Progress Review 1

Individual lab report – 02 | October 23, 2015

TEAM DAEDALUS

Submitted by:

Mohak Bhardwaj

Team Members:
Dorothy Kirlew
Richa Varma
Shivam Gautam

Pranav Maheshwari

1. Introduction

The work done by Team Daedalus for this progress review is spread across different subsystems and is in keeping with the goals specified by the team beforehand. The major tasks completed involve:

- Initial testing and prototyping platform has been acquired
- Literature Review and selection of communication architecture and hardware done.
- Literature Review of Vision Algorithms for obstacle avoidance and selection of hardware done.
- Interfacing of actuator control board with platform and initial testing of actuators done.

2. Individual Progress

The responsibilities taken up by me involved the following:

- Literature Survey on vision based obstacle detection algorithms to find the most appropriate algorithms for our purposes.
- Depending on algorithm, selecting the best suited camera for our application.
- Working on the android app.
    - GUI Development
    - Bluetooth Interface
- Primary Point of contact with sponsor

VISION BASED OBSTACLE DETECTION

As per the team's goals, I worked along with Pranav on selecting the appropriate algorithm and hardware for vision based obstacle detection. We initially decided to do our own literature survey independently and then exchange notes to narrow down the possible algorithms. We kept the requirements and time constraints of the project in mind while doing the survey.

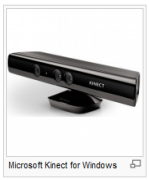
We needed an algorithm that was robust yet easy to apply because the main focus of our project is on the collaborative aspect I started by searching for general vision based obstacle avoidance algorithms ranging from straightforward optical flow to stereo and structure from motion. These algorithms although generic and robust, involve a lot of complexities and caveats in their implementation. The papers referenced for these methods are cited in the references section at the end.

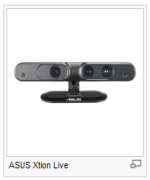
During the survey I came across a few interesting methods involving point clouds formed from depth estimation. This made me realize that the MS Kinect was easily available to us and could be used to similar effect. There are many papers on Kinect based navigation an obstacle avoidance. After going through a few of these papers, and exchanging notes with Pranav we decided that Kinect was the right way to go as it met our requirements, had a larger community of support compared to other RGB-D cameras by Asus or PrimeSense3D and was easily available in the inventory of testing. Here is a comparative study between Kinect and other RGBD cameras by ipisoft which shows that support and availability are the only major factors affecting the choice of cameras:

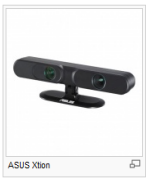
Depth Sensors Comparison

Starting from version 1.8.0.81, iPi Recorder supports ASUS Xtion and ASUS Xtion Live devices (and also PRO modifications for developers). They are similar to Microsoft Kinect sensor and can be used in the same way. But there are certain differences that you can take into account before making decision on which device to choose for use with iPi Motion Capture.

Starting from version 2.1.4.21, iPi Recorder officially supports PrimeSense Carmine 1.08 device, which is essentially an equivalent of ASUS Xtion Live under different brand name.


Microsoft Kinect for Windows


ASUS Xtion Live


ASUS Xtion


PrimeSense Carmine 1.08

Kinect vs Xtion / Carmine

Both Microsoft Kinect and ASUS Xtion (Live) / PrimeSense Carmine sensors are based on the same PrimeSense infra-red technology. So all basic characteristics critical for full-body motion capture are generally the same. But there are certain differences that you can take into account:

Device	Pros	Cons
Microsoft Kinect	▪ High quality of device drivers ▪ Stable work with various hardware models ▪ Has motor that can be controlled remotely by iPi Recorder application: this makes device positioning more convenient	▪ Bigger size (12" x 3" x 2.5" against 7" x 2" x 1.5") ▪ Higher weight (3.0 lb agains 0.5 lb) ▪ Require ACDC power supply ▪ Higher interference with another Kinect sensor in "Dual depth sensor" configuration ▪ Lower RGB image quality in comparison with MS Kinect
ASUS Xtion / PrimeSense Carmine	▪ More compact (7" x 2" x 1.5" against 12" x 3" x 2.5") ▪ Lighter weight (0.5 lb agains 3.0 lb) ▪ Does not require power supply except USB ▪ Lower interference with another ASUS Xtion / PrimeSense Carmine sensor in "Dual depth sensor" configuration ▪ Better RGB image quality	▪ Less popular device ▪ Lower drivers quality ▪ Does not work with some USB controllers (especially USB 3.0) ▪ No motor, allow only manual positioning

Now, the only task left was to narrow down on the exact algorithm we could apply. The following paper was finally narrowed down to:

[1] MS Thesis of Rasoul Mojtahedazdeh on Robot Obstacle Avoidance using MS Kinect –KTH Sweden

This thesis starts by describing in detail the technology and specification of the MS Kinect, its powers and limitations. A method for generating point clouds from the depth image data provided by the Kinect is then described. Only points with height between a minimum height and maximum height of the robot are selected from the point cloud and the rest of the points are discarded. These points are then projected onto the 2D space and using a clustering algorithm, obstacles are divided into polygons to form convex hulls. This information, after removal of outliers is used for obstacle detection and later for environment modelling using a 2D occupancy grid based approach.

We intend to pursue this method after acquiring a MS Kinect from the inventory.

ANDROID APP

I have had prior experience in working with Android Studio, so along with Dororthy, I took charge of the android app. We split the work with her putting more focus on the GUI and me on the Bluetooth interface.

For the Bluetooth interface, I first read up about Android's Bluetooth API but since it would take a longer time to implement, I decided to first start by implementing serial Bluetooth communications between my laptop which runs Ubuntu 14.04 and my Android smartphone. For this there were two options available

1. Using MIT AppInventor2 –an easy to use drag and drop interface for making apps.
2. Using a Bluetooth terminal app on android which could communicate with a serial port

I chose to go with the second option to develop the Ubuntu side of things first so that it would help us in testing while the app is being worked upon. I had to search the appropriate bash commands in Ubuntu and learn the fundamentals of bash scripting since I had no experience with it before. After spending some time in learning, I implemented a simple script in bash to start a serial service on

channel 22 and listen for Bluetooth devices sending data serially on it. The code for the bash script is shown below:

```
#!/bin/bash

sdptool add --channel=22 SP
xterm -e cat /dev/rfcomm0
gksudo rfcomm watch /dev/rfcomm0 22

trap ctrl_c INT

function ctrl_c(){
echo "Cleaning up..."
rfcomm release /dev/rfcomm0

sdptool del 0x10008

}
for i in `seq 1 5`; do
    sleep 1
    echo -n "."
done
```

After pairing the phone and laptop on Bluetooth, park and un-park commands can now be sent serially to the laptop. After appropriate error handling and reliability has been added to the bash program it can be easily integrated with all SBC's running Ubuntu or Linux. I also constantly kept checking on the work Dorothy was doing on the GUI and helping and advising her as much as I could because she had not worked with Android Studio before.

Fig: Screenshot of App GUI taken from Nexus 5

PRIMARY POINT OF CONTACT WITH SPONSOR

As my team's primary point of contact with the sponsor, I had to make sure that all communication with the sponsor go smoothly and there are no delays. Also, we are currently going through negotiations with the sponsor n ROS based platforms and arbitrating on that required me to keep the whole team in the loop at all times.

3. Challenges

The challenges that I had to face were:

- Deciding the vision algorithm and hardware to use was very difficult as we had to keep our requirements and constraints in mind while at the same time remember that the vision system is a critical system that as to perform perfectly.
- Working with bash scripting and Bluetooth serial connectivity were a challenge in the beginning as I had no bash experience and the language and commands are not intuitive or easy to master. Setting up a reliable Bluetooth connection is also difficult as it requires different commands and services to be understood and called.
- Keeping the whole team in the loop for all interactions with the sponsor and making decisions together.

4. Teamwork

The other members of my team were working on the following subsystems:

- Pranav Maheshwari: Worked with me on the vision based obstacle detection system literature survey and hardware.
- Richa Varma: She worked on the integration of the actuator control board with platform.
- Shivam Gautam: Worked on literature survey and coming up with hardware for communication system
- Dorothy Kirlew: Worked on GUI part of Android app along with me.

5. Plans

For the next progress the team intends to work on the following:

- Pranav and Mohak: Will work on acquiring Kinect and with ROS to start testing and calibration.
- Shivam: Will continue work on mesh network after ordering devices and testing.
- Dorothy and Mohak: Will continue work on the android app and Bluetooth functionality.
- Richa: Will work with mounting and interfacing sensors with the mobile platform.