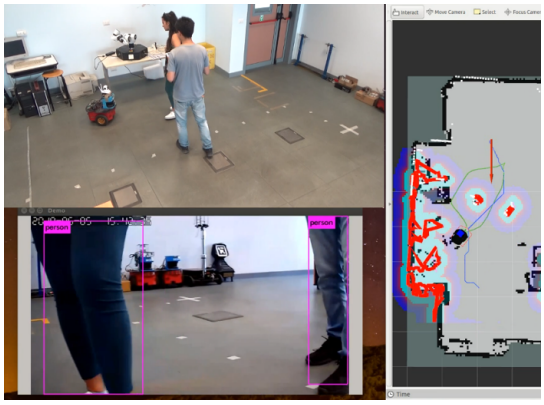

Fig. 7. Several human obstacles can be simultaneously detected fostering scalability (04:32).

HW/SW architecture allowing to share the information within a fleet of AMRs, acting as meta-sensors to support the working of standard AGVs. For the moment, the application is feasible when the relative motion between the robot and the human is slow, so that the robot has enough time to react and re-plan the trajectory, avoiding thus the human operator.

Since the adopted calibration method needs a sufficient amount of data, it has been performed offline: this can be considered a relatively easier approach since the process has to be executed once, but it obviously relies on the assumption that the involved sensors are well and definitely fixed in place. Enhanced solutions would be given by the implementation of an online calibration procedure, or the application of deep-learning algorithms for data association.

As future work, one of our objectives will be also to consider the proposed algorithm for the overall sensing system composed by multiple mobile robots as specified in the original scenario, where all mobile platforms (AGVs and AMRs) share the relevant information about their surroundings. Moreover, even though our solution allows to spread the information to other mobile robots so to influence their local behaviour, an ideal implementation would publish the detected humans as cost-map obstacles allowing for a more aware and efficient global plan computation. This feature, which we intend to consider in future developments, is shown in Figure 8.

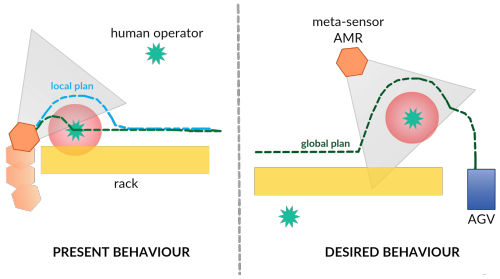

Fig. 8. Global and local human avoidance rule schematic representation.

Another possible improvement of the sensor fusion algorithm itself would require the training of the neural network involved at the object recognition step, so to adapt the reaction behaviour depending on the detected object.

REFERENCES

[1] M. Bader, A. Richtsfeld, M. Suchi, G. Todoran, W. Holl, and M. Vincze, "Balancing centralised control with vehicle autonomy in AGV systems for industrial acceptance," in *11th Int. Conf. on Autonomic and Autonom. Sys*, 2015.

[2] J. Theunissen, H. Xu, R. Y. Zhong, and X. Xu, "Smart AGV System for Manufacturing Shopfloor in the Context of Industry 4.0," in *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2018, pp. 1–6.

[3] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an autonomous mobile robot based on ROS," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2017, pp. 1–5.

[4] H. B. Mitchell, *Multi-sensor data fusion: an introduction*. Springer Science & Business Media, 2007.

[5] X. Wu, J. Ren, Y. Wu, and J. Shao, "Study on target tracking based on vision and radar sensor fusion," doi: 10.4271/2018-01-0613, SAE Technical Paper, Tech. Rep., 2018.

[6] Z. Yan, L. Sun, T. Duckctr, and N. Bellotto, "Multisensor Online Transfer Learning for 3D LiDAR-based Human Detection with a Mobile Robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7635–7640.

[7] V. De Silva, J. Roche, and A. Kondoz, "Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots," *Sensors*, vol. 18, no. 8, 2730, 2018.

[8] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[9] T. Linder, D. Griesser, N. Vaskevicius, and K. O. Arras, "Towards Accurate 3D Person Detection and Localization from RGB-D in Cluttered Environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Workshop on Robotics for Logistics in Warehouses and Environments Shared with Humans*, 2018.

[10] L. Spinello and R. Siegwart, "Human detection using multimodal and multidimensional features," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3264–3269.

[11] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2D laser scanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 726–733.

[12] V. D. Silva, J. Roche, and A. M. Kondoz, "Fusion of LiDAR and Camera Sensor Data for Environment Sensing in Driverless Vehicles," *CoRR*, vol. abs/1710.06230, 2017. [Online]. Available: http://arxiv.org/abs/1710.06230

[13] K. Ahmad Yousef, B. Mohd, K. Al-Widyan, and T. Hayajneh, "Extrinsic calibration of camera and 2D laser sensors without overlap," *Sensors*, vol. 17, no. 10, 2346, 2017.

[14] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[15] Z. Hu, Y. Li, N. Li, and B. Zhao, "Extrinsic calibration of 2-D laser rangefinder and camera from single shot based on minimal solution," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 915–929, 2016.

[16] J. Li, X. He, and J. Li, "2D LiDAR and camera fusion in 3D modeling of indoor environment," in *2015 National Aerospace and Electronics Conference (NAECON)*. IEEE, 2015, pp. 379–383.

[17] V. Digani, F. Caramaschi, L. Sabattini, C. Secchi, and C. Fantuzzi, "Obstacle avoidance for industrial AGVs," in *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sept 2014, pp. 227–232.

[18] N. M. Kakoty, M. Mazumdar, and D. Sonowal, "Mobile Robot Navigation in Unknown Dynamic Environment Inspired by Human Pedestrian Behavior," in *Progress in Advanced Computing and Intelligent Engineering*, C. R. Panigrahi, A. K. Pujari, S. Misra, B. Pati, and K.-C. Li, Eds. Singapore: Springer Singapore, 2019, pp. 441–451.

[19] M. Indri, L. Lachello, I. Lazzero, F. Sibona, and S. Trapani, "Smart Sensors Applications for a New Paradigm of a Production Line," *Sensors*, vol. 19, no. 3, 650, 2019.

[20] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[21] J. Heikkila, O. Silven *et al.*, "A four-step camera calibration procedure with implicit image correction," in *cvpr*, vol. 97, 1997, p. 1106.

[22] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.

[23] G. Marcus, "Deep Learning: A Critical Appraisal," *CoRR*, vol. abs/1801.00631, 2018. [Online]. Available: http://arxiv.org/abs/1801.00631

[24] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: a survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, 2018.

[25] M. Inc, *Pioneer 3 Operations Manual*.

[26] S. A. Waldkirch, *PLMS200/211/221/291 Laser Measurement Systems*.

[27] "ONVIF Official Site," Available online: https://www.onvif.org/.

[28] "Raspberry Pi site," Available online: https://www.raspberrypi.org/.

[29] "MATLAB Computer Vision Toolbox," Available online: https://it.mathworks.com/products/computer-vision.html.

[30] "Docker official website," Available online: https://www.docker.com.

[31] "GStreamer official webite," Available online: https://gstreamer.freedesktop.org/.

[32] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[33] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[34] P. Marin-Plaza, A. Hussein, D. Martin, and A. d. l. Escalera, "Global and local path planning study in a ROS-based research platform for autonomous vehicles," *Journal of Advanced Transportation*, vol. 2018, 2018.

[35] Human detection and avoidance with the Pioneer 3DX. [Online]. Available: https://youtu.be/lQfDR4PMuV0