**GPU Enhanced LIDAR for Real Time 3D Sensing**

Project Workbook

By

Amit Vasant Pachore

Anush Shankar

Bhupendra Naphade

Urvashi Agrawal

11/12/2016

**Advisor**: Dr. Harry Li

Contents

[List of Tables 3](#_Toc466577062)

[List of Figures 4](#_Toc466577063)

[Chapter 1. Literature Search, State of the Art 5](#_Toc466577064)

[1.1. Literature Search 5](#_Toc466577065)

[1.2. State-of-the-Art Summary 6](#_Toc466577066)

[1.3. References 7](#_Toc466577067)

[Chapter 2. Project Justification 8](#_Toc466577068)

[Chapter 3. Project Requirements 9](#_Toc466577069)

[3.1. Essential features 9](#_Toc466577070)

[Mechanical Housing 9](#_Toc466577071)

[Motor Drive 9](#_Toc466577072)

[Laser sensing and Data Processing 9](#_Toc466577073)

[3.2. Optional Features 10](#_Toc466577074)

[High quality user interface 10](#_Toc466577075)

[Object Detection and Marking 10](#_Toc466577076)

[3.3. Non-functional features 10](#_Toc466577077)

[Augmented Reality 10](#_Toc466577078)

[Chapter 4. Dependencies and Deliverables 11](#_Toc466577079)

[4.1. Dependencies 11](#_Toc466577080)

[4.2. Deliverables 11](#_Toc466577081)

[Chapter 5. Project Architecture 12](#_Toc466577082)

[5.1 Jetson TK1 Hardware Architecture 13](#_Toc466577083)

[5.2 Jetson TK1 Software Architecture 13](#_Toc466577084)

[5.3 Servo Motor 14](#_Toc466577085)

[5.4 Project Software Architecture 14](#_Toc466577086)

[Chapter 6. Project Design 15](#_Toc466577087)

[Chapter 7. QA, Performance, Deployment Plan 16](#_Toc466577088)

[7.1 Testing 16](#_Toc466577089)

[Unit Testing 16](#_Toc466577090)

[Functional Testing 17](#_Toc466577091)

[Integration Testing 17](#_Toc466577092)

[Performance and Stress Testing 17](#_Toc466577093)

[Acceptance Testing 17](#_Toc466577094)

[Issue Tracking 18](#_Toc466577095)

[Testing Schedule 18](#_Toc466577096)

[7.2 Deployment Plan 19](#_Toc466577097)

[Chapter 8. Implementation Plan and Progress 20](#_Toc466577098)

[Chapter 9. Project Schedule 23](#_Toc466577099)

[Chapter 10. 24](#_Toc466577100)

[Chapter 11. 25](#_Toc466577101)

# List of Tables

[Table 1. Project Implementation 22](#_Toc466120020)

[Table 2. Project Task Assignment and Tracking 23](#_Toc466120021)

# List of Figures

[Figure 1. System Block Diagram 12](#_Toc466575665)

[Figure 2. TK1 Embedded Board Top View 13](#_Toc466575666)

[Figure 3. TK1 Architecture and USB Camera 13](#_Toc466575667)

# Literature Search, State of the Art

## 1.1. Literature Search

Three-Dimensional imaging or sensing provides all necessary information of the scanned environment. It has been an integral part of autonomous vehicle, computer vision and machine learning technologies. There has been a lot of research and algorithm development to improve imaging and range detection. Three-dimensional imaging and range detection involve usage of a light emitter and detector as an integral part of the system. There are multiple ways of emission and detection of energy using flash lamp, laser, microwave emitting antenna, ultrasonic source, pulse modulator, sun light, different camera modules and shutter systems. Buffers are used to store the sum and differences extracted from the received beam derive the distance of the object from the camera and the beam emitting system. (Medina, 1992)

**3-D Imaging Techniques**

The 3-D sensing or imaging involves emission of energy to an object, extracting information from received energy and generating 3-D images. Non-optical and optical sensing are the two type methods for sensing. Non-optical sensing includes acoustic sensors, electromagnetic and others. Range is detected by measuring the time taken by the emitted energy to reflect back to the sensing system. Optical sensing systems light is used as a source for data extraction of the surrounding. There are multiple 3-D imaging techniques using optical sensors like laser triangulators, structured light, stereo vision, photogrammetry, Time of Flight, Interferometry and others (Sansoni, Trebeschi, & Docchio, 2009). In this section we review some techniques involves in 3-D sensing and mapping.

1. **Laser Triangulator**

Triangulation is the most widely used technique for sensing in short range up to 10 meters. Most of them support 0.5 to 2m of range detection. The distance of the object from the light source can be determined by analyzing the triangular geometry between the source, object and the detector. This method is accurate, relative insensitive to illuminating conditions and has structure texture effects.

1. **Structured Light**

This technique is similar to the one mentioned above. Structured light sensors simultaneously project bi-dimensional patterns of non-coherent light and analyzing them to obtain the range information for individual point observed

1. **Stereo Vision**

This method is passive approach in which two cameras simultaneously capture the same scene. It follows by camera modelling, feature extraction, corresponding analysis and triangulation. The challenge in this technique is to identify common point between the images captured by the two cameras. This is also done using a single camera capturing images from difference angles.

1. **Photogrammetry**

This technique is used to generate 3-D models with multiple photographs. The steps involved are camera calibration and orientation, image point measurements, 3D point cloud generation, surface generation and texture mapping. This technique can be used to extract information from images of moving objects.

1. **Time of Flight**

This method measures the time required for a light wave to travel from source on to a distant object and reflect back to the detection system. The intensity of the reflected signal along with the time of flight are used to generate 3-D data points. This technique is less accurate when analyzing reflected surfaces as the light is scattered.

1. **Interferometry**

In this technique uses the phenomenon of interference of light. The light is split using a beam-splitter and then recombined. The spatial shape of the resultant beam is used to measure the distance of the object.

## 1.2. State-of-the-Art Summary

The focus of our project is on 3-D imaging using involving LIDAR. The benefit of using LIDAR is its bright, directional and more coherent light source compared with other sources of light. There are a few 3D sensing techniques which include scanning imaging LIDAR and pulsed floodlight- illumination imaging LIDAR. Detection range is the limitation of these techniques. LIDAR is used to emit laser on distant object and measure the distance between LiDAR and the object. 3D images are generated on basis of the data collected by the LiDAR. Another aspect of 3-D sensing is the area measurement. It involves scanning of the area for data acquisition. It can be done with multiple techniques involving mechanical displacement, Multi-point and line projection, Fringe and coded pattern projection and Moiré effect. Multiple 3-D imaging techniques have been reviewed as a part of literature survey. Different sensing techniques are used for different application areas.

The prominent area of research in 3-D sensing involves optimal extraction and analysis of data to generate 3-D maps. The data retrieved from the light or imaging systems needs to be processed with high accuracy and real time. GPU is the new cutting edge technology to handle this data to generate highly accurate results. We are using Jetson TK1 GPU module along with business grade camera for 3-D modelling which would involve camera calibration, Depth-map Retrieval, Multi-view Registration, Multi-view Integration and Texture Mapping.

## 1.3. References

1. Medina, A. (1992). *U.S. Patent No. 5,081,530*. Washington, DC: U.S. Patent and Trademark Office.

This patent claims that a three-dimensional camera system comprises of beam emitter, detector for data extraction and depth measurement. It lists different types of emitter source that can be used for created stereoscopic image

1. Sansoni, G., Trebeschi, M., & Docchio, F. (2009). State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation. *Sensors (Basel, Switzerland)*, *9*(1), 568–601. <http://doi.org/10.3390/s90100568>

This articles briefly describes different types of sensing techniques involved in 3-D sensing. It also provides information of application areas and current available sensing devices.

1. Beraldin, J. A., Blais, F., Cournoyer, L., Godin, G., Rioux, M., & Taylor, J. (2003). Active 3D sensing.

This papers tells about 3-D sensing using active range cameras. It briefs about basic principles involved in majority of 3-D sensing and modelling systems. It also provides with multiple applications illustrating the technology described in the paper.

1. Tu, X. (2009). *Image based 3D Sensing and Modeling Technology for High-end Digital Cameras* (Doctoral dissertation, Stony Brook University).

This paper briefly reviews some existing technologies involved in 3-D sensing and modelling. It tells about the basic steps involved in the process and explains in detail procedure for 3-D modelling using high end digital cameras.

# Project Justification

In this project, we aim to develop a Light Detection and Ranging sensor(LIDAR) integrated with Augmented reality. The sensor uses laser source and detector unit to determine the distance of an object from it, and augments the sensed data to the camera feed to enhance the User Interface. Existing scan based LIDARs achieve up to 6 meters sensing range and lack a well-defined user Interface. The aim of this project is to provide an informative and graphic user interface with improved LIDAR sensing range of up to 20 meters.

The LIDAR will be developed using NVIDIA Jetson TK1 platform with Linux operating system. Linux operating system is an open source and is used for Embedded and other high end applications. User Interface design for the project will use Open source computer vision (OpenCV) library for computer vision and graphics, thereby enhancing the quality of algorithms developed. Among many other components used for the project, camera for augmented reality and laser source will play an important role towards the success of the project. The addition of OpenCV based user interface with augmented reality would help improve user experience and would provide an independently deployable system for distance sensing applications.

The project aims at using state of the art algorithms, powerful processing platform for precision sensing and high quality hardware to ensure that feature enhancements provide desired result. These factors are integral in overcoming existing flaws and providing a good quality product able to serve application needs. This project will provide a deep understanding of computer vision and graphics algorithms, motor drivers and CUDA parallel programming for faster processing.

Use of open source computer vision and graphics libraries would enable easy access to the implementations and help algorithm development in the future. OpenCV based implementations allows independence in design and reusability of algorithms for further improvements in design. Overall the project aims at eliminating current LIDAR drawbacks and add better features to the system, thereby improving its real-world deployment scenarios.

# Project Requirements

A LIDAR is used as a distance sensor providing high precision and 360 degrees of coverage. The data obtained from the sensor can be used in a wide variety of applications such as:

* Self-driving cars.
* Computer vision applications.
* Robotics.

The requirements for a LIDAR can be categorized into essential, optional and non-functional requirements. While essential requirements are necessary for the general product, optional requirements help design the product for specific applications. The following are some of the major requirements of the project:

## 3.1. Essential features

### Mechanical Housing

One of the major requirement for the project is the mechanical housing for the LIDAR. The housing should be able to accommodate the laser source, servo motor driver, camera module, laser detector unit and user interface.

### Motor Drive

The motor helps rotate the LIDAR to achieve 360-degree circular coverage. The motor needs to be mounted in the appropriate position to rotate the laser source, detector unit and camera at a constant speed. PWM drivers need to be used to ensure granular control over speed of rotation, while direction of rotation remains unchanged for the LIDAR application.

### Laser sensing and Data Processing

The core feature of the LIDAR is the detector unit and data processing. The detector unit detects laser light reflected from an object. Based on the time taken by laser light to make the trip from source to detector, distance of the object from the sensor is calculated. The detection unit needs to be sensitive enough to pick up weak laser light and process it without loss of information. This can be achieved with high speed processing platforms like NVIDIA Jetson TK1 and design of quality hardware for the detection unit.

## 3.2. Optional Features

### High quality user interface

A user interface provides ability to graphically view sensor data on a display screen. Use of a high-quality display capable of being housed on the LIDAR improves the portability of the product and allows it be used for handheld applications.

### Object Detection and Marking

The addition of object tracking and marking algorithms to the project’s software implementation further enhances the user experience of the product. This feature particularly helps detect and mark boundaries of known objects through video and image processing techniques. Such a feature would provide the user a better perception of the obstacle and its distance from the LIDAR.

## 3.3. Non-functional features

### Augmented Reality

The ability to integrate camera feed to the user interface and graphically represent LIDAR data proves to be a very useful feature. The feature allows the user to view the object in front of the LIDAR module with its distance. This enables user to get a clear picture of the target object when multiple objects are present in the field of view. This feature requires development of computer graphics and use of computer vision algorithms via OpenCV library.

# Dependencies and Deliverables

## 4.1. Dependencies

* The main challenge in our project is capturing the images with the rotation speed comparable to that of the existing LiDAR
* Processing the image for measuring the distance correctly
* Hardware support to interface servo motor using PWM
* Hardware support for prototyping the assembly
* Camera feed quality will affect the correctness of the distance sensing algorithm

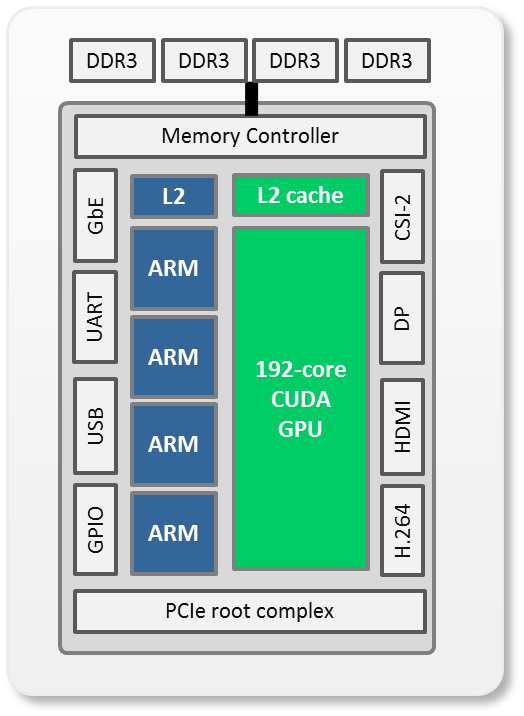
## 4.2. Deliverables

* Augmented reality (computer vision) based LIDAR prototype model on NVIDIA Jetson TK1 development board using industry grade camera
* Software package for enhanced sensing with augmented reality features.
* Achieving the capture speed of the images by the camera comparable to the speed of the existing LiDAR is the zenith aim of our project.
* Measuring the correct distance of the object in front of the complete unit is the main goal of the project.

# Project Architecture

3D mapping and sensing of the environment is the most researched topic in the automotive industry today for the successful navigation of the Self Driving car. The LiDAR is a device which can be used to sense the objects around us. However, it cannot give the exact details of the object. A camera acts as the vision of the car and helps the car capture the objects. A combination of the two can help the car map the object and also detect the exact distance. The image captured by the camera requires heavy image processing which can be handled only by a GPU. Hence we chose the NVIDIA TK1 as our platform. On the Jetson TK1 we will interface the camera, servo motor and the laser pointer. The camera will be made to rotate 360 degree with the help of the servo motor. And the intersection point of the laser source with the image taken by the camera will help in measuring the distance of the object from the camera. The project requires the detailed understanding of the hardware and the software to be used. This chapter describes the hardware and the software which we are going to use to implement the project.





Camera Feed



Servo Motor

Jetson TK1 Board

Laser Source

Figure . System Block Diagram

## 5.1 Jetson TK1 Hardware Architecture

The NVIDIA TK1 is the platform on which the project will be implemented. TK1 is an ideal platform for this project because the board is embedded with an on board Graphic Processing Unit(GPU) and has all the peripheral support required for the successful execution of the project. The NVIDIA TK1 has an onboard fully programmable NVIDIA Kepler GPU. GPUs are the state of the art parallel processors which can deliver 100s GFLOPS of performance. The GPUs offload the CPU from compute intensive tasks and hence leave more bandwidth on the CPU for the sequential processing. The major difference between a CPU and a GPU is that the CPU consists a couple of cores with a complex pipelined structure. The CPU has branch prediction and out of order execution, while a GPU has a relatively simple structure but is powered by hundreds of cores. The TK1 Kepler GPU consists of 192 such cores and can provide 300GFLOPS of 32-bit floating point computations.

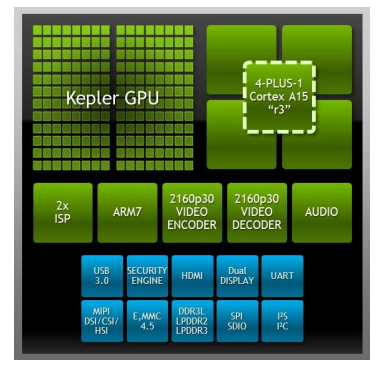


Figure 2. TK1 Embedded Board Top View



Figure 3. TK1 Architecture and USB Camera

The TK1 has a 32-bit quad core ARM Cortex A15 MPcore R3 CPU. The CPU has low power cores for better power efficiency. TK1 has 2GB RAM and 16 GB of onboard storage.

## 5.2 Jetson TK1 Software Architecture

On the software side the TK1 runs on Linux for Tegra(L4T) a modified package of Ubuntu 14.04. The TK1 supports CUDA 6.5, OpenGL 4.4 and NVIDIA VisionWorks ToolKit. Vision Works is an SDK which provides algorithms for NVIDIA-CUDA devise which are used to run OpenCV 3.0 applications. The board is also equipped with USB 3.0 and Ethernet ports which can be used to interface the camera to the board.

## 5.3 Servo Motor

The servo motor we are using is the HS-645MG ultra torque servo motor. It is a very reliable and robust motor. It has 3 metal gears, dual ball bearing and four adjustable splined horns. The motor will be mounted on the board and will help the camera to rotate and take 360 photos.

## 5.4 Project Software Architecture

The project will be on C/C++.However for image processing we will use OpenCV and to give the code the capability to process parallelly on the GPU we will use CUDA.

OpenCV: Open Source Computer Vision(OpenCV) is a library available in C/C++/Python useful for real-time computer vision. It has a range of inbuilt APIs which can be used for image/video capturing and processing. Motion understanding, object detection, facial recognition systems etc. are some of the fields where OpenCV is widely used.

CUDA: CUDA is a parallel computing platform and an API model created by NVIDIA. It is used for general carrying out purpose computations on the GPU. CUDA gives the complete freedom to the developer to choose the memory region on the GPU on which he wants to store his data. This makes the computations faster and flexible.

OpenCV contains a GPU module that contains the CUDA APIs for the GPU. With this module, it can accelerate almost all the APIs of the OpenCV. The code we will write in OpenCV can be made to run on the GPU with these APIs resulting in faster image processing and better performance.

# Project Design

# QA, Performance, Deployment Plan

## 7.1 Testing

One of the crucial aspects of product development is extensive testing and quality check to ensure high standard of performance. All such quality assurance activities, testing methodologies, test plan and deployment plan are discussed below. The design and implementation will be tested with different methodologies, under different test sets to eliminate bugs and improve system performance.

The scope of the test plan includes testing graphics features, image and video decoration features, motor drive mechanism, LIDAR detection and processing, sample acquisition rate test and object detection with marking test. The test plan covers different test methodologies like unit test, integration test, functional test, performance test and system test. The test methodologies and their test coverage is explained as below:

### Unit Testing

This involves the testing of each unit used in the project. Each unit is tested individually for proper operation and functionality verification. The test is done manually with future implementation of an automation script to speed up the process. Each functionality is tested extensively to ensure that the test results match with the expected outcome. Unit tests include 3D graphics test, image capture test, screen decoration test, video capture and decoration test, laser source test, laser detection unit test, motor drive test etc. Some of the unit tests and their expected outcome are as follows:

|  |  |  |
| --- | --- | --- |
| Test case No. | Test Description | Expected Results |
| UT\_01 | 3D Graphics Test | 3D graphics should be displayed on the screen |
| UT\_02 | Image and video capture test | Capture image and video from camera and display on window |
| UT\_03 | Motor test | Rotate motor with speed and direction control |
| UT\_04 | Screen decoration test | Graphic screen should be decorated with variable sized color patch |
| UT\_05 | Laser source test | Laser source powers on and emits laser light |
| UT\_06 | Laser detector test | Detector unit detects the laser light from source |

Table 1. Unit test plan

### Functional Testing

Each functionality of the product is tested to ensure that the feature performs as expected and does not deviate from standard outcome. This methodology involves testing features comprising of several units working in unison. These tests ensure that positive, negative and boundary test cases are considered to ensure that, the systems behavior and features are as expected.

### Integration Testing

After unit tests and functional tests have been successfully executed, the individual modules are interconnected and integration tests are carried out. These tests will help verify interactions between modules, verify module interfaces, data flow and control flow for the product. This methodology will involve both black box testing and white box testing. Data and control flow from LIDAR to the processing node will be checked with these tests.

### Performance and Stress Testing

The LIDAR will be tested extensively to check for performance improvement compared to state of the art technology. Measurement of obstacles at different distances ranging from few meters, up to 15 meters will be done to determine the improvement. Lower the time taken by the LIDAR to acquire samples and process the data helps achieve higher speed of operation and better precision of measurement.

This feature will be tested to ensure that the processing power is correctly utilized. These two metrics help determine the overall performance of the product. Furthermore, stress tests will be conducted to ensure greater life span for motor driver, laser source and laser detection units. Stress tests ensure that the product can provide reliable results for longer duration period.

### Acceptance Testing

Acceptance tests help decide, acceptability of the product based on requirements set forth earlier. This involves testing and verifying functional and non-functional requirements to ensure the product is acceptable.

The following test cases will be tested as part of the acceptance testing.

|  |  |  |
| --- | --- | --- |
| Test case No. | Test Description | Expected Results |
| AT\_01 | Laser light detection for different distances | Laser detector unit should be able to detect light up to 15 meters |
| AT\_02 | Distance measurement | Precise distance should be measured (with minimal error) up to 15 meters |
| AT\_03 | Motor driver | Motor should rotate at desired RPM to effectively capture reflected laser light |
| AT\_04 | Housing test | Mechanical housing should be able to remain intact and retain its form during operation |
| AT\_05 | Graphics display | Cube and screen graphics displayed on display |
| AT\_06 | Video capture test | Video should be captured and displayed on the graphics screen without lag (Good frame rate) |
| AT\_07 | LIDAR data mapping to graphics and video stream | Distance measured by LIDAR should be displayed as color patch around the cube |
| AT\_08 | Data communication test | Embedded server should be able to receive data from LIDAR on local area network |
| AT\_09 | Object detection and marking | Human in the video frame on the screen should be detected and his boundary should be marked using augmented points |

Table 2. Acceptance test plan

### Issue Tracking

To track issues in the system and manage them a google document is created. The document is shared across the entire team to allow everyone the ability to modify the document and update status.

### Testing Schedule

|  |  |  |
| --- | --- | --- |
| Test Type | Planned Dates | Time in hours |
| Unit Testing | 10/25/2016 – 12/15/2016 | 60 hours |
| Functional Testing | 12/01/2017 – 02/15/2017 | 80 hours |
| Integration Testing | 02/20/2017 – 03/15/2017 | 50 hours |
| Performance Testing | 03/20/2017 – 04/20/2017 | 65 hours |

Table 3. Test Schedule

## 7.2 Deployment Plan

The LIDAR system is interconnected on a network to be able to achieve centralized processing of distance information. The system consists of the LIDAR and a server over the network with various other nodes The deployment plan for both is as follows.

**LIDAR Application Deployment**

1. Download the software package for LIDAR distance sensing and augmented reality features from GIT repository. Optionally object detection and marking can be enabled in the package.
2. Install the package on the JETSON TK1 and reboot. An automation script runs the client side program upon channeling data from LIDAR to server node.

**Server-Side Application Deployment**

1. Download the package for server-side from GIT repository. The package contains code to handle the data from the LIDAR and further process it.
2. Install the package on the server and reboot.

# Implementation Plan and Progress

|  |  |  |
| --- | --- | --- |
| Sr. No. | Implementation Plan | Status |
| 1. | OpenCV 3.1 programming environment set up -   * + OpenCV 3.1 installation   + Environment variable setup for easy compilation and execution   + Finalizing naming convention for programs | Completed |
| 2. | Nvidia Jetson TK1 hardware (target) platform understanding   * + Schematic study from datasheet   + Device driver setup study | Completed |
| 3. | OpenCV 3.1 setup on Jetson platform   * + Download and compile OpenCV source code | Completed |
| 4. | OpenCV sample program understanding   * + Compile and execute camera interface, graphics design sample programs | Completed |
| 5. | Existing LIDAR module study   * + Interface existing LIDAR module with Jetson board   + Investigate the features and drawbacks of available module | Completed |
| 6. | Market survey for Camera module based on   * + Compatibility with Jetson   + Frame Rate   + Image Quality | Completed |
| 7. | Webcam interface with Jetson Platform   * + Interface webcam to test OpenCV program | Completed |
| 8. | Image and Video Capture Implementation   * + Implement image and video capture OpenCV program   + Test the software on Jetson platform | Completed |
| 9. | 3D graphics development   * + Realize OpenCV program to generate 3D graphics using 3D transformation pipeline | Completed |
| 10. | Video/Image Storage   * + Complete the software to store the images and videos | Completed |
| 11. | Images and videos on 3D graphics   * + Display captured videos and images on 3D view at different viewing angle | Completed |
| 12. | CUDA platform compatibility   * + Port OpenCV programs on CUDA platform for faster execution | In progress |
| 13. | Prototype assembly design   * + Design assembly of camera, laser and laser detection module | In progress |
| 14. | Servo motor selection and interfacing with Jetson   * + Market survey for servo motor based on     - Stable 60 RPM     - Long operational hours     - Compatibility with Jetson | In Progress |
| 15. | Laser detection module interface   * + Integrate laser detection module with Jetson board | Planned |
| 16. | Distance measuring   * + Write an algorithm to measure distance in 360 degrees using laser detection module and servo motor interface programs | Planned |
| 17. | Augmented reality   * + Implement augmented reality to display measured distance on the video feed for better user interface | Planned |
| 18. | Regression testing   * + Complete testing of prototype | Planned |
| 19. | Machine learning for object detection   * + Installing Caffe deep learning framework   + Training and modifying deep learning model for object detection primarily humans   + Integrating new machine learning algorithm with GPU Enhanced LIDAR | Planned |

Table 4. Project Implementation

# Project Schedule

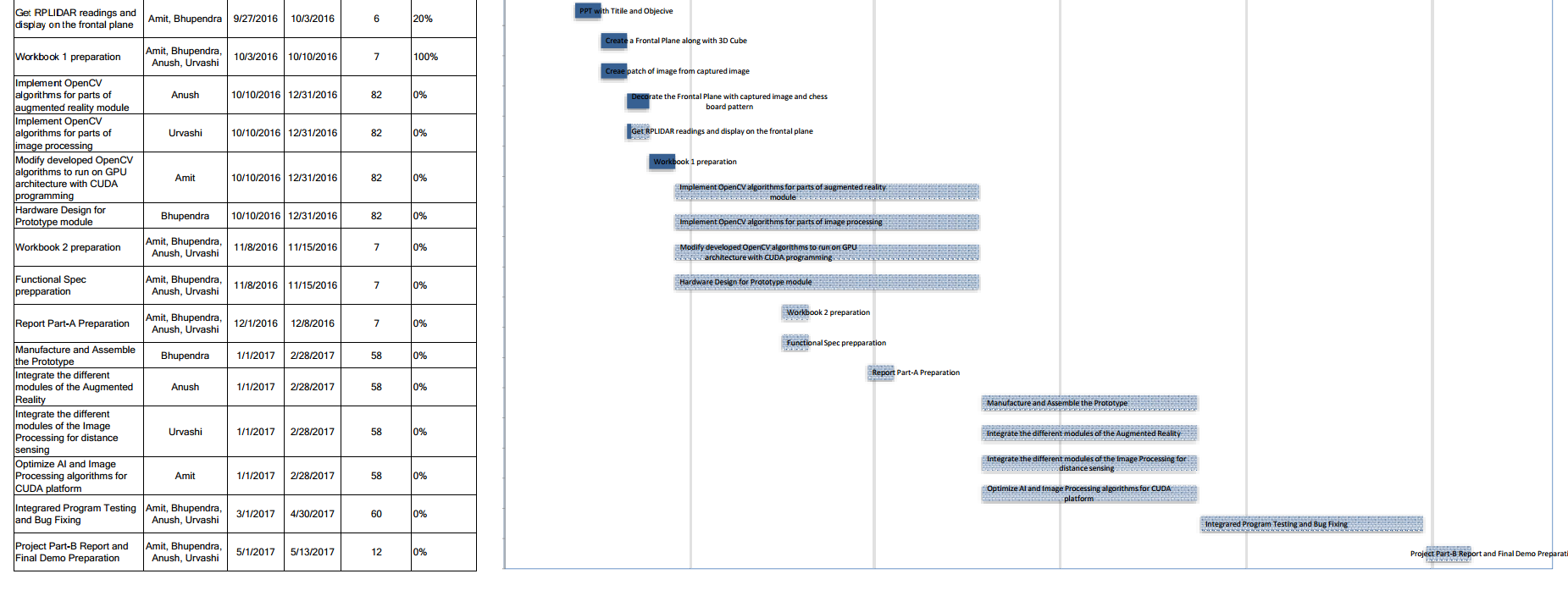
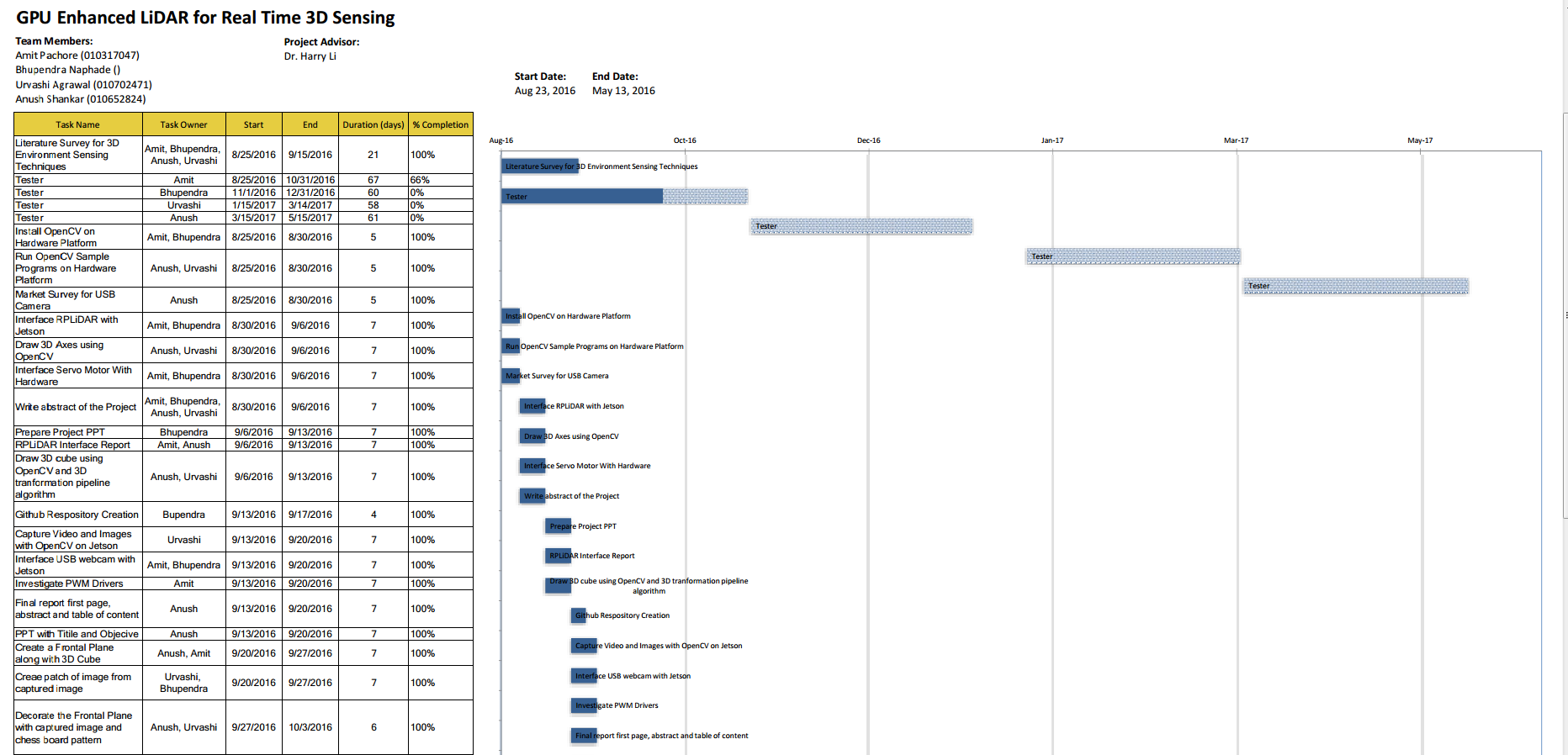


Table 5. Project Task Assignment and Tracking

# 

# 