# Detection, classification and tracking of persons with 3D LiDAR sensor in indoors environments*

Juan Gómez[1], Olivier Aycard[2]

***Abstract*—Person detection, tracking and classification are important aspects in applications such as social robotics, surveillance, human-robot collaboration and autonomous driving. In our solution, we present a modular approach that combines multiple principles: a robust implementation for object segmentation, a simple classifier with local geometric descriptors and a tracking solution with Global Nearest Neighbors data association and target's motion prediction. The main merit of this work relies in achieving a real time solution in a low-performance machine by reducing the amount of points to be processed by obtaining and predicting regions of interest via movement detection and motion prediction without any previous knowledge of the environment. Furthermore, even though our geometric classifier is simple, our prototype is able to successfully detect and track persons consistently even in challenging cases due to limitations on the sensor field of view or extreme pose changes such as crouching, jumping, and stretching. This is done thanks to a continuous interaction between the classification and tracking modules that allows the latter to correct misclassifications by predicting the target's location. Lastly, the proposed solution is tested and evaluated in multiple real 3D Light Detection and Ranging (LiDAR) sensor recordings taken in our office and shows great potential, particularly a high confidence in positive classifications.**

***Index Terms*—3D point cloud, person detection, tracking, classification, real-time**

## I. INTRODUCTION

In the field of robot perception, Detection And Tracking of Moving Objects (DATMO) is one of the main problems in almost every robotics application. Now more than ever, several autonomous robotic applications require a robust perception module to understand the environment the robot is in. Applications like autonomous or assisted driving, environment understanding for social robots and even human behavior understanding for commercial or scientific purposes are among them.

A lot of previous work has been done to tackle this particular problem in perception, most of them with depth or 3D sensors such as RGBD and stereo cameras, radar and even 2D Light Detection And Ranging (LiDAR) sensors as they became very popular due to their affordability. Even though the latter are affordable, precise and their data is not computationally expensive to process, the limitations of only perceiving one plane of the environment makes it difficult to detect people with a high confidence. On the other hand, RGBD and stereo cameras can often be slow to extract depth information and the results are often not as precise as with a LiDAR sensor.

Meanwhile, a 3D LiDAR offers a high resolution of points at a high speed with great precision, resulting in a very descriptive geometry of the environment. Objects in a 3D LiDAR reading can be described geometrically as they appear in the real world, with accurate representations for length, width and height up to a certain precision. Besides this, they are not affected by extreme lighting conditions as they depend on time of flight calculation.

In this work, we present a real time solution for multiple person detection, classification and tracking. The former is achieved by using techniques normally used in classic computer vision such as movement detection, regions of interest (ROIs) and tracking, in addition to classic techniques used in 3D point cloud non real time applications such as voxelization and segmentation. The two of which combined can result in a real time application.

The algorithm can be separated in the following stages: Voxelization, Segmentation, Movement Detection, Classification and Tracking. The focus of our work is to always process the least amount of points possible whilst never leaving a possible region of interest unprocessed.

Each of these techniques result in a real time solution for our problem without having any prior knowledge on the environment. As simple or as computationally expensive as some of the techniques can be on their own, they complement each other in order to accomplish our real time goal with positive results.

The contributions of our solution are the following:

- Modular structure consisting of multiple stages that work together to achieve the best performance and results. The modular nature makes it easier to enhance the performance of the solution by improving any of the modules presented.
- Although normally the classification and tracking parts of a solution are handled independently, our solution relies on a continuous interaction between the two that ultimately allows for a robust and consistent performance when dealing with extreme pose changes such as crouching, jumping and stretching, or even some of the sensor's limitations.
- A real-time implementation on a low-performance machine for detecting, classifying and tracking persons in an indoors environment in 3D point clouds. This is achieved

Identify applicable funding agency here. If none, delete this.

*This work was not supported by any organization, and its currently being contested for publication at ICTAI 2021

[1]Juan Gómez (juan.gomez@grenoble-inp.org) and Olivier Aycard[2] (olivier.aycard@imag.fr) are with Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

by dynamically extracting all the regions of interest that could contain a target. The extraction is done by both movement detection and motion prediction.
- The prototype is validated on a real 3D LiDar data set taken with our sensor in our work office. The results presented are overall satisfactory and show a specially high confidence for positive classifications.

This paper is divided as follows: In section II we present the related works in this field, section III gives context about the sensor's specifications and some of the most challenging cases of the problem, section IV explains the architecture and implementation of the solution, section V presents the validation and results and finally section VI presents the conclusion and discussion about future works.

## II. RELATED WORKS

Different types of range sensors have been used in previous work for DATMO, since depth information is crucial for systems that rely on accuracy and precision. 2D LiDAR sensors have been used mainly for autonomous driving applications since they are fast and precise, and the objects of interest (often other cars and motorbikes) are big enough to be accurately detected by the sensor despite its limitations [1]. Similar to these approaches, 3D LiDARs have also been used for these applications [2]. All of these works rely on a local vehicle map (occupancy grid) and create moving object hypotheses from first movement detection. In order to handle the high amount of data obtained with 3D LiDars and achieve a real-time implementation, prior 3D maps have been used for background subtraction to reduce the computation time [3], but this requires previous knowledge of the environment. In contrast to our approach, we rely on a background subtraction algorithm without any previous knowledge of the environment. And our solution does not rely on a first movement to detect a person. Overall, for person detection (even in close range) the data provided by a 2D LiDAR sensor is far too limiting to accomplish the ultimate task of analyzing and understanding a person's behavior.

Stereo vision and RGB-D cameras are also widely studied in the field of detection and tracking of objects, there are many works that aim to take advantage of these technologies. In the case of RGB-D cameras, they present an enticing offer since they provide both RGB information and depth data. RGB information could be useful in general vision applications like detection by skin detector and also for specific applications such as person re-identification. Some examples of the possible applications can be segmentation and classification of legs [4] and histograms for height difference and color [5]. Moreover, multiple detectors could be implemented depending on the object's distance [6]. Also, range data such as a target height can be used to introduce redundancy in order to improve accuracy [7].

Most of the previous work for detection and tracking of mobile objects use a moving platform for the sensor, so they have to deal first with Simultaneous Localization And Mapping (SLAM) and then with the DATMO problem. Despite this, some work has also been done for static sensors as it is in our case. The goal of our prototype is to serve as a proof of concept and it could be expanded to work with a mobile platform for the sensor by adding a SLAM stage or for any application with a stationary sensor. In the literature, works with stationary sensors have been done for applications such as person behavior understanding for commercial purposes [8].

For DATMO problems, occlusion of the targets is a problem with every sensor and it becomes more apparent when the sensor is stationary. Previous work to explicitly handle occlusions in these cases have been done by tracing the rays from positions near the target back to the sensor and see if there is an obstacle in the way [9].

Many previous works rely heavily on movement detection and clustering to track mobile objects, but imperfections of both techniques could result in a less precise solution. In our application we focus only on detecting and tracking people, so any other moving object can be discarded. Therefore, classification of the moving objects is a crucial step in our solution since we only care about the moving objects that are classified as a person. Classification of point clouds has been done with deep learning supervised models for person detection [10] which could replace the movement detection module since it segments directly the points in a scan that belong to an object of interest, but the process of classification of the whole frame can be expensive. Other solutions use trained R-CNNs to go directly from the raw input to the object being detected and tracked [11], but they were done only for tracking cars and do not fit in a modular solution.

Lastly, the approach for classification that best fits our needs of a fast, efficient and modular solution would be to segment the point cloud into a list of objects and then classify each object according to their geometric features [12]. Going from a total number of points that ranges from 32000 to 64000 to a list of a few objects reduces computational cost significantly. Furthermore, after classification we can focus specifically on those objects classified as a person and discard the others as background. In summary, we deal with a binary classification problem. Our work is therefore based on previous work done in [12], which presents a robust way to segment the points into individual objects and then classify them according to their geometric dimensions. This work was done to segment all the points of a single scan of a 3D LiDar sensor and classify them into multiple classes of objects and therefore is expensive in computation. Our scientific contribution relies on applying this method to take advantage of the robustness of the segmentation and the simplicity of the classification, while doing a temporal integration and tracking to process entire recordings of point clouds in real time. Moreover, we present an innovative interaction between the classification and tracking modules that is crucial to overcome the shortcomings of the simple classification process. Normally classification and tracking work independently, but previous works with 2D LiDars have also presented a similar interaction between the two [13].

## III. Implementation and Challenges

### *A. Sensor specs and positioning*

We used the Ouster OS1-32 3D LiDAR sensor. The sensor has a vertical resolution of 32 layers and the horizontal field of view is customizable up to a full 360-degree turn. On the other hand, the horizontal resolution (number of points per layer) is also customizable with three different options: 512, 1024 and 2048. Furthermore, the base frequency of operation for the three resolutions is 10hz. The minimum and maximum range of the sensor is 0.8m and 150m respectively. The precision of the sensor is of 1.1cm in the 1 - 20m range. A picture of the sensor can be seen in Fig. 1a

Even though the maximum sensor range is 150 meters, the resolution of the sensor makes it so that at greater distances it is difficult to differentiate a moving object from the background. Therefore, if we want to take advantage of the descriptive geometry that the sensor provides we need to have the most amount of points possible on our objects of interest, and that is why our solution is conceived for small indoors ranges of up to 8m in any direction.

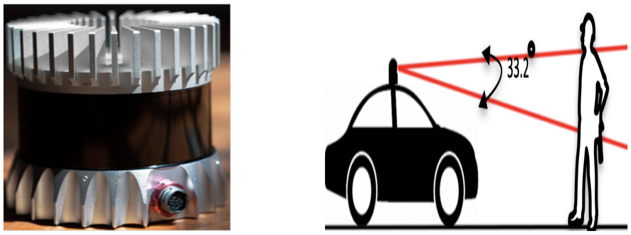
The recordings used for evaluation were done in two different offices, one big office of up to 12 meters in length and 5 meters in width, and a small office space of 4 meters in length and 5 meters in width. In every case, the sensor was positioned at a height of 1.2m, which is set to mimic a top-mounted sensor on a mobile robot platform.

### *B. Challenges and limitations*

*1) Restricted vertical field of view:* Despite our sensor having a complete 360-degree horizontal field of view, it is restricted with only a 33.2 degrees of vertical field of view. This means that even though objects are detected from a minimum range of 0.8m, an object that is close to the sensor could appear chopped in the resulting scan. This phenomenon depends on the object's height and the distance from the sensor. Taking into account that our application is in small indoors environments, we should expect people to walk close to the sensor, enough to appear chopped in the final scan. A person appearing chopped in the scan presents a challenge to the classifier since it relies on object dimensions and shape. An example of this situation can be seen in Fig. 1b.

*2) Extreme pose changes and occlusions:* Extreme pose changes are a challenge for any detector and tracker of people that involve a classification module. The wide variety of poses a person can assume makes it difficult for a classifier to be generalized enough to detect the person in normal poses and every possible combination of poses it might assume. Poses like crouching, stretching, jumping, pick and place objects are among them. Our solution aims to deal with this issue by stating an interaction between the classification and tracking modules to overcome their shortcomings.

Furthermore, another widely encountered problem in tracking are target occlusions [9]. Having a sensor at a specific position generates the problem of moving objects being occluded by static objects or each other. Our solution does not deal explicitly with occlusions but it is evaluated to see how it behaves in their presence.



(a) (b)

Fig. 1: In 1a a picture of the sensor, in 1b an example of the limitation of the restricted vertical field of view

## IV. Architecture and Implementation

In Fig. 2 the flow chart of our solution divided in each of its modules with the corresponding inputs and outputs is shown. The first module receives every point from the raw scan and it performs movement detection to find ROIs and therefore saving processing time. The second and third module receive the ROIs and do a process of voxelization and segmentation, their output is a list of segmented objects that were present in the ROIs. The fourth module classifies each one of the objects as either person or background. The fifth and final tracking module handles the creation, update and elimination of the tracks. This last module has two outputs: the final point cloud classified, and the predicted ROIs for each of the tracked persons, that are going to be used as feedback in the next time step.

This section gives a brief overview of the implementation of the modules 1, 2 and 3 and concentrates on the modules 4 and 5 because our scientific contribution relies on them.

### *A. Movement detection*

As already mentioned, in order to achieve a real time implementation we need to process the least amount of points possible at each scan. Given that in an office most of the environment is static, processing these points would be redundant and time consuming. This is why we rely on extracting ROIs that constitute a small part of the total points and that should contain the moving objects. One way to extract these ROIs is by doing movement detection. For a static sensor the most common and efficient way to do this is doing background subtraction. In our solution we do not rely on any previous knowledge of the environment to do this.

Therefore, the first step is to do an online creation and calibration of the background. This is done once at the start of the algorithm, we process every point in the scan through the entire pipeline in Fig. 2. In this way, if there is a person in the environment at the moment of the creation we can eliminate those points from the final model. Therefore, this part is the most expensive in computation as it is the worst case scenario (processing every point in the scan).

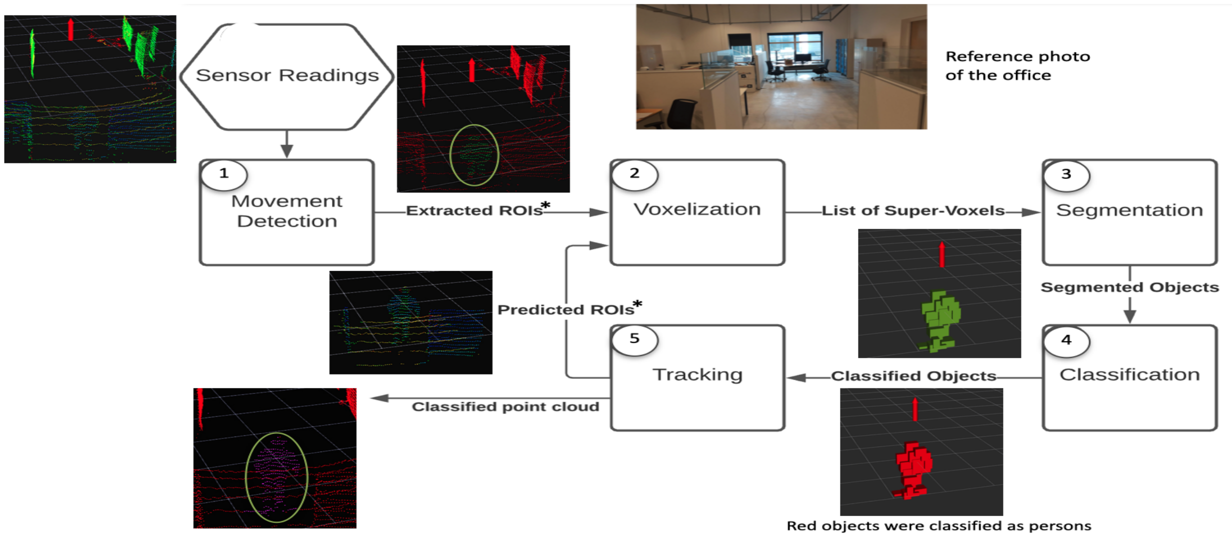


Fig. 2: Pipeline of our solution. Here we show the different modules implemented for our prototype. The red arrow represents the position of the sensor, red points represent static points, green points represent dynamic points and pink points represent points classified as part of a person.

After the background model creation we should have a list of only the static points of the environment and they can be compared to subsequent scans as it can be seen in Alg. 1.

**Algorithm 1:** Background subtraction

```
inputs : Voxel list from background model, point cloud
         from new scan
output: Moving voxel list $MV$ (ROIs)
for voxel in background model do
    Get voxel center $v_c$;
    Find neighbor points of $v_c$ within a maximum
      spherical distance of $2r$ in the new scan and save
      them;
end
Find points that did not fall in the vicinity of any of
  the voxels in the background model;
Apply the Voxelization method on them and save them
  on moving voxel list $MV$;
for each voxel $V$ in $MV$ do
    if number of points in $V$ ¡ minimum-density then
        erase $V$ from $MV$
end
```

### B. Voxelization and segmentation

In order to obtain a robust segmentation of the objects in a scan (or in the ROIs), we implement the solution presented by Trassoudaine et al. [12]. In this approach, voxelization is done by a nearest neighbor search in a radius around a center point (r-nn approach). The maximum voxel size has an effect over the final result of the segmentation and in the performance, and for our application is chosen to be 0.2m which strikes a good balance between computing time and resolution for short to medium range detection.

Segmentation is done as a link chaining method, where the links are the voxels. The links are chained together if they fulfil conditions regarding distance from each other and regarding other parameters provided by the sensor.

### C. Classification

After the segmentation process, we have a list of segmented objects (formed by voxels) that were inside the processed ROIs. The goal of the classifier is to solve for a binary classification problem, therefore it should assign a label to each object of either person or background. In our solution, we want to achieve a high confidence in a positive classification, since we want to focus only on moving objects that are classified as persons. As it is shown in Fig. 3, the classifier is made out of 3 weak classifiers. Each weak classifier has their own criteria for classification and they each cast a vote for the objects. In this way, the three weak classifiers work together to complement each other's limitations, and at the end it results in a highly confident classification. Finally, votes for each object are counted and we obtain our final classification. We present and detail every weak classifier in this section.

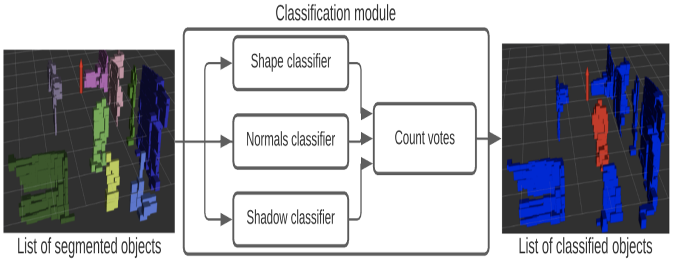


Fig. 3: Flow chart of the classification module. At the output the objects classified as persons are shown in red and blue otherwise.

*1) Shape classifier:* As already mentioned, one of the benefits of a 3D LiDAR with high resolution is that we can extract precise and multiple geometric features from the objects. The shapes classifier takes advantage of the descriptive geometry provided by the sensor. It is a simple classifier that compares the dimensions and shape of each object to one of the ”strict” geometric models of a person that we already have. These models represent a variety of attributes an object should have to be considered a person. These attributes include width, height, length, and the proportions between them. For example, if an object is taller than 2.3m we know it can not be a person, or if an object has almost the same width as height we also know it is not a person. With a variety of these simple thresholds this classifier can recognize a person when they are in their most basic poses (standing up or walking), this is why we say that the shape classifier compares to ”strict” models. The idea is to only start the tracking when we have a high confidence that the object is a person. Therefore, our classifier can not deal directly with the majority of poses a person can undergo other than walking and standing up, the integration between the classification and tracking can overcome this problem as it is explained in the tracking section.

*2) Normals classifier:* Even though the shape classifier is made to recognize a person under the most basic circumstances, its simplicity might also provide false negatives or false positives results. In order to fix this, we use the normal vectors classifier. For every object we calculate the normal vectors of each of the voxels that form them. Estimating the surface normals in a point cloud is a problem of analyzing the eigenvectors and eigenvalues or PCA (Principal Component Analysis) [12], and it is done with PCL (Point Cloud Library). Once we have the normals, we calculate the average contribution of each of the components of the vectors. Therefore, if the object has most of its normals parallel to the ground, it is most likely a wall or an office division. In this way, we can eliminate most of the false positives that result from a portion of wall or division that has dimensions similar to that of a person.

*3) Shade classifier:* Lastly, this classifier was made to correct possible miss classifications that result from an object of the background being partially occluded by a moving object. When this happens, the shape of the object could be altered enough to be considered a person, and the normal vectors calculation could also be affected because of the smaller number of points and therefore resulting in a false positive classification. This is what we called the shade effect. In order to avoid this, we created a weak classifier that checks if an object is being affected by this effect, by tracing the rays close to the object back to the sensor, if the rays hit another object in the way then we now are a product of the shade effect. The situation is explained graphically in Fig. 4. This process is similar to the one presented in [9], where it was done to detect partial and full occlusions of the tracked object.

*4) Vote system:* After the votes have been casted by each weak classifier, the final step is to count the votes for each class in each object. At the end, an object is classified as a person if the votes for this class are higher than the votes for the background class.

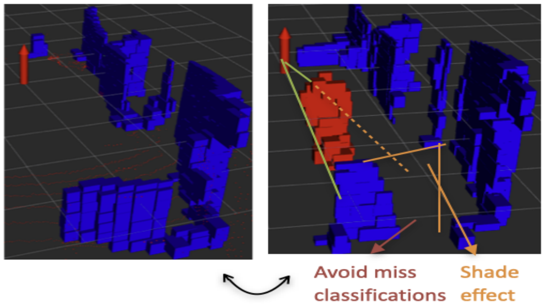


Fig. 4: Graphical example of the shade effect and the purpose of the shade classifier.

## *D. Tracking*

In Fig. 5 the flow chart for the tracking module can be seen. This module has the goal of creating, eliminating and updating the tracks. Moreover, it also handles the motion prediction stage where a predicted ROI is created for each tracked person. Finally, it can also correct some of the possible mistakes in the classification module. In this section we present every stage of the tracking module: track creation and elimination, motion prediction and the two possible cases for updating the tracks we can encounter.

*1) Track creation and elimination:* Our classifier is made to have a high confidence in positive classifications, so we can be confident on when to start a track. Despite this, it is still possible to get some false positives classifications, so we introduce more redundancy by fixing another condition to begin the track: a track is only created when the same object is classified as a person 3 scans in a row.On the other hand, a track is eliminated if the tracked person is not found after 20 consecutive scans (at 10hz it would be 2s).

*2) Motion prediction:* After a track is created, the tracking module produces two outputs: the final classified point cloud and the ROI obtained from the motion prediction as we can see in Fig 2. The motion prediction method is the second way in which we extract ROIs, other than the movement detection module. Keeping a track of the last object's positions, we can compute their previous velocity. Assuming a constant velocity we predict the next position of the target, and we create a gating area of 0.5 meters around the predicted position (in every direction). The radius of the gating area corresponds to

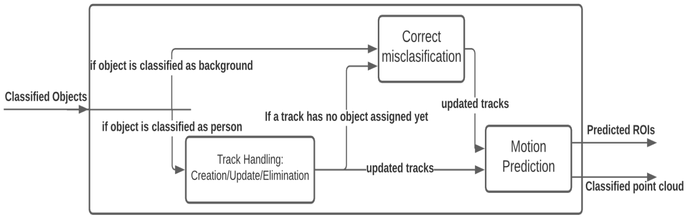

Correct misclasification
if object is classified as background
Classified Objects
If a track has no object assigned yet
updated tracks
if object is classified as person
Track Handling: Creation/Update/Elimination
updated tracks
Motion Prediction
Predicted ROIs
Classified point cloud

Fig. 5: Flow chart for the tracking module

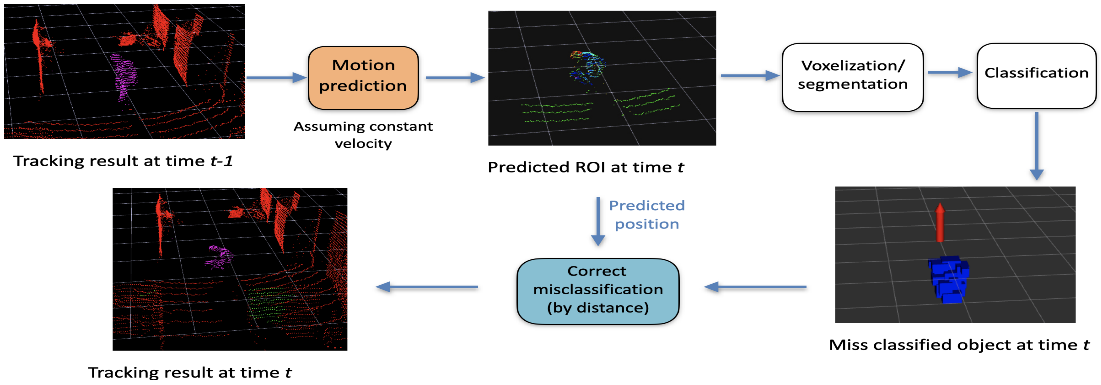

Tracking result at time *t-1*
Motion prediction
Assuming constant velocity
Predicted ROI at time *t*
Voxelization/ segmentation
Classification
Miss classified object at time *t*
Predicted position
Correct misclassification (by distance)
Tracking result at time *t*

Fig. 6: Example of the case 2: correcting a misclassification

the maximum distance a person can travel at a maximum speed in the time between frames. This means that the target should be somewhere in this ROI. The ROI becomes one of the inputs to the pipeline at the next time step from the voxelization module as we can see in Fig. 2. The idea is to have a group of ROIs that constitute a smaller part of the entire scan, and at the same time contain every dynamic object.

*3) Case 1, the track object is correctly classified as a person (track update):* This case corresponds to 80% of the cases. At every time step, every ROI goes through the voxelization, segmentation and classification process. The result is a list of classified objects that were in any of the ROIs. We take this list of objects classified as persons and assign them to their corresponding tracks by doing a method similar to GNN (global nearest neighbor) for data association. Basically, the person closest to the last recorder position of the track is assigned to it.

*4) Case 2, the tracked object is miss classified as background (correct misclassification):* This case corresponds only to 20% of the cases in a normal recording. It happens when a track is not assigned a person at any given time step because the person might have changed poses from frame to frame into one that the classifier was not able to recognize as a person. Despite it not being classified as a person, we know that the person should be close to the predicted position, somewhere in the predicted ROI. In this case, the object classified as background that is closest to the predicted position is a candidate for assigning it to the track. Before actually assigning it, the object has to fulfill some minimum conditions to be considered a person. An example of this case can be seen in Fig. 6. In this way we are able to overcome the limitations of the classifier, and we are able to detect and track persons consistently even in the challenging situations presented in the challenges section. If a track is still left with no object assigned after this, at the next time step we increment the gating area around the predicted position in which we search the object, in order to account for a possible displacement.

## V. VALIDATION AND RESULTS

This section is aimed to evaluate and discuss the results of our solution. To present the results, we start with the most simple cases up to the most complicated ones in two different environments as explained in the implementation section and with different persons. Videos of the results of all the recordings presented can be found in the url: https://lig-membres.imag.fr/aycard/html//Projects/JuanGomez/JuanGomez.html.
It is worth mentioning as a disclaimer, that due to time restrictions the results are presented and analyzed in a mostly qualitative way. In spite of this, quantitative results are presented using a simplified way to compare the ground truth to the results (whether the person is detected or not) in order to introduce more credibility to our evaluation. In this section we present the results and analysis of two different experiments.

### A. Experiment I

This experiment takes place in an office with a length of 10m and a width of 6m in the area captured by the sensor. It is a basic experiment that is aimed to validate the prototype from the most basic case to the most challenging. We present 4 different recordings: a baseline recording of a single person walking around the room normally, another single person recording focused on extreme pose changes like crouching, stretching, jumping and doing pick and place actions. Another single person recording where a big obstacle was placed in the middle of the office to simulate a large amount of occlusions, and a multiple person recording of two people walking around in the office. The results can be seen in Table I.
As expected, the results for the *Baseline* video are the best, since it is the simplest case. The person is correctly tracked at all times, with only a few false negatives. Some false positives can also be seen, and they are the product of miss classifications of segmented objects in the ROIs. Although they

TABLE I: Results for different situations in experiment I.

| Video | Precision | Recall | F1 Score | Freq |
|---|---|---|---|---|
| *Baseline* | 95.37 | 95.12 | 95.25 | 9.01 |
| *Poses* | 94.88 | 96.49 | 95.68 | 8.61 |
| *Occlusions* | 90.18 | 72.24 | 80.22 | 7.84 |
| *TwoPersons*: T1 | 93.65 | 95.16 | 94.40 | 6.80 |
| *TwoPersons*: T2 | 93.91 | 78.72 | 85.65 | 6.80 |

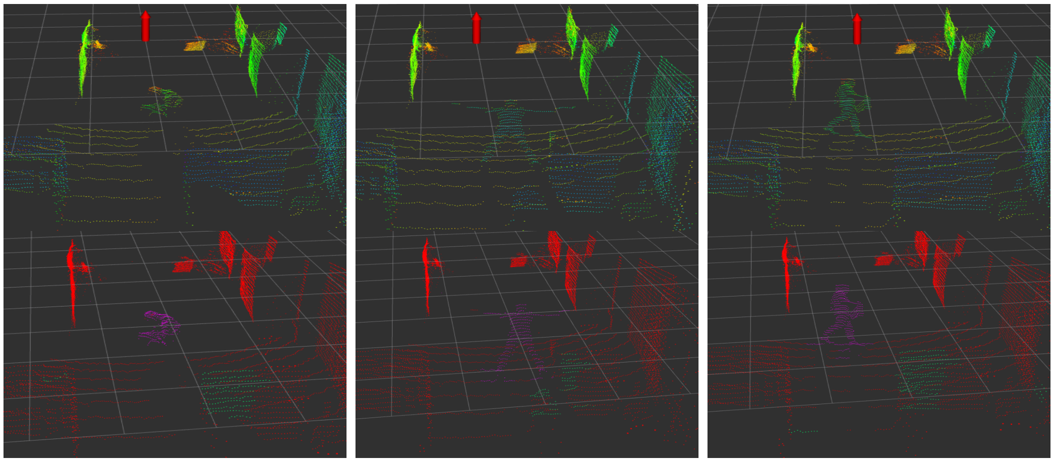


Fig. 7: Tracking results for extreme pose changes. Ranging from crouching, jumping jacks and holding an object. In the top row is the raw input data and the bottom row is the results of the tracking. Red points were classified as background and violet points as a person.

have an impact in the metrics, they never affect the tracking since a track is only started after 3 consecutive classifications and that is never the case for these false positives.

As seen in Table I in the $Poses$ video result, our prototype is indeed able to keep the tracking going even in the most difficult and extreme poses. This is possible thanks to the interaction between the classification and tracking modules. An example of some results of this video recording can be seen in Fig. 7.

On the other hand, the video $Occlusions$ is aimed to test how our prototype responds to a bad case of occlusions, even though they were not explicitly handled in the implementation. In the results we can see that even though the recall is highly affected by the amount of false negatives (due to the occlusions), the precision is still the highest metric and around 90%, which indicates that even in the most difficult cases our prototype seems to have a high confidence in the positive classifications.

Finally, the video $TwoPerson$ is evaluated as a proof of concept. Despite the fact that our solution was mostly intended for single person tracking it shows potential for multiple person tracking as well. The metric that decreases the most is the computing performance, outputting at 6.8hz down from the 10hz of input frequency. This is normal since the more persons there are in the scan, the more points the algorithm has to process. In Table I, the results are presented for each one of the targets, and we can see highly accurate results for the first one. On the other hand, the second target suffers from a high amount of false negatives which are caused by a mistake in the starting background model creation, but eventually the algorithm can correct the error.

### B. Experiment II

The second experiment is done in a smaller part of the building, it corresponds to a small rest area surrounded by halls. In this experiment, we recorded 4 different persons in similar situations as in the last experiment and the results can be seen in Table II.

Overall, we get good results for every situation. The baseline cases included the persons walking around, sitting in a chair and occasionally crouching to tie their shoes. As expected, we have good results for precision (above 90%) and decent results for recall, due to a high amount of false negatives.

The video $Poses2$ includes a person doing multiple ”aerobics-like” movements, and the person stays in the scan range for the whole video. Therefore, we see amazing results for precision, proving again that our prototype is robust against extreme pose changes and has an overall high confidence in positive classifications.

In the case of the video $Occlusions2$, the results for precision are similar to the ones in the first experiment, but it presents better results for recall because of a smaller number of false negatives. This only because the times when the target was partially or fully occluded were far less frequent than in the last experiment.

All in all, the results of both experiments are comparable and depend on the situation. The overall computation performance is slower in the second experiment. Since the room was smaller, the sensor was positioned closer to the targets. Therefore, it was more likely for the people to be very close to the sensor, which caused occlusions of the majority of the environment and therefore created problems in the algorithm that led to longer computation times.

TABLE II: Results for different situations in experiment II.

| Video | Precision | Recall | F1 Score | Freq |
|---|---|---|---|---|
| $Baseline2$ | 96.41 | 78.83 | 86.74 | 8.11 |
| $Baseline3$ | 90.6 | 83.33 | 86.82 | 7.21 |
| $Poses2$ | 97.51 | 91.35 | 94.33 | 7.71 |
| $Occlusions2$ | 91.23 | 87.93 | 89.55 | 7.37 |

## VI. CONCLUSION AND FUTURE WORKS

### A. Conclusion

In this work, we have presented a prototype for solving person detection and tracking in 3D point clouds in real time in a low performance machine. After discussing the different techniques for people detection and tracking with range sensors we have presented a method with robust object segmentation based on super-voxels and a chaining method. The classification of the objects plays an important role instead of being just a verification step like in other works, and it works in a simple and reliable way in order to classify objects that have the minimum required geometric features to be considered as persons.

Even though the classifier is not meant to deal with the variety of poses a person can assume, the integration of the classification module with the tracking result in an excellent handling of different extreme pose changes a person can undergo.

Furthermore, even though 3D point clouds are normally challenging because of the large amount of data to process, the achievement of a real-time implementation is a result of the techniques of movement detection and prediction, that limit the amount of points to be processed by finding and predicting regions of interest.

Overall, the preliminary validation and results presented in the last section prove that our prototype has a high confidence on positive classifications as it can be seen by the consistent results in the precision metric, even in harder situations like when dealing with occlusions.

### B. Future works

The continuation of our work on a bigger scale can be important for applications such as social robotics for following robots and applications in the industrial field with human-robot collaboration. The work presented in this article corresponds to the perception module in an autonomous system, and it could be expanded to work in conjunction with a robotic decision architecture that takes into account the way that the humans would react to the robot's movements.

More specifically, expanding upon the geometric methods to detect and segment a person in multiple levels: The person like an unique object for applications like robot following and the person as multiple objects (like arms, head, etc) linked together for applications like human-robot collaboration in industries.

On the other hand, another part would be the implementation of robust tracking methods to track a person in situations with complex dynamics and for tracking the different parts of a person with each their own dynamic. Besides, the introduction of a second 3D LiDar sensor with less limitations to do sensor fusion to capture the person from different positions can also be considered.

Finally, studying the interaction between the robot and the people, like movement and behavior prediction that can be integrated into the robot's movement planning taking into account the person's movement would be interesting for the application. Therefore, the robot would choose how to move according to how the person would react. Furthermore, an analysis of the implication of each one of these obstacles and of the social-relational values that humans will attribute to the robot's decision, and integration of these human factors into decision-making can be done.

In conclusion, our approach shows a lot of potential for future works and applications. And it shows the convenience and some of the advantages of working with a 3D LiDar sensor.

## REFERENCES

[1] Olivier Aycard Trung-Dung Vu. Laser-based detection and tracking moving objects usingdata-driven markov chain monte carlo.2009 IEEE International Conference on Roboticsand Automation, 2009, pp. 3800-3806, doi: 10.1109/ROBOT.2009.5152805.

[2] O. Aycard A. Azim.Detection, classification and tracking of moving objects in a3d environment.2012 IEEE Intelligent Vehicles Symposium, 2012, pp. 802-807, doi:10.1109/IVS.2012.6232303.

[3] Beñat Irastorza Renzo Verastegui Sebastian Süss et al. Bangalore Ravi Kiran, LuisRoldão. Real-time dynamic object detection for autonomous driving using prior 3d-maps.First International Workshop On Autonomous Navigation in Unconstrained Environments- In Conjunction with ECCV 2018, Sep 2018, Munich, Germany. hal-01890980.

[4] Armando Pesenti Gritti, Oscar Tarabini, JÃ©rÃ me Guzzi, Gianni A. Di Caro, VincenzoCaglioti, Luca M. Gambardella, and Alessandro Giusti. Kinect-based people detectionand tracking from small-footprint ground robots. In2014 IEEE/RSJ International Con-ference on Intelligent Robots and Systems, pages 4096–4103, 2014.

[5] Jun Liu, Ye Liu, Guyue Zhang, Peiru Zhu, and Yan Qiu Chen. Detecting and trackingpeople in real time with rgb-d camera.Pattern Recognition Letters, 53:16–23, 2015.

[6] Omid Hosseini Jafari, Dennis Mitzel, and Bastian Leibe. Real-time rgb-d based peopledetection and tracking for mobile robots and head-worn cameras. In2014 IEEE Interna-tional Conference on Robotics and Automation (ICRA), pages 5636–5643, 2014.

[7] Bihao Wang, Sergio Alberto Rodríguez Florez, and Vincent Frémont. Multiple obstacledetection and tracking using stereo vision: Application and analysis. In2014 13th In-ternational Conference on Control Automation Robotics Vision (ICARCV), pages 1074–1079, 2014.

[8] Takayuki Ikeda Tetsushi Miyashita Takahiro Brscic, Drazen Kanda. Person tracking inlarge public spaces using 3-d range sensors.Human-Machine Systems, IEEE Transactionson, 2013. 43. 522-534. 10.1109/THMS.2013.2283945.

[9] Florian Schler, Jens Behley, Volker Steinhage, Dirk Schulz, and Armin B. Cremers.Person tracking in three-dimensional laser range data with explicit occlusion adaption.In2011 IEEE International Conference on Robotics and Automation, pages 1297–1303,2011.

[10] Luciano Spinello, Matthias Luber, and Kai O. Arras. Tracking people in 3d using abottom-up top-down detector. In2011 IEEE International Conference on Robotics andAutomation, pages 1304–1310, 2011.

[11] Víctor Vaquero, Iván del Pino, Francesc Moreno-Noguer, Joan Solà, Alberto Sanfeliu,and Juan Andrade-Cetto. Dual-branch cnns for vehicle detection and tracking on lidardata.IEEE Transactions on Intelligent Transportation Systems, pages 1–12, 2020.

[12] Laurent Trassoudaine. Ahmad Kamal Aijazi, Paul Checchin. Segmentation et classifi-cation de points 3d obtenus à partir de relevés laser terrestres: une approche par super-voxels.RFIA 2012 (Re- connaissance des Formes et Intelligence Artificielle), Jan 2012,Lyon, France. pp.978-2-9539515-2-3.hal-00656538.

[13] R Omar Chavez-Garcia, Olivier Aycard. Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking. IEEE Transactions on Intelligent Transportation Systems, IEEE, 2015, PP (99), pp.1-10. 10.1109/TITS.2015.2479925 . hal-01241846