



# INDOOR 3D SCANNING AND NAVIGATION SYSTEM FOR AN AUTOMATED GUIDED VEHICLE

SORIN-DAN GRIGORESCU<sup>1</sup>, GEORGE-CALIN SERIȚAN<sup>1</sup>, BOGDAN-ADRIAN ENACHE<sup>1</sup>, FLORIN-CIPRIAN ARGATU<sup>1</sup>, FELILX-CONSTANTIN ADOCHIEI<sup>1</sup>, VICTOR STANCIU<sup>2</sup>

**Keywords:** 3D scanning; Automated guided vehicle; Light detection and ranging (LiDAR); Indoor mapping.

**The paper presents the development of a 3D laser scanning system that can produce an indoor navigation map for an automated guided vehicle. The system uses only a 1D light detection and ranging (LiDAR) as a measuring device and two stepper motors for positioning. Several tests are performed to determine the best trade-off between the time needed for the scan and the required resolution to produce an indoor navigation map, and the relationship between these variables is presented.**

## 1. INTRODUCTION

The recent pandemic caused by the novel coronavirus (COVID-19) restricted much of our human interaction and opened the door for automated guided vehicles (AGVs) as part of our social life. These vehicles can deliver goods to our door without human intervention, perform primary patient care, or help with house chores. However, the AGVs must navigate in indoor environments to perform all these tasks. These artificial environments have particularities, making them different from the exterior scenarios, which are custom for AGVs [1]. The scene can now suffer from structural variations (opened-closed doors) and various obstacles in the robot's trajectory (toys left on the ground, boxes, *etc.*). Another aspect is related to lighting, where different light sources, areas with vast shadows or very bright light, can make most of the cameras used by AGVs useless. Finally, the GPS used extensively for outdoor navigation has low or no reception indoors. To make it worse, the abovementioned issues cripple the navigation algorithms, making path planning, based on prior knowledge of the map, which is the most common navigation algorithm, an unpractical [2].

The first step towards solving this issue was to adapt the existing outdoor navigation systems to the indoor environment. Thus, vision-based mapless navigation methods using optical data flow, scene appearance feature detection, and image tracking were tested with some results. In [3], the AGV mimics the flight behavior of a bee by extracting optical flow to maintain between the corridor limits and optical flow intensity for obstacle detection. This approach requires that the medium around the robot to be textured enough so the optic flow can be relevant and high accuracy can be achieved. Another two-step approach combines potential fields with an appearance-based method for navigation and obstacle avoidance [1]. First, the appearance navigation separates the ground from the obstacles, and the results are used as input for the artificial potential field. While this approach was successful [4], the scene's lighting can be a significant source of obstacle detection errors. Other techniques based on image-based non-holonomic constraint [5], visual routes [6], and segment matching [7] between the current acquired image and reference images suffer from low robustness and poor generalization capability. These disadvantages, corroborated by heavy computational burden and insecure algorithms, steered researchers towards active sensor navigation methods.

A different approach is based on laser rangefinders, ultrasonic and infrared sensors, motion capture, and technical view systems. Combined with simultaneous localization and mapping (SLAM), all this can achieve the centimeter-level error needed in indoor environments [8]. Another advantage is that an absolute coordinate 3D point cloud map can provide the absolute location of the AGV. Nevertheless, visual odometry (VO) methods depend on high-accuracy sensors, high-speed data communication systems, and high computational capabilities [9]. Several implementations of VO make use of a multi-resolution visual fiducial and assistant navigation system [10], a monocular camera combined with MonoSLAM [11], a monocular visual-inertial system [12], *etc.* VO's core technology creates spares but persistent landmarks and a probabilistic framework to match the reference's feature points. The main drawbacks of this method are the lack of information on the scale of objects and high-accuracy sensors. A recently proposed method tries to overcome these issues by developing a factor graph [13] to integrate all sensor measurements into a navigation solution. It is based on a probabilistic graphical model which associates nodes to every system state. It uses historical information to convert the probability expression into a nonlinear problem. By applying statistical optimization methods, the best navigation trajectory is obtained. While this reduces the burden of high accuracy, high-cost sensors' gear, it shifts the focal point to high computational complexity systems.

The robotic navigation field is rapidly developing, but no universal solution exists. Depending on the environmental conditions and the AGV mission, the navigation method is usually a trade-off between the equipment and computational complexity [14]. In this context, the paper's main contribution is developing a 3D scanning system to produce an indoor navigation map with already available resources. It is designed for an AGV that operates in a residential area, a flat, or a one-floor house equipped with only one 1D LiDAR and additional positional motor gear. It can access the more powerful stationary house computer (server, laptop, or desktop). The proposed system is shown in section 2, while the calibration process that finds an optimum solution between the scanning time and data processing is presented in section 3. The provided results and the emerged navigation map are discussed in section 4, and the last part is dedicated to conclusions.

<sup>1-6</sup> Department Measurements of Electrical Apparatus and Static Converters, University Politehnica of Bucharest, Bucharest, Romania.  
E-mails: sorin.grigorescu@upb.ro, george.calin.seritan@upb.ro, bogdan.enache2207@upb.ro, florin.argatu@upb.ro, felix.adochiei@upb.ro, victor.stanciu@upb.ro

## 2. HARDWARE AND SOFTWARE SETUP

The proposed 3D scanning system consists of several components: an AVR microcontroller-based board equipped with a Proto Shield, a 1D LiDAR Lite V3, two stepper motors, and the associated motor drivers type A4988 – Fig. 1.

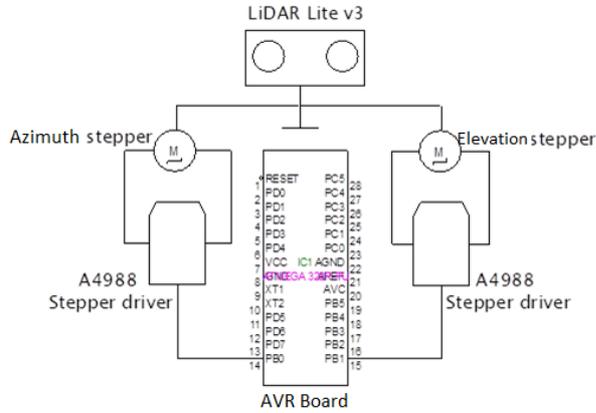


Fig. 1 – Block diagram of the proposed scanning system.

The microcontroller board is the core control component of the entire system. It receives the information from the LiDAR and computes the coordinates for every scanned point. It also issues the commands for the stepper motors to increment their position.

The 1D LiDAR Lite V3 is a standard distance measurement laser sensor. It determines the distance to an object using the Time-of-Flight method. It pulsed laser beam waves to the target and measured the time to reflect.

The data from one stepper motor represents the azimuth angle, while the other provides the elevation angle. The stepper motors have a  $0.9^\circ$  step resolution limited to a  $180^\circ$  view angle. Each motor can be controlled in a micro-stepping mode allowing up to 16 steps for each full step.

To process the data stream, the first attempt was to make all the required computations using the embedded AVR microcontroller running a program based on Eq. 1, but the process was time-consuming and prone to errors. Due to this, a different strategy was implemented. All the information gathered by the microcontroller: the spatial position and the distance measurements in spherical coordinates by the LiDAR are now sent to an onboard laptop using a serial interface. The personal computer uses a custom-designed sketch developed using Processing programming language, transforming the coordinates from spherical to Cartesian. These are displayed as a point cloud in an advanced software application – CloudCompare, for model and map development.

Suppose the AGV mechanical design is not large enough to sustain a laptop computer onboard. In that case, all the data can be sent to a stationary station through Wi-Fi, allowing the map to be constructed on a more powerful machine. This approach works well even if multiple AGVs are in the same area, only one performs the scan, and the stationary computer controls them all.

The operation of the system follows the sequence:

- The microcontroller calculates and sets the two stepper's absolute positions.
- The 1D LiDAR is turned on for that position and measures the average distance  $r$  to the nearest target.
- The measured distance information and the angles are sent to the laptop for processing.

- The step of the elevation stepper is incremented, and the new distance  $r$  and elevation angle  $\theta$  are recorded.
- When the elevation motor reaches its limit, its position is reset, and the azimuth motor's position is incremented, setting a new  $\varphi$  angle.
- The scanning is finished when both drives reach their limits i.e.  $180^\circ$  for azimuth and  $45^\circ$  for elevation.
- The spherical coordinate and data transformation from spherical to Cartesian coordinates is performed using

$$\begin{cases} x = r \cdot \sin \theta \cdot \cos \varphi \\ y = r \cdot \sin \theta \cdot \sin \varphi \\ z = r \cdot \cos \theta \end{cases} \quad (1)$$

After the scan, the collected data are processed and displayed as a point cloud. Several CAD applications like PointCloud or MeshLab are available for this task, but also custom ones can be easily created.

## 3. SYSTEM SETUP

The system setup aims to determine the optimal configuration of hardware settings, i.e. the number of steps required for both stepper drives and the software configuration for processing the original data point cloud.

The system's testing phase has involved varying the motors' number of steps to reduce the time needed for a complete scan without resolution loss in the generated point cloud.

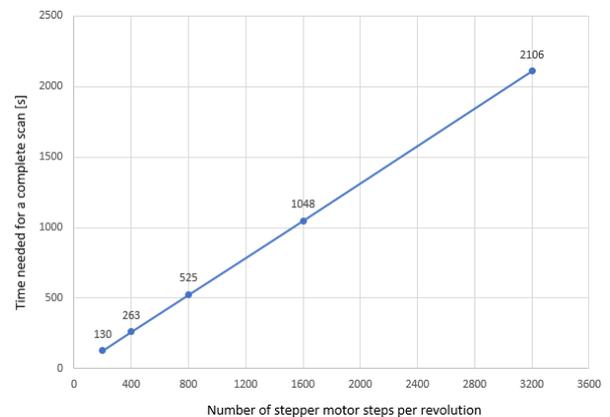
To reduce the complexity of the calibration stage, two scenarios were considered:

- The elevation stepper is set to perform 200 or 400 steps.
- The azimuth stepper is programmed to achieve 200, 400, 800, 1600, or 3200 steps for a complete scan.

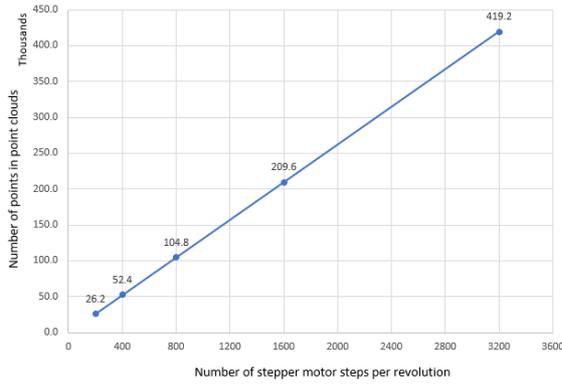
After the first two attempts with 200 steps for the elevation stepper and 200, respectively, 400 steps for the azimuth stepper, we concluded that the resolution of the point cloud was too low, and we excluded the 200 steps settings for the elevation stepper for the rest of the setup scenarios.

At the end of the hardware testing stage, a linear correlation between the number of steps of the azimuth stepper and the time for a complete scan – Fig. 2a, respectively, and the resolution – Fig. 2b, was established.

The point cloud was generated using the custom-designed sketch and CloudCompare in the second part of the setup process.



a) The correlation between the number of steps and time for a scan



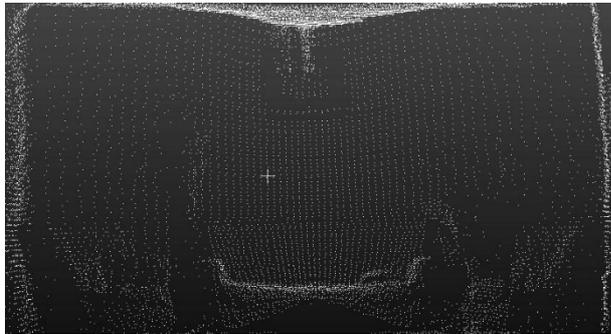
b) The correlation between the number of steps and resolution

Fig. 2 – Calibration results.

The first conclusion reached for developing a navigation map was that a minimum resolution of 800 steps for the azimuth stepper is required.



a) Point cloud for 400 steps, floating-point coordinates, and no grid.



b) Point cloud for 400 steps, coordinates grid set to 1 unit

Fig. 3 – Grid configuration for point clouds comparison.

Regarding the processing time, the first attempt used eq. 1, which produced floating-point data without any grid configuration and required almost 20 minutes on a laptop equipped with an i5 3GHz processor, and 4 Gb DDR3 RAM for an 800 steps scan. To reduce this time, in the second attempt, all coordinates were transformed for the three Cartesian axes X, Y, and Z from floating points into integers by using the processing *int* function, which is the equivalent of using a grid setting with the step of 1 unit. Before this conversion process, all floating point data were scaled with a factor of 1000 to preserve the most significant three decimals. This conversion process implies data loss, but the time was reduced to 27 seconds from 20 minutes.

#### 4. RESULTS AND DISCUSSION

The implementation of the system is presented in Fig. 4.



Fig. 4 – Robot with the proposed scanning and mapping system.

Using this system, a different room configuration was scanned beside the one used for setup. The new room has five by three meters in dimension and is located in an apartment, and the robot never scanned it before – Fig. 5.



Fig. 5 – The room used for testing.

In this environment, the robot performed the scan using the following configuration based on the setup process: azimuth stepper 800 steps, elevation stepper 400 steps, and the grid set to 1 unit. The scan process lasted for 17 minutes, and at the end, we obtained a point cloud with 209600 nodes after we excluded the errors from the 320000 gathered data. The whole process for representing the point cloud lasted 54 seconds, and the result is presented in Fig. 6.

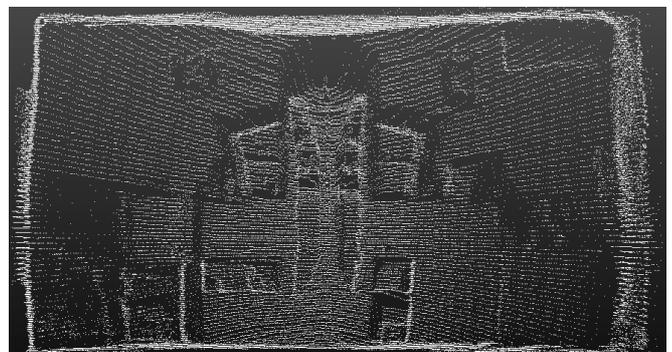


Fig. 6 – The representation of the point cloud for the scanned room.

From this data, we extracted a plan placed 0.9 meters from the ground (the robot's height with the system), which was further processed to turn into a 2D navigation map – Fig. 7.

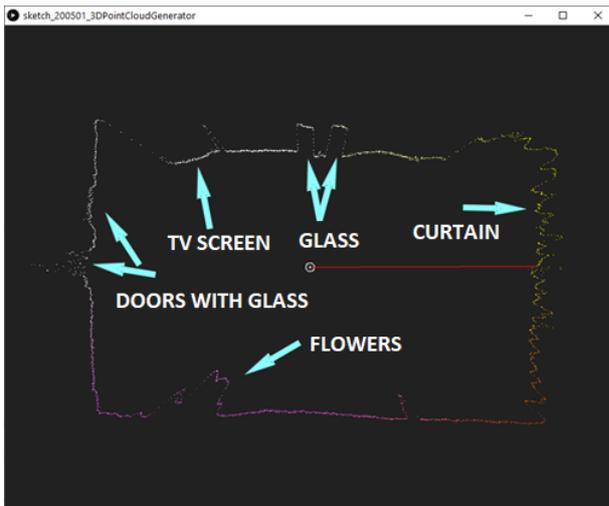


Fig. 7 – The 2D navigation map extracted from the 3D scan.

The obtained map shows the rough contour of the room, with some exceptions. The zig-zag line from the right represents the curtain's folds which shade the apartments' window, while the furniture door glass causes the two gaps from the center. The same effect is observed with a transparent glass component at the room door. Another mismatch is caused by the flowers used for decor, which perturb the measurements due to their intricate texture.

The overall process for constructing the navigation 2D map from the original 3D point cloud lasted almost 19 minutes, but this was done only once. All the other scans can be performed at low resolution because they identify the newly added objects to the room.

A similar approach was used in [15], where a LiDAR system produced a 3D map of the corridor of 15 m by 3 m. The scanning time was 16 minutes, but the number of steps was lower. The authors performed only 75 steps for azimuth and 60 for elevation. The main difference is that in [15], the authors focused more on identifying obstacles than producing a general map. Other than this, the mapping and processing times were similar.

Another key aspect is that all the processing was done using the laptop onboard at the AGV. Suppose the robot is not equipped with a high computation power machine. In that case, the presented methodology can still be implemented by transmitting the data to a fixed station that performs all the calculations and send the coordinates to the robot in the field.

## 5. CONCLUSIONS

The paper presents the development of a 3D scanning system that can produce an indoor navigation map for an AGV equipped with only a 1D LiDAR ranger. The system's performance depends on the computational processor power and the trade-off between the scanning time and the needed resolution. From the presented tests, one can establish a linear dependency between the steps performed to scan the room, the time for the scan, and the number of nodes in the point cloud.

The existing system was tested in a standard apartment room with all the furniture and accessories. The room's scan was done at the lowest determined resolution and lasted 17 min. Another aspect that influences the scan time and the number of errors is the presence of transparent or reflexive materials in the room.

## ACKNOWLEDGEMENT

This research was supported with the help of a project financed by the Ministry of European Investments and Projects, Managing Authority for the Operational Program Competitiveness (Directorate-General for European Competitiveness Programs) Competitiveness Operational Program, having Grant no. 121220.

Received on

## REFERENCES

- [1] Y. Song, L. T. Hsu, *Tightly coupled integrated navigation system via factor graph for UAV indoor localization*, *Aerosp. Sci. Technol.*, **108**, p. 106370 (2021).
- [2] M. Edl, M. Mizerák, J. Trojan, *3D laser scanners: history and applications*, *Acta Simulatio*, **4**, 4, pp. 1–5 (2018).
- [3] T. Ran, L. Yuan, J. B. Zhang, *Scene perception based visual navigation of mobile robot in indoor environment*, *ISA Trans.*, **109**, pp. 389–400 (2020).
- [4] X. Yang, R. Wang, H. Wang, Y. Yang, *A novel method for measuring pose of hydraulic supports relative to inspection robot using LiDAR*, *Meas. J. Int. Meas. Confed.*, **154**, p. 107452 (2020).
- [5] J. Li, X. Zhang, J. Li, Y. Liu, J. Wang, *Building and optimization of 3D semantic map based on Lidar and camera fusion*, *Neurocomputing*, **409**, pp. 394–407 (2020).
- [6] D. Rato and V. Santos, *LIDAR based detection of road boundaries using the density of accumulated point clouds and their gradients*, *Rob. Auton. Syst.*, **138**, p. 103714 (2021).
- [7] M. Petrov, A. Talapov, T. Robertson, A. Lebedev, A. Zhilyaev, and L. Polonskiy, *Optical 3D digitizers: Bringing life to the virtual world*, *IEEE Comput. Graph. Appl.*, **18**, 3, pp. 28–37 (1998).
- [8] F. B. P. Malavazi, R. Guyonneau, J. B. Fasquel, S. Lagrange, and F. Mercier, *LiDAR-only based navigation algorithm for an autonomous agricultural robot*, *Comput. Electron. Agric.*, **154**, pp. 71–79 (February 2018).
- [9] A. Schaefer, D. Büscher, J. Vertens, L. Luft, and W. Burgard, *Long-term vehicle localization in urban environments based on pole landmarks extracted from 3-D lidar scans*, *Rob. Auton. Syst.*, **136**, p. 103709 (2021).
- [10] E. Aldao, H. González-Jorge, and J. A. Pérez, *Metrological comparison of LiDAR and photogrammetric systems for deformation monitoring of aerospace parts*, *Measurement*, p. 109037 (2021).
- [11] A. Yilmaz and H. Temeltas, *Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LIDAR data*, *Rob. Auton. Syst.*, **122**, p. 103285 (2019).
- [12] P. Oğuz-Ekim, *TDOA based localization and its application to the initialization of LiDAR based autonomous robots*, *Rob. Auton. Syst.*, **131**, p. 103590 (2020).
- [13] T. Dehelean, C. Nafornita, A. Isar, *Enhanced metric for multiple extended object tracker*, *Rev. Roum. des Sci. Tech. Electrotech. Energ.*, **65**, 3–4, pp. 235–243 (2020).
- [14] R. H. E. Amar, L. Benchikh, H. Dermeche, O. Bachir, and Z. Ahmed-Foith, *Efficient trajectory reconstruction for robot programming by demonstration*, *Rev. Roum. des Sci. Tech. Electrotech. Energ.*, **65**, 1–2, pp. 117–121 (2020).
- [15] \*\*\* Athavale, R., Ram, D.H. and Nair, B.B., *Low cost solution for 3D mapping of environment using 1D LIDAR for autonomous navigation*, *IOP Conference Series: Materials Science and Engineering*, **561**, 1, pp. 1–7 (2019).