Proceedings of the ASME 2019 International Mechanical Engineering Congress and Exposition
IMECE2019
November 11-14, 2019, Salt Lake City, UT, USA

IMECE2019-11279

DESIGN AND DEVELOPMENT OF A COST-EFFECTIVE LIDAR SYSTEM FOR TRANSPORTATION

Theodore Wiklund, Mark Heim, Jaret Halberstadt, Michael Duncan, Deven Mittman, Thomas DeAgostino, and Christopher Depcik[1]
University of Kansas
Lawrence, Kansas, US

ABSTRACT

Light Imaging Detection and Ranging (LIDAR) cameras and Light Detecting and Ranging (LiDAR) rangefinders were initially implemented in the 1960s as a higher-resolution and increased capability alternative to radar. Since then, LIDAR and LiDAR (hereto called lidar) have expanded into applications in aerial geographical surveying and collision-detection systems for autonomous vehicles. Current commercial systems are relatively expensive and potentially oversized for non-commercial applications. Consequently, this deters their use on consumer products like bicycles, where lidar systems can enable safety advancements that are necessary to counter the rising numbers of hazards affecting riders. In addition, widespread usage of inexpensive lidar systems can facilitate a more complete picture of our transportation infrastructure by delivering information (e.g., pavement quality) suited for U.S. Department of Transportation Highway Performance Monitoring System (HPMS) reports. This will aid in the creation of a safer infrastructure by highlighting critical areas in need of improvement and repair.

As a result, this effort outlines the development of a compact and cost-effective lidar system. The constructed system includes the ability to generate a static image by collecting several hundred thousand distance signals measured by a lidar rangefinder. Since the rangefinder has no self-contained rotation or translation systems, an Arduino Mega 2560 v3 microcontroller operates a pair of stepper motors that adjusts its azimuthal angle and pitch. Coalescing these signals into an ASCII text file for viewing in MATLAB results in a reasonably accurate picture of the surroundings. While the current system takes 1-2 hours to complete a full sweep, it has the potential to provide sufficient accuracy for HPMS reports at a moderate expenditure: the entire system costs less than $300. Finally, upgrading to a more powerful microprocessor, implementing slip rings for enhanced electrical connectivity, and refining the code by including interpolation between points will enable faster point cloud generation while still maintaining an inexpensive device.

[1] Contact author: depcik@ku.edu

KEYWORDS: lidar, rangefinder, point cloud, transportation, three-dimensional mapping, inexpensive

NOMENCLATURE

ASCII	American Standard Code for Information Interchange
HPMS	Highway Performance Monitoring System
lidar	Light Detection and Ranging
LIDAR	Light Imaging Detection and Ranging
MATLAB	Matrix Laboratory
3-D	Three-Dimensional

1. INTRODUCTION

Methods of transportation can vary for individuals depending on weather, destination, purpose, or other factors. Some might use public transit, personal vehicles, bicycles, or walking as their preferred mode of transportation. Unfortunately, this wide variance of options with disparate speeds results in a complex environment with vehicular collisions accounting for a quarter (24.9%) of all accidental deaths in the United States in 2016 [1]. Understandably, safety is a major concern for most commuters and while the number of accidents has decreased in the past, there has been a recent rise since 2014 [2].

A primary safety concern is the existence of blind spots. Typically, rear and side-view mirrors help drivers monitor the area behind them. While additional mirrors are suggested to completely eliminate blind spots, watching multiple mirrors will slow drivers' reaction time [3]. Therefore, it is preferable to monitor the area surrounding the vehicle or bicycle via another system. A detection system to alert drivers, visually and/or audibly, would help improve reaction time while potentially providing more consistent benefit than mirrors alone. A secondary safety issue includes the condition of the road. Inadequate road infrastructure is listed as a frequent cause of single-vehicular mishaps, especially rollover accidents [4, 5]. With the United States infrastructure currently in poor condition

[6], having a detection system monitor road conditions in addition to blind spots would result in a significant opportunity to improve safety.

Lidar is one remote sensing method that can facilitate an effective monitoring of both safety concerns. Briefly, lidar works similar to radar systems by using near visible light waves instead of radio waves and can map the surrounding environment in three-dimensions (3-D) [7]. Current applications for drone-mounted aerial lidar systems include forest mapping to track growth, modeling forest fire behavior, classifying land and environmental types, and charting various other environments for a variety of purposes [8-11]. Additionally, ground-based mobile lidar systems can recognize various road types and identify defects in their respective surfaces while monitoring the environment surrounding roads for potential dangers [12]. In areas where valleys and other steep slopes are adjacent to roads, rail lines, and canals, landslides are detrimental to transportation and infrastructure. Here, lidar systems can be used to inspect surface material and identify changes and patterns that might lead to landslides [13].

While these applications illustrate lidar's propensity to provide accurate and detailed representations, it is often costly to collect these data while respectively difficult to analyze the point cloud files that result from the collocation of this information [8]. Commercial lidar systems are highly capable; however, their individual cost ($6k to $100k [14]) might be excessive for numerous vehicle-mounted systems. For instance, a vehicular system does not have to scan wide areas of land at a time, only the immediate vicinity if there is a targeted goal in mind (e.g., road conditions versus automated driving). Hence, designing an inexpensive and small lidar system to identify vehicle, pedestrian, and bicycle proximity along with road defects could significantly benefit transportation safety while providing for widespread implementation.

As a result, this effort describes the design of a more accessible and inexpensive lidar system while briefly discussing the potential impact it can have for transportation safety. First, two configurations of the hardware employed are presented highlighting a change from servomotors to stepper motors to enhance accuracy. Next, a straightforward methodology in data collection is indicated to generate the point cloud information via text files. Finally, the implementation of both configurations is presented stressing the lessons learned while culminating in the development of an instrument that should cost less than $300 and be capable of producing relatively accurate 3-D point clouds.

2. HARDWARE AND SOFTWARE

Since lidar uses infrared light, its wavelength (e.g., 905 nm) is reduced significantly in comparison to comparable radar systems (e.g., 50 cm). This provides it the capability to generate a high-resolution image (aka high point density point cloud). Lidar rangefinders determine the distance of objects by emitting short pulses of light and recording the time it takes for this light to return to the detector. Object distance is determined by multiplying the speed of light by half the time it took the laser pulse to return. Subsequently, combining this distance with known horizontal and vertical angles of the rangefinder determines the x, y, and z positions of individual points. A point cloud is generated from this 3-D map using an appropriate software program.

The fabrication of a complete lidar system includes integrating a lidar rangefinder with some mechanism of sweeping this component in three-dimensions. Furthermore, a microprocessor is required to process and store these data. Previous experience in creating this lidar system for the back of an electric bicycle demonstrated limited success and generated only two-dimensional information [15]. Building on this prior knowledge, this effort expanded the system's capabilities into 3-D via two successive hardware configurations. In both configurations, the Garmin LIDAR-Lite v3 module is employed since it has a greater range and accuracy (40 m ± 10 cm) than other inexpensive alternatives: e.g., Taidacent TOF 10120 (1.8 m ± 5%) and the Benewake TF Mini (12 m ± 6 cm). The LIDAR-Lite v3 also provides several different configuration settings that can be explored to enhance resolution accuracy.

2.1 Configuration I: Servo Motors

Learning from the preceding effort, an Arduino Mega 2560 Rev3 (16 MHz: henceforth Mega) was used as the microprocessor instead of an Adafruit Feather System or a Raspberry Pi 3B+. The open-source Arduino Integrated Development Environment and modified C++ programming language is well documented and respectively easy to learn for undergraduate students (the primary authors of this paper). Furthermore, Garmin officially supports the LIDAR-Lite v3 rangefinder on the Arduino platform and a library is supplied on Github [16]. In contrast, while the Adafruit system worked previously, it did not provide sufficient processing speed and tended to be unreliable. Moreover, while the greater processing speed of the Raspberry Pi 3B+ (1.4 GHz, 64-bit quad-core) is advantageous for mobile systems, the primary issue of the aforementioned efforts suggested the focus be on point cloud accuracy over computational speed. This goal, when combined with a greater difficulty in learning the native Raspbian operating system and Python programming language (along with the LIDAR-Lite v3 not being officially supported on the Raspberry Pi platform), further solidified the choice of the Mega.

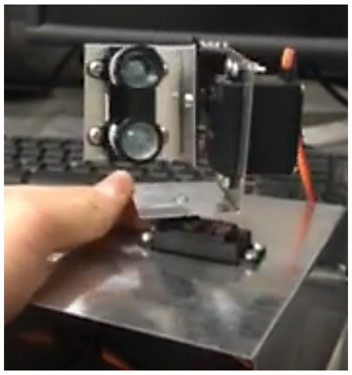


FIGURE 1. CONFIGURATION I ILLUSTRATING THE LIDAR-LITE V3 RANGEFINDER AND THE TWO TOWERPRO SERVOS ON A BENT ALUMINUM BASE.

In the first configuration shown in Figure 1, two TowerPro MG996R digital metal gear servomotors were employed to rotate in the *x*-*y* and *y*-*z* directions, respectively. A laptop computer supplied power for the entire system and a capacitor was used to protect the rangefinder from voltage spikes or current bursts. In Figure 2, the rangefinder and Mega communicated over the Inter-Integrated Circuit protocol using Serial Data Line and Serial Clock Line Mega pins, colored blue and green in the figure, respectively. Furthermore, signals to the servos were sent using the Mega's Pulse Width Modulation pins.

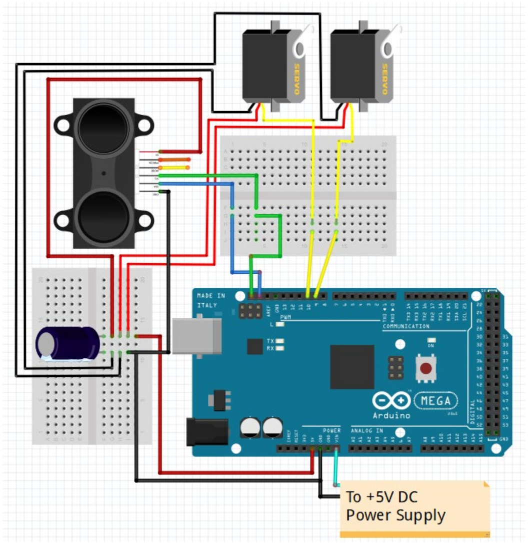


FIGURE 2. WIRING DIAGRAM FOR CONFIGURATION I WITH THE MEGA SUPPLYING ENERGY TO BOTH THE LIDAR RANGEFINDER AND THE SERVOS.

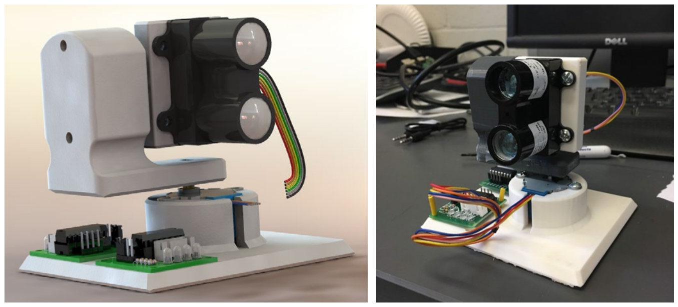


FIGURE 3. (LEFT) ISOMETRIC FRONT VIEW OF SOLIDWORKS COMPUTER AIDED DRAFTING MODEL OF THE SECOND CONFIGURATION HOUSING AND (RIGHT) ISOMETRIC FRONT VIEW OF PRINTED SECOND CONFIGURATION ASSEMBLY WITH STEPPER MOTORS, MOTOR CONTROLLER, AND RANGEFINDER ATTACHED.

Programming of the servomotors included adding a servo library to the Arduino code [17]. Using the angles of the motors retrieved from this library during operation, trigonometry was employed to calculate the *x*, *y*, and *z*-coordinates of each distance measured from the rangefinder. Unfortunately, the servos chosen could only rotate 180º in 1º increments. As a result, this configuration was unable to create a spherical point-cloud and had a relatively large degree per step value.

2.2 Configuration II: Stepper Motors

Similar to Configuration I, the Mega was used as the microprocessor for Configuration II. Now, two Kiatronics 28BYJ-48 5 VDC stepper motors, controlled by a Kiatronics ULN2003 motor controller, were employed to move the rangefinder. Each stepper motor had a gear reduction of 1/64 allowing for a rotation of 0.08º per step, facilitating a significant improvement in point cloud resolution (shown later in Section 3).

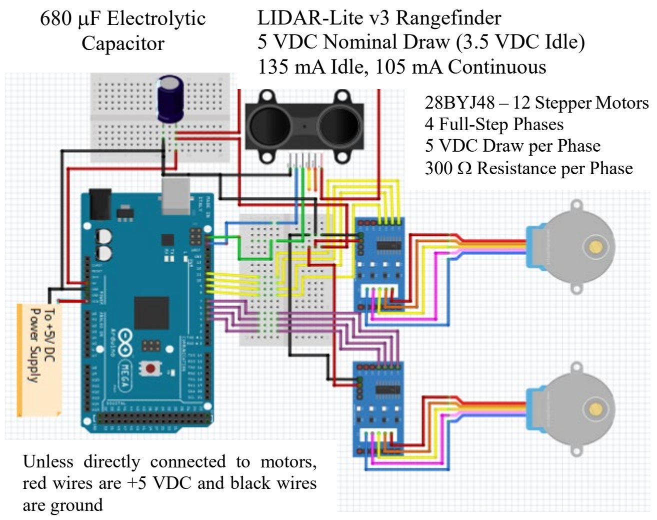

680 μF Electrolytic
Capacitor
LIDAR-Lite v3 Rangefinder
5 VDC Nominal Draw (3.5 VDC Idle)
135 mA Idle, 105 mA Continuous
28BYJ48 – 12 Stepper Motors
4 Full-Step Phases
5 VDC Draw per Phase
300 Ω Resistance per Phase
To +5V DC Power Supply
Unless directly connected to motors, red wires are +5 VDC and black wires are ground

FIGURE 4. WIRING DIAGRAM FOR CONFIGURATION II ILLUSTRATING AN ADVANCED COMPLEXITY OVER CONFIGURATION I (FIGURE 2) DUE TO THE ADDITION OF TWO STEPPER MOTORS.

During testing of Configuration I, the bent aluminum structure (Figure 1) flexed during operation resulting in the rangefinder not rotating around a fixed point in space. Moreover, the motor shafts did not line up to the fixture point of the rangefinder resulting in data that did not have a common origin. Instead, the second configuration included a 3-D printed housing, as illustrated in Figure 3, which provided a solid base, minimized vibration during usage, and created a common origin. This housing was printed from acrylonitrile butadiene styrene using two Stratasys Mojo fused deposition modeling printers and took 9.1 hours to complete while utilizing 5.7 in^3 of material.

Like the first configuration in Figure 2, wiring of the second version in Figure 4 involved power being supplied by a laptop computer and a capacitor was implemented to protect the rangefinder. Now, the Mega communicated with two motor controllers connected to the stepper motors that operate using four electromagnets. These motors can be rotated at half steps between the magnets enabling an advanced resolution. Unfortunately, the available Arduino stepper motor library did not properly communicate with these motor controllers [18]. Therefore, code was written to directly change the voltages of the electromagnets inside these motors, one magnet at a time.

2.3 Point Cloud Software

The data coming from a rangefinder includes the raw distance; hence, the most straightforward format for generating point clouds is through the American Standard Code for Information Interchange (ASCII) .xyz file type that features three columns of *x*, *y*, and *z*-coordinates for the thousands of points in a point cloud. Most commercial software packages that generate point clouds are setup to read the industrial standard .las and .laz lidar data. While initially the Trimble Realworks Viewer 11.0 was used because it can plot both .xyz and .las formats enabling a transition between the generated raw distance data into the industry format, it was decided to employ MATLAB as an alternative point cloud processing tool.

The LIDAR-Lite v3 rangefinder utilized is not capable of detecting color and is not officially supported to provide signal strength data. Whereas, .las and .laz file types allow for incorporation of color and signal strength. Furthermore, the students involved in this effort are familiar with MATLAB programming through their undergraduate curriculum. As a result, MATLAB code was generated that can parse data arrays from the Arduino system and concatenate this information into *x*, *y*, and *z*-coordinates. When reviewing these data in the following section, it was found that some datasets had points that were not near the subject of interest; i.e., random outliers. Code was added to filter this outlying data to ensure presentation of only the area of interest. These data are then plotted using the 3-D scatter plot option in MATLAB with color used as the legend to determine the distance away from the rangefinder. Except for one instance, ASCII .xyz text files were used to generate the point cloud images in the next section.

3. RESULTS AND DISCUSSION

The first point cloud generated using Configuration I and plotted using the Trimble Realworks Viewer is illustrated in Figure 5. Overall, these data took four minutes to capture and the servomotors were programmed to rotate 30º horizontally and 45º vertically. While the edge of the monitor on the right is somewhat visible in the point cloud at a slightly different angle, the overall point cloud resolution is poor. It is possible that the monitor screen material interfered with the rangefinder's laser pulses by absorbing or reflecting them away from the rangefinder; hence, it shows up as an empty screen area. In addition, while the monitor on the left is partially visible in the point cloud, the window behind the monitors prevented any further details from appearing as the laser pulses went through into the next room and did not return to the rangefinder.

At this point, a second set of data were taken using Configuration I to see if any improvements could be made to the setup or the underlying Arduino code. This time MATLAB was used to generate the point cloud with the corresponding picture and point cloud shown in Figure 6. It took eight minutes to generate these data and during this process, the aluminum mount was seen to wiggle after each horizontal sweep was completed, resulting in the double image seen in this point cloud. The point cloud still has a respectively poor resolution and the system loses accuracy as the distance from the rangefinder increases; i.e., the points get further apart the farther they are away from the rangefinder.

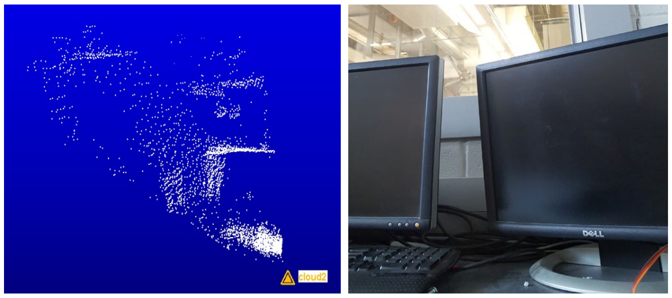


FIGURE 5. (LEFT) FIRST POINT CLOUD GENERATED USING CONFIGURATION I AND (RIGHT) THE CORRESPONDING PICTURE LOCATION.

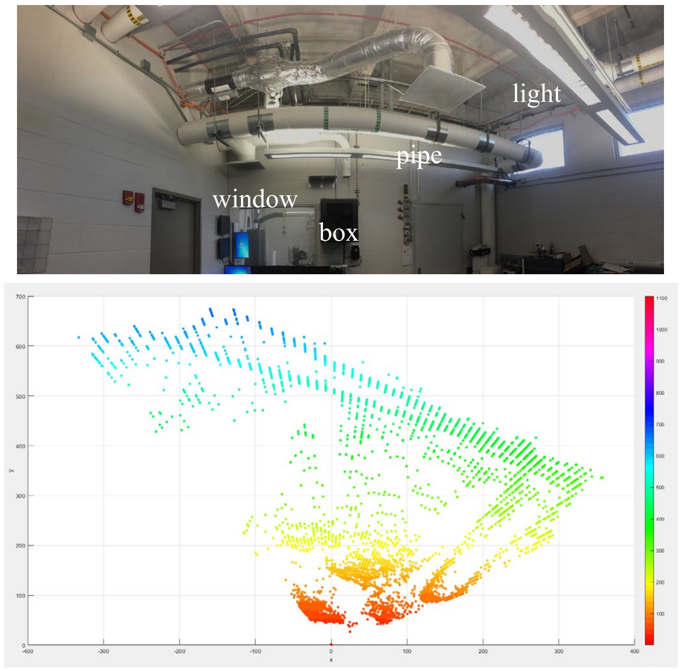


FIGURE 6. (TOP) PHOTO OF CONTROL ROOM FOR ENGINE TEST CELL ON CAMPUS AND (BOTTOM) CORRESPONDING TOP VIEW OF THE POINT CLOUD GENERATED OF THIS ROOM IN MATLAB.

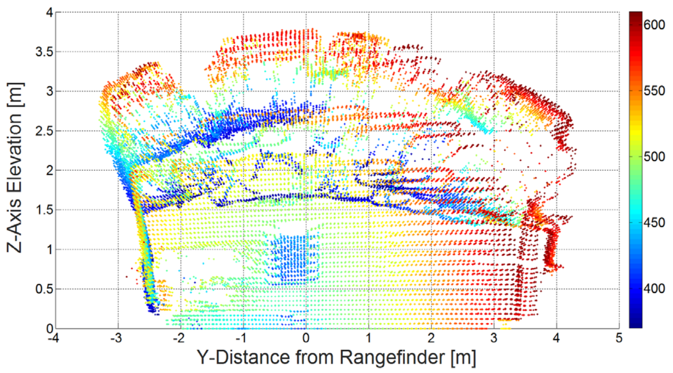
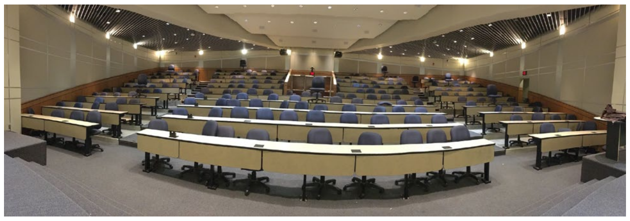


FIGURE 7. FRONT VIEW OF POINT CLOUD USING CONFIGURATION II WITH THE SERVER BOX AND WINDOW (SEE FIGURE 6) NOW DISTINGUISHABLE.



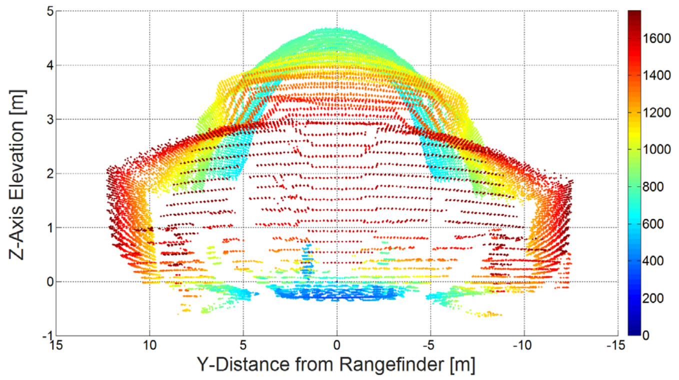


FIGURE 8. (TOP) FRONT VIEW REFERENCE PHOTO FOR AN AUDITORIUM CLASSROOM ON CAMPUS WITH ONE CHAIR PLACED ON TOP OF THE DESK AND (BOTTOM) POINT CLOUD GENERATED USING CONFIGURATION II.

It was at this point that Configuration II was constructed to increase the number of data points taken to around 15,000-20,000 up from about 3,000 data points in Configuration I. This upgraded system took ten minutes to collect the same picture location as Figure 6. Figure 7 illustrates that the service box on the wall to the right of the window is now more clearly seen jutting out of the wall along with the window itself becoming distinguishable. The walls are now discernable and a large cylindrical pipe near the ceiling is present. The respectively bright rectangular light can be (somewhat) seen lower in the image and closer to the rangefinder. Of importance, the filtering routine implemented in MATLAB removed data behind the window because it skewed the overall point cloud picture. In the point cloud figures moving forward, the legend color indicates the distance in [cm] from the rangefinder in all three-directions.

However, when attempting to capture a classroom on campus with numerous objects (Figure 8), respectively few distinguishing characteristics are seen. Except for the overall shape of the auditorium and the ceiling, there are not many recognizable features. Upon reviewing the Arduino code, it was found that there was a mistake in the electromagnet voltage specifications that limited the horizontal resolution of the point clouds. Many unique step numbers were counted as the same step that caused the resulting point clouds to have multiple points in one location. Furthermore, this version of the code incremented the vertical motors as a full rotation around the magnets. This caused a relatively large jump in the angle upwards when it could have been respectively smaller. Both Figure 7 and Figure 8 illustrate these issues with numerous points in the horizontal direction missing along with a reduced accuracy (i.e., jumps) in the vertical direction.

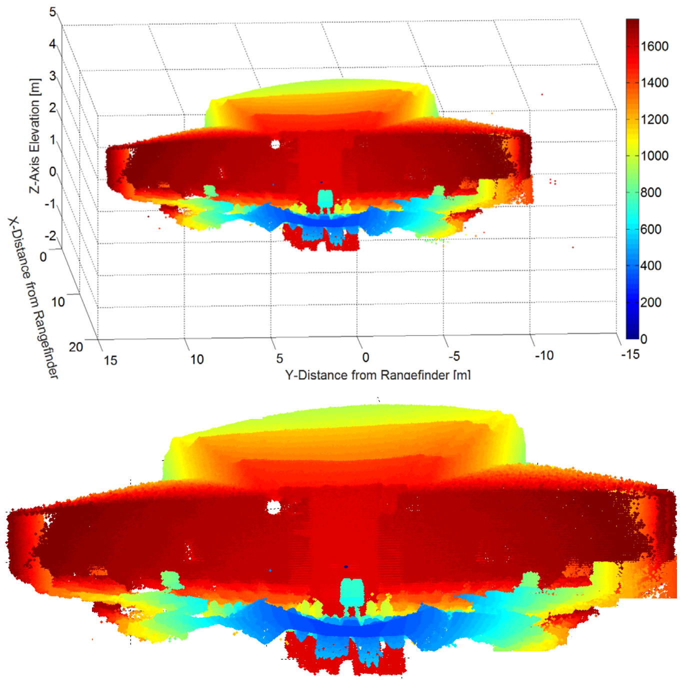


FIGURE 9. POINT CLOUD (TOP) WITH AXES AND (BOTTOM) WITHOUT AXES OF THE AUDITORIUM CLASSROOM IN FIGURE 8 AFTER IMPLEMENTING CODE UPGRADES TO CONFIGURATION II.

A subsequent upgrade to the code fixed the horizontal bug that augmented the resolution in this direction by seven times. Moreover, additional code was written to loop half steps between each electromagnet of the vertical motor. Rather than a full rotation around all four magnets, the motor rotated once between magnets one and two. After another horizontal sweep, the vertical motor then moved to magnet two. After another horizontal sweep, the motor moved between magnets two and three, and so on. This amplified the vertical resolution by seven times; hence, bringing the total resolution growth to forty-nine times the previous code. Unfortunately, this increased resol-ution created a data collection issue. After 300,000 data points are collected, the serial monitor within the Mega began deleting distance measurements collected from the beginning of a test. Currently, a third-party serial monitoring program (CoolTerm [19]) is installed in the laptop that uses the same communication port connected to the Arduino and writes these data directly to a text file. Ideally, direct communication between Arduino and MATLAB would allow MATLAB to read these serial monitor data and plot the point cloud in real time while fixing the data deletion issue.

Figure 9 presents the updated Configuration II point cloud for the same location as Figure 8. This data set took 130 minutes to create, contained over 700,000 points, and generated a text file with a size of 8 MB. Nearly all seats are clearly visible, especially those close to the front. Moreover, the chair placed on top of the desk in the middle of the classroom is seen clearly. On the left side of the auditorium and to the right of the left walkway, one outlet on every desk starting from the front and ending towards the back was lifted. While difficult to see in this figure, after expanding the image to a larger size, these outlets are shown as small bumps in the point cloud.

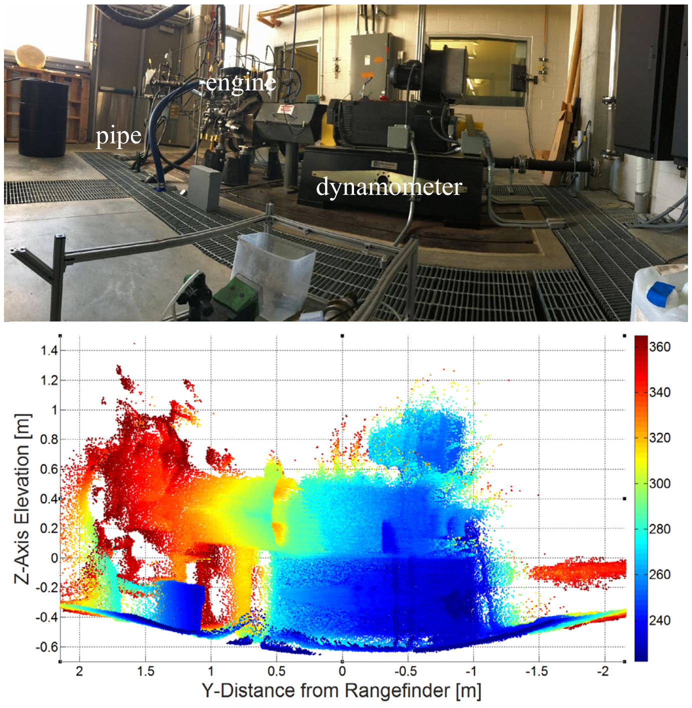


FIGURE 10. (TOP) REFERENCE IMAGE FOR THE MULTI-CYLINDER ENGINE TEST CELL ON CAMPUS AND (BOTTOM) THE CORRESPONDING POINT CLOUD.

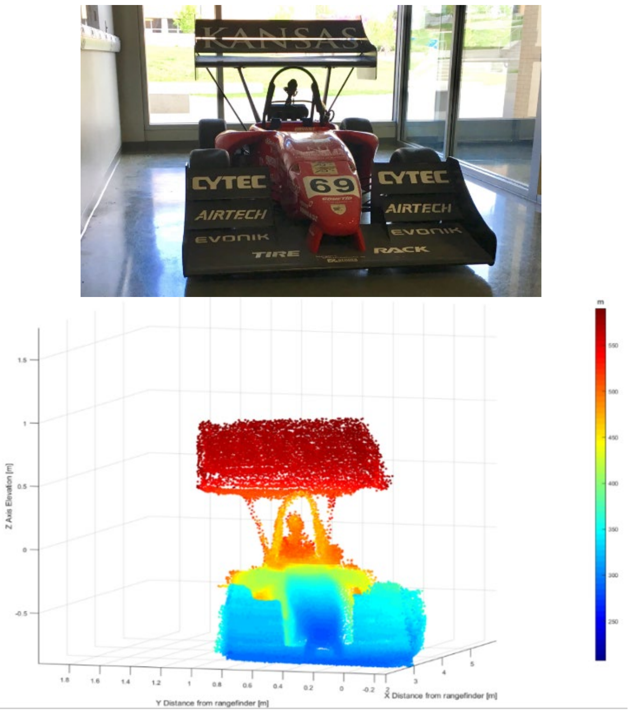


FIGURE 11. (TOP) REFERENCE IMAGE OF THE FORMULA SAE CAR AND (BOTTOM) THE RELATED POINT CLOUD.

This success led to another point cloud being taken of a multi-cylinder engine test cell on campus in Figure 10. This dataset took 85 minutes to create and data beyond a certain range were removed to better utilize color grading within MATLAB for the subjects of interest. After deletion, there remains about 200,000 points with the dynamometer on right hand side of the engine clearly seen. In addition, the curved pipe starting at the floor and ending at the engine is noticeable. It is important to note that only the default settings on the rangefinder were used; hence, configuring it to its short-range option might increase the detail in scenarios, such as Figure 10, where the objects are closer to the rangefinder.

To highlight how data post-processing can improve point cloud detection of the subject of interest, Figure 11 presents a picture and point cloud of a Formula SAE car. By strategically removing data points beyond a certain distance, the picture of the vehicle becomes rather recognizable. This demonstrates that successful lidar usage requires the fabrication of a capable hardware system coupled to efficient software routines.

Overall, this effort illustrates that higher resolution point clouds take significantly longer to create. Placing a system with this level of detail onto mobile platforms (e.g., electric bikes) where immediate knowledge of threats is needed appears unfeasible. Instead, like the previous effort, use of a rangefinder in conjunction with a camera can sweep an area significantly faster; hence, detecting vehicles more quickly along with the distance of that vehicle to alert riders of potential danger. Other possibilities include integrating this rangefinder with more extensive software algorithms that can track objects of interest [20]. However, this system appears suitable for delivering information for HPMS reports including, but not limited to: traffic information to mitigate roadway delays, accident/crash investigation, soil and rock slope stability, flood risk mapping, pavement quality monitoring, and clearance data for highway overpasses and power lines [21]. Since the total system cost is less than $300 (not considering the 3-D printed mount estimated at less than $30), it is possible to facilitate widespread implementation of lidar across the entire transportation infrastructure to enhance the information gathered. Finally, moving to a Raspberry Pi 3B+ microprocessor and implementing slip rings in the setup can help create a stand-alone system that is robust, fast, and, in combination with code upgrades that include interpolation between points, can generate high quality point clouds at a minimum expenditure.

4. CONCLUSIONS

The extensive application of lidar systems throughout the transportation infrastructure can facilitate a safer environment for travelers. These systems can enable the public to be aware of imminent threats while helping highlight critical areas in need of improvement and repair. However, current commercial lidar systems are relatively expensive, subsequently reducing their potential widespread feasibility. This effort endeavored to minimize expenditures when attempting to generate a lidar system of similar accuracy to commercial options. This was accomplished by utilizing a Garmin LIDAR-Lite v3 as the rangefinder and an Arduino Mega 2560 v3 microcontroller in combination with two stepper motors. Overall, it was possible to generate relatively accurate point clouds in MATLAB from ASCII text files with upwards of 700,000 data points. With a cost less than $300 (not including a 3-D printed mounting), this increases the possibility of wide-ranging implementation. Currently, this system is not suitable for mobile applications as data collection time took around 1-2 hours. Nevertheless, the system appears suitable for delivering information for public transport reports. Finally, potential upgrades to the system (e.g., microprocessor and slip rings) can further improve speed, robustness, and accuracy while not significantly growing its cost.

ACKNOWLEDGEMENTS

The research described in this paper is funded, in part, by the Mid-America Transportation Center via a grant from the U.S. Department of Transportation’s University Transportation Centers Program, and this support is gratefully acknowledged. The contents reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein, and are not necessarily representative of the sponsoring agencies.

REFERENCES

1. Xu, J.Q., Murphy, S. L., Kochanek, K. D., Bastian, B., and Arias, E., 2018, "Deaths: Final Data for 2016," National Vital Statistics Reports, **67**(5), National Center for Health Statistics: Hyattsville, MD.
2. National Center for Statistics and Analysis, 2018, "Summary of Motor Vehicle Crashes: 2016 Data," Traffic Safety Facts Report No. DOT HS 812 580, National Highway Traffic Safety Administration: Washington, DC.
3. Mole, C. D. and Wilkie, R.M., 2017, "Looking Forward to Safer HGVs: The Impact of Mirrors on Driver Reaction Times," Accident Analysis and Prevention, **107**, pp. 173-185 doi: 10.1016/j.aap.2017.07.027.
4. Goniewicz, K., Goniewicz, M., Pawłowski, W., and Fiedor, P., 2015, "Road Accident Rates: Strategies and Programmes for Improving Road Traffic Safety," European Journal of Trauma and Emergency Surgery, **42**(4), pp. 433-438, doi: 10.1007/s00068-015-0544-6.
5. Anarkooli, A. J., Hosseinpour, M., and Kardar, A., 2017, "Investigation of Factors Affecting the Injury Severity Of Single-Vehicle Rollover Crashes: A Random-Effects Generalized Ordered Probit Model," Accident Analysis and Prevention, **106**, pp. 399-410, doi: 10.1016/j.aap.2017.07.008.
6. American Society of Civil Engineers, 2019, "Infrastructure Report Card," Available from: https://www.infrastructurereportcard.org.
7. Puente, I., González-Jorge, H., Martínez-Sánchez, J., and Arias, P., 2013, "Review of Mobile Mapping and Surveying Technologies," Measurement, **46**(7), pp. 2127-2145, doi: 10.1016/j.measurement.2013.03.006.
8. Kelly, M. and Di Tommaso, S., 2015, "Mapping Forests with Lidar Provides Flexible, Accurate Data with Many Uses," California Agriculture, **69**(1), pp. 14-20, doi: 10.3733/ca.v069n01p14.
9. Garcia-Gutierrez, J., Gonçalves-Seco, L., and Riquelme-Santos, J. C., 2011, "Automatic Environmental Quality Assessment for Mixed-Land Zones using Lidar and Intelligent Techniques," Expert Systems with Applications, **38**(6), pp. 6805-6813, doi: 10.1016/j.eswa.2010.12.065.
10. Yang, B. S., Wei, Z., Li, Q., and Li, J., 2013, "Semiautomated Building Facade Footprint Extraction from Mobile LiDAR Point Clouds," IEEE Geoscience and Remote Sensing Letters, **10**(4), pp. 766-770, doi: 10.1109/LGRS.2012.2222342.
11. Chiang, K. W., Tsai, G.-J., Li, Y.-H., and El-Sheimy, N., 2017, "Development of LiDAR-Based UAV System for Environment Reconstruction," IEEE Geoscience and Remote Sensing Letters, **14**(10), pp. 1790-1794, doi: 10.1109/LGRS.2017.2736013.
12. Kromer, R. A., Hutchinson, D. J., Lato, M. J., Gauthier, D., and Edwards, T., 2015, "Identifying Rock Slope Failure Precursors Using LiDAR for Transportation Corridor Hazard Management," Engineering Geology, **195**, pp. 93-103, doi: 10.1016/j.enggeo.2015.05.012.
13. Neupane, S. R. and Gharaibeh, N. G., 2019, "A Heuristics-Based Method for Obtaining Road Surface Type Information from Mobile Lidar for Use in Network-Level Infrastructure Management," Measurement, **131**, pp. 664-670, doi: 10.1016/j.measurement.2018.09.015.
14. Lienert, P., and S. Nellis, 2019, "Cheaper Sensors Could Speed More Self-Driving Cars to Market by 2022," Available from: https://www.reuters.com/article/us-autos-autonomous-lidar/cheaper-sensors-could-speed-more-self-driving-cars-to-market-by-2022-idUSKCN1TD2MY
15. Blankenau, I., Zolotor, D., Choate, M., Jorns, A., Homann, Q., and Depcik, C., 2018, "Development of a Low-Cost LIDAR System for Bicycles," SAE Technical Paper 2018-01-1051, doi: 10.4271/2018-01-1051.
16. Garmin Ltd., 2018, "LIDAR-Lite Arduino Library," Available from: https://github.com/garmin/LIDARLite_Arduino_Library.
17. Arduino, 2019, "Servo Library," Available from: https://www.arduino.cc/en/reference/servo.
18. Arduino, 2019, "Stepper Library," Available from: https://www.arduino.cc/en/reference/stepper.
19. Meier, R., 2019, "Roger Meier's Freeware: CoolTerm," Available from: https://freeware.the-meiers.org/.
20. Jeon, W., and Rajamani, R., "Active Sensing on a Bicycle for Simultaneous Search and Tracking of Multiple Rear Vehicles," IEEE Transactions on Vehicular Technology, **68**(6), pp. 5295-5308, doi: 10.1109/TVT.2019.2911572.
21. Williams, K., Olsen, M. J., Roe, G. V., and Glennie, C., 2013, "Synthesis of Transportation Applications of Mobile LIDAR," Remote Sensing, **5**(9), pp. 4652-4692, doi: 10.3390/rs5094652.