IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 July 2023, accepted 21 July 2023, date of publication 25 July 2023, date of current version 31 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298602

RESEARCH ARTICLE

Data Homogenization Method for Heterogeneous Sensors Applied to Reinforcement Learning

ELIZABETH PALACIOS-MOROCHO, (Student Member, IEEE), PABLO LÓPEZ-MUÑOZ, MANUEL A. COSTÁN, AND JOSE F. MONSERRAT, (Senior Member, IEEE)

iTEAM Research Institute, Universitat Politècnica de València, 46022 Valencia, Spain

Corresponding author: Elizabeth Palacios-Morocho (mapamo3@iteam.upv.es)

The work of Elizabeth Palacios-Morocho was supported by the Research and Development Grants Program, Universitat Politècnica de València, under Grant PAID-01-19.

ABSTRACT In autonomous navigation and route planning, the data obtained by the different sensors play a significant role. On the one hand, more data will lead to faster learning of the behavioral policy. On the other hand, agents equipped with different sensors will need more computing power to process the data, thus requiring more robust equipment and increasing the cost of implementation. In addition, the complexity of the algorithms increases as different types of data, i.e., data with different structures, have to be synchronized. Therefore, this paper addresses the problem of homogenization and synchronization of data provided by heterogeneous sensors. Furthermore, it presents a novel method of estimating data in order to provide the agent with a 360-degree view of the environment, similar to that provided by a laser. The method's performance compares the different behavioral policies obtained by different viewing angles of a camera with the policy obtained by a laser. The data obtained from the different viewing angles of each sensor are used in a path planning algorithm, which was designed to use a single 24-scan laser as an input source. The results show that the proposed method is robust since the behavior policies can be reused regardless of the viewing angle with which the sensor (camera) is provided. Furthermore, the proposed novel algorithm achieves an average efficiency of 68% and 94% using a 90 and 360-degree camera, respectively.

INDEX TERMS Artificial intelligence, reinforcement learning, heterogeneous data, homogeneous data, point cloud, laser scan, interpolation.

I. INTRODUCTION

The field of Artificial Intelligence (AI) has produced a series of challenges and opportunities that have led to significant technological achievements, where Machine Learning (ML) has been one of the pillars of digital transformation. Enabling the development of many use cases in the fields of entertainment, medicine, mining, education, military space, agriculture, robotics, and transportation [1].

One of the main case studies in the field of robotics and transportation is the implementation of autonomous systems. These include autonomous ground or airborne vehicles, robots providing services such as food preparation, space robots, marine robots, and smart factories [2]. However, providing autonomy to an agent represents a great challenge since it means that the agent is able to move in an unknown environment and successfully react to sudden changes that may occur in the environment [3].

Many algorithms have been proposed to provide a solution to this type of scenario, such as the case of Pfeiffer et al., who presented a model based on laser mapping, which can determine the position of the different targets and select the best decision. However, being a supervised learning method, it is limited by the quality of the experiences obtained by the simulation performed with a motion planner [4]. To overcome these limitations, Gregory et al. presented a Reinforcement Learning (RL) algorithm that estimates the collision probability of the robot [5]. Due to the drawback of its high dimensionality, Quan et al. decided to incorporate Deep Learning (DL) into RL, developing the Deep Reinforcement Learning (DRL) method [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Turgay Celik.

Other authors proposed using Double Deep Reinforcement Learning (DDRL) to reduce the instability during the training process, resulting in a better navigation policy and reducing the overestimation of the behavioral policy [7]. However, the efficiency of traditional methods is based on the data obtained by their sensors. Therefore, the better the accuracy of the sensors, the higher the efficiency of the algorithm.

Considering the above, having different sensors in the same agent, such as lasers and radars, represents a high efficiency, as well as a high computational and economic cost. Consequently, processing and synchronizing data collected in real time represents a significant challenge. That is why, to reduce costs, Yue et al. carried out the mapping and optimal selection of the robot action from a Red Green Blue and Depth (RGB-D) camera [8], [9], [10], [11].

In order to provide a solution to this challenge, this study focuses on the homogenization of data obtained by a sensor (camera) with different viewing angles. The homogenization process is necessary because the data is sought to be compatible with the type of input data required by the planning algorithm published in previous works [12]. This algorithm is based on DDRL and uses a matrix of 24 laser scan data to feed the neural network. It is important to mention that the path planning algorithm will be used only to test the performance of the data homogenization and estimation proposed in this work.

Additionally, due to the need to develop new methods for sharing behavior policies with agents equipped with heterogeneous sensors, this work also addresses the estimation of unknown data using geometric considerations and data interpolation. This estimate reduces the computational capabilities required in the equipment inside each agent to process the data. Because, instead of processing the data obtained by a laser, a lower technology sensor such as a camera can be used, and achieve similar performance.

A. RELATED WORK

Current technology has enabled progress in the development of autonomous agents. However, their efficiency is affected by the data received by their sensors. Sensors represent the source of information about the environment and each provides different types of data depending on their defined structure. Therefore, working with several of them at the same time represents a huge challenge. To reduce this complexity, it is necessary to process these data simultaneously after homogenizing the output data.

Oliveira et al. developed an extrinsic calibration method for multiple heterogeneous sensors in intelligent vehicles integrated into the Robot Operating System (ROS). This method optimizes all the sensors, achieving precise calibrations that allow them to be used in parallel [13]. Continuing with the treatment of heterogeneous data from different sensors, Ratasich et al. proposed a configurable multi-sensor measurement fusion node implemented in ROS. This fusion package incorporates different fusion techniques and allows the concatenation of one-dimensional value sensors [14]. On the other hand, Kumar et al. focused on analyzing only two types of sensors, radar, and camera. To do this, they calibrate and fuse data from both heterogeneous sensors. However, they focus on synchronization, which is vital in applications such as autonomous driving. As this application requires the merged data to be valid and accurate at every point in time [15].

Kokovkina et al. proposed an algorithm combining synchronized camera and laser data to create a 3D model of the environment in which the robot is located [16]. Several authors have continued with this line of research, developing different methods for data integration and fusion of 3D camera data and 2D laser data, which are synchronized and implemented in real-time robotic applications such as object detection, localization, and navigation [17], [18], [19], [20], [21].

Ohnishi et al. presented a method of concatenating data from the same sensors to test the performance of the processing of heterogeneous data in Q-Learning (Q-L), Deep Q-Learning (DQL), and Double Deep Q-Learning (DDQL) algorithms [22]. Another alternative multisensor data fusion method using cubic interpolation was presented by Zhou et al., which solves the temporal alignment problems that occur when existing data from multiple data sources [23].

In order to use depth cameras for robot navigation, Sandfuchs et al. proposed an algorithm that transforms 3D depth images into a 2D laser data format, aiming at reducing the computational cost of processing 3D data [24]. Continuing with this approach, Faria et al. proposed to transform the data from two RGB-D cameras into a 2D laser format to be the input data of an artificial intelligence algorithm that creates estimated depth images that are used for robot mapping and navigation [25].

Surmann et al. focused their work on the autonomous navigation of robots in unknown environments using a Two-dimensional (2D) laser scanner and an RGB-D camera. The heterogeneous data processing from the two sensors was done by transforming the Three-dimensional (3D) data from the camera to the laser data format [26]. Whereas Yoo et al. proposed a method for planning trajectories in a real environment using a map of the current field obtained by interpolating real tidal current observation data [27]. Finally, Choi et al. suggested a method to select between different path-planning strategies by reinforcement learning adaptively. This is achieved by obtaining high interpolation accuracy over the explored environment, thus reducing the total distance of the path [28].

B. CONTRIBUTION OF THE PAPER

The proposed method allows the compatibility of the behavioral policies created by the different sensors. This leads to satisfactory performance and accuracy, regardless of the policy or sensor used by the agent. In summary, the contributions of this work are:

- A data synchronization and sorting algorithm.
- A data estimation strategy using geometric considerations.
- A data estimation strategy using linear interpolation.
- The source code of the algorithm proposed in this work is available at https://github.com/ELIZABETH1611/Homogenization.git.

The rest of the paper is organized as follows: Section II covers the technical background and formulation of the problem. Section III proposes the different phases for data processing, while Section IV presents the main characteristics of the method proposed in this paper for the estimation of unknown data. Section V discusses and compares the results obtained with the different sensors using the same planning algorithm. Section VI presents conclusions and future lines.

II. TECHNICAL BACKGROUND AND FORMULATION OF THE PROBLEM

This section briefly introduces laser scan data, point cloud data, interpolation, RL and DDQL techniques that are the foundation to understand the processing and homogenization of the data used in this work.

A. LASER SCAN

It is characterized by returning values corresponding to the distance from the center of the laser to surrounding obstacles. The distance data it provides depends on the angle of view of the laser and the separation between the laser single scans. This type of data represents one of the primary sources for many existing mapping and navigation techniques [29], [30].

B. POINT CLOUD

Point cloud data come from sensors such as cameras. They are characterized by containing a 3D point cloud in the viewing angle of the sensor. Each 3D point has its position in free space, with 32 bits of precision. Moreover, this data type is widely used in automatic modeling [31]. Considering that the objective is always to work with laser scan type data, it is necessary to specify the reference frame of the point cloud in order to get correct rotational translations [32].

C. INTERPOLATION

Interpolation methods use way-points to calculate a continuous trajectory [33]. In other words, it allows the estimation of data using neighboring data. This interpolation can be performed in different ways; in the case of linear interpolation, a smooth transition between neighboring points is obtained [34].

D. REINFORCEMENT LEARNING

The techniques of RL allow a system to learn using its experiences acquired by interacting with its environment. In other words, to achieve learning, the agent first obtains information about its environment and then acts based on that information. Whether the action is good or bad, it will receive a reward or a punishment. This process will be repeated until the agent achieves the desired learning [35].

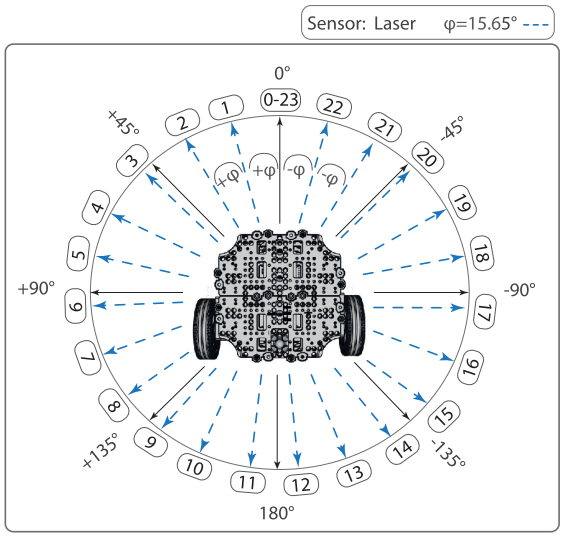


FIGURE 1. Distribution of the 360° laser vision through 24 single scans. Where φ represents the angle of separation between the single scans.

E. DOUBLE DEEP Q-LEARNING

In those applications where huge amounts of raw data are used as input, DL techniques combined with RL are applied [36]. The basic DDQL algorithm used two neural networks; one of them is called an online network and is the one that determines the action to be performed. While the other is called an objective network since it evaluates how good the performed action was. The online network is continuously trained to obtain an optimal behavioral policy, while the objective network receives constant updates from the online network, preventing overfitting of the policy [37].

III. DATA PROCESSING

A. DATA HOMOGENIZATION

This paper focuses on the homogenization of data collected from heterogeneous sensors. Such data will be used to train the behavioral policy of a path-planning algorithm. The algorithm requires a vector of 24 data (Y) as input. These data correspond to the distances read by the laser, which has a viewing angle of 360 degrees (see Fig. 1).

Since the objective is to obtain compatible behavioral policies between heterogeneous sensors, the data obtained by the camera (Point Cloud) are converted into laser data (Laser Scan). In other words, the 3D points obtained with the camera are transformed into a laser scan with 2D data. In each tested configuration, the camera and the laser are processed unless the agent has 360 degrees of vision. However, the laser data are only used to corroborate the collision detected by the camera or the estimated data, which will be explained in the following sections.

The homogenization process is carried out by a ROS node, where the source of data (camera) and the angle of view of the

sensor (θ) are indicated. The angle of view is decomposed into a minimum angle (θ_{min}), maximum angle (θ_{max}), and increment angle (θ_Δ) that defines the number of single scans (N) to be obtained from the input data. As a result of this process, a vector X composed of N laser scan samples will be obtained.

$$X = [sample[0], sample[1], \ldots, sample[N-1]]. \quad (1)$$

It is important to mention that for each sensor, the aim is to obtain a number of samples similar to the number obtained by the laser at that angle of view. Therefore, the equations shown below were designed taking into account the above-mentioned. For cases where the sensor has a viewing angle of fewer than 360 degrees, the following equations will be used:

$$\theta_{max} = (\frac{\theta}{2} \cdot \frac{\pi}{180}) + \frac{1}{10}, \quad (2)$$

$$\theta_\Delta = \frac{\theta \cdot \frac{\pi}{180}}{N-1}, \quad (3)$$

$$N = \frac{\theta}{15} + 2. \quad (4)$$

whereas for those sensors that provide a 360-degree viewing angle, a variation in the equations is made to ensure that both sensors provide the same total number of single scans.

$$\theta_{min} = \theta_{max} = \frac{\theta}{2} \cdot \frac{\pi}{180}, \quad (5)$$

$$\theta_\Delta = \frac{\theta \cdot \frac{\pi}{180}}{N}, \quad (6)$$

$$N = \frac{\theta}{15}. \quad (7)$$

Once the data homogenization process has been completed, the data received is synchronized and sorted.

B. DATA SYNCHRONIZATION

The synchronization module, as its name suggests, is responsible for synchronizing data when more than one sensor is used at the same time. For this purpose, a ROS node was used, which is responsible for synchronizing messages coming from two different data sources.

To start with the process, a vector Y is defined, which represents the 24 data used as input for the algorithm. These data are sorted in such a way that the first element of the vector corresponds to the scan at the 0.1-degree position and the last element corresponds to the scan at the -0.1 degree position (see Fig.2). Since the vector X, obtained from the homogenization process has a different order than the one used in the vector Y, the data is sorted as shown in (8).

$$\begin{aligned} Y = [&X[-1], X[-2], \ldots, X[-c_x], \\ &Z[0], Z[1], \ldots, Z[N_u - 1], \\ &X[0], X[1], \ldots, X[c_x]]. \end{aligned} \quad (8)$$

As the above equation defines, the vector Y is formed by the known values represented by the vector X and the unknown values which are defined by the vector Z. The number of unknown values (N_u), in other words, the number of missing data to form an input vector of 24 data is determined by (9). Whereas the angle of separation between each one of them (ξ) is determined by (10).

$$N_u = 24 - N, \quad (9)$$

$$\xi = \frac{\Gamma}{N_u}. \quad (10)$$

To calculate ξ the first step is to define the vectors which contain the angular position of each of the known values (X). Where the first one defines the positions of positive angles (φ_l) in the range of zero degrees to $\theta/2$. The second one defines the positions of negative angles (φ_s) in the range of zero degrees to $-\theta/2$ (see Fig.2). While the size of each vector is defined by c_x which represents the central position of X.

$$c_x = \frac{N}{2} - 1, \quad (11)$$

$$\varphi_l = \theta_\Delta * [0, 1, \ldots c_x], \quad (12)$$

$$\varphi_s = -\varphi_l. \quad (13)$$

Then, the unobserved angle Γ (14) corresponding to the area not covered by the sensor is calculated by taking into account the last positive and negative angle of (12), and (13), respectively.

$$\Gamma = 180 + |90 - \varphi_l[-1]| + |-90 - \varphi_s[-1]|, \quad (14)$$

Finally, the vector B represents the angular position of each of the unknown values.

$$B = [\varphi_l[-1] + \xi] + ([0, 1, \ldots \frac{\Gamma + \varphi_l[-1] - \xi}{\xi}] * \xi). \quad (15)$$

Therefore, the final distribution of the data after sorting can be seen in Fig. 2. While the process to obtain the initial Y data vector formed by 24 values is summarized in the flow chart of Fig. 3. Additionally, Table 1 summarizes the main parameters of each of the implemented configurations, and the features configured within the simulation are listed in Table 2.

IV. ESTIMATION OF UNKNOWN DATA

It is known that large data sets influence the performance of the learned policy and reduce overfitting. For instance, data augmentation techniques are used in image classification to accelerate the learning process [38], [39], [40].

A 360-degree view of the environment provides more information about it than a 120-degree view. Therefore, in the first case, the route planning algorithm will need less training time to obtain an optimal policy since it has more data about the environment. In order to accelerate this learning process for agents that have a limited view of the environment (less data), a method is proposed in this paper. This method allows a reliable estimation of the missing data by geometric estimation and interpolation.

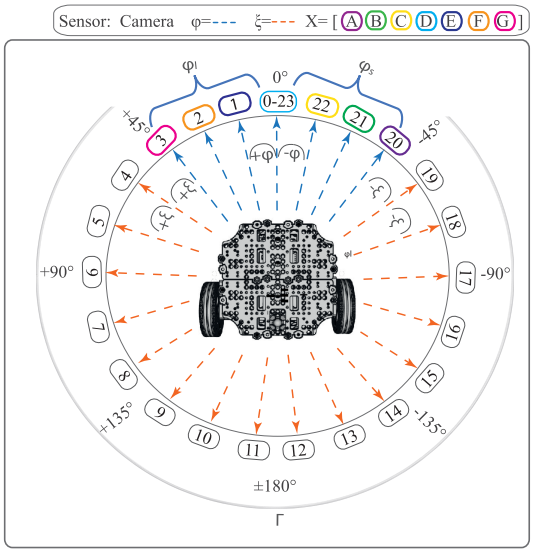


FIGURE 2. Distribution of the 24 single scans, using a 90-degree camera, where the eight known data (*X*) are represented in different colors. On the other hand, the unknown data (*Z*) are represented by a single color. However, since the known data have different sort order, the colors represent the position of each of the data within the data vector obtained by the sensor (*X*). Additionally, the angle of view that is not covered by the sensor (Γ) is represented.

TABLE 1. Configurations of the sensors used, specifying their minimum, maximum, increment, and total angle. Also, the number of known data, unknown data, and the position of the known data within the vector is used to feed the neural network.

Sensor	θ	θ_{min}	θ_{max}	θ_{Δ}	N	N_u	Positions of N
Camera	90º	-0.68	0.88	0.22	8	16	[0:3] [20:23]
Camera	120º	-0.95	1.14	0.23	10	14	[0-4] [23-19]
Camera	150º	-1.20	1.40	0.23	12	12	[0-5] [23-18]
Camera	180º	-1.45	1.65	0.23	14	10	[0-6] [23-17]
Camera	360º	-1.55	1.55	0.25	24	0	[0-24]
Laser	360º	0.00	π	0.27	24	0	[0-24]

Consequently, even if the agent has a camera with a viewing angle of less than 360 degrees, after applying the proposed method, the agent will have a very close estimation of the part of the environment that the sensor cannot observe.

A. GEOMETRIC CONSIDERATIONS

As each sensor has a different viewing angle, this implies that as the viewing angle increases, the number of known data will increase, while the number of unknown data will decrease. Therefore, the more known data available, the better the estimation of the unknown data.

The first phase of data estimation is performed using trigonometry and the data read by the sensor in the previous step. To do this, it is taken into account that linear and angular actions define the agent's movement. Therefore, the estimation of the data will vary depending on whether its motion is affected by linear velocity or a combination of both.

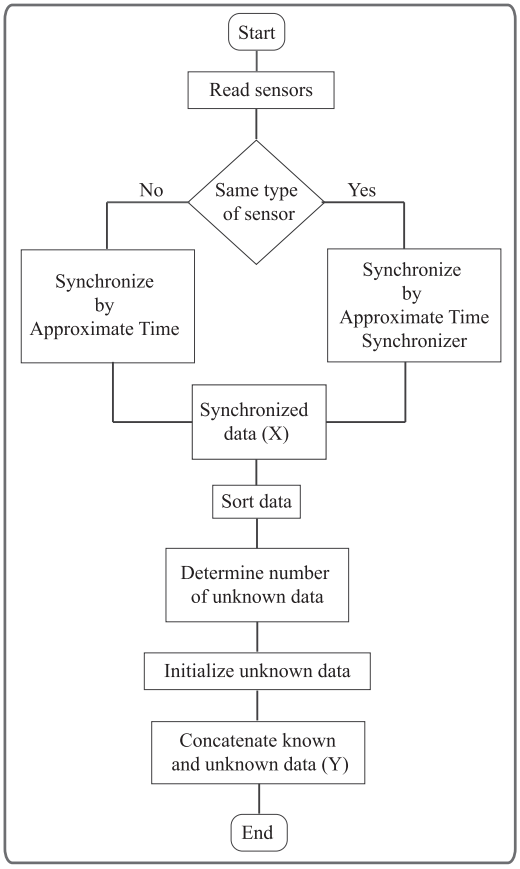


FIGURE 3. Synchronization and sorting process of the data obtained by the sensor. In addition, the process of the concatenation of the unknown with the known data is described in order to obtain a vector with 24 data.

1) LINEAR MOTION

As mentioned above, for the estimation of the unknown values, it is necessary to have knowledge of the data read by the sensor in the previous step. In addition, the current and previous position of the agent is also needed to accurately calculate the distance traveled.

The process consists of calculating the angle (η) formed between the current direction and position of the agent towards each of the different known points obtained by the sensor reading in the previous step. If the η is lower than θ of the agent, the position is discarded, but if η is higher, the corresponding beam (single scan) is calculated.

$$\eta = (\pi - \rho) \cdot sgn\, \varphi. \tag{16}$$

So the value of the beam will correspond to the distance from the current position of the agent to the reference point. In order to calculate (η), first the distance traveled by the agent is calculated by (17), then the distance x between the

TABLE 2. Main characteristics of the sensors used, such as the characteristics of the images obtained by the camera, type of noise, and standard deviation of the laser.

Sensor	Type	Amount	Image			Noise		
			Width	Height	Format	Type	σ	Resolution
Camera 90° Camera 120° Camera 150° Camera 180° Camera 360°	Realsense R200	1 2 (180°)	1920	1080	R8G8B8			
Laser 360°	HLS-LFCD-LDS	1				Gaussian	0.01	0.015

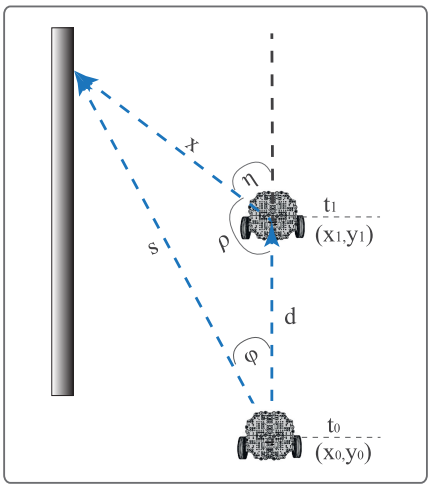


FIGURE 4. Calculation of the different angles and distances used in the estimation of unknown data from the input data vector when the agent performs linear motion actions.

new position of the robot (t_1) to one of the known points (s) from the previous step (t_0) is calculated.

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}, \tag{17}$$

$$x = \sqrt{s^2 + d^2 - (2sd \cdot \cos\ |\varphi|)}. \tag{18}$$

As seen in Fig. 4, the distances form a triangle, where φ corresponds to the beam angle of the data position used as the reference point. On the other hand, ρ is the complementary angle of η. To calculate it, use (19), and (16), respectively. This entire process is described in the Algorithm 1.

$$\rho = \cos^{-1}\left(\frac{d^2 + x^2 - s^2}{2dx}\right). \tag{19}$$

2) ANGULAR MOTION

The process performed to estimate the values is similar to the one used for linear motion, with the difference that the agent will perform a uniform circular movement since it is affected by a linear velocity (v) and angular velocity (w).

$$r = \frac{v}{w}. \tag{20}$$

Algorithm 1 Data Estimation Using Linear Motions

```
Read the array of data from the previous step (Y')
Calculate the distance traveled (d)
for Each data (s) in Y' do
    Calculate the distance x between s and the current position
    Calculate η
    if η < θ then
        pass
    else
        Search compatible beam angle
        Replace in Y the position corresponding to the compatible angle for the value of x
    end if
end for
return Y
```

Therefore, first, the radius (r) of the circle is calculated. Then the angle of rotation (Θ) during the action time (t_a) is calculated by taking the center of the circle as the point of origin and using (21).

$$\Theta = |wt_a|. \tag{21}$$

Considering that d, r, r forms an isosceles triangle as shown in Fig. 5, λ can be computed by (22).

$$\lambda = \frac{\pi - \Theta}{2}. \tag{22}$$

Consequently, η is calculated using (28). And the process of calculating the estimated values in the turning actions is shown in the Algorithm 2.

$$\beta = \frac{\pi}{2} - \lambda, \tag{23}$$

$$\alpha = |\varphi| + \beta, \tag{24}$$

$$x = \sqrt{s^2 + d^2 - (2sd \cdot \cos\ \alpha)}, \tag{25}$$

$$\zeta = \cos^{-1}\left(\frac{d^2 + x^2 - s^2}{2dx}\right), \tag{26}$$

$$\rho = \zeta - \beta, \tag{27}$$

$$\eta = (\pi - \rho) \cdot sgn\ \varphi. \tag{28}$$

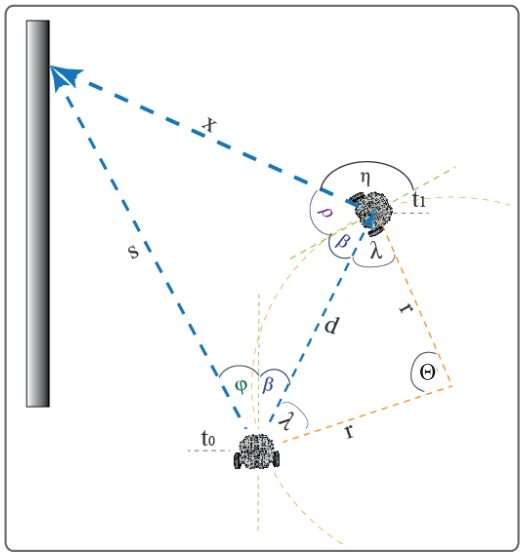


FIGURE 5. Calculation of the different angles and distances used to estimate unknown data from the input data vector when the agent performs angular motion actions.

Algorithm 2 Data Estimation Using Angular Motions

```
Read the array of data from the previous step (Y')
Calculate the distance traveled (d)
Calculate the radius of the circle (r)
Calculate the angle of rotation (Θ)
Calculate the internal angle (λ) formed by d, r, r
for Each data (s) in Y' do
    Calculate the distance x between s and the current position
    Calculate η
    if η < θ then
        pass
    else
        Search compatible beam angle
        Replace in Y the position corresponding to the compatible angle for the value of x
    end if
end for
return Y
```

3) ESTIMATED BACK VALUES

Due to the fact that the different camera configurations have a viewing angle of 180 or less, it is not possible to estimate the back beams using the above methods. This is because the angle of view only covers the front beams, which allows only side beams to be estimated.

To estimate the values of the back beams ($Y[\varphi]$), which correspond to positions 10 to 13 (Fig. 2) the initialization value is used as the starting value(s). Since the agent's starting point is located at a collision-free distance, the initialization value was assigned a distance less than the maximum distance read by the sensor (3.5). Then, at each step, the values assigned to each beam will be updated with the value corresponding to

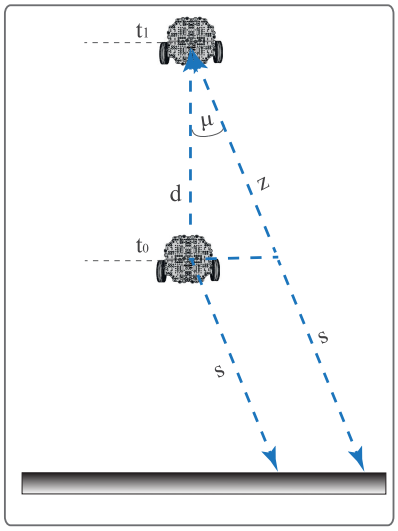


FIGURE 6. Variables calculated to obtain the estimated back values.

the sum or subtraction of the distance (z) and the value of the current beam. This distance is added or subtracted depending on the angle of rotation and the direction of their velocities (see Fig. 6).

$$Y[\varphi] = s + z. \tag{29}$$

It is important to mention that, as in the previous methods, s is the distance value for each of the single scans obtained in the previous step. On the other hand, d is the distance that the robot has traveled when performing the action, and μ is the angle that the beam of the analyzed back position forms with the 180-degree angle of the robot.

In addition, z represents the distance added or subtracted in each beam in t_1 with respect to t_0. Algorithm 3 summarizes the process it performs to estimate these values.

$$\mu = |180 - \varphi|, \tag{30}$$

$$z = \frac{d}{cos\ \mu}. \tag{31}$$

B. DATA INTERPOLATION

Thanks to geometric estimation, more data are known in this part of the process and can be used to estimate the remaining data by interpolation (see Fig. 7).

For this purpose, linear interpolation (see Fig. 8) is used, as it allows us to approximate any position within the domain defined by the input data. Moreover, given that data will always correspond to the front and rear beams, the interpolation will be more accurate.

In order to evaluate the different policies obtained, five tests will be carried out. The performance obtained with a laser will be taken as a general reference.

V. RESULTS DISCUSSION

In those scenarios where it is intended to have a global behavior policy for different agents or when they can exchange

Algorithm 3 Data Estimation of Back Values

for Each step **do**
  if step < 1 **then**
    Initialize the distance values of positions $Y[10:14]$ to a collision-free value, less than 3.5 (laser maximum).
  else
    Read the array of data from the previous step (Y')
  end if
  Calculate the distance traveled (d)
  for Each data (i) in $Y'[10:14]$ **do**
    s= Data from the previous step ($Y'[i]$)
    Calculate the angle μ
    Calculate the distance z between s and the current position
    $s' = s + z$
    Replaces s' at the corresponding position in Y (Eq. (29))
  end for
end for

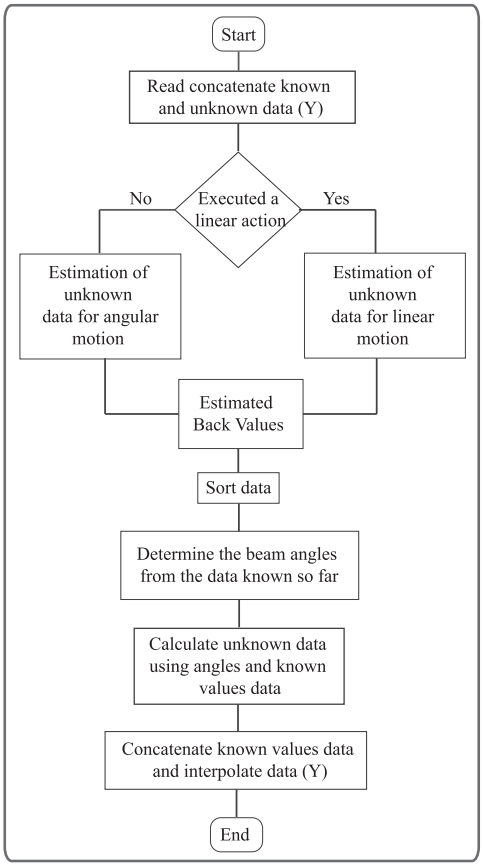


FIGURE 7. Process of interpolation of the remaining unknown data using the known data so far.

their learned policies, the agents must have the same type of sensors or, in the case of using other sensors, their data must go through a homogenization process. If, after the homogenization process, the data obtained are less than those needed to feed the neural network, a data estimation process is required.

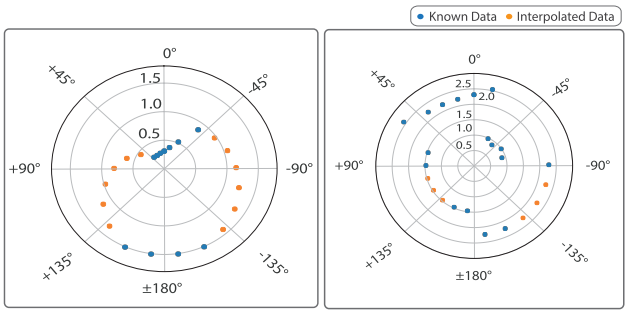


FIGURE 8. Interpolation of the remaining unknown data using the known data and the data obtained by geometric considerations.

In this work, the performance obtained by a laser with a 360-degree view of the environment within a route planning algorithm was compared with that obtained by cameras with five different viewing angles. The method proposed in this paper of homogenization and data estimation was applied to these cameras in order to satisfy the requirements of the path planning algorithm used.

In the first four cases, only one camera is used, while in the last configuration, two cameras are used, each with a 180-degree viewing angle (Table 1 and Table 2).

As previously mentioned, the same path-planning algorithm was used in all tests, which requires an input vector of 24 laser scan data. In addition, the behavioral policy used for the comparisons was obtained after an eight-hour training. The training and test environment contains sixteen objectives located in different parts of the environment, each at a different distance from the agent's starting point. Hence, an agent with an optimal policy will need less time to reach a close objective than one that is in a distant position.

Since a large data set will help the algorithm learn faster and the lack of data will reduce the learning efficiency, this work aims to have a learning rate similar to that achieved by a laser but using sensors with smaller viewing angles and with the estimation of unknown data.

As can be seen in Fig. 9, all sensors achieved more than 70% of their task during the whole training process. Note that even if an episode has a certain duration (500 steps) the total number of objectives to be reached during each episode will be different since the sampling of the objectives is random. This means that in those training phases in which the closest and lower difficulty objectives are sampled more often, as long as the trained policy is optimal, the agent will have a higher success rate.

As can be seen in Fig. 9 in the case of the two 180-degree cameras, which in this paper is denoted as a 360, the first observation from the training sessions is that the algorithm learns faster when it has a smaller number of estimated data. It has a higher number of targets achieved (successes) during the entire training session. This is because the sensor has

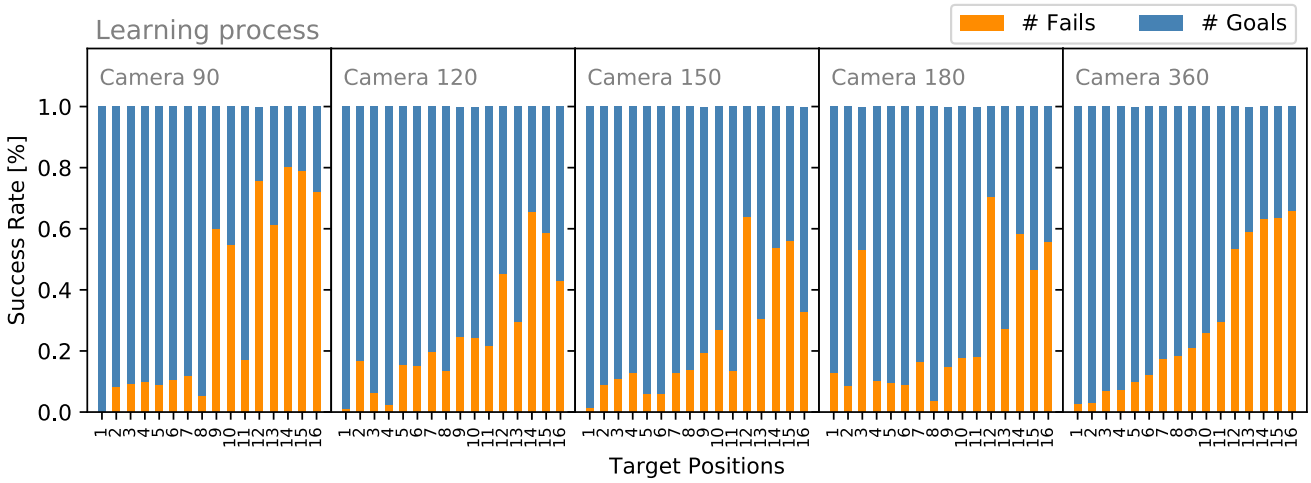


FIGURE 9. Number of goals achieved (successes) and the number of collisions the agent has had during the training process. The training was carried out for 8 hours, in the same environment and with the same targets but randomly selected. In total, there were 16 different target positions used for training and testing, which are represented on the x-axis.

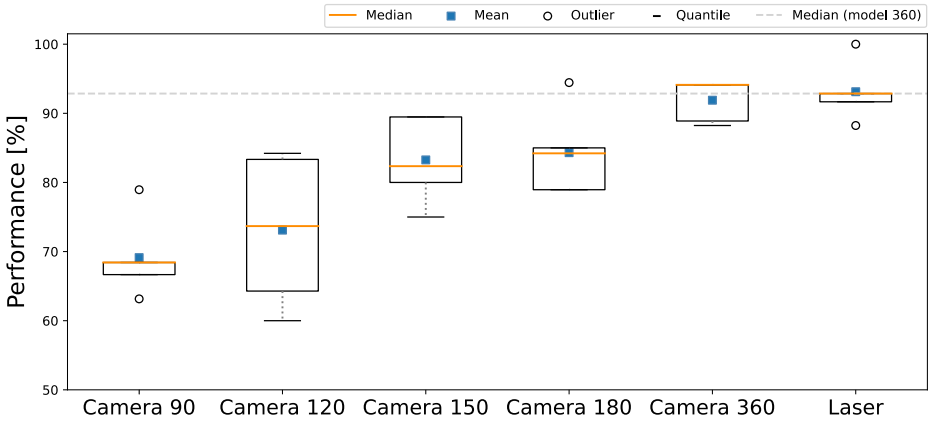


FIGURE 10. Performance obtained for each of the different viewing angles of the different cameras compared to that obtained by a laser.

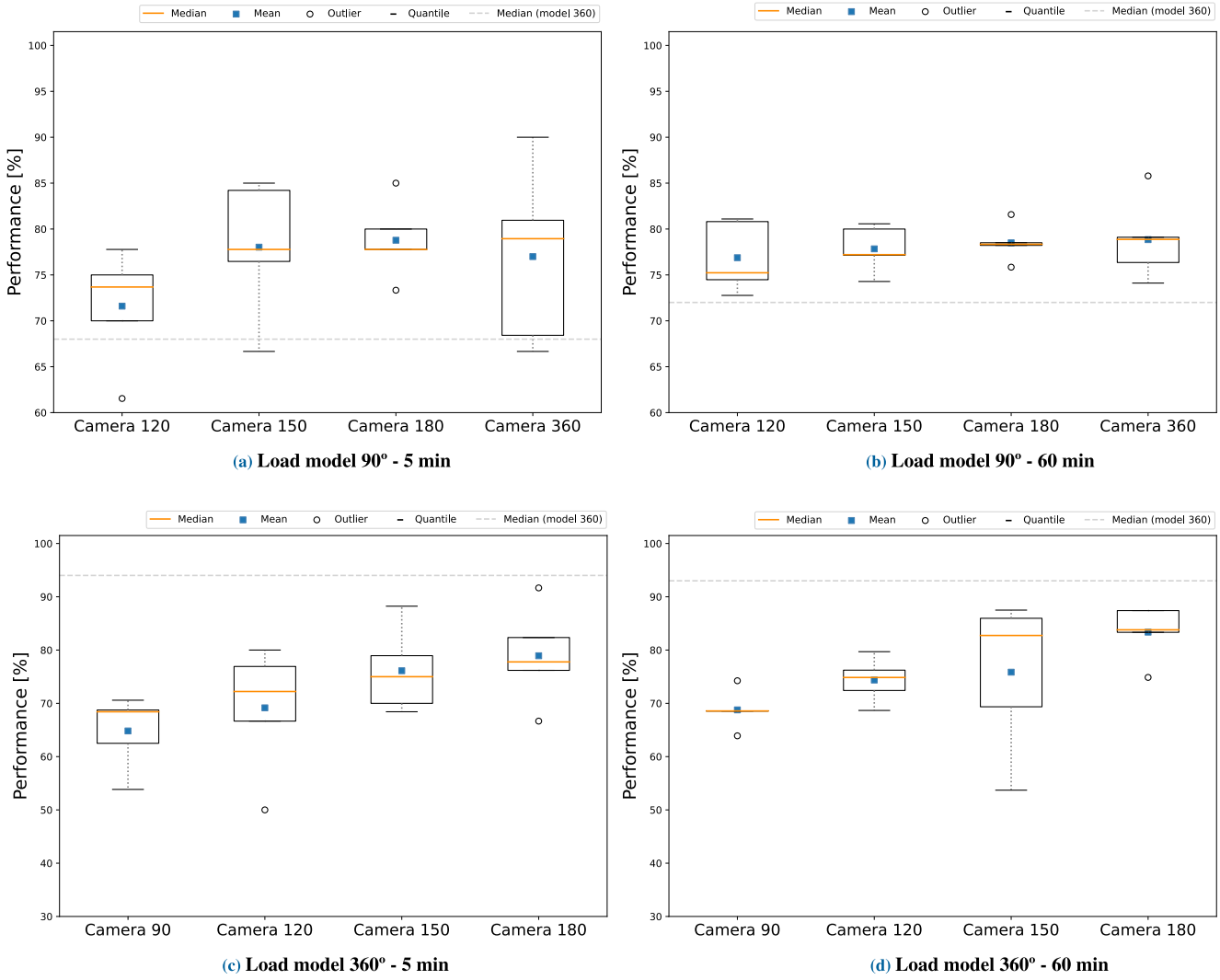
a complete view of the environment, making its data more accurate.

For instance, if an agent is provided by a sensor whose angle of view does not correspond to a complete view of the environment, it will need more time and, therefore, more episodes to reach its optimal policy. However, thanks to the estimation of unknown data, this agent will have an estimated view of the whole environment, and their learning process will be faster.

First Test: The optimal behavioral policy obtained by each sensor is loaded five times during five minutes (see Fig. 10). As a result, the best-performing optimal behavior policy corresponds to the 360-degree camera. Given that it has a median success rate of 94%, similar to that obtained with the reference policy of comparison (93% - laser). This is because the distance values corresponding to the 24 individual scans are calculated directly from the data collected by the camera without requiring a previous data estimation process since all 24 values are known.

Regarding the other configurations, the 90-degree camera has the lowest success rate, with 68%. In contrast, the remaining configurations show a 74%, 82%, and 84% of success rate, respectively. In these configurations, it has been necessary to estimate the unknown values and it can be demonstrated that these estimated values are very close to reality since the behavioral policy achieves a percentage of success of more than 70%.

Second Test: In contrast, in this test the optimal performance policy obtained by the 90-degree camera is loaded five times for five minutes on each of the other sensors (see Fig. 11a). This test is used to check the compatibility of loading the behavior policy obtained by a sensor with a different viewing angle on an agent using a different sensor. The results show that the 120-degree camera maintains the same rate as when using its model, corresponding to an average success rate of 74%. This is because there are only two data differences between the number of known data for a 90-degree camera and a 120-degree camera.



(a) Load model 90º - 5 min

(b) Load model 90º - 60 min

(c) Load model 360º - 5 min

(d) Load model 360º - 60 min

FIGURE 11. The subfigures show: a-b) Comparison of the performance obtained using the 90-degree policy on the rest of the sensors, c-d) Comparison of the performance obtained using the 360-degree policy on the rest of the sensors. The gray line represents the median of the policy loaded on its own sensor.

TABLE 3. Summary of the medians obtained when the agent uses the model created by its sensor and when using models created by a sensor with higher vision and one with lower vision.

Sensor	θ	time (min)	model load	Mdn [%]	model load	Mdn [%]	model load	Mdn [%]	time (min)	model load	Mdn	model load	Mdn [%]
Camera	90º		90º	68		68		68			69		72
Camera	120º		120º	74		72		74			75		76
Camera	150º	5	150º	82	360º	75	90º	78	60	360º	83	90º	77
Camera	180º		180º	84		78		78			84		78
Camera	360º		360º	94		94		79			93		79
Laser	360º		360º	93	-	-	-	-	-	-	-	-	-

Although the other sensors do not reach the median achieved with their own models (see Fig. 10), they have a median of over 78%. This shows that the models are compatible since a high success rate is achieved even if the agent uses a model created with a sensor with a viewing angle limited to a few degrees.

This shows that the models are compatible and a high success rate is achieved even if a model created with a sensor with a low viewing angle is used.

Third Test: The same process is performed as in the previous one, with the difference that now the tests have a duration of one hour. This is done in order to analyze how

the performance of the loaded model is affected by the new data used to feed the neural network, which is obtained from a different sensor than the one from which the policy was trained. Since the robot runs for a longer period of time, i.e. one hour instead of five minutes as in the previous case, the policy will be fitted using this data that can make it better or worse. The results (see Fig. 11b) show that the data estimation is good since the median in all cases is equal to or higher than 76%.

Fourth Test: The process followed in this test is the same as in the second one, but now the optimal performance policy loaded is the 360-degree camera policy, with the objective of analyzing the success rate achieved when the agent uses a model created with a sensor that provides the 24 distance values directly from the sensor. In each configuration, the optimal performance policy is loaded five times for five minutes to ensure that the results are consistent and have not been altered by random events (see Fig. 11c). The results have shown that the median hit rate of the camera configurations with the smallest viewing angle has a median of 68%, while the median hit rate for the other sensors is over 72%.

Fifth Test: The same policy that was used in the last test is loaded five times for one hour (see Fig. 11d). The results obtained are similar to those obtained in the previous test in which the agent with a 90-degree camera achieves a success rate of 69%, and with the rest of the sensors, it is higher than 75%.

VI. CONCLUSION

The homogenization and synchronization processes are necessary when information about the environment from more than one sensor is fed to the agent because, in this way, the data would be collected in the same format and time.

In order to obtain compatible policies between heterogeneous sensors, a method of homogenization and data estimation was proposed. As a result, it was shown that, regardless of the loaded policy, a median above 71% was achieved for all sensors with a viewing angle higher than 90 degrees. Furthermore, with the proposed method, it is possible to use cameras with different viewing angles and use them in algorithms designed for laser scan data.

On the other hand, it was observed that the data estimation process helps to learn the behavioral policy much faster. This can be seen in Fig. 9, where sensors with viewing angles less than 180 degrees have a similar success rate in the same number of hours. Therefore, with a 180-degree camera, an efficiency of 84% is achieved, the computational resources needed by the agent to process the agent's data are reduced, and the economic cost of implementing these use cases is reduced.

On the other hand, if higher accuracy is required, it is possible to use two 180-degree camera-type sensors since this configuration achieves the same percentage of accuracy as with a laser while using fewer resources.

As a future work, the homogenization method is intended to be used in the case of the use of collaborating agents with heterogeneous sensors is proposed.

REFERENCES

[1] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1343–1350.

[2] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: A survey," *Auto. Robots*, vol. 46, no. 5, pp. 569–597, Jun. 2022.

[3] T. Zhang, Q. Li, C.-S. Zhang, H.-W. Liang, P. Li, T.-M. Wang, S. Li, Y.-L. Zhu, and C. Wu, "Current trends in the development of intelligent unmanned autonomous systems," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 1, pp. 68–85, Jan. 2017.

[4] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1527–1533.

[5] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," 2017, *arXiv:1702.01182*.

[6] Y. Liu, Z. Chen, Y. Li, M. Lu, C. Chen, and X. Zhang, "Robot search path planning method based on prioritized deep reinforcement learning," *Int. J. Control, Autom. Syst.*, vol. 20, no. 8, pp. 2669–2680, Aug. 2022.

[7] M.-F.-R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, p. 2748, Dec. 2022.

[8] P. Yue, J. Xin, H. Zhao, D. Liu, M. Shan, and J. Zhang, "Experimental research on deep reinforcement learning in autonomous navigation of mobile robot," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 1612–1616.

[9] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," in *Proc. 4th Student Conf. Res. Develop.*, Jun. 2006, pp. 183–188.

[10] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," *IEEE Trans. Syst., Man, Cybern., B*, vol. 39, no. 3, pp. 672–689, Jun. 2009.

[11] Q. Yang and S.-J. Yoo, "Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms," *IEEE Access*, vol. 6, pp. 13671–13684, 2018.

[12] E. Palacios-Morocho, S. Inca, and J. F. Monserrat, "Multipath planning acceleration method with double deep R-learning based on a genetic algorithm," *IEEE Trans. Veh. Technol.*, early access, May 26, 2023, doi: 10.1109/TVT.2023.3277981.

[13] M. Oliveira, A. Castro, T. Madeira, P. Dias, and V. Santos, "A general approach to the extrinsic calibration of intelligent vehicles using ROS," in *Proc. 4th Iberian Robotics Conf.* Cham, Switzerland: Springer, 2020, pp. 203–215.

[14] D. Ratasich, B. Frömel, O. Höftberger, and R. Grosu, "Generic sensor fusion package for ROS," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 286–291.

[15] R. Kumar and S. Jayashankar, "Radar and camera sensor fusion with ROS for autonomous driving," in *Proc. 5th Int. Conf. Image Inf. Process. (ICIIP)*, Nov. 2019, pp. 568–573.

[16] V. A. Kokovkina, V. A. Antipov, V. P. Kirnos, and A. L. Priorov, "The algorithm of EKF-SLAM using laser scanning system and fisheye camera," in *Proc. Syst. Signal Synchronization, Generating Process. Telecommun. (SYNCHROINFO)*, Jul. 2019, pp. 1–6.

[17] S. Syntakas, K. Vlachos, and A. Likas, "Object detection and navigation of a mobile robot by fusing laser and camera information," in *Proc. 30th Medit. Conf. Control Autom. (MED)*, Jun. 2022, pp. 557–563.

[18] M. Basavanna, M. Shivakumar, K. R. Prakash, and P. Bhomkar, "ROS based 3D mapping of an indoor environment using fusion of orbbec astra camera and LiDAR on turtlebot mobile robot," in *Proc. 5th Int. Conf. Electr., Electron., Commun., Comput. Technol. Optim. Techn. (ICEECOT)*, Dec. 2021, pp. 323–327.

[19] A. Gunatilake, L. Piyathilaka, S. Kodagoda, S. Barclay, and D. Vitanage, "Real-time 3D profiling with RGB-D mapping in pipelines using stereo camera vision and structured IR laser ring," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 916–921.

[20] J. Zou, H. Zheng, and F. Wang, "Real-time target detection system for intelligent vehicles based on multi-source data fusion," *Sensors*, vol. 23, no. 4, p. 1823, Feb. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/4/1823
[21] H. Bozorgi, X. T. Truong, H. M. La, and T. D. Ngo, "2D laser and 3D camera data integration and filtering for human trajectory tracking," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2021, pp. 634–639.
[22] S. Ohnishi, E. Uchibe, Y. Yamaguchi, K. Nakanishi, Y. Yasui, and S. Ishii, "Constrained deep Q-learning gradually approaching ordinary Q-learning," *Frontiers Neurorobotics*, vol. 13, p. 103, Dec. 2019.
[23] T. Zhou, M. Chen, and J. Zou, "Reinforcement learning based data fusion method for multi-sensors," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 6, pp. 1489–1497, Nov. 2020.
[24] S. Sandfuchs, M. P. Heimbach, J. Weber, and M. Schmidt, "Conversion of depth images into planar laserscans considering obstacle height for collision free 2D robot navigation," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Aug. 2021, pp. 1–6.
[25] J. M. Faria and A. H. J. Moreira, "Implementation of an autonomous ROS-based mobile robot with AI depth estimation," in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2021, pp. 1–6.
[26] H. Surmann, C. Jestel, R. Marchel, F. Musberg, H. Elhadj, and M. Ardani, "Deep reinforcement learning for real autonomous mobile robot navigation in indoor environments," 2020, *arXiv:2005.13857*.
[27] B. Yoo and J. Kim, "Path optimization for marine vehicles in ocean currents using reinforcement learning," *J. Mar. Sci. Technol.*, vol. 21, no. 2, pp. 334–343, Jun. 2016.
[28] T. Choi and G. Cielniak, "Adaptive selection of informative path planning strategies via reinforcement learning," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Aug. 2021, pp. 1–6.
[29] G. Vosselman, "Fusion of laser scanning data, maps, and aerial photographs for building reconstruction," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jun. 2002, pp. 85–88.
[30] R. Wang, X. Li, and S. Wang, "A laser scanning data acquisition and display system based on ROS," in *Proc. 33rd Chin. Control Conf.*, Jul. 2014, pp. 8433–8437.
[31] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.
[32] A. Luis Ballardini, S. Fontana, A. Furlan, and D. G. Sorrenti, "Ira_laser_tools: A ROS LaserScan manipulation toolbox," 2014, *arXiv:1411.1086*.
[33] L. Butyrev, T. Edelhäußer, and C. Mutschler, "Deep reinforcement learning for motion planning of mobile robots," 2019, *arXiv:1912.09260*.
[34] E. L. Dan, M. Dînsoreanu, and R. C. Muresan, "Accuracy of six interpolation methods applied on pupil diameter data," in *Proc. IEEE Int. Conf. Autom., Quality Test., Robot. (AQTR)*, May 2020, pp. 1–5.
[35] Z. Ding, Y. Huang, H. Yuan, and H. Dong, *Introduction to Reinforcement Learning*. Singapore: Springer, 2020, pp. 47–123.
[36] S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: An overview," in *Proc. SAI Intell. Syst. Conf.*, Jun. 2018, pp. 426–440.
[37] W. Zaher, A. W. Youssef, L. A. Shihata, E. Azab, and M. Mashaly, "Omnidirectional-wheel conveyor path planning and sorting using reinforcement learning algorithms," *IEEE Access*, vol. 10, pp. 27945–27959, 2022.
[38] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022.
[39] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif. Intell. Rev.*, vil. 55, pp. 2351–2377, Mar. 2022.
[40] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 13001–13008.

ELIZABETH PALACIOS-MOROCHO (Student Member, IEEE) received the degree in electronics and telecommunications engineering from the University National of Loja (UNL), in 2018, and the M.Sc. degree in systems technologies and communication networks from Universitat Politècnica de València (UPV), in 2019. She is currently participating as a Predoctoral Researcher with the iTEAM-UPV—Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València.

PABLO LÓPEZ-MUÑOZ received the double degree in telecommunications engineering and in business administration from Universitat Politècnica de València (UPV), in 2023. He was a Research Engineer in a six-month internship with the iTEAM-UPV—Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València. His research interest includes data processing for artificial intelligence algorithms.

MANUEL A. COSTÁN received the double degree in telecommunications engineering and in business administration from Universitat Politècnica de València (UPV), in 2022. He was a Research Engineer in a six-month internship with the iTEAM-UPV—Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València, in 2022. His research interests include vehicular communications, mobility, and data processing for artificial intelligence algorithms.

JOSE F. MONSERRAT (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in telecommunications engineering from Universitat Politècnica de València (UPV), in 2003 and 2007, respectively. He holds six patents. He has also published more than 50 journal articles. He was a recipient of the First Regional Prize of Engineering Studies, in 2003, for his outstanding student record, also receiving the Best Thesis Prize from UPV, in 2008. In 2009, he was awarded the best young researcher prize in Valencia. In 2016, he received the merit medal from the Spanish Royal Academy of Engineering in the young researcher category.