Survey of state of the art VR Driving Simulation for Physical Test Car Using LiDAR for Mapping the Surrounding Environment

Raigo Kõvask
University of Tartu
Tartu, Estonia
raigo.kovask132@gmail.com

***Abstract*—This document gives an overview of the pipeline from collecting from the LiDAR sensor all the way to visualizing it in the modelling software. It will look into the different issues with mapping the LiDAR data into the correct coordinates with different SLAM algorithms and visualizing issues within software for the point cloud. In this we will make short description on different SLAM algorithms, and their pros and cons on which one to choose. The result was achieved by doing a systematic mapping study on various articles published on the web.**

***Index Terms*—LiDAR, SLAM, modelling, localization**

I. Introduction

The LiDAR sensors have become a large research area in recent years, with the rise of autonomous vehicles research. Because of that, the price of different LiDAR has seen a massive price drop in the past three years according to market research consultants Yole Développement. This means that LiDAR sensors could be more affordable and can be used for different use cases. The sensor is already used widely on autonomous driving to detect the surrounding environment for collisions and mapping for driving. This paper goes in another direction and tries to find solutions for remotely controllable vehicles through virtual reality. This could have multiple use cases such as creating remotely controllable vehicles so that the drivers don't need to be physically in the car or taking over the control of fully autonomous vehicles in case of errors. The focus of this paper is to address main issues that should be addressed when attempting to create an environment with LiDAR sensor. These are localization, mapping and modelling.

II. Method

Findings in this paper were discovered by doing a systematic mapping study. The mapping study consisted of the following steps:

- defining the research questions
- data search on different articles from ACM (∼60% of sources) and articles from various data sources(∼40% of sources)
- applying inclusion and exclusion criteria on the papers
- data extraction
- summary of the extracted data and results on findings

A. Research Questions

The mapping study investigates popular use cases for LiDAR vehicles, the latest algorithms used to track test vehicle position and thought that get correct position for LiDAR points, their pros and cons, and how to show correctly this data in the VR tools. This leads to the following research question:

- RQ1: What are the different sensors that have been used with LiDAR sensors to map surrounding environment?
- RQ2: What are the effective and accurate methods to see and track the position of the dummy car in the real-world using LiDAR?
- RQ3: What are the pros and cons in using different sensors and different algorithms?
- RQ4: What are possible solutions to generate and handle polygon meshes from the point cloud?

B. Search

Two different sources were chosen for this paper: ACM and other sources. Reason why the other internet sources were selected is that there are a lot of different well done papers that can be used in this paper. The found resources were checked for validity. ACM research was done using the following query:

> ([[All: location] OR [All: position] OR [All: navigation] OR [All: slam]] AND [All: lidar])

After the subset of articles were found from the articles, the "snowballing" was used to find more relevant articles based on the found articles.

C. Study Selection and Quality Assessment

For excluding and including articles following criterions were used:

Inclusion criterions:

- IC1: The study's title refers to information about LiDAR SLAM or model creation on LiDAR data (If it is questionable, then the abstract is reviewed).
- IC2: Studies contain information on how to create meshes from LiDAR point cloud.

- IC3: Studies describe certain SLAM algorithm to use with LiDAR sensor.
- IC4: Studies explain the positive and negative sides of using LiDAR.

Exclusion criterions:

- EC1: The study is a book.
- EC2: The study is not available in full text.
- EC3: The study is not in English.

In the beginning, there were 1517 articles. First, the criteria IC1 was used to sort out articles that were not related to the topic at all. Applying it reduced the number of articles to 40. After that, the articles were filtered by IC2, IC3, and IC4 (This means that the articles were already read through). This reduced the number of articles to 18. The exclusion criterions were applied to the articles found after the snowballing or finding them on various data sources. After the relevant papers were found additional ”snowballing” was applied to find extra articles. After snowballing there are total of 26 articles.

D. Data Extraction

To extract data from the given studies, a template was created (Table I). The table consists of the following columns: data item that describes the searchable item in the article, the value that shows the type of the data, and RQ that shows the research questions connected with the given data item.

TABLE I
DATA EXTRACTION TABLE

Data item	***Value***	***RQ***
Authors	Set of authors	
Title	Title of the article	
Solutions developed	Different solutions that are developed based on the LiDAR sensor	RQ1
Technologies used	Different hardware/software that was used in the given paper	RQ1
Methods used	Different methods used to track the dummy car with LiDAR	RQ2
Pros and cons of the methods/ solutions	Set of advantages and disadvantages of different methods. solutions	RQ3
Point cloud processing	Set of methods to process LiDAR point cloud to objects.	RQ4

III. FINDINGS

A. RQ1 - Different sensors used

Table II shows all of the practices found in certain papers. Different LiDAR sensors under this are counted as equal except for the Kinect. The reason for this is that different algorithms used LiDAR sensors with different configurations (i.e. RPLidar that rotates and receives input from that) or static LiDARs that have a limited view area that is usually 65 degrees.

B. RQ2 - Methods to track the car

This section gives an overview of different methods that were found in the papers and brief descriptions on how they work. In the end there's also a count on how many times different techniques were mentioned in different papers. Table III shows all of the different methods that use LiDAR scanner for estimating the position of the vehicle. Some of the methods that are defined in the table are not defined in the list, because the papers only mentioned the name of the method that was used, but did not describe it. Description is necessary to find out the overall idea of the method.

TABLE II
DATA EXTRACTION TABLE

Method	***Mentions***
LiDAR + IMUs	[1], [5], [8], [16], [21] [22]
LiDAR + Camera	[1], [5], [6], [18]
LiDAR + Kinect	[4], [24]
Kinect	[1]
GPS + IMU + LiDAR	[7], [12], [22]
LiDAR + GPS	[14], [25]
GPS + LiDAR + IMU + other surrounding vehicles and pinpoints	[20]
Radar + GPS + IMU + Camera [2]	
Camera + IMU + LiDAR	[26], [27]
LiDAR	[1], [3], [4], [9], [13] [15], [17], [19], [23]

Gmapping - A two-dimensional SLAM algorithm based on particle filtering, which integrates LiDAR and odometer information and has strong robustness[6]. It is the most widely used algorithm for 2D-SLAM based on laser radar. It is based on RBPF. The main difference of RBPF is that GMapping puts forward an improved proposal of distribution and selective resampling is proposed to reduce the frequency of resampling[15].

Hector-SLAM - This algorithm relies on high resolution LiDAR data. The map model used in this algorithm is grid map and is located by scanning matching[15].

Monte Carlo localization (MCL) - The Monte Carlo positioning merges data from LiDAR, map messages, coordinate transformation messages, odometer messages, and initial pose messages. Result is the highest probability through state estimation based on particle filtering[6].

Cartographer - It can provide 2D and 3D real-time SLAM in a multi-platform and sensor configuration. The principle of it is based on a graph optimization (eliminate cumulative error generated by loop closure detection during mapping process)[15].

Rao-Blackwellized particle filter SLAM (RBPF-SLAM) - Algorithm that estimates the posterior probability of the robot trajectory, and then calculates the posterior probability of the environment map based on the trajectory. The trajectory posterior probability is estimated by Rao-Blackwellized particle filter, where each particle represents a potential track and a map of environment[15].

Adaptive Monte Carlo localization (AMCL) - This is based on Monte carlo localization method. It adds adaptive KLD (Kullback-Leibler distance) method to update particles, which can effectively prevent particle degradation[6].

Normal Distribution Transform(NDT) - The NDT method divides space into cubes and generates a local probability density function (pdf) to represent each cube, so that each pdf may be thought of as an approximation of the local surface, defining the surface's position. It uses Newton's optimization method to find the rotation and translation between the two point clouds during the registration step, looking for the best fit between the two scans' pdfs[3].

Extended kalman filter(EKF) - Kalman filter algorithm involves predicting the state based on the system model and updating the state based on the measurements[7].

Iterative closest point(ICP) - ICP-based registration methods solves correspondence estimation between previous scans and transformation[4]. There are a lot of different submethods that have been developed from initial ICP method. The following methods are Correspondance estimation methods with ICP:

- ***ICP*** - It is a common algorithm for point cloud processing. Its purpose is to solve the registration problem of two sets of point clouds with similar shapes and deviated positions[16].
- ***PL-ICP*** - It is a common algorithm for point cloud processing. Its purpose is to solve the registration problem of two sets of point clouds with similar shapes and deviated positions. Unlike ICP, which uses the distance between a point and a point as an optimization function, PL-ICP uses a point-line distance as an optimization function [16].
- ***EfficientVarICP*** - summarizes the ICP processes and proposes strategies to improves the algorithm speed of the ICP process[4].
- ***IMLP*** - improves the ICP by incorporating the measurement noise in the transformation estimation[4].
- ***point-to-plane ICP***[4]
- ***generalized ICP*** - allows for the inclusion of arbitrary covariance matrices in both point-to-point and point-to-plane variants of ICP[4].
- ***plane-to-plane ICP***[4]

Transformation estimation methods for ICP:

- ***Singular value decomposition (SVD)*** - SVD-based estimation methods perform to the difference of correspondences. It obtains much faster efficiency and more accuracy then other estimation methods[4].
- ***Lucas-Kanade (LK) algorithm*** - This method estimates transformation with Jacobian of feature difference and approximation methods(e.g. Gauss-Newton). There's also submethod that has been extended from the given method called LM-ICP. It leverages the Levenberg-Marquardt algorithm to estimate the transformation by adding a damping factor to the original LK algorithm. It replaces the Euclidean distance with the Chamfer distance. [4]
- ***Procrustes analysis*** - This method converts the transformation estimation as a linear least-squares problem.[4].

Graph-based registration - In a graph-based registration the goal is to find point-to-point correspondences between two graphs(graph consists of edges and vertexes) by considering vertexes and edges.[4].

Gaussian mixture models (GMM) - The idea in GMM-based methods is to formulate the registration problem of Equation into a likelihood maximization of input data.[4]. There are some methods that have been developed from this method:

- ***CPD*** - CPD introduces a motion drift idea into the GMM framework. It adds constraints to transformation estimation.[4]
- ***CH-GMM*** - This method reduced the computation complexity of GMM by combining the convex hull(a tighter set of original point set) and GMM.[4].
- ***JRMPC*** - It recasts the registration as a clustering problem, where the transformation is optimizing by solving the GMM. [4]
- ***DeepGMR*** - This method uses deep learning to learn correspondences between GMM components and points. The transformation and the next GMM parameters can be estimated by next forward step.[4].

Feature learning methods - The main idea of it is to use the deep feature to estimate accurate correspondences. The transformation calculations can be later done with one-step optimization algorithms (e.g. SVD or RANSAC). There is no iteration between correspondence estimation and transformation estimation with this method[4]. The methods that fall under this category are used to extract distinctive features. Some of the methods that go under this category:

Learning on volumetric data

- ***3DMatch*** - This feature learning method learns on the RGBD(RGB-depth) images. It receives RGBD image as input and output is a 512-dimensional feature for a local patch.[4]
- ***3DSmoothNet*** - It introduces a pre-processing method to align the 3D patches and calculate the volumetric data based on the aligned 3D patches. It takes aligned volumetric data as the input for the model and outputs the geometric correspondence for the points[4].

Learning on point cloud - Difference from feeding the network the volumetric data, this method instead learns local descriptors on pure geometry and is aware of the global context[4].

- ***PPFNet*** - It uses the global and local features as input in an MLP block to generate the correspondence search feature. This method uses a point pair feature (PPF) to pre-process the input point cloud patches and a PointNet to extract a local feature. a global feature is received by applying a max-pooling operation[4].
- ***PPF-FoldNet*** - This is an unsupervised method to remove the annotation requirement constraint from PPFNet[4].
- ***SiamesePointNet*** - This method produces the descriptor of interest points by a hierarchical encoder-decoder architecture[4].
- ***3DFeatNet*** - a weakly-supervised approach that leverages alignment and attention mechanisms to learn feature

correspondences from GPS/INS tagged 3D point clouds without explicitly specifying them.[4].

- ***RPMNet*** - a less sensitive to initialization and more robust deep learning-based approach for rigid point cloud registration. This method's network can get a soft assignment of point correspondences and can solve the point cloud partial visibility.[4].
- ***Deep closest point(DCP)*** - The deep closest point (DCP) employs a dynamic graph convolutional neural network for feature extraction and an attention module to generate a new embedding that considers the relationships between two point clouds[4].
- ***IDAM*** - This method incorporates both geometric and distance features into the iterative matching process. It includes computing similarity scores based on the entire concatenated features of the two points of interest [4].

End-to-end learning-based methods - The Idea behind the end-to-end learning-base registration methods are that two-point clouds are fed into the neural network, and output is the transformation matrix[4]. This means that two steps are done simultaneously. Following methods are used to achieve this: **Registration by regression** - Idea for this is that the registration of transformation is done using the neural network to fit a regression model for the transformation matrix estimation[4].

- ***relativeNet*** - The pose is estimated directly from the features[4].
- ***DeepVCP*** - Uses probabilities learned among a group of candidates to detect keypoints, which can boost the registration accuracy[4].

Registration by optimization and neural network - It combines the conventional registration-related optimization theories with deep neural networks to solve the registration problem in Equation.

- ***PointNetLK*** - It uses the PointNet to extract global features for two input point clouds and then use a inverse compositional (IC) algorithm to estimate the transformation matrix[4].
- ***DeepGMR*** - It uses a neural network to learn pose-invariant point-to-distribution parameter correspondences. After this step the correspondences are fed into the GMM optimization module to estimate the transformation matrix[4].
- ***DGR*** - This method proposes a 6-dimensional convolutional network architecture for inlier likelihood prediction. It estimates the transformation by a weighted Procrustes module[4].

C. RQ3 - Positive and negative sides of different sensors/methods

The following section contains information about positive and negative sides of different methods and sensors.

Point cloud analysis overall

- Noise and outliers - Because the acquisition environment, sensor noise and sensor image mechanisms are different at different acquisition time, the captured point clouds will contain noise and outliers around the same 3D position[4].
- Partial overlap - Due to different viewpoint and acquisition time, the captured point cloud is only partial overlapped.
- Density difference - Due to different imaging mechanisms and different resolutions, the captured point clouds usually contain different densities[4].
- Scale variation - Since different imaging mechanisms may have different physical metrics, the captured point clouds may contain scale differences[4].
- Large datasets - This is an issue because the large datasets that LiDAR collects need significant computation power and additional disk storage[12].

TABLE III
DATA EXTRACTION TABLE

Method	***Mentions***
EKF	[5], [7], [11], [15], [16], [21]
HectorSLAM	[6], [15], [16], [27], [28]
GMapping	[6], [15], [16], [27]
Monte carlo particle filter	[6], [11]
LOAM	[1], [26]
Cartographer	[15], [16], [28]
Rao-Blackwellized Particle filter (RBPF)	[15], [16], [24]
ICP	[3], [4], [16], [22]
SegMap	[1]
LIO-SAM	[1]
KartoSLAM Triple-Surface Structure Extraction and Fitting.	[6]
Kalman Filter	[11]
Markov chains	[11]
AMLC	[6], [16], [17]
PL-ICP	[16]
MVAEE	[20]
point-to-plane ICP	[4], [26]
plane-to-plane ICP	[4], [26]
EfficientVarICP	[4]
generalized ICP	[4], [22]
DeepPCO	[26]
DeepICP	[26]
LO-net	[26]
CPD	[4]
CH-GMM	[4]
IDAM	[4]
Normal Distribution Transform(NDT)	[3]
RANSAC	[9]
kernel consensus-based robust estimator	[9]
LOL, a LIDAR-only Odometry and Localization algorithm	[1]
JRMPC	[4]
DeepGMR	[4]
3DMatch	[4]
3DSmoothNet	[4]
PPFNet	[4]
PPF-FoldNet	[4]
SiamesePointNet	[4]
3DFeatNet	[4]
RPMNet	[4]
Deep closest point(DCP)	[4]

Sensors pros and cons

- **Multiple sensors** - The errors from using multiple sensors can come from individual sensor calibration or measurement errors, lack of synchronization, or misalignment between the different sensors. [9]

- **Single sensor** - The limitation when using only single sensor in a complex environment is that there are several factors that can not be resolved with a single sensor such as a camera. A camera will not perform well in a low light environment, whereas a LiDAR sensor will not perform well in the presence of highly reflective surfaces such as a mirrors.[10]
- **GPS** - GPS is known to fail in tunnels so this is one of the issues with this solution. [14]
- **Kinect** - As the Low-cost RGB-D device, it provides ease of use and mobility but often it suffers from noisy alignments and global drifting[23]. But it can generate dense point clouds, while the view range is usually limited to 5 meters[4].
- **LiDAR** - It has a long view range while generating sparse point clouds[4]. the LiDAR based scanners produce highly accurate global 3D point clouds but are often restricted to rigid anchor points[23].

Mathematical theories vs deep learning

- When considering large datasets, mathematical theories are a significant shortcoming as they require both compute time and additional disk storage[12].
- Applying the mathematical theories on registration will have huge computation times[4].
- Applying deep learning algorithms will not guarantee accuracy[4].
- Directly combining deep learning and mathematical theories still require high computation time[4].

LIDAR-only Odometry and Localization - Main pro for this algorithm is that it can process the measurements more robustly and it needs input only from single sensor. Negative side is that the error could be accumulated in the long trajectories in the estimation, because the drift is continuously accumulated. This needs to be canceled by localization algorithm when the correct match has been found between the LiDAR stream and the offline reference map[1].

EKF - It is a loosely coupled method. Compared with loosely coupled, tight coupling is able to reduce inertial drift for those cases where scan images do not contain enough features to estimate location and orientation[7].

RANSAC - The accuracy is low when data contains multiple inlier structures[9].

Kernel consensus-based robust estimator - The accuracy is low when data contains multiple inlier structures[9].

Normal Distribution Transform(NDT) - Pros: The NDT algorithm is significantly quicker than it's closest counterpart ICP[3]. Cons: All reflected points from moving objects are counted as outliers[3].

Gaussian mixture models (GMM) - Pros: GMM-based method are robust to noise and outliers, since these methods align the distributions.[4].

Gmapping - Compared with the Cartographer, Gmapping does not require too many particles and has no loop detection when constructing small scene maps. Therefore, the amount of calculation is less than the Cartographer and the accuracy is not much worse. Gmapping can be used to build indoor map in real time, which requires less computation and higher precision in building small scene map[15]. It effectively takes advantage of the odometer information of wheels, which is also the reason why it has low requirements on LiDAR frequency: the odometer can provide robot pose prior[15].

Hector-SLAM - Hector SLAM, it requires less frequency and is more robust. Because Hector doesn't use a speedometer, it can be used in situations like disaster relief where the ground is uneven. But Hector is prone to mismatches when the robot makes a quick turn, and the resulting map is misaligned, mainly because the optimization algorithm is prone to falling into local minima. It is more suitable for corridor type environment[15]. The mapping results from it may be inaccurate due to the unexpected events. For example if a robot's velocity dramatically changes due to an external force causing the returned data from scanning to be distorted.

RBPF - It requires a lot of particles to build the map and it performs resampling frequently. When the number of particles is larger, the algorithm complexity will be higher, and frequent resampling will cause particle degradation. [15]It also has problems with high calculation cost and over memory consumption. In order to reduce the number of particles and samples, particles should be extracted from the proposal distribution function before the estimation of the next generation is determined. [24]

3DMatch - This method has two major cons: volumetric data requires large Graphic process unit (GPU) memory and it is quite sensitive to LiDAR rotation variations[4].

3DFeatNet - It doesn't require manual annotation of matching point cluster[4].

ICP - It requires hard assignments of closest points, it is sensitive to the initial transformation and noisy/outliers. Therefore, the ICP usually converges to the wrong local minima[4]. The registration efficiency is another remaining research problem, which also is a future research direction. The recent point clouds usually contain millions of points making ICP extremely slow. However, many current advanced methods are all required ICP to do the refinement to obtain high accuracy[4]. Since ICP generally registers two adjacent scans directly, the position of the new scan is only affected by the position of the previous scan, which causes the error to easily increase linearly with the increase in the number of matched scans. The rapid accumulation of errors is a difficult problem in most of the ICP based SLAM methods.[16] It also focuses on the global features without capturing the local pattern, which limits its accuracy.[26]

D. RQ4 - Polygons from point cloud

In this section the information about certain methods to handle the LiDAR point cloud from the meshes is presented.

Poisson surface reconstruction algorithm - This algorithm is used to join together multiple scans and to reduce the noise of the environment.[3]

The followings steps were done in the reconstruction pipeline to generate the surface[17]:

- Remove noise from the point cloud.
- Find K nearest points that create a plane.
- Find normals for the planes
- Reconstruction of the points based on Poisson surface reconstruction algorithm
- the Laplacian and Tubain Smoothing were applied on each part separately.

IV. Discussion

A. RQ1 - Different sensors used

The results from different papers show that the most popular methods were usually the ones based on the data recieved from LiDAR. The other methods in the papers supplemented the LiDAR sensor with other sensors and provided more accurate data. LiDAR and IMUs were also popular solutions as the IMUs improve the overall positioning of the car and it helps to avoid the overall drift of the point cloud. Multiple solutions also mentioned that LiDAR data can be merged with Camera data and the estimation could be built upon the results of this. But the solutions using Camera used their own methods to track the positioning and as this paper focuses on the solutions with LiDAR then these were not included here.

B. RQ2 - Methods to track the car

The results show that the most popular method that is used is Extended Kalman Filter(EKF). As these were only counted when it had a real comparison, the EKF was also mentioned in the other papers as a method that is used a lot. The next popular ones that were described were HectorSLAM, GMapping, and ICP. These are a bit older methods that are usually used to track the vehicle position in the world. As investigating different algorithms, a conclusion was made that different algorithms don't focus on the whole cycle of the SLAM. There are two sides to the SLAM that are usually focused on. These are correspondences and transformation estimates. For example, ICP is focused only to find out estimated correspondences, but some of them are end-to-end based methods like RelativeNet, that could also calculate the transformations. ICP wasn't the most popular method, but it had a lot of different submethods that were created to optimize the functions and make the algorithms more efficient. Almost 40% of the methods were using different machine learning algorithms and neural networks. Based on the articles release dates, it shows that these methods have become more popular in recent years. As this paper focused only on finding out different solutions, the estimates on how each method performs are not presented. Research "A comprehensive survey on point cloud registration"[4] gave a good insights on how some of the methods performed with each other.

C. RQ3 - Positive and negative sides of different sensors/methods

The negative and positive sides were collected based on the data obtained from different papers. The results had very few methods that were compared to some other method or some specs of it were brought out. The main con of using multiple sensors is that these sensors may be miscalibrated or they might be out of sync with each other. The pro for using multiple sensors is that the more sensors you have the more accurate reading you could get from it. Using multiple sensors in different environments gives more reliability when one of the sensors cannot handle the surroundings (GPS doesn't have a signal in tunnels or the surfaces are too reflective for LiDAR sensor). When comparing the methods, there were two larger types of methods: mathematical and methods using deep learning. Methods using mathematical theories are usually slower and can't handle large data sets, but methods using deep learning on the other hand have smaller accuracy. As for the comparison of different methods there wasn't much information about it and some of the methods differ on where it is used in SLAM algorithm. That is why we can't directly compare them. The pros and cons are still available in the data extraction section for the methods that were discussed in the papers.

D. RQ4 - Polygons from point cloud

As there weren't many articles that were describing algorithms or processes to create objects from the mesh clouds then this research question might need more input before deciding on what method to use for the given purpose. The papers that did describe reconstructing the environment described the Poisson surface reconstruction algorithm for the given purpose. This could be used as a basis for the given problem if there isn't any other research on how to achieve this result.

V. Conclusion

This paper focused on finding out different solutions to construct the environment from LiDAR point cloud into the 3D virtual world. This focused on three issues with given solution: mapping, localization and modelling the environment. For this the literature study was done and total of 28 paper were obtained from different sources to answer 4 different research questions.

This study concludes that there are a lot of different solution that could be used for localization and mapping the environment. As there was no comparison between these algorithms then we can't conclude on what algorithm to choose. The paper provides a insight on what methods exist and what could be used for this purpose. The last research question that was formulated to get information on different solution on how to handle point clouds in the game world got only a single match, thus it needs to be researched more before making any claims on it.

References

[1] Csaba Benedek, Andras Majdik, Balazs Nagy, Zoltan Rozsa, Tamas Sziranyi, "Positioning and perception in LIDAR point clouds", Digital Signal Processing, Volume 119, 2021, 103193, ISSN 1051-2004,

[2] Yanjun Wang, Qi Chen, Qing Zhu, Lin Liu, Chaokui Li, Dunyong Zheng Csaba Benedek, Andras Majdik, Balazs Nagy, Zoltan Rozsa, Tamas Sziranyi, "A Survey of Mobile Laser Scanning Applications and Key Techniques over Urban Areas", Remote Sensing, 2019, MDPI,

[3] Attila Borcs, Balazs Nagy and Csaba Benedek, ”Dynamic 3D environment perception and reconstruction using a mobile rotating multi-beam Lidar scanner”, Institute for Computer Science and Control (MTA SZTAKI), 2014

[4] Xiaoshui Huang, Guofeng Mei, Jian Zhang, Rana Abbas ”A comprehensive survey on point cloud registration” The University of Sydney, University of Technology Sydney, 2021

[5] Alicja Wasik, Rodrigo Ventura, Jose N. Pereira, Pedro U. Lima, Alcherio Martinoli, ”Lidar-Based Relative Position Estimation and Tracking for Multi-Robot Systems”, Distributed Intelligent Systems and Algorithms Laboratory, Ecole Polytechnique Federale de Lausanne, Switzerland 2014

[6] HongRun Wang, Hao Wang, JunYu Yi, ”Research on Intelligent Navigation and Positioning of Inspection Robot Based on SLAM”,China, RICAI, 2020

[7] Xiaorui Wang, Shuai Han, Baoshan Wei, ”LiDAR Aided Integrated Navigation System for Indoor Environments”,China, RICAI, 2019

[8] Tianci Chen, Lixue Zhu, WeiFeng Huang, Genping Fu, Chenyu Yang, Zhenlong Wu, ”Construction of Positioning and Navigation System for Mobile Vehicle Based on Simultaneous Localization and Mapping”,China, RICAI, 2019

[9] Wenwen Qu, Cheng Wang, Hanyun Wang, Ming Cheng, Jonathan Li, ”Triple-Surface Structure Extraction and Fitting for Control Point Positioning in LiDAR Point Clouds”

[10] Naman Patel, Prashanth Krishnamurthy, Yi Fang, Farshad Khorrami, ”Reducing Operator Workload for Indoor Navigation ofAutonomous Robots via Multimodal Sensor Fusion”, HRI, 2017, Vienna, Austria

[11] Yuri D. V. Yasuda ,Luiz Eduardo G. Martins ,Fabio A. M. Cappabianco, ”Autonomous Visual Navigation for Mobile Robots: A Systematic Literature Review”, ACM Computing Surveys, Volume 53, 2020

[12] James D. Lynch, Xin Chen, Roger B. Hui, ”A multimedia approach to visualize and interact with large scale mobile LiDAR data”, MM ’10: Proceedings of the 18th ACM international conference on Multimedia, October 2010, Pages 1689–1692

[13] E. W. Lam, T. D. C. Little, ”Refining light-based positioning for indoor smart spaces”, SMARTOBJECTS ’18: Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects, June 2018, Article No.: 9, Pages 1–8

[14] Andi Zang, Xin Chen, and Goce Trajcevski. 2015. ”Digital Terrain Model Generation using LiDAR Ground Points” In Proceedings of the 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics (UrbanGIS’15). Association for Computing Machinery, New York, NY, USA, 9–15. DOI:https://doi.org/10.1145/2835022.2835024

[15] Dong Shen, Yuhang Xu, and Yakun Huang. 2019. Research on 2D-SLAM of Indoor Mobile Robot based on Laser Radar. In Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering (CACRE2019). Association for Computing Machinery, New York, NY, USA, Article 64, 1–7. DOI:https://doi.org/10.1145/3351917.3351966

[16] Yibo Cao, Zhihui Zhang, Xiaosheng Chen, Haiwen Zhu, and Pengfei Zhao. 2021. Indoor SLAM Algorithm Based on PL-ICP and Map Matching. In Proceedings of the 2021 International Conference on Control and Intelligent Robotics (ICCIR 2021). Association for Computing Machinery, New York, NY, USA, 295–299. DOI:https://doi.org/10.1145/3473714.3473765

[17] Jan Hovad and Jitka Komarkova. 2014. Adaptive lidar grid filtering and utilization in the automated photorealistic surface modelling. In Proceedings of the 18th International Database Engineering Applications Symposium (IDEAS ’14). Association for Computing Machinery, New York, NY, USA, 362–365. DOI:https://doi.org/10.1145/2628194.2628253

[18] João Guerreiro, Daisuke Sato, Saki Asakawa, Huixu Dong, Kris M. Kitani, and Chieko Asakawa. 2019. CaBot: Designing and Evaluating an Autonomous Navigation Robot for Blind People. In The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS ’19). Association for Computing Machinery, New York, NY, USA, 68–82. DOI:https://doi.org/10.1145/3308561.3353771

[19] Rawan A. AlRashid Agha and Wrya Monnet. 2019. A virtual an augmented reality application of the Rashid Agha’s diwakhana in the citadel of erbil using 360° 3D Lidar. In Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems (DATA ’19). Association for Computing Machinery, New York, NY, USA, Article 32, 1–9. DOI:https://doi.org/10.1145/3368691.3368723

[20] Yanxing Song, Richard Yu, Yuchuan Fu, Li Zhou, and Azzedine Boukerche. 2019. Multi-Vehicle Cooperative Positioning Correction Framework Based on Vehicular Blockchain. In Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet ’19). Association for Computing Machinery, New York, NY, USA, 23–29. DOI:https://doi.org/10.1145/3345838.3356004

[21] Shi Han-hai, He Fa-jiang, Dang Shu-wen, and Wang Kang-le. 2017. Research on slam algorithm of iterated extended Kalman filtering for multi-sensor fusion. In Proceedings of the 3rd International Conference on Communication and Information Processing (ICCIP ’17). Association for Computing Machinery, New York, NY, USA, 242–246. DOI:https://doi.org/10.1145/3162957.3162999

[22] Henry Haugsten Hansen, Sayed Muchallil, Carsten Griwodz, Vetle Sillerud, and Fredrik Johanssen. 2020. Dense LIDAR point clouds from room-scale scans. In Proceedings of the 11th ACM Multimedia Systems Conference (MMSys ’20). Association for Computing Machinery, New York, NY, USA, 88–98. DOI:https://doi.org/10.1145/3339825.3391862

[23] James Pieszala, Gabriel Diaz, Jeff Pelz, Jacqueline Speir, and Reynold Bailey. 2016. 3D gaze point localization and visualization using LiDAR-based 3D reconstructions. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research Applications (ETRA ’16). Association for Computing Machinery, New York, NY, USA, 201–204. DOI:https://doi.org/10.1145/2857491.2857545

[24] YONG CHEN, Dang Shu wen, and LI Lu jun. 2020. Research and Application of Improved RBPF-VSLAM Algorithm. In 2020 the 6th International Conference on Communication and Information Processing (ICCIP 2020). Association for Computing Machinery, New York, NY, USA, 52–56. DOI:https://doi.org/10.1145/3442555.3442564

[25] Saleh Albeaik, Mohamad Alrished, Salma Aldawood, Sattam Alsubaiee, and Anas Alfaris. 2017. Virtual Cities: 3D Urban Modeling from Low Resolution LiDAR Data. In Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL ’17). Association for Computing Machinery, New York, NY, USA, Article 86, 1–4. DOI:https://doi.org/10.1145/3139958.3140025

[26] Ce Zheng, Yecheng Lyu, Ming Li, and Ziming Zhang. 2020. LodoNet: A Deep Neural Network with 2D Keypoint Matching for 3D LiDAR Odometry Estimation. In Proceedings of the 28th ACM International Conference on Multimedia (MM ’20). Association for Computing Machinery, New York, NY, USA, 2391–2399. DOI:https://doi.org/10.1145/3394171.3413771

[27] Ou Zheng and Hala ElAarag. 2018. Simultaneous localization and mapping using UAVs equipped with inexpensive sensors. In Proceedings of the Communications and Networking Symposium (CNS ’18). Society for Computer Simulation International, San Diego, CA, USA, Article 1, 1–10.

[28] Agustin Garcia-Rodriguez, Jose Gabriel Castillo-Garcia, Hugo G. Gonzalez-Hernandez, Jorge A. Reyes-Avendano, and Roberto J. Mora-Salinas. 2021. Autonomous Navigational System for an Industrial AGV using ROS and ZED Stereo Camera. In The 4th International Conference on Electronics, Communications and Control Engineering (ICECC 2021). Association for Computing Machinery, New York, NY, USA, 117–122. DOI:https://doi.org/10.1145/3462676.3462695