Unmanned Aerial Vehicle Black Box

By
John Diebold

A THESIS

Submitted to
Oregon State University
University Honors College

In partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Mechanical Engineering
(Honors Scholar)

Presented May 26th, 2016
Commencement June 2016

AN ABSTRACT OF THE THESIS OF

John Diebold for the degree of Honors Baccalaureate of Science in Mechanical Engineering presented on May 26, 2016. Title: Unmanned Aerial Vehicle Black Box .

Abstract approved: ____

Geoff Hollinger

This project created a stand-alone unit capable of providing on-board sensing and processing for robotic vehicles. This unit was designed to be compact, lightweight, inexpensive, and adaptable. The goal of this project is to help advance robotics research in mapping and multi-agent coordination by making it easier to equip vehicles to perform mapping and localization functions. Software integration for the sensing components was accomplished using the LSD-SLAM (Simultaneous Localization and Mapping) Package in ROS (Robot Operating System) and localization testing was done to verify functionality.

Key Words: UAV, Robotics, Mapping, Localization, SLAM, Sensing, Processing

Corresponding e-mail address: dieboldj@oregonstate.edu

Honors Baccalaureate of Science in Mechanical Engineering project of John Diebold presented on May 26, 2016.

APPROVED:

Geoff Hollinger, Mentor, representing Mechanical Engineering

Daniel Gillins, Committee Member, representing Civil and Construction Engineering

Brian Whiteside, Committee Member, representing VDOS Global

Toni Doolen, Dean, University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

John Diebold, Author

Acknowledgements

I would like to acknowledge and thank my mentor, Dr. Geoff Hollinger, for providing me with this research opportunity and for all of his support along the way. I would also like to thank my committee members Dr. Daniel Gillins and Brian Whiteside for their time and input. Thank you to Ryan Skeele for his help and guidance with the technical details of the project. Also thanks to the other students in the Robot Decision Making Lab for their support.

This research has been funded in part by Department of the Air Force award number FA8651-14-c-0135 and NASA award number NNX14AI10G.

Table of Contents

1. Introduction

Currently in the Robotic Decision Making Laboratory (RDML), several unmanned aerial vehicles (UAVs) are used for research purposes. Due to the nature of the research, these vehicles must be equipped with on-board computer processing and a variety of sensors. Commercial solutions either do not meet the requirements of the lab or are prohibitively expensive. When a new vehicle is acquired by the lab, a significant portion of time must be spent wiring, programming, and testing these processors and sensors. The creation of a box with processing and sensing capabilities that could be easily attached to aerial vehicles would decrease setup time and costs for research projects in the lab.

Previously, the lab has used custom built quadcopters for mapping and coordination research. This has several advantages over buying commercial products. Custom-made UAVs are generally significantly cheaper than comparable commercial models. They also allow for a much greater degree of customization. The main drawback to custom built vehicles is the design and manufacturing time needed to produce them. It can also be difficult to find or manufacture replacement parts and perform maintenance when needed.

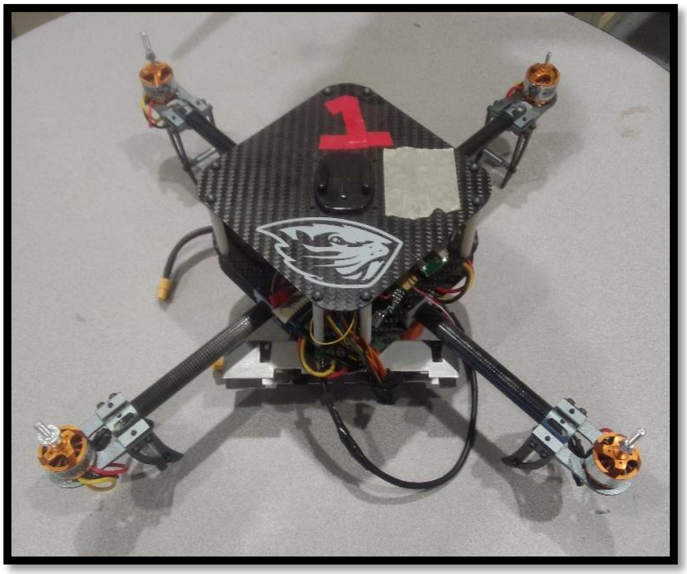


Figure 1.1: A custom made quadcopter in the RDML lab.

Shortly before the start of this project, the lab purchased three DJI Matrice 100 UAVs for research purposes. The DJI Matrice is a quadcopter that was designed to be useful for developers. It has multiple expansion bays and universal communication ports to allow the flight controller to connect to other devices [1]. The lab also purchased DJI guidance systems for all three vehicles. The guidance system attaches to the underside of the Matrice and consists of five sensor modules, one downward facing and one on each of the four sides. Each sensor module contains a camera and two ultrasonic sensors.

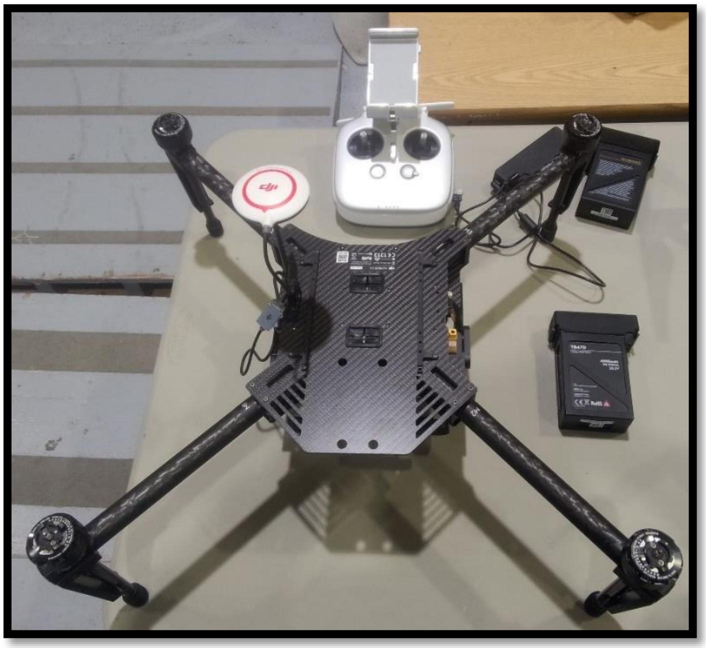


Figure 1.2: A stock DJI Matrice 100 with controller and TB47D Battery

The inspiration for this project was the lab's need to attach a processor and sensors to the Matrices. The mounting system needed to be lightweight to keep the total system weight below the maximum takeoff weight. It also needed to be compact to fit on the top plate of the Matrice. To meet these requirements, a significant amount of time needed to be invested in designing this mounting system. Rather than invest time and resources into creating a mounting system that was only compatible with the Matrice, it was determined that a stand-alone system capable of being attached to a wide variety of robotic vehicles would be more beneficial to the lab in the long-term.

Chapter 2 provides a literature review of sensing and processing systems used for robotic mapping and state estimation. Chapter 3 describes the design development and software integration of the box. Chapter 4 presents the final design and the manufacturing methods used. Chapter 5 discusses the testing methods and results. Chapter 6 concludes the thesis and provides recommendations for future work.

The goal of this project was to create a stand-alone unit capable of providing sensing and processing for robotic vehicles. The primary objective was to create a package that contained all of the components necessary for sensing and processing and that could be attached to UAVs. This included a metal frame to provide structural rigidity and attachment points for the components as well as a plastic shell to protect the components from water and debris. The secondary objective was to perform the necessary software integration and testing necessary to make the box functional. This objective was not as important because other lab members could finish the software integration if it was not completed before the end of the project.

2 Literature and Product Review

Chapter 2 provides a review of literature written by research groups who have used robots and aerial vehicles for mapping and state estimation in unknown environments. The focus is on the hardware setup used to provide the on-board sensing and or processing for the vehicles. Chapter 2 will also present a review of existing products that provide onboard sensing and processing.

2.1 Literature Review

Mobile robot localization and mapping using sonar sensors has been studied for over 25 years [2] and laser range finders for nearly as long [3]. Thus, there are many examples of different sensing and processing setups used for these applications.

In [4], a UAV capable of autonomous mapping and processing using a Samsung Galaxy S2 Android smartphone was created. The advantage of this system is that the processor and camera are combined into one unit, the smartphone, which is more weight and space efficient. The main limitation is the processing power of the smartphone which is capable of mapping a defined environment but not capable of performing simultaneous localization and mapping (SLAM).

The approach in [5] uses an Ardupilot Mega as the autopilot and sensing unit. This autopilot contains a gyrometer, accelerometer, barometer, and global positioning system (GPS). A PandaBoard processing board was connected to the autopilot via a universal serial bus (USB) to transistor-transistor logic (TTL) cable to run the localization and mapping algorithms. The system is capable of tracking moving targets with high precision (97.4% average). The downside of this setup is the use of two processing boards, the Ardupilot Mega and PandaBoard, which increases system weight and size.

Similar systems have been used for mapping and navigation in GPS-denied environments [6]. In [7] a Microsoft Kinect sensor mounted to the base of a quadcopter and tilted slightly downwards was used to provide RGB-D camera data to create a 3D map of the environment. An IMU was also used for state-estimation. A 1.86GHz Core2Duo Processor with 4GB of RAM was used for on-board processing. The project was successful in accurately mapping and path planning in indoor environments and was able to conduct all sensing and computation required for positional control on-board the vehicle.

There has also been research performed using programmable logic devices such as field-programmable gate arrays (FPGA) as the primary processing units for robotic mapping systems [8]. The advantage of FPGAs are that they are more customizable than standard computer processors. For example, in [9] an FPGA was used in a robotic system to control an ultrasonic sensor circuit containing eight sensors as well as a voltage regulator circuit and several stepper motors. However, FPGA setups are more difficult to implement and generally not as size or energy efficient as standard processors.

2.2 Existing Product Review

One commercial product capable of mapping and localization is the Stencil from Real Earth Inc. [10]. The Stencil is a stand-alone system that can be hand-held or mounted to a vehicle. It consists of a square aluminum box which holds an i7 dual-core computer processor and a six degrees of freedom micro-electro-mechanical-system (MEMS) inertial measurement unit (IMU). The exterior of the box has power, USB, and Ethernet connections as well as a mounting plate for a Velodyne VLP-16. The Velodyne has a 360 degree laser range finder and a range of 100 meters to provide measurements of the surrounding area. The stencil weighs 700 grams without the Velodyne attached and 1530 grams with it attached. It is powered externally by a 12-19V DC power supply.

The stencil performs many of the SLAM tasks required for research by the RDML. However, it has several disadvantages compared to a custom designed system. One of these is its selling price of several thousand dollars. Only one Stencil would have been able to be purchased and the lab would function best with several of these systems. Another disadvantage of the Stencil is that is does not have a camera, instead relying on the Velodyne for camera measurements. The Velodyne is large and heavy which makes it not possible for use on UAV systems where size or weight are a concern.

The xOEMcore from Oxford Technical Solutions is another commercial sensing system that can be used in robotic systems [11]. It contains dual GPS units and three IMU sensors as well as an ARM (Advanced RISC Machine) processor. These sensors provide highly accurate localization and navigation data. However, the unit does not include a camera or a light detection and ranging (LIDAR) unit but does have communication ports that allow for the addition of these components. The high price would also be prohibitively expensive for RDML.

3 Design Development

Chapter 3 provides information on why each specific component in the box was selected. It also shows how each component was integrated into the system and how the layout of the components evolved over time through CAD models and prototyping.

3.1 Hardware Selection

The first component selected was the Inertial Measurement Unit (IMU). IMUs have either 6 or 9 degrees of freedom. Degrees of freedom refers to the number of axes which the device measures. A 6 degree of freedom IMU usually has two sensors, a gyroscope and an accelerometer [12]. The accelerometer measures acceleration in the x, y, and z directions. The gyroscope measures the rotation of the IMU in the roll, pith, and yaw axes. A 9 degree of freedom IMU also adds a magnetometer. Magnetometers detect the earth's magnetic field along the x, y and z axes. This is used to measure the IMU's orientation in reference to the earth's magnetic pole.

The main criteria that were used to select an IMU were: price, accuracy, size, weight, and compatibility with a Linux-based processor. IMU's range in price from under $5 to several thousands of dollars. The primary difference between an inexpensive and a higher cost IMU is the accuracy and range of the sensors and the latency of the system. The box needed an accurate IMU but it was also important that the IMU not cost too much so that the whole budget wasn't spent on one component. For this reason the search focused on mid-range IMU's costing in the range of $100-$300.

This led to the selection of the UM7 IMU. The UM7 dimensions are 1.06 x 1.02 x 0.26 inches, and it weighs 3.5 grams [13]. The small size and weight are ideal for the box system. The static pitch/roll accuracy is ±1°, and the static yaw accuracy is ±3°. The accelerometer has a range of ±8 g, and the gyrometer has a range of ±2,000 degrees per second. These specifications show that the UM7 has the accuracy and range needed to provide the box with meaningful acceleration and rotation data.

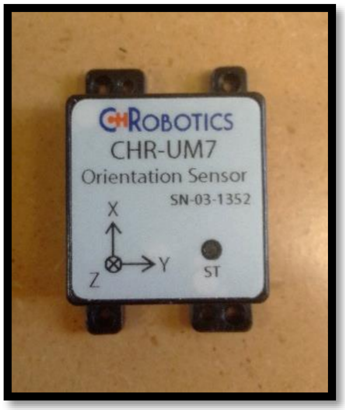


Figure 3.1.1: UM7 IMU

The main requirement for the camera was that it needed to have a global shutter rather than the more common rolling shutter. Digital camera sensors are made up of an array of photo-diodes called pixels which each collect a certain amount of light [14]. In a rolling shutter camera each row of pixels collects light before the next row, meaning that all of the pixels do not collect light at the same time. This causes motion blur if the camera is trying to take a picture of a moving target because the target will be in a different position when each row of pixels triggers. A global shutter camera solves this issue by having each pixel collect light at the same time. For this application the camera will need to be able to accurately take data when the UAV is moving at high speeds so a global shutter is required. It was also decided to use a wide-angle lens to be able to capture more of the surrounding environment.

The basic requirements for the global shutter camera were: price, size, weight, and compatibility with Linux and with a wide-angle lens. To be able to compare cameras that met all of these requirements these factors were also considered: FPS (Frames per Second), number of pixels, and the resolution. To get a baseline for what was necessary in these categories, papers from other research groups who had used cameras for localization and mapping on aerial vehicles were examined. These findings gave a baseline of at least 30 FPS, at least 0.3 million pixels, and a resolution of at least 640 x 480.

This led to the selection of the Chameleon3 1.3 MP Color USB3 Vision camera from Point Grey which costs $295. The Chameleon has a frame rate of 30 FPS and a resolution of 1288 x 964

[15]. It has 1.3 million pixels with a charge coupling device (CCD) sensor and a global shutter. It also has a USB 3.0 port and Point Grey provides free software that is compatible with Linux and Windows. It can be equipped with either a C/CS-mount lens holder or a M12 lens holder giving it a large variety of possible lenses. The main reason for the selection of the Chameleon over other similar cameras is its smaller size. The chameleon has dimensions of 44 x 35 x 19.5 mm, roughly half the size of other comparable cameras.

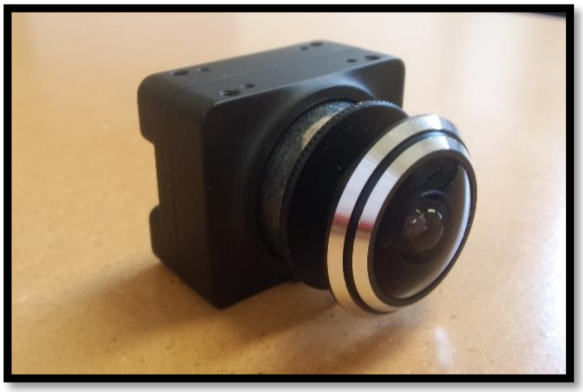


Figure 3.1.2: Chameleon3 camera with fish-eye lens

The next component to be selected was the downward-facing lidar. Lidar works by shining an ultraviolet light at objects. The light hits these objects and reflects back to the lidar sensor [16]. The sensor measures the time it takes light to return to the lidar. The distance of the object can then be determined using the travel time and the speed of light. The main criteria to be considered for a lidar system are price size, weight, maximum range, accuracy, and scan rate. This led to the selection of the Lidar Lite V2, which has a range of 40m, an accuracy of ±0.025m, and a scan rate of 50 Hz [17]. The main reason for selecting the Lidar Lite V2 was the small size and weight. The module is 20 x 48 x 40 mm and weighs 22 grams. It is also very affordable, selling for a price of $115. Other lidar modules can cost several thousand dollars.

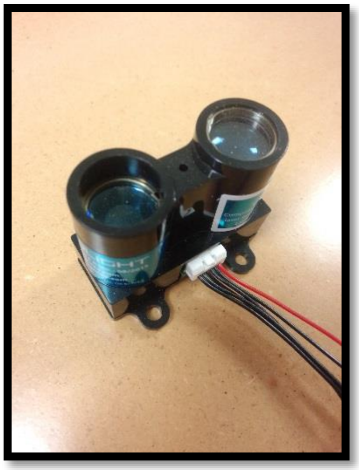


Figure 3.1.3: Lidar Lite V2

The computer processing unit (CPU) was selected primarily based on size, weight, processing power, and energy consumption. Computers in the mini-PC category, designed to be small and light, tend to perform well in these categories [18]. Prior to this project, the lab was already in possession of several Gigabyte Brix mini-PCs. These CPUs were used by the lab because the processors are more powerful than other mini-PCs and the case can be easily removed to decrease weight and size. Several other mini-PCs were researched for use in this project but it was determined that the Gigabyte Brix models were best suited because of their superior processors and removable case. The specific Brix model chosen was the GB-BXi7-4500. It has a 1.8 GHz Intel i7 processor and measures 29.9 x 107.6 x 114.4 mm [19]. It also has 4 USB 3.0 ports and a maximum power consumption of 80 watts.

Figure 3.1.4: Gigabyte Brix GB-BXi7 with case [20]

The Brix requires an input voltage of 12V DC. The box is powered with lithium polymer (LiPo) batteries which come in a variety of voltage ratings. Their voltage also varies somewhat based on the charge level of the battery. To be able to provide the Brix with a steady, 12V DC current a voltage regulator is needed. Voltage regulators take a variable input current and regulate it to a steady output current. The main requirement for selection was the power rating of the voltage regulator, size, and weight. The Brix CPU, which provides power to the other components, has a max power rating of 80W so the voltage regulator needs to be rated at least that high. This criteria led to the selection of the DROK LTC3780 Voltage Regulator which is rated for 100W [21]. It measures 60 x 50 x 20 mm and weighs 21 g. It also has all of the input and output connections grouped together on the same side which makes wire routing easier. The regulator takes DC current between 4-32 V and transforms it into 0.8-32V DC. The output voltage is controlled by turning a small screw clockwise or counter-clockwise.

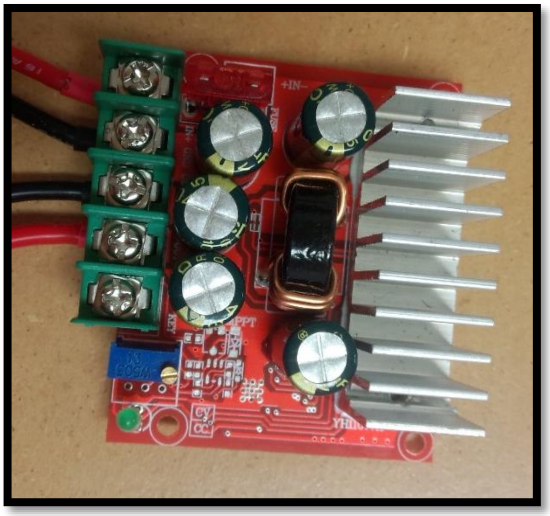


Figure 3.1.5: DROK LTC3780 Voltage Regulator

The last component which needed to be incorporated into the design was the Velodyne Lidar Puck VLP-16. The Velodyne is a 360° Lidar sensor. It weighs 830 g and has a diameter of 103mm and a height of 73mm [22]. The Velodyne was purchased for research by RDML prior to the start of this project. It has a much greater field of view and accuracy than the Lidar-Lite but is also significantly larger and heavier. Because of this, it is important for the Velodyne to be easily removable from the rest of the box so that the box can be used on aircraft with weight or space constraints. It is also important for the Velodyne to be near the center of the vehicle to which it is attached to be able to provide accurate data.

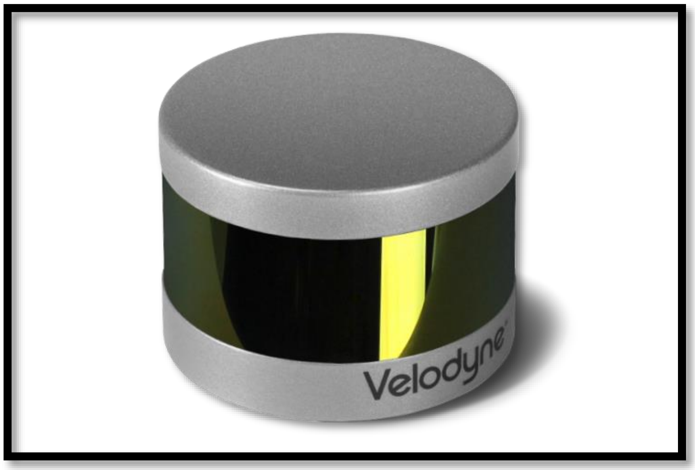


Figure 3.1.6: Velodyne Lidar Puck VLP-16 [22]

3.2 Hardware Integration

Each sensing component in the box needs to be able to connect to the Brix CPU via the USB 3.0 ports. The Chameleon3 camera has a micro B USB 3.0 port and is able to connect directly to the Brix via a micro B to A USB 3.0 cable. The UM7 IMU has a TTL serial communication port. This cannot be directly connect with a USB port. A USB expansion board, sold by CHRobotics, is needed to allow the UM7 to connect to the CPU [23]. The expansion board has a TTL serial communication port for the input from the UM7 and a mini-B USB port for the output to the Brix. It measures 42 x 13 x 6 mm and weighs 7 g.

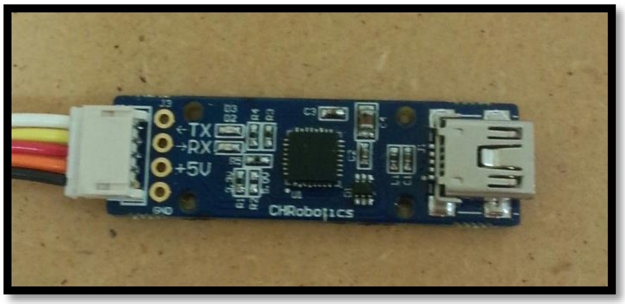


Figure 3.2.1: UM7 USB Expansion Board

The Lidar Lite is also unable to interface directly with the CPU. It has an I^{2}C communications port. An I^{2}C to USB adapter is needed to connect the Lidar to the CPU. The adapter chosen for the box measures 35 x 18 x 12 mm and weighs 4 g [24]. It has an I^{2}C port on one end and a USB Type-B port on the other. A USB Type-B to Type-A cable allows the adapter to connect to the Brix CPU.

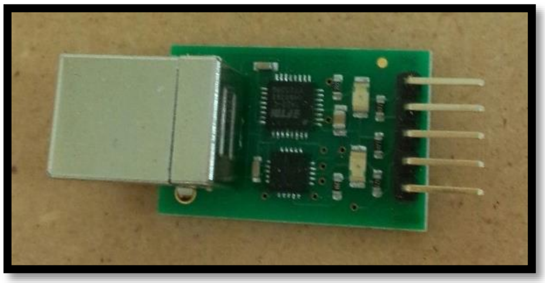


Figure 3.2.2: I^{2}C to USB Adapter

To connect the voltage regulator to the Brix CPU, wires were soldered directly to the Brix printed circuit board (PCB) rather than using the power port on the exterior of the Brix. This allowed the wiring to be contained inside of the box. An XT60 male and female connector were added in the middle of the wire to allow the brix and voltage regulator to be separated without having to unsolder the connections or cut the wires.

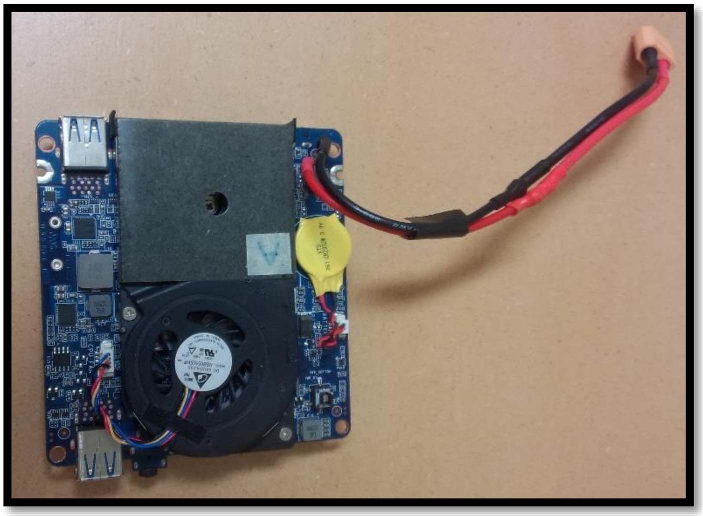


Figure 3.2.3: Gigabyte Brix PCB with power wires

3.3 Design Iterations

The initial prototype created consisted of an aluminum plate mounted directly to the rails on the top of the DJI Matrice quadcopter. The front of the plate hung off the front edge of the DJI to allow the downward facing Lidar to be attached. The rest of the components were mounted to the top of the plate. The Velodyne was mounted onto a separate plate which was then attached to standoffs that attached to the main plate. The Velodyne board was mounted to the underside of the Velodyne plate.

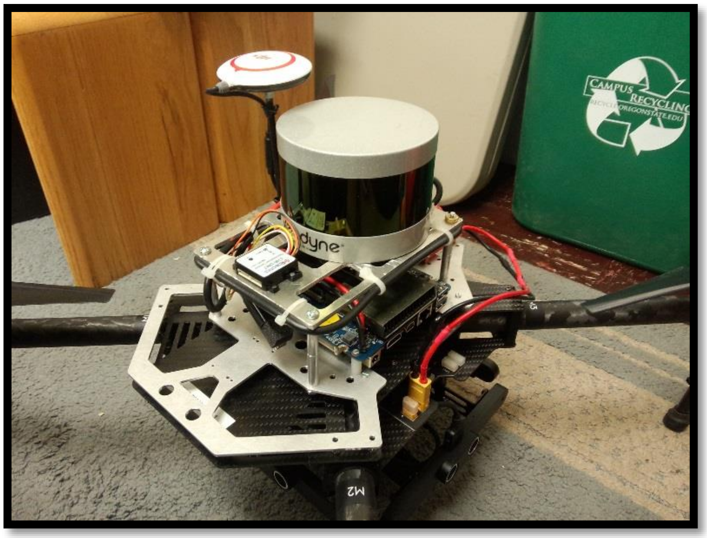


Figure 3.3.1: First prototype design attached to DJI Matrice

This initial prototype worked well for testing. However, it had several disadvantages. It was not very space efficient and took up almost all of the top of the Matrice. It also was not compatible with other UAV systems. It was also not very weight efficient, some portions of the plate were not used and the Velodyne was fairly thick to allow tapped holes for the Velodyne board. This prototype also did not include a shell around the plate so the components were open to the air. For this reason the prototype was not flown very often.

The second iteration of the design was created in Solidworks but not manufactured due to time constraints. This iteration focused mostly on space efficiency. All of the components were fit into a fairly square shaped box. There were two main plates that most of the components were attached to and a side plate that the voltage regulator attached to. This side plate was positioned vertically, at a 90 degree angle to the other two plates. This helped a lot with space efficiency but required several brackets to hold in place. This along with other aspects of the design made it quite heavy which would have decreased the flight time of a UAV while it was attached and possibly made the aircraft unable to take-off. This design was also too tall for the Velodyne to be put on top while attached to the DJI Matrice due intersecting with the GPS mount on the DJI.

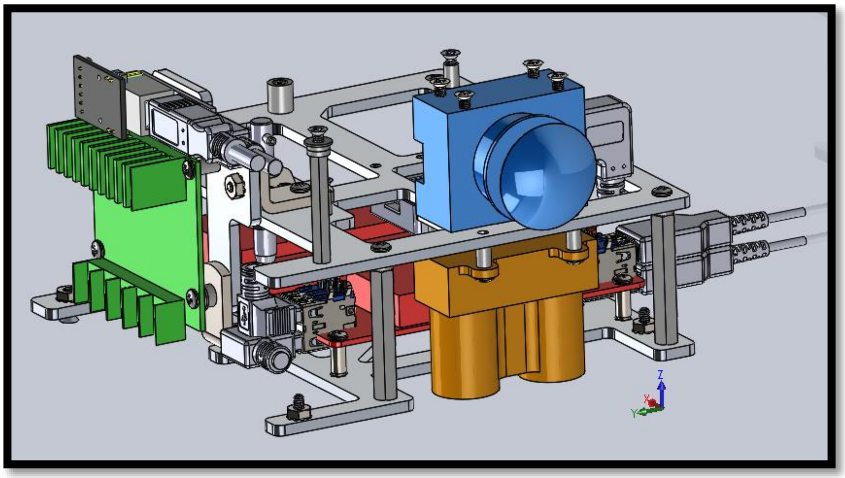


Figure 3.3.2: Second design iteration in Solidworks

3.4 Software Integration

Ubuntu, a Linux-based operating system, was installed on the Gigabyte Brix. Ubuntu was used because it is compatible with the Robot Operating System (ROS). The Indigo version of Robot Operating System (ROS) was also installed. ROS is an open-source framework of tools and libraries that make it easier to create software for robotics [25]. ROS has many packages relating to robotic vision and mapping which made it useful for the software integration of the sensing components of the box.

There are several ROS packages that can be used for SLAM. The LSD SLAM package, which is compatible with the Indigo version of ROS, was chosen. LSD SLAM differs from other SLAM packages because it uses image intensity, rather than keypoints, for tracking and mapping [26]. SLAM packages that use keypoints identify multiple points of interest, or keypoints, in each image taken. The location of these points in different images is compared to determine the location of the image in regard to the other images that were taken. This allows the SLAM software to stitch the images together to form a map of the environment and the location of the camera at different points in time. This process is known as Scale-Invariant Feature Transform (SIFT) [27]. LSD SLAM compares the image intensity differences of full images, not just the difference in specific keypoints. This leads to higher accuracy, especially in environments where few keypoints can be identified. Many indoor environments fit this category due to the uniformity of walls and flooring.

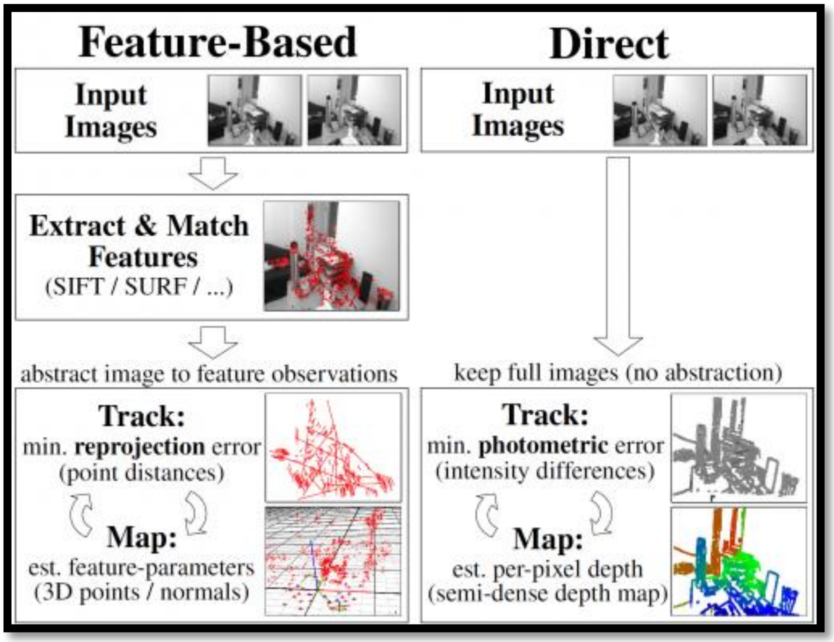

Feature-Based
Direct
Input Images
Input Images
Extract & Match Features (SIFT / SURF / ...)
abstract image to feature observations
keep full images (no abstraction)
Track: min. reprojection error (point distances)
Map: est. feature-parameters (3D points / normals)
Track: min. photometric error (intensity differences)
Map: est. per-pixel depth (semi-dense depth map)

Figure 3.3.1: Diagram explaining the difference between the keypoint tracking method and the image intensity method used by LSD SLAM [26]

Prior to use of the LSD SLAM package, a calibration file for the camera must be created. Two lenses were purchased for the camera, a standard angle lens, the Boowon BW38BLF, and a wide angle lens, the COP-USA L017S. Each of these lenses needed to be calibrated separately due to the radial distortion created by the wide angel lens. The camera parameters needed for LSD SLAM to function properly are the normalized focal length in the x and y directions, normalized camera center in the x and y directions, and normalized distortion factor. The ROS camera calibration package was used to obtain these parameters. The calibration process involves moving a checkboard pattern around the camera. The software identifies the points where the squares of the checkerboard meet and calculates the distance between these points to determine the image distortion [28].

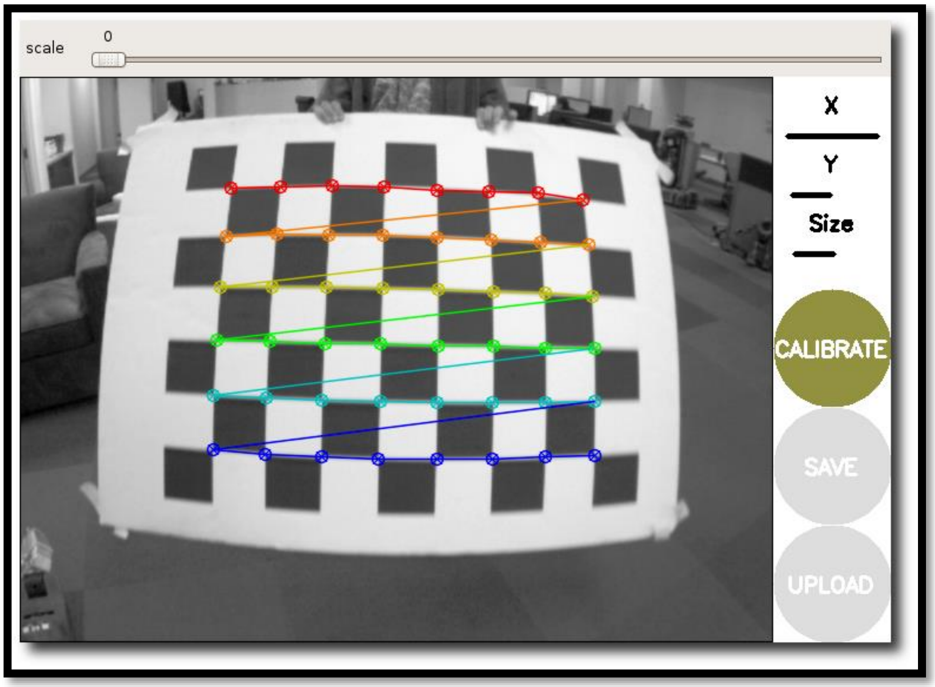


Figure 3.3.2: Example of the ROS camera calibrator process [28]

To make use of the UM7 IMU the ROS um7 package was needed. This package creates a ROS node that allows the orientation messages created by the IMU to be used by other ROS packages, such as LSD SLAM [29]. A similar package, the lidar-lite-ros-scanner-driver, is needed to convert the data produced by the downward-facing Lidar into messages that can be used by ROS [30]. Currently there is no SLAM package that can use data from a camera, IMU and Lidar but it is possible in the future that such a package will be created. In the meantime, the Lidar data could be used to verify the mapping and localization accuracy of LSD SLAM.

4 Final Design

Chapter 4 provides information about the design of the final product and the manufacturing methods used to create the product.

The main goals of the final design were:

1. Minimize space used
2. Minimize total weight
3. Adaptability with other UAV systems
4. All ports easily accessible
5. Easy to manufacture and maintain

The total weight of the box with the Velodyne is 1655 grams. Without the Velodyne it is 825 grams. The dimensions with the Velodyne are 178 x 136 x 127 mm and without the Velodyne 178 x 136 x 55mm. The total cost to build a box including all components except the Velodyne is approximately $1400 as of May 2016.

4.1 Layout

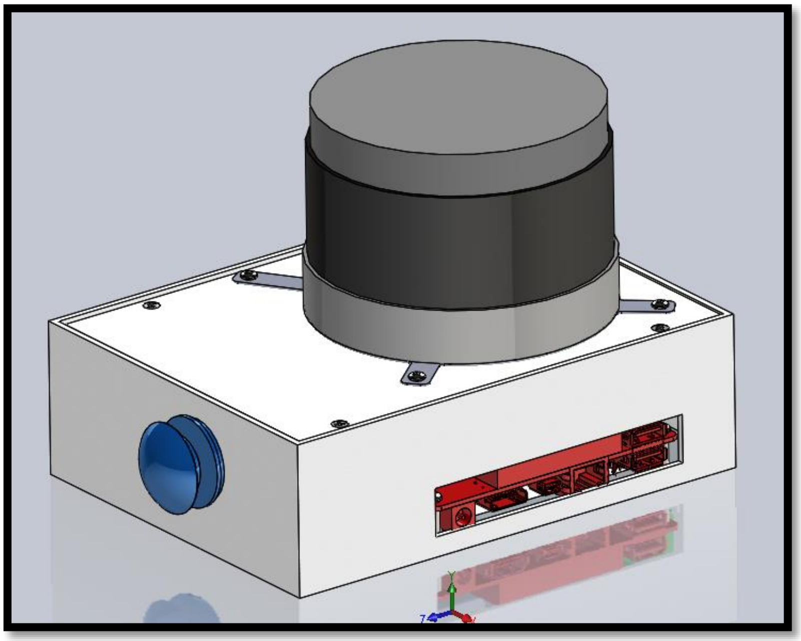


Figure 4.1.1: Box Exterior in Solidworks

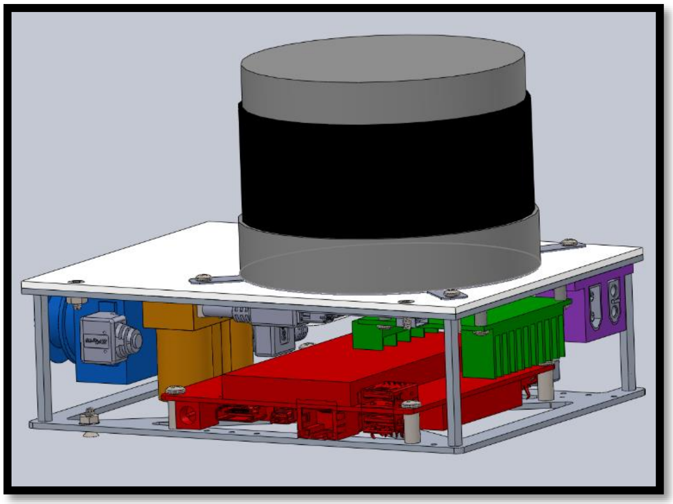


Figure 4.1.2: Box Interior in Solidworks, the components are color-coded: Camera = Blue, Lidar-Lite = orange, Brix CPU = red, voltage regulator – green, XT60 connector holder = purple

The layout of the components is fairly simple allowing for only three aluminum frame pieces (two without the Velodyne) which allowed the weight to be kept to a minimum. There is a bottom frame plate, a top frame plate, and a Velodyne frame plate. There are two 3D printed shell pieces which enclose the frame and keep the components safe from outside elements. The bottom shell piece attaches to the underside of the bottom frame and also includes vertical walls to surround the box. The top shell attaches to the top of the top frame plate to cap the box. Having two shell pieces is much easier for manufacturing and also allows the box to be partially disassembled more easily to help with maintenance. The bottom frame plate has two main sets of standoffs. The first set is for the gigabyte brix, allowing it to be mounted to the frame while still having enough clearance for the wiring on the bottom of the board. The second set are much longer and attach to the top frame plate, allowing the boards to be connected and providing a strong, rigid frame around the sensitive components inside the box.

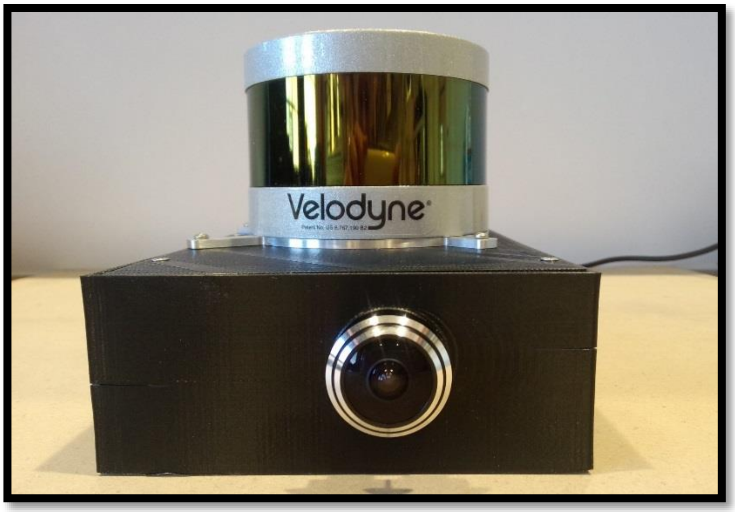


Figure 4.1.3: Front View of Box Exterior

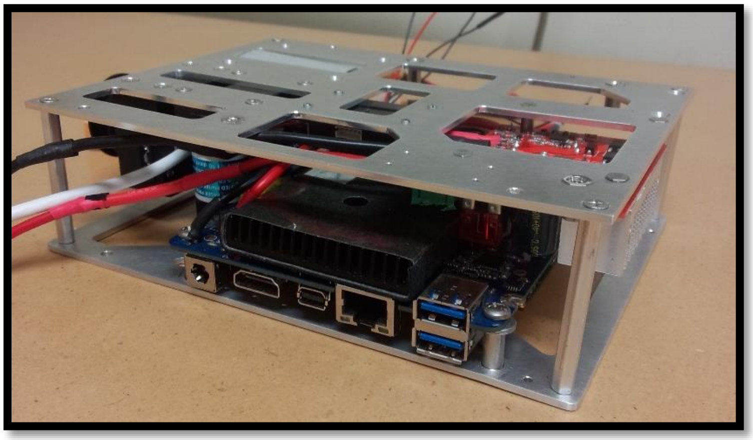


Figure 4.1.4: Side View of Box Interior

The top plate has mounting holes for the rest of the components inside of the box. One of these is the camera, which attaches to the underside of the top frame via fasteners which can be screwed directly into tapped holes in the camera case. Another one of these components is the downward facing lidar, which also attaches to the underside of the top frame. The Lidar was positioned so that the bottom frame plate would not be in the way of its sensors and circular cutouts were added to the bottom shell so as to not block the lidar. The IMU attaches to the center of the underside of the top frame plate. It is attached upside down which reverses the acceleration data it produces. This can be corrected for in the software.

Also attached to the underside of the top plate is the voltage regulator. Standoffs were used so that the bottom of the regulator would not be touching the top frame plate, which could potentially cause an electrical short. The original connectors were removed from the voltage regulator allowing wires to be directly soldered to the terminals. This saved space and gave a more secure power connection.

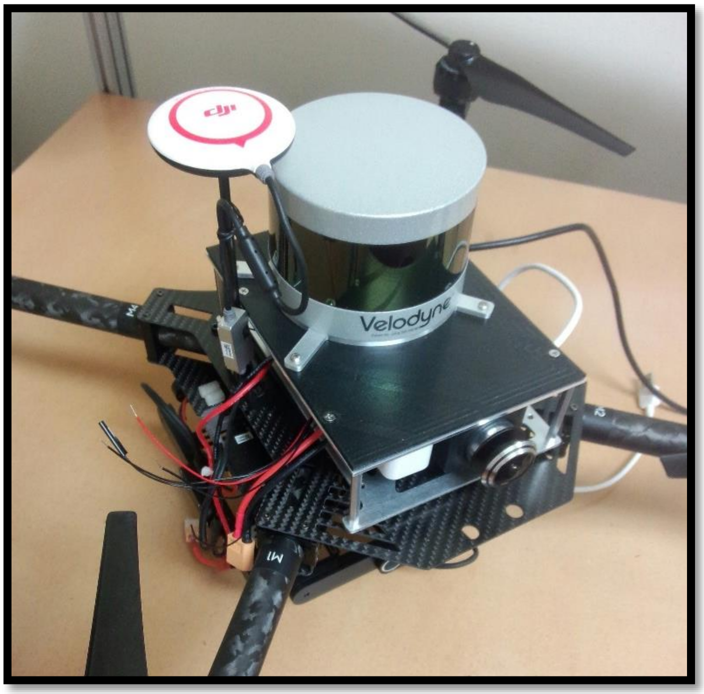


Figure 4.1.5: Box attached to DJI Matrice UAV

The voltage regulator receives power via an XT60 connector mounted on the back side wall of the box. The output of the voltage regulator is split into two paths. One path provides power to the Gigabyte Brix, which then provides power to the rest of the components inside the box via the USB ports. The other path provides power to the Velodyne circuit board.

The Velodyne frame plate is the smallest of all three frame pieces. It is a circular-shaped plate that has mounting holes for the Velodyne puck and also has tabs that come off the circle to allow the plate to be mounted to the top of the box. On top of the Velodyne there is a 3D printed cap which houses the Velodyne board. This allows the board to be completely out of the line of sight of the Velodyne to allow the Velodyne to take more accurate data.

4.2 Manufacturing Method

The box was modeled and designed using Solidworks CAD software. This allowed for the creation of accurate drawings and part files to be sent to a local machine shop in order to create the 3 aluminum frame pieces. The frame pieces we're manufactured using 6061-T6 Aluminum Alloy Sheet. 6061-T6 is a lightweight, high strength alloy that is commonly used in aerospace applications. The plates were cut out of the sheet metal on a CNC mill.

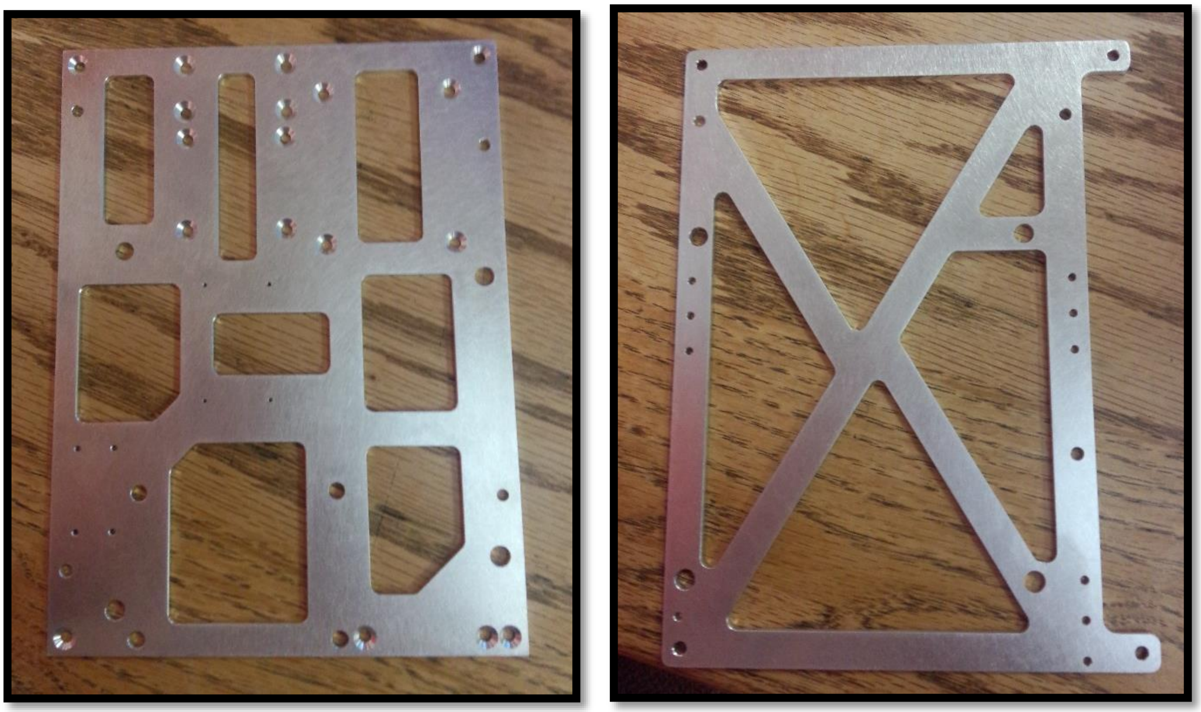


Figure 4.2.1: Top Frame Piece (left) and Bottom Frame Piece (right)

The 3D printed shell pieces were created using the Oregon state University Library 3D printer. The parts were modeled in Solidworks and then the STL files (Stereo Lithography) were sent to the printer by using the online submission form. The fasteners were ordered from McMaster Carr's online catalog. Many had to be purchased in packs of 25, 50, or 100. Initially this is more expensive but it will allow for easy replacement of any missing or damaged parts and will make it easier to build a second box. The USB hub and connectors were purchased on Amazon and the USB wire was purchased from DigiKey.

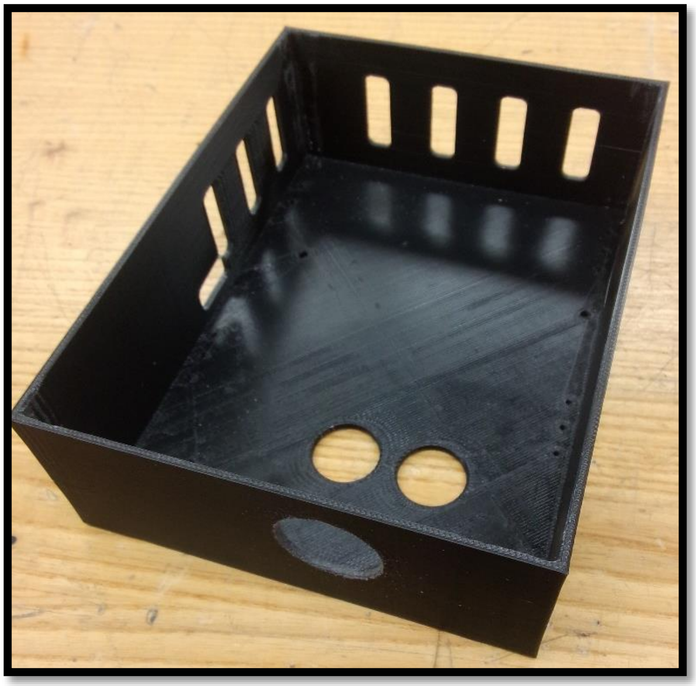


Figure 4.2.2: 3D Printed Bottom Shell Piece

After all of the components had arrived, the first step in the manufacturing process was to attach the standoffs to both the bottom and top frame plates. An arbor press was used to press-fit the standoffs into the plate. This makes the standoffs very secure but also makes it impossible to remove them in the future. The next step was to attach the components to the bottom and top frame plates. This was done using the fasteners purchased from McMaster. After this step some initial testing was done to determine that all of the components were functioning correctly.

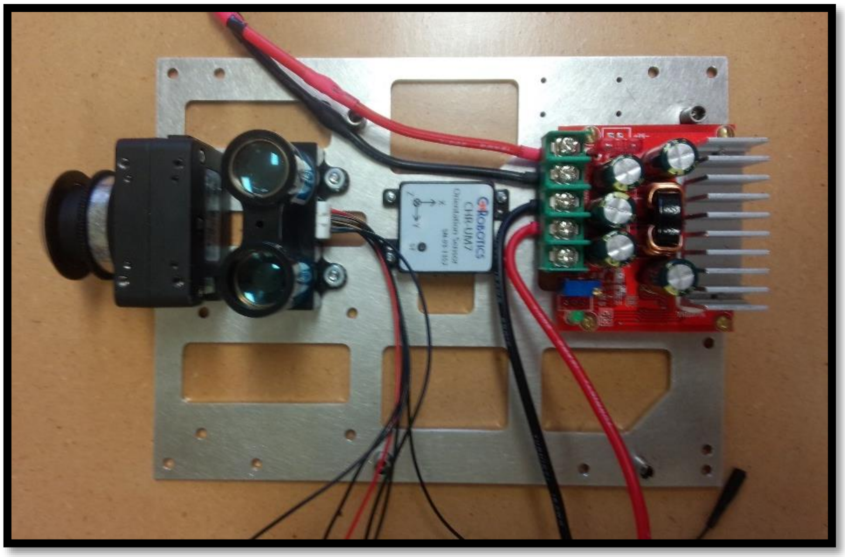


Figure 4.2.3: Top Plate with Components Attached

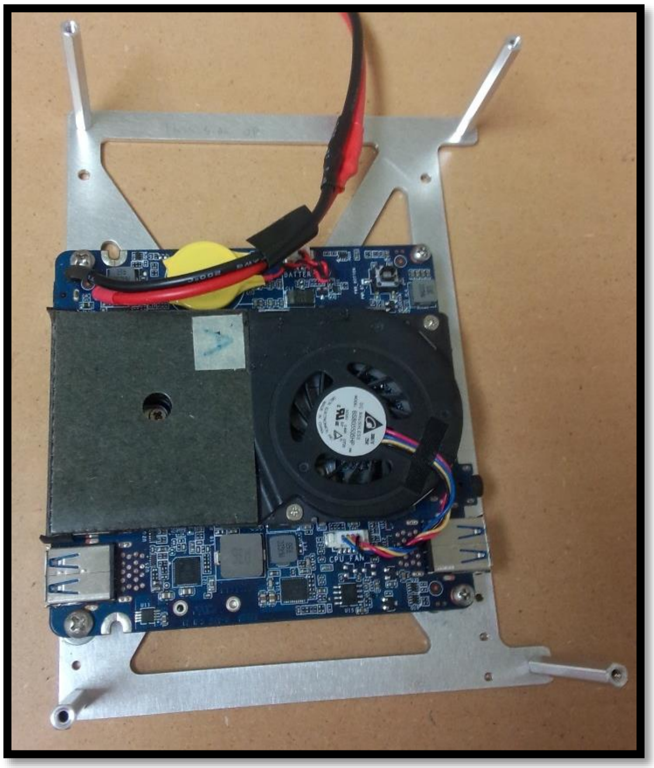
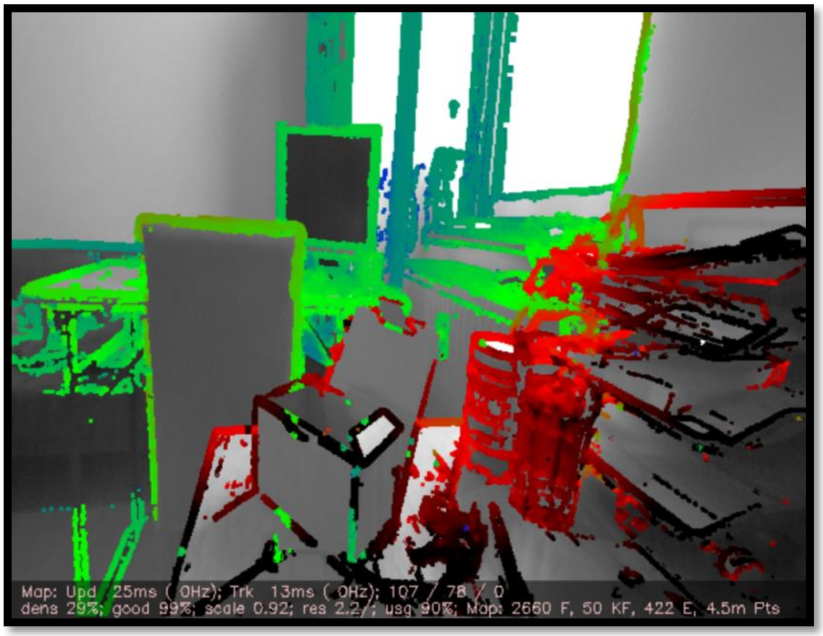


Figure 4.2.4: Bottom Plate with Brix CPU and Standoffs attached

Once it was determined that everything was functioning properly, the bottom and top shell pieces were attached to the plates with flat head screws. The Velodyne was then attached to the Velodyne plate which was then attached to the top of the box. The total assembly time was approximately three hours. This could be decreased in the future with better component tolerancing and manufacturing instructions.

5 Testing and Results

The first test performed was to run an example dataset on the Brix to test the LSD SLAM functionality. An example dataset of a camera moving around a small room is provided in the LSD SLAM tutorial. This dataset is a video of a camera being moved around a small room. It is known to work well with LSD SLAM so it can be used to test SLAM functionality on different processors. LSD SLAM was launched and the dataset was played to generate a point cloud map of the room and localization data for the camera. Figures 5.1.1 and 5.1.2 show the results.



```
Map: Upd  25ms ( 0Hz); Trk  13ms ( 0Hz); 107 / 78 / 0
dens 29%; good 99%; scale 0.92; res 2.2/; usg 90%; Map: 2660 F, 50 KF, 422 E, 4.5m Pts
```

Figure 5.1.1: LSD SLAM viewer while dataset test was in progress. The points show the outline of objects and their distance to the camera.

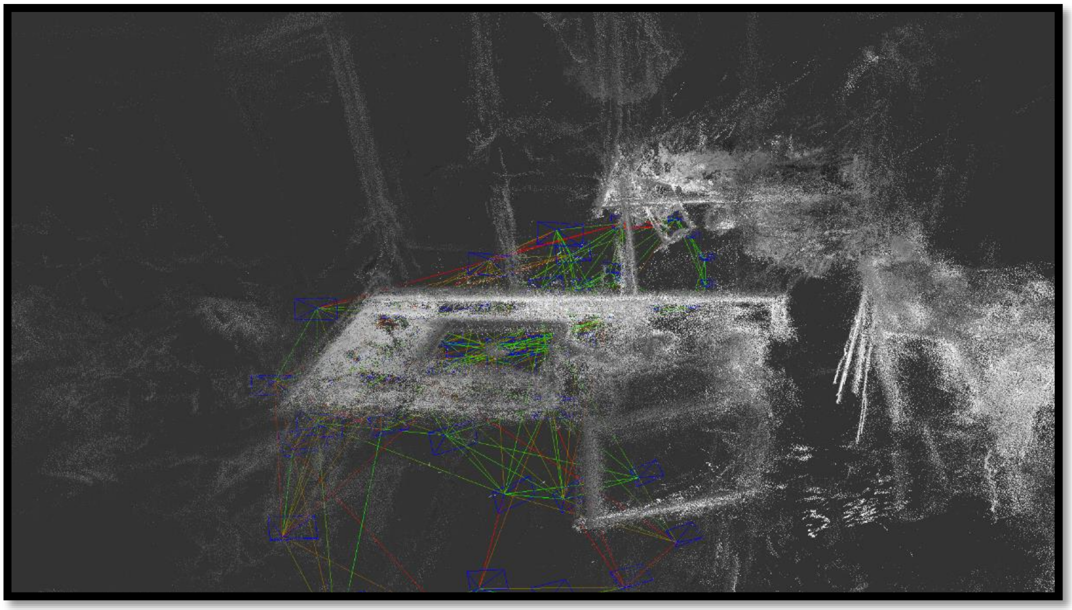


Figure 5.1.2: Point cloud map created of the dataset. The white points show objects in the room such as a desk and shelves. The blue boxes show the location of the camera at different points in time.

The point cloud created by the dataset is fairly clear and the rough outline of most of the objects present can be seen. It is also possible to tell the approximate path taken by the box around the room. Testing of the live SLAM feature of LSD SLAM was not as successful. When the camera was moved around while LSD SLAM was running tracking was frequently lost resulting in incomplete and inaccurate maps. This problem was lessened by moving the camera very slowly but the maps created were still very poor quality.

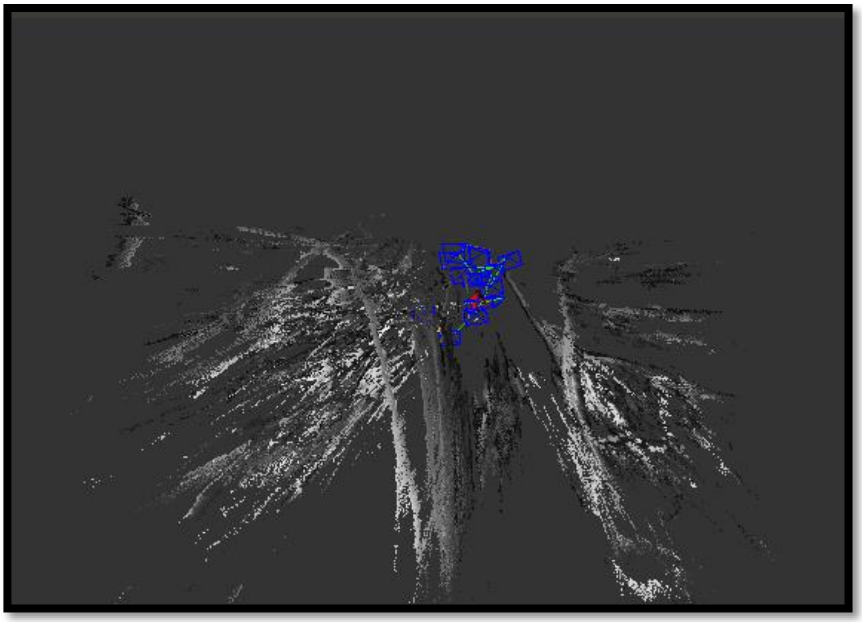


Figure 5.1.3: Point cloud map created using the live SLAM feature of LSD SLAM and moving the camera around the RDML. It is very hard to identify objects in the room and the camera localization is very inaccurate.

Currently, the cause of the poor quality seen in the live SLAM point cloud map is unknown. It is suspected that poor camera calibration could be playing a role. Poor camera calibration could also be causing the issues with the lost tracking while LSD SLAM is running. Possible solutions include rerunning the ROS camera calibration and seeing if the camera parameters produced are significantly different than the original parameters. Another possible solution would be using a different camera calibration package in ROS which could yield more accurate parameters. If neither of these solutions are successful, testing with other ROS SLAM packages could be performed.

The UM7 IMU driver was successfully installed and tested. Orientation messages were published to the correct ROS node. However, integration of these messages with LSD SLAM has not yet been completed. Once this integration is completed, testing should be performed to compare the SLAM accuracy of the system with only the camera compared to the accuracy of the system using both the camera and IMU. The downward-facing Lidar has not been successful connected to the CPU using the I^2C to USB Adapter. Troubleshooting will need to be performed to determine the cause of this. Data has been successfully received using an Arduino microcontroller instead of the CPU so the Lidar does appear to be functional.

6 Conclusion and Recommendations

This project was successful in creating a compact and lightweight box to hold all of the components necessary for on-board processing and sensing. An aluminum frame provides structural strength for the box and a plastic shell protects the components from debris. The box is capable of being attached to the DJI Matrice UAV and could be adapted for other robotic systems. The biggest portion of work that needs to be completed to make the box functional is the software integration. Currently, the box is capable of performing SLAM and creating point cloud maps of the surrounding environment. However, these maps are inaccurate and not useful for research purposes. These software issues will need to be resolved before the box can be of use to the RDML.

There are also many mechanical improvements that could be made to increase the functionality of the box. The first of these is to increase the accessibility of components inside of the box. With the current design, there is little to no accessibility to allow for the repair or replacement of components. Significant disassembly of the system is needed to reach some of the components. Further design iterations could be performed to increase this accessibility. However, this would most likely come at the expense of space efficiency so there would be a limit to how accessible the box could be made.

Another idea that should be considered for future design iterations of the box is an alternate material for the outer shell. The 3D-printed ABS plastic currently used is easy to manufacture but it is also weak and doesn’t have highly accurate resolution for the holes and other features in the shell. One alternative method that could be considered is injection molding. Injection molding is much more precise than 3D printing. An alternative material that could be considered is carbon fiber. Carbon fiber is much stronger than ABS and not much heavier.

Strength and durability testing should be performed on the current or future box designs. This would help identify weak points in the frame that could potentially be damaged if the box was dropped or hit. These points could potentially be reinforced while material could be removed from stronger parts of the frame to compensate.

Another design consideration is the cooling inside of the box. Currently the only feature to allow air to flow through the box is several cutouts in the outer shell. Other cooling methods such as fans and heatsinks should be considered. Thermal analysis of the system could be performed to determine where these active cooling methods should be placed in the box to be the most effective.

Bibliography

[1] "Matrice 100: The quadcopter for developers | DJI", *Dji.com*, 2016. [Online]. Available: http://www.dji.com/product/matrice100. [Accessed: 17- May- 2016].

[2] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons", *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 376-382, 1991.

[3] D. Fox, W. Burgard, H. Kruppa and S. Thrun, "A Probabilistic Approach to Collaborative Multi-Robot Localization", *Autonomous Robot*, no. 8, pp. 325-344, 2000.

[4] M. Leichtfried, C. Kaltenriner, A. Mossel and H. Kaufmann, "Autonomous Flight using a Smartphone as On-Board Processing Unit in GPS-Denied Environments", *Proceedings of International Conference on Advances in Mobile Computing & Multimedia - MoMM '13*, 2013.

[5] M. Siam, R. ElSayed and M. ElHewl, "On-board Multiple Target Detection and Tracking on CameraEquipped Aerial Vehicles", in *International Conference on Robotics and Biomimetics*, Guangzhou, China, 2012.

[6] J. Rogers, J. Fink and E. Stump, "Mapping with a ground robot in GPS denied and degraded environments", in *American Control Conference*, Portland, OR, 2014.

[7] A. Bachrach, S. Prentice, R. He, P. Henry, A. Huang, M. Krainin, D. Maturana, D. Fox and N. Roy, "Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments", *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1320-1343, 2012.

[8] L. Vachhani and K. Sridharan, "Robotic Mapping with Simple Sensing and Processing Hardware - Algorithm and Architecture", in *Control, Automation, Robotics and Vision Conference*, Singapore, 2010.

[9] P. Kumar and K. Sridharan, "VLSI-Efficient Scheme and FPGA Realization for Robotic Mapping in a Dynamic Environment", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 1, pp. 118-123, 2007.

[10] "Stencil", *Real Earth - Fast, Accurate, & Affordable 3D Scene Reconstruction*, 2016. [Online]. Available: http://www.realearth.us/stencil.html. [Accessed: 17- May- 2016].

[11] "xOEMcore", *OxTS*, 2015. [Online]. Available: http://www.oxts.com/products/xoemcore/. [Accessed: 17- May- 2016].

[12] "A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications. « Starlino Electronics", *Starlino.com*, 2016. [Online]. Available: http://www.starlino.com/imu_guide.html. [Accessed: 17- May- 2016].

[13] "Pololu - UM7 Orientation Sensor", *Pololu.com*, 2016. [Online]. Available: https://www.pololu.com/product/2741. [Accessed: 17- May- 2016].

[14] "Point Grey Knowledge Base", *Ptgrey.com*, 2016. [Online]. Available: https://www.ptgrey.com/KB/10028. [Accessed: 17- May- 2016].

[15] "Chameleon3 board level USB3 Vision cameras for industrial, life science, traffic, and security applications.", *Ptgrey.com*, 2016. [Online]. Available: https://www.ptgrey.com/chameleon3-usb3-vision-cameras. [Accessed: 17- May- 2016].

[16] *Lidar-uk.com*, 2016. [Online]. Available: http://www.lidar-uk.com/how-lidar-works/. [Accessed: 17- May- 2016].

[17] "LIDAR-Lite 2 Laser Rangefinder (PulsedLight)", *Robotshop.com*, 2016. [Online]. Available: http://www.robotshop.com/en/lidar-lite-2-laser-rangefinder-pulsedlight.html. [Accessed: 17- May- 2016].

[18] "Mini PCs with Intel Inside®", *Intel*, 2016. [Online]. Available: http://www.intel.com/content/www/us/en/mini-pc/mini-pc-overview.html. [Accessed: 17- May- 2016].

[19] "GIGABYTE - Desktop PC - Mini-PC Barebone - GB-BXi7-4500 (rev. 1.0)", *Gigabyte.com*, 2016. [Online]. Available: http://www.gigabyte.com/products/product-page.aspx?pid=4736#ov. [Accessed: 17- May- 2016].

[20] "Amazon.com: Gigabyte Brix Ultra Compact PC Intel i7-4500U 3.0/1.8 GHz Wi-Fi/BT4.0 Processor (GB-BXi7-4500): Computers & Accessories", *Amazon.com*, 2016. [Online]. Available: http://www.amazon.com/Gigabyte-Compact-i7-4500U-Processor-GB-BXi7-4500/dp/B00FNPCKUU/ref=sr_1_8?s=pc&ie=UTF8&qid=1463433103&sr=1-8&keywords=gigabyte+brix. [Accessed: 17- May- 2016].

[21] "DROK LTC3780", *Amazon.com*, 2016. [Online]. Available: http://www.amazon.com/DROK-Synchronous-Converter-Transformer-Efficiency/dp/B00JJHPF0A. [Accessed: 17- May- 2016].

[22] "VLP-16", *Velodynelidar.com*, 2016. [Online]. Available: http://velodynelidar.com/vlp-16.html. [Accessed: 17- May- 2016].

[23] U. Sensor, "USB Expansion Board | CH Robotics", *Chrobotics.com*, 2016. [Online]. Available: https://www.chrobotics.com/shop/usb-expansion-board. [Accessed: 17- May- 2016].

[24] "Devantech USB to I2C Adapter | Acroname", *Acroname.com*, 2016. [Online]. Available: https://acroname.com/products/DEVANTECH-USB-I2C-INTERFACE-ADAPTER?sku=R286-USB-I2C. [Accessed: 17- May- 2016].

[25] "ROS.org | About ROS", *Ros.org*, 2016. [Online]. Available: http://www.ros.org/about-ros/. [Accessed: 10- May- 2016].

[26] "Computer Vision Group - Visual SLAM - LSD-SLAM: Large-Scale Direct Monocular SLAM", *Vision.in.tum.de*, 2016. [Online]. Available: http://vision.in.tum.de/research/vslam/lsdslam?redirect=1. [Accessed: 13- May- 2016].

[27] "OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform)", *Docs.opencv.org*, 2016. [Online]. Available: http://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html#gsc.tab=0. [Accessed: 08- May- 2016].

[28] "camera_calibration/Tutorials/MonocularCalibration - ROS Wiki", *Wiki.ros.org*, 2016. [Online]. Available: http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration. [Accessed: 15- May- 2016].

[29] "um7 - ROS Wiki", *Wiki.ros.org*, 2016. [Online]. Available: http://wiki.ros.org/um7. [Accessed: 12- May- 2016].

[30] "r3n33/lidar-lite-ros-scanner-driver", *GitHub*, 2016. [Online]. Available: https://github.com/r3n33/lidar-lite-ros-scanner-driver. [Accessed: 12- May- 2016].

[31] "tum-vision/lsd_slam", *GitHub*, 2016. [Online]. Available: https://github.com/tum-vision/lsd_slam. [Accessed: 11- May- 2016].