CS CAPSTONE

FINAL REPORT

DEPTH SENSING WITH COMPUTER VISION AND LIDAR

TUESDAY 12TH JUNE, 2018

PREPARED FOR

D. KEVIN MCGRATH

BY

GROUP 69

DSCVL-OVERCOMERS

LUCIEN-ARMAND T. TAMNO

Abstract

This Depth Sensing with Computer Vision and Lidar report describes the project development from its historical genesis to the final result. The report includes the chronological development,shareholders, participants and their different roles, technical and management skills learned, as well as reflections.

CONTENTS

1 Table of Contents

Contents

2 Definitions

IR:

IR stands for the infrared technology.

IR Depth Sensor:

A device that calculates distances by emitting infrared signals.

LIDAR:

Light Detection And Ranging - A method that uses lasers to measure distance

Microsoft Kinect:

A product that uses an IR Depth sensor to measure distances.

Logitech Brio Webcam: [?]

The webcam model this project shall be using.

RPLidar A1: [1][**?**]

A low-cost LIDAR unit that this project shall be using.

RPLidar Solid-state: [**?**]

A High-cost LIDAR unit single direction with better performances of detecting, measuring and locating liquids and people

Computer Vision:

The methods for acquiring, processing, analyzing, and classifying digital images and extracting information.

Python/C API: [**?**]

API: application programming interface between Python and C programs

DSCVL:

Acronym that stands for Depth sensing with computer vision and lidar

3 Introduction

The DSCVL project today a solution, was originally our client idea to have a new technology by overlaying two signals from different sources. The first signal being captured the existing Lidar technology and the second signal coming from a camera technology. Therefore, the goal of project was to turn the descriptive above concept in a viable application to address the gap left by the infrared technology in outdoor environment.

In fact, the Lidar technology is reliable in outdoor environment and capable to provide with precision objects distance measurements. And one of the reliable device uses by this technology is the 16-Segment SOLID-STATE Lidar equipment, at times called the Leddar M16 8. The second technology we use in this project is the webcam specifically the Logitech Brio Webcam 8 and it is used to get images.

More importantly, the fact that the lidar technology (Leddar M16) was more reliable to provide better objects measurements than the IR technology 8, was presented to us the ideal technology of choice. so both, Lam and I as suggested by Kevin our client, decided to use it.

4 Current State of the DSCVL

Though, the past winter term progress report described the RPLidar A1 capability 8 of outputting useful information but the reality is that the DSCVL project needed some changes for the sake of better accuracy. some changes took place at different level of the project. And the most important being the replacemnt of RPLiadr A1 by the 16 segment solid state M16 that provides better detection, localization, and distance measurements when is to compares to the RPLidar A1.

Additionally, those differences can even be more tangible at the code level, because the M16 model has its code literally written in C program while the RPLidar code is closely related to the python environment. And this M16 factor code-based caused me to conduct researches to figure out how can I make the C program code work in python environment. The result of that research was to discover the python **??** functionality to play the intermediary role of having the C program code runs in a python environment.With the same idea, I dug a little bit more to finally find out that Visual Studio is the suitable Windows application to implement the python/C API.

4.1 Preliminaries and the Python/C API implementation

4.1.1 Are there any special hardware, OS, or runtime requirements to run your software?

To run the Leddar M16 technology, I have to implement the python/c API2 using the C-code that was shipped with the M16 device. And to do so, I have to make sure first that the M16 can properly run in a Windows environment by default. And for me to test that functionality, I installed and configured the default M16 settings, as shown in figure 1 below.

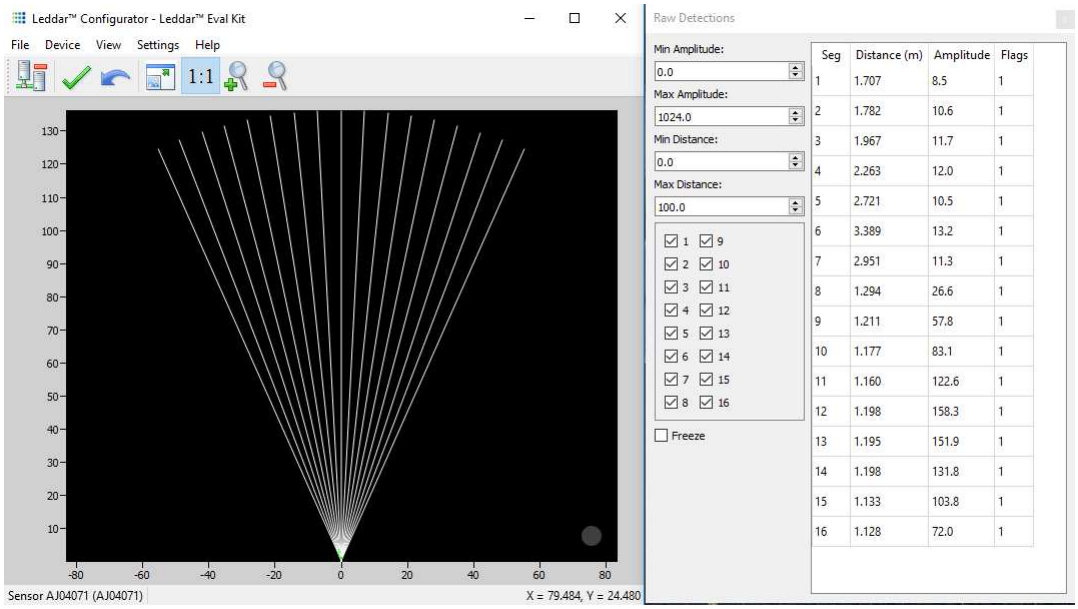

Fig. 1. figure
Leddar M16 outputs distance measurements using the default windows installation [right] and default M16 screen [left].

That figure 1 displays a sample distance outputs as it is supposed to be visualized by a user at the final stage of the DSCVL project. Though, differences may be seen in term of data disposition on the screen,because the M16 code has to run in conjunction with webcam technology to overlay images form the python environment. Hence, the implementation of the DSCVL application has an additional technological constraint to be done via an API.

4.1.2 How does your project work?

To get the written C++ language code to python, the Visual studio 2017 application was the relevant tool to use. Therefore, I installated visual Studio in Windows 10 system, imported any necessary library to connect the visual Studio to the Leddar M16 C-code,using the appropriate [**?**] documentation related to the topic. The remaining bulk of work is to write and debug the API implementation code to make sure the M16 interacts properly from the visual studio perspective and secondly, to integrate in a single package the python and the M16 C codes, as shown in figure 2.

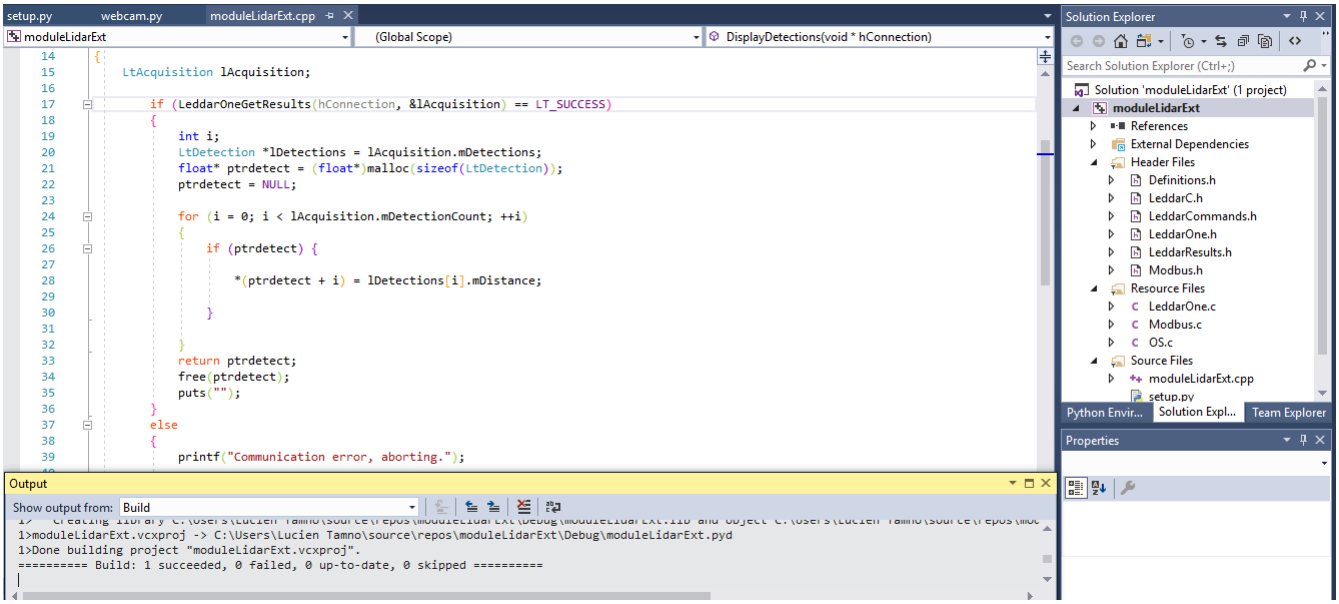

Fig. 2. figure
Python/C API built in Visual Studio 2017.

However, when working on the above API, I encountered multiple setbacks.among others, the two most important issues are:the slight Leddar M16 differences seen in the the M16 files, for instance I had to change one default header, as shown in figure 3 below.

```
#pragma once

#ifdef WIN32
  #ifdef LEDDARC_LIBRARY
  #define LEDDARC_DLL __declspec( dllexport )
  #else
  #define LEDDARC_DLL __declspec( dllimport )
  #endif
#else
  //#define LEDDARC_DLL __attribute__ ((visibility ("default")))
#define LEDDARC_DLL __declspec( dllimport )
#endif

#ifdef __cplusplus
extern "C"
{
#endif
```

Fig. 3. figure
Snippet code changed in the Leddar M16 header file.

And the worst being found in the current debugging phase that presents a successful result in code checker but generates several linkage errors.

Finally, I also wrote an additional decorator function in the python module to speed up the python execution module as shown in figure 3here underneath.

setup.py webcam.py moduleLidarExt.cpp

```python
import numpy as np
import cv2
import os
from ctypes import *
from os import sys
from moduleLidarExt import distance



api_data =[]

def executer_distances(function_2_exe):# function to execute any callback function passed as argument
    def lidar_update_distance(*args, **kwargs): # wrapper function
        return function_2_exe(*args, **kwargs)  # function to execute
    return lidar_update_distance


@executer_distances
def distanceZaxis():

    api_data = distancelidar  # storing API returning values
    i = 1
    for value in api_data:
        print('{} distance value is {}'.format(i, value)) # test of values return by the API
```

Fig. 4. figure

Python decorators code.

4.2 Legacy: GUI Interface

In order to prioritize the DSCVL backend, the Frontend was relegated because the backend is more important and represent the project core functionality as envisioned in the final stage. The GUI downgrade, yet still provides the basic services it has before, but it is not clear whether the final DSCVL system would make use of the legacy graphical user interface.

5 Final Gantt Chart

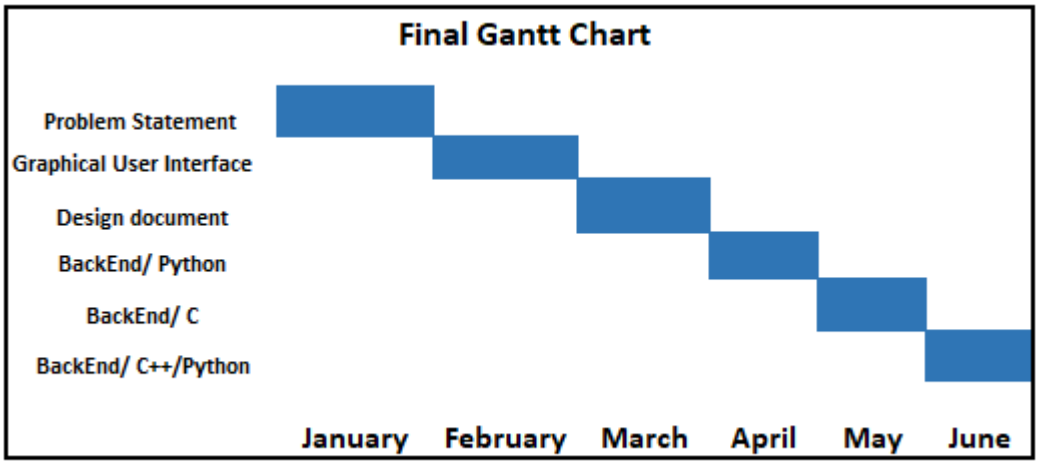

Fig. 5. figure

Final Gantt Chart

6 Design Document

7 OVERVIEW

As a roadmap, this document is to help its users understand how to build accurate overlay images from two existing, reliable technologies with distinctive features. The Lidar technology description would help the reader to understand how accurate and reliable are the pulsing laser distance measurements in this outdoor environment while the Webcam would aide to get the two-dimensional image to overlay to the depth Lidar scanner sensor data inputs.

7.1 Scope

This document is to describe a new application that will combine both the Lidar technology to Webcam stream technology for specific overlay images to define accurate objects distances in outdoor environment to be utilized by the stakeholders and specially client of the DSCVL project.

7.2 Purpose

The purpose of this document is to elaborate on technical aspects of the software to implement, to allow stakeholders of DSCVL project to understand what the Lidar and camera technologies do the best, each technology characteristics,their differences and what is used to make these technologies work together in order to accomplish the ultimate goal that is to get an overlay image as required by the client.Thus, the technicality of the project different compounds would be described with great details in the following sections.

7.3 Intended Audience

The content of this document is to primarily address the needs and requirements of the depth sensing using with computer vision client in particular and can be added more broadly, stakeholders and even generally readers with interest on this descriptive document.

8 DEFINITIONS

IR: IR refers to the infrared light spectrum.

IR Depth Sensor: A device that calculates distances by emitting infrared patterns.

LIDAR: Light Detection And Ranging - A method that uses lasers to measure distance

Microsoft Kinect: A product that uses an IR Depth sensor to measure distances.

Logitech Brio Webcam: The webcam model this project shall be using.

RPLidar A1: A low-cost LIDAR unit that this project shall be using.

RPLidar Solid-state: RPLidar: A High-cost LIDAR unit single direction with better performances of detecting, measuring and locating liquids and people

Computer Vision: The methods for acquiring, processing, analyzing, and classifying digital images and extracting information.

GUI: GUI: Graphical User Interface

Videocapture: videocapture: Stream of subsequent images

DSCVL: Acronym that stands for Depth sensing with computer vision and lidar

SDD: software design descriptions as defined by the IEEE 1016

9 Design Description Information

9.1 Introduction

The design description information presents to the reader what kinds of information the DSCVL processes, what part of the Lidar or the camera technology handles what and how.

9.2 Design Description Identification

Essentially,there are two blocks or components that are ought to interact to make the Depth sensing using computer vision become a feasible project, and those two components are the hardware and the software. The hardware uses the robopeak lidar A1 scanner and the Logitech Brio webcam of which, we provide more details after. The software component basically having a bunch of open source libraries among others the OpenCV library, the Numpy library and others.

9.2.1 Object processing At the hardware level

9.2.1.1 **Robopeak Lidar or RPLidar Inputs**: We use the RPLidar as input device to our system to provide accurate measurements of distances via its sensors. In detecting, locating and measuring objects that include people and liquids, the Lidar's scanner challenges the limits imposed by the same objects to other technologies such as the commercial infrared technology that uses depth sensors in a natural outdoor environment but is confused by the sunlight. However, as depicted by the figure1 underneath, the most effective RPLidar model to use would be the solid-state, for it has no additional time caused by the spinning engine rotation as seen in the 360 RPLidar A1 model.

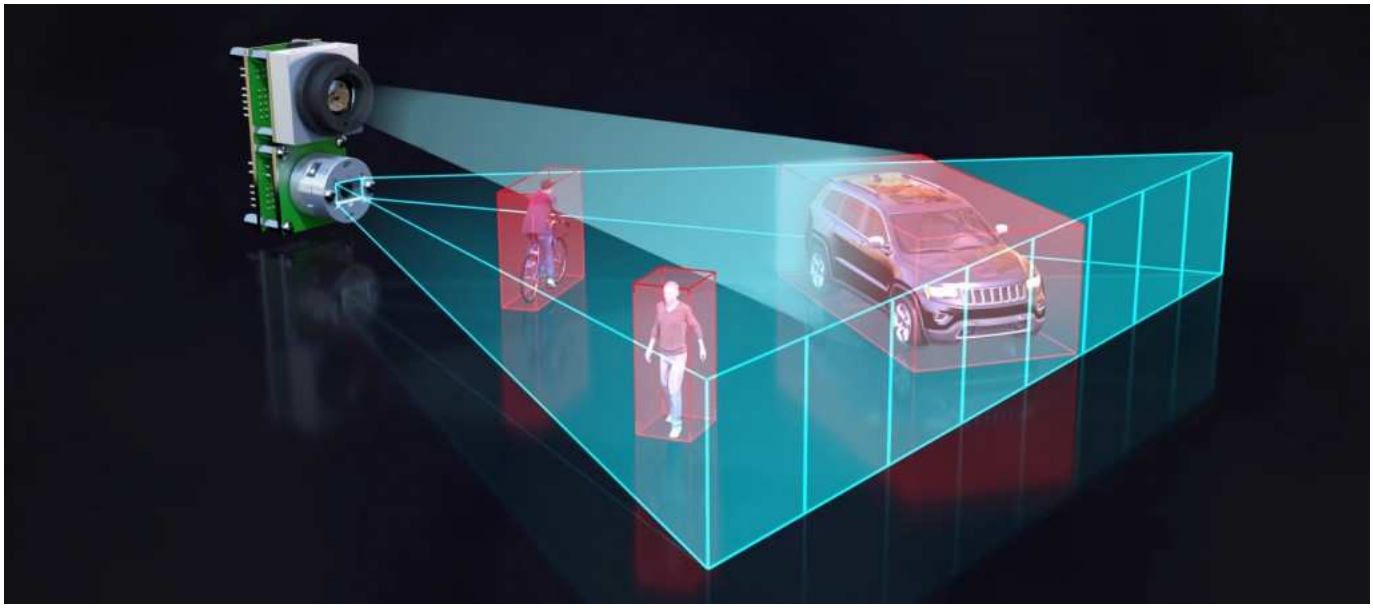

Fig. 6. figure

LIDAR SENSOR FUNDAMENTALS, derived from *leddartech* [1]

9.2.1.2 **The Webcam Inputs** : The camera is the other input device source that we will use to get video streams to match to the RPLidar sensors laser pulses, that enables us to see what kin of object the system is processing. And both Lidar scanner and the webcam having their data process at the software level.

9.2.2 Object processing At the software level

what we are using at the software level is a array of libraries and most are python library dealing with images. we have for instance, the OpenCV library through which images found in the system would be isolated from the stream of images, processed and ultimately painted on a screen.

9.2.3 Library Structure

The library structure aims to integrate different software functions and defines the path to the final overlay image implementation.In technical terms, the library structure is known by the name API.

9.3 Identified design stakeholders

The depth sensing using computer vision and Lidar project stakeholders are users interested by what the Lidar technology combines to the webcam image processing can accomplish in either production or experimental environment.

9.3.1 Image Data processing

The image data processing has to comply with user predefined conditions and take into account physics and mathematics parameters to yield excepted outcomes as shown by the following figure for which a scanner rplidar A1 sensors mapping is capturing targeted point object.

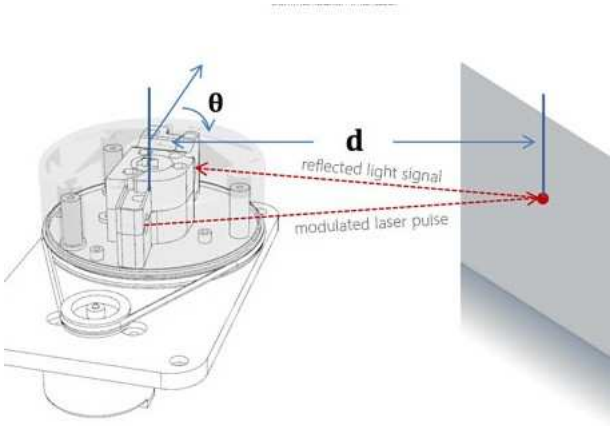

Fig. 7. figure

, derived from *seeeds* [2]

9.3.2 the principal stakeholder

When it comes to the question to know who is the principal stakeholder of this python-based project, the response goes without saying that our client is D. Kevin McGrath, who has the property right on the future final DSCVL product and therefore, all the project specifications have to meet his requirements.

9.3.3 the other stakeholders

May fall into the category, users of the Lidar technology to either import new features in production environments or those who would likely get the outcomes from the DSCVL software.

10 SELECTED DESIGN VIEWPOINTS

The selection of design viewpoints aims to define and present how the new system would likely interact with its environment, address concerns from its users,choose the language to use as well as depict relations between parts.

10.1 Interface viewpoint

The virtual visible interface would be the user interface that accounts for what kinds of information a user would like to see displayed as the output.Basically, the user would like to have an overlay images in a range within the value defined on the user interface. That said and regardless the OS that hosts the DSCVL application, the user interface has to be standardized across platforms and its components interaction process as shown in the following figure.

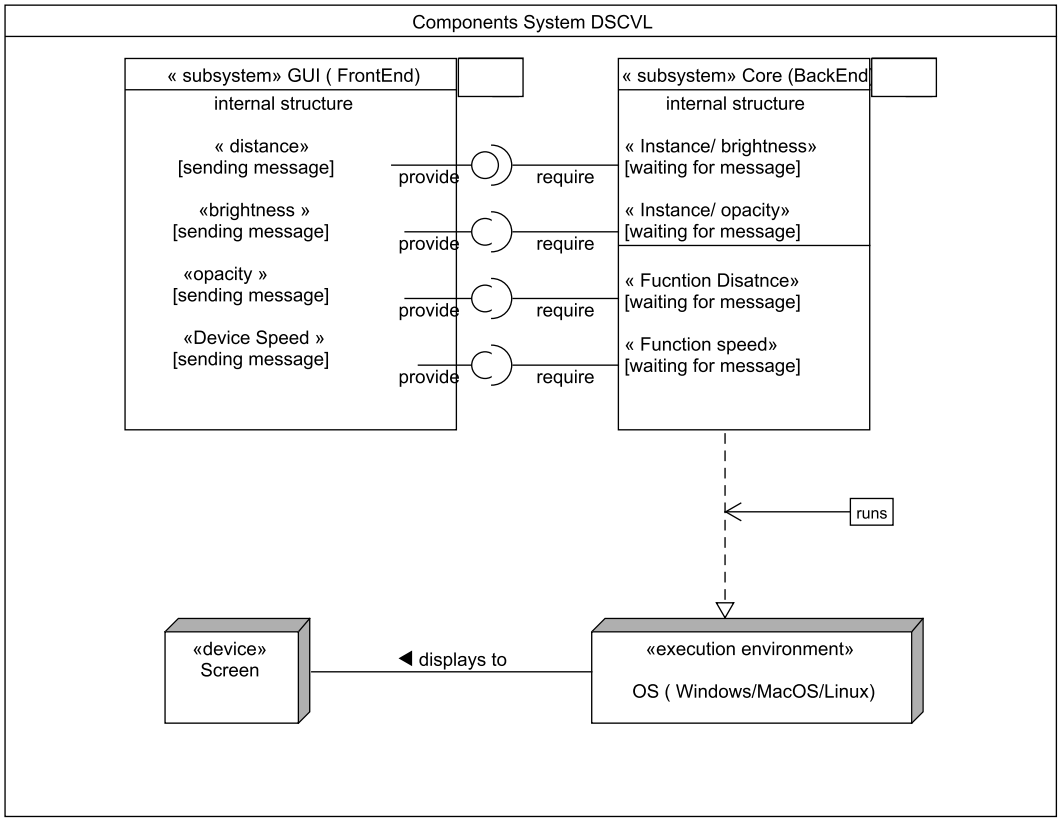

Fig. 8. figure

UML Component diagram for DSCVL

10.1.1 Design Elements

Other elements of the design are libraries encompassing the almost entire backend. However, this set of libraries is made to be transparent to the user but available to developers.

10.1.2 Design languages

The application to be created would be made in python language though in addition the language, other programming languages may be added such as the C++ language (if necessary) in order to run subroutines. Among others, the openCV library plays an important role in the DSCVL software implementation.

10.1.3 Design Elements - OpenCV and Library Structure

One sample code functionality of the openCV library implemented in the DSCVL application would look something like this(code hereafter):

```
from rplidar import RPLidar

PORT_NAME = '/dev/ttyUSB0'


def run(path):
    '''Main function'''
    lidar = RPLidar(PORT_NAME)
    outfile = open(path, 'w')
    try:
        print('Recording measurments... Press Crl+C to stop.')
        for measurment in lidar.iter_measurments():
            line = '\t'.join(str(v) for v in measurment)
            outfile.write(line + '\n')
    except KeyboardInterrupt:
        print('Stoping.')
    lidar.stop()
    lidar.disconnect()
    outfile.close()

if __name__ == '__main__':
    run(sys.argv[1])
```

record_measurments.py(code lines that capture scanner measurements)[3] and the above code is a snippet code that also portrays the internal program structure. Which on module is to capture data from the RPLidar device that has features as described by following figure.

Specs

Measurement Performance

Item	Unit	Min	Typical	Max	Comments
Distance Range	Meter(m)	TBD	0.2 - 6	TBD	White objects
Angular Range	Degree	n/a	0-360	n/a	
Distance Resolution	mm	n/a	<0.5	n/a	<1.5 meters
			<1% of the distance		All distance range*
Angular Resolution	Degree	n/a	≤1	n/a	5.5Hz scan rate
Sample Duration	Millisecond(ms)	n/a	0.5	n/a	
Sample Frequency	Hz	n/a	≥2000	2010	
Scan Rate	Hz	1	5.5	10	Typical value is measured when RPLIDAR takes 360 samples per scan

*Note:triangulation range system resolution changes along with distance change, the below chart showed the theoretical resolution change of RPLIDAR:

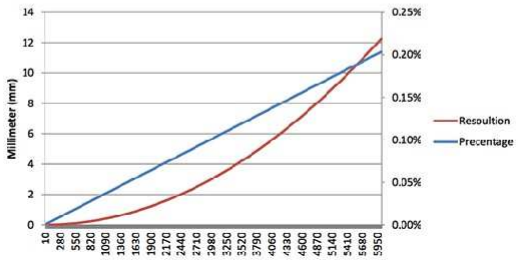

Fig. 9. figure

RPLidar device[above]- measurement performance [underneath], derived from *seeds*

10.2 Dependency viewpoint and diagram

This part is to see how the whole system parts come together as a single entity to perform the require work of getting the overlay image. We have the Interconnection, sharing, and parameterization of Logitech Brio Webcam and the RPLidar scanner with syncing services happening at the coding level.

Therefore, we have the following diagram to represent those services.

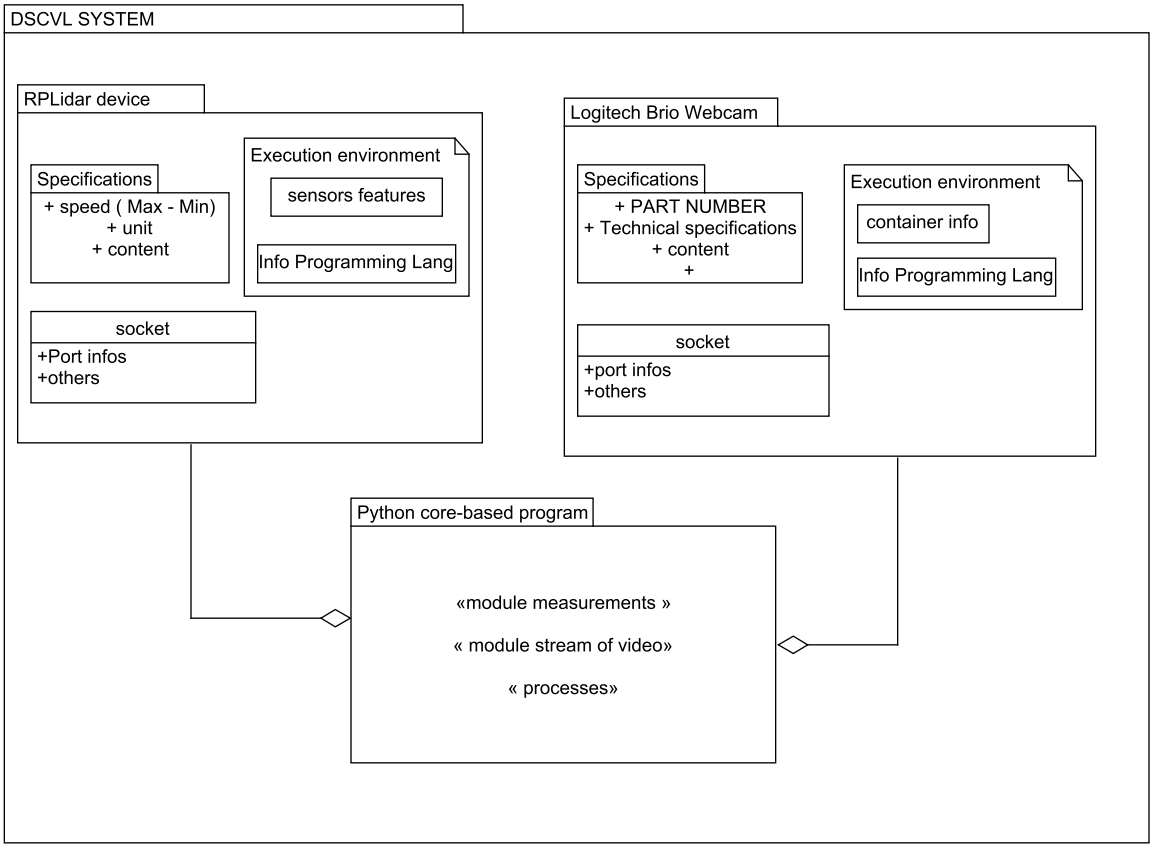

Fig. 10. figure

UML package diagram and component diagram

11 REFERENCES

[1]: https://leddartech.com/technology-fundamentals/

[2]: https://www.seeedstudio.com/RPLIDAR-360-degree-Laser-Scanner-Development-Kit-p-1823.html

[3]: https://github.com/SkoltechRobotics/rplidar/blob/master/examples/record_measurments.py

12 Weekly Blog Posts over six Months

12.1 Ten-Week Term Retrospective

Week	Positives	Deltas	Actions
1-Jan	-	-	-
2-Jan	• back and forth email communication with Kevin regarding to way to get into the CS 462 and CS 406	• Kevin emailed me back regarding the way i wanted to work in a new project: to whether i have decided to work with a partner or to do it in solo. • The next day exactly on Friday, January the 19th he informed me that he is making an arrangement.	• I emailed Kevin to know what would the next step after registering to the cs 406 and cs 462. • I replied to Kevin that " working with a partner would be beneficial to me and allow to simulate a real-world experience.
3-Jan	• This week, the Group 69 is born • For the first time I had a fruitful exchange with my partner Lam	• At this point, I did not know to jump onboard and get started with the project though i could note a positive interaction with Lam.	• i reached Lam out to get the insight about the new project and for the first amazingly, the promptly respond to my email by outlining the key points of the Depht sensing with computer vision and lidar project, which at that times was titled ROS lidar project. • I was encouraged by Lam to reach out Junki as he was introduced to us by Kevin as our TA and ultimately Junki accepted to be our TA.
4-Jan	• if week 3 was the week that our group was born, week 4 actually indicates when i effectively got started on searching what could be one way of solving the DSCVL problem.	• I got into my first struggle trying to install the python libray on Windows 10 • I had no idea how to get the rplidar library worked on Windows.	**weekly tasks list** 1) Install RPLiar python library [5] 2) Build the appJar file from the appJar library for the GUI [3]
5-Fev	• noticeable progress done by building python libraries and configuring paths to properly work in windows environment	• I met with Kevin during the weekly office hour session and put forward an sketched figure of how I was designing final overlay image and also how would the relative distance of the object to be captured by the Robotpeak lidar device.	**weekly tasks list** 1) Download and install Numpy wheel files 2) Download and install OpenCV wheel file 3) Configure " pip" tool to run with the Windows 10 4) explore how to work with tkinter library

6-Fev	• week for the midterm progress report write up • built the pdf slide presentation file along with its video	•	**weekly tasks list** 1) Turn in midterm process report (slide, write up, audio recording) 2) wrote the first function to capture the video stream from the my built-in webcam and store images on the Hard drive.
7-Fev	• explore other possibilities to build a virtual Linux layer on top of the windows OS. • work with the Group 40 learder to have a private critique poster session	• Faced a serious setbadk to make the windows 10 system work with the rplidar device. • I was having buggy results regardless changes made in the source code program	**weekly tasks list** 1) Read documentation to perform test reading from the lidar device 2) Helped out by Kevin to successfully read data from the lidar A1 scanner
8-Fev	• we missed the class poster critique session, and planned to attend the one with Group 40.	•	**weekly tasks list** 1) raised questions that prompted me to go back and take a closer look at how the rplidar A1 is trying to read data and how i can utilize those data for Z-axis calculations.
9-March	• Poster Critique session meeting in Kelly Hall with Kirsten as supervisor, Group 40 team. • outcomes of the session/ poster format correction: the need of labeling the central image with axis was presented, Include group members contact information if necessary and include the client of the project. • explored some the benefits of the Tensorflow technology and in particular the ”eager execution” version	• I Was told by Junki and Lam that Tensorflow will not be helpful for the z-axis calculus.	**weekly tasks list** 1) contact Nathan leader of Group 40 for the final arrangement for the critique session 2) loop back to update my team partner Lam abot the critique session. 3) i successfully wrote functions to turn rplidar stream data into an array of float values. And then, for that array of float points to be array of values to represent the mobility of the object in front of the lidar device. 4) Install of the eager execution python wheel file on windows 10
10-March	• compute the z-axis object movements using geometrical maths functions • explore how i would be using the numpy spacial functions to get around with the standard maths functions [6]	• Still struggling to transform the array float points of data into 2-D array to pass it as one of arguments for the numpy function.	**weekly tasks list** 1) Install spicy spacial numpy wheel file on windows 10 2) write the final winter progress report 3) write the slide of the report presentation and build the video

11-Apr	• read the python documentation for C extension • explore the possibility to use command line for module extension	• still unable to clearly understand how C can be extended in python.	**weekly tasks list** 1) search more info in the python documentation 2) find information about static libray
12-Apr	• Find out website to provide samples of project where dynamic Library C/python is used in command lines.	• Still do not know how to interpret data form the M16 documentation.	**weekly tasks list** 1) Download and Install of the Leddar M16 exe file for Windows 10 (64 bits) 2) Configuration of the M16 in Windows 10 environment 3) Import the Ctypes function library in python
13-Apr	• commands to Create a dynamic library to link files in Linux • commands how to compile dynamic library in Linux	• keep searching how to implement DLL in Windows	**weekly tasks list** 1) save all the linux commands for DLL 2) sample command: and gcc -shared -o lnkfile.so file1.c file2.c ..filex.c
14-Apr	• Get the main website describing how to create C extension in python	• confront to the problem of Visual Studio.	**weekly tasks list** 1) Install Visual Studio But computer freezes 2) Perform some extra work on the computer to keep on with the project
15-May	• • Explore other possibilties to call Leddar M16 function.	• Erreur prompt by visual studio on the C function called	**weekly tasks list** 1) determine what function C to run on API 2) Define of the module python on visual studio 2017 3) document the first process to call function in Visual Studio
16-May	• examine how to include and link SDK files in Visual Studio (VS)	• Still facing failure issue at the python/C API and VS	**weekly tasks list** 1) Headers importation in VS 2) Write some main function in the API module
17-May	• explore other paths of writing the API	• drawback, the Seasnake implemenmtation could not work in VS 2017	**weekly tasks list** 1) Get around the Visual Studio issues of linkage 2) Attempt to use seasnake 3) document seasnake installation process 4) Keep working on the previous Python/C API

18-May	• Explore every single possibility to include SDK in VS	• Struggle with Windows System errors and problems of machine failure.	**weekly tasks list** 1) Maintain Machine and re-install the environment Visual Studio Windows 2) reconfigure the entire project 3) continue testing integration of the C Leddar in VS 4) successfully include SDK links in linker
19-June	• successfully compile of the ptython/C++ API	• Struggle to call the function Leddar in python code	**weekly tasks list** 1) Include all headers in the C/API 2) Include all files in different modules 3) Include links and other files in the VS properties option
20-June	• Narrow the problem of the module python not validating extension at the visual studio level	• Module python still not working	**weekly tasks list** 1) Write final report Presentation 2) write final winter progress report

12.2 How does one run it?

The Python/C++ API run but the module python to handle the call function from API accuses a Windows Visual Studio issue.

12.3 What web sites were helpful?

This needs to be detailed enough to recreate and/or use your project!

12.4 Recommended Technical Resources for Learning More

To dive into this topic, one can use links provided in the appendix section to learn more about the topic.

12.5 What web sites were helpful?

The bulk of the this project from start to finish was mostly done using websites as listed underneath. the first set of websites consists of sites used to develop the Python/C API and the second set lists sites related to libraries for python module.

12.5.1 websites for the python/Cpp API

1) python/C++ API guidelines Using Visual Studio[1]
2) Download File Leddard M16 [2]
3) Install Visual Studio 2017 [3]
4) Building C and C++ Extensions[4]
5) Foreign function library for Python[5]

12.5.2 other libraries for python

1) OpenCV[6]
2) Numpy [7]

12.6 reference books really Helped

Basically,some of the books I was able to glimpse a book Advance C++ programming[8]. But as I have said it in the section website that helped the most and my research was carried out via internet, on Windows web pages, python repositories and others like Github repository.

12.7 Other Party Involvement

When is to say who else was involved in this project except my teammate, I have to admit the fact that our client was helpful by providing some guidelines and paths for me to look into. I received from our client the main website in which all the first instructions related to the python/C API implementation could be found. And from that point on, though most of the student did not have any past experience with python and C working in visual studio IDE, I found one graduate student o whom I showed the python error to recognize the extension module from C and He tried to provide some instructions unfortunately that did not work.

13 CONCLUSION

Over the course of this project I was able to the first able to implement resident application API in windows environment, integrate python module in visual studio by specially customizing Python environment to work in Windows. Additionally, I was able to to engage in research activities whenever the project was not working to get the project going by googling or reading in other material suitable for the area of the project.

For non-technical skills, I learned to work in a team by considering the perspectives of the other group member. But beyond that, I learned to ask help by clearly explaining to the person to whom I needed help from, the challenge I was facing. Due to the hidden hurdles that he project work presented to me. And to elaborate about the project work, one of the key point i would always remember is that a project does not only need technical skills to be successful, but also non-technical skills as well. It is submitted to my attention to be solve but more importantly, it is an opportunity to learn new skills, to explore new ways to work with others, to accept inputs from them changes and adapt to those transformations that presented to me either as an advice or a reproach. The latter, can only be possible in a teamwork environment and good management is the way to go. By Management I mean, to learn how to handle stress that can be imposed to you by the amount work to do or the speed at which that work has to be done.

Finally, if I could go back and do this project all over this project, the first thing i would do is to get started on the python/C API because this module appear to be straightforward to implement but unfortunately that is not the case, Because the module has hidden aspects like how Visual studio interacts with Windows platform to provide services in python environment, which is one important phase of the python/C API development in Windows system.

14 APPENDIX

[1]python/C++ API guidelines Using Visual Studio

[2]Download File Leddard M16

[3]Install Visual Studio 2017

[4]Foreign function library for Python

[5]Foreign function library for Python

[6]: OpenCV

[7]: Numpy

[8] advance C++ programming [9]: legacy GUI

[10]: rplidar