2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

Object Detection and Navigation of a Mobile Robot by Fusing Laser and Camera Information

Spyridon Syntakas, Kostas Vlachos, *Member, IEEE,* and Aristidis Likas, *Senior Member, IEEE*

***Abstract*— While state-of-the-art YOLO approaches have revolutionized real time object detection in mobile robotics, most of the publicly available models are trained on datasets with a small number of available classes. In addition, the difficulty in creating large datasets with many available classes for 2D object detection sets limitations to real world robotic applications and specialized use cases. This paper presents a solution that tackles these limitations by approaching object detection via fusion of 2D laser and RGB camera information resulting to a detector with 1000 learned classes. Object localization is performed in the 3D world by clustering the point cloud provided by the 2D laser scanner using the DBSCAN algorithm. The clusters are projected onto the image plane providing Regions of Interest (ROI), where proposed object bounding boxes are obtained, that are labeled with distance information. Object recognition is achieved using a pretrained, on the ImageNet dataset, ResNet and a voting schema among proposed bounding boxes, that also estimates the objects height. The detection system is used in combination with a navigation system that employs artificial potential field. The combination of the two, makes the robot's perception easily adaptable to specialized applications and the robot's behaviour adjustable to the complexity and variability of unstructured and unknown workspaces. The method has been implemented in ROS and tested both in simulation as well as in real case scenarios using the mobile robot Pioneer 3-DX. The work is aimed at robots with limited hardware and sensor capabilities and tries to enable detection via fusion, despite the limitations.**

I. INTRODUCTION

Robotic navigation and motion planning is a field of great importance in robotics that is nowadays greatly enhanced by the advances in sensory technology and machine learning. Deep learning revolutionized robotic vision which is now a core concept in robotic navigation. Of great importance is the task of object detection which gives mobile robots the ability to interact with objects of interest in the surrounding environment. State-of-the-art models like YOLO [1] provide real time object detection with high accuracy, capable of reinforcing reliability in navigation, using only camera imagery information. Although the state-of-the-art models have shaped and revolutionized the task of 2D object detection, the small number of available classes in commonly used datasets (e.g., COCO [2]), the difficulty in training these models to new and large datasets, the difficulty to create new datasets for specific applications and the existence of various other sensors attached to robots give space to new approaches.

The exploitation of point cloud data provides another way to tackle the problem of object detection. Convolutional Neural Networks like PointNet [3] and VoxelNet [4] are trained to detect objects using only 3D point cloud data. Available multi-modal datasets [5] enable multi-modal detection via fusion of data [6]. Fusion techniques [7], [8], [9], [10] based on 3D LiDAR and RGB camera information are common approaches to sensor fusion. Low level fusion achieved by projecting the point cloud provided by a 3D LiDAR onto the camera image plane, as in [11], is a common approach for multi-modal object detection. Detection is achieved via 3D point cloud projection and usage of YOLOv4 in [12]. The combination of PointNet and InceptionV3 is studied in [13] and a fusion network is presented in [14]. Approaches like Expandable YOLO [15] and Complexer-YOLO [16] fuse depth and RGB camera data to perform 3D object detection.

Although a lot of work has been done in multi-modal detection, the majority of applications concern autonomous vehicle perception, thus focusing on outdoor urban environments and aiming at pedestrian and vehicle detection. In addition the expenses in hardware regarding 3D LiDARs prohibit the application of these approaches to a wide variety of indoor mobile robots that rely on cheaper hardware.

The usage of 2D laser scanners waives the constraint regarding sensor cost and also reduces the computational power required to process the provided point cloud. The fusion of 2D laser and imagery data, in most works, is subject to constraints of small number of available classes. For example [17] uses an SSD as the backbone network for recognition. Specializing on specific classes of interest as in [18], that performs human detection via HOG focusing on imagery ROI provided by the laser, is also subject to the same constraint. A pretrained CNN is used for object classification from a mobile robot in [19] although focusing on one specific learned object and trying to follow it as a target, without low level sensor fusion. A disadvantage of 2D lasers is that information corresponding to the height of objects is lost due to sensor limitations.

The intuition behind the designed perception system presented here is that of a detector based on data fusion with a huge number of classes available out of the box. A very generalized detector can be obtained by taking advantage of a pretrained state-of-the-art CNN on ImageNet dataset and the fusion of 2D laser and camera data. However, as stated above, the objects height information is unknown due to the limitations of the 2D laser scanner. This information,

*This research was supported by project "Dioni: Computing Infrastructure for Big-Data Processing and Analysis" (MIS No. 5047222) co-funded by European Union (ERDF) and Greece through Operational Program "Competitiveness, Entrepreneurship and Innovation", NSRF 2014-2020.

*We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Jetson TX2 used for this research.

Spyridon Syntakas, Kostas Vlachos and Aristidis Likas are with the Department of Computer Science and Engineering, University of Ioannina, 45110 Ioannina, Greece. (Email: {ssyntakas; kostaswl; arly}@cse.uoi.gr)

that is essential for object localization on the image plane, is estimated via a voting schema during the classification of the object. Voting based on classification results is what enables the usage of a 2D laser scanner for 2D object detection in the presented work. As a result, the system as a whole is lifting sensor high cost and computational power limitations, while achieving object detection with 1000 available classes.

This paper focuses on the design and implementation of a perception system, that tackles the problem of object detection via fusion of 2D laser and RGB camera information. In the proposed method the two sub problems of object detection, i.e., object localization and object recognition, are solved in different spaces using different sensory modalities. Object localization takes place in the 3D world surrounding the robot by segmenting the point cloud given by a 2D laser scanner. By clustering the point cloud using the DBSCAN [20] algorithm, areas of high point density, i.e., the clusters, correspond to individual objects. By projecting the clusters on the image plane of the camera sensor using extrinsic and intrinsic sensor calibration, the localization of the object is propagated to that of the digital image plane. Regions of interest (ROI) in the form of several proposed bounding boxes, with known width and varying height, are obtained on the image plane and the second core component of detection, i.e., recognition, can be achieved with the usage of a pretrained CNN that performs classification of the content of the boxes. A cumulative voting schema among the boxes finalizes the class label and estimates the boxes height dimension, thus bounding boxes labeled with class and distance information are inferred.

CNNs such as ResNet [21] are trained on huge datasets, like the ImageNet [22], and are capable of recognizing hundreds of objects with high accuracy and fast speed of inference. These CNNs can also be easily adapted to specialised use cases via transfer learning and fine tuning, techniques that are nowadays commonplace. In addition, the classification of the detected object is enhanced with distance information provided by the laser scanner.

Compared to YOLOv3 approach for 2D object detection, the first clear advantage of the proposed detection system is the ability to detect 1000 objects due to ResNet18 being trained using ImageNet dataset, in comparison to YOLO that in most cases is trained to detect 80 objects available in COCO dataset. A second advantage comes from the easiness and convenience in adding more classes and retraining the backbone CNN, adapting the detection system to many different and specialized use cases. Transfer learning and fine tuning of pretrained models, available in many frameworks, is an easy task compared to retraining YOLO approaches in order to add new classes. The third advantage of the proposed approach comes from the fusion of the laser with the camera sensor, that gives accurate information of the detected objects distance from the robot. That particular advantage enhances the interaction with the perceived surrounding environment and increases the reliability.

A navigation system employing "artificial potential field" [23] is also implemented that achieves autonomous navigation, while the robot perceives the surrounding environment and interacts with it using the proposed detection system. The combination of the two systems makes the robot's perception easily adaptable to specialized applications and the robot's behaviour adjustable to the complexity and variability of unstructured and unknown workspaces. The knowledge of the distance of the detected objects makes the control of the robot a lot more convenient. The proposed object detection system, that combines laser and camera information, has been implemented and tested on a real robot, the Pioneer 3-DX, as well as in simulation. The Pioneer 3-DX is equipped with the Jetson TX2 embedded AI computing device by NVIDIA. With software developed in the ROS ecosystem [24], the robot perceives and navigates through the unknown workspace.

II. Approach

The proposed method is implemented as two distinct systems, i.e, the perception and the navigation system. Each of them constitutes a stand-alone system, but this paper focuses on the combined application of the two, which offers autonomous navigation and interaction within the perceived, initially unknown, workspace. As proof of concept the robot is given a goal pose and moves towards it, by incrementally defining a free path, while it interacts with objects of interest that are detected using the proposed method.

A. Perception System

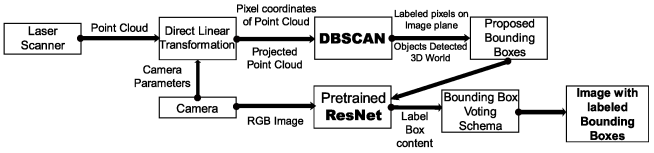
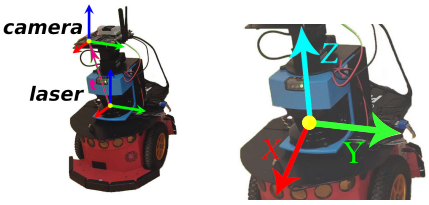


Fig. 1: Block diagram of the Perception System.

A block diagram of the proposed perception system is depicted in Fig. 1. The two heterogeneous sensory data, i.e., the 2D point cloud and the RGB image, are fused in a serial manner to achieve real time object detection. The sensory data, given the different sampling rates of laser and camera, are first synchronized using temporal information in the form of timestamps provided by the ROS messages, thus we ensure that the data provided by the two sensors correspond to the same scenery in front of the robot. The information fusion of the two sensors is achieved via the Direct Linear Transformation (DLT). Every point $\mathbf{P} = [X, Y, Z, 1]^T$ of the 2D point cloud that is provided by the 2D laser (where $Z = const$), is mapped to its corresponding $\mathbf{p} = [x, y, 1]^T$ pixel location on the image plane. The projection is performed as,

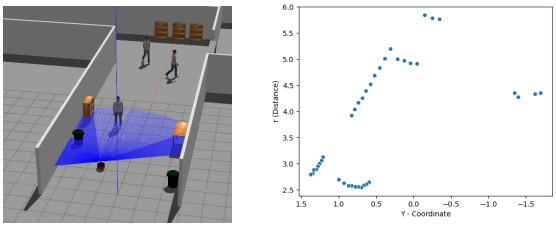
$$\mathbf{p} = K\, R\, [I_3| - \mathbf{t}]\, \mathbf{P} \tag{1}$$

with $K_{(3\times3)}$ denoting the Camera Matrix that contains the intrinsic parameters and $R_{(3\times3)}$, $-\mathbf{t}_{(3\times1)}$ corresponding to the pose of the camera frame, i.e., the 3D rotation and translation of the camera frame w.r.t. the laser frame, which are the extrinsic parameters.



(a) (b)

Fig. 2: (a) Pioneer 3-DX Sensor Frames, (b) Laser Frame



(a) (b)

Fig. 3: (a) Simulation World, (b) 2D Point Cloud

The six extrinsic parameters are obtained from the relative configuration of the sensors of the Pioneer 3-DX robot used in this application, that is depicted in Fig. 2. The five intrinsic parameters, that are contained in K, i.e., f corresponding to the focal length of the camera, a_x, b_y corresponding to the inverse height and width of the pixels in the photosensitive sensor and c_x, c_y corresponding to a translation between the center of the sensor and the center of the image plane, are obtained via camera intrinsic calibration which was performed using a ROS package implementation based on Zhang's Method [25]. With knowledge of the eleven parameters of the Direct Linear Transformation every point $\mathbf{P}$ referred from the laser frame can be mapped to its corresponding $[x, y]$ pixel location via the homography given in (1) or using the analytic form as,

$$x = width - (\frac{^cY\, f_x}{^cX} + c_x) \tag{2}$$

$$y = height - (\frac{^cZ\, f_y}{^cX} + c_y) \tag{3}$$

with $^cX, ^cY, ^cZ$ denoting each point of the point cloud coordinates w.r.t. the camera frame, f_x, f_y denoting the focal length in pixels and $width, height$ the image size.

The 2D laser scanner data are given initially in the form of corresponding Euclidean distances and angles of the vectors between the laser frame and the points on the object surfaces that the laser beams are reflected from, i.e., (r, θ). The 2D laser scanner has a fixed pose w.r.t. the robot's base link and the data are given in Cylindrical Coordinates, from which the corresponding Cartesian Coordinates can be easily obtained. The Cartesian Coordinates $[X, Y, Z]$ of all the points w.r.t. the laser frame constitute the point cloud dataset, which is a 2D layout of the surrounding environment providing a 2D model of the sensed workspace. The point cloud dataset is updated with a frequency equal to the sampling frequency of the 2D laser scanner and is made more informative with the addition of the Euclidean distance of each point, i.e, $PCL = \{X, Y, Z, r\}$, with PCL denoting the point cloud dataset and r the corresponding distance of each point.

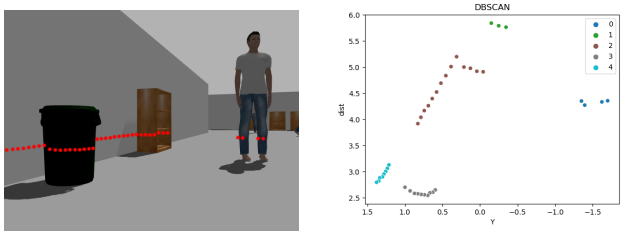
1) Object Localization: Object Localization is based on the DBSCAN algorithm. Using the Direct Linear Transformation (2), (3), the dataset is enriched with the $[x, y]$ pixel coordinates of each point $\mathbf{P}$, thus $PCL = \{X, Y, Z, r, x, y\}$. The $\{Y, r\}$ data, that correspond to the Y - coordinate and distance of each point of the 2D point cloud, are passed to the density based clustering algorithm DBSCAN. Every cluster inferred corresponds to one object detected by the laser in the 3D world, thus knowledge is obtained of which points of the point cloud belong to each object. The usage of $\{Y, r\}$ data facilitates clustering in a way that enhances accuracy in terms of correct number of objects. Given that the environment surrounding the robot is unstructured, there is no prior knowledge of neither the number nor the shape of the clusters. DBSCAN is the algorithm of choice for this application as it tallies with the complexity of the unstructured environment because it does not require the number of clusters as a hyperparameter and can discover clusters of arbitrary shapes. In addition, the speed of inference it provides, fits the real time application.

Given that DBSCAN returns K clusters, the dataset after the point cloud clustering, has the form described in Table I.

TABLE I: Clustered Point cloud for K objects surrounding the robot.

X	Y	Z	**Distance**	x	y	**Label(Object)**
X_1	Y_1	Z_1	r_1	x_1	y_1	0
X_2	Y_2	Z_2	r_2	x_2	y_2	0
X_3	Y_3	Z_3	r_3	x_3	y_3	0
X_4	Y_4	Z_4	r_4	x_4	y_4	1
X_5	Y_5	Z_5	r_5	x_5	y_5	1
...	...	...	...	...	...	...
X_N	Y_N	Z_N	r_N	x_N	y_N	K

Given the simulation environment depicted in Fig. 3 (a), the perspective 2D layout of the corresponding point cloud is plotted in 3 (b). The projection of the point cloud onto the image plane of the camera sensor and the inferred clusters, after DBSCAN application on the $\{Y, r\}$ 3D world information, can be seen in Fig. 4.



(a) (b)

Fig. 4: (a) Point Cloud projection, (b) Clustered Point Cloud

Given the $\{x, y, Label(Object)\}$ subset of the data set, we obtain knowledge of which image pixels correspond to each object detected in the 3D world by the laser scanner. As seen in Fig. 5, the clustering in the 3D world is propagated in the digital image 2D plane. The width of each object in pixel units on the image plane is the only dimension that can be directly inferred from the point cloud as the laser scanner is 2D. The height of the objects cannot be estimated from the 2D information.

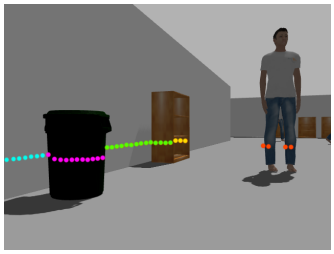


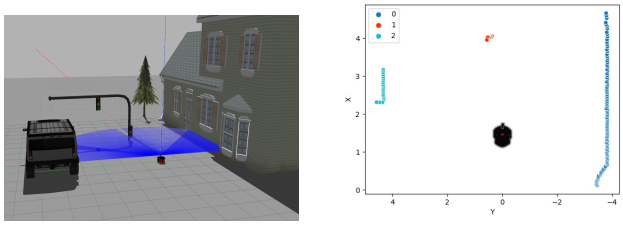
Fig. 5: Clustering of pixels.

A design choice is made to focus only on the closest cluster (object) to the robot. Thus the closest cluster is the only one that provides the ROI via projection on the 2D image plane from now on. This choice is made w.r.t. the application of the perception and navigation system, as the closest object is immediately interactable with the robot, either by manipulation or by sensor measurement.

So far, identification of the location of the object on the image plane is accomplished, but for object localization to be completed as a task, a bounding box must be drawn around the object, which is nontrivial since height information of the object is not known. To tackle this problem we use the voting approach described below.

2) Object Recognition: Object Recognition is based on ResNet. The intuition is to focus on the ROI on the image plane provided by the laser. To apply the ResNet, a bounding box should be specified to define the image window to be classified. Since we only know the width of the object, we consider several candidate bounding boxes of fixed width but different height, in order to enclose the object. A ResNet18, pretrained on the ImageNet dataset, is used, that takes as input the image window and outputs the corresponding 1000 class probabilities for each box. To estimate the object's height dimension, a voting schema based on the ResNet18 inference is applied among the several potential boxes. The voting schema outputs the object's class and the box that best encloses the object, in the sense that it more reliable provides the classification results. In general, most objects have a specific width to height ratio, despite the intra-class size variation. In addition, most workspaces contain specific categories of objects. Thus, the selection of the candidate bounding boxes can be assisted by an abstract prior knowledge of the robot's workspace and the possible objects in it. Below we provide an illustrative example.

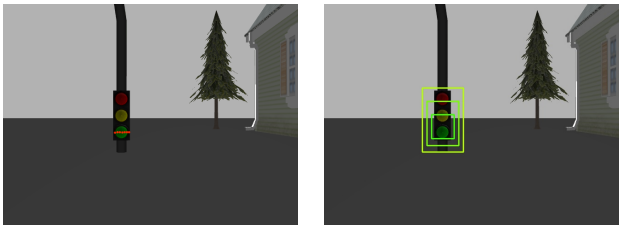
Given the environment simulation depicted in Fig. 6 with the clustered point cloud, the closest object to the robot that is captured by the camera sensor is that of the traffic light.



(a) (b)

Fig. 6: (a) Simulation World, (b) 2D Point Cloud

Applying the object localization method via fusion described above, the ROI width that corresponds to the traffic light can be seen in Fig. 7 (a). Three potential bounding boxes with increasing height dimension are placed around the ROI, given the width, as is depicted in Fig. 7 (b). The width of each box can be constant, i.e., the width provided by the laser, making the three boxes have the same width or incremented by a small scaling factor with each box as in Fig. 7 (b).



(a) (b)

Fig. 7: (a) Closest object ROI via DLT, (b) 3 candidate boxes

The content of each box is isolated from the rest of the image by cropping the image in the boxes perimeter. Thus, three images are created, one for each corresponding box. The images are forward passed to the pretrained, on the ImageNet dataset, ResNet18 model as seen in Fig. 8.

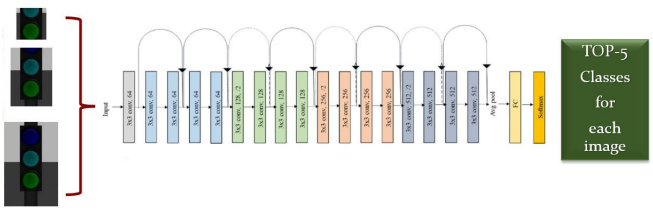


Fig. 8: ResNet18 Classifier.

For each one of the three images, 1000 class probabilities are inferred. The top 5 class probabilities of each image, i.e., the 5 classes that the ResNet18 is more confident about, are the ones that the proposed method uses via the following cumulative voting schema. Each one of the three boxes proposes 5 labels. A map data structure is used with the inferred labels as keys, as in Tables II, III, IV, V that correspond to the example of Fig. 6. As value for each

TABLE II: 1^{st}Box

Label	Prob.%
traffic light	64.87
loudspeaker	23.44
oscilloscope	1.94
digital clock	0.96
cassette player	0.74

TABLE III: 2^{nd}Box

Label	Prob.%
loudspeaker	72.32
traffic light	5.04
switch	4.05
face powder	2.13
lipstick	1.99

TABLE IV: 3^{rd}Box

Label	Prob.%
traffic light	71.67
loudspeaker	13.30
spotlight	1.57
switch	1.14
knee pad	0.77

TABLE V: Voting Score

Label	Score
traffic light	47.12
loudspeaker	36.35
switch	1.73
face powder	0.71
lipstick	0.66
oscilloscope	0.64
spotlight	0.52
digital clock	0.32
knee pad	0.26
cassette player	0.24

label/key, a score is used, computed as the average of the corresponding label's probabilities among the three boxes. The label with the highest value/score is chosen as the label for the detected object. The box that proposed the chosen label with the highest average probability is the chosen box, that is drawn around the objects extent. That way, the missing height information is approximated through voting of feasible boxes and the class of the chosen box is inferred.

Following this voting method, in the example that is discussed, the voted label is that of "traffic light" which stands correct and the third bounding box is the one chosen, as it proposed the chosen label with the highest probability. So accurate detection is achieved as shown in Fig. 9. Note that, since distance information of the object w.r.t. to the robot is also available via the laser scanner, the detected object is also labeled with this information.

Note that before the forward pass, the cropped images are preprocessed the same way the ImageNet dataset images are transformed prior to the ResNet training, i.e., resized as 256 x 256 images, normalized with $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$. After the transformation, the three images are stacked into a batch, which is transferred to the GPU. The pretrained ResNet18 model is also transferred to the GPU, thus the inference can be done using CUDA operations further enhancing the speed and making use of the Jetson TX2 computational power.

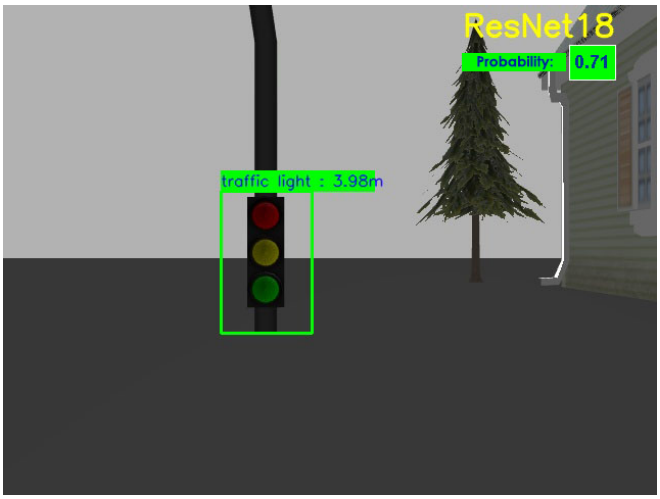


Fig. 9: Object Detection and Distance Information.

An algorithmic overview of the proposed perception system is shown in Algorithm 1.

Algorithm 1: Perception System

Input : PCL: Point cloud dataset $\{X, Y, Z, r\}$
Input : RGB_{image}: Camera RGB Image
Input : C: Camera Parameters
Output: Detected object $[label, box, distance]$
Data : $projected\ pixels$: Pixels on image plane from PCL projection
Data : $Cluster\ labels$: Labels of point cloud clusters
while PCL *and* RGB *image data are received* **do**
  Synchronize PCL and RGB image data temporally
  $projected\ pixels \leftarrow$ **DLT**(PCL, C)
  $PCL \leftarrow$ **EnchancePCLwithInfo**$(projected\ pixels)$
  $Cluster\ labels \leftarrow$ **DBSCAN**$(PCL[Y], PCL[r])$
  $PCL \leftarrow$ **EnchancePCLwithInfo**$(Cluster\ labels)$
  Get closest cluster, cluster with $\min(r)$ from PCL dataset
  ROI (cluster width in pixels) $\leftarrow$ **GetWidth**$(closest\ cluster)$
  Potential Boxes $\leftarrow$ **PlacePotentialBoxes**(ROI)
  boxes content $\leftarrow$ **CropImage**$(RGB_{image}, Potential\ Boxes)$
  foreach *box content* **do**
    box content $\leftarrow$ **preprocess**$(box\ content)$
    1000 class probabilities $\leftarrow$ **ResNet**$(box\ content)$
    Get Top 5 class probabilities for box content
  end
  box,
    label$\leftarrow$ **VotingSchema**$(Top\ 5\ probabilities\ of\ all\ boxes)$
  Draw box and **mark** with label and distance (r) information
end

B. Navigation System

The proposed approach to motion planning achieves autonomous navigation without prior knowledge of the robot's workspace by employing "artificial potential field" [23]. The robot is given a goal position and orientation and navigates through the unknown unstructured environment by updating a free path incrementally until the goal pose is reached by the use of velocity commands as a control law.

Following the artificial potential field method, we assume that the robot is an artificially positive charged particle with configuration $\mathbf{q} = [x, y, \theta]^T$ that moves according to the forces that are applied to it by the total potential field generated by the goal configuration, that is assumed negative charged and the obstacle configurations, that are assumed to be positive charged.

Commanding a velocity vector proportional to the gradient of the potential, the robot moves to the goal configuration avoiding the obstacles and responding to the dynamics of the workspace. The velocity commands denoted as $\mathbf{u} = [u_x, u_y, u_\theta]^T$ are given as,

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \mathbf{u} = -k\nabla U(\mathbf{q}) = -k \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{\partial U}{\partial \theta} \end{bmatrix} \tag{4}$$

with k denoting the gain and $U(\mathbf{q})$ the total potential at configuration $\mathbf{q}$.

In the absence of obstacles within a distance threshold, a P-controller with a higher gain than that of attractive field controller is activated. The switch between the two controllers makes the robot's movement more smooth and more responsive throughout navigation.

III. Experimental Results

Both the perception and the navigation systems have been implemented using ROS and the combined application of the two has been tested both in real case scenarios as

well as in simulation. The computational graph of the ROS implementation is depicted in Fig. 10.

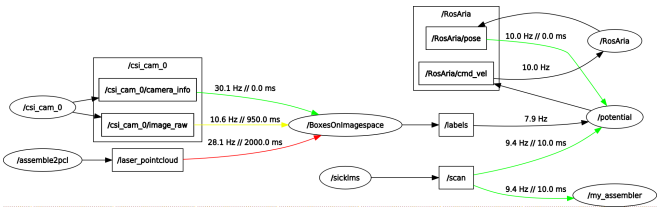


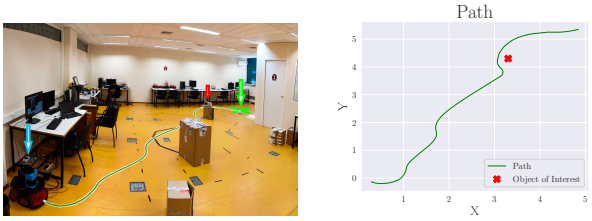
Fig. 10: ROS Graph of Computations.

The node $/BoxesOnImageSpace$ implements the perception system. The node takes as input the point cloud, the raw RGB image and the intrinsic and extrinsic parameters and performs the object detection in real time as described. The navigation system implemented in the node named $/potential$ has as input the odometry data provided by the $/RosAria$ node, as well as the distances of closest objects provided by the laser scanner. The last input to the navigation system is the inferred label of the closest detected object provided by the $/BoxesOnImageSpace$ node.

The mobile robot is equipped with the LMS200, a 2D infrared $(905nm)$ laser scanner with maximum range, with 10 % reflectivity, at 10m, 180° aperture angle and angular resolution at $0.5°$ at 75 Hz. The camera that provides the RGB imagery data is a CMOS 5 megapixel (2592×1944) image sensor with OmniBSI-2TM technology and has a maximum transfer rate of 30 fps for 5 Mpixel. The scan mode is progressive with lens size of $1/4$" and non-linear lens chief ray angle of $29.7°$. The image area is $3673.6\mu m \times 2738.4\mu m$ with pixel size of $1.4\mu m \times 1.4\mu m$. The camera is attached to the NVIDIA Jetson TX2 Module that acts as the computational unit of the robot.

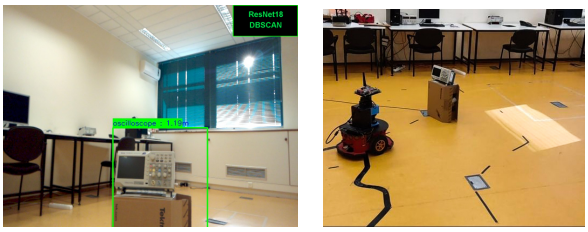
In order to experimentally validate the proposed system, the robot navigates autonomously in an unknown workspace using the navigation system until a given goal pose is reached. If a given object of interest is detected via the perception system the robot stops and interacts with it in the form of a sensor measurement. As seen in Fig. 11(a) the object of interest is an oscilloscope placed in the laboratory environment, whose pose is unknown to the robot, and the goal pose is that of $\mathbf{q_{goal}} = [5m, 5.5m, 15°]^T$. The robot navigates, avoiding perceived obstacles, following velocity commands derived by the navigation system until it reaches the goal pose. While navigating the detection system perceives the oscilloscope and the robot stops for 10 seconds, captures an image and then continues navigating until the goal pose is reached. The path that the robot follows can be seen in Fig. 11. The corresponding trajectories are depicted in Fig 13.

The detected object as captured by the camera sensor can be seen in Fig. 12 as well as the pose of the robot during the sensor measurement. The results of more experiments that took place are given in Fig. 14.



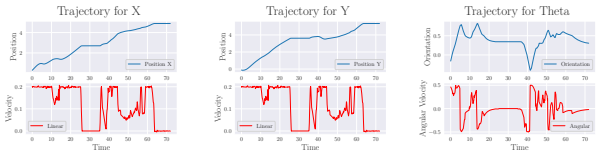
(a) (b)

Fig. 11: (a) Experiment Environment, (b) Path to goal



(a) (b)

Fig. 12: (a) Detection of Oscilloscope, (b) Pose of the robot



(a) (b) (c)

Fig. 13: Trajectories: (a) X, (b) Y, (c) θ

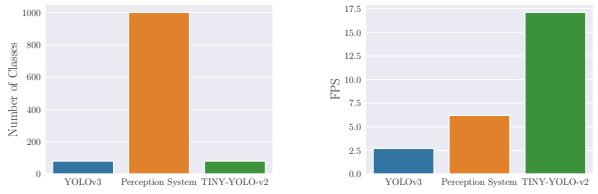
(a) (b) (c)

Fig. 14: Detection cases: (a) Joystick, (b) Space heater, (c) Car

IV. Conclusion

In this work, the proposed perception and navigation systems achieve the goal of autonomous navigation of a mobile robot in an unknown environment while it interacts with objects of interest. The fusion of 2D laser and camera information successfully completes the task of real time object detection and distance evaluation, with 1000 classes of objects available, which is a significant improvement compered to YOLOv3 and TINY-YOLO-v2, see Fig. 15 (a). The proposed voting schema provides an estimation of the objects height, although not always successfully, despite the fact that it is not directly inferrable from the point cloud. The implementation also achieves reliable sensor synchronization as it manages to temporally match the heterogeneous sensory

data with accuracy. In terms of speed comparison of the real time 2D object detection, given the specific hardware specs used in this work, the proposed system $(6.2fps)$ is faster than YOLOv3 [26] $(2.7fps)$, but slower than TINY-YOLO-v2 model [26] $(17.1fps)$, as seen in Fig. 15 (b).



(a) (b)

Fig. 15: (a) Classes Comparison, (b) Fps Comparison

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 06 2016, pp. 779–788.

[2] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014. [Online]. Available: https://arxiv.org/abs/1405.0312

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1612.00593

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018.

[5] H. Yin and C. Berger, "When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets," 10 2017.

[6] D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, C. Glaser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, mar 2021. [Online]. Available: https://doi.org/10.1109%2Ftits.2020.2972974

[7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," 2016. [Online]. Available: https://arxiv.org/abs/1611.07759

[8] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," 12 2017.

[9] C. Ruizhongtai Qi, W. Liu, C. Wu, H. Su, and L. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," 11 2017.

[10] T. Kim and J. Ghosh, "Robust detection of non-motorized road users using deep learning on optical and lidar data," 11 2016, pp. 271–276.

[11] G. H. Lee, J. D. Choi, J. H. Lee, and M. Y. Kim, "Object detection using vision and lidar sensor fusion for multi-channel v2x system," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 1–5.

[12] X. Xu, L. Zhang, J. Yang, C. Cao, Z. Tan, and M. Luo, "Object detection based on fusion of sparse point cloud and image information," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[13] G. Melotti, C. Premebida, and N. Gonçalves, "Multimodal deep-learning for object recognition combining camera and lidar data," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2020, pp. 177–182.

[14] S. Pang, D. Morris, and H. Radha, "Clocs: Camera-lidar object candidates fusion for 3d object detection," 2020. [Online]. Available: https://arxiv.org/abs/2009.00784

[15] M. Takahashi, Y. Ji, K. Umeda, and A. Moro, "Expandable yolo: 3d object detection from rgb-d images," in *2020 21st International Conference on Research and Education in Mechatronics (REM)*, 2020, pp. 1–5.

[16] M. Simon, K. Amende, A. Kraus, J. Honer, T. Samann, H. Kaulbersch, S. Milz, and H.-M. Gross, "Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds," 06 2019, pp. 1190–1199.

[17] A. Mulyanto, R. Indra, P. Prasetyawan, and A. Sumarudin, "Implementation 2d lidar and camera for detection object and distance based on ros," *JOIV : International Journal on Informatics Visualization*, vol. 4, 12 2020.

[18] B. Wu, J. Liang, Q. Ye, Z. Han, and J. Jiao, "Fast pedestrian detection with laser and image data fusion," 09 2011, pp. 605 – 608.

[19] V. Popov, S. Ahmed, A. Topalov, and N. Shakev, "Development of mobile robot target recognition and following behaviour using deep convolutional neural network and 2d range data," *IFAC-PapersOnLine*, vol. 51, pp. 210–215, 01 2018.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.

[24] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," vol. 3, 01 2009.

[25] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[26] M. Bjelonic, "YOLO ROS: Real-time object detection for ROS," https://github.com/leggedrobotics/darknet$_ros$, 2016 − −2018.