FPGA-based Architecture for a Low-Cost 3D Lidar Design and Implementation from Multiple Rotating 2D Lidars with ROS

J. Peña Queralta[1], F. Yuhong[1,2], L. Salomaa[1], L. Qingqing[1,2], T. N. Gia[1],
Z. Zou[2], H. Tenhunen[3] and T. Westerlund[1]

[1] Department of Future Technologies, University of Turku, Finland
[2] School of Information Science and Technology, Fudan Universtiy, China
[3] Department of Electronics, KTH Royal Institute of Technology, Sweden
Emails: [1]{jopequ, yuhong.y.fu, laolsal, tunggi, tovewe}@utu.fi, [2]{qingqingli16, zhuo}@fudan.edu.cn, [3]hannu@kth.se

***Abstract*—Three-dimensional representations and maps are the key behind self-driving vehicles and many types of advanced autonomous robots. Localization and mapping algorithms can achieve much higher levels of accuracy with dense 3D point clouds. However, the cost of a multiple-channel three-dimensional lidar with a 360°field of view is at least ten times the cost of an equivalent single-channel two-dimensional lidar. Therefore, while 3D lidars have become an essential component of self-driving vehicles, their cost has limited their integration and penetration within smaller robots. We present an FPGA-based 3D lidar built with multiple inexpensive RPLidar A1 2D lidars, which are rotated via a servo motor and their signals combined with an FPGA board. A C++ package for the Robot Operating System (ROS) has been written, which publishes a 3D point cloud. The mapping of points from the two-dimensional lidar output to the three-dimensional point cloud is done at the FPGA level, as well as continuous calibration of the motor speed and lidar orientation based on a built-in landmark recognition. This inexpensive design opens a wider range of possibilities for lower-end and smaller autonomous robots, which can be able to produce three-dimensional world representations. We demonstrate the possibilities of our design by mapping different environments.**

***Index Terms*—FPGA; Lidar; 3D Mapping; Point Clouds; Laser scanner; laser rangefinder; SLAM; Autonomous Robots;**

I. Introduction

The design and development of small autonomous robots have gathered increasing attention over the past decades. The use of lidars, light detection and ranging sensors, has enabled more reliable self-driving cars, but also a wider range of autonomous robots [1]–[5]. Lidars are able to produce accurate measurements, and the data they generate is usually processed as point clouds, a set of points in space.

Over the past two decades, 2D and 3D lidars have been adopted in a wide range of robots and autonomous vehicles [3], [6], [7]. However, 3D lidars are mostly reserved for high-end products and complex applications, such as self-driving cars, or object detection and classification [8], [9]. Autonomous robots operating in complex environments require 3D lidars for an enhanced situational awareness. Nonetheless, 3D lidars are expensive [10], [11]. On the other hand, the rapid development of lidar technology has enabled the commercialization of inexpensive 2D lidars under 100$. These are used in more basic setups [6]. Due to the difference of at least one order of magnitude between the prices of 2D and 3D lidars, the design

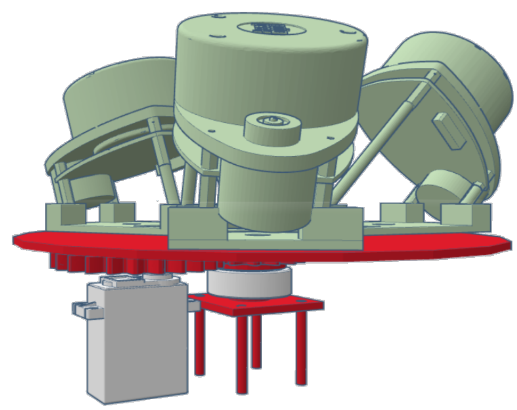


Fig. 1. Sensor setup with a servo motor and three 2D lidars.

of systems able to produce 3D world visualizations from 2D scanners can have a considerable impact on the performance of mobile robots relying on 2D sensors [12], [13]. It creates new possibilities in mapping [14] and localization [15].

Existing solutions utilize servo motors or stepper motors to move or rotate a single 2D lidar scanner. Then, a program is written to transform the two-dimensional information acquired by the lidar into three-dimensional point clouds. As 2D lidars already have built-in rotation, adding a new movement limits the speed of the additional rotation and therefore the refresh rate for 3D information is often a few Hz at most, significantly lower than the 2D counterpart [14]. Moreover, the three-dimensional point clouds generated by a single 2D lidar are sparse and traditional simultaneous localization and mapping (SLAM) algorithms cannot be directly applied [15]. We propose to overcome these two limitations by extending the same approaches to an arbitrary number of 2D lidars. However, while the computation needed to create the 3D point cloud is insignificant when compared to the analysis of the data for real-time localization or similar types of algorithms, adding multiple lidars increases the CPU load in terms of data acquisition and requires of multiple ports. Therefore, we propose the utilization of an FPGA board as a bridge between the 2D sensors and the CPU processing the data.

We introduce a pure VHDL implementation that acquires data from multiple channels, controls the rotation motor and outputs the 3D point cloud information to the CPU. This reduces the load and creates a flexible and extendable standalone sensor. The mapping and localization programs running on the CPU do not need to be modified as the setup consisting of multiple lidars and an FPGA can be utilized as a single 3D scanner.

One of the first attempts to utilize a moving 2D lidar for creating a 3D point cloud was carried out by Klimentjew *et al.* [12]. The authors applied the results for obstacle detection and avoidance in a service robot, enabling a more comprehensive situational awareness. Morales *et al.* used a servo motor to provide a fast and precise implementation of a moving 2D lidar over a fixed angle [13]. In these early attempts, continuous rotation had not been yet introduced. In the past few years, the topic has regained attention. Murcia *et al.* developed a package for the Robot Operating System (ROS) and a rotating 2D lidar with a stepper motor to obtain 3D point clouds of different scenarios [14]. The authors focused on static mapping to create very dense and large point clouds. In this paper, we present an approach that relies on multiple lidars to create generate point clouds that contain significant information about the environment in a fast scan. Recently, Bauersfeld *et al.* have also proposed the design of a low-cost 3D laser scanner [15]. The authors also present a review of SLAM algorithms suitable for the sparse point clouds generated by their design. In our case, by integrating multiple 2D lidars into a single sensor more generic SLAM methods can be utilized.

The main contributions of this work are the following: (i) the design and development of an FPGA-based solution that can accommodate an arbitrary number of lidar scanners and output 3D point cloud information to a PC; and (ii) a ROS package that processes the data received from the FPGA board and publishes to multiple ROS topics and enables integration with common mapping and navigation packages. Moreover, by providing an easily extendable and flexible FPGA-based solution, multiple other sensors can be easily integrated within the platform with a low impact on energy consumption and computational cost due to the inherent parallelization of the HDL-based design. Therefore, the main CPU that runs mapping and localization or other algorithms can be freed from the load of reading data from multiple channels.

II. Scanner Design

The rotating laser scanner that we have designed is illustrated in Figure 1. Multiple scanners are placed over a rotating platform with different inclination angles. The platform sits on a ball bearing and is rotated with a set of two gears and a continuous rotation servo motor. The lidars are connected to an FPGA board using a slip ring. The FPGA runs VHDL code that uses generic GPIO pins as inputs and outputs. In this project, we have utilized RPLidar A1 scanners are connected via a serial port with a baud rate of 115200. UART serial data, at a higher and configurable baud rate, is transmitted to a PC via a TTL-level USB adapter. As only generic GPIO pins are used and all the code has been written in VHDL, it is easily portable to virtually any FPGA platform with enough programmable gates and memory. A built-in landmark next to the motor, not shown in Figure 1 is used to continuously calculate its speed from lidar readings directly at the FPGA, which in turn controls the speed of the motor. The system architecture is illustrated in Figure 2. The system is designed to accommodate an arbitrary number of 2D lidars. If higher speed output is required to the CPU processing the data, then a different interface can be used. As the VHDL code is fully modular, this has a small impact on the code structure.

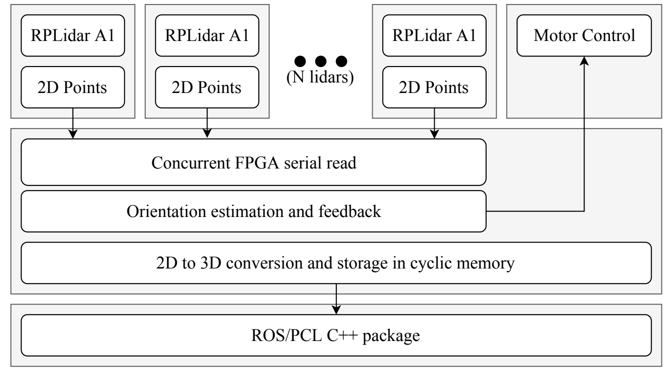


Fig. 2. System Architecture

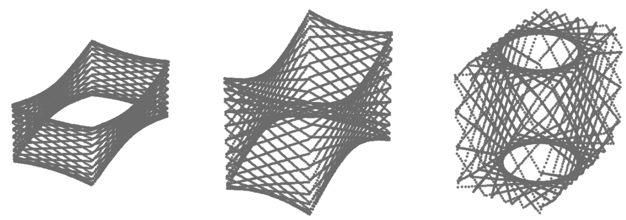


Fig. 3. Simulation of the point clouds generated by a rotating 2D lidar with inclinations of 9°, 16.6°and 45.5°in a room measuring 3.484 m by 6 m.

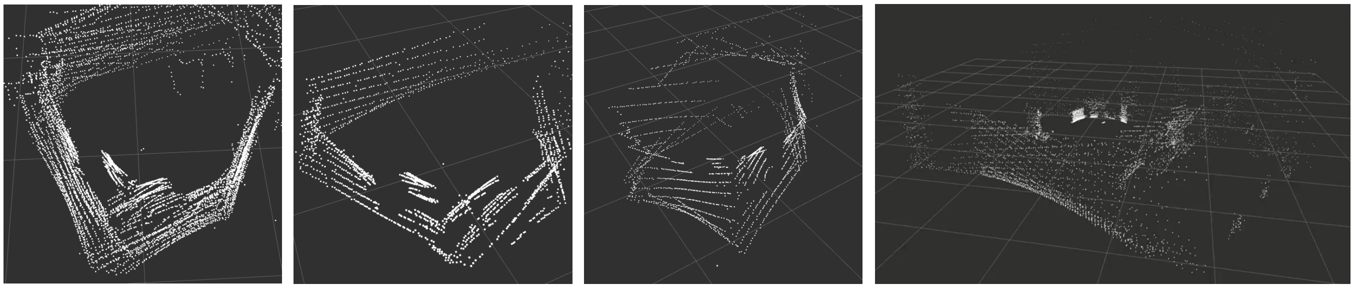
The following notation is used through this section. Each of the lidars is placed with an inclination ψ, which impacts the angular resolution of the 3D scanner. A pair (R, φ) represents the distance and angle measurements as directly obtained from each of the 2D lidar sensors. The final 3D point calculated taking into account the rotation θ and inclination ψ of the lidar is given by coordinates (x, y, z), while the coordinates (x', y') represent the projection of the measurements over the plane $z = 0$, and we define $R' = \|(x', y')\|$.

A. Three-dimensional point cloud

The FPGA board has two roles: to act as a data bridge between multiple lidar sensors and a CPU, and to transform two-dimensional sensor information to three-dimensional points.

Given a maximum sensing distance R_{max}, the points measured by the lidars at a certain distance lay can be projected over an ellipse of radii R and $R\cos(\theta)$. Then, the projected coordinates (x, y), without taking into account the rotation or orientation of the 2D lidar are given by

$$x' = R\sin\theta,\ y' = R\cos\theta\cos\psi,\ z = \sqrt{R^2 - x'^2 - y'^2} \quad (1)$$



(a) Full map with three lidars. (b) Lidar inclined 16°. (c) Lidar inclined 30°. (d) 3D map with multiple obstacles and furniture.

Fig. 4. Visualization of the 3D map as a point cloud with ROS and rviz. Subfigure (a) is the combination of (b), (c). Subfigure (d) shows a larger room.

and then the final (x, y) coordinates are given by

$$x = R' \sin(\theta + \varphi), \quad y = R' \cos(\theta + \varphi) \tag{2}$$

where $R' = \sqrt{x'^2 + y'^2}$.

B. 2D Lidar Inclination

The inclination of each of the lidars has a significant impact on the quality and usability of the 3D point cloud data. Figure 3. The utilization of multiple lidars allows for a three-dimensional point cloud with much richer information. Lidars with low inclination angle provide a detailed view of objects at a similar height, while lidars with larger inclination angle provide more sparse point clouds but include information about big surfaces such the floor and roof, as well as corners between walls, which is critical for many mapping and localization algorithms. Figure 3 (a) shows a point cloud of high density, while (c) provides a full view of the room including roof and floor, but with very sparse data on the walls. Figure 1 illustrates the setup with three lidars and different inclinations. The lidars are inclined approximately 16°, 23°and 30°. Using multiple lidars enables, on one hand, a higher density of points in different areas of space, if they have different inclinations. On the other hand, it allows for a higher refresh rate of measurements given a particular angle.

III. Experiment and Results

In order to test the usability and accuracy of the proposed 3D lidar design, we have used three RPLidar A1 M8, a low-cost 2D lidar priced under 100$. The RPLidar A1 is able to scan up to 10Hz and output up to 8000 measurements per second. In our setup, we use the lidar at 7Hz. The three lidars are connected to a Zybo Z7-20 board featuring the larger Xilinx XC7Z020-1CLG400C. The board has multiple peripherals and a dual-core ARM Cortex-A9 processor. However, in the design of this lidar only generic GPIO ports and the FPGA logic have been utilized, with all code written in VHDL as described in the previous section. This simplifies the process of portability to other platforms. A unoptimized version utilizes 401 LookUp Table (0.75%), or LUTs, 500 Flip-Flops (0.47%), and 13 IO ports (10%), leaving room for new functionalities or more lidars. The main limitation is the IOs. However, these can be multiplexed and only one extra RX line per additional lidar is needed to receive the data. The TX line sends basic commands, and thus can be shared by all connected lidars.

We have first compared the mapping capabilities of each lidar independently and the combination of all three. The results are shown in Figure 4. A simple room setup with three walls in different inclinations has been chosen to illustrate more clearly the impact of the lidar inclinations. Figure 4 (b) shows the point cloud generated by the most horizontal lidar, with dense scans of the walls but no view of the floor or roof. The most vertical lidar is shown in Figure 4 (c), where very sparse data is available at the walls but the roof starts to be visible. Figure 4 (a) shows the combination of these two and a third scan from a lidar with an inclination of 23°. The only objects in this setup were a chair, visible on the top-right side of the point cloud, and the built-in calibration landmark, which can be clearly seen in subfigure (b). During these scans, the rotation speed of the servo motor was 1/4Hz, which was incremented to 1Hz for the lidars using a pair of gears. The scans shown were acquired in half rotation, taking 500ms. Scans in (b) and (c) contain roughly 2000 points, for a total of 6000 in (a). Figure 4 (d) shows the scan of a larger room with a length over 7m and multiple objects. The sensor we propose is able to detect all objects in the room within its field of view.

IV. Conclusion and Future Work

We have presented a prototype of a low-cost 3D lidar based on multiple rotating 2D lidar scanners. The main contribution of this work is the sensor architecture and the integration of an FPGA bridge between the 2D sensors and the CPU that transforms two-dimensional data into a three-dimensional point cloud. To the extent of our knowledge, this is the first approach using FPGA technology. The solution we propose is able to accommodate an arbitrary number of 2D lidars, only limited by the number of GPIO pins of the board. Moreover, it could be easily extended to integrate other different sensors. We have shown how the proposed sensor can be used to obtain 3D maps of complex environments

Due to the limited space of the conference paper, this work has been focused on the design and implementation of the 3D lidar scanner, and on the utilization of using an FPGA board as a bridge between sensors and CPU. Applications of this sensor and its efficiency for more complex tasks such as real-time localization or mapping of larger areas will be included and analyzed in future work.

References

[1] X. Chen, H. Ma, J. Wan, B. Li, and T. Xian. Multi-view 3d object detection network for autonomous driving. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[2] B. Li, T, Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

[3] B. Schwarz. Lidar: Mapping the world in 3d. *Nature Photonics*, 4(7):429, 2010.

[4] R. Agishev, B. Gross, F. Moshary, A. Gilerson, and S. Ahmed. Range-resolved pulsed and cwfm lidars: potential capabilities comparison. *Applied Physics B*, 85(1):149–162, Oct 2006.

[5] J. Peña Queralta, C. Mccord, T.N. Gia, H. Tenhunen, and T. Westerlund. Communication-free and index-free distributed formation control algorithm for multi-robot systems. *Procedia Computer Science*, 151:431 – 438, 2019. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019).

[6] A. N. Catapang and M. Ramos. Obstacle detection using a 2d lidar system for an autonomous vehicle. In *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 441–445, Nov 2016.

[7] W. Maddern, A. Harrison, and P. Newman. Lost in translation (and rotation): Rapid extrinsic calibration for 2d and 3d lidars. In *2012 IEEE International Conference on Robotics and Automation*, pages 3096–3102, May 2012.

[8] M. Himmelsbach, T. Luettel, and H. . Wuensche. Real-time object classification in 3d point clouds using point feature histograms. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 994–1000, Oct 2009.

[9] 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*, pages 2798–2805, May 2011.

[10] J. Zhang and S. Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, page 9, 2014.

[11] S. A. Hiremath, G. W.A.M. van der Heijden, F. K. van Evert, A. Stein, and C. J.F. ter Braak. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture*, 100:41 – 50, 2014.

[12] D. Klimentjew, M. Arli, and J. Zhang. 3d scene reconstruction based on a moving 2d laser range finder for service-robots. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1129–1134, Dec 2009.

[13] J. Morales, J. L. Martínez, A. Mandow, A. Pequeño-Boter, and A. García-Cerezo. Design and development of a fast and precise low-cost 3d laser rangefinder. In *2011 IEEE International Conference on Mechatronics*, pages 621–626, April 2011.

[14] H. F. Murcia, M. F. Monroy, and L. F. Mora. 3d scene reconstruction based on a 2d moving lidar. In *Applied Informatics*, pages 295–308, Cham, 2018. Springer International Publishing.

[15] L. Bauersfeld and G. Ducard. Low-cost 3d laser design and evaluation with mapping techniques review. In *2019 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, March 2019.