# General-Purpose Model-Free Object Tracking with 3D LiDAR

**Bruno Comesaña Cuervo**

**School of Electrical Engineering**

Thesis submitted for examination for the degree of Master in Space Science and Technology.

Espoo September 24, 2022

**Thesis supervisor:**

Ph.D. Tomasz Kucner

Aalto University
School of Electrical
Engineering

Author: Bruno Comesaña Cuervo

Title: General-Purpose Model-Free Object Tracking with 3D LiDAR

Date: September 24, 2022 Language: English Number of pages: 5+44

Department of Electrical Engineering and Automation

Major: Space Robotics and Automation

Supervisor: Ph.D. Tomasz Kucner

This thesis presents two different methods developed by the author in the area of 3D tracking of multiple objects using 3D LiDAR data.
The first method attempts to improve the overall performance of a reference tracking system [1] by using distance-dependent motion modelling. Two of the most common motion models, Kalman Filter and Constant Velocity, are combined using a weight function that depends on the distance between the tracked object and the sensor. As the results show, this technique proved to increase the complexity of the tracking algorithm and its processing load without providing any mayor improvements.
On the other hand, the second method was developed with the aim to improve the behavior of 3D multi-object trackers against occlusions and point cloud sparsity. This was achieved by simplifying the life management of the tracklets, consisting of removing both tracklet scoring from the tracklet life or death evaluation and the minimum hits required to generate a track, as well as increasing the number of frames in which a tracklet can survive without being associated with a detection. An important improvement was obtained regarding the number of identity switches for both vehicles and pedestrians. In the pedestrian case, the lowest value seen, at least in the field of 3D model-free multi-object tracking, was obtained, to the best of the author's knowledge.

Keywords: model-free, 3D LiDAR, motion model, score refinement, life cycle management, Tracklet, occlusions, point cloud sparsity, autonomous

## Acknowledgments

I want to start my acknowledgements section by expressing my deepest gratitude to my supervisor, Ph.D Tomasz Kucner, who has been my main source of helpful insights, constant guidance and support. There is nothing, in my opinion, that positively affects more the result of an extensive work like a master's thesis than having a great supervisor who is committed to guide you throughout the whole process.

On the more technical side, I feel necessary to mention the authors of CenterPoint [2] and SimpleTrack [1], who were really supportive. They provided different detection files and models to be used as the basis for the tracking systems presented here, allowing to speed up the development process by removing the necessity of implementing the whole detection system and instead focus directly on the tracking algorithm.

Lastly, I want to mention my parents and my partner, whose emotional support during these last six months was key to endure the characteristic uncertainty of developing this thesis.

Switzerland, 30.08.2022 Bruno Comesaña Cuervo

# Contents

# Abbreviations

| | |
|---|---|
| DATMO | detection and tracking of multiple objects |
| MOT | multi-object tracking |
| LiDAR | light detection and ranging |
| RADAR | radio detection and ranging |
| 2D | two-imensional |
| 3D | three-dimensional |
| SOTA | state-of-the-art |
| TP | true positive |
| FP | false positive |
| TN | true negative |
| FN | false negative |
| NMS | non-maximum suppression |
| WOD | Waymo Open Dataset |
| IDS | identity switches |
| FOV | field of view |
| KF | Kalman Filter |
| CV | constant velocity model |
| GT | ground truth |
| IoU | intersection over union |
| TOF | time of flight |

# 1 Introduction

Robotics is a research area of great interest due to its potential benefits for humanity. As recent examples like last-mile delivery robots [3] or service robots for catering applications [4] demonstrate, humans aim to automatize a wide range of tasks, focusing principally on boring, repetitive, physically demanding, and even dangerous ones. This is bound to free up people's energy and capabilities to be used on more satisfying and fulfilling endeavours, replacing activities that are "practical or necessary" for society with activities that are "chosen" by each individual. The concept of "It is not great but someone has to do it" will be vanished, taking the human species to a whole new level of minimum living standards.

In robotics research, the tracking of multiple objects plays an important role, not only for autonomous vehicles but also for a long spectrum of mobile platforms operating autonomously in shared environments. This capability equips the system with situation awareness, being able to locate and predict to a certain degree the spatial behaviour of all surrounding objects, which is essential if a robotic system is to act with a certain level of autonomy without affecting or threatening its environment. In consequence, detection and tracking of multiple objects, or DATMO for short, has been a rich area of research withing robotics for a many years.

In the following subsections, background knowledge is provided for a better understanding of the contents of this thesis, together with a clear exposition of the research problem at hand and an outline of the whole document.

## 1.1 Background

Four fundamental elements can be identified in any tracking system using data from a 3D LiDAR sensor: preprocessing, motion model, association and life cycle management [1]. Each one being focused on one of these components, this thesis presents two different techniques that have been developed in an attempt to create a 3D tracking system with a better performance than the current state-of-the-art.

One of these techniques affects the motion model, which is the module in charge of predicting and updating the state of all objects being tracked. Without creating a new motion model algorithm, this thesis aims to improve how commonly used models are applied to the task of tracking.
The other technique works with the life cycle module, which basically determines when the different detections are likely to correspond to real objects and therefore, worth to start tracking, and when this tracking has become unreliable or unnecessary, and therefore, not worth to continue. By modifying the tracks life and death policies, this thesis aims to decrease object misidentification.

Although the mentioned techniques are focused on just part of what constitutes a complete tracking system, all aspects of the system are conditioned by the particular applications it is designed for. Since the goal was to develop a model-free general-purpose outdoors system, neither of the two different tracking techniques that are proposed in this document make use of prior information about the objects that the sensor may encounter. In this way, the obtained system is highly flexible with regards to the environments where

it can be deployed and the objects with which it can co-exist.
The "general-purpose" aspect of the system refers to the possibility of applying it to any scenario, implying the absence of a target application for it. In consequence, the use of the methods developed and presented in this document is not conditioned by whatever the robotic platform's goal is. This feature can be considered part of the presented systems as they were developed without any particular application in mind, apart from it being a system aware of the state of any object in its surroundings, whatever they may be. More in particular, this generality is deemed achieved because the tracking techniques have been evaluated on the Waymo Open Dataset [5] which, although it is meant mainly for autonomous driving, provides scenarios complex and varied enough for the development of a tracking system intended for a wide variety of outdoor applications.

In order to arm robotic systems with the capacity to detect and track multiple objects, sensors that gather information about surrounding objects are required. Initially, RADAR technology, which uses radio waves in order to detect objects at relatively long distances and provide information about their position with respect to the sensor itself, was in the center of this field of investigation, mainly for military purposes [6][7].
With the development of new technology, DATMO research moved later on towards video and laser data, opening to a whole new level of information available for the detection and tracking tasks. Invented shortly after laser technology itself [8], LiDAR technology became slowly one of the main sensors utilized in DATMO research.
In order to provide a general idea of why LiDAR was used in this thesis as the sensor that provides the data, its main features are presented in comparison to other sensors typically found in the bussiness of detection and tracking of objects. Unlike RADAR, LiDAR uses light, predominantly in the infrared spectrum, to determine the location of objects. This is done by measuring the time of flight (TOF) that the laser beam emitted by the LiDAR sensor takes to come back to it after being reflected by an object.
Contrary to cameras, both RADAR and LiDAR provide information about the environment independently of the daylight, although LiDAR is negatively impacted by poor weather (e.g rain, fog). On the other hand, despite RADAR having a greater range than LiDAR sensors, it shows substantially more error in measurements at short distances, as it can be seen in Figure 1. The superior image quality and performance at short distances makes LiDAR sensors a superior alternative for autonomous robots, since the closer an object is to the vehicle where the sensor is mounted, the greater the need to know its state with accuracy. Nevertheless, RADAR, LiDAR and other sensors are often combined to benefit from their respective strengths [9][10][11].

Nowadays, LiDAR sensors can be 2D or 3D depending on how much information about the environment they are capable of gathering. Nevertheless, the first laser scanners were just 2D, which create a horizontal plane by emitting a single laser beam while spinning, creating in this way a horizontal plane that gathers the 2D range information of the surrounding environment. This can be better understood with Figure 2.
On the other hand, 3D LiDAR sensors add an extra degree of freedom on the vertical plane, making them a very attractive option but also a more complex one when it comes to interpret and use the raw data they produce. They generate a certain number of 2D planes at different degrees of inclination to provide information about the surrounding three-dimensional space. An example of raw 3D point cloud data can be seen in Figure 3.
Despite being initially focused on detection and tracking methods that employ 2D LiDAR [13], nowadays the interest and effort of the scientific community lies on approaches that

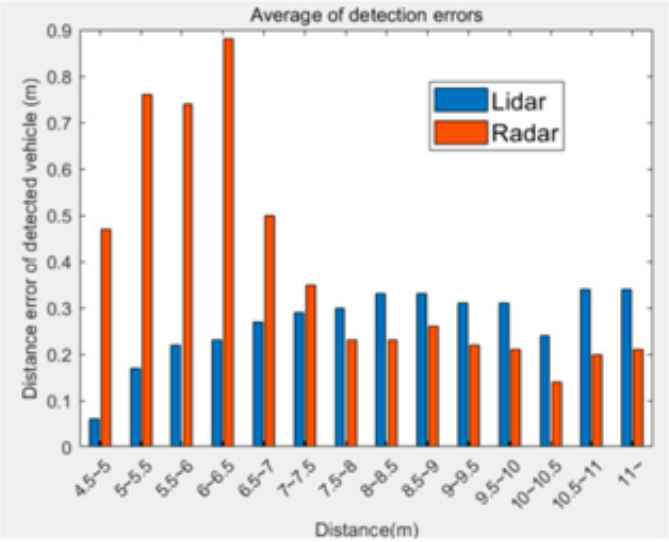



Figure 1: Measurement error of lidar and radar sensors depending on the distance [12]

use 3D LiDAR [14]. [6] documents quite a few examples of the combination of 3D data and 2D tracking methods since, initially, there were no methods to handle the 3D data directly. The usual approach to 3D tracking consisted in projecting the 3D point cloud into a 2D representation, as it can be seen in papers [15], [16] and [17], for example. Other methods would use 2D slices of the 3D point cloud in order to deal with hills [18]. Later on, algorithms capable of segmenting the 3D point cloud were developed [19], opening the door to tracking systems that directly use the 3D data.

Figure 2: On the left, an example of 2D laser scans in an outdoor scenario projected into an image. On the right, the corresponding raw point cloud data showing the different objects on the horizontal plane. [20]

This thesis has used 3D LiDAR data to carry out multi-object tracking, for which the already mentioned open source data set WOD was utilised. Its most relevant characteristics are discussed in Section 4.
These type of data sets are required because, as mentioned before, this document has focused on the concept of 3D multi-object tracking, or 3DMOT for short. This term refers

to the action of tracking more than one object at a time using data from a 3D sensor. For the tracking to occur, a 3D detector must provide a virtual representation of each object it finds for each frame, an object hypotheses known as bounding boxes, of which the MOT system does the tracking. Providing a visual example to help with the understanding of the concept, 3D bounding boxes can be seen in Figure 3.
Tracking itself consists in "connecting" or "linking" the bounding box given to an object in each frame where it has been located by the detector. This is done in order to identify this object as the same single object. The "connecting element" is known as trajectory. This process of first detecting the object and then assigning a trajectory to it is referred to as "tracking-by-detection", and it is one the most commonly used tracking paradigms [21][22]. When the action of tracking pertains more than a single object a time, each object has its own trajectory. In this case, the so called "tracklet-based trackers" divide this independent trajectories into smaller portions, known as tracklets, which connect the same objects found in consecutive frames. This type of tracker uses tracklets as the base for object association, obtaining in this way longer trajectories [23]. This is further discussed in the association part of Section 3.

To the present time, one of the most commonly used approaches to the tracking of multiple objects is model-based. This approach counts with the advantage of knowing a priori what to look for, which increases the amount of cues available comparing to the model-free alternatives, making it easier to obtain a higher precision MOT system. On the other hand, this system is less flexible with respect to unexpected obstacles and scenarios, which is exactly the strength of model-free methods. An example of this can be seen in [24], where the authors avoided the use of models in order to make their system, which consists into applying SLAM to the detection and tracking of multiple objects, usable in any type of environment. Not only to benefit from its advantages, but also to contribute to the less researched alternative, this thesis has focused on the area of model-free 3D MOT.

## 1.2 Research problem

Among the many different sensors available for 3D MOT [15], 3D LiDAR stands out due to its accuracy, wealth of spatial data and light independency. Despite the advantages it provides, 3D LiDAR technology must deal with two mayor challenges: increasingly sparse point cloud as the distance from the sensor grows, and the occultation phenomenon caused by obstacles, which is known as occlusions. One of the main consequences of this with respect to object tracking is the difficulty to prevent identity switches (IDS), which represent the number of times that a tracked trajectory changes its matched ground truth identity [25]. That is why these challenges have shaped the purpose of this thesis and the two MOT techniques presented in it.

For the purpose of reducing the mentioned negative implications of point cloud sparsity and occlusions, this thesis was built around two different areas of a typical 3D MOT system: the motion model and the life cycle management.
In any MOT algorithm, there are different motion models that can be used for the prediction and update of the robot's state estimation. Among them, the ones that stood out the most due to the frequency with which they are used are Kalman Filter and Constant Velocity Model [1]. Due to its central role in the tracking of objects, the motion model could be described as one of the essential building blocks of any MOT system. Therefore, an improvement in this area could potentially generate a significant improvement in the system as a whole. This, as it will be seen in Section 3.4, is tightly connected to the mentioned challenges faced by 3D LiDAR MOT.
In an attempt to benefit from the advantages that these common motion models posse, while at the same time compensating their respective drawbacks, this thesis comes to answer the following question:

**Does the distance-dependent combination of two of the most common motion models affect, in a significant manner, the overall performance of a 3D MOT system?**

On the other hand, while researching possible novel approaches to life cycle management, it was found that, despite some interesting ways of dealing with occlussions and sparsity seen in recent papers [26] [27], this continues to suppose a problem with potential for improvement in the area of tracklet life management. Here is where the second research question was determined:

**How can the negative impact of occlusions and point cloud sparsity on the performance of 3D MOT systems be reduced?**

## 1.3 Overview

The next section covers the literature review that has been carried out on relevant papers in order to gather enough knowledge about the past and current SOTA. Moreover, this research was also aimed at looking for gaps in knowledge in the field of model-free multi-object tracking using 3D LiDAR data.

Section 3 presents all the necessary details to comprehend the two different techniques developed for this thesis, both the method that pertains the motion-model subdivision shown in Subsection 3.4 and the life cycle management method shown in Subsection 3.5. This entails a detailed description of the common aspects that both techniques share regarding the complete 3D MOT pipeline, the preprocessing and the association steps, as well as the 3D object detector used to provide the bounding boxes. Performance results on the WOD and nuScenes benchmarks are provided to justify the detector used.
To ensure a clear understanding of what has been done in both cases, the thought process that lead to the different assumptions and actions taken is presented as clearly as possible, together with clarifying code whenever necessary.

After a clear exposition of what has been done to produce new results, the results themselves are presented in Section 4. This consists of two subsections. In Subsection 4.1, a description is provided of the data sets employed as a substitute for a real sensor, together with the different metrics those benchmarks utilize to evaluate the performance of 3D MOT systems. In Subsection 4.2, the numerical results are presented in tables together with the results obtained by other relevant 3D MOT systems.

At this point, both the actual contribution made by this thesis to the area of 3D MOT using 3D LiDAR data without appearance models, as well the attempt to obtain further improvements via the combination of the existing models according to the separation between the sensor and the object being tracked, have been presented. Furthermore, the methods used to evaluate the results should also be clear. Therefore, the following is Section 4.3, which is used to discuss the author's understanding of the presented results.

Lastly, conclusions obtained after finishing this thesis and analysing methods and results are presented in Section 5, together with possible future improvements to be made on top of this work.

# 2 Related Work

As indicated in Subsection 1.3, this section contains most of the references used to build, from scratch, the background knowledge for this thesis. Furthermore, it was this literature review that provided the knowledge upon which this work is sustained.

As this thesis focuses on the use of model-free approaches to tracking using 3D LiDAR data, model-free papers relevant to this field have been thoroughly analysed.
[28] shows an approach to model-free tracking based on motion cues, which are extracted sequentially from consecutive frames and then used to segment objects with a Bayesian approach. This paper provided an initial new perspective on possible alternative sources information that could be used for tracking instead of prior knowledge of the environment and/or the objects in it.
[27] shows how, without relying on models of the objects or the environment, the negative impact that occlusions and point cloud sparsity have on the performance of a tracking system can be reduced. This paper shows a rather alternative approach to it. Instead of focusing on the software aspect of the system, they propose a mobile 3D LiDAR sensor located on an articulated arm in order to scan occluded objects from their flanks. For the sparsity challenge, they modify the angular step of the scanner according to the point cloud density on the object targeted.
[24] approaches both detection and tracking of objects as a single problem, having the sequence of 3D measurements provided by either laser scanners or time-of-flight (TOF) cameras as the only input information. This constitutes yet another example of how information about the precise objects and/or the environment is not the only way of approaching neither tracking nor detection.
[29] is another example of model-free tracking. In this case, detection and tracking are also approached as joint problem in order to avoid the limitations of the more usual "detection and tracking" setting. They focus on single object tracking (SOT) instead of multiple object tracking (MOT).
[30] proposes the use of an object-related local grid and a particle filter with an adaptive probability function to reduce the area for measurement association, avoiding the need for a geometric model. This was the first paper found during the research for this thesis where a particle filter was used, as the most common case in this literature review is the application of kalman filter to motion modelling.
[31] is a master's thesis that suggests a new freespace querying algorithm to distinguish between dynamic and static objects when tackling the problem of object detection. For this motion-compensated scan alignment is used, obtaining finally a model-free setting-independent detection method that uses 3D LiDAR data. This paper provided useful insights towards simple object classification that does not require previous information.
[32] chooses to focus in the differentiation between ground and obstacles in order to reduce the processing load natural to model-based detection methods. This is done via a graph based a local convexity criterion. As the previous paper, this is another example found of simple classifications that can be carried without previous information about the objects.
[14] provided a clear overview of the generic algorithm necessary to detect objects with laser scanners, together with an extensive list of DATMO papers with different setups, which provided not only a useful overview of the past of detection and tracking systems, but also an understanding of the chronology of the different technologies and algorithms, including the transition from 2D towards 3D systems. Although oriented towards the

use of 2D LiDAR data, [33] proposes a hierarchical approach to solve the problem of detecting and tracking moving objects without using prior information about them, giving a good introduction into the model-free paradigm and its advantages and disadvantages.

Some literature review has also been conducted on model-based methods in order to have a clear understanding of the advantages and disadvantages that implies choosing not to exploit prior information on potential objects the tracking system may encounter. [34] presents a hybrid of neural network model based and occupancy map based approaches, where dynamic objects are removed from 3D point could maps, which affect their quality. [35] tackles the problem of encoding point clouds into a format ready for the downstream detection. This became state-of-the-art (SOAT) for model-based 3D detection methods. It makes use of PointNets to learn a representation of the point cloud organizing it in vertical columns.
[36] takes advantage of knowledge about object model, measurement model and motion model to apply a Bayesian framework. This is used to make sense of laser measurement sequences, using a Data-Driven Markov Chain Monte Carlo (DDMCMC) to find the optimal solution.
[37] encodes model and candidate shapes into a compact representation. This is done by applying a Siamese tracker for shape completion in 3D point clouds with the goal of tracking objects.
[38] presents an efficient segmentation method for 3D LiDAR data. The clusters of raw data are then either processed via supervised or unsupervised classification algorithms, or increased with information provided by other sensors.
[39] applies a recurrent neural network to the incoming raw LiDAR data, obtaining in this way the object location and their identity.

[6], [2], [40] and [41] represent, among other systems, objects as points instead of the more commonly bounding box, which are two fundamentally different approaches to object detection. While [6] centres itself on the effect this object representation has on simple 3D tracking algorithms, [2] offers a whole new framework for 3D object detection and tracking of multiple objects. Also class-independent (model-free) and without the use of maps, [42] detects objects via a freespace querying algorithm that compensates the movement of the LiDAR scanner while performing the scanning, which adds complexity to the task but removes the distortion causes by the movement of the vehicle while sensing. Examples of works where bounding box representations are used are [26], [1], [28] and [24], among many others.

[1] and [43] dive into the current solutions proposed to the 3D MOT problem, subdividing them into four basic building blocks: pre-processing of detection, association, motion model, and life cycle management. In this way, these papers clarify the contributions to the general MOT performance made by each individual building block, highlighting also the different interactions between blocks. This leaves the door open, which has been heavily taken advantage of in this thesis, for new SOTA solutions that use the "tracking-by-detection" paradigm, predominant in the field of 3DMOT.
An example of trackers directly dependent on the detector's performance is [44], which attempts to improve object tracking performance by modifying the manner in which tracklets are rated, from the common approach of just giving tracklets the same score as the associated detection to providing tracklets with their own scoring logic. This logic makes use of historic information about the tracklet's previous score and whether it has been

matched with a detection in the current frame or not.
Another example is [43], which presents a baseline system constituted by standard methods to generate SOTA results.
[2], [26] and [1], which have already been mentioned, also make use of the tracking-by-detection paradigm.
The difficulties associated with having a "detection-dependent" tracking method are avoided, and the consequent benefits highlighted, by [29].

Finally, a simple yet effective solution to the identity-switch phenomena caused by occlusions and point cloud sparsity that 3D MOT faces is proposed by [26]. The basic idea is to modify the classic attitude adopted towards track's life management, which usually consist in killing the track if it is stays invisible for a certain number of frames, and it is instead kept alive indefinitely (immortal tracks), outperforming previous LiDAR methods. This idea is utilized in a special way by one of the techniques presented in this thesis, in Subsection 3.5.

# 3 Methods

In this section, the two methods around which this thesis has been built are described. It has already been mentioned that within a common 3DMOT system, four fundamental modules can be identified: preprocessing, motion model, association, life cycle management. As indicated in Subsection 1.2, this thesis covers two different approaches to the usual policies seen in motion modelling 3.4 and life cycle management 3.5. Although these methods offer novel views on two building blocks within the 3D MOT pipeline, they share the other elements that conform it. Moreover, the 3D detector is the same for both alternatives. Therefore, in order to clearly show what this thesis intended to contribute with the two developed methods, the commonalities together with the utilized 3D detector are presented first, while the special aspects of each method are presented in their respective subsections.

## 3.1 3D Object Detector

The detection segment of a 3D DATMO system is in charge of processing, in this case, the point cloud inputs coming from a 3D LiDAR sensor and generating 3D bounding boxes, which work as virtual representations of the objects detected by the 3D LiDAR sensor, which in turn the 3DMOT system employs for the purpose of tracking the surrounding objects. This high-level view of the DATMO pipeline is simplified on Figure 4.
As mentioned in the background subsection 1.1, although the raw data utilized by the detector comes from a 3D LiDAR, this thesis has made use of the Waymo Open Dataset instead of using the data coming directly from a sensor.

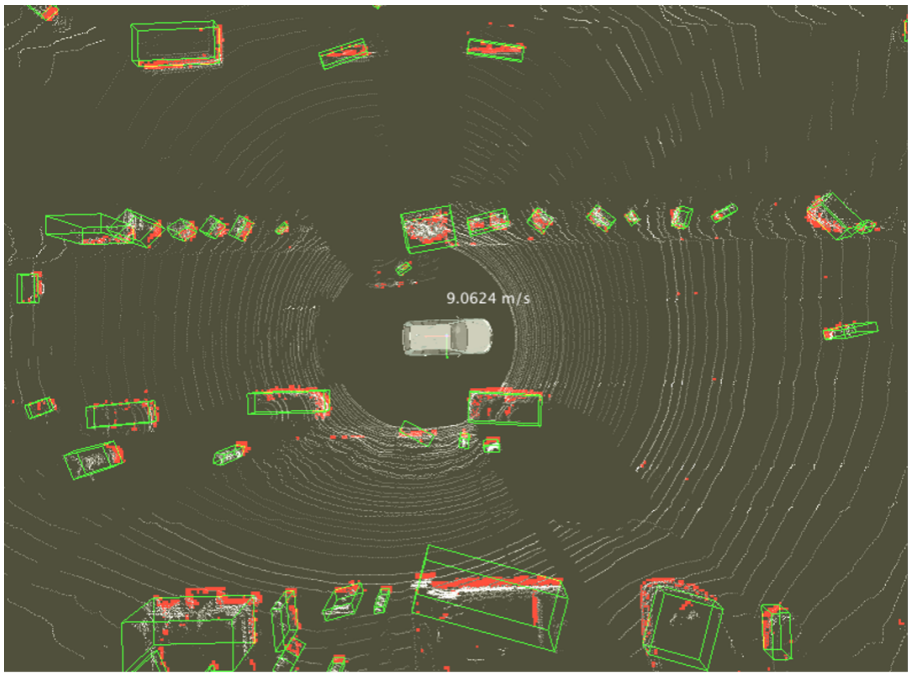



Figure 3: Red dots show cloud points colliding with objects and green boxes are bounding boxes [20]

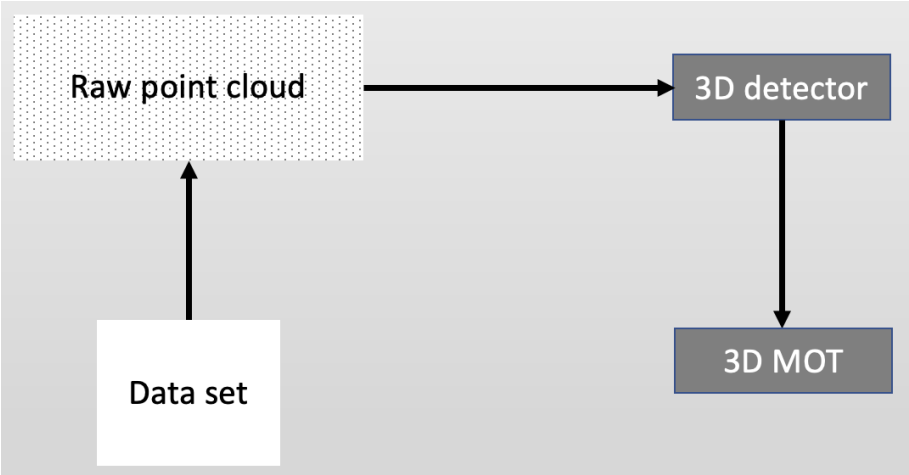



Figure 4: Simplified DATMO pipeline

In this thesis, the 3D obect detector CenterPoint presented in [2] was used for both techniques, as it was found to be the best performing with regards to object detection using 3D LiDAR. This can be seen in tables 1, 2 and 3, where the results they published show superior performance according to every metric used by the WOD and nuScenes benchmarks.

| Difficulty | Method | Vehicle | | Pedestrian | |
|---|---|---|---|---|---|
| | | mAP↑ | mAPH↑ | mAP↑ | mAPH↑ |
| Level 1 | StarNet [45] | 61.5 | 61.0 | 67.8 | 59.9 |
| | PointPillars [35] | 63.3 | 62.8 | 62.1 | 50.2 |
| | PPBA [45] | 67.5 | 67.0 | 69.7 | 61.7 |
| | RCD [46] | 72.0 | 71.6 | | |
| | **CenterPoint** [2] | **80.2** | **79.7** | **78.3** | **72.1** |
| Level 2 | StarNet [45] | 54.9 | 54.5 | 61.1 | 54.0 |
| | PointPillars [35] | 55.6 | 55.1 | 55.9 | 45.1 |
| | PPBA [45] | 59.6 | 59.1 | 63.0 | 55.8 |
| | RCD [46] | 65.1 | 64.7 | | |
| | **CenterPoint** [2] | **72.2** | **71.8** | **72.2** | **66.4** |

Table 1: Comparison table of 3D SOTA detection showing the superiority of CenterPoint on the WOD test set [5]

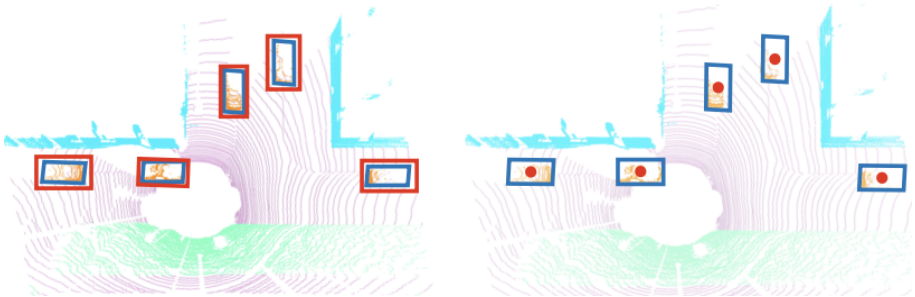


Figure 5: Visual difference between bounding box and point representation [2]

| Method | mAP↑ | NDS↑ | PKL↓ |
|---|---|---|---|
| WYSIWYG [47] | 35.0 | 41.9 | 1.14 |
| PointPillars [35] | 40.1 | 55.0 | 1.00 |
| CVCNet [48] | 55.3 | 64.4 | 0.92 |
| PointPainting [49] | 46.4 | 58.1 | 0.89 |
| PMPNet [50] | 45.4 | 53.1 | 0.81 |
| SSN [51] | 46.3 | 56.9 | 0.77 |
| CBGS [52] | 52.8 | 63.3 | 0.77 |
| **CenterPoint** [2] | **58.0** | **65.5** | **0.69** |

Table 2: Comparison table of 3D SOTA detection showing the superiority of CenterPoint on the nuScenes test set [53]

| Difficulty | Method | Vehicle | | Pedestrian | |
|---|---|---|---|---|---|
| | | mAP ↑ | mAPH ↑ | mAP ↑ | mAPH ↑ |
| Level 1 | DOPS [54] | 56.4 | | | |
| | PointPillars [35] | 56.6 | | 59.3 | |
| | PPBA [45] | 62.4 | | 66.0 | |
| | MVF [55] | 62.9 | | 65.3 | |
| | Huang et al. [56] | 63.6 | | | |
| | AFDet [57][58] | 63.7 | | | |
| | CVCNet [48] | 65.2 | | | |
| | Pillar-OD [59] | 69.8 | | 72.5 | |
| | PV-RCNN [60] | 74.4 | 73.8 | 61.4 | 53.4 |
| | **CenterPoint-Pillar** [2] | **76.1** | **75.5** | **76.1** | **65.1** |
| | **CenterPoint-Voxtel** [2] | **76.7** | **76.2** | **79.0** | **72.9** |
| Level 2 | PV-RCNN [60] | 65.4 | 64.8 | 53.9 | 46.7 |
| | **CenterPoint-Pillar** [2] | **68.0** | **67.5** | **68.1** | **57.9** |
| | **CenterPoint-Voxel** [2] | **68.8** | **68.3** | **71.0** | **65.3** |

Table 3: Comparison table of 3D SOTA detection showing the superiority of CenterPoint on the WOD validation set [5]

As indicated in section 2, while the most common approach to the representation of 3D objects in a point cloud is via bounding boxes, CenterPoint [2] introduced the idea of representing them as points, which as the authors claim, removes the difficulties bounding boxes have with enumerating all orientations and with fitting boxes aligned with the axis to objects that rotate with respect to the sensor. The different between these two representation approaches can be better visualized with Figure 5.
For the detection, it uses a keypoint detector and then regresses the 3D features of the detected object, generating more reliable bounding boxes than those obtained directly. Due to intrinsic lack of orientation points have, the search space the detector must handle is smaller than in the case of bounding boxes, simplifying downstream task like tracking. Furthermore, the its two-stage refinement module is faster than other center-based works.

### 3.2 Preprocessing

Once within the tracking system, the first module to be described is the preprocessing. The preprocessing step entails the processing of all bounding boxes put out by the 3D detector. It is characterized by the selection of the boxes that will actually be used by the tracking system, working in this way as sort of filtering element. For this, two new concepts are introduced: (1) confidence score and (2) detection threshold. Confidence score is a value provided by the object detector in a percentage form that represents the probability of an object having been correctly detected [61], whereas the detection threshold makes reference to a particular value of this score which is used as a threshold for a certain change in the behaviour of the tracking system with regard to that detection.
Both systems presented in this thesis make use of a detection threshold as part of their policy to determine the bounding boxes that will be used by the 3D MOT, filtering out the ones with a confidence score below such threshold.
Although this approach is shared by both tracking methods presented below, the threshold itself is not exactly the same. Therefore, this detection threshold is again mentioned in the pertinent subsections, in order to keep separated the shared aspects from the uniqueness of each method.

When only this threshold is used as the "filtering policy", the following applies: If the score of a certain detection is above or equal to a given threshold, it is used by the MOT, and *vice versa*. This can be visualized in Figure 6.

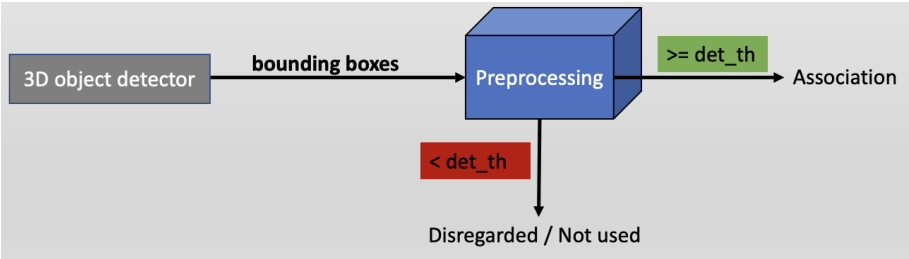



Figure 6: Preprocessing with just a detection threshold (det_th)

With this simplified policy, the recall of the system is negatively affected by high detection thresholds. Recall, also known as sensitivity, is calculated according to the following equation [62]:

$$sensitivity = recall = \frac{TP}{TP + FN} \tag{1}$$

Where TP refers to "true positive" detections, which are the detections taken into account by the MOT system that correspond to correctly identified objects, and FN refers to "false negative" detections, which are the detections falsely disregarded by the MOT system.
The following equation shows the relation between FN, TN, FP and TP, corresponding their total sum to the total number of detections:

$$1 = (FN + TP) + (FP + TN) = positives + negatives \tag{2}$$

The detection threshold acts as a quality filter. If it increases, the minimum confidence that the detector must have on a detection in order to be taken into account by the MOT also increases. In consequence, the average quality of the accepted detections increases:

$$\uparrow det_{th} = \uparrow Q \tag{3}$$

Where $det_{th}$ is the detection threshold and $Q$ is the quality of the boxes accepted by the MOT system.
Therefore, as the tracking system becomes "more demanding" regarding the minimum detection score, the likelihood of the MOT accepting detections falsely identified as real decreases:

$$\uparrow det_{th} = \downarrow FP \tag{4}$$

On the other hand, the sheer number of detections disregarded as incorrect (FN) increases. Although a low confidence score means that the likelihood of the detection being incorrect is high, this does not necessarily mean that it is actually incorrect.
This is the problem with this simplistic approach to preprocessing. There are occasions in which the detector is only capable to provide poor quality bounding boxes for real objects, but they are nevertheless correctly detected.

$$\uparrow det_{th} = \uparrow FN = \downarrow TP \tag{5}$$

Where FN and TP are complementary, as they represent the total number of detections corresponding to real objects: $1 = FN + TP$.

This logic shows, by combining Equation 1 with 5, that increasing the detection threshold directly decreases the sensitivity of the tracking system, proving what was previously stated. The denominator in the recall equation 1 increases more than the numerator.

$$\uparrow det_{th} = \downarrow sensitivity \tag{6}$$

In order to keep the recall high for high detection thresholds so that the benefits of a high threshold can be reaped, meaning decreasing the FP without increasing the FN, [1] and [26] apply a strict version of the non-maximum suppression score, or NMS for short, to the input detections.
NMS is a fundamental preprocessing step for most computer vision applications [63], including 3D MOT. It removes overlapping bounding boxes of lower quality and keeps the diverse low-quality ones. Adding NMS to the preprocessing module would modify the

algorithm symbolized in Figure 6, which would look more like what is shown in Figure 7. The two techniques presented in this thesis make use of this pre-processing tool due to the benefits just mentioned.

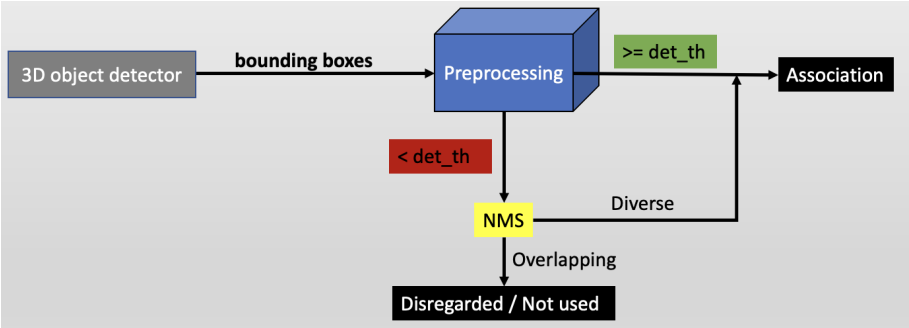



Figure 7: Preprocessing based on a detection threshold (det_th) and NMS

### 3.3 Association

As the preprocessing module, the association approach is the same for both tracking methods presented in this document.
Association is another of the fundamental modules of a MOT pipeline [21]. Both tracking methods presented in this thesis utilize tracklet-based association which, as mentioned in Subsection 1.1, makes use of tracklets in order to associate the state predictions of the different detected objects with the detected objects themselves. These predictions are obtained via a motion model, which is described in depth in Subsection 3.4. The result of this process is the classification of detections and tracklets in three groups:

- Matched: the tracklet was successfully associated, according to the pre-established conditions, with an existing detection
- Unmatched detections: corresponding to either newly detected objects or to already existing objects that couldn't be successfully associated to any of the existing tracklet
- Unmatched tracklets: for tracklets that could not be associated to any of the existing detections

As seen before, DATMO using a tracking-by-detection paradigm starts by detecting all objects in each frame, and then they are linked based on their similarity. The connection of these links between frames form the trajectories.
In tracklet-based association, two steps can be identified [64][1]:

- Step 1: A score is computed to indicate the level of similarity between a bounding box provided by the detector in the current frame and the predicted state in the current frame of the bounding box that forms part of a tracklet from the previous frame.
- Step 2: Some sort of matching strategy is used to link the tracklet and the current detection, building up trajectories.

Using the mentioned tracklet-based association, the MOT system is better protected against error detector responses and individual missing detections [23][64].
Furthermore, a two-stage approach to association has been applied in order to reap the benefits presented by SimpleTrack [1]:

- Stage 1: The previous two steps are applied to the detections and tracklet predictions, obtaining the three mentioned types of objects: matched detections, unmatched detections and unmatched tracklets.
- Stage 2: Unmatched tracklets go through stage 1 all over again, this time making used of a lower matching score.

The extra stage enables to keep the number of missidentified objects low, since tracklets are deemed "unmatched" as easily as when only the first stage is used. The similarity score required for matching a detection with a tracklet prediction is lower in the second stage than in the first stage. This associations of lower quality would imply an increase of false positives [1], hence these tracks are not output. They are simply kept alive in memory.

A simplification of the MOT steps in the first two frames is shown below to give a clearer idea of how this type of association works:

- $1^{st}$ frame:
  - The detector assigns a bounding box to all detected boxes in the frame
  - The MOT selects only those with the required quality to be tracked
  - A tracklet is initialized for each bounding box selected by the MOT, which includes a copy of the bounding box itself
- $2^{nd}$ frame:
  - The detector assigns a bounding box to all detected boxes in the frame
  - The MOT selects only those with the required quality to be tracked
  - Using a motion model, the state of the previously initialized tracklets is predicted in order to estimate where they could be in this frame
  - The similarity between the predictions and the detections is computed
  - Based on the similarity scores, tracklets and detections are associated using a matching algorithm
  - Depending on whether existing tracklets and detections are matched, they are classified as shown above

Obviously, this is not a thorough description of the MOT process. These tracklets are initiated and removed depending on the life cycle policies in place, which are covered in each of the following two subsections because they are different for each presented technique.

As already mentioned, the association step can itself be subdivided in two steps: similarity computation and matching.
Similarity computation consists basically in determining how close the prediction and

the detection are. Some of the methods most commonly used to compute the similarity are either IoU-based, as seen in tracking systems like the baseline 3DMOT presented in AB3DMOT [43], ImmortalTracker [26] and the single-object tracking system, SOT for short, presented in [29], or distance-based, which is used by MOT systems like [65] and [2]. IoU, or intersection over union, is a term used to describe to what extent two intersecting boxes are overlapping each other [1][66]. The greater the IoU, the greater the overlap and, in consequence, the greater the likelihood they will be matched in the following step of the association process. This concept can be better visualized with Figure 8.

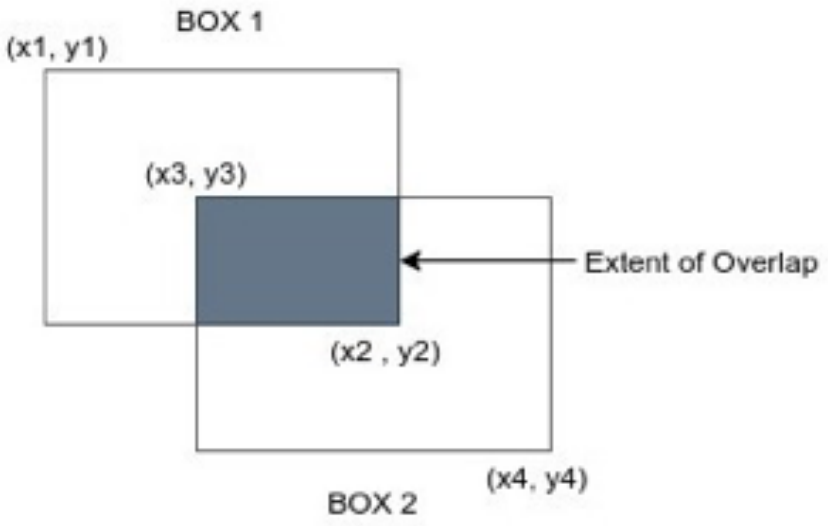



Figure 8: The grey area shows the overlap between box 1 and box 2 [66]

The equation to compute the IoU score looks as follows:

$$IoU = \frac{I}{U} \tag{7}$$

Where I corresponds to the area of intersection of two boxes, as shown in Figure 8, and U corresponds to the area of union of two boxes, meaning the total area covered by both boxes.
IoU values range from 0 to 1, and if the two boxes do not overlap each other, IoU is 0, as Equation 7 shows. On the other hand, if the boxes overlap each other completely, IoU will be 1.

Distance-based metrics on the other side score the "closeness" of the prediction and the detection in terms of the distance that separates them. Distance-based alternatives usually found in 3DMOT are Mahalonobis distance [67], like in [65], and L2 norm [68], also known as Euclidean distance, like in [2].
L2 norm is the straight line between two points, which can be calculated with the following equation:

$$L2 = \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2} \tag{8}$$

Where the coordinates of point 1 are (x1,y1,z1) and the coordinates of point 2 are (x2,y2,z2). On the other hand, Mahalonobis distance makes reference to the distance between a point and a distribution. It is basically the multivariate equivalent of the L2 norm:

$$L_{Mahalonobis} = \sqrt{(x-m)^T * c^{-1} * (x-m)} \tag{9}$$

Where $L_{Mahalonobis}$ is the Mahalonobis distance, $x$ is the observation vector, $m$ is the vector of mean values of independent variables and $c$ is the covariance matrix of independent values.

As just mentioned, the IoU association metric doesn't allow the association of two bounding boxes if they don't overlap each other at all, this is, when IoU is 0. This is a common value to be found among newly created tracklets and objects with abrupt motions. Therefore, this is a substantial drawback of this similarity computation method. Distance-based metrics are well suited for these occasions but they lack sensitivity for low quality detections at close distance, as they don't provide orientation discrimination [1]. GIoU [69], or Generalized IoU, is a sort of combination of both IoU and distance-based metrics, which manages to avoid the disadvantages presented by each method while benefiting from the their respective strengths. [1] generalized GIoU for its use in 3D applications.

Regarding matching algorithms, some of the most common are either the Hungarian algorithm [70], applied by tracking methods like [43], [71] and [72], which used it to solve a bipartite problem, and the greedy algorithm, used for the solution of the association problem by approaching it as the association of nearest pairs by MOT systems like [65] and [2]. As demonstrated in [1], both matching algorithms go well with IoU-based metrics, whereas the greedy algorithm only performs well with distance-based metrics. This comparison can be seen in Figure 9.

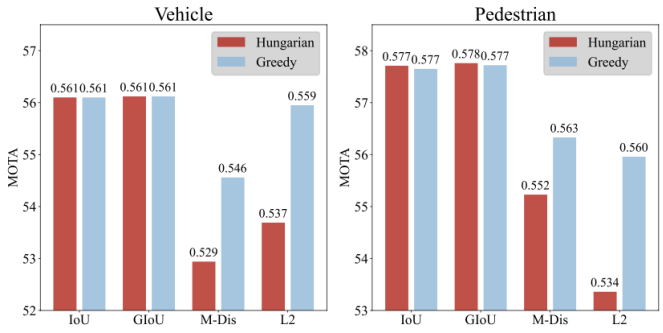



Figure 9: MOTA results for both vehicles and pedestrians for different IoU and distance-based metrics applied together with either the Hungarian or a greedy algorithm [1]

Both tracking systems presented in the following two subsections make use of the 3D GIoU metric, while the selected matching approach is the Hungarian algorithm. This decision is justified by the following two reasons: 1) To benefit from the advantages presented above, and 2) to create a base MOT system similar enough to different SOTA

trackers like SimpleTrack [1], ImmortalTracker [26] and AB3DMOT [43], which make use of the same configurations. In this way, a fair comparison is obtained that highlights the key differences of the present document and works as a reference to understand the results generated.

## 3.4 Motion model modification

This section contains the details that characterize the first MOT system developed for this thesis. It aims to answer the first of the research questions presented. This question, as already shown in Subsection 1.2, is:

**Does the distance-dependent combination of two of the most common motion models affect, in a significant manner, the overall performance of a 3D MOT system?**

The purpose of the motion model in object tracking is to predict and update the state of tracklets from the previous frame into the current frame. The presented motion modelling based method consists in the "combination" of two different models with the purpose of benefiting from their respective advantages, as well as observing whether their individual drawbacks can be complemented by each other. In particular, Kalman Filter and Constant Velocity Model were selected to test this idea, as they are some of the motion models most commonly used nowadays in 3D MOT, which can be seen in some recent papers already referenced like [1], [2], [26] and [43], as well as in older papers like [71] and [72].

During the literature review conducted with the aim of finding a knowledge gap to focus this thesis on, the characteristics of these two motion models stood out due to an apparent distance-dependent quality. KF takes advantage of multi-frame information in order to make the results smoother when the detections are of low quality, whereas CV does not offer such level of motion smoothing but provides a better behavior when dealing with unexpected and unpredictable movements [1]. SimpleTrack shows the respective performance of the abovementioned models on both WOD and nuScenes data sets. From these results it was drawn the conclusion that KF is better suited for high frame rates due to the higher movement predictability, whereas CV performs better in low frame rate situations thanks to the explicit speed predictions.

The motion model's characteristics previously highlighted triggered the distance-dependency idea upon which the approach to motion modelling presented here is based. The particular line of reasoning is clarified as much as possible below. Regarding the applicability of KF:

- KF performs better with low quality detections [1]
- At long distances, the sparsity of the point cloud is higher than at short distances, which means that there is a direct correlation between point cloud sparsity and sensor-object separation
- If the number of cloud points that collide with a certain object decreases, the quality of the detected object's virtual representation worsens. It is therefore assumed that low quality detections are predominant at high distances between the detected object and the sensor
- Conclusion: the performance profile a Kalman Filter motion model could imply it is a superior approach at large distances

And regarding the applicability of CV:

- CV is better than KF when it comes to abrupt and unpredictable motions [1]
- These types of movements can be expected both at short and long distances. This assumption is based on the following reasoning:
  - Simple geometry can justify the higher amount of abrupt motions perceived at shorter distances. As it can be seen in Figure 10, depending on the sensor-object distance, a certain movement is appreciated as more or less abrupt. If the object moves the same distance at two different object-sensor separations, the movement is perceived as greater when done at the shorter separation, this is, the angle between its initial and end position is greater
  - On the other hand, the direct correlation between point cloud sparsity and sensor-object separation explained for the KF case, together with an increased likelihood of an object being occluded by another when at greater distances, seems to indicate that the frequency of the detection updates decreases as the distance increases. This "update frequency" can be also described as smoothness (opposite of abrupt), and based on the logic just exposed, it seems to be reasonable to expect it to worsen as the distance increases
- CV makes use of an extra dimension of information: speed [1]
- Conclusion: the extra dimension of CV, together with the motion abruptness, for which this motion model is better suited and that could be found at different distances, indicates a certain level of distance codependency which this thesis attempts to test

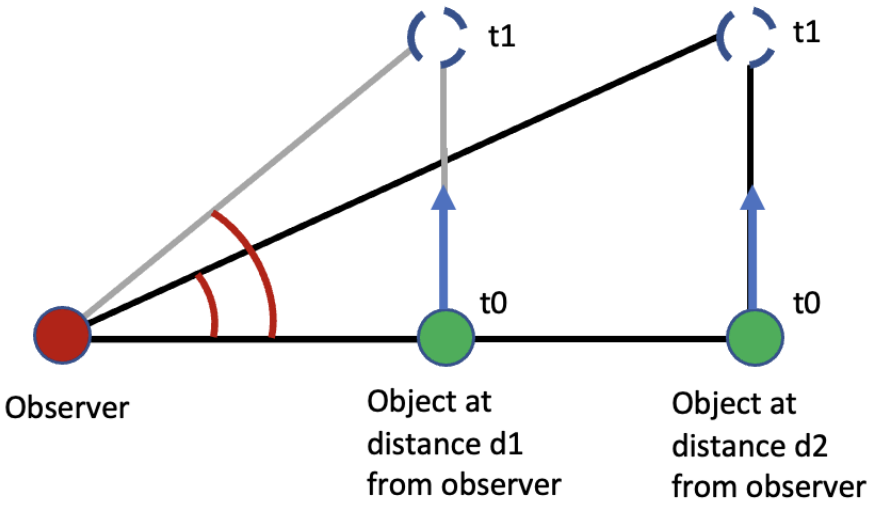



Figure 10: How abrupt a movement appears to be is influenced by the distance from the observer at which it occurs. The angle, shown as a red arc, of the movement between position at $t_0$ and $t_1$ indicates how abrupt it appears to the observer. A movement with a higher relative angle seems "more abrupt" than one with a smaller angle, given that it is the same displacement taking the same amount of time

As it can be deduced from the reasoning presented above, testing different combinations of KF and CV depending on the distance was deemed necessary in order to clarify whether

the predominant use of either KF or CV at shorter or longer distances could provide an overall improvement on the tracking performance.
The initial approach to the practical application of this idea was simple: In order to make use of two different motion models, a way of combining them depending on the distance between the sensor and the detected object was needed, a way of mathematically expressing the idea "If the object is at a distance less than $d$, then motion model A will be used. If not, motion model B will be used". Of course, this was just the initial simplified concept, which can be better visualized with the example code shown in Appendix A.1.

Soon, it was concluded that some sort fuzzy logic [73] would be needed in order to smooth the transition between the state estimations provided by each motion model, instead of the initial idea of applying each independently depending on the distance, which implied the creation of discrete separated regions separated by a selected distance.
The motivation behind this decision was that applying a motion model for objects up to a certain distance, lets say 30 meters from the sensor, and the other motion model for objects further away, could generate an irregular behavior for objects at, or near, 30 meters (in this example). Here is where the concept of "weighted average" came in. From the internet [74], its definition is the following:

Weighted average is a calculation that takes into account the varying degrees of importance of the numbers in a data set. In calculating a weighted average, each number in the data set is multiplied by a predetermined weight before the final calculation is made.

Since the motion model is used for the prediction and update steps, the combination must occur there. This means that the outcome of those two functions is based on the metioned combination instead of a single model.
The outcome of the prediction function is a bounding box. Therefore, it was decided that the average of the prediction functions from two different motion models would be the average of the predicted bounding boxes' 3D coordinates. This implies that the tracklet state prediction would have to be calculated twice, once with KF and once with CV.
On the other hand, the update function takes in the bounding box and updates the related tracklet information without outputting anything. Therefore, for the update step, two update functions are applied, each to its corresponding motion model.

An example of the code for the "fuzzy" attempt is shown in Appendix A.2, which can be compared with the code in Appendix A.1 to understand more clearly the improvement that was being pursued.
This improved approach to the combination of motion models supposes that one of the models would be prioritized over the other to different degrees depending on the sensor-object distance, but they both would be taken into account everywhere. As mentioned before, both model are utilised for both the prediction and the update step. Combining them, it was theorised before testing this method that the estimation errors produced by each motion model independently could be dampened. This "prioritization" was intended to highlight the impact of each method in those areas where it was believed to be better than the other alternative method. Consequently, the prioritization was tested in many different ways, but the most relevant ones are CV dominant at short distances and KF at greater distances, as well as the opposite.
For this, the area surrounding the sensor was initially divided in many small sub-areas,

code of which can be seen in Appendix A.3, with the intention to obtain a very high degree of smoothness in the result of combining the two models. After carrying out more tests, it was observed that by just dividing the space in three subdivisions was enough to obtain the best results. The distances chosen are the same three used by the WOD data set: 0-30 meters, 30-50 meters and +50 meters. A representation of his definitive version of the code for the weighted average function can be seen in Appendix A.4.

Summing up, the main features of this tracking system are:

- The model-free 3D detector CenterPoint [2] is used to feed the tracking system with bounding boxes
- The preprocessing and association steps are as presented previously, as they are shared by both tracking techniques in this thesis
- The distance used by the weight function was obtained from the ego files from WOD
- The life module used by this first tracking system has the same characteristics as the one used by SimpleTrack. This has been done in this way in order to isolate the possible effects of my novel approach to motion modelling
- The name of this 3DMOT system is **mmTracker**

### 3.5 Life cycle management

This section contains the details that characterize the second MOT system developed for this thesis. It aims to answer the second research question.
For that purpose, first, the output of the association step must be brought back to mind. This consists, as already seen in Subsection 3.3, of three different objects: matched detections, unmatched detections and unmatched tracklets. Once this classification is done, the life module is in charge of determining what to do with the unmatched tracklets and the unmatched detections.

As previously indicated, one of the main consequences of point cloud sparsity and occlusions is the switching of the tracked objects identifier. In order to understand why IDS are relevant, it must be once again reminded what they represent: the number of times that a tracked trajectory changes its matched ground truth identity [25].
MOT systems assign unique identifiers to each target in order to differentiate them. In the best case scenario, when there are no IDS, each target keeps the same identifier until it goes out of reach.
As to be expected, the performance of the tracking system worsens when it has to create unnecessary new identifiers. This occurs because the MOT wrongly "thinks" that there are two different targets when, in reality, both detections correspond to the same target that has been, at some point during the tracking, not correctly associated and therefore misidentified as a new, different object.
As it can be seen in the MOTA equation 10, the system's accuracy is negatively impacted by identity switches. The processing speed is also expected to decrease proportionally to the increment of IDS, since it constitutes an extra load of unnecessary processing.

$$MOTA = 1 - \frac{\Sigma_t(FP_t + FN_t + IDS_t)}{\Sigma_t GT_t} \tag{10}$$

Where FP corresponds to False Positives, which are detections that do not correspond to real objects, FN to False Negatives, which have already been explained, IDS to Identity Switches, which refer to the occasions in which objects have been misidentified, and GT to Ground Truth, which refers to all objects that should be detected (the sum of TP and FN).

There are three scenarios where a tracked object can be granted the wrong label [75]:

- When the object leaves the FOV of the sensor during a certain number of frames and then reappears. This can be caused by either occlusions or simply by the object going further than the sensor's reach and then falling back within the FOV
- The structures formed by moving cloud points that correspond to each object vary strongly
- When a track is associated with the wrong object

Inspired by the simple yet effective principle depicted in [26] regarding tracklet death determination, a more intelligent approach to tracklet birth determination and score management seen in [44] [5], and a more complex view on use of tracklets beyond simply keeping the ones that matched with detections presented in [1], this thesis aims to provide a substantial improvement to this problem, both for vehicles and pedestrians, by taking advantage

of the information already available from the 3D detector about the quality of the detections and simplifying the logic used to determine when to create a new tracklet or kill an existing one. This will provide an answer to the second research question faced by this work:

**How can the negative impact of occlusions and point cloud sparsity on the performance of 3D MOT systems be reduced?**

Interestingly enough, the discoveries made here were made while attempting to apply confidence-based tracklet scoring [44] to the 3D tracking system developed by SimpleTrack [1] in order to reduce the number of IDS they were observing.
Whereas the usual approach, count-based scoring, consists in simply assigning the detection score to the associated tracklet, confidence-based scoring makes use of a more complex logic to determine the score of each tracklet.
On the one hand, a score decay factor is applied to a tracklet for every frame in which it is not associated with a detection, decreasing its score. In this way, tracklets that have not being matched are "punished" by dropping their score by such factor. Once the score of the tracklet drops below a certain "deletion threshold", the tracklet is removed. In this regard, count-based methods just keep the score of the previously associated detection until the tracklet reaches a certain number of frames without having being matched, after which the tracklet is discarded. Different score decay factors were tested in order to obtain the best results, since the refined parameters obtained by [44] could not be assumed to be the best suited for this application, since the rest of the MOT system presented here differs from their system.
On the other hand, in the same way as unmatched tracklets are "punished", matched tracklets are rewarded. [44] proposes different scoring functions that can be applied to those tracklets that are actually successfully associated with detections, increasing their scores for every frame in which they are matched. Among these different functions, the following was selected, as the paper showed its superior performance:

$$trk_{score} = 1 - ((1 - trk_{score}) * (1 - det_{score})) \tag{11}$$

Where $trk_{score}$ is the score of the tracklet and $det_{score}$ is the score of the associated detection.
In this way, confidence based scoring takes advantage of historic information about tracklets and detections, this being the tracklet and detection scores in the previous frame, in order to determine how "trustworthy" a tracklet is and decide whether it should be disregarded or kept alive.

Careful testing of this tracking system was carried out, with several iterations made of different score decay factors, deletion thresholds and maximum ages, as well as different score update functions, in order to find out what was causing what. In doing this, it was found that there was no need for minimum hits, tracklet initialization and tracklet scoring altogether. Therefore, the deletion threshold was also dismissed. Here, minimum hits refers to the number of frames in which a detection must be repeatedly detected before being considered "stable enough" as to start being tracked and therefore initialize a tracklet.
The interesting aspect here is that, apparently, the scores provided by the 3D object detector are already a good enough indicator of their trustworthiness when deciding whether to feed them to the MOT or disregard them. As per the improvement in the performance, it

results now obvious that it is totally unnecessary to wait for a number of frames where these detections are "re-detected" before making this decision. This does not provide an extra layer of security against false positives, whereas removing it must reduce processing time and memory footprint.
The other most meaningful feature of this approach is that, by increasing the maximum age of the unmatched tracklets, this is the number of frames that the tracking system waits for an unmatched tracklet to be associated again, the number of identity switches is directly decreased. As it seems, a high maximum age limit for the tracklets fights back early tracklet termination, which is identified by [1] as one of the main causes of IDS. This coincides with the findings made by [26]. Unlike Immortal Tracker, which never kills an unmatched tracklet, in this document a peak number of frames was identified at which the improvement generated basically reaches a plateau.
The higher the maximum age of tracklets is, the more "hopeful" the tracking system is. The word "hopeful" refers in this case to how much time the MOT system waits for a detection to reappear before giving up and consider it to have disappeared, consequently removing its assigned tracklet.

By combining these two main features, an important improvement on the number of IDS was obtained, while at the same time keeping the rest of metrics from worsening. Summing up, the main features of this tracking system are:

- The model-free 3D detector CenterPoint [2] is used to feed the tracking system with bounding boxes
- The preprocessing and association steps are as presented previously, as they are shared by both tracking techniques in this thesis
- The detection threshold for vehicles is 0.6
- The detection threshold for pedestrians is 0.5
- The motion model used is Kalman Filter, in order to keep the all features outside the scope of this system as common as possible, providing the possibility of a fair comparison with SimpleTrack and other relevant SOTA 3DMOT systems
- The maximum age of vehicles is 15 frames
- The maximum age of pedestrians is 30 frames
- The name of this 3DMOT system is **SimpleLife**

# 4 Results

In this section, the results obtained with both methods presented in section 3 are shown. For this purpose, the data set used and the metrics applied for evaluation are explained. Furthermore, the performance is presented in comparison to other relevant SOTA tracking systems in order to provide useful context that allows to understand what has been achieved with each MOT system.

## 4.1 Data Set and Metrics

The data set utilised to evaluate the 3D MOT systems presented in the previous Section 3 is Waymo Open Dataset [5] (WOD).

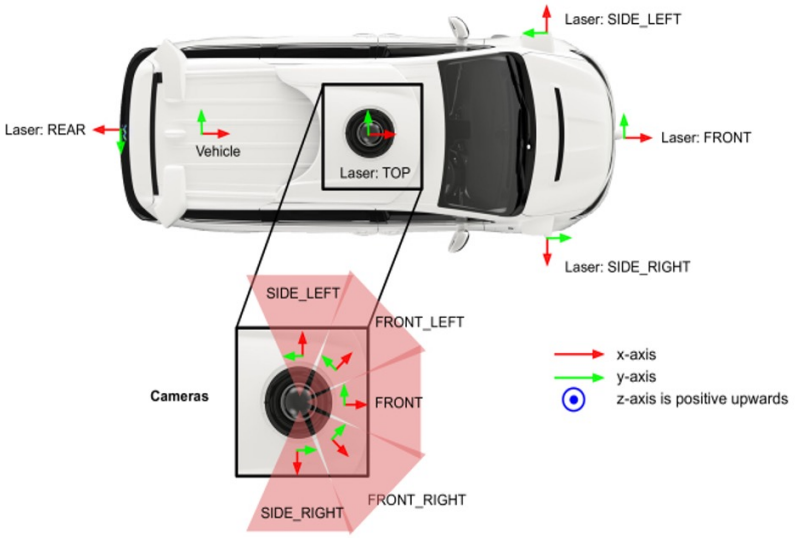



Figure 11: Distribution of the sensors and cameras used for data collection

As already indicated, this data set is meant for the testing and training of systems used in autonomous driving. Therefore, the data it contains is from city roads. This provides a good enough variety of scenarios to confidently test the MOT developed in this thesis, as they consist in general-purpose model-free tracking systems.
WOD has the following features, which is supported by Figure 11:

- Five LiDAR sensors are installed aboard the Waymo car used for the collection of data in the form of point clouds: a mid-range sensor goes on the top of the vehicle, four medium-range sensors go on the sides, the front and the rear
- All five cameras go on the top of the vehicle, covering different degrees of the surrounding area
- The data obtained is classified in vehicles, cyclists and pedestrians, allowing to evaluate tracking systems based on these three different type of objects. The cyclists are excluded from the results and the discussion, as they are not mentioned by most of SOTA 3D MOT systems, rendering the comparison impossible

- It actually consists of two data sets: perception and motion. The perception data set was the one used for the evaluation of the tracking systems in this thesis
- The data set consists of 390000 frames collected at 10Hz in different geographies and atmospheric conditions
- The perception data set is divided into test and training. The test data set was used, as it is smaller in size, making it more manageable, and it is used by all relevant SOTA systems mentioned in Subsection 4.2

The approach to the evaluation of the results obtained for both MOT systems in this document is the following:

- Four different metrics are used: MOTA, Miss, FP and Mismatch
  - MOTA: It gives a measure of the tracking accuracy by taking FP, FN and IDS into account. The equation for it has already been seen in Equation 10. The higher this value is, the better
  - Miss: Also known as False Negatives, or FN. This indicates how many detections are considered to not correspond to a real object when they actually do. The lower this value is, the better
  - FP: False Positives account for detections that should have been disregarded, as they do not correspond to real objects. The lower this value is, the better
  - Mismatch: Also known as Identity Switches, or IDS. This metric refers to the number of times the tracker has misidentified an object. The lower this value is, the better
- Both vehicles and pedestrians were taken into account in the evaluation of the results
- In order to give a useful and fair context of the values obtained, these are compared with relevant state-of-the-art 3D tracking systems: AB3DMOT, CenterPoint, Immortal Tracker and SimpleTrack

## 4.2 Comparison

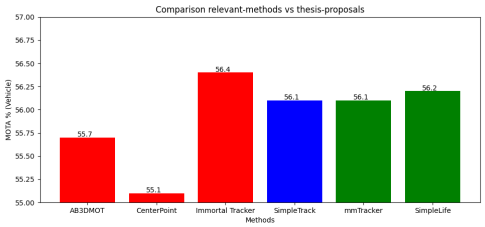
In this subsection, the performance of both tracking systems developed in this thesis is presented in the form of a comparison of key metrics results obtained in this work and in relevant SOTA 3D MOT systems.

In the Table 4, a first view of all results together is shown. In it, the best and second best results are shown in bold text, as well as the tracking methods presented on this document.

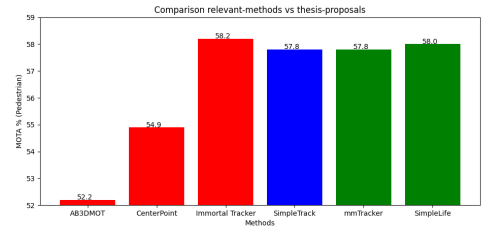
| Method | Vehicles | | | | Pedestrian | | | |
|---|---|---|---|---|---|---|---|---|
| | MOTA % ↑ | Miss % ↓ | IDS % ↓ | FP % ↓ | MOTA % ↑ | Miss % ↓ | IDS % ↓ | FP % ↓ |
| AB3DMOT | 55.7 | - | 0.40 | - | 52.2 | - | 2.74 | - |
| CenterPoint | 55.1 | 33.9 | 0.26 | 10.8 | 54.9 | 34.0 | 1.13 | **10.0** |
| Immortal Tracker | **56.4** | **33.4** | **0.01** | **10.2** | **58.2** | **30.5** | **0.26** | 11.3 |
| SimpleTrack | 56.1 | 33.5 | 0.08 | 10.4 | 57.8 | **30.9** | 0.43 | **10.9** |
| **mmTracker** | 56.1 | **33.4** | 0.08 | 10.4 | 57.8 | **30.9** | 0.42 | **10.9** |
| **SimpleLife** | **56.2** | **33.4** | **0.02** | **10.3** | **58.0** | **30.9** | **0.18** | **10.9** |

Table 4: Metrics comparison between SOTA 3D MOT systems, mmTracker and SimpleLife

In order to provide a closer look into the results obtained, each metric is covered individually, both for vehicles and pedestrians, in the form of coloured histograms. In them, this thesis' tracking systems are shown in green, relevant tracking systems are shown in red and the system used as reference, due to having the most in common, is in blue.
First, MOTA results are shown in Figure 12. In it, it can be seen that, although mmTracker, the tracking system focused on the combination of two motion models, does not provide an improvement over the reference system, SimpleLife does for both vehicles and pedestrians. In the pedestrian case, it almost reaches the best MOTA value seen in model-free 3D MOT.



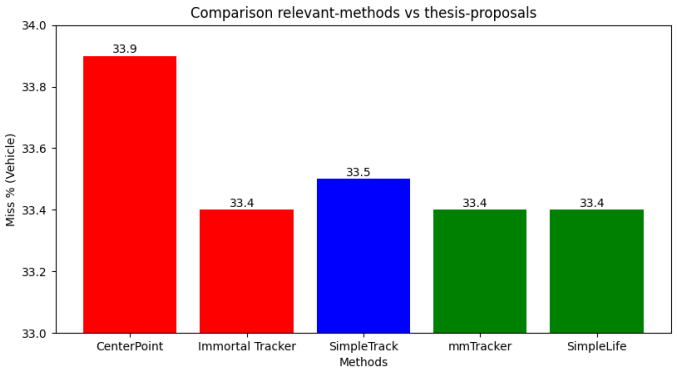

(a) Vehicles



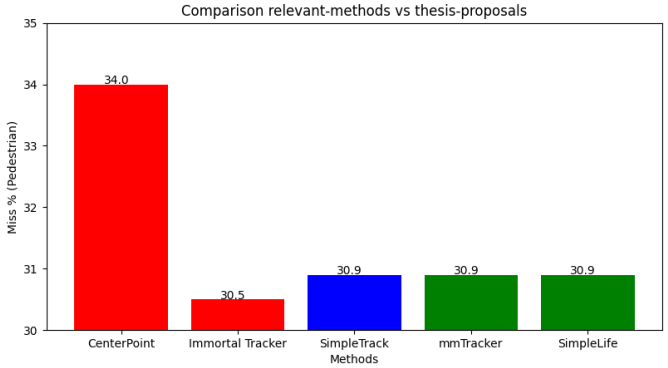

(b) Pedestrians

Figure 12: MOTA results

The Miss results, or false negatives, are shown in Figure 13. This figure indicates that no improvement, also no worsening, was generated in the case of pedestrian tracking. On the other hand, both mmTracker and SimpleLife produce an improvement in the tracking of vehicles in comparison with SimpleTrack [1], reaching the best values on this metric by a model-free 3D MOT method, shown by Immortal Tracker [26].



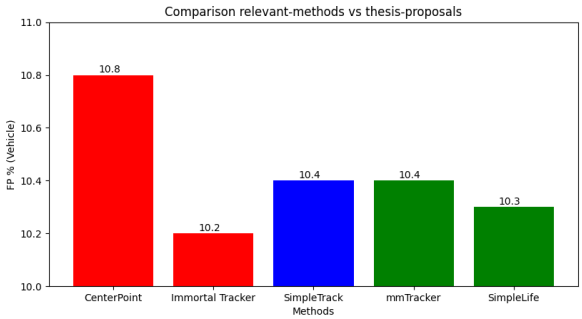

(a) Vehicles



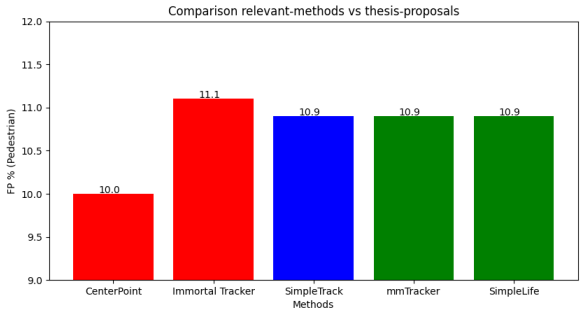

(b) Pedestrians

Figure 13: Miss results

The results obtained regarding False Positives, presented in Figure 14, indicate that only an improvement was obtained by SimpleLife for the case of vehicle tracking. For the rest, neither improvement nor worsening was produced.



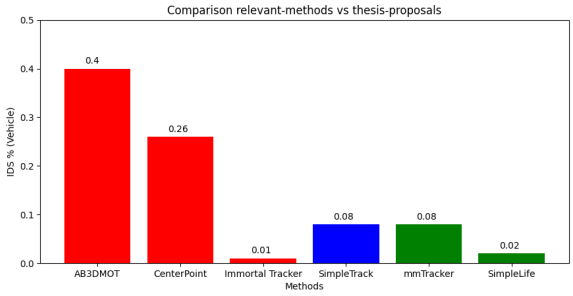

(a) Vehicles



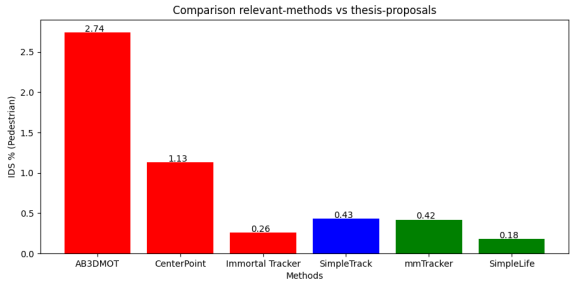

(b) Pedestrians

Figure 14: FP results

Finally, the identity switches, or mismatch, generated by each method can be seen in Figure 15. This shows the most exciting results. Whereas the IDS in the case of mmTracker were not altered, those obtained by SimpleLife improved greatly. On the one side, IDS for vehicles improved by a 75% with respect to the reference MOT system, almost reaching the best value, that of Immortal Tracker [26]. On the other side, IDS for pedestrians improved by 58% with respect to the reference MOT system, obtaining the best, to the the author's knowledge, IDS value for pedestrians that a model-free 3D MOT system has reached up to date.




(a) Vehicles




(b) Pedestrians

Figure 15: IDS results

## 4.3 Discussion

In order to discuss the results presented in Section 4, a reminder of the research questions is deemed appropriate.
The question for which the 3D MOT system mmTracker was developed is the following:

**Does the distance-dependent combination of two of the most common motion models affect, in a significant manner, the overall performance of a 3D MOT system?**

On the other hand, the question for which the 3D MOT system SimpleLife was developed is the following:

**How can the negative impact of occlusions and point cloud sparsity on the performance of 3D MOT systems be reduced?**

Based on the results obtained from the tests carried while developing this thesis, answers to both research questions have been obtained, considering in consequence successfully terminated the enterprise started with the writing of this thesis.

Regarding the use of distance-dependent motion modelling, it was found that there is a negligible effect on the overall performance compared to the usual single-model approach. From this, it is understood that motion-model combination based on sensor-object distance does not improve, or worsen, the performance of the object tracker in a meaningful way. Nevertheless, it does increase the complexity and footprint of the task.
Regarding the second research question, it was discovered that the negative effects of occlusions and point cloud sparsity can be decreased in a significant manner by simplifying how tracklets are managed. Moreover, simpler life management policies generate an improvement in tracking accuracy, FN, FP and IDS in the case of vehicles and it provides the lowest number of identity switches ever seen for pedestrians, at least among model-free multi-object tracking systems using 3D LiDAR.

# 5 Conclusion

In the process of creating this master's thesis document, several conclusions have been reached. Furthermore, a wide variety of tools and knowledge previously alien to the author have been understood and acquired.

First and foremost, the literature research, for which a considerable amount of time was dedicated, laid the foundation for the creation of this work. The utter importance of this phase when developing a document of this magnitude could be considered one of the important conclusions generated by its creation.

Initially, understanding the advantages and disadvantages of using a model-free instead of a model-based approach to 3D multi-object tracking made clear that, to develop a different and somehow new approach to this problem that could be generalised into different contexts, the model-free alternative was to be the one this thesis should be focused on. The lack of previous knowledge about the environment and the objects in it, with which the sensor would have to interact, constitutes the exact prerequisite of a system that can be considered to be of "general purpose". This in itself is another key lesson drawn from this thesis.

The work of Ziqi Pang et al. with SimpleTrack [1] has been central in the development of this document. It has not only inspired some of the decisions taken regarding what characteristics some of the different elements of the tracking systems should have, but also has provided a very clear view of how a 3D model-free multi-object tracking system that uses 3D LiDAR data works and what its fundamental building-blocks are. It has proven extremely helpful to approach 3D MOT as a four-block problem, understanding that modifications within single blocks can affect the overall performance greatly.

Before going into some direct conclusions drawn from the tracking algorithms themselves, it is worth mention another lesson learnt by the author. This is, the deep impact that open data sources like Waymo Open Dataset, nuScenes or KITTI have on facilitating research like the one done with the present document. This also goes for the use of supercomputers, the Finnish Puhti supercomputer in this case, which sped up the processing of data to an astounding degree. It is no exaggeration to say that it has cut in half the time that the large amount of tests carried for this thesis would have required. Therefore, a byproduct of generating this document has been the learning of how to deal with public supercomputers: how to connect a personal computer to a public supercomputer in order to upload important amounts of data, as well as the code used to process it; how to prepare scripts that tell the supercomputer the program to be used, the data on which the programs is to be applied, and what number of CPUs and nodes are needed, together with the amount of memory and time expected for the uploaded tracking code to the raw point cloud data. This was a quite challenging but enriching experience.

Regarding the detection side of any DATMO system, it was found that center-based detection is one the best starting points for a tracking system using “tracking-by-detection", where the tracking performance is directly influenced by the performance of the detector. Therefore, the use of a good detector is of paramount importance if the tracking system is to show its best performance improvement capabilities compared to previous versions of somehow similar tracking methods. In this sense, the use of the CenterPoint [2]

has played a central role in obtaining some of the positive results covered in previous sections.

The development of the mmTracker contributed greatly to the author's understanding of the big picture of multiple object tracking, mostly with the use of 3D LiDAR for the obtention of the raw data. A clear view was obtained of the different basic elements that constitute a tracking algorithm, together with all the different challenges faced by this problem. Moreover, seemingly clear signs of distance-dependent performance were found in different aspects of this algorithm, choosing to focus in this case on the motion modelling. With the mmTracker system presented, many different weight distributions of the motion models were tested, attaining stable results only for a few of them. From these experiments it is concluded that this line of research is not worth following, since the tracking system was made more complex without providing any worthwhile benefit. Nevertheless, it is very well possible that the approach applied to test this distance-dependent theory was not the appropriate one, consequently implying that this theory should not yet be completely disregarded as of no value, but instead further investigated in search of a different way of applying it.
On the other hand, the second tracking method, SimpleLife, demonstrated once again that some discoveries occur in the course of an investigation that did not have such intention in the first place. Although the end result, improving the performance of a multiple object tracking system by improving the life cycle management building block, was maintained, the means to achieve it turned out to be different from the initially intended. The approach with which the experiments started was focused of the refinement of tracklet scoring. Interestingly enough, it was found that tracklet scoring is unnecessary, at least in the presented application of it and with the purpose of improving the life management module. Another unexpected finding is that the requirement of minimum hits to generate a track, while at the same time having detections that posses confidence scores, is pointless, at least in the shown application of it. Finally, the SimpleLife tracking system developed here shows that increasing the time tracklets can survive without being associated reduces identity switches, which is consistent with the Immortal Tracker paper [26], although no reasonable justification for keeping them alive indefinitely was found.

# 6 Future Work

This section is meant to provide some sort of guidance and light regarding knowledge gaps that were found while creating this thesis, which could be worthwhile to further investigate.

Due to the natural time restrictions of any master's thesis, this work only made use of one, although widely used in the autonomous vehicles community, data set, the Waymo Open Dataset. It would be interesting to test the methods developed here on other datasets, like nuScenes or KITTI, which would provide a wider context and a better understanding of the tracking systems developed, as well as more robust results.

It could also be interesting to consider applying the mmTracker logic to the association metrics, as they also seem to show some degree of distance-dependent behavior.

Lastly, it is deemed almost certain that combining SimpleLife's logic with model-based tracking methods would also show positive results. Therefore, this is a recommended path for those looking for potential improvements to be made on that side of the 3D tracking approaches.

# References


[1] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," *arXiv preprint arXiv:2111.09621*, 2021.

[2] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[3] T. Hoffmann and G. Prause, "On the regulatory framework for last-mile delivery robots," *Machines*, vol. 6, no. 3, p. 33, 2018.

[4] J. M. Garcia-Haro, E. D. Oña, J. Hernandez-Vicen, S. Martinez, and C. Balaguer, "Service robots in catering applications: A review and future challenges," *Electronics*, vol. 10, no. 1, p. 47, 2020.

[5] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[6] P. Morton, B. Douillard, and J. Underwood, "An evaluation of dynamic object tracking with 3d lidar," in *Proc. of the Australasian Conference on Robotics & Automation (ACRA)*, 2011.

[7] R. C. Watson Jr, *Radar origins worldwide: history of its evolution in 13 nations through World War II*. Trafford Publishing, 2009.

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

[9] D. Y. Kim and M. Jeon, "Data fusion of radar and image measurements for multi-object tracking via kalman filtering," *Information Sciences*, vol. 278, pp. 641–652, 2014.

[10] R. Mobus and U. Kolbe, "Multi-target multi-object tracking, sensor fusion of radar and infrared," in *IEEE Intelligent Vehicles Symposium, 2004.* IEEE, 2004, pp. 732–737.

[11] S. Kim, S.-Y. Oh, J. Kang, Y. Ryu, K. Kim, S.-C. Park, and K. Park, "Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2005, pp. 2173–2178.

[12] T. Kim and T.-H. Park, "Extended kalman filter (ekf) design for vehicle position tracking using reliability function of radar and lidar," *Sensors*, vol. 20, no. 15, p. 4126, 2020.

[13] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proceedings 2007 IEEE international conference on robotics and automation.* IEEE, 2007, pp. 3402–3407.

[14] C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppe, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe *et al.*, "Moving object detection with laser scanners," *Journal of Field Robotics*, vol. 30, no. 1, pp. 17–43, 2013.

[15] M. Darms, P. Rybski, and C. Urmson, "Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments," in *2008 IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 1197–1202.

[16] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[17] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2, pp. 123–139, 2009.

[18] L. E. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian detection and tracking using three-dimensional ladar data," *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1516–1528, 2010.

[19] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3d lidar point clouds," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2798–2805.

[20] M. Himmelsbach, A. Mueller, T. Lüttel, and H.-J. Wünsche, "Lidar-based 3d object perception," in *Proceedings of 1st international workshop on cognition for technical systems*, vol. 1, 2008.

[21] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[22] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[23] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Lyu, W. Ke, and Z. Xiong, "Long-term tracking with deep tracklet association," *IEEE Transactions on Image Processing*, vol. 29, pp. 6694–6706, 2020.

[24] F. Moosmann and C. Stiller, "Joint self-localization and tracking of generic objects in 3d range data," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1146–1152.

[25] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104 423–104 434, 2019.

[26] Q. Wang, Y. Chen, Z. Pang, N. Wang, and Z. Zhang, "Immortal tracker: Tracklet never dies," *arXiv preprint arXiv:2111.13672*, 2021.

[27] J. Lin, S. Li, W. Dong, T. Matsumaru, and S. Xie, "Long-arm three-dimensional lidar for antiocclusion and antisparsity point clouds," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[28] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3d lidar scans," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 4508–4513.

[29] Z. Pang, Z. Li, and N. Wang, "Model-free vehicle tracking and state estimation in point cloud sequences," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8075–8082.

[30] P. Steinemann, J. Klappstein, J. Dickmann, F. von Hundelshausen, and H.-J. Wünsche, "Geometric-model-free tracking of extended targets using 3d lidar measurements," in *Laser Radar Technology and Applications XVII*, vol. 8379. SPIE, 2012, pp. 104–115.

[31] J. D. Yoon, "Model-free setting-independent detection of dynamic objects in 3d lidar," Ph.D. dissertation, University of Toronto (Canada), 2019.

[32] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *2009 IEEE Intelligent Vehicles Symposium*. IEEE, 2009, pp. 215–220.

[33] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2d lidar," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1039–1063, 2015.

[34] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla, "Robust method for removing dynamic objects from point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 765–10 771.

[35] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[36] T.-D. Vu and O. Aycard, "Laser-based detection and tracking moving objects using data-driven markov chain monte carlo," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3800–3806.

[37] S. Giancola, J. Zarzar, and B. Ghanem, "Leveraging shape completion for 3d siamese tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1359–1368.

[38] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3d laser data," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008, pp. 4043–4048.

[39] P. Ondruska, J. Dequaire, D. Z. Wang, and I. Posner, "End-to-end tracking and semantic segmentation using recurrent neural networks," *arXiv preprint arXiv:1604.05091*, 2016.

[40] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[41] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[42] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3d lidar," in *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE, 2019, pp. 113–120.

[43] X. Weng and K. Kitani, "A baseline for 3d multi-object tracking," *arXiv preprint arXiv:1907.03961*, vol. 1, no. 2, p. 6, 2019.

[44] N. Benbarka, J. Schröder, and A. Zell, "Score refinement for confidence-based 3d multi-object tracking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8083–8090.

[45] J. Ngiam, B. Caine, W. Han, B. Yang, Y. Chai, P. Sun, Y. Zhou, X. Yi, O. Alsharif, P. Nguyen *et al.*, "Starnet: Targeted computation for object detection in point clouds," *arXiv preprint arXiv:1908.11069*, 2019.

[46] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu, "Range conditioned dilated convolutions for scale invariant 3d object detection," *arXiv preprint arXiv:2005.09927*, 2020.

[47] P. Hu, J. Ziglar, D. Held, and D. Ramanan, "What you see is what you get: Exploiting visibility for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 001–11 009.

[48] Q. Chen, L. Sun, E. Cheung, and A. L. Yuille, "Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 224–21 235, 2020.

[49] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.

[50] J. Yin, J. Shen, C. Guan, D. Zhou, and R. Yang, "Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 495–11 504.

[51] X. Zhu, Y. Ma, T. Wang, Y. Xu, J. Shi, and D. Lin, "Ssn: Shape signature networks for multi-class object detection from point clouds," in *European Conference on Computer Vision*. Springer, 2020, pp. 581–597.

[52] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[53] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[54] M. Najibi, G. Lai, A. Kundu, Z. Lu, V. Rathod, T. Funkhouser, C. Pantofaru, D. Ross, L. S. Davis, and A. Fathi, "Dops: Learning to detect 3d objects and predict their 3d shapes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 913–11 922.

[55] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*. PMLR, 2020, pp. 923–932.

[56] R. Huang, W. Zhang, A. Kundu, C. Pantofaru, D. A. Ross, T. Funkhouser, and A. Fathi, "An lstm approach to temporal 3d object detection in lidar point clouds," in *European Conference on Computer Vision*. Springer, 2020, pp. 266–282.

[57] Z. Ding, Y. Hu, R. Ge, L. Huang, S. Chen, Y. Wang, and J. Liao, "1st place solution for waymo open dataset challenge–3d detection and domain adaptation," *arXiv preprint arXiv:2006.15505*, 2020.

[58] R. Ge, Z. Ding, Y. Hu, Y. Wang, S. Chen, L. Huang, and Y. Li, "Afdet: Anchor free one stage 3d object detection," *arXiv preprint arXiv:2006.12671*, 2020.

[59] Y. Wang, A. Fathi, A. Kundu, D. A. Ross, C. Pantofaru, T. Funkhouser, and J. Solomon, "Pillar-based object detection for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020, pp. 18–34.

[60] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[61] S. Mandal, S. M. B. Mones, A. Das, V. E. Balas, R. N. Shaw, and A. Ghosh, "Chapter four - single shot detection for detecting real-time flying objects for unmanned aerial vehicle," in *Artificial Intelligence for Future Generation Robotics*, R. N. Shaw, A. Ghosh, V. E. Balas, and M. Bianchini, Eds. Elsevier, 2021, pp. 37–53. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323854986000058

[62] G.-H. Fu, L.-Z. Yi, and J. Pan, "Tuning model parameters in class-imbalanced learning with precision-recall curve," *Biometrical Journal*, vol. 61, no. 3, pp. 652–664, 2019.

[63] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 850–855.

[64] H. Shen, L. Huang, C. Huang, and W. Xu, "Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking," *arXiv preprint arXiv:1808.01562*, 2018.

[65] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3d multi-object tracking for autonomous driving," *arXiv preprint arXiv:2001.05673*, 2020.

[66] Vineeth S Subramanyam, "Iou (intersection over union)," accessed June 6, 2022. [Online]. Available: https://bit.ly/3zF1Pen

[67] P. C. Mahalanobis, "On the generalized distance in statistics." National Institute of Science of India, 1936.

[68] L. Rüschendorf and S. T. Rachev, "A characterization of random variables with minimum l2-distance," *Journal of multivariate analysis*, vol. 32, no. 1, pp. 48–54, 1990.

[69] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[70] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[71] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[72] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[73] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[74] Akhilesh Ganti, "Weighted average," accessed June 15, 2022. [Online]. Available: https://bit.ly/3xSOAp3

[75] Y. Ma, J. Anderson, S. Crouch, and J. Shan, "Moving object detection and tracking with doppler lidar," *Remote Sensing*, vol. 11, no. 10, p. 1154, 2019.

# A Motion Model combination code

## A.1 Non-fuzzy

```
if 0 <= d < 30:
    result.x = motion_model_A.x
    result.y = motion_model_A.y
    result.z = motion_model_A.z
else:
    result.x = motion_model_B.x
    result.y = motion_model_B.y
    result.z = motion_model_B.z
```

## A.2 Fuzzy

```
if 0 <= d < 30:
    result.x = motion_model_A.x * 0.7 + motion_model_B.x * 0.3
    result.y = motion_model_A.y * 0.7 + motion_model_B.y * 0.3
    result.z = motion_model_A.z * 0.7 + motion_model_B.z * 0.3
else:
    result.x = motion_model_B.x * 0.3 + motion_model_B.x * 0.7
    result.y = motion_model_B.y * 0.3 + motion_model_B.y * 0.7
    result.z = motion_model_B.z * 0.3 + motion_model_B.z * 0.7
```

## A.3 Fuzzy - highly smooth

```
if 0 <= d < 5:
    result.x = motion_model_A.x * 0.9 + motion_model_B.x * 0.1
    result.y = motion_model_A.y * 0.9 + motion_model_B.y * 0.1
    result.z = motion_model_A.z * 0.9 + motion_model_B.z * 0.1
if 5 <= d < 10:
    result.x = motion_model_A.x * 0.8 + motion_model_B.x * 0.2
    result.y = motion_model_A.y * 0.8 + motion_model_B.y * 0.2
    result.z = motion_model_A.z * 0.8 + motion_model_B.z * 0.2
if 10 <= d < 15:
    result.x = motion_model_A.x * 0.7 + motion_model_B.x * 0.3
    result.y = motion_model_A.y * 0.7 + motion_model_B.y * 0.3
    result.z = motion_model_A.z * 0.7 + motion_model_B.z * 0.3
if 15 <= d < 20:
    result.x = motion_model_A.x * 0.6 + motion_model_B.x * 0.4
    result.y = motion_model_A.y * 0.6 + motion_model_B.y * 0.4
    result.z = motion_model_A.z * 0.6 + motion_model_B.z * 0.4
if 25 <= d < 30:
    result.x = motion_model_A.x * 0.5 + motion_model_B.x * 0.5
    result.y = motion_model_A.y * 0.5 + motion_model_B.y * 0.5
    result.z = motion_model_A.z * 0.5 + motion_model_B.z * 0.5
if 30 <= d < 35:
    result.x = motion_model_A.x * 0.4 + motion_model_B.x * 0.6
    result.y = motion_model_A.y * 0.4 + motion_model_B.y * 0.6
    result.z = motion_model_A.z * 0.4 + motion_model_B.z * 0.6
```

```
if 35 <= d < 40:
    result.x = motion_model_A.x * 0.3 + motion_model_B.x * 0.7
    result.y = motion_model_A.y * 0.3 + motion_model_B.y * 0.7
    result.z = motion_model_A.z * 0.3 + motion_model_B.z * 0.7
if 40 <= d < 45:
    result.x = motion_model_A.x * 0.2 + motion_model_B.x * 0.8
    result.y = motion_model_A.y * 0.2 + motion_model_B.y * 0.8
    result.z = motion_model_A.z * 0.2 + motion_model_B.z * 0.8
else:
    result.x = motion_model_B.x * 0.1 + motion_model_B.x * 0.9
    result.y = motion_model_B.y * 0.1 + motion_model_B.y * 0.9
    result.z = motion_model_B.z * 0.1 + motion_model_B.z * 0.9
```

## A.4 Fuzzy - smooth

```
if 0 <= d < 30:
    result.x = motion_model_A.x * 0.7 + motion_model_B.x * 0.3
    result.y = motion_model_A.y * 0.7 + motion_model_B.y * 0.3
    result.z = motion_model_A.z * 0.7 + motion_model_B.z * 0.3
if 30 <= d < 50:
    result.x = motion_model_A.x * 0.5 + motion_model_B.x * 0.5
    result.y = motion_model_A.y * 0.5 + motion_model_B.y * 0.5
    result.z = motion_model_A.z * 0.5 + motion_model_B.z * 0.5
else:
    result.x = motion_model_B.x * 0.3 + motion_model_B.x * 0.7
    result.y = motion_model_B.y * 0.3 + motion_model_B.y * 0.7
    result.z = motion_model_B.z * 0.3 + motion_model_B.z * 0.7
```