Title

Elina Trier Brigisson
Technical Faculty of Electronics and IT
Aalborg University
Aalborg, Denmark
mail@student.aau.dk

Anna Maria Maj
Technical Faculty of Electronics and IT
Aalborg University
Aalborg, Denmark
amaj17@student.aau.dk

Daria Oskina
Technical Electronics and IT
Aalborg University
Aalborg, Denmark
doskin20@student.aau.dk

***Abstract*—**
***Index Terms*—**

I. Introduction

Write about Veovo.
An airport is a place that is very critical in terms of time and punctuality therefor efficient planning and processes are essential. The use of computer vision can be a good tool to optimise bottleneck areas like counters and queues. It would also allow to plan and organise a smooth workflow by enabling the staff to evaluate the movements of the airports guests in real-time and giving them an overview of the situation. Taking quick decisions like opening of another counter can eliminate delays and the distribution of people waiting in a queue. To be able to retain hygiene and safety standards, it is of high priority to make it possible for people to keep a safe distance from each other and avoid accumulations of crowds in limited areas.

Additional information gained through long-term data acquisition of common movement patterns can be very insightful and allow to eliminate obstacles and optimise paths for more convenience. //Add reference To be able to gain such knowledge it is necessary to continuously monitor the proceedings in certain areas of the airport. Veovo has been working with classic cameras capturing two-dimensional image data from a real-time environment which then was analysed and human features were detected and tracked. Finally, the system tracked and identified people. This processes can potentially be improved by implementing the same functionality with Light Detection and Ranging (LiDAR).
The increasing use of 3D LiDAR sensors in various fields like autonomous driving or robotics shows that the interest in the LiDAR technology is growing and that research has started picking up pace and investigate on how to use the technologies advantages in certain applications. Since vehicles equipped with LiDAR for automated driving functions were released in 2017, public interest and production have increased, which on the other hand decreased their manufacturing cost and improved the robustness of the sensors [1].
In comparison to data captured in 2D the comparatively wide area covered by a 3D LiDAR scan can be used to enlargen the field of view and thus the area of detection. Additionally LiDAR technology provides an increase in ranging accuracy since the information density is higher in three-dimensional data [1], [2].

II. Methods

A. LiDAR

Light Detection and Ranging (LiDAR) systems are able to measure their environment very precisely by scanning their surroundings with a laser and capture the reflected light with a sensor.
Li et al. [1] describe the role of LiDAR in perception and localisation systems and divide such a systems' output into 3 different types of information:

- The physical description containing the pose, velocity and shape of an object
- The semantic description containing the categories of the different objects
- An intention prediction which describes the likelihood of an objects behaviour

This kind of information makes it possible to apply LiDAR for object detection tasks, classification, tracking and intention prediction.
Consisting of a laser rangefinder and a scanning system (beam steering system) a LiDAR can calculate the distance of an object with signal processing electronics and principles [1].

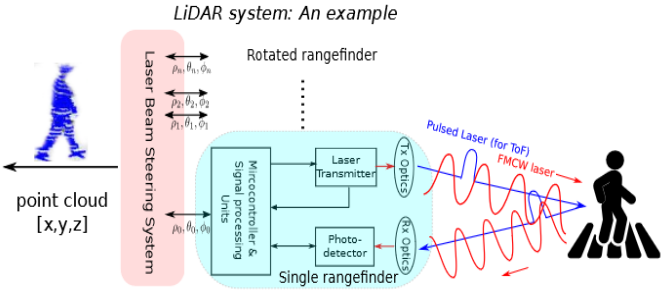


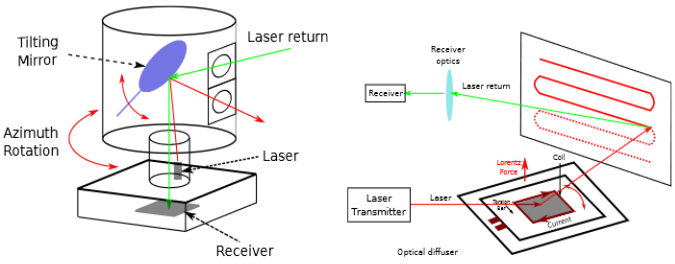
Fig. 1. A LiDAR human detection system [1].

There are three scanning techniques:

- A more often used "mechanical spinning" scanner:
  It can provide a 360° Field of View but has the disadvantage of many moving parts, it is less convenient for

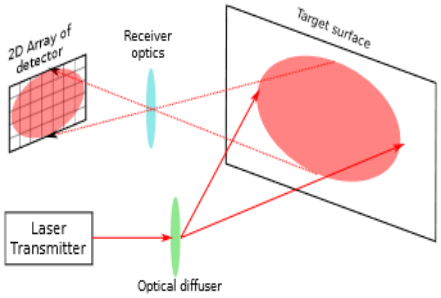
integration as it is rather huge and is prone to vibrations at its mount [1].

- There are "in-between-type" scanners like Micro-Electro-Mechanical Systems (MEMS) based LiDAR which is a "near-solid-state" technology containing some moving part [1].
- Truly "solid state" scanners like 3D flash LiDARs scan a single scene at once but have a limited detection range of below 100 m due to the small power threshold of the laser that is applied for eye safety reasons. They also have a limited field of view since they have a distinct scanning direction [1].



(a) Principle of mechanical spinning LiDAR (b) Principle of MEMS LiDAR

Fig. 2. Mechanical spinning LiDAR and Micro-Electro-Mechanical LiDAR [1].



(c) Principle of flash LiDAR

Fig. 3. Solid state flash LiDAR [1].

B. Representation of 3D Data

The raw data given by a LiDAR consists of a 3D point cloud at a frame, representing the environment as a collection of three-dimensional points with attached information like color, distance, location and the dedicated intensities of the lasers' reflection, which are indicating the properties of surface materials. [1]

To achieve a scan of a certain desirable precision the parameters Resolution, Quality and Colour have to be considered. [3]

C. Challenges in the Processing of Point clouds

Due to the huge amount of information and the memory requirements the processing of point cloud data sets is challenging.

Since point clouds are the output of many 3d scanning devices and are a convenient way to represent 3D data, the challenge is to know how to process them properly to make use of them later for training and testing a network.

The main difficulty in processing data captured with depth sensors, like LiDAR, is the difference in dimensionality of the captured data compared to regular 2D image processing. According to Guo et al. [4] most research in Machine learning and Computer Vision is focusing on improving the performance of algorithms working with 2-dimensional data. Whereas 2D images can be represented as pixel arrays and conveniently read and processed - working with 3D data is different. Three dimensional data can be represented in multiple ways like e.g with volumetric pixel grids, polygonal meshes and faces or point clouds. This leads to different file formats like (.off, .pcd, ...) and displays the difficulty of the fact that there is no international standard in the data acquisition with LiDAR [1].

Nevertheless point clouds provide the original 3D information without discretizing the data and this format is widely used in the field.

There are some other constraints on point clouds: Since they have no order, reading point clouds must not be prone to permutations.

A different problem of an object that is captured in 3D is that it can be captured from multiple angles - thus a network processing point clouds has to be invariant to rigid transformations.

Also capturing the interaction between points and other neighbouring points has to be taken into account during processing of the data, which can be challenging.

Specifically for human detection there are other challenges like the varying forms of LiDAR Scans of the human shape, depending on the distance from the sensor as shown in figure 4.

The limited vertical angular resolution of LiDAR sensors

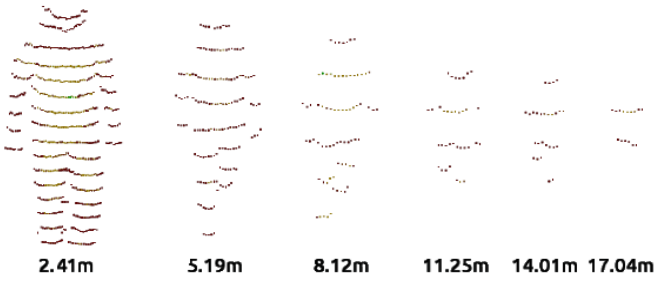


Fig. 4. A human 3D LiDAR Scan from different distances [2]

causes the vertical distance between a series of horizontal points to differ, even though they reflect the same object. This makes the choosing of a fitting distance threshold more difficult and correct cluster extraction more complicated [2], [5], [6].

Another weak spot of three-dimensional detection is the relatively small amount of datasets, in contrast to a large amount of labelled data sets for two-dimensional object detection -

and even smaller for indoor environments [6].

D. Datasets

Guo et al. [6] provide a good overview of existing datasets for 3D data. In general data sets can be either synthetically generated or consist of recordings of real-world environments. Those that can be used for 3D object detection and tracking can be divided into two categories: indoor environments and outdoor environments. For 3D object detection only a handful of edited datasets exist that are appropriate for training a network. Listed datasets for indoor environments are for example ScanNet and Sun RGB-D but since these datasets are in RGB-D format they are not fitting for training for an eventual LiDAR point cloud input.
Data sets acquired with LiDAR are focusing on urban outdoor environments, as most of them are specifically made for autonomous driving and focus on the detection of objects in an outdoor traffic situation [2]. Even though there are some publicly available data sets, not many are focusing on indoor environments since research on LiDAR is mostly used to improve autonomous driving in urban outdoor environments. However, there are some labelled data sets focusing on human detection in indoor environments which are coming from the area of robotics. Three data sets that were made available by the researchers are L-CAS and InLiDA. After extensive researching and comparison of the details, it was decided that the dataset that would fit this project's needs best would be the L-CAS dataset.

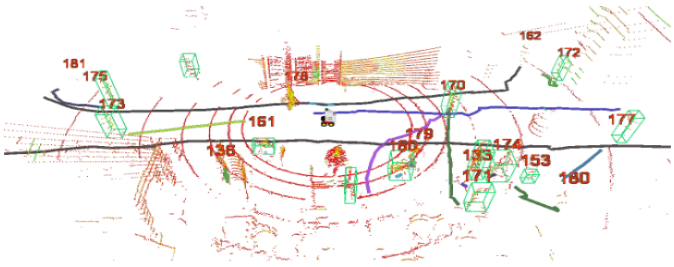


Fig. 5. Overview of the L-CAS data set with the LiDAR mounted on a robot found at the center and classified objects [2].

Specifically configured for human-robot interaction the L-CAS dataset provides a basis for detection and tracking of humans in an indoor environment. In contrast to other datasets, an increased amount of labels can be found in the L-CAS dataset, as it contains the categorisation of groups and crowds of people, small children, sitting people, people walking up stairs and people with trolleys and luggage, seen in figure 6, which fits the goal of human detection in the scenario of an airport very well.

The L-Cas Dataset was recorded with a Velodyne VLP-16 3D LiDAR in a publically accessible building. (16 scan channels, 360° horizontal, 30°vertical FoV, mounting height: 0.8m, LiDAR rotating at 10Hz, maximum scan range 100m,

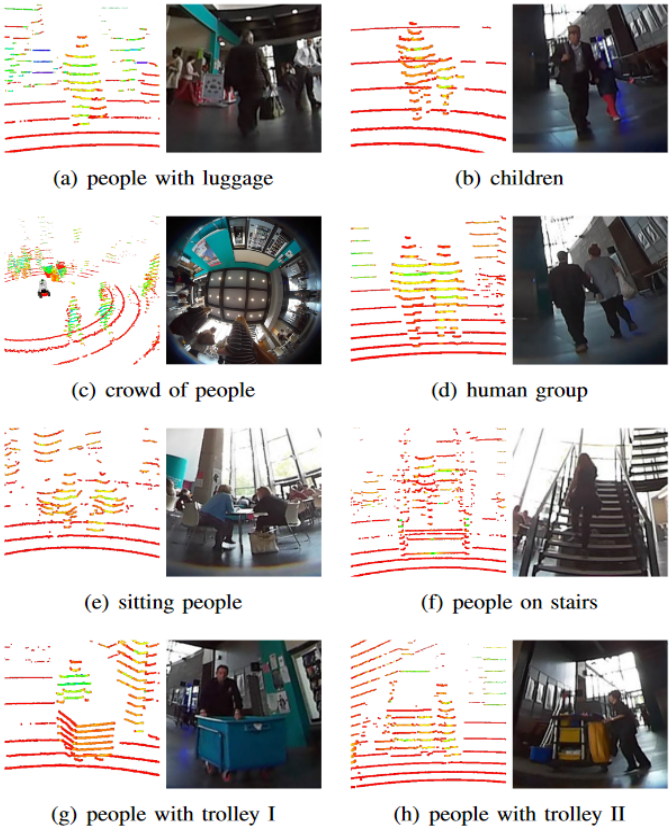


Fig. 6. Different classifications withing the L-CAS data set [2].

28,002 scan frames (less when stationary), 30,000 3D points per frame, recording of odometry, coordinate transformation, panoramic image surrounding robot also recorded = ground-truth, about 20% manually annotated, transformation between coordinate frames with ROS tf package) [2].

E. Processing of LiDAR

The processing of LiDAR data at a basic point cloud level can be done in multiple ways. The subdivision of the point cloud data into parts that bear information can be achieved by a row of different methods like Point Cloud Semantic Segmentation (PCSS), in which the point cloud is divided into clusters that supposedly belong to one object [7].
Segmentation can be done at multiple levels, like on a global scene level, an object level or only partially [6].

Two segmentation methods that are often used are discrete segmentation and point-based segmentation.
Point-based approaches work on raw point clouds. This approach avoids information being lost during projections or conversions, like it is the case with discrete segmentation.
Discrete methods use discrete transformations on the point-cloud input and even though they have to face information loss, they are efficient.
A form of discrete segmentation is a grid-based approach, where the scan is divided into polar grids and the grid cells are being classified as free, occupied and occluded, indicating

the number of points in them. Consecutively occupied cells are grouped into object clusters. A downside of discrete segmentation is that along the way raw information from the LiDAR measurements is lost, which can be a problem for object detection of objects that are far away [1], [6].
Depending on the angle and the quality of the scan the points in an object are unevenly distributed which makes classification more difficult. A workaround is to uniformly sample points on an object's surface.
Sample Data in an uniformly distributed way (assigning probability of a point proportionally to an area) Since networks usually have dense layers, a fixed number of points in a point cloud is required (sample one point per chosen face).
Depending on the distance from the LiDAR scanner the height of a person and its position have to be distinguishable. To unitise those differences for the network the object is translated to the origin (subtract mean from all its points) and normalized into a unit sphere [1].
Processing of LiDAR data in spherical coordinates has the advantage of naturally creating a range image sphere and can increase processing speed [1], [6].

F. Deep Learning with LiDAR

As Guo et al. [6] show in their work, there are many different networks specified for different purposes.

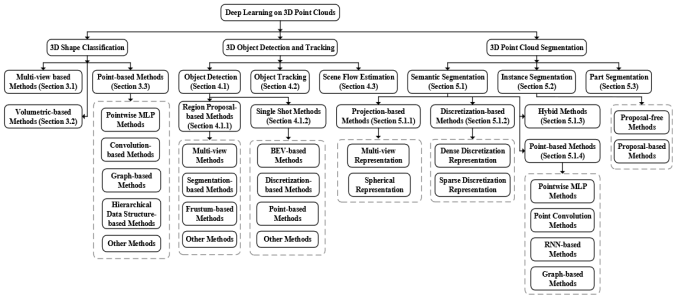


Fig. 7. Deep learning tasks with 3D data and differnt methods [6]

As can be seen in figure 7, three main purposes are defined: Shape Classification, Point Cloud Segmentation and object detection and tracking. For the final goal of recognizing objects and predicting their movements, methods for these three classes of processing LiDAR data are often interacting or building on top of each other inside of a network.
Object recognition techniques based on machine learning often consist of two consecutive parts: Firstly feature extraction and secondly classification [1].

1) Feature Extraction: During feature extraction object descriptors are calculated. Features can be global and apply to a whole object like the size, the maximum intensity etc., or they can be local and apply for each point. Features that are often used are surfaceness, linearness and scatterness.
A natural limitation to the complexity of the extracted features is the real-time data acquisition.

2) Classification: During classification the prediction of a category based on the previously extracted features is done by a classifier. Often classifiers are pre-trained by a ground truth data set.
For classification many supervised machine learning algorithms are applicable like Naive Bayes, Support Vector Machines, KNN, Random Forest, Gradient Boostin Tree, and evidential classifiers that can handle unknown classes [1].

For 3D shape classification there exist three different approaches: Multi-view-based, volumetric and point-based.
Multi-view-based methods project 3D point clouds into multiple views to find specific features for each view. Those features are then fused to give global features and create a global shape descriptor. Although Multi-view-based approaches are effective, they are also computationally expensive and slow [6].
In volumetric-based methods 3D point clouds are converted into a geometrical 3D representation like a Voxel or a Pillar. Then 2D/3D CNNs are applied on those representations for shape classification.
With a point-based approach the feature learning happens for each point. Pointwise features are learned which are then combined in a symmetric aggregation funcion (e.g. max pooling), so global features can be extracted. Typical architectures for point-based approaches are pointwise Multi-Layer Perceptron (MLP), convolution-based, (which are good with dealing with irregularities in 3D point clouds) or graph-based methods.
One of the very first networks processing point cloud input directly was PointNet [8]. It is well-known and many other networks are based on PointNet due to its simpicity.
As mentioned earlier the problem of permutation due to the unordered nature of point clouds is tackled in PointNet with a symmetric aggregation function (max-pooling) and thus a permutation invariance is achieved.
Building on T-Net, a network predicting input-specific transformation matrices (eg. rotated input) and fusing them with globally trained features, Pointnet is also invariant to geometric transformations, as it aligns input data along a certain axis.

However the interaction between the points can not be captured as the MLP layers are independantly applied to each point, but follow-up Networks like PointNet++ offer solutions.
In general it can be said that pointwise MLP Networks like PointNet that provide feature learning per point functionalities are often the first step in the building of a network.

3) Object detection: The main task of object detection is to apply a 3D bounding box to each detected object. To detect objects, either region proposal or single shot methods are used. They can be implemented with different techniques like a multi-view-based, a segmentation-based or a frustum-based approach.
Region proposal based methods are proposing regions with objects in them. Then features are extracted from each region

and the category of each proposal is determined.
Segmentation-based region proposal proofed to be quite effective. The idea behind segmentation-based methods is to remove background points and noise and then create proposals from the remaining points. This method is computationally efficient and produces high object recall rates. It is also suited for complicated scenes dealing with occlusion and crowded objects [6].

Single shot methods on the other had predict class probabilities directly which can be achieved with techniques like Bird's Eye View (BEV - a 2D scene-view from above), discretization-based and point-based methods.
Discretization-based methods are used the frequently in other networks. After the discrete transformation of the point-clouds they use Fully Convolutional Networks (FCN) for prediction.

The later steps in the Processing Pipeline of LiDAR data in deep learning can again be divided into object detection, tracking, recognition and motion prediction [1](Himmelsbach).

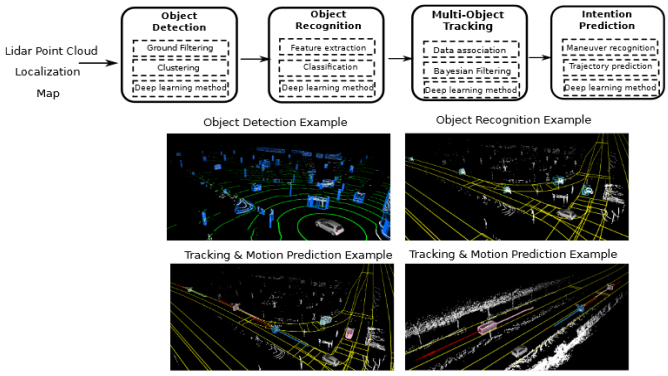


Fig. 8. Pipeline of classic LiDAR perception system [1].

4) Learning methods: Learning can be divided into data-driven and model-driven learning methods.
Data-driven methods can be computationally very expensive, but improve the semantic information provided by the data in comparison to model based learning [1].
The object recognition part of the pipeline maps semantic class information onto the detected objects.
Another distinction between learning models is off-and online learning.
Offline-learning: This method has a static annotated data set which is used to train a network.

Online-learning: With this approach data is coming in constantly and provides continiously new data. Also this method is more appropriate for a real-time environment and thus is more time-critical and the labeling process can be rather difficult.

Focusing on the previous scan of a LiDAR system, Real-time classification of humans can be achieved with online learning [2].
Even though pre-trained off-line classifiers have to be trained for each new environment, contrary to a moving LiDAR sensor like it is the case for autonomous driving and robotics, this is not problematic in an environment with a fixed camera.

G. PointPillars

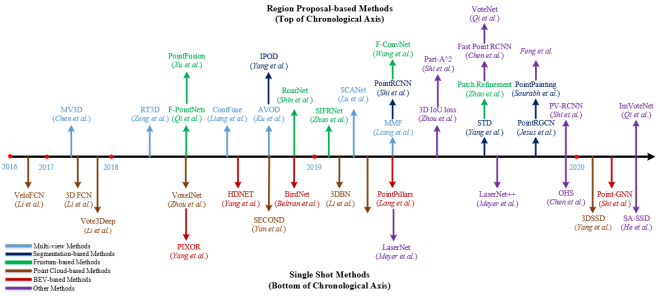


Fig. 9. Networks made for object detection [6]

The constant strive to improve the results of object-detections can be seen in the example of the different Networks trying to improve the results on the Kitti data set, in which, to date, more than 150 different Networks are competing to improve the detection of pedestrians based on the estimation accuracy of the detection, orientation and in percentage. [9], [10]
As previously stressed, following the importance of the data set, the processing of the available data is at the core of the first stages of a network. Since there are many methods and networks in 3D object detection as can be seen in figure 9, the focus on their comparisons will lie on the ones that achieve object detection using point cloud data.
One of these networks is PointPillars. What makes PointPillars very appealing for the purpose of this project is the aim to increase efficiency and speed with the goal to operate on real-time data while still having accurate results.
Reporting very good comparative results in 3D detection at the time of release (2019) and dealing with the challange of high dimensionality of LiDAR data, the PointPillars network is a candidate to look more into.

Making use of the fact that Convolutional Neural Networks (CNNs) achieve excellent results in detecting objects in images, and a fast and simple single-stage approach combining the regression of anchor boxes and their classification into one step, PointPillars is building on top of a simplified version of PointNet and following a similar architectural approach as VoxelNets' follow-up Network SECOND, PointPillars central achievement is the replacement of a 3D Convolutional Neural Network with a 2D CNN. Since 3D convolutional networks are slow, PointPillars is avoiding the problems that most volumetric-based methods using 3D CNNs are facing, which

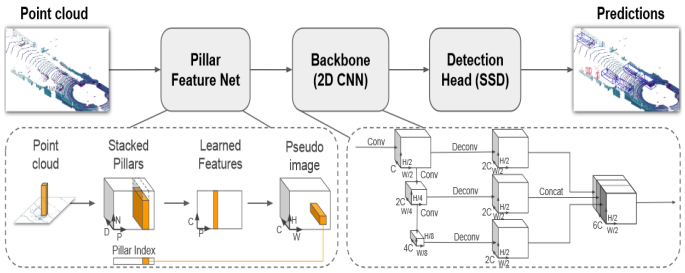


Fig. 10. Architecture of the PointPillars Network [11].

is that they don't scale well to dense 3D data since the computation and memory grow cubically with the resolution [6].

The PointPillar network consists of four main parts as can be seen in figure 10.

The Pillar Feature Net

To be able to apply a 2D CNN to point clouds, the 3D input is reduced to a 2D pseudo-image. This is done by the Pillar Feature Net, that is firstly dividing the point cloud into a vertical grid (x-y plane) and thus creating pillars. After calculating distances for each point in a pillar, calculating the amount of non-empty pillars and the number of points per pillar, it returns a tensor containing a new 9-dimensional LiDAR point. This data is then processed by the second part of the Pillar Feature Net which is based on PointNet. As Point Pillars does feature detection directly it is independent of potential influences of pre-defined encoders and reduces information-loss in the raw data. Another advantage of the self-reliant feature-learning is the adaptiveness to different point cloud configurations, which can vary depending on the used LiDAR sensor.

Using the PointNet feature detection functionalities, PointPillars utilizes them in their object detection pipeline. After applying PointNets to each point and encoding the features into a pseudo-image. [11], [12]

III. Tracking

Most tracking is done with some form of Kalman Filter (Unscented Kalman Filter + Nearest Neighbour in [2]) Mostly different variations of Kalman filter *elaborate [1]

IV. Testing

V. Results

VI. Conclusions

As LiDAR sensors only gather data from real physical objects, the data is not susceptible to difficult illumination conditions or similar and is highly reliable. [1], [2]

With increasing computational power it is easier to utilise three-dimensional data than it was a decade ago. While working with two-dimensional data was possible for most hardware, the processing of three-dimensional datasets needed access to hardware with a higher performance [2]. The processing speed and technologies to process and build real-time systems are now provided.

Less positive aspects of LiDAR like the sucseption to weather conditions like heavy rain, dust or fog that represent particles in the air that can hinder a proper reflection of the laser can be neglected in an indoor scenario. [1], [5]

Due to the laser transmitter threshold defined by the eye-safety standard:IEC 60825 it is also safe for use in an environment focused on human detection [1]

Tracking: A Kalman filter can be a good approach for efficient 2 dimensional tracking since people move on a plane [2].

The requirement to be a network working on real-time data is definetley something to consider in the choosing of the right building blocks for a suitable network.

VII. Acknowledgment

References

[1] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[2] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for human classification in 3d lidar-based tracking," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 09 2017, pp. 864–871.

[3] C. Thomson, "Point cloud survey specification — how to build a point cloud," https://info.vercator.com/blog/point-cloud-survey-specification-how-to-build-a-point-cloud, January 2020, (Accessed on 05/05/2021).

[4] A. Conner-Simons, "Deep learning with point clouds," https://news.mit.edu/2019/deep-learning-point-clouds-1021, October 2019, (Accessed on 05/04/2021).

[5] ——, "Deep learning with point clouds," https://lidarradar.com/info/advantages-and-disadvantages-of-lidar, (Accessed on 05/04/2021).

[6] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 06 2020.

[7] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3d: A review of point cloud semantic segmentation."

[8] C. Ruizhongtai Qi, H. Su, K. Mo, and L. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 12 2016.

[9] A. Conner-Simons, "Deep learning with point clouds," http://www.cvlibs.net/datasets/kitti/eval$_object.php?obj_benchmark = 3d, October2019, (Accessedon05/05/2021)$.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] A. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 06 2019, pp. 12 689–12 697.

[12] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," 06 2018, pp. 4490–4499.