Senior Design Capstone Spring Midterm Progress Report

May 5, 2018

Depth Sensing with Computer Vision and Lidar

Prepared for

Oregon State University

D. Kevin McGrath

Prepared by

Group 69

Kin-Ho Lam

Abstract

Depth Sensing with Computer Vision and Lidar proposes combining computer vision and lidar to create a reliable depth sensor. This document details its project member's progress toward a final design.

1 Table of Contents

Contents

References

[1] "Logitech brio webcam with 4k ultra hd video & rightlight 3 with hdr." [Online]. Available: https://www.logitech.com/en-us/product/brio

[2] T. Huang, "Rplidar a1." [Online]. Available: https://www.slamtec.com/en/lidar/a1

[3] "Leddar solid-state lidar: M16 multi-element sensor module." [Online]. Available: https://leddartech.com/modules/m16-multi-element-sensor-module/

[4] "tensorflow/models." [Online]. Available: https://github.com/tensorflow/models

[5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and et al., "Speed/accuracy trade-offs for modern convolutional object detectors," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] "Coco - common objects in context." [Online]. Available: http://cocodataset.org/

[7] "openimages/dataset," Jan 2018. [Online]. Available: https://github.com/openimages/dataset

[8] [Online]. Available: http://www.cvlibs.net/datasets/kitti/

[9] "Tensorflow framework & gpu acceleration from nvidia data center." [Online]. Available: https://www.nvidia.com/en-us/data-center/gpu-accelerated-applications/tensorflow/

2 DEFINITIONS

2.1 IR

IR refers to the infrared light spectrum.

2.2 IR Depth Sensor

A device that calculates distances by emitting infrared patterns.

2.3 lidar

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

2.4 Microsoft Kinect

A product that uses an IR Depth sensor to measure distances. Referred as a benchmark.

2.5 Logitech Brio Webcam

Webcam made by Logitech. [1]

2.6 RPLidar A1

A budget lidar device made by Slamtec. [2]

2.7 Leddar M16

A solid-state lidar device made by Leddar. [3]

2.8 Computer Vision

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

3 PROJECT PURPOSE

Infrared (IR) depth sensors such as the model used in Microsoft's Kinect 2.4 can quickly calculate distances in indoor scenarios. However, IR depth sensor readings can be obfuscated by other infrared emitting sources such as other IR depth sensors or natural sunlight. IR depth sensors cannot be used in self-driving cars, outdoor robots, or any any device that requires high accuracy distance measurement in varying conditions. Depth Sensing with Computer Vision and Lidar proposes combining computer-vision image classification with lidar technology to create a robust and reliable depth sensor.

4 DESIGN

The Logitech Brio webcam provides a high-resolution, two-dimensional image but lacks depth perception. The Leddar M16 provides accurate depth measurement in a horizontal dimension but lacks vertical perspective beyond a 40-degree spread. This project proposes bridging the utility of both devices by securing them in stationary positions, then using software to combine their outputs.

Figure 1 illustrates different dimensions measured by the M16 Lidar and Brio Webcam. The red cube represents the Logitech Brio webcam and M16 Lidar secured in stationary positions. The flat purple triangle represents the M16 Lidar's horizontal range detection. The transparent green rectangle in front of the person represents the computer vision model recognizing that there is a person in-front of the sensor. The transparent teal pyramid represents the Brio webcam's field-of-view.

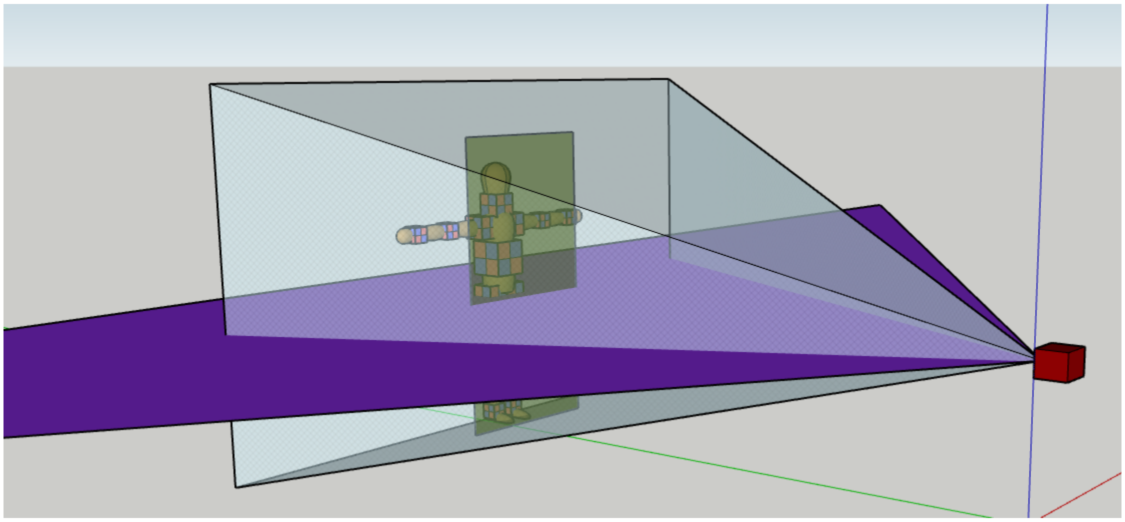

Fig. 1. Visualizing different dimensions measured by the M16 Lidar and Brio Webcam.

5 Device Mount

Using some spare plywood, I created a mount for the Logitech Brio webcam and Leddar M16 as shown in Figure 2. This mount serves to stabilize the webcam and lidar devices in a stationary positions relative to each other so that accurate distance/visual calibrations can be performed. If the webcam and lidar devices are not placed in consistent positions, distance information will not be synchronized with object recognition. Writing an algorithm to compensate for automatic distance and object recognition calibration is an overly complex task and beyond the scope of this project.

Fig. 2. Part of the mount for .

Progress with the Leddar M16 is slow and I do not foresee it to be working and integrated in time for expo. Fortunately, I was able to read distance information with the RPlidar A1. I adjusted the mount to accommodate the RPlidar A1 by drilling a few shallow holes in the base of the mount. This allows the RPlidar A1's four plastic standoffs to fit into the base of the mount as shown in Figure 3.

Fig. 3. Visualizing different dimensions measured by the M16 Lidar and Brio Webcam.

The device mounting system is complete. Our project's components can be fixed in stable positions to ensure consistent readings in different environments. This mounting system is important to our final design because it ensures operational consistency and simplifies the overall problem.

6 Computer Vision

During Winter Term, I started experimenting with OpenCV's pre-trained facial/pedestrian support-vector-machine (SVM) classifier. This SVM is a combination of several other SVMs that detect the upper body, eyes, mouths, and noses. The combined SVM is intended to detect faces with high accuracy. However, when applied to our design, I could not consistently replicate good results. This was due to several factors, namely the SVM used was meant to perform classification on still images where the camera's perspective is far from the subject.

Our design specifications envision a system that quickly tracks multiple subjects in a crowded expo scenario. In a such a scenario, human subjects will be unpredictably shifting their positions and moving in or out of the field-of-view. As seen in Figure 4, the OpenCV SVM model does not perform to our specification. If the human subject were to turn their head or move too quickly, the SVM will have difficulty tracking their body. Additionally, the SVM performs intensive calculations on the computer's CPU, severely limiting the video output's frame-rate and resolution.

Fig. 4. SVM face classification (Left) fails when subject slightly turns their head (Right)

Tensorflow's open source object detection classifier presented a better computer-vision alternative. [4] The Tensorflow object recognition library is better suited for this project because its library has already been trained to recognize a large dataset of objects. [5] These pre-trained datasets in Tensorflow's library are sourced from other machine learning datasets and employ advanced algorithms such as Single Shot Multibox Detection and Region-Based Fully Convolutional Networks. [6] [7] [8]

Fig. 5. Our pre-trained Tensorflow model can detect multiple subjects in near real-time

Using this pre-trained Tensorflow model, our project can now accurately outline and label over 90 subjects as they come into view of the webcam. The current state of the code will enable us to selectively edit the output video frames to draw bounding boxes on subjects as they move in and out of the camera's field of view.

Tensorflow also enables us to take advantage of NVIDIA CUDA, a driver that moves intensive calculations to the GPU. While this increases our list of material requisitions for our physical expo demo, moving calculations to the GPU greatly improves the output video quality, frame rate, resolution, and classification speed. [9]

The computer vision aspect of our project is now complete. Combined with a stable mount, we now have a versatile system that can recognize over 80 distinct models such as humans, bags, or animals in near real-time.

7 Current Problems

7.1 RPlidar A1 and Computer Vision Integration

To ensure our project will be in a presentable state by expo, I am modifying my design to use the RPlidar A1. Progress with the M16 lidar device has been slow, I do not foresee it being and integrated in time for expo. I have successfully tested the RPlidar A1 and have achieved reading some point-distance angle information. However, the python library used by the device calls a generator function to store its distance data. This type of data structure is hard to adapt for our purposes because the device's buffer will overfill and overwrite readings if it is not called fast enough. Figure 6 demonstrates this issue. To remedy this, I will experiment adjusting the loop calling the Tensorflow image recognition functions. I will also experiment running the RPlidar A1 in parallel with the Tensorflow image recognition.

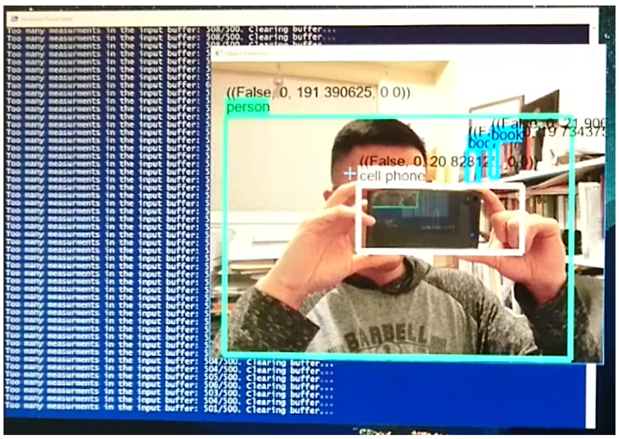

Fig. 6. Due to the overfilling buffer my current implementation does not calculate distances correctly.