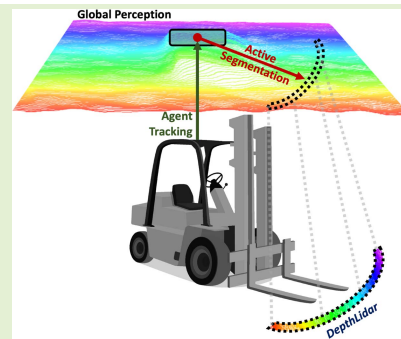


DepthLiDAR: Active Segmentation of Environment Depth Map Into Mobile Sensors

Marcelo Limeira, Luis Piardi^{ID}, Vivian Cremer Kalempa^{ID}, Paulo Leitão^{ID}, *Senior Member, IEEE*,
and André Schneider de Oliveira^{ID}, *Member, IEEE*

Abstract—This paper presents a novel approach for creating virtual LiDAR scanners through the active segmentation of point clouds. The method employs top-view point cloud segmentation in virtual LiDAR sensors that can be applied to the intelligent behavior of autonomous agents. Segmentation is correlated with the visual tracking of the agent for localization in the environment and point cloud. Virtual LiDAR sensors with different characteristics and positions can then be generated. This method is referred to as the *DepthLiDAR* approach, and is rigorously evaluated to quantify its performance and determine its advantages and limitations. An extensive set of experiments is conducted using real and virtual LiDAR sensors to compare both approaches. The objective is to propose a novel method to incorporate spatial perception in warehouses, aiming to achieve Industry 4.0. Thus, it is tested in a low-scale warehouse to incorporate realistic features. The analysis of the experiments shows a measurement improvement of 52.24% compared to the conventional LiDAR.

Index Terms—Virtual sensors, LiDAR, point cloud, active segmentation, Industry 4.0.



I. INTRODUCTION

INDUSTRY 4.0 philosophy was envisaged to promote the advancement of manufacturing, achieve a shorter product life cycle and extreme mass customization in an economical and efficient process [1], [2]. It increases the overall level of industrialization, informatization, and digitalization of manufacturing systems, aiming to achieve greater efficiency, competency, and competitiveness [3]. As one of the essential items in Industry 4.0, smart factories integrate several technologies, making them more complex and precise for the improvement of aspects such as the performance, quality, and transparency of manufacturing processes [4]. The transition from traditional to intelligent manufacturing is believed to have

a profound and lasting impact on the global manufacturing industry [1]. However, it presents a significant challenge when converting stand-alone automated equipment into intelligent interconnected agents.

Industrial environments are evolving because of emerging technologies, increasing their automation level and the incorporation of robotic devices. Mobile robots can adapt to manufacturing changes, meeting the sector's diverse needs, and incorporating new techniques [5], [6]. Main manufacturers have begun the journey towards implementing smart robots in factories environments. Amazon is a prominent example with the amazon robots [7]; other examples are Ocado [8] and DHL [9]. Owing to their flexibility, autonomous robots are one of the pillars to achieving Industry 4.0 requirements; however, they need new skills to interact with this dynamic environment.

One of the challenges in smart factories is to have mobile robots carry loads from one point to another, be self-organized and able to use all the free space in the factory [10]. To autonomously perform tasks in the environment, mobile robots must be able to self-locate, find a safe navigation path, and reach the desired location in an unstructured environment [11]. This behavior is opposed to that of the standard automated guided vehicle (AGV), which normally follow ground fixed lines. A novel vehicle class must then be adopted to explore the benefits of Industry 4.0, including comprehensive and complex perception systems, intelligent and adaptable behaviors, and dynamic management.

Autonomous behaviors are highly dependent on the reliability, detailing, and description of the environment.

Manuscript received May 7, 2021; accepted June 7, 2021. Date of publication June 9, 2021; date of current version August 31, 2021. This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES)—Finance Code 001 and in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). The associate editor coordinating the review of this article and approving it for publication was Prof. You Li. (*Corresponding author: Vivian Cremer Kalempa.*)

Marcelo Limeira and André Schneider de Oliveira are with the Graduate Program in Electrical and Computer Engineering, Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba 80230-901, Brazil (e-mail: limeira@alunos.utfpr.edu.br).

Luis Piardi and Paulo Leitão are with the Research Center in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança (IPB), Campus de Santa Apolónia, 5300-253 Bragança, Portugal.

Vivian Cremer Kalempa is with the Department of Information Systems, Santa Catarina State University (UDESC), Florianópolis 89283-081, Brazil (e-mail: vivian.kalempa@udesc.br).

Digital Object Identifier 10.1109/JSEN.2021.3088007

Simple collision avoidance behaviors are limited to perception capability. Smart warehouses aim to achieve optimized and automatic load management using autonomous agents. In this case, the traditional forklifts must be updated to execute autonomous and cooperative behaviors.

Most industrial mobile robots base their perception of the environment on a laser range finder (LRF) sensor, also known as a light detection and ranging (LiDAR) scanner. These sensors are preferred owing to their high resolution and fidelity, which are slightly affected by variations in illumination and temperature, thus ensuring accurate and robust measurements for almost all surface types. Many navigation algorithms, simultaneous localization mapping (SLAM), and path planning in both research and industrial applications are based on the data of these devices. However, fixing the sensors on conventional forklifts is complex because of the movements required for handling loads. At the front, the forklift lifts forks that cause the laser beam's occlusion, making it impossible to perceive frontal movements and small or medium size obstacles. Additionally, the high cost of this sensor constraints the development of some applications.

This paper presents an innovative approach to the spatial perception of autonomous vehicles in industrial scenarios. The proposed method aims to avoid individual perceptions of fixed and expensive sensors. The *DepthLiDAR* consists in virtual sensor created through the active segmentation of a global perception source of the environment based on depth and RGB data from a top view camera. This approach facilitates the conception of multiple heterogeneous perception sources for application to intelligent and cooperative behaviors in industry. The proposed approach has been extensively experimented; consequently, few global perception sources can produce many spatial sensors, which are more reliable and precise than standard laser scanners.

This paper is structured as follows. Section II discusses the related work. In Section III, a detailed description of the proposed approach is presented. In Section IV, the results of the experiments performed are presented and discussed. Section V presents the overall analysis of the experiments and a comparison with standard LiDAR sensors. Finally, Section VI presents the conclusions and suggestions for future work.

II. RELATED WORK

Some researchers have been developing solutions to use alternative sensors that maintain the same data type and perception of LiDAR sensors, whether due to cost, size, or technical issues. In [12], a 2D laser scan is extracted from a 3D point cloud provided by a depth sensor attached to the robot. This method has a lower cost and, unlike conventional LiDARs in which only obstacles in the same plane as the sensor are detected, the entire field of view (FOV) of the sensor is analyzed. A disadvantage is that each robot needs a depth sensor. Similarly, in [13] the author uses data from a Microsoft Kinect attached to the robot to produce 2-D and 3-D SLAM. It is observed that the lower field of view and the high processing requirements limit the application of this method in practice.

In [14], a low-cost device was introduced that uses ultrasound sensors to detect curbs and replaces LiDARs. A combination of sensors and a series of algorithms was used to overcome the low precision of these sensors, employing four ultrasound sensors, where accuracy of 13.5 cm was achieved.

Rhee and Seo [14] introduced milliMap, a device that uses a millimeter-wave chip (mmWave) to generate a dense grid map with precision comparable to LiDAR. Its main advantage, in addition to the low price, is the ability to provide data from distances even in low visibility environments (e.g., dust, fog, smoke).

In [15], a simulation of a laser scanner based on a single ray approach was presented. Based on this approach, the dynamic effects resulting from the carrier system's movement or other dynamic elements in the simulated scene can be considered in detail. Additionally, the simulation of a laser scanner based on a single ray can run in real-time. In [16], a method for generating virtual scans for intelligent vehicles driving in complex urban environments was proposed. The technique uses a new data structure referred to as the fundamental virtual scan matrix (BVSM) to represent the 3D point cloud around the vehicle. The authors also presented a simultaneous road filtering and obstacle detection method that works on top of the BVSM to generate VScan and a sorted array-based acceleration method for VScan generation in real-time.

In [17], a new method for automatically extracting 3-D LiDAR point clouds with point-level ground truth labels from an autonomous driving simulator 3-D data analysis was presented. The results concluded that training with synthetic data could improve the performance of a deep model; this improvement becomes more prominent as the real-world data set becomes smaller. Similarly, in [18], synthetic data were used for experiments in urban environments. Yue *et al.* [19] presented a framework to generate artificial point clouds from video games. They achieved a +9% accuracy improvement by supplementing the training data with synthetic data.

Beltran *et al.* [20] presented a method for generating new point cloud datasets captured by a real LiDAR device. The proposed pipeline uses multiple frames for the accurate 3D reconstruction of a scene in the spherical coordinate system that allows the simulation of the scans of a virtual LiDAR sensor, configurable both in location and internal specifications.

Gusmão *et al.* [21] presented a guideline for the development of LiDAR simulators based on parallel raycasting algorithms. A sensor simulator is presented as a case study demonstrating a generation of synthetic point clouds in different scenarios. Finally, [22] proposed a support scheme for autonomous movement for estimating road direction using virtual LiDAR signals generated through semantic segmentation results and geometric information from the camera.

The review conducted indicates that LiDAR sensors are an essential tool for the autonomy of mobile robots. However, for the development and proof of concept for smart factories, these sensors' costs are relevant, being a financial challenge to embark on a sensor in each robot. Approaches with depth camera are an alternative, where a camera is used for each robot to create the LiDAR sensor. However, it is still necessary to have multiple cameras for multiple robots.

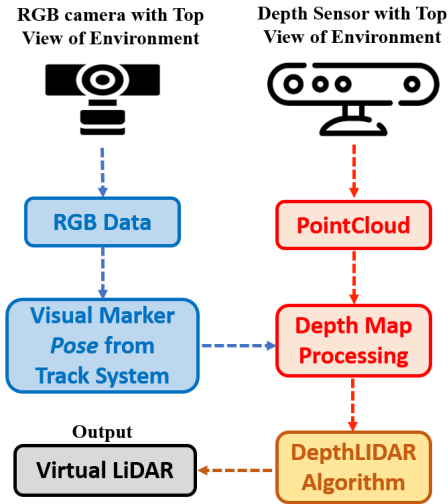


Fig. 1. Generic architecture for DepthLiDAR virtual sensor.

This paper presents an innovative approach that avoids individual perceptions of fixed and expensive sensors. The solution consists of a virtual sensor for autonomous vehicles, developed from the active segmentation of a source of global perception of the environment, based on RGB and depth cameras with top view.

III. DepthLiDAR - PROPOSED APPROACH

This section presents the structure of the virtual LiDAR sensor, that is, the *DepthLiDAR* and the tools required. The relevant configuration aspects are presented and discussed to obtain the right conditions for the virtual sensors.

A. Generic Tools for DepthLiDAR

The proposed approach is highly useful for small-scale environments prototypes and proof of concept. To develop the virtual LiDAR sensor, three components are fundamental: a camera that obtains the depth data (stereo or RGB-D), an RGB camera, and finally, a visual marker, also referred to as a tag.

Fig. 1 presents the generic architecture of the proposed system. Both cameras need to be positioned parallel to the ground and on the same elevation, obtaining a top view of the entire useful environment navigated by the robot. The visual marker will be positioned on top of the small-scale mobile robot (like Kilobot [23] or WsBot [24]), and this way, it will be visible to the RGB camera to allocate the virtual sensor in the same pose as the robot.

In short, the process of compose the virtual LiDAR involves obtaining the *pose* (x, y, z, θ) of the tags from the data processing of the RGB camera. The first 3 parameters (x, y, z) correspond to the tag position at Cartesian plane of the fixed inertial system, and the last one θ corresponds to the rotation of tag in relation to the Z axis of inertial system plan. With this information, data is crossed between the RGB and depth cameras, locating the tag's position on the depth map. An exploration process is then performed to determine if there are obstacles in the depth of information around the tag. This process is a scan of an area around the tag, which is, methodically increasing the sensor's angle and range. When a lower depth dimension than the depth of the test bench

floor is verified, it is deduced that there is an obstacle at a certain distance for that determined angle. Combining the data from all the laser beams, the virtual LiDAR or *DepthLiDAR* is obtained.

B. Tracking System of the Visual Marker Pose

Visual tracking facilitates the correct localization of agents in the environment. To achieve this, a specific marker is attached to each agent consisting of 2D coded targets with a predetermined geometric pattern and parameterized physical dimension, named as tags. Image processing techniques are used to extract the *pose* (i.e., position (x, y, z) and orientation θ) of each tag in the Cartesian \mathbb{R}^3 inertial system plan.

The open-source library Alvar developed by the Technical Research Center VTT of Finland [25] was used with a ROS package *ar_track_alvar* [26] to track the visual markers. This library contains computer vision algorithms to locate multiple and different markers in an image. It is necessary to perform camera calibration to obtain intrinsic geometric and lens parameters and correct possible distortions that vary according to the RGB camera model. The calibration process is described in [27].

The RGB camera is installed at the center of the structure, resulting in a complete top view of the small-scale environment. The tags containing encodings with ID information are positioned on the upper surface of the small robots, and are completely visible to the image acquisition system. To implement *DepthLiDAR*, the *ar_track_alvar* library detects all visual markers and supplies the *Pose* of the respective tag as an output to the system. The position of the *DepthLiDAR* is determined using these coordinates. In other words, the virtual sensor is positioned precisely on top of each virtual marker corresponding to the position of the mobile robot.

C. Depth Map Processing

The data from the depth camera play a fundamental role in the proposed approach. The depth sensor provides structured data as a point cloud, aligned with the point cloud library (pcl) [28]. Briefly, the point cloud is a data format where for each pixel of the image, in addition to RGB, its depth in relation to the sensor is given. To facilitate the LiDAR sensor's virtualization, the pcl data is restructured in matrix M_{res_x, res_y} with three channels. The matrix size corresponds to the resolution of the camera, whose number of columns corresponds to the pixel resolution of the camera res_x , and the number of lines corresponds to the resolution of the camera res_y . The matrix channels "1", "2", and "3" decompose \mathbb{R}^3 and have float data that store a point of the geometric coordinates X, Y, and Z of an underlying sampled surface. In other words, in a given $pixel_{i,j}$ contained in the matrix M_{res_x, res_y} , channel "1", stores the value of a point referring to component X of the space, whereas channels "2", and "3", store the respective values in Y and Z of a surface sampled by the depth sensor, respectively.

The matrix M_{res_x, res_y} , resulting from the restructuring of the depth sensor data contains information on points in the \mathbb{R}^3 space, which are processed to obtain the virtual LiDAR sensor data. Considering that the camera samples are synchronized,

there is a depth frame for each RGB frame, and it is necessary to determine the relationship between the result of the tracking system (tag position (x, y, z)) and restructured point cloud (matrix M_{res_x, res_y}).

The first procedure is to estimate the pixel in matrix M_{res_x, res_y} that corresponds to the spatial position (x, y, z) obtained from the tracking system. To achieve this, it is necessary to estimate the pixel size, that is, how much a pixel measures in meters for this system setup. Equation 1 presents the calculation to obtain the *pixel_size*. The association of three variables is used: the resolution of the depth sensor, the distance between the camera and surface, that is, the floor (h_floor), and the field of view fov , which is an intrinsic property of each depth sensor model. The fov is given in degrees and corresponds to the angle that determines the dimension of the area reached by the depth sensor. For Equation 1, assuming that the pixels are squared, the same size is used for the horizontal and vertical pixel edges. Knowing the pixel dimension, identifying the element i, j of M_{res_x, res_y} that corresponds to the position obtained by *ar_track_alvar* becomes a trivial process, as described by equations 2 and 3. Therefore, to determine the accuracy of the process, just compare the tracking system's position (x, y, z) corresponding to the point cloud position of the element $pixel_{i,j}$ of M_{res_x, res_y} of channels "1", "2", and "3", respectively.

$$pixel_size = \frac{2 * h_floor * \tan(\frac{fov}{2})}{res_x} \quad (1)$$

$$i = \text{int} \left(\frac{tag_x}{pixel_size} \right) \quad (2)$$

$$j = \text{int} \left(\frac{tag_y}{pixel_size} \right) \quad (3)$$

The next process involves exploring the area around the element $pixel_{i,j}$. The aim is to identify obstacles in the field of view of the virtual sensor (see Fig. 2). First, the parameters Ang_min and Ang_max are defined, and remembering $Laser_beam_k$ is applied in the clockwise scan (left to right), as in a real sensor. $Ang_increment$ defines the *DepthLiDAR*'s angular resolution. To explore the environment around the tag, $pixel_{i,j}$ is dynamically increased/decreased starting at $Range_min$ and ending at $Range_max$ according to the orientation of the $Laser_beam_k$ in question. If a quota in z in the pcl is significantly lower than the quota z on the floor (i.e., h_floor), an obstacle is then identified and returned at an increased distance.

D. DepthLiDAR Algorithm

The proposed approach is based on the depth map's active segmentation, as detailed in Algorithm 1, where the labels in *italics* are the parameter definitions and others are the variables. The *DepthLiDAR* function receives two parameters: the depth matrix M_{res_x, res_y} , and the *Tags* object vector containing the coordinates of all the identified visual markers from the tracking system. All parameter usefull for the *DepthLiDAR* function are summarized in Table I.

The *config_Parameters()* function initializes all parameters necessary for the operation of the *DepthLiDAR*,

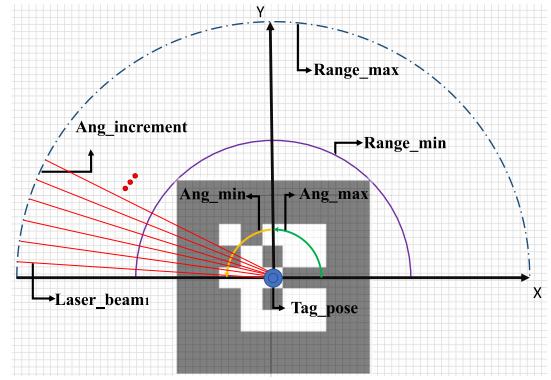


Fig. 2. The crossing of information between the tracking system and the point cloud virtualize the *DepthLiDAR*.

TABLE I

PARAMETERS TO THE DEVELOPMENT OF DEPTHLIDAR

Parameters	Description
M_{res_x, res_y}	Depth array or point cloud
<i>res_x</i>	X-axis resolution of depth sensor
<i>res_y</i>	Y-axis resolution of depth sensor
$pixel_{i,j}$	Depth matrix element in position i, j
h_floor	Distance from depth sensor to the floor
<i>fov</i>	Field of vision of depth sensor
<i>pixel_size</i>	Pixel size in meters
<i>Angle_min</i>	Reference for initial angle scan
<i>Angle_max</i>	Reference for final angle scan
<i>Angle_increment</i>	<i>DepthLiDAR</i> angular resolution
<i>Range_min</i>	Initial distance of the laser beam
<i>Range_max</i>	Maximum laser beam distance
$Laser_beam_k$	Distance value in laser beam k

namely $Range_min$, $Range_max$, $Angle_min$, $Angle_max$, $Ang_increment$ and $pixel_size$. Subsequently, the *DepthLiDAR* object that contains the attributes *id*, *pose*, *LaserBeam*-*Ranges* (vector of floats with the values of each beam of the virtual sensor) is instantiated. The rest of the code was structured in three nested loops.

The first loop, referred to as the localization loop, is dedicated to setting the center of the virtual sensor in the same position as the tag (position and orientation), identifies the index i, j of the reference pixel element to the tag's pose in the point cloud and calculates the initial beam orientation based on the *DepthLiDAR* orientation. The second loop is referred to as the laser beam. This loop is responsible for covering the entire angular extension of the sensor with a clockwise scan, increasing according to the *DepthLiDAR*'s angular resolution.

Finally, the third loop, which is within the laser beam loop, is the linear incrementing loop. It is responsible for constructing the beams and follows the rectilinear trajectory according to the $laser_beam_k$ orientation (*current_Ang*). The analysis starts with the parameter $Range_min$ and with each iteration of the loop, the pixel resolution distance is increased until an obstacle is detected or the maximum distance $Range_max$ is reached ($Range_min < range < Range_max$). The *pixel_Increment* function takes the pixel i, j as parameter referring to the center of the virtual sensor, the angle, and the current range of the result of the iteration that increases the laser beam and returns the coordinate m, n relative to the iteration.

The depth matrix M_{res_x, res_y} is analyzed to check for obstacles. The sensors have a certain degree of imprecision

Algorithm 1 *DepthLiDAR* Function

```

1 Function depthLidar( $M_{res\_x, res\_y}, Tags$ )
2   init.config_Parameters();
3   for tag in Tags do
4     sensor = init.depthLidar();
5     sensor.id = tag.id;
6     sensor.pose = tag.pose;
7     (i,j) = get_Pixel(tag.pose);
8     currentAng = sensor.pose.θ - Angle_min;
9     while currentAng < sensor.pose.θ + Angle_max do
10      range = Range_min;
11      while range < Range_max do
12        (m,n) = pixel_Increment(i,j,currentAng,range);
13        if  $M_{res\_x, res\_y}[m][n] < h\_floor + \sigma$  then
14          break ;
15        end
16        range = range + pixel_size;
17      end
18      sensor.ranges.add(range);
19      currentAng = currentAng + Ang_increment;
20    end
21    return sensor;
22  end
23 end
24 Function get_Pixel (tag.pose)
25   i = tag.pose.x / pixel_size;
26   j = tag.pose.y / pixel_size;
27   return (i,j);
28 end
29 Function pixel_Increment (i,j,currentAng,range)
30   m = i + int(range * cos(currentAng) / pixel_size);
31   n = j + int(range * sin(currentAng) / pixel_size);
32   return (m,n);
33 end

```

that was considered a standard deviation (σ) of the measurements obtained from h_floor , that is, the distance between the camera and the floor. Therefore, if the sensor detects any obstacle, that is, a smaller depth value, the linear increment is interrupted, the value of the current range is stored in the *DepthLiDAR* object's distance vector, and the angle is increased. At the end of the iterative process, the function responds with an object of the *DepthLiDAR* type containing the sensor ID, position, and range measurements stored in a vector.

IV. ANALYSIS OF *DepthLiDAR*

A rigorous analysis of the *DepthLiDAR* approach is presented to highlight its advantages, limitations, and performance. The proposed method has significant differences from conventional LiDAR scanners, mainly related to the camera lens and object shadow, in which its response is significantly attenuated. Despite this, the proposed method has a response with fewer errors than the conventional LiDAR. It aims to improve the mobile agents' perception in indoor environments, allowing clear obstacle sensing with direct global localization.

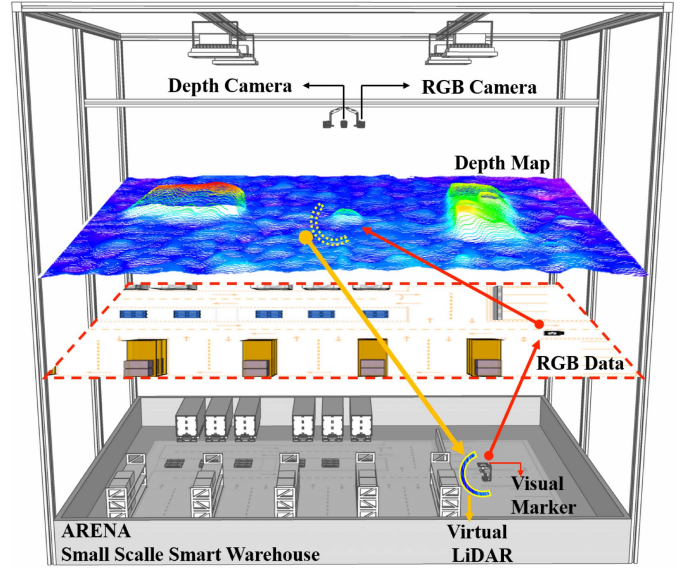


Fig. 3. Sketch of ARENA, highlighting the elements for *DepthLiDAR*.

The *DepthLiDAR* is evaluated in six categories specially designed to quantify its characteristics. The categories assess the angular asymmetry, peripheral distortion, effect of lens curvature, limitation on hollow objects, object shadow impact and scan time. All this experiments were conducted on a test bench, the ARENA, composed of the tools and configurations necessary for *DepthLiDAR* previously presented. Finally, a comparative assessment was conducted between LiDAR and *DepthLiDAR*, using a larger environment (outside the test bench) to verify the influence of larger dimensions applied to *DepthLiDAR*.

A. Test Bench, Tools and Setup for the Validation of *DepthLiDAR*

The structure developed to represent the activities and interactions in a smart warehouse on a small scale is referred to as ARENA, illustrated in Fig. 3. It consists of a metallic support and a wooden top, 2 m long, 1.2 meters wide, and 1.4 meters high. This real structure supports small-scale autonomous mobile robots, the WsBot [24], localized by visual markers. The RGB camera and the depth camera are positioned to cover the entire useful area traveled by the robots in a top perspective. ARENA is a tool with great potential for developing and validating studies and optimizations dedicated to smart warehouse management systems, minimizing the risks and costs involved in testing in real environments. More details of the architecture and its applications related to ARENA can be found in [29].

For the validation of the virtual sensor, mobile robots were replaced by a conventional LiDAR sensor. A tag was attached to the top of this real sensor, not obstructing its field of view, therefore not influencing the measurements. *DepthLiDAR* is virtually fixed at the center of a conventional LiDAR with the same resolution and range. Subsequently, both approaches are compared and the differences highlighted. The reference LiDAR used in the experiment was the Hokuyo URG04-LX, the RGB camera was Logitech C920 and the depth sensor was an Intel Real Sense D435. The hardware used to

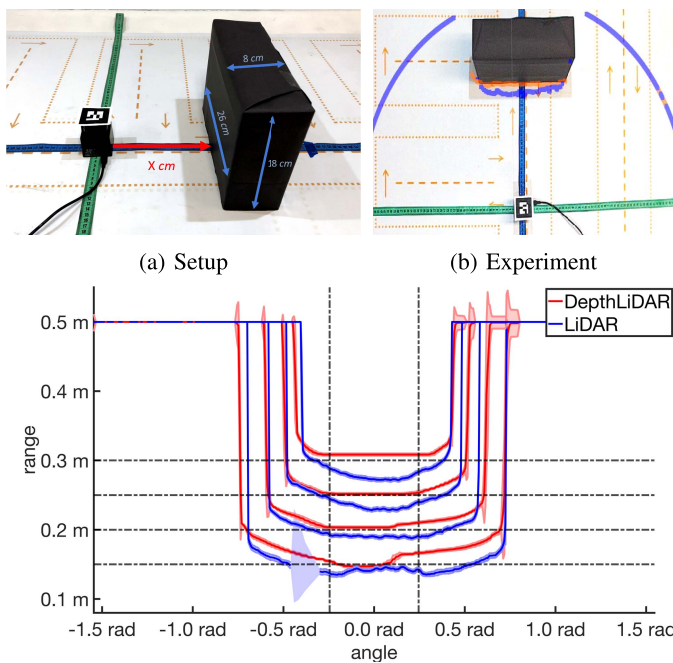


Fig. 4. Experiment of the Angular Asymmetry.

process *DepthLiDAR* consist in Intel Core i7 2.40 GHz with 8 Gb of RAM.

Twenty three experiments, divided into six categories, were conducted to analyze the performance of the *DepthLiDAR*, where in each experiment, 200 measurements were made on each sensor, with a variation of 180° and 512 points. All experiment data is available in the repository [30]. A statistical analysis was conducted to quantify the performance of the proposed approach.

B. Angular Asymmetry

The *DepthLiDAR* approach is based on global perception through a top-view RGB-D camera to construct the environment's depth map. The RGB-D camera has lens distortions that attenuate the object perceptions in peripheral regions (i.e., far away from the center of the camera). This experiment aims to evaluate the influence of the distance of the camera center on obstacle detection, as shown in Fig.4.

Additionally, the experiment aims to evaluate the angular asymmetry caused by the lens of the depth camera. This characteristic introduces a non-linear difference between the measurements near the lens center and far from them. Particularly, this class of tests compares the alignment of the $beam_n$ of the virtual sensor with the $beam_n$ of the real sensor. Both sensors are located at a variable distance (represented by the red arrow in Fig.4a) from a box. The box was 15 cm high, 26 cm wide, and 8 cm deep. Therefore, it is possible to identify, the beam that detects the first edge of the box and the one that identifies the box's last edge in both sensors.

The angular asymmetry is evaluated in two sectors, near the camera center (diameters of 0.15 and 0.20 cm) and far from the camera center (diameters of 0.25 and 0.30 cm). These distances were chosen based on the camera range on the small-scale factory floor. A comparison window is

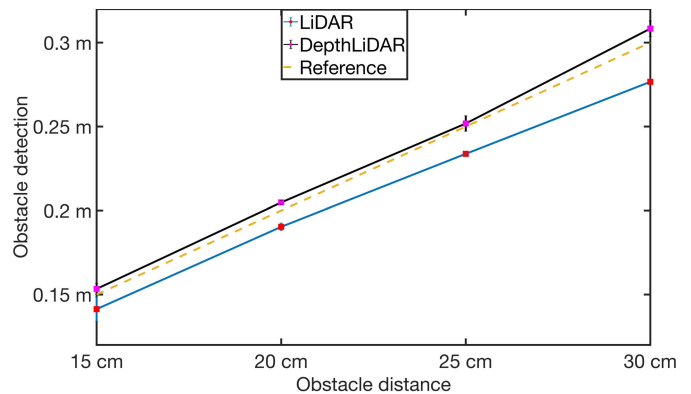


Fig. 5. Analysis of the Angular Asymmetry.

stipulated at the center of the box detection (80 points), between -0.2454 and $0.2454 \pi \text{ rad}$, as shown in Fig.4c.

Both sensors have more accurate measurements with obstacles close to the sensor and increased errors in the peripheral sectors, as shown in Fig.5. The *DepthLiDAR* has a 5.25 mm error near the center and 7.90 mm far from it. Conversely, the conventional LiDAR has a 7.60 mm error near the center and 23.65 mm far from it.

C. Peripheral Distortion

The depth camera captures non-square images in widescreen resolution (1920×1080 pixels). Thus, the angular asymmetry is distinct horizontally and vertically. The smaller side (vertical with 1080 pixels) attenuates the effects of the angular distortion more intensely than the bigger side (horizontal with 1920 pixels), in a factor of 16:9. This experiment aims to quantify these effects.

A scenario with a cylindrical obstacle is designed with the sensor fixed at the center. The obstacle is also established using several small cylinders to evaluate different laser beam reflection cases, as shown in Fig.6. The circumference radius was changed to 15, 20, 25, and 30 cm. A group of 200 successive measurements were obtained in each case for both sensors, as shown in Fig.6c.

The analyses focus on the frontal and peripheral measurements. The *DepthLiDAR* has less error in both cases, a mean of 2.97 mm in the frontal and 10.25 mm laterally; however, it has a significant distortion in its frontal and peripheral measurements (distortion factor of 3.44), as shown in Fig.6c. However, the LiDAR obtains similar measurements in the frontal (mean error 13.40 mm) and peripheral (mean error 10.25 mm), with a distortion factor of 1.33, but introduces more errors, as shown in Fig.7.

The conventional LiDAR sensor loses its capability of obstacle detection over long distances, mainly because of laser beam reflection disturbances that lead to several null measurements. This condition is not observed in the *DepthLiDAR*, which maintains detection. The LiDAR has a mean of 11.31 invalid points per measure, which corresponds to 2.20% of the message.

D. Effects of Lens Curvature

The RGB-D camera has disturbances because of the curvature lens that also introduces error in obstacle detection.

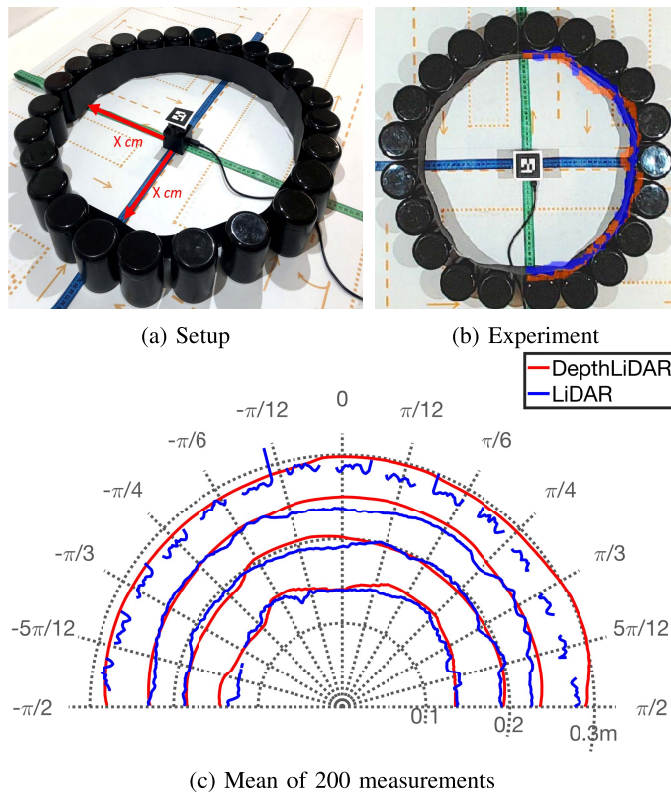


Fig. 6. Experiment of peripheral distortion.

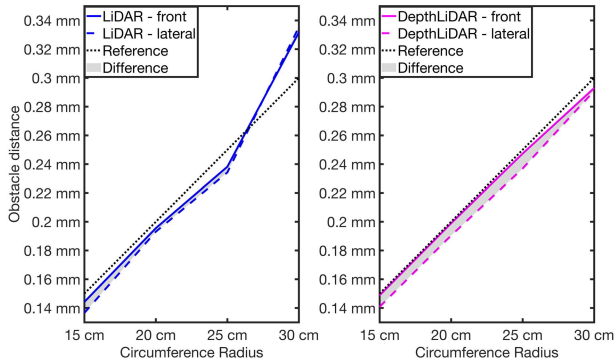


Fig. 7. Analysis of peripheral distortion.

This experiment aims to quantify this influence in different directions.

A box ($15 \times 26 \times 8 \text{ cm}$) was used as an obstacle, and the virtual and real sensors were positioned at a fixed distance of 20 cm from the box as shown in Fig. 8. Subsequently, an experiment was carried out with the set (box and sensors) at a distance of 20 and 40 cm to the right of the central axis of the camera and 20 and 40 cm to the left of the center of the camera axis. Another experiment 20 cm below the center of the image and 20 cm above the camera axis was conducted, totaling six experiments for this test class.

The DepthLiDAR presents more error in the South direction measurements than other directions. However, the detection is more precise in all the directions than the regular LiDAR sensor, as shown in Fig. 9. The DepthLiDAR has less than half of the errors of the conventional sensor.

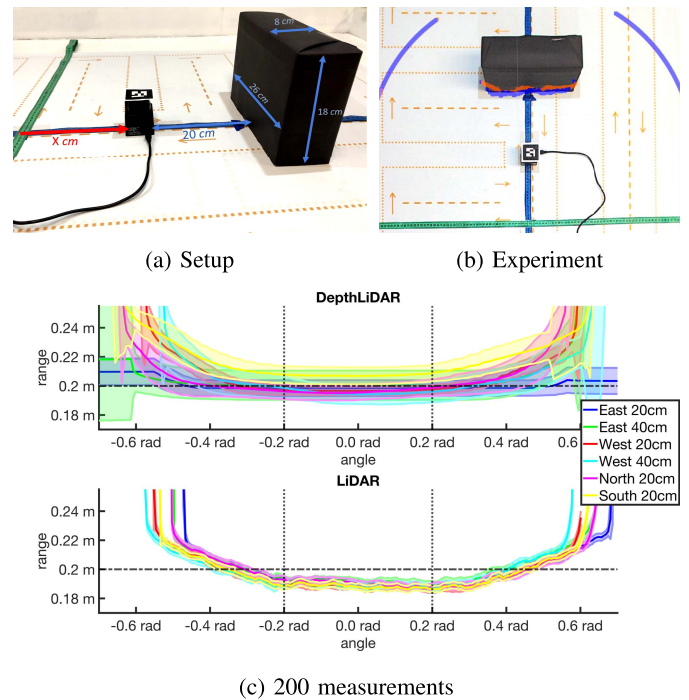


Fig. 8. Experiment of lens' curvature.

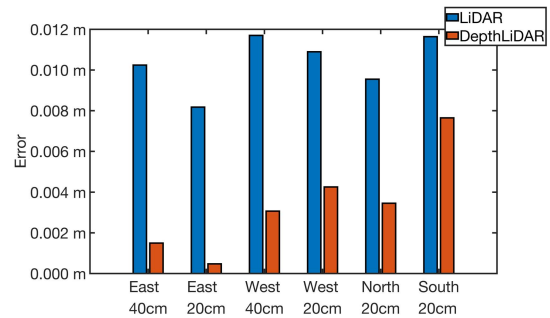


Fig. 9. Analysis of lens' curvature.

E. Restriction of Suspended Objects

This experiment aims to evaluate the difference between the real and virtual sensors in the presence of a suspended object, which is detectable by the camera's RGB-D sensor but not detectable by the real sensor because it is hollow or has no surface. Fig. 10 illustrates the setup for the experiment. The real and virtual sensors were placed at a distance of 25 cm from the shelf.

Fig. 10c presents a comparison of the laser message of the virtual and real LiDAR. The gray area is the suspended obstacle when the DepthLiDAR can detect this obstacle instead of a regular LiDAR.

F. Shadow Limitation

The experiment aims to evaluate the influence of the shadow or blind areas on the virtual sensor because of the projection between a larger obstacle covering a smaller object's view. This experiment used two objects, a box measuring 18.5 cm high, 26.5 cm wide, and 8.5 cm deep and a smaller cylindrical box measuring 9 cm high and 6 cm in diameter. The larger box was placed at a distance of 20 cm from the real and virtual

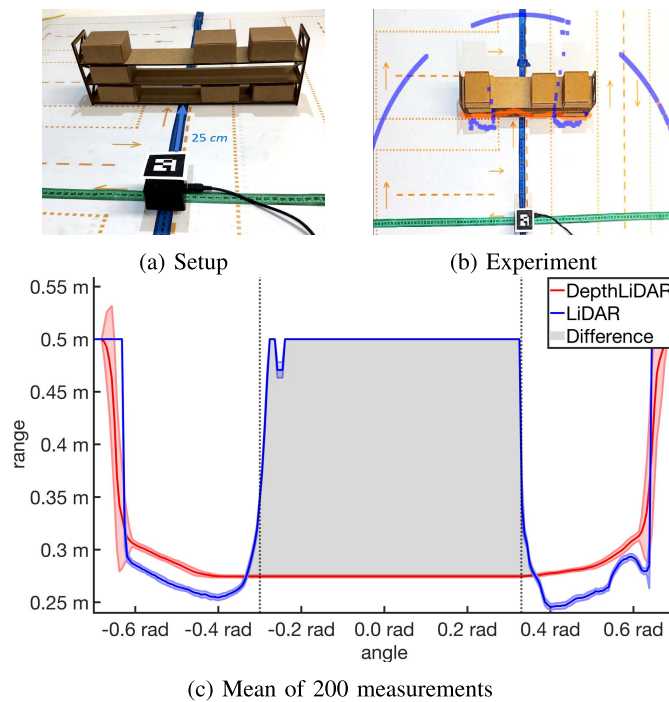


Fig. 10. Experiment of restriction of suspended objects.

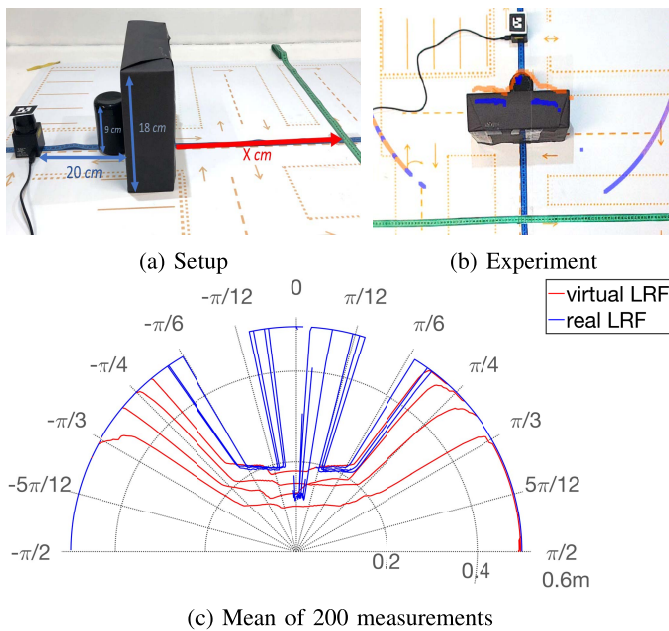


Fig. 11. Experiment of shadow limitation.

sensors, whereas the smaller box was positioned between the sensor and larger box, as shown in Fig. 11.

Fig. 12 illustrates a comparison between the mean error of the *DepthLiDAR* and regular *LiDAR*. The influence of the objects' shadow is only visible in the *DepthLiDAR* because of this top-down perception. The error is emphasized by other factors, such as the lens curvature, peripheral distortion, and angular asymmetry. Despite this, superior perception allows the sensor to detect objects behind larger objects (such as inversion in this experiment).

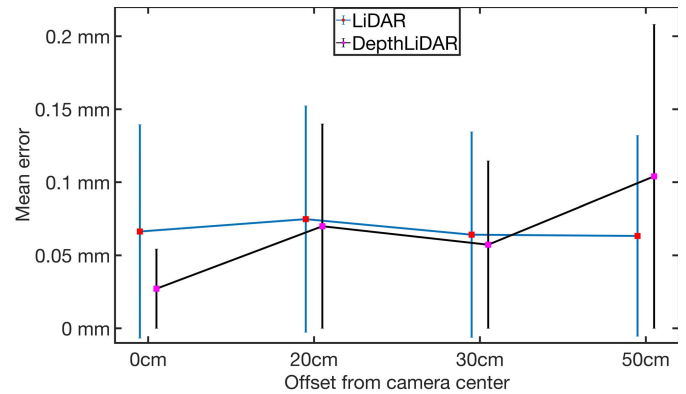


Fig. 12. Analysis of shadow limitation.

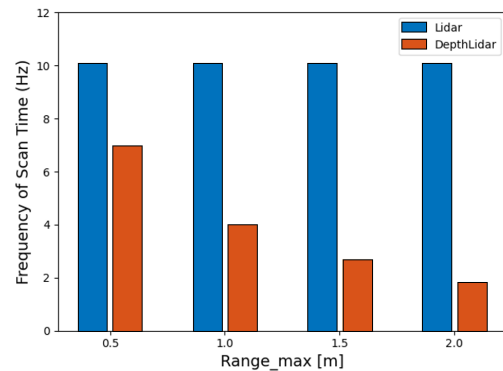


Fig. 13. Scan time experiment.

G. Scan Time Evaluation

This experiment aims to evaluate the difference in scan time between the real and virtual *LiDAR*. In this experiment, it was analyzed the frequency that each sensor makes available the data of the scan related to 180° , i.e., of the 512 points of measured depth, varying the *Range_max* of the sensor, that is the maximum measurement defined between 0.5, 1.0, 1.5 and 2.0 meters.

Fig. 13 presents the results of the scan time comparing the real sensor and the virtual sensor. According to the manufacturer, the Hokuyo URG04-LX has a reading time of 100 ms, thus generating an update frequency of 10 Hz, and this corresponds to the samples obtained in the three tests performed. The scan performance of *DepthLiDAR* suffers an attenuation in the scan time frequency as the value of *Range_max* increases. This fact is directly linked to the volume of data processed in the point cloud when realizing the exploration to detect objects.

H. Accuracy in Real Dimensions

All previous experiments have been conducted at ARENA, the small-scale test bench for developing prototypes for intelligent warehouses. Therefore, the range of the virtual sensor is small, meeting the needs of the bench. An experiment was conducted outside the ARENA to evaluate the virtual sensor for systems with real dimensions. The objective of this experiment is to evaluate the ability of *DepthLiDAR* to detect obstacles with distances in a similar way to that found in real applications. For this, the similar setup experiment realized in subsection B of this chapter was performed (Fig. 4). The setup

TABLE II
ANALYSIS OF THE *DepthLiDAR* HIGHLIGHTING THE IMPROVEMENTS

ANALYSIS	OBJECTIVE	VARIATION	LiDAR	deviance	DepthLiDAR	deviance	Improvement
Angular Asymmetry	Measurement error [mm]	Obstacles near to camera center	9.15	0.47	4.10	0.73	55.19%
		Obstacles far to camera center	44.80	0.52	21.75	0.18	51.45%
Peripheral Distortion	Frontal measurement error [mm]	obstacles near to camera center	53.60	3.60	11.90	4.30	77.80%
		Obstacles far to camera center	13.40	0.90	2.98	1.08	77.80%
	Peripheral measurement error [mm]	Obstacles near to camera center	71.50	11.20	41.00	9.20	42.66%
		Obstacles far to camera center	17.88	2.80	10.25	2.30	42.66%
Effects of Lens' Curvature	Measurement error with shifted sensor [mm]	North	9.50	0.40	3.50	1.20	63.16%
		South	11.60	0.40	7.60	1.20	34.48%
		East (near)	10.20	0.40	1.50	0.40	85.29%
		East (far)	8.20	0.40	3.10	0.00	62.20%
		West (near)	11.70	0.30	3.10	0.60	73.50%
		West (far)	10.90	0.30	4.30	2.00	60.55%
Shadow limitation	Measurement error over shadow limitation [mm]	Near	70.55	0.42	48.50	0.00	31.25%
		Far	63.70	0.47	80.65	0.01	-26.61%
Restriction of suspended objects	Can measure suspended objects?	-	No	-	Yes	-	-
Measurement failure	Number of null points per measurement	-	11.31	-	0.00	-	-
Overall error [mm]			29.05	1.61	17.44	1.66	52.24%

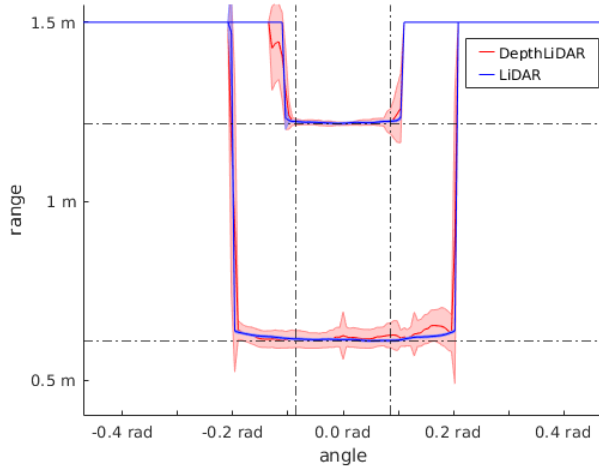


Fig. 14. Long Distance Accuracy Experiment with *RealSense*.

distances were changed, the cameras were positioned at 2.5 m in height and, the LiDAR and *DepthLiDAR* at the center of the coordinate system and the obstacle at two different distances, first at 0.6 m and then at 1.2 m. Fig.14 shows the results obtained.

Due to the greater distance from the obstacle, the comparison window of this experiment has a smaller dimension: from -0.0859 to $0.0859 \pi rad$ with 28 points. With the obstacle positioned at 0.6 m, the average error obtained in *DepthLiDAR* was 25.0 mm while in LiDAR it was 19.2 mm, with the obstacle positioned at 1.2 m in error *DepthLiDAR* was 21.6 mm while in LiDAR it was 20.4 mm. The light red band on the graph represents the standard deviation of the virtual sensor data, where it is possible to notice a high variation. The high inaccuracy in this experiment may make the use of *DepthLiDAR* impractical in some applications that require high precision. According to the manufacturer of the depth sensor used (Intel RealSense 435), the depth measurement error is proportional to the distance.

V. OVERALL EVALUATION OF THE *DepthLiDAR*

The different experiments evaluating the *DepthLiDAR* approach aim to determine and quantify the characteristics of virtual sensors in comparison with the real LiDAR sensor. A set of twenty three experiments, in six categories applied to ARENA workbench, was performed, where each experiment consisted of 200 similar and successive measurements with both sensors. Each sensor has 512 points in 180° . A total of 3.891.200 points were captured and analyzed. Additionally, two experiments were conducted to evaluate the applicability of *DepthLiDAR* on large scale.

A. *DepthLiDAR* Improvement

Table II summarizes the first five experiments comparing the measurements of distance, distortions and restrictions between LiDAR and *DepthLiDAR*.

The *DepthLiDAR* has a better measurement in almost all experiments, presenting less measurement errors. The overall evaluation, considering all experiments, delimits that the traditional LiDAR has a mean error of 29.05mm with standard deviance of 1.61mm. The *DepthLiDAR* has a mean error of 17.44mm with a standard deviance of 1.66mm, which characterizes a 52.24% measurement improvement compared to the conventional sensor.

Another relevant difference is that the classic LiDAR loses the same measurements owing to the incorrect reflection of laser beams, introducing null points, with a mean of 11.31 points per measurement, or 2.2% of the LiDAR measurement. This phenomenon does not occur in the *DepthLiDAR* approach, which promotes 100% of the measured points.

Classic LiDAR sensors can be perceived around the center in a 2D plane. This characteristic introduces a "false" perception of clear spaces in the presence of suspended objects such as the shelves of warehouses. The top-down perception of the *DepthLiDAR* approach avoids this effect, merging the detection of suspended objects in a lower 2D measurement plane.

TABLE III
ANALYSIS OF THE *DepthLiDAR* HIGHLIGHTING ITS LIMITATIONS

ANALYSIS	OBJECTIVE	VARIATION	LiDAR	deviance	<i>DepthLiDAR</i>	deviance	Difference
Scan time	Measurement of scan frequency [Hz]	<i>Range_Max</i> = 0.5 [m]	9.91	0.01	6.98	0.22	29.56%
		<i>Range_Max</i> = 1.0 [m]	9.92	0.01	3.92	0.19	60.48%
		<i>Range_Max</i> = 1.5 [m]	9.91	0.01	2.70	0.19	72.75%
		<i>Range_Max</i> = 2.0 [m]	9.91	0.01	1.80	0.17	81.83%
		Error [Hz]	9.91	0.01	3.85	0.19	61.15%
DepthLiDAR applied to large scale	Measurement error [mm]	Obstacle at 0.6 m from camera center	19.30	7.0	25.00	31.30	29.53%
		Obstacle at 1.2 m from camera center	20.40	3.0	21.60	7.90	5.88%
		Error [mm]	37.50	5.0	23.30	19.60	16.20%

The point cloud is segmented in standard laser scan messages because it has some advantages compared with the point cloud, mainly concerning processing time. The 2D LiDAR promotes a spatial perception around the robot in a 2D half-circle, which detects the main obstacles to movement and manages avoidance motions. The information about obstacle heights is irrelevant to the autonomous behaviors of ground vehicles.

B. DepthLiDAR Limitations

The first limitation observed in *DepthLiDAR* was identified in the experiment that involves shadows. The *DepthLiDAR* degenerates its measurements in the shading of objects, which is emphasized when the object is still far from the center of the camera. This failure is a consequence of top-down perception and does not occur in the frontal perception of classic LiDAR sensors. In these cases, the *DepthLiDAR* has 26.61% more errors than the traditional LiDAR.

Table III summarizes the limitations of *DepthLiDAR*. The experiment analyzes the sensor's scan time to a cutting drop in the availability of sensor data as the LiDAR's range increases, i.e., *Range_Max*. Due to the amount of data to be analyzed in the point cloud, this becomes the most worrying limitation of the *DepthLiDAR*. For small-scale environments, such as the workbench used for the tests, the 6.98 Hz time scan is acceptable. However, for a more extensive coverage radius, the latency of the virtual sensor becomes an enemy for the application in autonomous robots since the time scan decreases considerably. However, increasing the specs of the hardware used to process the *DepthLiDAR* and using the GPU can attenuate the difference between the *DepthLiDAR* and the LiDAR sensor. The *DepthLiDAR* scan time is dependent on the computer performance, and the experiments adopt a standard I7, generation four, with 2.4 GHz and 8 GB of RAM. This characteristic can be improved with the use of more powerful computers.

Finally, the last limitation identified in *DepthLiDAR* is related to the dimension of the environment where the virtual sensor is applied. However, it is not a critical limitation since the obstacle was detected and its contoured silhouette defined, with a precision error in the order of 20 mm. Another factor to be taken into account is the depth sensor adopted in the experiments, i.e., Intel Real Sense D435, which has intrinsic noises as the depth at which it operates increases. Using cameras from other, more stable manufacturers to operate at

greater depths, such as Microsoft Kinect V2.0, will mitigate the error in real environments.

VI. CONCLUSION

This paper presented an approach for the active segmentation of an up-view point cloud to create a virtual LiDAR. The approach aimed to introduce precise perception in multiple mobile agents to achieve advanced intelligent behaviors in smart warehouse prototypes, facilitating the interconnected and autonomous cooperative performance required by Industry 4.0. The proposed approach facilitated the introduction of different perception systems with higher precision than regular LiDAR sensors, and exhibited a better measurement effect in almost all experiments; furthermore, it obtained fewer measurement errors. Compared with the conventional LiDAR, the measurement improvement is 52.24%. Additionally, it is possible to use a single RGB-D sensor to virtualize multiple LiDAR sensors.

Future work will evaluate the introduction of different kinds of *DepthLiDAR* in autonomous forklifts to determine the most appropriate characteristics for autonomous behaviors and research the best approach to mitigate the limitation related to scan time in large coverage radius of proposed virtual LiDAR.

REFERENCES

- [1] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018.
- [2] E. Hozdić, "Smart factory for industry 4.0: A review," *Int. J. Modern Manuf. Technol.*, vol. 7, no. 1, pp. 28–35, 2015.
- [3] T. Stock and G. Seliger, "Opportunities of sustainable manufacturing in industry 4.0," in *Proc. CIRP*, vol. 40, 2016, pp. 536–541.
- [4] L. Cheng *et al.*, "Registration of laser scanning point clouds: A review," *Sensors*, vol. 18, no. 5, p. 1641, May 2018.
- [5] A. Kroll and S. Soldan, "Survey results on status, needs and perspectives for using mobile service robots in industrial applications," in *Proc. 11th Int. Conf. Control Automat. Robot. Vis.*, Singapore, Dec. 2010, pp. 621–626.
- [6] M. Salhaoui, A. Guerrero-González, M. Arioua, F. J. Ortiz, A. El Oualkadi, and C. L. Torregrosa, "Smart industrial IoT monitoring and control system based on UAV and cloud computing applied to a concrete plant," *Sensors*, vol. 19, no. 15, p. 3316, Jul. 2019.
- [7] J.-T. Li and H.-J. Liu, "Design optimization of Amazon robotics," *Automat., Control Intell. Syst.*, vol. 4, no. 2, pp. 48–52, 2016.
- [8] R. Mason, "Developing a profitable online grocery logistics business: Exploring innovations in ordering, fulfilment, and distribution at Ocado," in *Contemporary Operations and Logistics*. Cham, Switzerland: Springer, 2019, pp. 365–383.
- [9] C. K. M. Lee, "Development of an industrial Internet of Things (IIoT) based smart robotic warehouse management system," in *Proc. Int. Conf. Inf. Resour. Manage. (CONF-IRM)*, 2018, pp. 1–14.

- [10] A. Yilmaz and H. Temeltas, "Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LiDAR data," *Robot. Auto. Syst.*, vol. 122, Dec. 2019, Art. no. 103285.
- [11] C. Wang *et al.*, "Safe and robust mobile robot navigation in uneven indoor environments," *Sensors*, vol. 19, no. 13, p. 2993, Jul. 2019.
- [12] F. Nardi, M. T. Lázaro, L. Iocchi, and G. Grisetti, "Generation of laser-quality 2D navigation maps from RGB-D sensors," in *Robot World Cup*. Springer, 2018, pp. 238–250.
- [13] A. Oliver, S. Kang, B. C. Wünsche, and B. MacDonald, "Using the Kinect as a navigation sensor for mobile robotics," in *Proc. 27th Conf. Image Vis. Comput. New Zealand (IVCNZ)*, 2012, pp. 509–514.
- [14] J. Rhee and J. Seo, "Low-cost curb detection and localization system using multiple ultrasonic sensors," *Sensors*, vol. 19, no. 6, p. 1389, Mar. 2019.
- [15] M. Emde and J. Rossmann, "Validating a simulation of a single ray based laser scanner used in mobile robot applications," in *Proc. IEEE Int. Symp. Robot. Sensors Environ. (ROSE)*, Oct. 2013, pp. 55–60.
- [16] H. Mengwen, E. Takeuchi, Y. Ninomiya, and S. Kato, "Robust virtual scan for obstacle detection in urban environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 683–690.
- [17] F. Wang, Y. Zhuang, H. Gu, and H. Hu, "Automatic generation of synthetic LiDAR point clouds for 3-D data analysis," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 7, pp. 2671–2673, Jul. 2019.
- [18] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: An open urban driving simulator," 2017, *arXiv:1711.03938*. [Online]. Available: <https://arxiv.org/abs/1711.03938>
- [19] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR point cloud generator: From a virtual world to autonomous driving," 2018, *arXiv:1804.00103*. [Online]. Available: <https://arxiv.org/abs/1804.00103>
- [20] J. Beltran *et al.*, "A method for synthetic LiDAR generation to create annotated datasets for autonomous vehicles perception," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1091–1096.
- [21] G. F. Gusmão, C. R. H. Barbosa, and A. B. Raposo, "Development and validation of LiDAR sensor simulators based on parallel raycasting," *Sensors*, vol. 20, no. 24, p. 7186, Dec. 2020.
- [22] M. Adachi and R. Miyamoto, "Model-based estimation of road direction in urban scenes using virtual LiDAR signals," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 4498–4503.
- [23] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 3293–3298.
- [24] M. A. Limeira, L. Piardi, V. C. Kalempa, A. S. de Oliveira, and P. Leitão, "WsBot: A tiny, low-cost swarm robot for experimentation on industry 4.0," in *Proc. Latin Amer. Robot. Symp. (LARS), Brazilian Symp. Robot. (SBR) Workshop Robot. Educ. (WRE)*, Oct. 2019, pp. 293–298.
- [25] V. T. R. C. of Finland. *Alvar—3D Tracking*. Accessed: Jan. 2, 2021. [Online]. Available: <https://virtual.vtt.fi/virtual/proj2/multimedia/alvar>
- [26] A. Cantieri *et al.*, "Cooperative UAV-UGV autonomous power pylon inspection: An investigation of cooperative outdoor vehicle positioning architecture," *Sensors*, vol. 20, no. 21, p. 6384, Nov. 2020.
- [27] A. De la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," *Sensors*, vol. 10, no. 3, pp. 2027–2044, Mar. 2010.
- [28] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 1–4.
- [29] L. Piardi, V. C. Kalempa, M. Limeira, A. S. de Oliveira, and P. Leitão, "ARENA—Augmented reality to enhanced experimentation in smart warehouses," *Sensors*, vol. 19, no. 19, p. 4308, Oct. 2019.
- [30] L. Piardi. (2021). *Depthlidar*. [Online]. Available: <https://github.com/LuisPiardi/DepthLiDAR>



Marcelo Limeira received the bachelor's degree in electrical engineering from Positivo University (UP) and the master's degree from the Federal University of Technology–Parana (UTFPR), where he is currently the Ph.D. degree. His research focuses on industrial automation, mobile robotics, and artificial intelligence.



production systems, smart warehouse, autonomous mobile robot and collaborative robots.



Luis Piardi received the bachelor's degree in electronic engineering from the Federal University of Technology–Parana (UTFPR), Brazil, in 2018, and the master's degree in industrial engineering from the Instituto Politécnico de Bragança, Portugal, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with UTFPR. Since 2018, he has been a Researcher with the Research Centre in Digitalization and Intelligent Robotics (CeDRI). His research has been focused on intelligent production systems, smart warehouse, autonomous mobile robot and collaborative robots.

Vivian Cremer Kalempa received the bachelor's degree in computer science from Santa Catarina State University (UDESC) and the master's degree in computer science from Santa Catarina Federal University (UFSC). She is currently pursuing the Ph.D. degree with the Federal University of Technology–Parana (UTFPR). She is also a Professor with UDESC. Her research has been focused on intelligent production systems, collaborative robots, and autonomous mobile robot.



Paulo Leitão (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Porto, Portugal, in 2004. In 1995, he joined the Polytechnic Institute of Bragança, where he is a Full Professor with the Department of Electrical Engineering and Scientific Coordinator, Research Centre in Digitalization and Intelligent Robotics (CeDRI). He has published four books and more than 300 articles in high-ranked international scientific journals and conference proceedings (peer-review). His research interests include industrial informatics, intelligent and reconfigurable systems, cyber-physical systems, the Internet of Things, distributed data analysis, factory automation, multi-agent systems, holonic systems, and self-organized systems. He served as the General Co-Chair of several international conferences, namely IEEE INDIN 2018, SOHOMA 2016, IEEE ICARSC 2016, HoloMAS 2011, and IFAC IMS 2010. He is a Senior Member of the Systems, Man, and Cybernetics Society (SMCS), the Chair of the IEEE IES Technical Committee on Industrial Agents, and a member at-large of the IEEE IES Administrative Committee (AdCom). He was the Chair of the established IEEE 2660.1-2020 Standard. He received six scientific paper awards and is the coauthor of three patents. He has coordinated/participated in several national and international research projects (FCT, Portugal2020, EU FP7, and EU H2020) and networks of excellence.



André Schneider de Oliveira (Member, IEEE) received the M.Sc. degree in mechanical engineering, concentrate in force control of rigid manipulators from the Federal University of Santa Catarina (UFSC) in 2007 and the Ph.D. degree in engineering of automation and systems in 2011. His Ph.D. thesis was on differential kinematics through dual-quaternions for vehicle-manipulator systems. He is an Associate Professor with the Federal University of Technology–Parana (UTFPR) and a member of Advanced Laboratory of Robotics and Embedded Systems (LASER) and Laboratory of Automation and Advanced Control Systems (LASCA). His research interests lie in the areas of robotics, mechatronics and automation with particular focus in navigation and localization of mobile robots, autonomous and intelligent systems, perception and environment identification, cognition, and deliberative decisions, human-interaction, and navigation control.