2018-01-1051 Published 03 Apr 2018

# Development of a Low-Cost LIDAR System for Bicycles

**Isaac Blankenau, Daniel Zolotor, Matthew Choate, Alec Jorns, Quailan Homann, and Christopher Depcik**
University of Kansas

***Citation:*** Blankenau, I., Zolotor, D., Choate, M., Jorns, A. et al., "Development of a Low-Cost LIDAR System for Bicycles," SAE Technical Paper 2018-01-1051, 2018, doi:10.4271/2018-01-1051.

## Abstract

Transportation surveys illustrate that one of the most significant deterrents to bicycles as a form of conveyance is the concern with safety. Moreover, crash statistics also indicate that motor vehicles pose a severe risk to bicycles. As a result, this paper focuses on the development of a bicycle-mounted traffic monitoring system with the potential of providing early crash warnings to bicyclists. The system designed has a low monetary cost ($280.84) and is small enough to mount on a bicycle (94 mm × 56 mm × 89 mm). Moreover, it has sufficient range to track cars before they get dangerously close to the bike. The foundation of the system includes a Light Detection and Ranging (LIDAR) module that includes direct compatibility with microcontrollers. This LIDAR module interacts with a camera, stepper motor, and small computers through interfacing hardware and software. While robust, one limitation of the system is processing power. Specifically, its ability to detect cars is contingent on the performance of the computer that processes a video stream from the camera. With increased computing power, the system is capable of detecting lanes aiding in the search for cars, subsequently eliminating most false positives. Hence, augmenting processing capabilities of the current system would allow the vehicle recognition software to be more sensitive, ensuring that an automobile on the road is always detected.

## Introduction

Environmental and health issues within cities resulting from traffic emissions have led to some municipalities banning or restricting internal combustion engines [1, 2, 3]. In response, commuters often adapt by using bicycles and electric-assisted bicycles (e-bikes), subsequently making cycling more popular in urban areas [3, 4]. In addition to environmental benefits, many are urged to bicycle to improve health through exercise [1, 3]. While the large-scale adoption of bicycling as a primary source of transportation has tremendous potential to increase the quality of people's lives, it can only do so after mitigating the hazards that cyclists face [1].

Generally, several factors make people reluctant to use e-bikes and conventional bicycles as transportation. Weather and the impact cycling has on one's appearance can deter some [2, 5]. However, the concern for safety is the most substantial barrier to adopting cycling as primary means of transportation [6]. This potential for harm is attributed primarily to infrastructure, motorists, and the absence of protection for the cyclist [2]. In particular, cyclists in the United States (US) reported that motorists are their principal concern [7]. This is understandable considering that in the US, there were 818 cyclist fatalities and 45,000 cyclist injuries from motor vehicle-related accidents in 2015 [4]. Overall, the number of cyclist deaths per year has been increasing, with cyclist fatalities steadily becoming a more significant percentage of the total transportation fatalities [4, 8].

In this area, the enforcement of strict adherence to road rules for both cyclists and motorists will improve cyclist safety [9, 10, 11]. Additionally, competency and awareness can reduce the likelihood of collisions [9, 10, 12]. However, motorists are by no means the only reason for accidents as cyclists also have lapses in judgment. Specifically, many cyclists stop adhering to traffic laws when they are not held to the same standards as motorists [5, 13]. For instance, cyclists will continue riding even though there is a stop sign or stop light with those on e-bikes more likely to do so due to improved acceleration capabilities [5, 11, 13]. In addition, the existing infrastructure contributes to safety issues. Many bicyclists view integrated road conditions as four times more onerous than the environment in dedicated bike lanes; thus, there have been efforts to separate cyclists from motorists [14]. Hence, decreasing the interaction between cyclists and motorists improves cyclist safety, potentially through integrated bike lanes [15]. Of note, this modification does not prevent collisions in intersections and considerations must be made for the costs incurred. While a long-term infrastructural design shift will foster safer conditions for bicyclists, such changes are unlikely until cyclists represent a more significant portion of transportation [16, 17].

Therefore, while the infrastructure slowly evolves and adapts, an immediate solution is required to improve cyclists' safety. This answer depends on the conditions cyclists face and the shortcomings of current safety measures. Because of highly variable speeds, road surfaces, and live traffic

conditions, it can be difficult to maintain rear facing awareness [4]. A standard resolution to this problem is to install mirrors, on either the handlebars or helmet, to reduce the time taken by rearward observations. However, the field of view in mirrors is often limited and tends to offer poor depth perception. Moreover, mirrors provide intermittent performance by giving feedback only while being observed. Additionally, there is the added risk that the rider's attention is distracted from their front, which is a significant risk since 84% of cyclist fatalities occur from head-on collisions [4]. Instead, a rear-mounted system capable of continuously tracking motor vehicles along with their distances and speeds could provide an early warning system for cyclists, subsequently reducing the occurrence of accidents.

However, any system designed specifically for use on a bicycle faces unique constraints. It must be affordable and not negatively influence the ride experience. Taking lessons from the helmet, bicyclists tend to be reluctant to accept these costs in exchange for safety [18]. Therefore, reception hinges on providing a reasonable sense of security and reliability, all while reducing cost, weight, and maintenance. In consideration of these expectations, a Light Detection and Ranging (LIDAR)-based system is feasible, given its capabilities of both high speed and accurate monitoring of traffic situations with relatively low computational requirements for data processing [19, 20, 21, 22]. In this area, there have been several previous attempts to equip bicycles to monitor road conditions and improve safety.

As early as 2011, a team at Rutgers University began developing a computer vision system to detect cars [23]. In 2014, a team at Northeastern University created a distance based sensor system that would provide feedback to riders based on the distance of an object [24]. The system used a small array of stationary ultrasonic distance sensors situated on both the front and rear of the bike and feedback was presented through light and noise notifications. In the same year, Wallich built a system using a prior version of the LIDAR sensor that is used here. Employing an Arduino-based platform, he used LIDAR as the rangefinder to detect any oncoming traffic from the rear [25]. A few years later, a team from the University of Minnesota developed a multi-sensor bicycle safety system that included the same LIDAR element that is used in this project and mounted it to a stepper motor to add a second dimension of measurement [26]. Because of the low acquisition rate of the sensor, the team built an algorithm to track objects instead of measuring through a continuous sweep. Currently, Garmin has a commercial product available (Varia™) that uses radar to detect the presence and relative velocity of approaching traffic [27].

While all of these efforts had varying levels of success, there remains a fundamental need by the cycling community for a low-cost system that can effectively monitor traffic conditions and improve rider safety. Moreover, seeing as how most cyclist fatalities caused by vehicles occur at the front of the vehicle [28], cyclists are especially concerned with incoming hazards. Hence, monitoring vehicles from the rear of the bike will allow cyclists to better focus on navigation and oncoming potential issues while still being alert to rearward threats.

As a result, this effort describes the integration of inexpensive commercial microcontrollers with LIDAR based distance measurements for use on the rear of an e-bike. The following sections first describe the hardware and software of the system illustrating an iterative process at creating the least expensive solution while incorporating an open-platform software package for vehicle recognition. Subsequent testing on the rear of an e-bike finds successful automobile identification; however, processing limitations preclude on-road efforts. Therefore, this paper ends with a discussion of future upgrades required to handle traffic conditions.

# Hardware and Software

The base component of the system incorporates the LIDAR-Lite v3 module produced by Garmin. It is capable of communicating over either Serial Peripheral Interface (SPI) or Inter-integrated Circuit ($I^2C$) connections and can provide distance measurements with an accuracy of +/− 2.5 cm at a frequency of up to 500 Hz [29]. Because the LIDAR module can only perform one-dimensional measurements, it is mounted directly on a 400-count stepper motor (StepperOnline®, 14HR05-0504S) that traverses in a horizontal direction providing a second dimension to track targeted objects on the road. LIDAR calculations and its control are based upon the communication between two subsystems: a microcontroller and a small computer. Over the course of the research, two different subsystems were built. The first was designed primarily as a testbed and, therefore, each part was chosen for its versatility with lower priority placed on size and cost. Construction of the second system focused primarily on size and cost.

## Testbed Subsystem

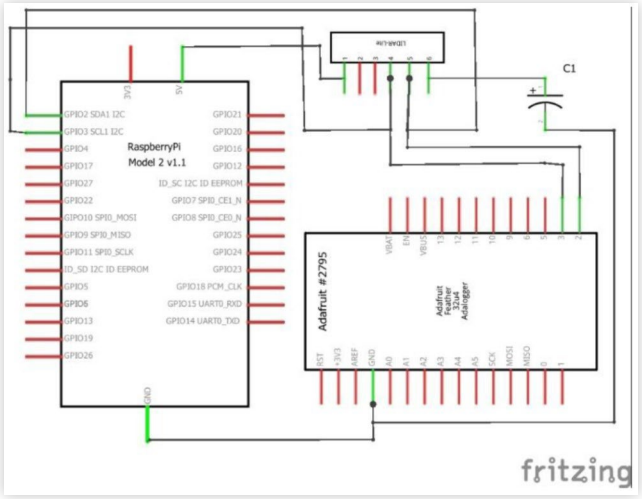
The microcontroller implemented in the first iteration was an Arduino Mega 2560 R3. This is an open source product built around the Atmega2560 8-bit Atmel Microcontroller and operates at 16 MHz. It is capable of powering sensors at either 3.3 or 5 VDC while requiring 7-12 VDC to run. It has four built-in hardware serial ports for expedient use with sensors; hence, it does not need to emulate serial ports with General-purpose input/output (GPIO) pins, which is considerably less efficient. Notably, a serial port emulator must process the data bit-by-bit, and since it does not contain a bus to save the incoming bits, it cannot process in clusters. This prohibits processing while data are being read.

The Raspberry Pi 3 Model B was chosen as the computer subsystem for the first design iteration. It is a single board computer optimized for running Raspbian, a Debian-based Linux distribution Operating System (OS) and is capable of running the Open Source Computer Vision Library (OpenCV) C++ software package [30]. This computer has a Quad-core 1.2 GHz Broadcom BCM2837 64-bit CPU with 1 GB of RAM. It comes with both Wi-Fi and Bluetooth modules built in, four Universal Serial Bus (USB) ports, two Camera Serial Interface (CSI) ports, an Ethernet port, an auxiliary (AUX) port, a High-Definition Multimedia Interface (HDMI) port, and 40 GPIO pins. It was integrated with Raspberry Pi's Camera Module v2 (Pi Cam) through one CSI connection to track cars through a live video stream [31]. The Pi Cam was chosen based on its compatibility and its low video quality configuration (480p), which is ideal for image processing with limited resources.

Communication between the Model B and Mega was conducted initially over a serial connection. However, the intermittent nature of the data produced by the Raspberry Pi resulted in slow and unreliable data transfer. The communication protocol was switched to $I^2C$, which provides simple short distance intra-board communication within a single system and only requires two signal wires from each board to exchange information [32]. Switching to $I^2C$ solved the serial connection issues while making data transfer quicker and more reliable.

Regarding this application, the primary downside to both the Mega and Model B are their cost and size. Despite being relatively inexpensive (both under $50), the goal of a bicycle mounted system provides unique constraints where cost and size are heavily weighted. After testing the initial system and providing its validity, a smaller and lower cost system was constructed with the intention of better suiting the design goals.

**FIGURE 1** Wiring schematic for the final revision of the LIDAR system.






## Final Subsystem

The first change was to implement the Adafruit Feather System as a replacement for the Arduino Mega. Feather is a complete line of development boards that are stackable, expandable, and Arduino programmable [33]. It is the platform for motor control, data logging, Global Positioning System (GPS) tracking, and application of the LIDAR sensor. Feather allows for the integration of most elements that are required for the project in predesigned chips. The master board is the Adafruit Feather 32u4 Adalogger that is an 'all-in-one' data logger with built-in USB and battery charging [34]. One of two wings for the master board is the Adafruit Direct Current (DC) Motor + Stepper FeatherWing [35]. It allows for the use of two bipolar stepper motors or four brushed DC motors (or one stepper and two DC motors) and is used here to control the 400-count stepper motor. The second wing is the Adafruit Ultimate GPS FeatherWing [36]. It provides a precise, sensitive, and low power GPS module for location identification anywhere in the world. It can also keep track of time after synchronizing with satellites using an inbuilt Real-Time-Clock (RTC).

Furthermore, the Raspberry Pi Model B was replaced in the second iteration with the Raspberry Pi Zero, currently the cheapest computer available at only $5. The Zero only has a single-core operating at 1 GHz with 512 MB of RAM while also not containing either Bluetooth or Wi-Fi modules. Because of the similarities in computer architecture between the Model B and the Zero, it was possible to transfer directly the Secure Digital (SD) card containing the memory and OS from the Model B. Because of this, the OpenCV software running on the Zero is similar to the OpenCV software developed for the Model B, with only a few minor changes to account for the low processing capabilities of the Zero. Communication between the Feather system and the Zero is achieved via the $I^2C$ serial bus in the same way as the first testbed. Figure 1 provides the final connection schematic between the Zero and Feather data logger.

Since the LIDAR, Zero, Feather, and motor control all operate over $I^2C$, it is essential to keep separate message addresses in order to maintain stable and reliable communication. At this time, the GPS sensor is used solely for an accurate timestamp and communicates over a serial connection; hence, it does not interfere with the $I^2C$ interface. This timestamp along with the most recent vehicle distance and angle measurements (discussed in the next section) are saved to an SD card for post-processing.

For the final system illustrated in Figure 2 and Figure 3, the stepper motor is powered by a battery pack consisting of eight AA batteries. This provides enough voltage for the motor controller to run reliably, while also lasting long enough for extended testing. This battery pack is connected to the motor controller via a 9 VDC socket connector so power can be quickly cut off when testing is not taking place. The stacked Feather system runs on a 3.7 VDC lithium polymer battery (DataPower [37]) using a SubMiniature version A (SMA) connection integrated onto the Adalogger board. A 5 VDC USB powers the Zero from the 4000 mAh, 5 VDC external

**FIGURE 2** Garmin LIDAR-Lite v3 (right top) mounted to stepper motor (middle). The Raspberry Pi Cam (right middle) is connected via a ribbon cable to the Adafruit Feather stack (left middle).

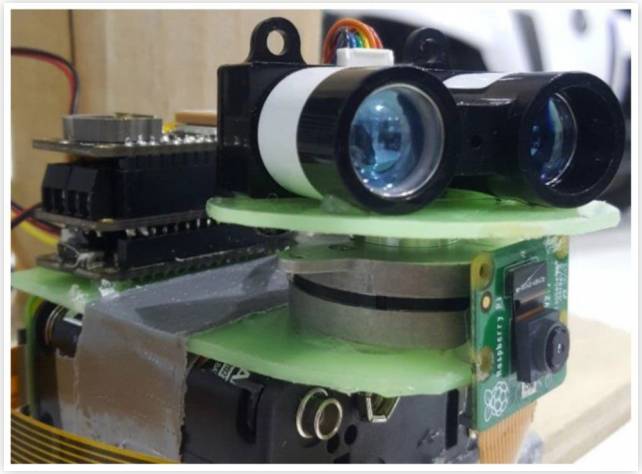
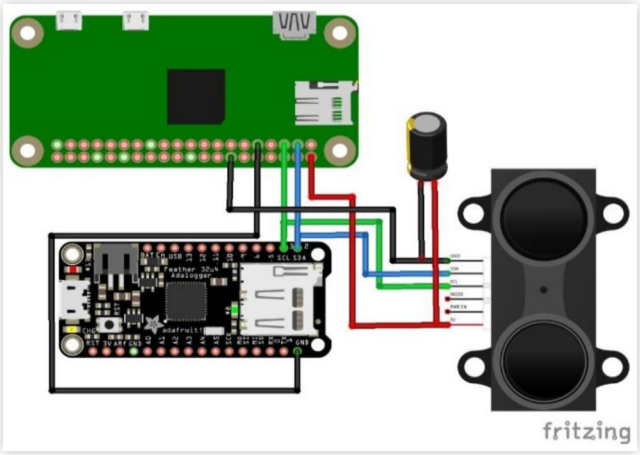


**FIGURE 3** Simplified connection diagram joining the Raspberry Pi Zero (top) with the Feather stack (bottom left), a 680 μF capacitor specified by Garmin to regulate LIDAR power requirements (middle right), and Garmin LIDAR-Lite v3 (bottom right).






battery (Prime Line PL-1365), initially intended for charging cell phones and tablets.

## Vehicle Recognition

Vehicle recognition is achieved via the OpenCV software package, as previously mentioned, with the authors' code provided in the following reference [38]. When the Zero starts, a Python script begins and runs in the background. Then, the vehicle detection program begins, and the video stream from the Pi Cam opens. Because the camera is moving with the bike, a background subtraction algorithm cannot be used. Instead, a cascade (via an Extensible Markup Language (XML) file) is loaded into the program. This cascade is an image classifier that was trained by feeding over a thousand positive and negative samples of cars [39]. Subsequently, each frame of the video stream is passed to the classifier, and if a car is found, the car's attributes are added to a list. Specifically, the location of each car from the left-hand side of the screen is converted to an angle in degrees and added to a list of angles. The list is sorted, and the smallest angle is saved to a text file. This angle has +/− 1° of uncertainty due to the non-uniform curvature of the camera's lens. Then, the Python script looks for a change in the angle stored in the file and sends it to the Feather (as an integer value proxy) over $I^2C$ if one is found. These values determine if a car expected; hence, a boolean value of one (zero otherwise) is saved alongside the measurements from the Feather stack at the input angle. Since the stepper motor has no feedback indicating its current position, the program calibrates itself periodically by continuously rotating in one direction on a low torque setting, butting up against a stopper at a known angle.

In an effort to decrease the number of false positives, lane detection can be implemented, providing a region of interest to search for cars. If a lane is detected, its vanishing point will be calculated, and a region of interest will be defined: the area bounded by the lines tangent to the outermost lane markings and the road's vanishing point. All area outside of this region will become black in the frame, and then it will search for cars in this new frame with a higher tolerance. If a lane is unable to be detected, the program will run as described in the prior paragraph. For the testbed subsystem, lane detection was tried with limited success. Hence, it was removed for the final subsystem due to Zero's processing limitations.

## Results and Discussion

Initial testing and troubleshooting of the testbed system were conducted by substituting a video stream of traffic in lieu of live video collected by the Pi Cam. This method expedited adjustments to both the OpenCV program sensitivity and the communication between the subsystems. Subsequently, to demonstrate that the final system is capable of both detecting and determining the position of cars, a stationary test was conducted utilizing a single vehicle (2000 Infinity G20). Specifically, the front of the vehicle was directed toward the system in an otherwise empty parking lot, simplifying the process of taking manual measurements.

At a height of 3 feet, LIDAR measurements were found to be unreliable. This is likely due to the angle and reflectivity of the hood and windshield, which reflects light away from the light detection sensor. This is to be expected given the acknowledgment within the Garmin LIDAR manual that unless the sensor is normal to a specular surface, it will be incapable of taking accurate measurements. [29]. Furthermore, smooth reflective surfaces may not disperse light back towards the receiver [40, 41]. However, by lowering the system height to 1.5 feet, the results became significantly more dependable. At this height, the front of the car is at an angle more conducive to the LIDAR sensor, subsequently allowing for accurate measurements.

Data were then recorded for ten different car positions as a function of distance from the LIDAR system to the center of the vehicle as shown in Table 1. The direct distances and angle measurements were found by finding the x and y

**TABLE 1** Measurement data from the stationary test that describes the vehicle position as found by both direct measurement with a measuring tape (columns 2 and 3) and as determined by the LIDAR system (columns 4 and 5).

| Test | Measured Distance [in] | Measured Angle [deg] | LIDAR Distance [in] | Raspberry Pi Angle [deg] |
|---|---|---|---|---|
| 1 | 292.3 | 40.95 | 298 | 41 |
| 2 | 322.0 | 57.67 | 334 | 58 |
| 3 | 227.8 | 18.16 | 226 | 18 |
| 4 | 152.8 | 11.60 | 155 | 15 |
| 5 | 497.0 | 31.33 | 509 | 31 |
| 6 | 385.0 | 31.70 | 395 | 30 |
| 7 | 259.0 | 31.10 | 265 | 34 |
| 8 | 673.1 | 32.04 | 660 | 31 |
| 9 | 623.9 | 37.54 | 588 | 39 |
| 10 | 119.9 | 47.57 | 130 | 48 |

distances to the vehicle using a tape measure and exercising Pythagorean's theorem and trigonometry, respectively. The LIDAR distance in this table is the distance to the vehicle as determined by the corresponding sensor. Furthermore, the OpenCV program determined the Raspberry Pi Angle measurements in this table. Figure 4 plots these positions on an x-y plane and it is important to note that the frontal area of the car cannot be accurately described as a single point. Hence, in Figure 4 a line is drawn between the measurement taken at the leftmost and rightmost positions of the front of the vehicle. Moreover, solid circles in this figure provide the positions of the vehicle as determined by the LIDAR system. Overall, the LIDAR system appears to find the vehicle successfully. Moreover, accuracy can be determined by the proximity of each point in comparison to the front center of the vehicle. On average, the LIDAR system found the center of the vehicle with 82.3% and 96.7% accuracy in the x- and y-directions, respectively. Here, the deviations between the indirect and direct measurements may be a function of the curvature of the front of the car along with the relative reflectivity of the lights and grill.

Subsequently, dynamic tests helped to evaluate the performance of the system in active situations. The first test consisted of mounting the LIDAR system on the back of an e-bike at a height of 1.5′ and riding through a parking lot past parked cars while also being followed by a car (2000 Oldsmobile Bravada). The path chosen consisted of riding up one lane of the parking lot and then down the adjacent lane. During the ride, the LIDAR sensor rotates through 75 unique angle measurements from 0 to 67.5°, where 33.75° is designated as the angle directly behind the e-bike. Henceforth, a sweep refers to the LIDAR rotating from 0 to 67.5°, then returning to 0°.

Figure 5 provides insight into the operation of the LIDAR system. Specifically, as the LIDAR-Lite performed a sweep

**FIGURE 4** Position plot of the stationary test. Points indicate the coordinates of the car as determined by the LIDAR system. The lines show the frontal span of the vehicle as measured directly.

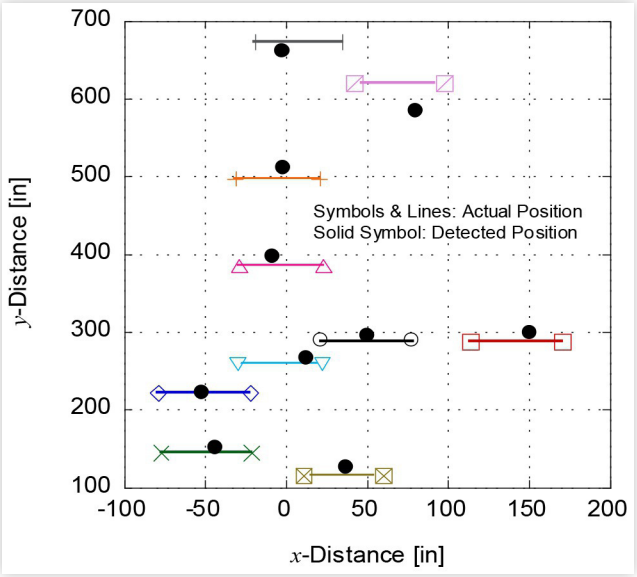
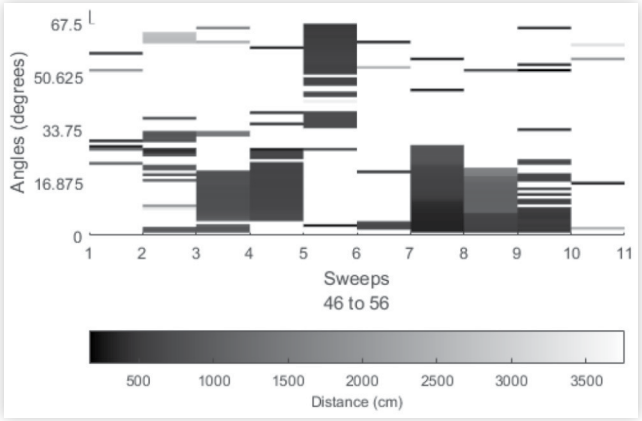





**FIGURE 5** LIDAR system data while being followed by a car during the period of 106 to 131 seconds.






(*x*-axis), it captured the distance of surrounding vehicles (color bar) along with their relative angle to the e-bike (*y*-axis). Furthermore, darker bars represent objects nearer to the LIDAR system. Of note, data below 1 m was removed because the manual states there is non-linearity in measurements below this level and visual inspection via OpenCV photos did not find anything. Overall, determination of the relative movement of objects occurs through the change in angle and distance of these objects between consecutive sweeps. However, unlike its potential fixed location on an automobile, the relative orientation of the LIDAR camera changes dynamically during a ride as the e-bike jostles, rolls, and sways. Hence, reviewing the data from Figure 5 illustrates that it becomes difficult to determine definitive locations of vehicles without visual cues (e.g., OpenCV photos). In other words, the three-dimensional space continuously varies as the horizon seemingly tilts. Therefore, differentiating between objects becomes onerous and the implementation of an accelerometer or gyroscope synced to the LIDAR data should help with fidelity.

In order to improve clarity behind how the LIDAR system operates, Figure 6 provides an array of data during the small window of time corresponding to the first sweep in Figure 5. Each sweep takes approximately 2.45 seconds, which may lead to discrepancies while measuring the same object. Moreover, this makes comparisons to photo taken by the Pi Cam difficult. Specifically, this camera takes photos nearly instantaneously as compared to the duration of a sweep; however, the OpenCV software is not quick enough to analyze photos at the same frequency as the LIDAR sensor. For example, the image corresponding to approximately the same time as sweep 46 is shown in Figure 7 with approximate LIDAR angles indicated on the picture. Here, sweep 46 begins before taking the photo and the LIDAR sensor returns zero values when it does not find an object within its range of 40 m.

The first two peaks from the left in Figure 6 appear to correspond to the cars (green arrows) on the left edge of Figure 7 and the tallest peak should link to the distant car (yellow arrow) just left of the Oldsmobile Bravada. As anticipated, the LIDAR sensor captures other objects beyond vehicles, as it appears the light pole next to the vehicle highlighted with a blue arrow appears in these data. In addition, the distance

**FIGURE 6** LIDAR system data from the 46th sweep, or at approximately 106 seconds.

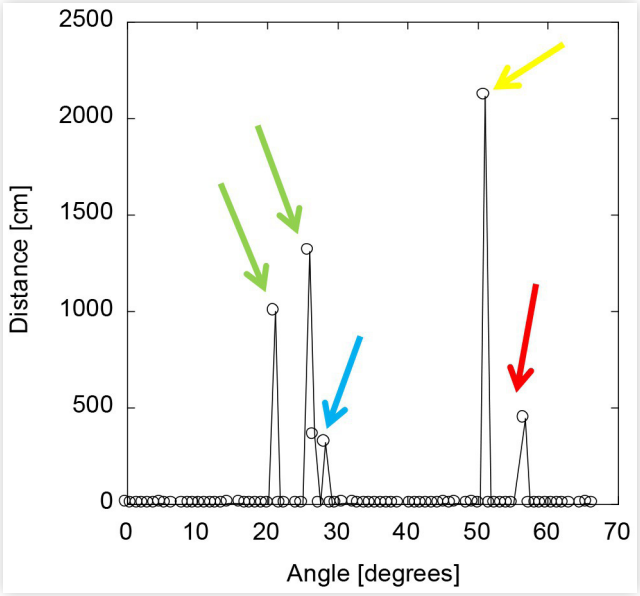
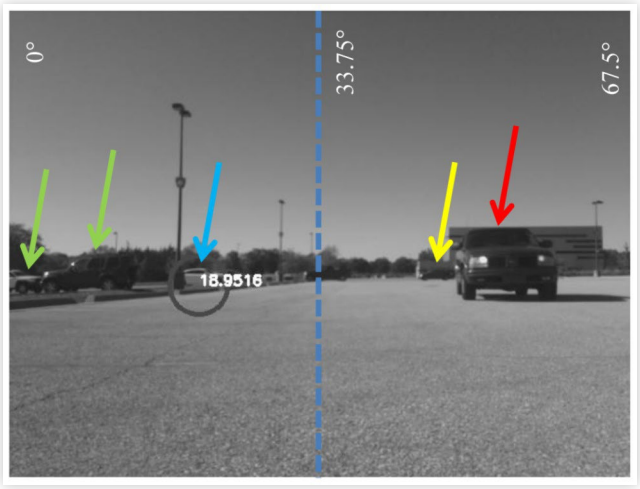





**FIGURE 7** Photo taken by the OpenCV software at 106 seconds into testing (about 4.05° into the 46th sweep). In this frame, the central vehicle is missed by the software but is picked up by the LIDAR.






measurements compared with the photo for the blue and green highlighted vehicles do not seem correct. This may be because the LIDAR sensor is sweeping through the range and the photo only illustrates a snapshot of this sweep. For instance, the LIDAR sensor sweeps from left to right; hence, the green highlighted vehicles would be closer initially as the e-bike travels away from them. Moreover, while the LIDAR sensor appears to provide fidelity in picking up all vehicles, the OpenCV software misses all but one since its sensitivity is too low and insufficient processing speed was provided. Hence, future work should ensure that LIDAR sensor recognition and OpenCV data collection occur at the same frequency.

**FIGURE 8** Photo taken by the OpenCV software at 239 seconds into testing. Limitations in sensitivity lead to inaccurate vehicle identification. Example of bicycle sway and tilt shown.



Interestingly, because of fixed system integration on the e-bike, both the LIDAR sensor and OpenCV recognition software will experience the same roll as the bicycle. However, this should not affect the photos as much as the LIDAR measurements because photos cover a much larger area. Instead, the roll and pitch of the e-bike will make the LIDAR sensor less likely to detect an object at the bounds of its rotation (see Figure 8). Furthermore, the OpenCV system will be less capable of detecting vehicles when not level, because of limitations in learning the appearance of a vehicle. Similarly, as mentioned prior, the sensitivity set on the OpenCV system has a substantial impact on its vehicle detection accuracy. Limitations on computing power led to a reduction in sensitivity of the OpenCV system. Hence, this is why it misses detecting some vehicles (e.g., Figure 7 & 8), and occasionally other objects are labeled as vehicles, as seen in Figure 9.

**FIGURE 9** Photo taken by the OpenCV software at 201 seconds into testing. Not all objects identified OpenCV are vehicles.

These issues should not be as prevalent for automotive LIDAR usage; however, systems designed for bicycles and e-bikes will need to account for these facets because of their unique constraints (e.g., low cost and small size). As a result of the found limitations, it was decided to forgo experimentation in real collision scenarios and instead focus on enhancement of the system for future efforts.

## System Diagnosis

A significant drawback of the camera vision system running on the reduced processing power of the Raspberry Pi Zero is the relatively low framerate that the system is capable of handling. Specifically, the video stream was processed at a rate of one frame every 2.1 seconds during testing. A processing rate this low can lead to the issue of insufficient reaction time for the rider to vehicular threats. Furthermore, with a LIDAR range of 40 m, any vehicle traveling over 19 m/s (42.5 mph) relative to the e-bike could potentially move through the entire detection range without being sensed.

Ideally, when the subsystems are effectively communicating, the video feed sends angle measurements directly to the Arduino-based subsystem for subsequent saving. However, communication between the Feather and the Zero proved to be unreliable and would often cause the Feather system to crash. Hence, testing did not employ the interface between these subsystems, consequently requiring a combining of data in post-processing as mentioned prior.

Moreover, it takes an average of 2.45 seconds for the LIDAR system to make all 75 unique angle measurements; i.e., distances at each angle are found only once every 2.45 seconds. This low repeat frequency is undesirable for the same reasons the low framerate is detrimental: a vehicle can go undetected for too long. Furthermore, the relatively low memory and processing speeds of the Feather contributes significantly to the process times. Specifically, data are written to the SD card once every 15 measurements. Therefore, it takes on average 0.07 s to save the data to the SD card, which means that 0.4 s of every rotation can be eliminated by using a microcontroller with more embedded memory.

Finally, the limiting factor in the system is the maximum frequency of the LIDAR-lite v3 at 500 Hz. Hence, the minimum time possible for the system to complete full cycle would be 0.15 s. Since it takes approximately 0.5 s for the stepper motor to traverse 75 angles, the system is currently set up to complete a sweep in less than one second. However, processing the data leads to the greater time mentioned prior. Therefore, to equate the system speed to that of the LIDAR sensor, a different stepper motor is required. To increase speed beyond this level, multiple LIDAR sensors would need to be employed, or the number of measurements per sweep would have to be lowered.

# Conclusion

Safety concerns deter many people from cycling, but currently available solutions have difficulty gaining traction because of unique constraints. Specifically, riders wish to employ affordable and unobtrusive solutions that do not negatively influence their experience. In this area, LIDAR-based systems provide an opportunity to improve safety dramatically from the rear because of their high speed and precision monitoring capabilities at relatively low computation requirements. Furthermore, recent advances in sensor technology are beginning to bring costs into a range that allows implementation on all bicycles and e-bikes.

As a result, this effort endeavored to generate the lowest cost system possible using the Garmin LIDAR-Lite v3 module interacting with Adafruit Feather boards and a Raspberry Pi Zero as the microcontroller and computer, respectively. This system can measure the distance of a stationary vehicle accurately after training the OpenCV software package running on the Zero. However, subsequent dynamic tests found limited success as the LIDAR sensor was able to find surrounding vehicles; whereas, the limited processing capability of the Zero dramatically reduced OpenCV's ability. Moreover, data provided by the LIDAR sensor are difficult to analyze to find moving vehicles given its recognition capacity; i.e., it notices many items beyond just vehicles. Furthermore, at a proposed cost of $280, the system (while cheaper than many commercial LIDAR sensors) is still too expensive for use by the cycling community.

Overall, this novel integration and application of existing microcontrollers in tracking vehicles has applicability beyond cyclist safety and into areas, such as unmanned aerial vehicles and robotics, where object detection can be crucial. In addition, this study demonstrates that the employed methodology has limited success when price constraints (currently) cause reduced computational capabilities. Hence, future efforts should enhance processing capabilities to provide a better linking of the LIDAR sensor and OpenCV recognition software in order to remove false vehicle positives while endeavoring to do so at a minimum of expense.

## Acknowledgments

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. National Safe Routes to School Task Force, "Safe Routes to School: A Transportation Legacy," Washington, D.C.: U.S. Department of Transportation, 2008, Available from: http://archive.saferoutesinfo.org//sites/default/files/task_force_report.web_.pdf.
2. Weinert, J., Ma, C., Yang, X., and Cherry, C., "Electric Two-Wheelers in China: Effect on Travel Behavior, Mode Shift, and User Safety Perceptions in a Medium-Sized City," *Transportation Research Record: Journal of the*

*Transportation Research Board* 2038:62-68, 2007, doi:10.3141/2038-08.

3. Rose, G., "E-Bikes and Urban Transportation: Emerging Issues and Unresolved Questions," *Transportation* 39(1):81-96, 2012, doi:10.1007/s11116-011-9328-y.
4. National Highway Traffic Safety Administration, "Quick Facts 2015 (Updated)," 2016, Available from: https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812348.
5. Du, W., Yang, J., Powis, B., Zheng, X. et al., "Understanding On-Road Practices of Electric Bike Riders: An Observational Study in a Developed City of China," *Accident Analysis & Prevention*. 59:319-326, 2013, doi:10.1016/j.aap.2013.06.011.
6. Götschi, T., Garrard, J., and Giles-Corti, B., "Cycling as a Part of Daily Life: A Review of Health Perspectives," *Transport Reviews* 36(1):45-71, 2016, doi:10.1080/01441647.2015.1057877.
7. Schroeder, P. and Wlibur, M., "*National Survey of Bicyclist and Pedestrian Attitudes and Behavior, Volume 2: Findings Report*," (Washington, DC, National Highway Traffic Safety Administration, 2012) Available from: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/811841b.pdf.
8. Bureau of Transportation Statistics, "Chapter 6: Transportation Safety," In Transportation Statistics Annual Report. Washington, D.C. : U.S. Department of Transportation, 2016, Available from: https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/transportation_statistics_annual_report/2016/chapter_6.
9. Räsänen, M., Koivisto, I., and Summala, H., "Car Driver and Bicyclist Behavior at Bicycle Crossings under Different Priority Regulations," *Journal of Safety Research* 30(1):67-77, 1999, doi:10.1016/S0022-4375(98)00062-0.
10. Summala, H., Pasanen, E., Räsänen, M., and Sievänen, J., "Bicycle Accidents and Drivers' Visual Search at Left and Right Turns," *Accident Analysis & Prevention* 28(2):147-153, 1996, doi:10.1016/0001-4575(95)00041-0.
11. Yang, X., Huan, M., Abdel-Aty, M., Peng, Y., and Gao, Z., "A Hazard-Based Duration Model for Analyzing Crossing Behavior of Cyclists and Electric Bike Riders at Signalized Intersections," *Accident Analysis & Prevention* 74:33-41, 2015, doi:10.1016/j.aap.2014.10.014.
12. Walker, I., "Signals Are Informative but Slow Down Responses when Drivers Meet Bicyclists at Road Junctions," *Accident Analysis & Prevention* 37(6):1074-1085, 2005, doi:10.1016/j.aap.2005.06.005.
13. Wu, C., Yao, L., and Zhang, K., "The Red-Light Running Behavior of Electric Bike Riders and Cyclists at Urban Intersections in China: An Observational Study," *Accident Analysis & Prevention* 49:186-192, 2012, doi:10.1016/j.aap.2011.06.001.
14. Hunt, J.D. and Abraham, J.E., "Influences on Bicycle Use," *Transportation* 34(4):453-470, 2007, doi:10.1007/s11116-006-9109-1.
15. DiGioia, J., Watkins, K.E., Xu, Y., Rodgers, M., and Guensler, R., "Safety Impacts of Bicycle Infrastructure: A Critical Review," *Journal of Safety Research* 61:105-119, 2017, doi:10.1016/j.jsr.2017.02.015.
16. Macmillan, A., Connor, J., Witten, K., Kearns, R. et al., "The Societal Costs and Benefits of Commuter Bicycling: Simulating the Effects of Specific Policies Using System Dynamics Modeling," *Environmental Health Perspectives* 122:335-344, 2014, doi:10.1289/ehp.1307250.
17. Flusche, D., "*The Economic Benefits of Bicycle Infrastructure Investments*," (Washington, DC, League of American Bicyclists, 2009) Available from: https://www.aarp.org/content/dam/aarp/livable-communities/old-learn/transportation/economic-benefits-bicycle-infrastructure-report.pdf.
18. Finnoff, J.T., Laskowski, E.R., Altman, K.L., and Diehl, N.N., "Barriers to Bicycle Helmet Use," *Pediatrics* 108(1):e4-e4, 2001, doi:10.1542/peds.108.1.e4.
19. Williams, K., Olsen, M., Roe, G., and Glennie, C., "Synthesis of Transportation Applications of Mobile LIDAR," *Remote Sensing* 5(9):4652, 2013, doi:10.3390/rs5094652.
20. Puente, I., González-Jorge, H., Martínez-Sánchez, J., and Arias, P., "Review of Mobile Mapping and Surveying Technologies," *Measurement* 46(7):2127-2145, 2013, doi:10.1016/j.measurement.2013.03.006.
21. Glennie, C., "Kinematic Terrestrial Light-Detection and Ranging System for Scanning," *Transportation Research Record: Journal of the Transportation Research Board* 2105:135-141, 2009, doi:10.3141/2105-17.
22. Jeon, W. and Rajamani, R., "Active Sensing on a Bicycle for Accurate Tracking of Rear Vehicle Maneuvers," in *ASME 2016 Dynamic Systems and Control Conference*, Paper No. DSCC2016-9772, 2016, doi:10.1115/dscc2016-9772.
23. Smaldone, S., Tonde, C., Ananthanarayanan, V.K., Elgammal, A., and Iftode, L., "The Cyber-Physical Bike: A Step Towards Safer Green Transportation," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications ACM*, Phoenix AZ, 2011, doi:10.1145/2184489.2184502.
24. Castellanos, S., "Northeastern Students Develop 'Smart Bike' Tech to Curb Cycling Deaths," 2014, Available from: http://www.bizjournals.com/boston/blog/startups/2014/01/northeastern-students-develop-smart.html.
25. Wallich, P., "An Early-Warning System for your Bike: Low-cost LIDAR Can Detect Approaching Cars," *Spectrum*,IEEE.org, 2015, doi:10.1109/MSPEC.2015.7131685.
26. Woongsun, J. and Rajamani, R., "A Novel Collision Avoidance System for Bicycles," in *2016 American Control Conference (ACC)*, Boston, MA, 2016, doi:10.1109/acc.2016.7525451.
27. Garmin, "Varia™ Rearview Radar," 2017, Available from: https://buy.garmin.com/en-US/US/p/518151.
28. NHTSA's National Center for Statistics and Analysis, "Traffic Safety Facts: Bicyclists and Other Cyclists," 2017, Available from: https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812382.
29. Garmin International Inc., "Lidar Lite v3 Operation Manual and Technical Specifications," 2017, Available from: https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf.
30. OpenCV team, "OpenCV Library," 2017, Available from: http://opencv.org/.
31. Raspberry Pi, "Raspberry Pi Camera Module," 2017, Available from: https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/RPiCamMod2.pdf.

32. NXP Semiconductors, "UM10204: I²C-Bus Specification and User Manual," 2014, Available from: https://www.nxp.com/docs/en/user-guide/UM10204.pdf.
33. Adafruit, "Feather - A Complete Line of Development Boards from Adafruit that are Both Standalone and Stackable," 2017, Available from: https://www.adafruit.com/feather.
34. Adafruit, "Adafruit Feather 32u4 Adalogger ID: 2795," 2017, Available from: https://www.adafruit.com/product/2795.
35. Adafruit, "DC Motor + Stepper FeatherWing Add-on for All Feather Boards ID: 2927," 2017, Available from: https://www.adafruit.com/product/2927.
36. Adafruit, "Adafruit Ultimate GPS FeatherWing ID: 3133," 2017, Available from: https://www.adafruit.com/product/3133.
37. DataPower Technology Limited, "Product Specifications: Polymer Li-Ion Rechargeable Battery," 2015, Available from https://cdn.sparkfun.com/datasheets/Prototyping/spe-00-502535-400mah-en-1.0ver.pdf.
38. Zolotor, D., "Bicycle-Mounted Vehicle Tracking System," 2017, Available from: https://github.com/dzolotor/LidarCV.
39. Fergus, R., Perona, P., and Zisserman, A., "Object Class Recognition by Unsupervised Scale-Invariant Learning." in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003, doi:10.1109/CVPR.2003.1211479.
40. Beraldin, J.-A., Blais, F., and Lohr, U., "Three Dimensional Laser Scanning Technology,". In: Vosselman G., Mass H.-G., editors. *Airborne and Terrestrial Laser Scanning.* (Scotland, UK, Whittles Publishers, 2010).
41. Lichti, D. and Skaloud, J., "Registration and Calibration,". In: Vosselman G., Mass H.-G., editors. *Airborne and Terrestrial Laser Scanning.* (Scotland, UK, Whittles Publishers, 2010).


## Contact Information


**Christopher Depcik**
Associate Professor
Department of Mechanical Engineering
depcik@ku.edu
Ph: 785--864-4151
3144C Learned Hall, 1530 W. 15th Street
Lawrence, KS, 66045-4709


## Acronyms

**CPU** - Central Processing Unit
**CSI** - Camera Serial Interface
**GPIO** - General Purpose Input/Output
**GPS** - Global Positioning System
**HDMI** - High-Definition Multimedia Interface
**I2C** - Inter-Integrated Circuit
**LIDAR** - Light Detection And Ranging
**MLS** - Mobile Laser Scanning
**OS** - Operating System
**RAM** - Random-Access Memory
**RTC** - Real-Time-Clock
**SD** - Secure Digital
**SMA** - SubMiniature Version A
**SPI** - Serial Peripheral Interface
**USB** - Universal Serial Bus
**XML** - Extensible Markup Language



ISSN 0148-7191