University of Zurich[UZH]

Bruno Rodrigues, Lukas Müller, Eder J. Scheid,
Muriel Franco, Christian Killer, Burkhard Stiller

LaFlector: a 2D LiDAR-based Indoor Tracking Approach

TECHNICAL REPORT – No. IFI-2022.03

February 2022

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

LaFlector: a 2D LiDAR-based Indoor Tracking Approach

Bruno Rodrigues, Lukas Müller, Eder J. Scheid,
Muriel F. Franco, Christian Killer, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
E-mail: [rodrigues,scheid,franco,killer,stiller]@ifi.uzh.ch, lukas.mueller11@uzh.ch

***Abstract*—Light Detection and Ranging (LiDAR) is used in various applications, from mapping the environment's topography to self-driving vehicles. Among such applications, the use of LiDAR for indoor tracking and quantifying visitors' interest (and ensuring safe distancing) is still not widely explored. Technologies based on wireless signals and video captures are typically used for indoor tracking, but they deficits concerning the lack of accuracy of captured signals or the lack of user privacy in the case of traditional surveillance cameras. Despite introducing tracking challenges inherent in the detection based on light reflection, LiDAR-based approaches represent a relatively low-cost solution for accurate indoor tracking. Thus, this paper presents LaFlector, a LiDAR-based indoor tracking system introducing tracking heuristics capable of detecting, classifying, and tracking several objects simultaneously, which are recorded and dynamically displayed in a 2D coordinate system. LaFlector was evaluated based on a low-cost 2D LiDAR hardware (Slamtec Mapper M1M1) and capable of detecting moving objects with high precision, showing in practice that a LiDAR-based can be used to track visitors' interest and count the number of moving objects.**

***Index Terms*—Indoor Tracking, Security, LiDAR**

I. Introduction

Tracking visitor's interest across a Web-page is a relatively straightforward task in contrast to the physical, analog world. For example, digital tracking can be accomplished by tracking mouse movements [35] that, in its turn, can be used to predict user experience [25] and optimize the sales process. However, these insights are equally crucial in physical stores to understand how customers interact with a product or service, which attracts most customers' interest. In addition, counting the number of persons in a closed environment recently became crucial to reduce the spread of viruses (*e.g.,* SARS-CoV-2) by ensuring minimum social distancing. Therefore, the research on approaches and systems to track people within a restricted area determines a viable research field with commercial and public health benefits.

Several indoor tracking approaches and techniques based on wireless signals and video feeds exist. Camera-based approaches often rely on the employment of one or more cameras [43], [31], [19], that while providing high accuracy over wireless signal-based approaches, they pose privacy concerns [16], since the individual's data is being processed and captured by a device. Furthermore, wireless approaches are susceptible to various interference from other wireless signals and multi-path fading depending on the internal environment (*e.g.,* WiFi probing [36] and Bluetooth beacons [32], [30]) and are also susceptible to privacy concerns (*e.g.,* under the EU General Data Protection Rule (GDPR) [29]). Thus, approaches that do not affect an individual's privacy to perform such a tracking are required, but still lacking.

LiDAR scanners use laser's reflection to estimate the distance between the sensor and the object without incurring privacy concerns, based on the Time Difference of Arrival - TDoA [43]. Furthermore, LiDARs are applied in different fields from mapping a given environment's topography [42] to self-driving vehicles [41]. For example, in autonomous driving systems, a LiDAR (combined with other sensors) can be used to create a map of a vehicle's surroundings. Therefore, a precise tracking system based on LiDAR can be used for on-the-spot tracking at expositions, trade shows, or stores to acquire valuable data for exhibitors and companies enhancing advertisements campaigns.

Therefore, this work developed LaFlector, a LiDAR-based indoor tracking system that introduces heuristics capable of detecting, classifying, and tracking several objects simultaneously, recorded and dynamically displayed in a 2D coordinate system. LaFlector is designed as a modular engine that receives tracking data from one or more distributed LiDARs for subsequent positioning data processing. Objects are detected and distinguished considering the dynamic mapping of the environment (limited by the LiDAR's range) and the logical path of moving objects. Unique classification is based on heuristics that consider previous positions, path, and speed of dynamic objects to predict their next steps. Furthermore, data processed is output in a standardized form, consisting of timestamped X and Y coordinates so that data can be correlated with other data sources (*e.g.,* wireless or video). In addition to documenting the procedures here for the design, implementation, and evaluation of LaFlector (single evaluations available in video [24]), the source code is open-sourced [23] for future contributions.

The remainder of this paper is structured as follows. Section II describes the technical aspects of the LiDAR technology and related concepts. While Section III presents the design and implementation of LaFlector, Section IV details the evaluation performed and discusses its results. Section V compares related work with LaFlector. Finally, Section VI summarizes this paper and lists future work.

II. Background

While the background on LiDAR technology is presented, including its strengths and weaknesses, properties on data management (reading, segmenting, classifying, and tracking) are introduced. Further, it presents details on object tracking with LiDAR.

LiDAR is a sensing technology that observes ranges (*i.e.,* distances) backscattered by a laser, in which a sensor is used to measure the backscattered light in the environment [20]. Thus, the distance can be estimated by measuring the time required for the laser light to return, given the speed of light. The basic principle for LiDAR originated in the early 1960s, and military institutions drove its technical developments for measuring distances and weapon guidance [22].

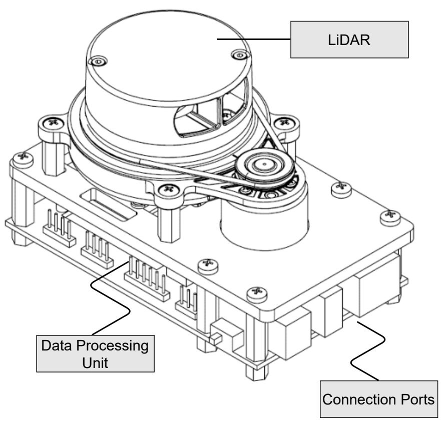


Fig. 1: Slamtec Mapper M1M1 LiDAR Device [38]

The device employed in the prototype of `LaFlector` here is the Slamtec Mapper M1M1 [38] (*cf.* Figure 1). It is a 2D LiDAR able to perform 7,000 measurements per second and to achieve a maximum range distance of 20 m with a 5 cm resolution in both indoor and outdoor environments. A Data Processing Unit (DPU) processes data in real-time, outputs a high-precision map, and poses with a maximum data mapping area of 90,000 m^2. Further, it provides a 10/100 Mbps Ethernet port, a 802.11a/b/g/n/ac WiFi module, and Universal Serial Bus (USB) for communication. Thus, it is suitable for the tracking of visitors in presentations stands without introducing deployment complexities.

1) Benefits and Drawbacks: In contrast to tracking solutions such as WiFi-based [30] and Bluetooth-based [32], LiDAR tracking offers the following **benefits**:

- **Precision:** LiDAR scanners present a high precision in comparison to WiFi-based and Bluetooth-based solutions. *E.g.,* [30] implements a WiFi-based system, which achieved a deviation of 1.1 m at a maximum distance of 10.8 m under good conditions, since Received Signal Strength Indicator (RSSI) values are not reliable, whereas LiDAR devices can achieve a precision of 5 cm.
- **Capturing objects, not devices:** While the Android OS has enabled Medium Access Control (MAC) address randomization by default since version 10 [21], Apple has extended MAC address randomization with iOS 14 [12]. Thus, device fingerprinting will become increasingly difficult, which does not affect LiDARs. Moreover, it allows tracking objects without an intelligent device, unlike Bluetooth or WiFi tracking.
- **Privacy:** Data protection regulations have become increasingly strict, which "light" adheres to automatically. Tracking via WiFi or Bluetooth data could be complicated because the MAC address is classified as personal data according to the GDPR [5]. Thus, LiDAR tracking does neither process nor even holds information concerning tracked objects identities.
- **Line of Sight:** LiDAR sensors can provide accurate results over long distances (up to 200 m in vehicles) [18]. Hence, LiDAR's are often used in vehicles in combination with other sensors [40].

However, LiDAR techniques show **drawbacks** due to their dependency on physical properties of light:

- **Hidden objects:** Due to the principle of operation, a LiDAR scanner cannot detect objects that are behind other objects. This limitation can be compensated by a second scanner capturing objects from a different angle.
- **Large data sets:** Depending on the resolution of the scanner (and also the aperture angle and the use case), the number of data points collected can become extensively large and data points deriving from static objects that are usually not of interest. Henceforth, the post-processing of LiDAR's data can be more extensive than that of other measurement techniques (*e.g.,* WiFi-based [30] and Bluetooth-based [32]).
- **Robustness:** The performance of a LiDAR scanner strongly decreases in heavy rain or fog, which can reduce the detection rate of objects by up to 50% [6], [18].

2) Legal situation: The Slamtec Mapper M1M1 [38] device considered in this paper contains laser with a power of 28 watts; thus, it is classified according to the IEC 60825-1 standard [14] as a Class 1 laser. Further, even powerful devices, such as the Slamtec M2M1 model [39] with an extended range of 40 m, is classified as Class 1. According to the Federal Office of Public Health of Switzerland a Class 1 laser is safe, and even direct exposure to the laser beam will not cause any damage to the human eye [27]. Thus, due to the classification as a Class 1 laser, such a LiDAR scanner can be used anywhere and at any height (even eye level).

A. Object Tracking using LiDAR

Tracking can be divided into passive and active tracking. Active tracking requires participation and configuration of the client for a specific measurement technique. Passive tracking works without adaptations in the environment to be analyzed. This does not mean that signals from devices to be tracked are not necessary. However, these signals occur without pre-configuration. For example, in the case of WiFi tracking, these are outgoing probe requests from the clients [36]. In particular, LiDAR tracking is a passive tracking technique as it works independently of the tracked object's signals. This part is about the steps that are usually performed to identify and track objects from the unprocessed laser data. Since the scanner used in this thesis is a 2D scanner, no 3D-specific process steps are observed, such as surface matching.

1) Reading Data: Due to the design of a 360-degree LiDAR scanner [38], it is convenient to use the Polar Coordinate System to specify data points. In such a system, a data point is defined as the distance (r) and angle (ϕ) from a starting point resulting in a tuple (r, ϕ). The starting point corresponds to the position of the scanner. In the first step, these data are converted into Cartesian coordinates (x, y), where $x = r \sin \phi$ and $y = r \cos \phi$. It is sensible to filter out non-valid data points to reduce the number of data points and increase the relative data quality.

2) Segmenting Data: A data set of a LiDAR scanner contains reflecting objects in a room, too. As a rule, static objects, such as walls, windows, or fixed objects, are no longer of interest after the initialization phase. By subtracting the set of static objects from a newly acquired data set, the resulting difference is the set of points associated with objects in motion. If the difference is zero, no moving objects exist in the room that could be tracked [37].

3) Classifying Data: If an object is detected (*i.e.,* not belonging to the static objects' data set), it must be classified. Based on a 2D scanner, the possibilities are limited by the lack of surface matching. Thus, it is needed to consider the relation between the distance and number of data points to distinguish between static and dynamic objects. For example, if an object is close to the LiDAR scanner and covers fewer data points than expected, it will not be identified as a human.

4) Tracking Data: If an object could be identified as a human, the movement is recorded. However, if persons leave the room or exceed the maximum measurement distance, a LiDAR scanner cannot distinguish person A from person B. Therefore, to solve biometric features [2] or motion analysis techniques, it should be applied to maintain the tracking even if two people cross each other.

III. LaFlector's Design

LaFlector (a wordplay from *laser beam* emitted by the LiDAR and the laser's *reflection*) is designed to reach a flexible deployment and operation of individual hardware components (*i.e.,* sensors, nodes, and sink). It is based on a distributed architecture in which multiple nodes are connected to the respective LiDAR sensors sending data to a sink. The sink is responsible for collecting positioning data and determining the positioning of dynamic (people) and static (environment) objects. Thus, pre-processing steps are performed as early as possible to avoid bottlenecks downstream and reduce the amount of data forwarded by the sink [23]. The following sections the assumptions defined for its design, detail LaFlector's design, and provide implementation details. LaFlector's code is open-source[1].

[1]LaFlector's source-code: https://gitlab.ifi.uzh.ch/rodrigues/laflector

A. Assumptions

These are considered for an ideal operation of LaFlector:

- **LiDAR's Placement:** The scanner is to be placed to minimize dead spots, horizontally aligned with the floor.
- **LiDAR's Height:** Object classification expects a solid body with a pre-defined width concerning the measured distance. Thus, the laser must be operated at upper body height. Placing the scanner at foot level would lead to incorrect classifications.
- **Disturbances:** No other devices are continuously transmitting signals at the same wavelength to which the LiDAR scanner could respond to. This applies, for example, to laser pointers or infrared remote controls.

B. Architectural Components

LaFlector's design (*cf.* Figure 2) is based on a distributed architecture consisting of a server acting as a sink and one or multiple nodes. A node is connected to a LiDAR device over WiFi or Ethernet, depending on the node's capabilities. This node-sink design was chosen to ensure extensibility and allows for multiple nodes (*i.e.,* LiDAR and data collection) to be run on one sink (*i.e.,* data processing).

- **LiDAR:** consists of no other visible sub-components and is basically handled as a black box. The Slamtec SDK provides the needed functions to work with the device.
- **Node:** The node's software runs either on traditional X86 or ARM architectures (*e.g.,* System-on-Chip devices, such as ASUS Tinkerboard or Rasperry PI devices) being directly connected via a USB port to the LiDAR sensor. Subcomponents include the SDK Parser, Socket Client, and Data Pre-Processing as follows:
  - The *SDK Parser* collects the data from the LiDAR and passes it to the pre-processing.
  - The *Socket Client* connects to the Socket Server of the sink and receives commands.
  - The *Data Pre-processing* converts the distance (r) received and the angle (ϕ) for each data point into timestamped Cartesian coordinates (x, y).
- The **Sink** is the largest component consisting of six subcomponents.
  - The *Socket Server* receives data from one or more nodes via a TCP connection.
  - The *Command Line Interface (CLI)* provides an interface capturing user inputs, forwarding them to the socket server, and controlling the behavior of the Data Processing.
  - The *Data Processing* is responsible to segment, classify, and track dynamic objects in the line-of-sight of multiple nodes.
  - TRhe *Database* supports the data processing component with a time-series database (*e.g.,* InfluxDB) to store data points delivered by the node.
  - The *External (Ext.) Storage* maintains data being exported as a standardized timestamped coordinate to be contrasted or combined with different tracking sources,

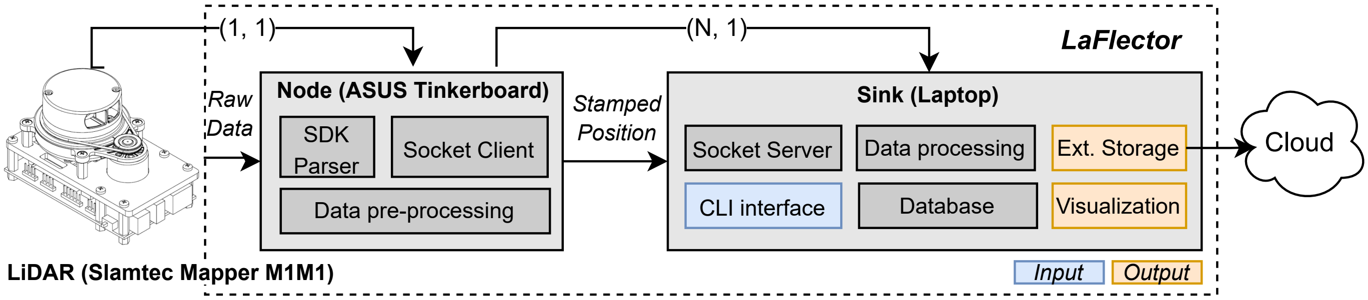


Fig. 2: LaFlector's Node-Sink Architecture

for instance, to support measurements from wireless tracking (*e.g.,* ASIMOV [30] or BluePIL [32]).

– The *Visualization* provides instant feedback of objects.

To account for the data streaming in the distributed node-sink architecture, data received by the sink work in a retrospective fashion, *i.e.,* work with a local subset of the data that only uses values from the past. Thus, a rolling time window is implemented only containing values from the interval $[t_c - \Delta t, t_c]$, with t_c being the current time and Δt the window size, determined by the update frequency of the nodes and the expected variance of the data (timestamped position).

C. Merging Upstream Data

To compute a location from pre-processed positions, a strategy has to be determined to merge data streams in a rolling time-window from different nodes. Bilinear interpolation is used to fit individual upstreams sent by nodes at different cycles with a weighted average of the nearest coordinates. In a rolling time-window of Δt s, pre-configured by the sink, nodes accumulate timestamped (x, y) coordinates until t_c is reached and any missing (x, y) coordinate in the rolling time window is estimated by the interpolation at the sink. LaFlector builds upon the assumption that the update cycles of individual nodes are short enough to legitimize the interpolation between two data points as a valid estimation of the true state of the system. To enable the inference of position values at a certain point in time, measured values must be available preceding and succeeding a said point.

D. Location Algorithm

If the same object is detected by multiple nodes, a measurement or a set of measurements should be associated with this object. This association considers the position *i.e.,* placement of LiDARs and timestamped coordinates of tracked objects reported by nodes. However, as LiDARs do not gather information from the tracked objects *i.e.,* persons, the challenge is to model the movement pattern so that the same person is not accounted multiple times. Thus, it is considered that people follow an almost constant velocity pattern in an indoor environment, in which there are no sudden acceleration movements in contrast to their own average velocity. A nearly constant velocity model is given as follows [9]:

$$x(k) = Fx(k-1) + w(k-1) \tag{1}$$

where F is the state transition matrix of each person in a given time t_c, and $w(k-1)$ is a zero-mean white noise Gaussian process with covariance Q. The white noise is a statistical model defined in [3] that represents the "near constant" velocity variation. F and Q are given as:

$$Fx = I \otimes \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix} \tag{2}$$

and:

$$Q = I \otimes \begin{matrix} \frac{T^3}{3} & \frac{T^3}{2} \\ \frac{T^3}{2} & T \end{matrix} \tag{3}$$

where Q is the power spectral density [9] of the process noise, T is the scan time, $\otimes$ is the Kronecker product and I is the identity matrix. This expresses a belief that the object will have moved in the direction gathered from the last measurement and that the velocity of said movement will not have changed abruptly. Values obtained from the previous step in the pipeline are used as current observations. Therefore, the following observation vector is used: $z_k\left(\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right)$ Then, the following observation matrix is defined to express that the observation corresponds to the x and y coordinates of the state vector: $r_k\left(\begin{smallmatrix} 0.3 & 0 \\ 0 & 0.3 \end{smallmatrix}\right)$ Those values used were determined experimentally and work for the evaluation scenario in this paper. The following heuristics are applied after data collected by nodes are prepared.

1) Segmentation of Environment: Data segmentation consists of two tasks. While the first task is to detect static objects *i.e.,* environment, the second task compares a different (x, y) in $T+1$ with the static objects to determine moving objects. An implementation detail specific to the Slamtec Mapper M1M1 LiDAR Device [38] is that matching has to be done with every laser rotation, and each rotation contains hundreds of points. Thus, the angle and distance list are written to a Python dictionary that has the advantage a $\mathcal{O}(1)$ lookup time.

```python
for angle in scan_dict.keys():
temp_angle = angle
while temp_angle not in static_dict.keys():
    temp_angle = temp_angle + 0.001
```

```
    if temp_angle - angle > 0.005:
        break
if temp_angle in static_dict.keys():
    # do not allow points that lay behind static
        objects, if this happens, something is wrong
         in the
    # room and if the distance difference is bigger
        than a threshold it must be a moving object
    if static_dict.get(temp_angle) > scan_dict.get(
        angle) and (
            static_dict.get(temp_angle) - scan_dict.
                get(angle)) > self.
                _static_moving_distance:
        angle_list_moving.append(angle)
        distance_list_moving.append(scan_dict.get(
            angle))
```

Listing 1: Comparing Dynamic to Static Objects

This snippet shows the acquisition and construction of the Static Dictionary. This dictionary remains unchanged for the rest of the current measurement. Once the static dictionary is created, detection of moving objects can start. Angle and distance are now retrieved again per rotation and written to the moving objects dictionary. The moving dictionary is compared with the static dictionary and as soon as a threshold (parameter: static-moving-distance) is exceeded, the value is stored in a separate angle and distance list. These values are then checked and classified in the next step.

2) Object Classification: Starts with the two lists *angle_list_moving* and *distance_list_moving*, which contain the segmented measured values of the last three rotations. The first task of classification is to distinguish between multiple objects. To do this, one starts at the first point and compares it with following one. If the distance between the points is smaller than a certain threshold (parameter: split-distance) it must still be the same object. The check points are shifted and it is checked again. This process is repeated until the object is completely captured. After the objects from the lists are separated, they are identified in the next step.

3) Object Identification: Decide whether a set of points that were previously classified as a human object, can be assigned to an already existing object (*i.e.,* same person). The set of points is matched with all objects that are still active.There are three cases that need to be distinguished:

- **No object match**: The points do not correspond to any previously known object. The object must be new accordingly.
- **One object match**: The points can be clearly assigned to an object. In this case, the new position of the object is set.
- **Two or more matches**: In this case, it is not possible to decide which points belong to which object. Assuming that the objects continue to move in the same direction, the expected positioning is determined by means of a direction vector derived from the last way-points.

4) Language and Libraries: The Sink was written in Python since Python has more and easier to use extensions for data collection, manipulation, and visualization. That is why Python is the most used programming language in the field of Data Science [15]. Python interpreter was running in version 3.9. Apart from the Python standard libraries, the following additional packages were used:

- **InfluxDB:** To read the values of the node from the Influx database, this package was used in version 5.3.1. The package is also officially recommended by Influx when using Python [13].
- **PyYAML:** Instead of writing a simple YAML parser like in the node, the PyYAML package in version 5.4.1 was used. This is also because the parameters for the sink are grouped and therefore a bit harder to read.
- **NumPy:** This very comprehensive package provides data structures and functions for mathematical operations. It was used for the conversion of cartesian and polar coordinates and the midpoint and distance determination. It is also a dependency for the matplotlib library. Version 1.20.1 was installed for this thesis.
- **Matplotlib**: To visualize the results the matplotlib library (version 3.3.4) was used. It can represent numpy arrays in coordinate systems.

5) Output: The output is available in two forms. **Plot**. The position of the detected objects and their direction vectors are displayed in a dynamic X/Y coordinate system. This output form serves for visualization. **Logger**. The system has a logger which creates a file with the start time at startup. The verbose level determines which data is written to the log file and which is not. This output is used for debugging. Furthermore, the logger function can be quickly exchanged with a database client, so that the corresponding data ends up in a database instead of the log file. This is in case the data should be further processed or combined with another data source.

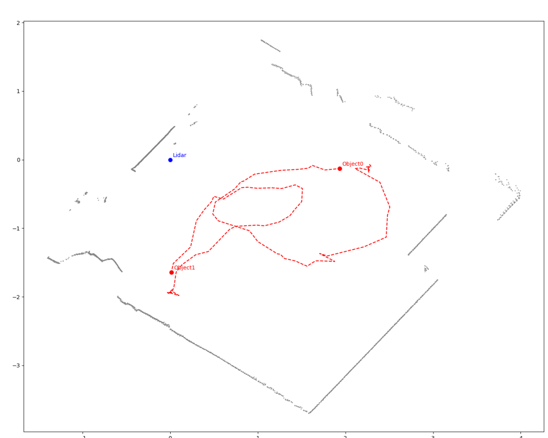


Fig. 3: Scenario #2: Two Persons Tracking Plot

IV. Evaluation and Discussion

The evaluation of LaFlector was conducted in a 18 m^2 room. The windows were covered and there were static objects in the room (*e.g.,* sofa, table, TV, and chairs). Runs were performed with the parameter's default value. To simulate objects' appearance, both entering the measurement room and spontaneously joining the measurement height were attempted. To test an object's disappearance, the tracked person left the room or suddenly went below the measurement height. The measurement height was 144 cm above the ground for all test runs. Four scenarios were run five times each:

1) **Single Person Tracking**: One person moves through the room at a walking pace. There are no other moving objects. This scenario is the basis for further evaluations.
2) **Two Persons Tracking**: Two people move simultaneously at a walking pace in the room. They never stand behind each other and are always at least one meter apart. Their paths might intersect at different times.
3) **Two Persons Crossing**: Two people cross paths. For a short time, one person stands behind the other. They do not make any hard changes on their direction.
4) **Single Person Behind Static Object**: A static object stands in the room, which was captured at the beginning. A person moves behind the object for a short amount of time and continue the walking direction.

These runs were measured using the following criteria. A criterion can be considered as passed or failed.

- **Positioning (Pos.)**: The tracking is accurate and continuous. The criterion is considered as passed if the object is tracked accordingly to the real position and there are no unexpected jumps in the way-points.
- **Classification (Class.)**: The person is recognized and correctly classified. No static objects or objects are classified as human objects. The criterion is considered as passed when the number of human objects is correctly detected.
- **Identification (Id.)**: The same object is always identified as the same. If a person reappearing from behind a static object is classified as a new object, the criterion is considered failed. If two people cross each other and the system can no longer assign the objects because the object identification was lost, the criterion is considered failed.

A. Results

Table I presents how often a criterion was not met in 5 runs. In general, it can be stated that the location algorithm worked as expected for most scenarios. However, there were failures in the identification and classification in Scenarios #3 and #4, which are explained in the next sections. Results

TABLE I: Failures in Each Scenario

#	Scenario	Pos.	Class.	Id.
1	Single Person Tracking	0	0	0
2	Two Persons Tracking	0	0	0
3	Two Persons Crossing	0	0	2
4	Person behind Object	0	0	1

are represented with the created plots of the LaFlector. A series of snapshots visualize the first scenario. These snapshots were taken at a regular interval (every 5 seconds) during the run, in which the last snapshot represents the remaining three

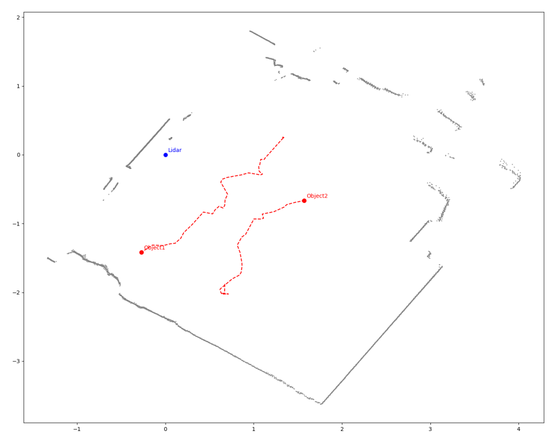


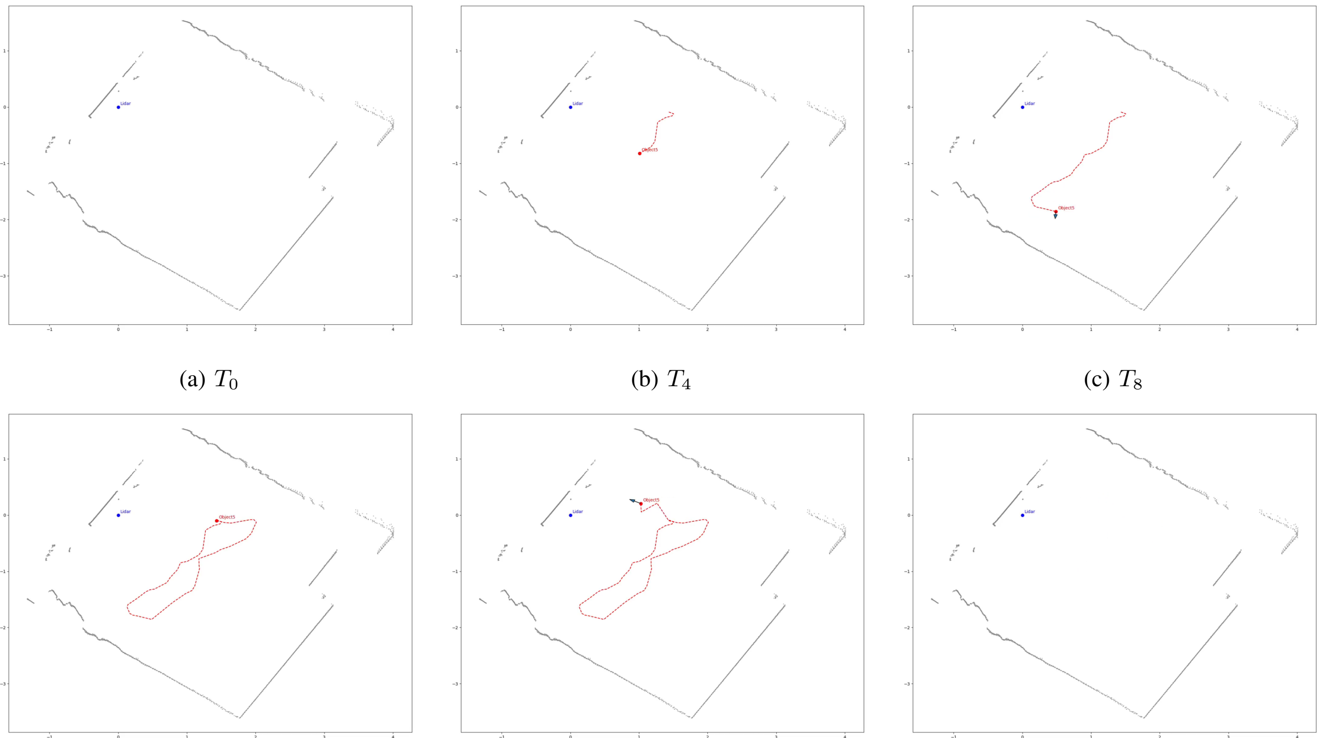
Fig. 4: Scenario #3: Two Persons Crossing Plot

scenarios before the object disappears/dies. Way-points are recorded for the evaluation. In the usual tracking mode, the way-points are not displayed by default. If desired, the plotting of way-points can be enabled in the plotter class.

1) Scenario #1: Single Person: In the first scenario, during all five runs, the person was correctly detected, tracked without interruption, and removed again after disappearing (*i.e.,* marked as "dead"). No static objects were classified as people. Figure 5 depicts the progress of the run with a running time of 40 s, in which the object alternately moved and remained stationary.

2) Scenario #2: Two or More Persons not Crossing Paths: In the second scenario, two people were successfully tracked, as illustrated in Figure 3. Both persons were recognized at the same time, and their path was tracked correctly. Even when two people/objects were close to each other (but not behind each other), the identification worked without any issues in five runs. The tracking was performed during 35 s.

3) Scenario #3: Two Persons Crossing Paths: In the third scenario, two people cross paths, assuming that they do not change their speed significantly. If a person started walking while being covered by the other person, there would be no heuristic possibility to detect this. Since according to the last confirmed information, the person was at rest. The five crossings were performed with the following estimated crossing angles: twice 180 degrees, once 120 degrees, once 90 degrees, once 60 degrees. In two out of five cases, the individuals could not be positively identified after crossing. The smaller the crossing angle, the higher the probability that the recognition fails because the persons are not distinguishable for the LiDAR for a longer time. Accordingly, the identification of the objects failed at 90 and 60-degree crossing angles. The best results can be achieved when the persons cross at a straight angle (180 degrees). The plot of a run with a duration of 8 seconds and a straight angle is depicted in Figure 4.



(a) T_0 (b) T_4 (c) T_8

(d) T_{12} (e) T_{16} (f) T_{20}

Fig. 5: Snapshot Series of a tracking run, where (a) Object not detected in T_0, (b), Object is detected and tracking started in T_4, (c) In T_8, active tracking and direction vector is visible, (d) Object stopped and no direction vector is calculated in T_{12} (e) Active tracking and direction vector visible in T_{16}, and (f) Object died and all way-points cleared in T_{20} [T measured in seconds]. Video captures of the evaluation can be found at [24].

4) Scenario #4: Single Person Behind Static Object: A flaw in the location algorithm occurred in the second run of Scenario #4, after reappearing behind the static object. The object was identified as a new object instead of the original due to a sudden change in speed or change of walking directly behind the static object. Once the model defined in [3] represents a "near constant" velocity variation, the heuristic prediction model failed considering the covariance Q defined in the white noise Gaussian process. This highlights the importance of choosing a suitable location model for the characteristics of the environment. Considering that abrupt accelerations are exceptions for an indoor tracking reproducing the behavior of an exhibition, it is possible to tolerate such outliers.

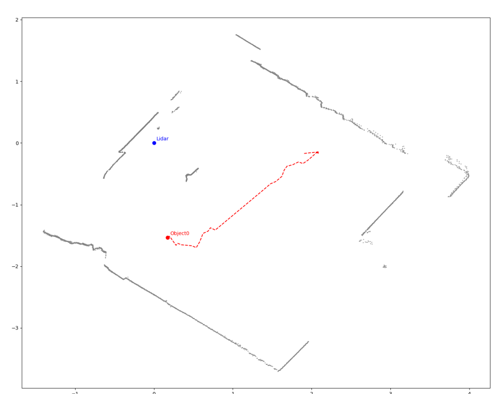


Fig. 6: Scenario #4: Single Person behind Object Plot

B. Discussion

The results clearly reflect the strengths and weaknesses of a LiDAR sensor from Section II. As expected, the positioning was accurate. However, as a natural disadvantage of tracking devices based on light reflection, it is not possible to track when objects are obscured by other objects regardless of whether they are static or in motion. Herein, it can only be evaluated with a certain probability whether the object still exists and where it is located when reappearing. How well this case works also depends significantly on the parameterization. For example, if people are expected to move close to each other, the split-distance and existing-new-threshold parameters should be decreased. This increases the probability that the persons will be detected. At the same time, it increases the risk that single persons will be detected as several.

The LiDAR interface was responsible for making the data from the LiDAR scanner available for data processing. The data processing always received the latest laser data via the Influx database and the data acquisition worked without any interruptions via the socket.

From the first scenario, it can be deduced that the segmentation of static and moving objects works reliably. The moving object was recognized as such and tracked. Further, scenario 2 shows that the system can also detect and track two or more people simultaneously. Errors occurred in scenario 3, in which LaFlector could no longer determine which of the intersecting persons was in two out of five cases. Scenario 4 further demonstrates that heuristic to predict movement worked as expected being able to associate the path with the object.

Objects were recognized after disappearance except for one case. It follows that when two crossing objects are in motion, the system appears to be less accurate than when one of the objects is static. It is possible that even better results could be obtained with more complex, finer-grained heuristic prediction. However, this would also require a higher resolution LiDAR scanner. In particular, far away objects from the scanner would otherwise take too long to be reliably detected since the density of measurement points decreases with distance.

The graphical illustrations from the results are from the plotter of the system. The log file created with each run contains the desired information according to the selected log level. It is, therefore possible to read (x, y) points that an object had during a run. With a simple code extension, the logger's data could be made available for further processing or combination with another data source. Therefore, the requirement of expandability can also be confirmed.

V. Related Work

This section covers existing tracking techniques using LiDAR scanners. To properly classify the related work, it is important to categorize how a LiDAR device can be used since classification methods depend on the deployment scenario. This classification is depicted in Figure 7 and detail in the next sections. Further, existing work regarding tracking humans using LiDAR are presented and compared with `LaFlector`.

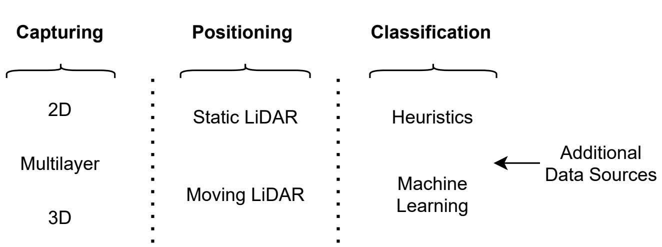


Fig. 7: LiDAR Related Work Categorization

In [10], a 2D LiDAR is used on vehicles to distinguish flat and uneven surfaces and derive the course of a road. This is achieved by evaluating the distance between the single measuring points. On roads without any curbs the system achieves at least a score of 92% in detecting the edge of the road. If curbs are present the value drops to 80%. This can be explained by smaller variations in the measured values caused by a sidewalk than in uneven ground.

[7] employs multiple 2D LiDAR sensors mounted at different heights to provide more data than a single 2D scanner but less than 3D LiDAR. This allows an object measured in a particular layer to be verified with the other layers in terms of tracking. In their evaluation with LiDARS mounted on a car, the pedestrian detection rate increased from 70.5% with one layer to 91.6% with 4 layers.

A 3D LiDAR captures multiple angles and provides significantly more data points than a 2D or a multilayer approach. Further, The points distributed over different heights allow effective surface matching. For example, if an object is completely flat over its entire height it can concluded that is not human. [37] presents an experiment in which two people walk through a room and hide behind obstacles and come out again, the false-negative rate was halved from 30 frames to 15 frames using surface matching for a total of 696 frames.

A. Positioning

While static LiDAR scanners usually focus on detecting objects, for moving LiDAR sensors the own positioning is of additional interest. This process is also known as SLAM (Simultaneous Localization and Mapping). This can for example be used for measuring rooms. Samsung also uses LiDAR sensors for this purpose in its new robot vacuum cleaners [4]. To create the map of an environment, a robot can move to different points. Its movements can be determined by the LiDAR itself and depending on the construction also by wheel positions. To correct errors, positions are also approached multiple times to correct errors accumulated during the movement. This is also called loop detection (approaching a location twice) and loop closure (error correction) [11].

While LiDAR is used mounted directly on vehicles, as shown previously, they can also be used statically at road intersections. Based on the size of the set of points, the distance from the LiDAR, and the direction taken, it is possible to decide whether it is a pedestrian or a vehicle. This information can be used to optimize the flow of traffic at intersections. In a potentially further step in connected driving, approaching vehicles could be warned of pedestrians or even force an emergency brake [44].

B. Object Classification Strategies

Further, most approaches are based on the ROS (Robot Operating System). The ROS is an open-source framework for robot software development [33]. For example, Leg Detector (LD) is the most popular ROS package for people detecting with a LiDAR sensor. Recognition at LD is based on a machine learning classifier [34].

PeTra (People Tracking) [8] is a system based on a Convolutional Neural Network (CNN). PeTra was developed for use on mobile robots. This means that the scanner's working height was assumed to be 30 to 50 cm above the ground. Accordingly, the system was designed and trained to detect human legs. In comparison with the LD Package PeTra was able to achieve higher accuracy.

[26] presents an approach for tracking people using heuristics and a LiDAR sensor. The team was able to identify an object that was obscured by another object through the prediction from the reappearance. But it was also noted that a changing velocity of a hidden object leads to a mismatch between the hypothesis and the reallocation. In any case, this leads to the result that the object can no longer be identified with confidence.

C. Heuristic Approach

Heuristics in the sense of computing is defined as *"proceeding to a solution by trial and error or by rules that are only loosely defined"* in the Oxford Dictionary of English [28]. In terms of tracking, this means expecting a moving object's appearance at a certain time and place. As a loose rule, the object's current motion can give an expected value for a future point in time.

D. Machine Learning Approach

While in the heuristic approach, rules are defined manually, in machine learning, these rules are automatically constructed based on a training data set. The classification of objects belongs to the supervised learning methods. This means that, for example, to recognize a leg, you need a training set that contains legs and non-legs representations and the information whether it is a leg or not. From this information, a model can be derived using a learning method. This model can then be used to make predictions [17]. Neural networks are suitable as a learning method for the classification of image and sound files [1].

The same goals can be addressed by using both moving or static LiDAR sensors. The detection of persons can take place at intersections or at the vehicle itself. For the purpose of this paper (*i.e.,* the tracking of people in a room) it is more appropriate to keep the LiDAR static. Due to the static positioning, the height of the LiDAR can be freely chosen. LaFlector is designed to be operating at heights of 1.20 m to 1.50 m. Thus, the system does not recognize legs but only upper bodies, which does not affect the accuracy.

LaFlector uses a 2D LiDAR, allowing a deployable system to be constructed at lower cost than with 3D sensors. Since no training data set is available for a neural network supporting the given LiDAR, the heuristic approach was chosen. In the classification, as in related works, the width of the object is used. Techniques such as surface matching cannot be used since the measurement consists only of a single layer.

Efforts have been done on LiDAR tracking, but most focus on moving LiDAR sensors whereas LaFlector follows the static placement approach. The system is designed to be extensible with a second LiDAR. In combination of with a low-cost 2D LiDAR sensor and detection at body height instead of leg, this paper provides additional contributions over existing related work. In contrast to the recognition of legs, where it must always be taken into consideration that legs could be behind each other, the recognition at body height offers chances for higher recognition rates.

VI. Summary and Future Work

This paper presented LaFlector, a LiDAR-based indoor tracking system that introduces heuristics capable of detecting, classifying, and tracking several objects simultaneously, recorded and dynamically displayed in a 2D coordinate system. A key takeaway is that LiDAR scanners were shown in the experiments to be sensitive to environmental influences (*e.g.,* light incidence) and how many data points it takes to get a reliable result. In the initial phase, this also led to many trial and error attempts. Many parameters depend on the LiDAR device itself and its specifications. The experiments performed demonstrated the effectiveness of LaFlector's tracking algorithm in different indoor scenarios, in which situations were explored, where people made abrupt movements and remained behind other objects, preventing the reflection of light. In this sense, the use of heuristics allowed LaFlector to estimate the positioning of the previous object.

As next steps the addition of further sensors (besides LiDARs, such as based on Wireless and Bluetooth) is planned for to allow for the extension of the measurement range for improvements of the localization accuracy and to counterattack errors. While algorithms used for localization are theoretically capable of working with multiple sources, approaches to extend ranges will have to include strategies for the selection of the most useful measurements during the merging stage.

Acknowledgments

This work was supported by Innosuisse, the Swiss Agency for Innovation Demands under Grant No. 42193.1 IP-ICT. Also, the authors would also like to thank Simon Tuck, Livealytics AG, who supported the research and development during those experiments in Corona times of limited public accessibility and with many discussions on different setups and interpretations.

References

[1] S. Albawi and T. Mohammed, "Understanding of a Convolutional Neural Network ," *The International Conference on Engineering and Technology*, 2017.

[2] D. Baltieri, R. Vezzani, R. Cucchiara, Utasi, C. Benedek, and T. Szirányi, "Multi-View People Surveillance Using 3D Information," in *IEEE International Conference on Computer Vision Workshops (ICCV 2011 Workshops)*, Barcelona, Spain, November 2011, pp. 1817–1824.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

[4] CNET, "Samsung's new robot vacuum uses lidar and empties its own bin like a fancy Roomba," 2021. [Online]. Available: https://cnet.co/3dwp9ze

[5] B. Consulting, "MAC-Adresse und der Datenschutz – Fingerabdruck des Computers?" 2018. [Online]. Available: https://brands-consulting.eu/mac-adresse-und-der-datenschutz-fingerabdruck-des-computers

[6] A. Filgueira, H. Gonzalez-Jorge, S. Laguela, L. Diaz-Vilarino, and P.Arias, "Quantifying the Influence of Rain in LiDAR Performance," *Measurement*, Vol. 95, pp. 143–148, 2017.

[7] S. Gidel, P. Checcin, C. Blanc, and T. Chateau, "Pedestrian Detection and Tracking in Urban Environment using a Multilayer Laserscanner," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 3, pp. 579–588, 2010.

[8] A. Guerrero-Higueras, C. Alvarez-Aparicio, M. Olivera, F. Rodriguez-Lera, C. Fernandez-Llamas, F. Rico, and V. Matellan, "Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments," *frontiers in Neurorobotics*, Vol. 12, No. 85, 2019.

[9] B. Habtemariam, R. Tharmarasa, T. Thayaparan, M. Mallick, and T. Kirubarajan, "A Multiple-Detection Joint Probabilistic Data Association Filter," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 7, No. 3, pp. 461–471, 2013.

[10] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Road Boundary Detection and Tracking for Structured and Unstructured Roads using a 2D LiDAR sensor," *International Journal of Automotive Technology*, Vol. 15, No. 4, pp. 611–623, 2014.

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-Time Loop Closure in 2D LIDAR SLAM," *IEEE International Conference on Robotics and Automation*, 2016.

[12] A. Inc., "Apple's Report on Wi-Fi Privacy," 2021. [Online]. Available: https://support.apple.com/guide/security/wi-fi-privacy-secb9cb3140c/web

[13] InfluxData, "Getting Started with Python and InfluxDB," 2018. [Online]. Available: https://www.influxdata.com/blog/getting-started-python-influxdb/

[14] International Electrotechnical Commission, "IEC60825-1 Standard - Safety of Laser Products," August 2001, https://bit.ly/32sEISb, last visit April 20th, 2021.

[15] Kaggle, "Data Science Survey - 2018," 2018. [Online]. Available: https://www.kaggle.com/sudhirnl7/data-science-survey-2018

[16] S. T. Kouyoumdjieva, P. Danielis, and G. Karlsson, "Survey of Non-Image-Based Approaches for Counting People," *IEEE Communications Surveys Tutorials*, Vol. 22, No. 2, pp. 1305–1336, 2020.

[17] A. Krause, "Introduction to Machine Learning," 2020. [Online]. Available: https://las.inf.ethz.ch/courses/introml-s20/slides/introml-01-introduction-annotated.pdf

[18] M. Kutila, P. Pyykonen, H. Holzhuter, M. Colomb, and P. Duthon, "Automotive LiDAR Performance Verification in Fog and Rain," *IEEE Intelligent Transportation Systems Conferenc*, 2018.

[19] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 10, pp. 1683–1698, 2008.

[20] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and M. Flood, "LiDAR Remote Sensing of Forest Structure," *Progress in physical geography*, Vol. 27, No. 1, pp. 88–106, 2003.

[21] G. LLC, "Privacy: MAC Randomization," 2020. [Online]. Available: https://source.android.com/devices/tech/connect/wifi-mac-randomization

[22] P. McManamon, "History of LiDAR," in *LiDAR Technologies and Systems*, 2019, pp. 29–34.

[23] L. Mueller, "LaFlector's Source Code," 2021, https://gitlab.ifi.uzh.ch/rodrigues/laflector, last visit May 21st, 2021.

[24] ——, "Single Person Evaluation Videos of LaFlector," 2021, https://owncloud.csg.uzh.ch/index.php/s/LrqgR3sAa93ozdL, last visit April 20th, 2021.

[25] V. Navalpakkam and E. Churchill, "Mouse Tracking: Measuring and Predicting Users'Experience of Web-based Content," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2963 – 2972, 2012.

[26] H. Nemati, S. Shahbandi, and B. Åstrand, "Human Tracking in Occlusion based on Reappearance Event Estimation," *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotic*, Vol. 2, pp. 505–512, 2016.

[27] F. O. of Public Health FOPH, "Faktenblatt Laserpointer," 2018. [Online]. Available: https://www.bag.admin.ch/dam/bag/de/dokumente/str/nis/laser/faktenblatt_laserpointer.pdf

[28] O. U. Press, "heuristic," 2015. [Online]. Available: https://www.oxfordreference.com/view/10.1093/acref/9780199571123.001.0001/m_en_gb0375970

[29] Privacy Company, "What does the GDPR say About WiFi Tracking?" May 2019, https://www.privacycompany.eu/blogpost-en/what-does-the-gdpr-say-about-wifi-tracking, last visit May 1st, 2021.

[30] R. Ribeiro, B. Rodrigues, C. Killer, L. Baumann, M. Franco, E. J. Scheid, and B. Stiller, "ASIMOV: A Fully Passive WiFi Device Tracking," in *IFIP Networking 2021*. Espoo, Finland: IFIP, jun 2021, pp. 1–3.

[31] B. Rodrigues, I. Cepilov, M. Franco, E. J. Scheid, C. Killer, J. von der Assen, and B. Stiller, "CCount: Correlating RFID and Camera Data for High Precision Indoor Tracking," 2022.

[32] B. Rodrigues, C. Halter, M. Franco, E. J. Scheid, C. Killer, and B. Stiller, "BluePIL: a Bluetooth-based Passive Indoor Localization Method," in *IFIP/IEEE International Symposium on Integrated Network Management (IM 2021)*. Bordeaux, France: IEEE, may 2021, pp. 1–9. [Online]. Available: http://dl.ifip.org/db/conf/im/im2021/210969.pdf

[33] ROS.org, "ROS Introduction," 2018. [Online]. Available: http://wiki.ros.org/ROS/Introduction

[34] ——, "leg_detector," 2021. [Online]. Available: https://wiki.ros.org/leg_detector

[35] C. Schefels, S. Eschenberg, and C. Schöneberger, "Behavioral Analysis of Registered Web Site Visitors with Help of Mouse Tracking," in *IEEE 14th International Conference on Commerce and Enterprise Computing (CEC 2012*, Hangzhou, China, September 2012, pp. 33–40.

[36] J. Scheuner, G. Mazlami, D. Schoni, S. Stephan, A. De Carli, T. Bocek, and B. Stiller, "Probr - A Generic and Passive WiFi Tracking System," *IEEE 41st Conference on Local Computer Networks*, Vol. 88, pp. 71–78, 2016.

[37] J. Shackleton, B. VanVoorst, and J. Hesch, "Tracking People with a 360-Degree Lidar," *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010.

[38] Shanghai Slamtec. Co., Ltd, "SLAMTEC Mapper M1M1 Datasheet," 2019, https://bit.ly/3tBsqCX, last visit April 20th, 2021.

[39] ——, "SLAMTEC Mapper M2M1 Datasheet," 2019, https://bit.ly/3n3Rn7M, last visit April 20th, 2021.

[40] H. Wang, B. Wang, B. Lui, X. Meng, and G. Yang, "Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle," *Robotics and Autonomous Systems*, Vol. 88, pp. 71–78, 2017.

[41] R. W. Wolcott and R. M. Eustice, "Visual Localization Within LIDAR Maps for Automated Urban Driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, Chicago, IL, USA, September 2014, pp. 176–183.

[42] J. Wozencraft, "Complete Coastal Mapping with Airborne LiDAR," in *MTS/IEEE OCEANS 2002*, Vol. 2, Biloxi, MI, USA, October 2002, pp. 1194–1198 vol.2.

[43] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," Vol. 38, No. 4, p. 13–58, December 2006.

[44] J. Zhao, H. Xu, H. Liu, J. Wu, and Y. Zheng, "Detection and tracking o fpedestrians and vehicles using roadside LiDAR sensors," *Transportation Research: Part C*, Vol. 100, pp. 68–87, 2019.