EPICS SMART CITY

SPRING 2018

EPICS Smart City - Spring 2018

	Name	Major/Year
Project Manager	Erika Lin	Electrical Engineering / 3rd
Hardware Design Lead	Romita Biswas	Electrical Engineering / 2nd
Data Analysis Design Lead	Kalpan Jasani	Computer Science / 2nd
Website and App Development Design Lead	Kartik Mittal	Computer Science / 1st

What is a Smart City?

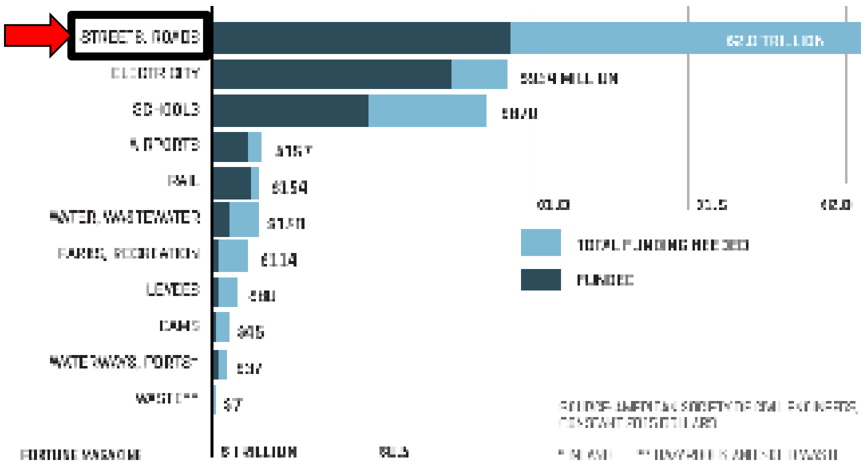
- **The ultimate goal: Implement technology and the internet of things to improve quality of life.**

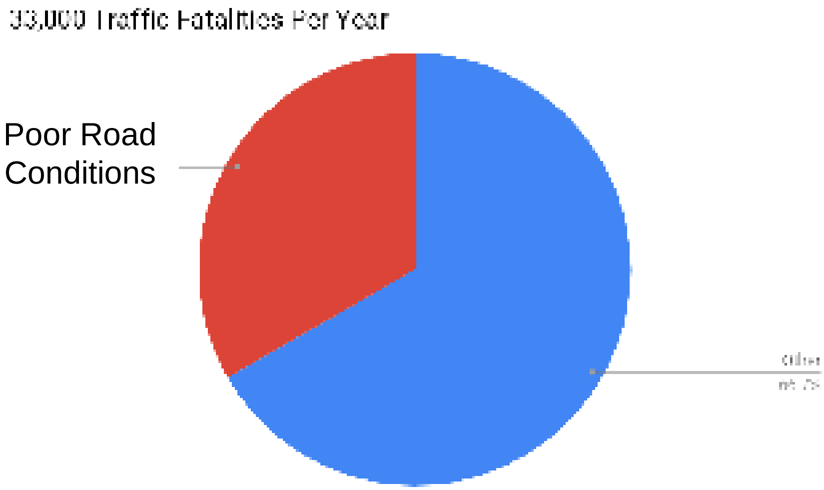
placemeter

PARKWHIZ

Why Potholes?

U.S. INFRASTRUCTURE NEEDS OVER THE NEXT 10 YEARS





https://www.pothole.info/the-facts/

multi-million dollar lawsuits

Existing Pothole Detection Systems

https://patents.google.com/patent/US9626763B1/en

Google Patented Pothole Detection System

http://www.arkansashighways.com/System_Info_and_Research/pavement_management/pavement_management.aspx

ARAN Vehicle
$1.5 Mil Military-Grade

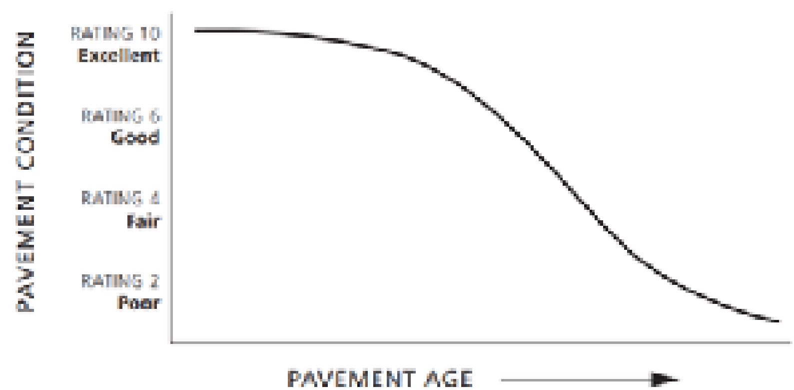
http://www.nydailynews.com/autos/street-smarts/ford-pothole-mitigation-article-1.2874552

Ford Fusion V6 Pothole Detection System

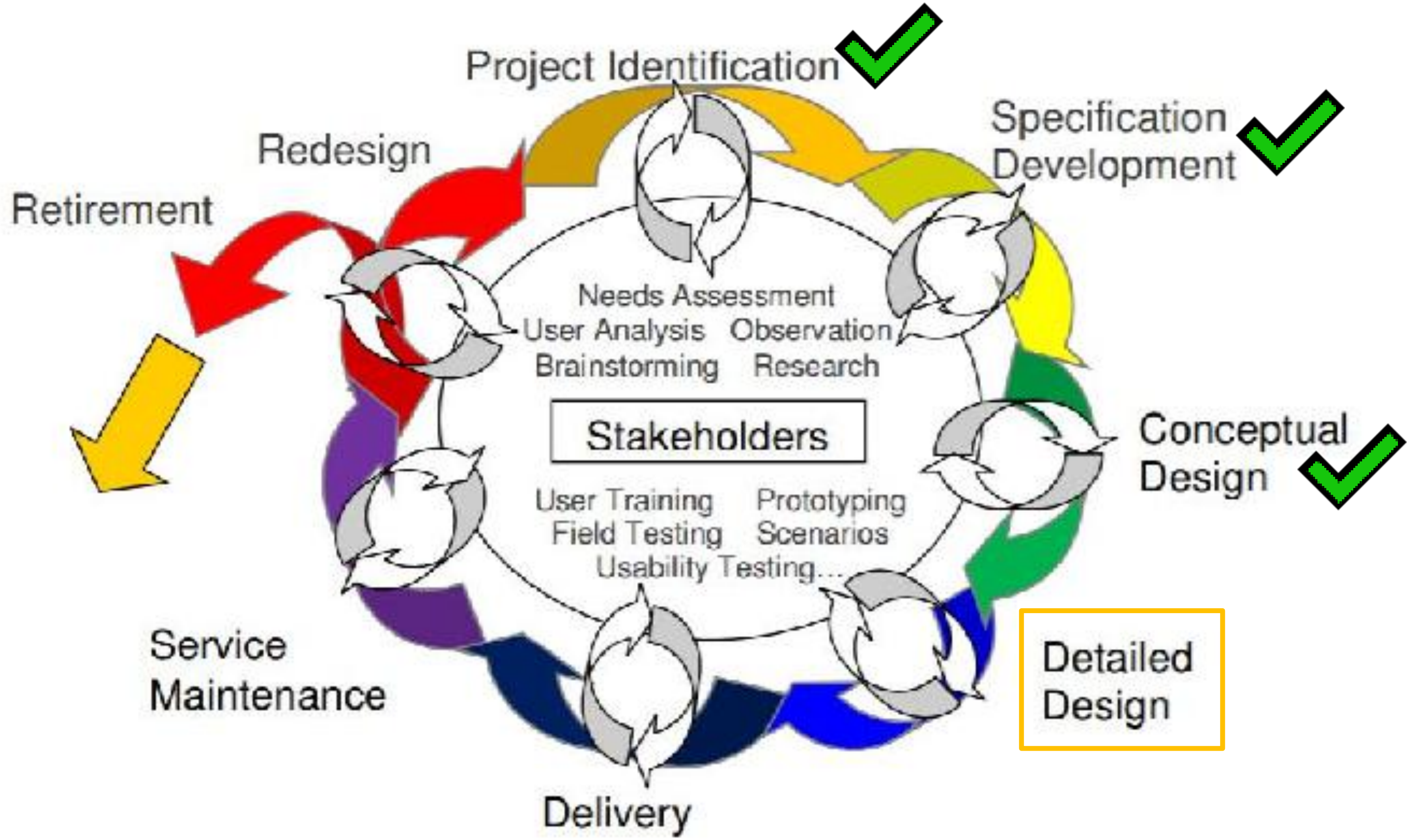
Project Background

CITY OF
WEST LAFAYETTE

PASER Manual



Smart City Design Approach



Target Needs

Automated System

Compact and Affordable

Reliable Analysis

EPICS SMART CITY

Hardware Team

Meet the Hardware Team

Name	Major/Year
Romita Biswas	Electrical Engineering / 2nd
Benjamin Hutchins	Mechanical Engineering / 2nd
Brian Sutanto	First Year Engineering / 1st
Kyaw San (Steven) Thway	Civil Engineering / 2nd

Needs and Specifications

Needs	Specifications
• Collect depth data for potholes • Locate potholes • Continuous data stream	• Continuous collection of depth data • Continuous location data • Synchronized data • Collection fps greater than vehicle speed

Currently Existing Hardware

Current Hardware - Automatic Road Analyzer

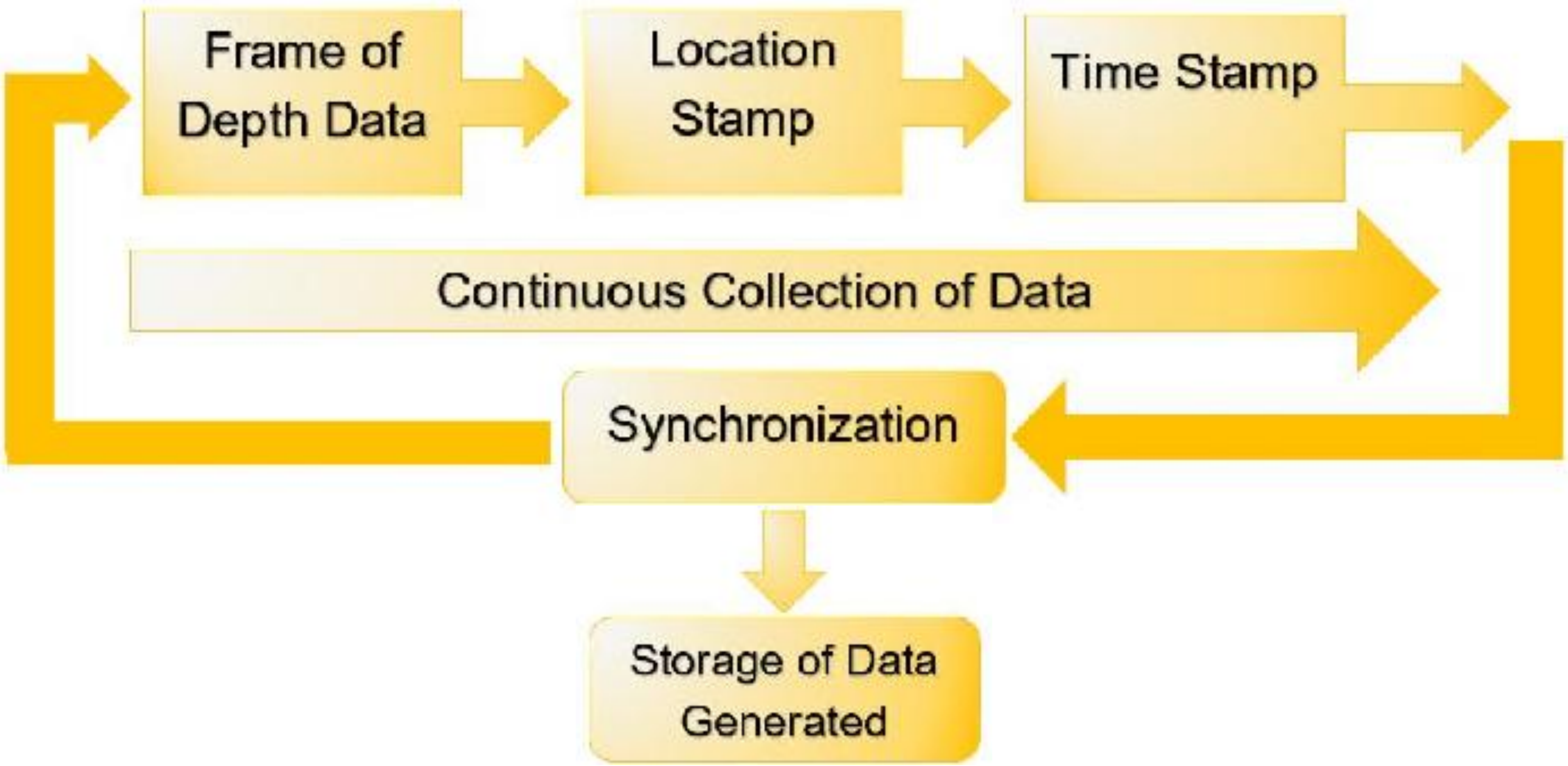
Benefits	Drawbacks
High precision survey and software systems	$130,000-$150,000 per pavement laser sensor
Finds cracks, potholes, drainage issues	$1.3 million per ARAN Truck
Hardware: GPS, distance measuring instrument for depth, texture data	Used once every 1-2 years

Hardware Project Motivation

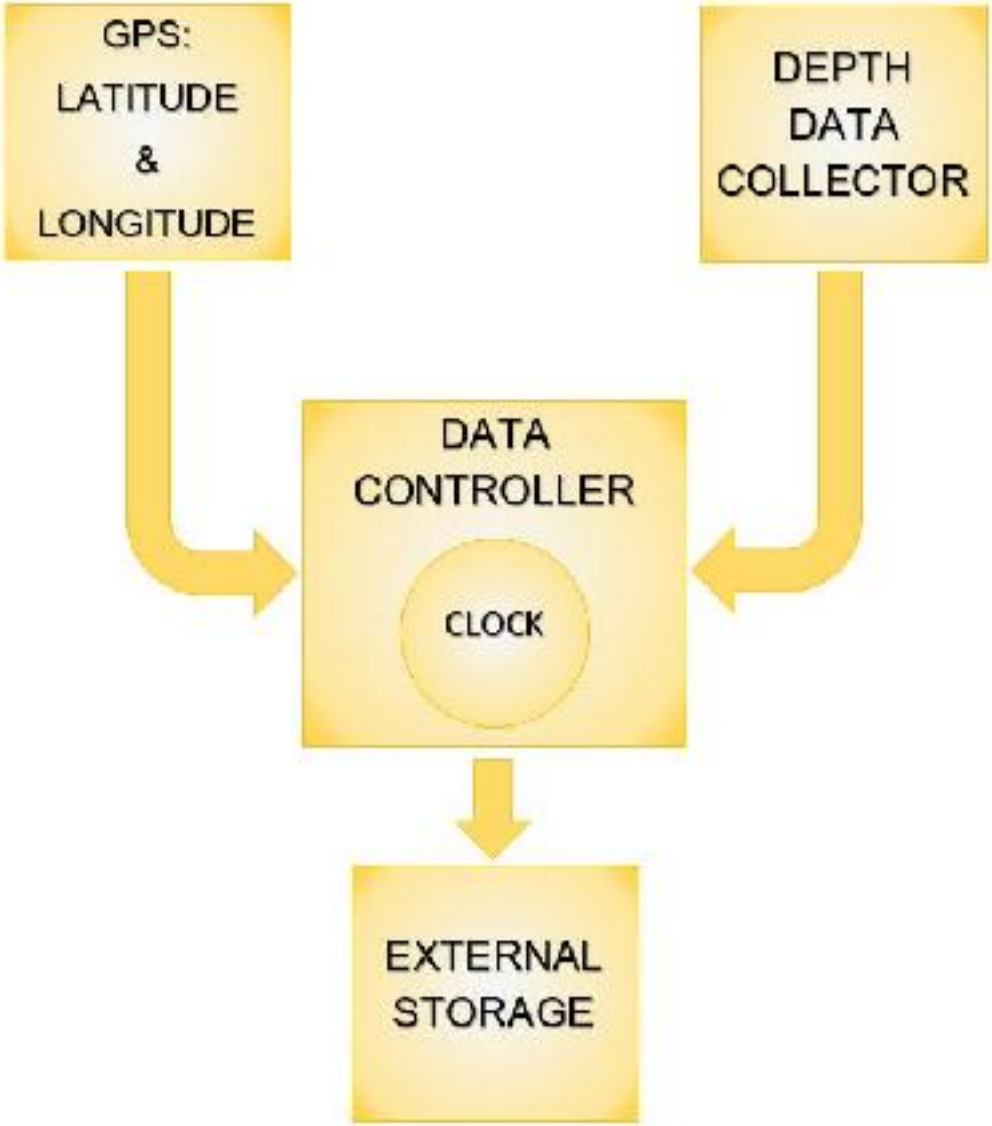
Complementary System:

- Affordable for all cities
- Regular surveying
- Quantitative and qualitative data
- Mountable on multiple vehicles
- Location and time of pothole detection

Hardware Background



System Overview



Depth Data

- Data taken from a sensor like a point cloud
- Measures distance from sensor to object of interest

Depth Data Collector

Kinect for Xbox

Kinect for Xbox

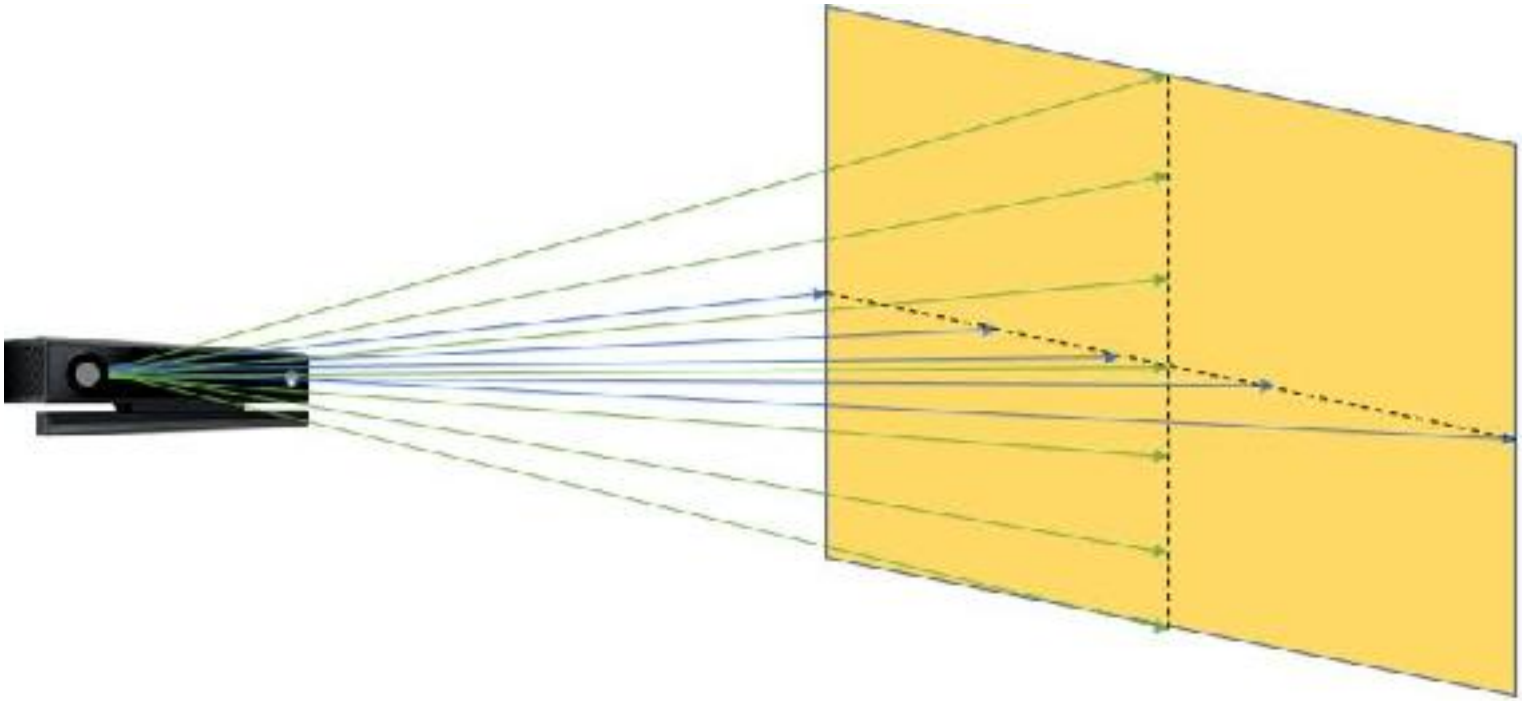
- Commercially produced for gaming industry
- Wide angle time of flight sensor
    - Measures a scene of distances from a light pulse’s time
- Can track at night with IR sensor
- Detects objects up to 4.6 feet
- High resolution images (1080p)

Kinect for Xbox

Benefits	**Drawbacks**
$84.90	UV ray sensitivity
IR sensor: At night	Resolution up to mm
Time of flight sensor: measures distances from sensor	Small cracks can't be detected
Automatically generates a matrix	Distorted outer frame
60 frames/sec	

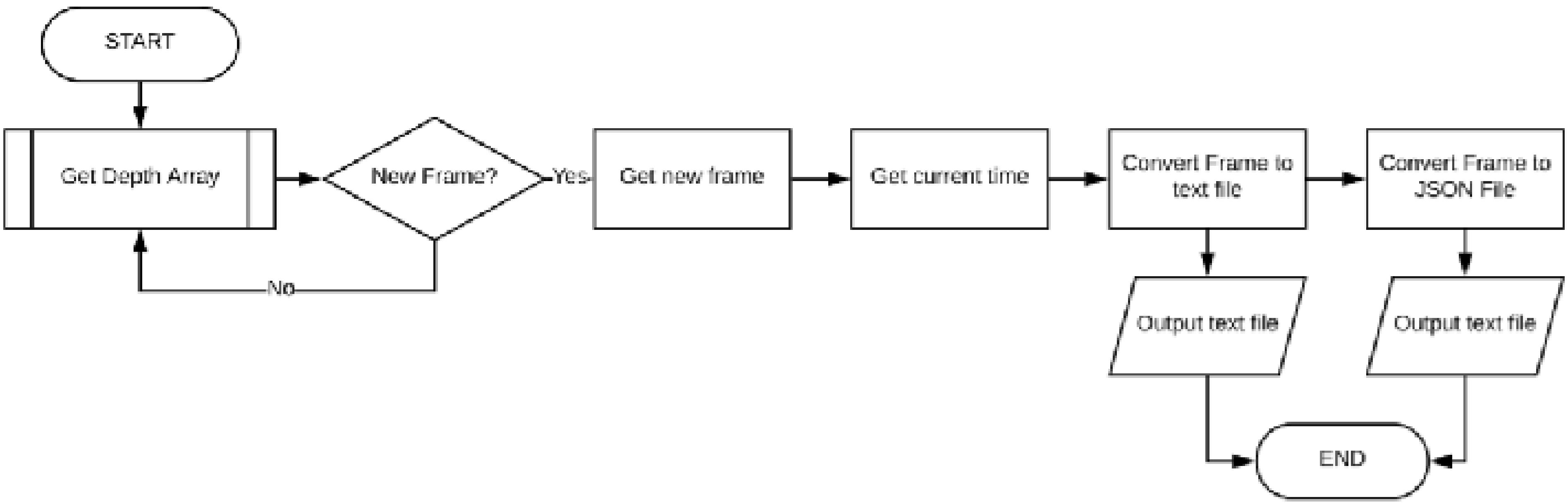
Wide Angle Time of Flight Sensor

Kinect for Xbox: Matrix Field of View



Kinect Code High Level Overview

Kinect Flowchart



FPS Calculation Result

Speed (mph)	FPS
10	5.081
20	10.161
30	15.242
40	20.323
50	25.403
60	30.484

Adafruit GPS

Global Positioning System (GPS)

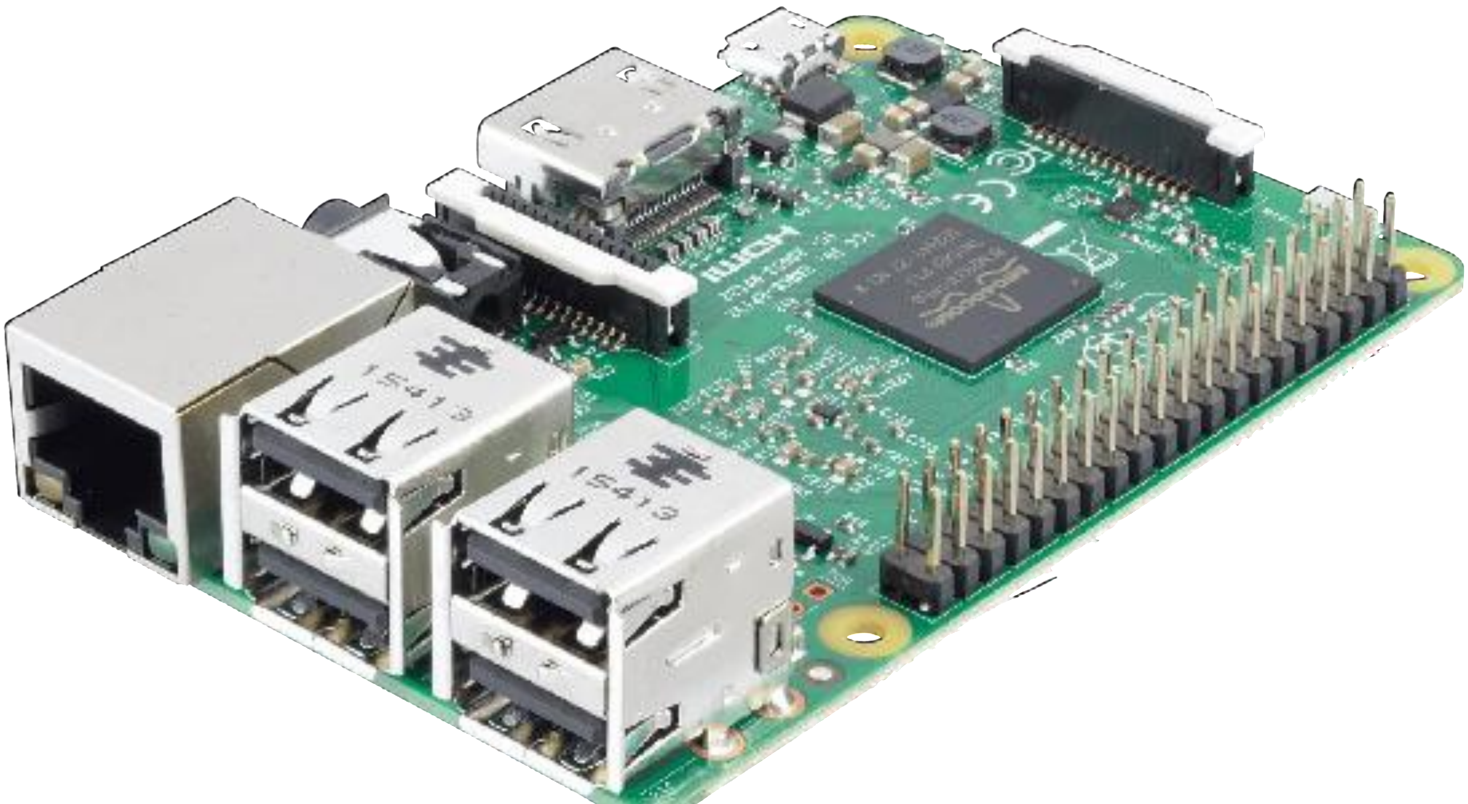
- Utilized the satellite-based navigation system
- Location data: latitude & longitude and the corresponding time collected at a max rate of 10 times per second
- Tracker and a receiver included

Collecting GPS Data

- Adafruit GPS built for Raspberry Pi
- Raspberry Pi will be connected to parent microcontroller
- Microcontroller will run GPS through Raspberry Pi via server
- Execution and collection of data will take place at microcontroller

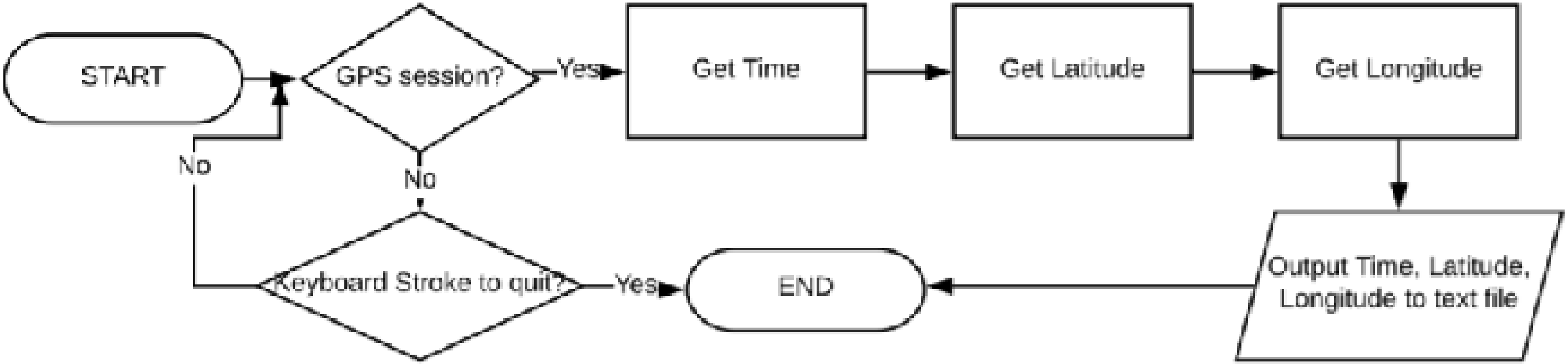
Child Microcontroller

Raspberry Pi 3



GPS CODE HIGH LEVEL OVERVIEW

GPS Flowchart



Micro-Controller

Intel® NUC Kit NUC7i7BNH

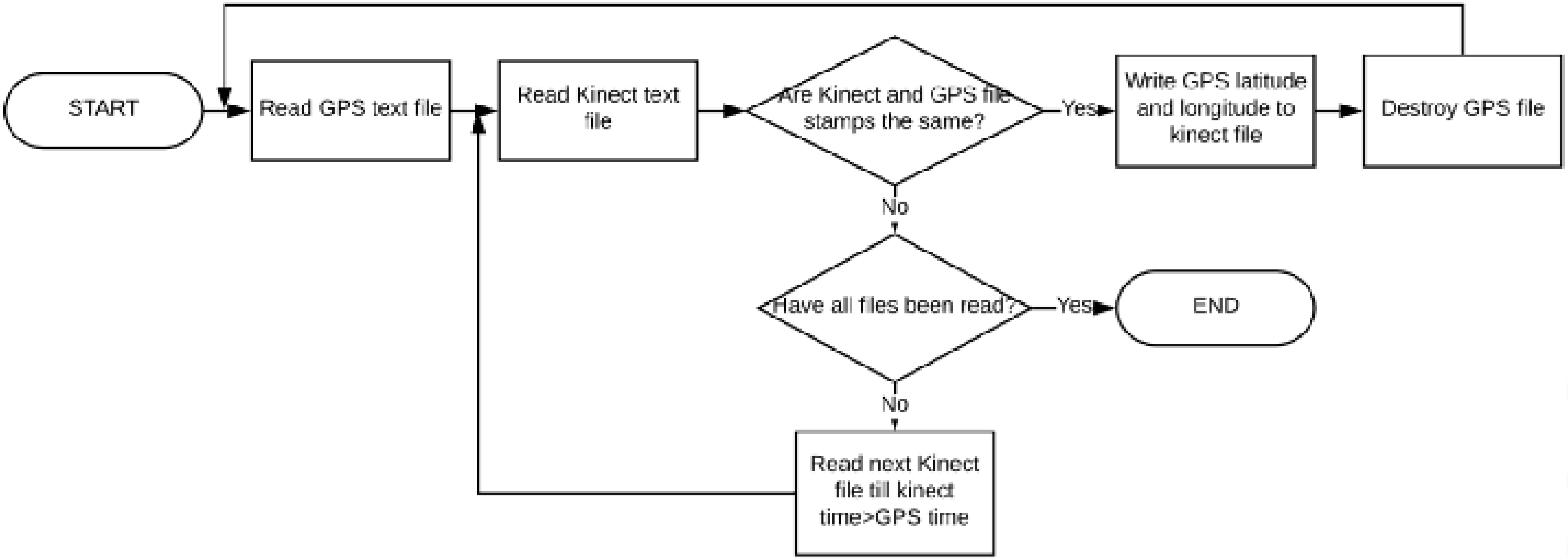
Micro-Controller

Intel NUC Micro-Controller

- Runs Kinect and GPS
- Processes data generated
- Stamps GPS location on each matrix generated
- Stamps time on each matrix
- 512 GB SSD memory installed
- 16GB DDR4 RAM installed

SYNCHRONIZATION HIGH LEVEL OVERVIEW

Synchronization Flowchart



Attachment & Demonstration

Current Iteration of Prototype

- Project Partner Meeting: can be attached to waste management trucks to run at night
  - UV ray sensitivity
- Waste management trucks take all routes through city
- All roads can be checked weekly

Various Garbage Truck Types

Various Garbage Truck Types

- 3 active garbage trucks
- 2 active recycling trucks
- Different types of trucks -> Multiple solutions
- Initial Goal: concentrate on one truck specification

Kinect Attachment Location

Attachment Area

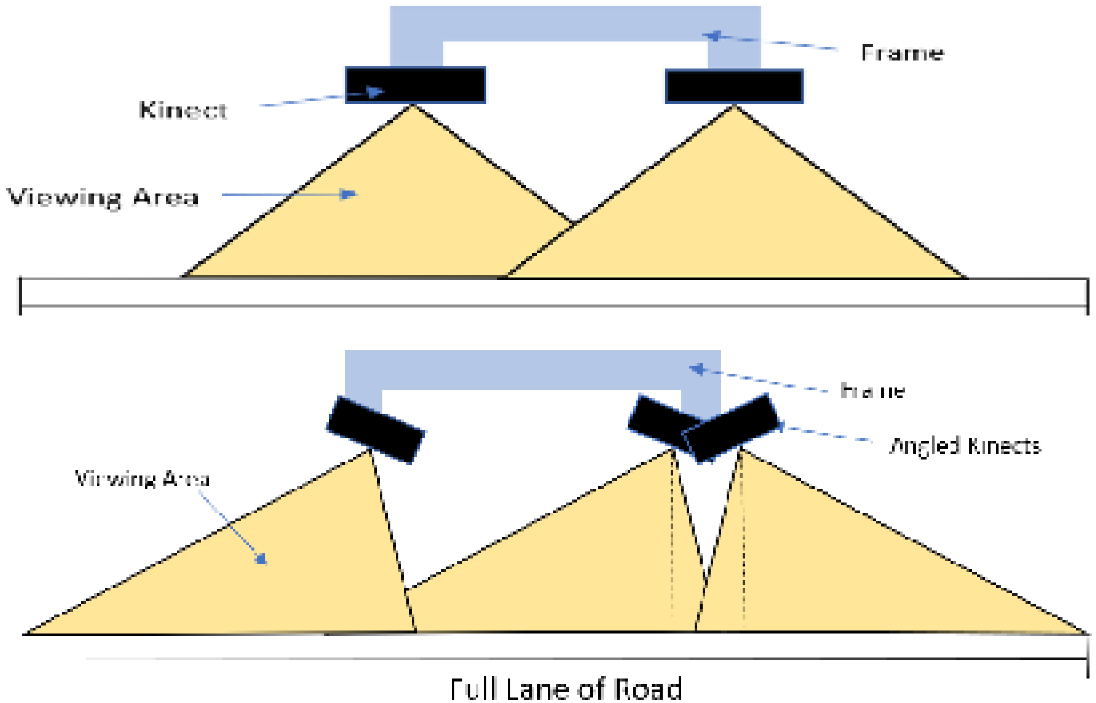
Current Iteration

Prototype Placement

Future Possible Iteration

Kinect Angle Visual

Kinect Angle Calculations



Calculation Results: Three kinects attached to the truck's frame with an angle of around 20 degrees from the horizontal is sufficient to cover a full lane of road (10 feet width).

Weather and Safety Precaution

Weather and Safety Consideration

Issue:
- Rain
- Snow
- Flying gravel

Issue:
- Kinect data loss
- Refraction

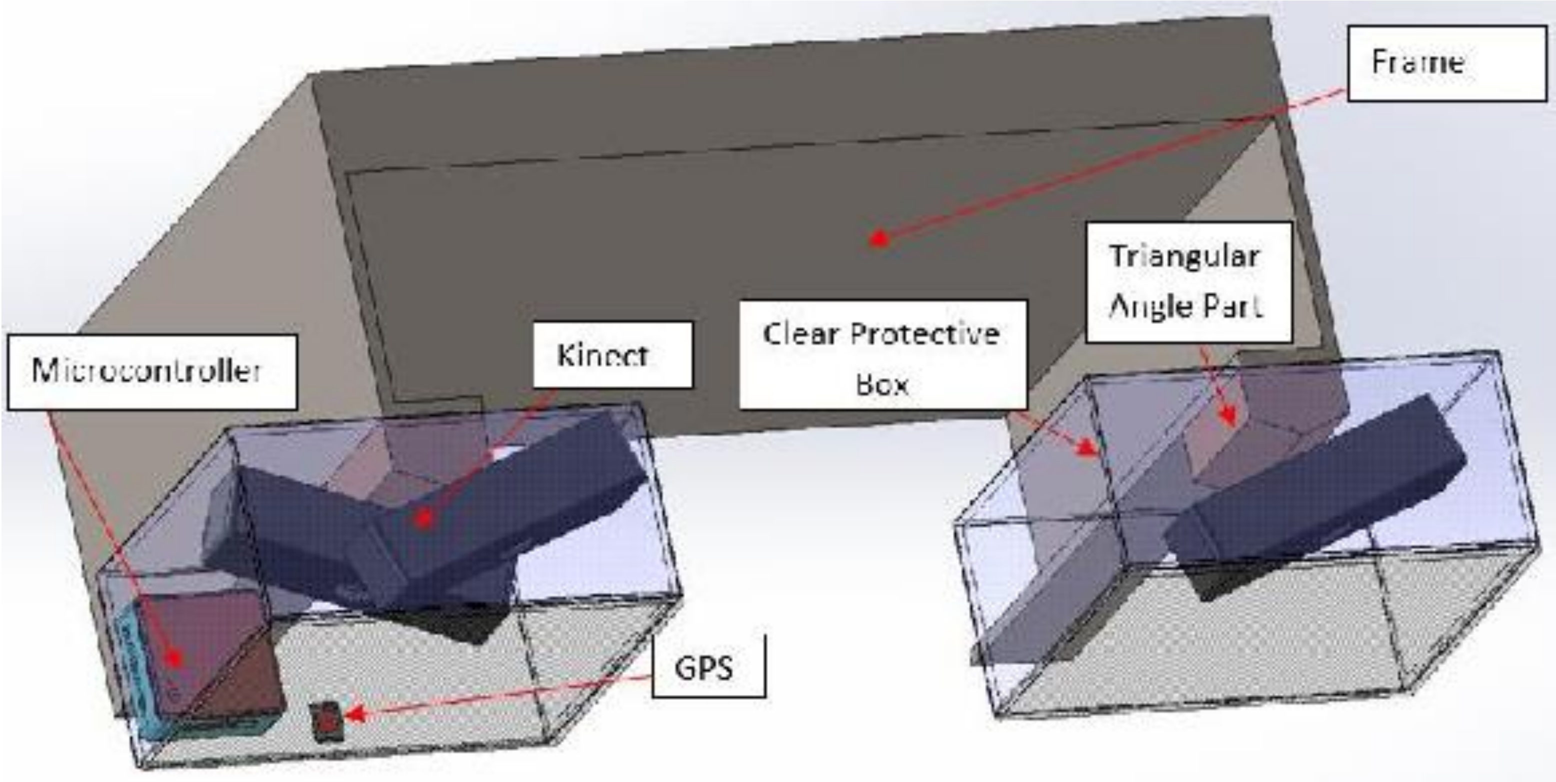
Issue:
- Case Safety

Solution:
- Clear case to house system

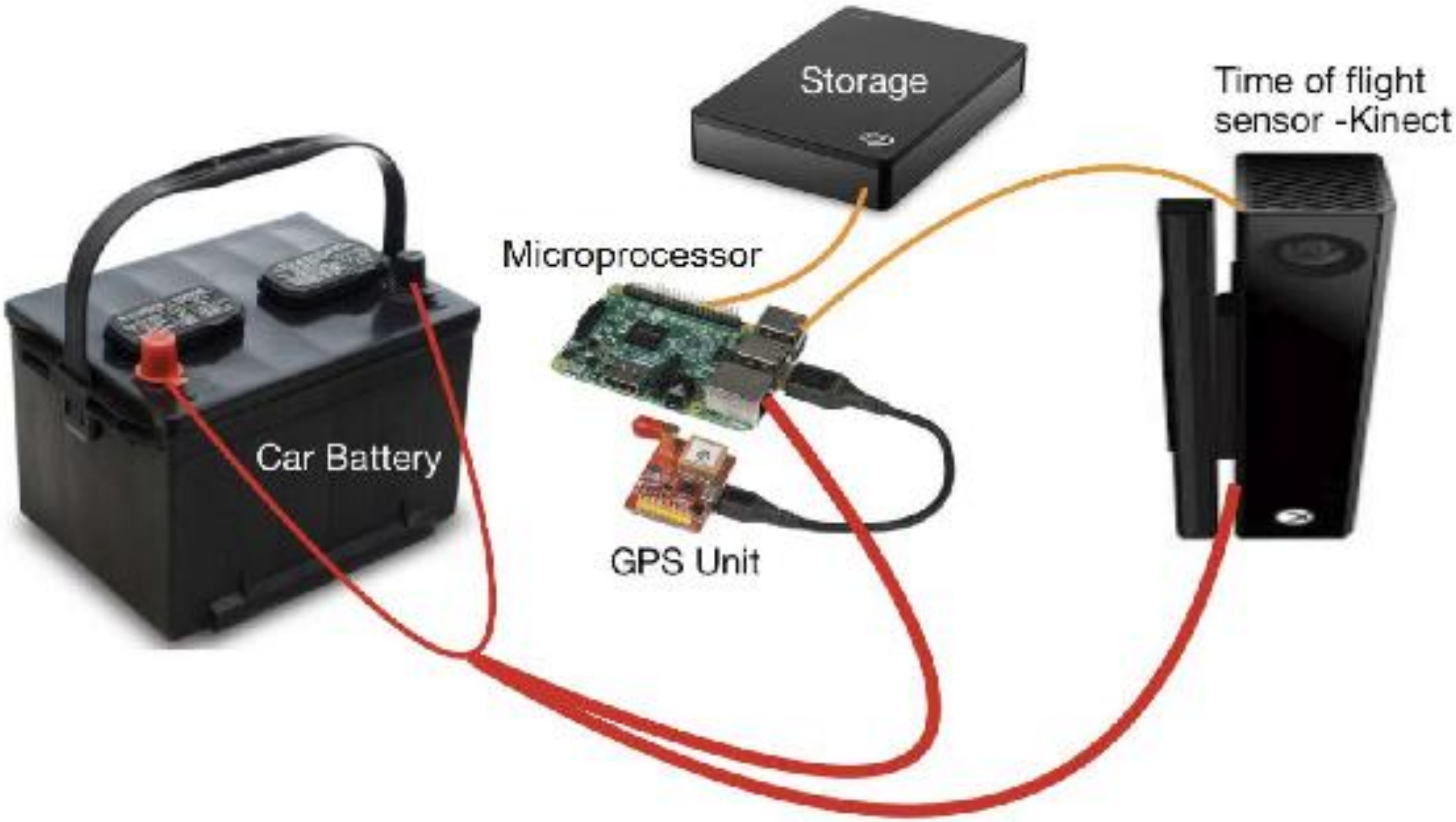
Solution:
- Safe and detachable
- Noise reducing clamps

Potential Design

CAD Model of Design



Progress: Detailed Design



Progress: Detailed Design

- Power source (car battery)
- Decision on depth sensor
- Installation of new microcontroller
- Able to collect continuous depth data using Kinect
- Consideration of safety and weather precautions

Issues

Learning

- Programming
- Python CV capabilities
- Multiple servers

Storage

- Data generated
  - 1406.5 MB/minute
- Python Speed matching desired fps

Tentatives

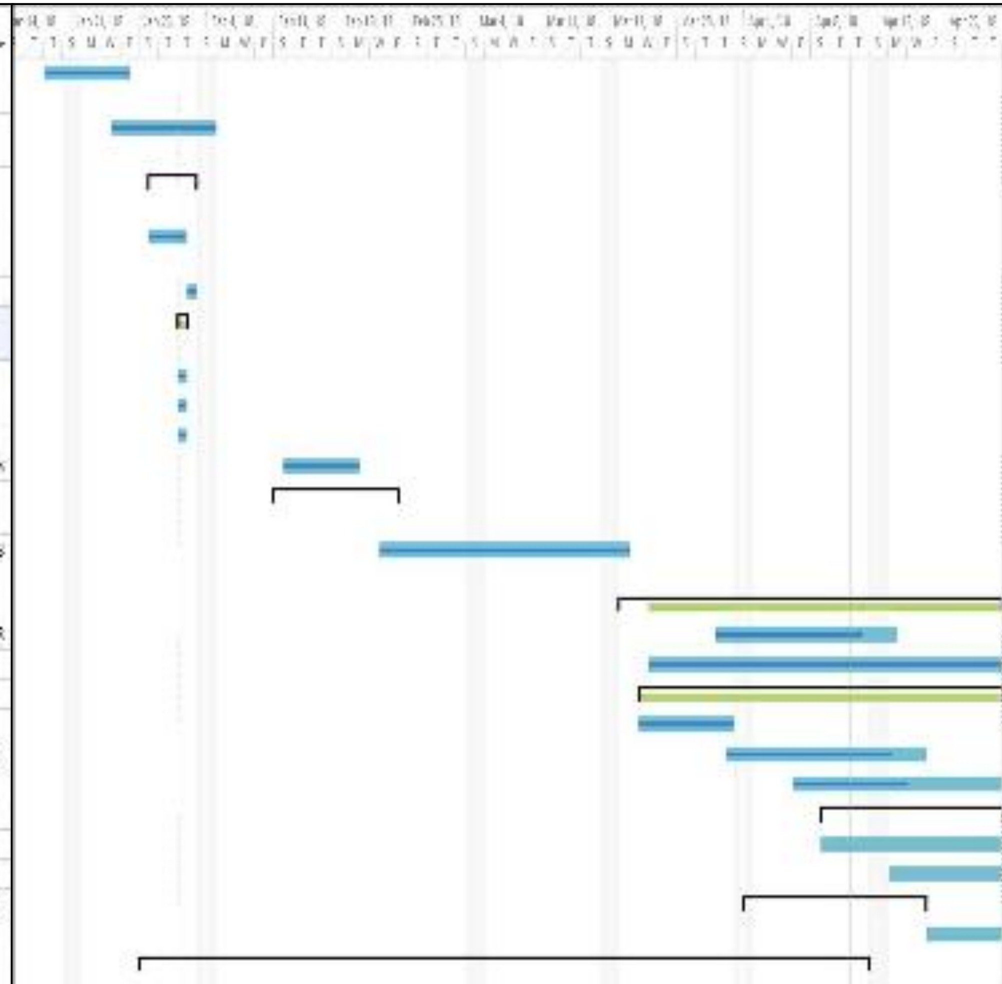
- Calibration to new road levels
- Calibration to different speeds
- Safety and protection

Budget for Spring 2018

Hardware Team Budget	
Item	Estimated Cost
Microcontroller	$494.93
SSD	$296.00
16GB RAM	$159.95
TOTAL	**$950.88**

Hardware Timeline

	Task Name	Duration	Start	Finish
✓	Brainstorming & Preliminary Actions	7 days	Thu 1/18/18	Fri 1/26/18
✓	Understand current code and functions	8 days	Thu 1/25/18	Sun 2/4/18
✓	**Project Partner Communication**	**5 days**	**Mon 1/29/18**	**Fri 2/2/18**
✓	Come up with questions for PP	4 days	Mon 1/29/18	Thu 2/1/18
✓	Meet with PP	1 day	Fri 2/2/18	Fri 2/2/18
✓	**Modify Design after PP Meeting**	**1 day**	**Thu 2/1/18**	**Thu 2/1/18**
✓	Mounting Frame Design	1 day	Thu 2/1/18	Thu 2/1/18
✓	Frame Overlap Design	1 day	Thu 2/1/18	Thu 2/1/18
✓	Garbage Truck Mounting Des	1 day	Thu 2/1/18	Thu 2/1/18
✓	Operate GPS Code	7 days	Mon 2/12/18	Mon 2/19/18
✓	**Mid Semester Design Review Presentation**	**11 days**	**Sun 2/11/18**	**Fri 2/23/18**
✓	Operate Kinect Code Successfully	18 days	Thu 2/22/18	Mon 3/19/18
	Modify Kinect Code	**30 days**	**Mon 3/19/18**	**Fri 4/27/18**
	Multiple frames per second	13 days	Thu 3/29/18	Mon 4/16/18
✓	Generate Continuous Data	27 days	Thu 3/22/18	Fri 4/27/18
	Prepare Microcontroller	**26 days**	**Wed 3/21/18**	**Fri 4/27/18**
✓	Complete Hardware Installat	8 days	Wed 3/21/18	Fri 3/30/18
	OS and Software Installation	13 days	Fri 3/30/18	Thu 4/19/18
	Operate Kinect Code	16 days	Fri 4/6/18	Fri 4/27/18
	Raspberry Pi to NUC Microcontr	**15 days**	**Mon 4/9/18**	**Fri 4/27/18**
	Output GPS data to Microcon	15 days	Mon 4/9/18	Fri 4/27/18
	Use Microcontroller time	10 days	Mon 4/16/18	Fri 4/27/18
	Final Review Presentation	**14 days**	**Sun 4/1/18**	**Thu 4/19/18**
	GPS and Depth Data Merge	6 days	Fri 4/20/18	Fri 4/27/18
✓	**Notebook Reviews**	**56 days**	**Sun 1/28/18**	**Fri 4/13/18**



| Jan | Feb | March | Apr

EPICS SMART CITY

Data Analysis Team

EPICS PURDUE

Meet the Data Analysis Team

Name	Major/Year
Kalpan Jasani	Computer Science / 2nd
Ethan Tan	First Year Engineering / 1st
Ayyub Jose	Electrical Engineering / 4th
Don Yerkozhanov	Chemical Engineering / 3rd

Needs and Specifications

Needs	Specifications
• Accept and suggest Hardware team's inputs • Detect potholes • Quantify severity of potholes • Send to server for Web and App team	• Plane fitting • Otsu's binarization • Rectangle bounding

Flow Chart for Processing

Data for a frame → Correct tilt in data (Plane fitting) → Isolate pothole (Otsu's Method) → Quantify severity (Rectangle bounding) → Send to server

Plane-Fitting

Random Sample Consensus (RANSAC)

- Compensate for depth data accuracy
- “Correcting the tilt”

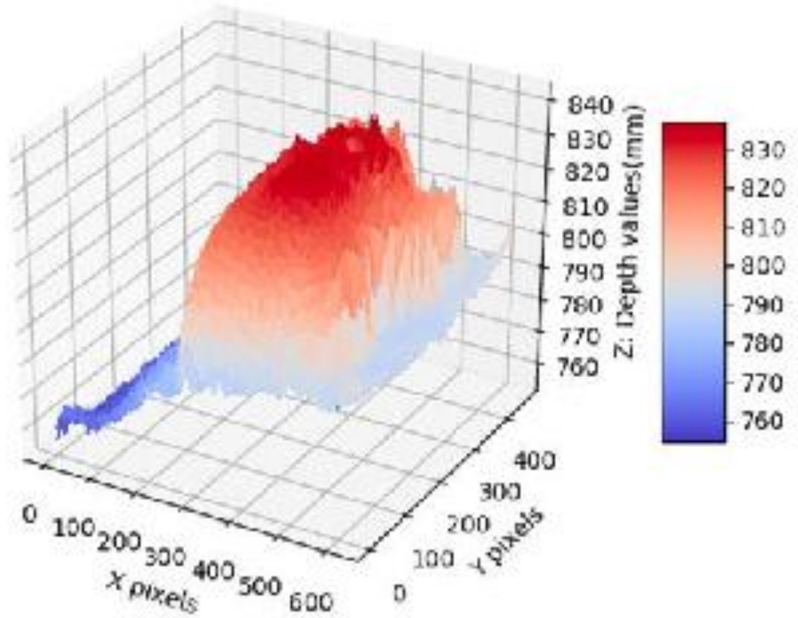


Figure 1

Plane Fitting Intuition

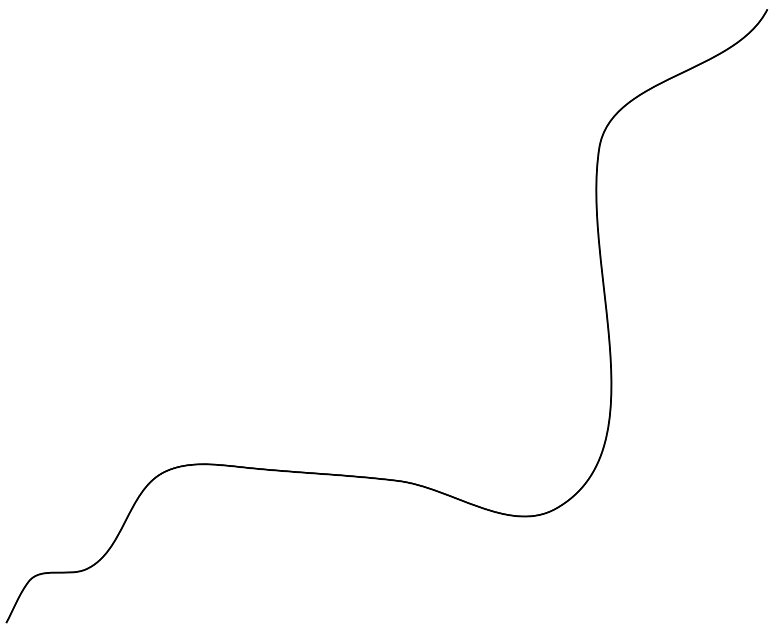


Fig: Before plane fitting

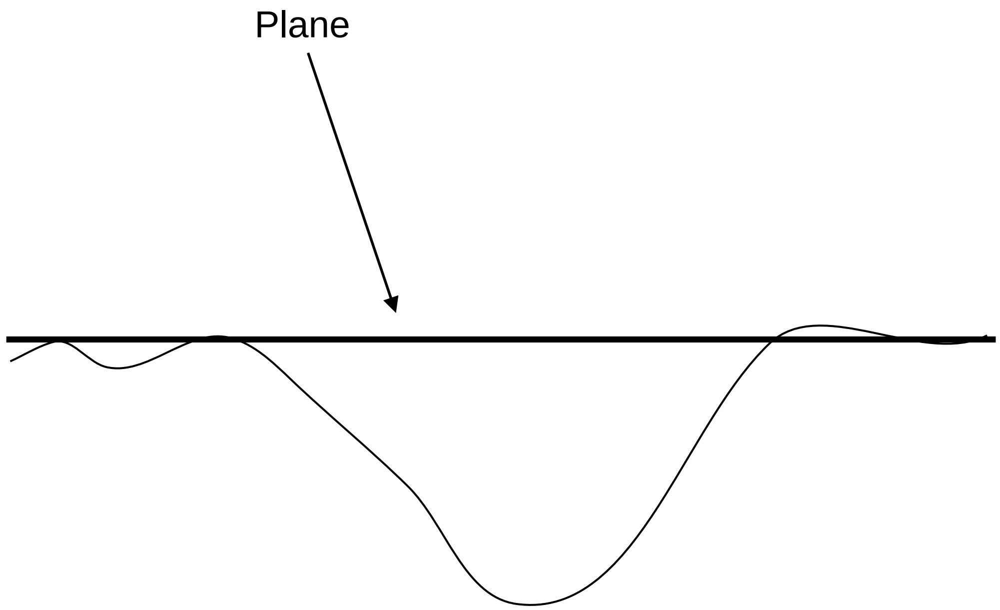
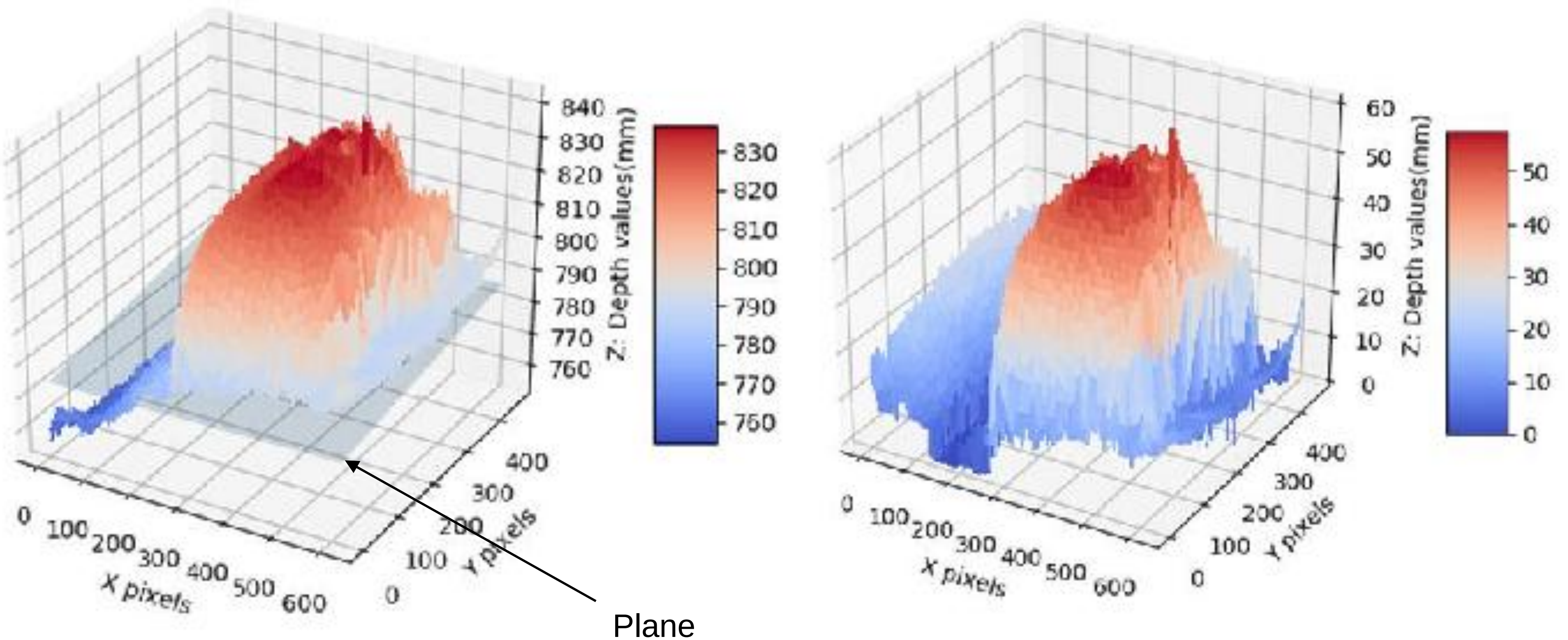


Fig: Plane fitted data

Plane Fitting

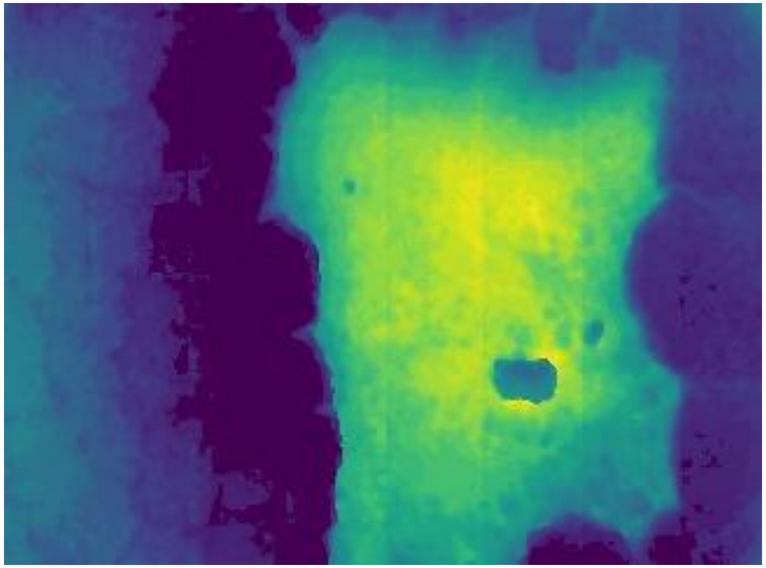
Plane-Fitted and Skewness Corrected Images



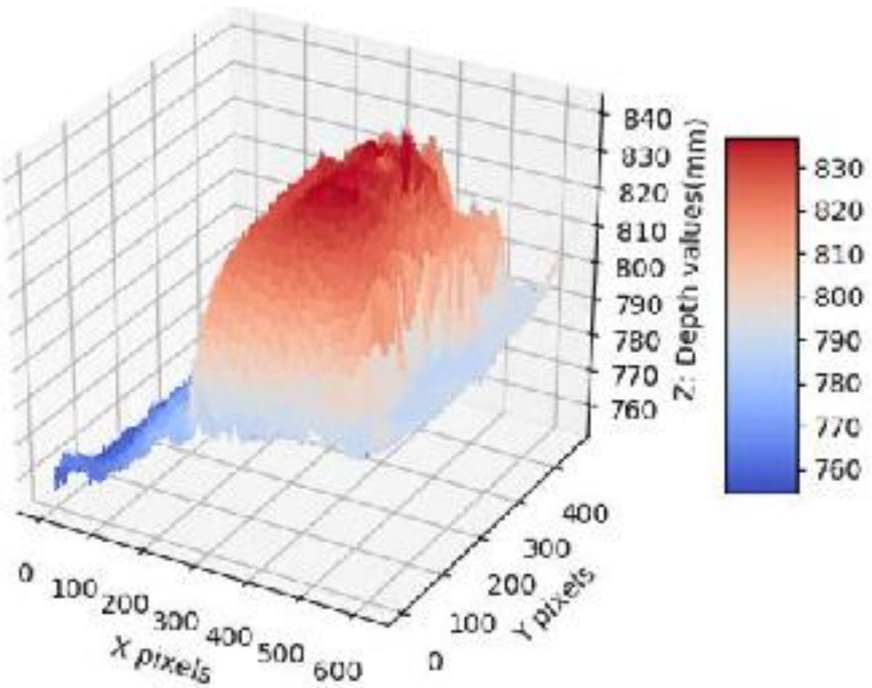
Binarization

Otsu's Method

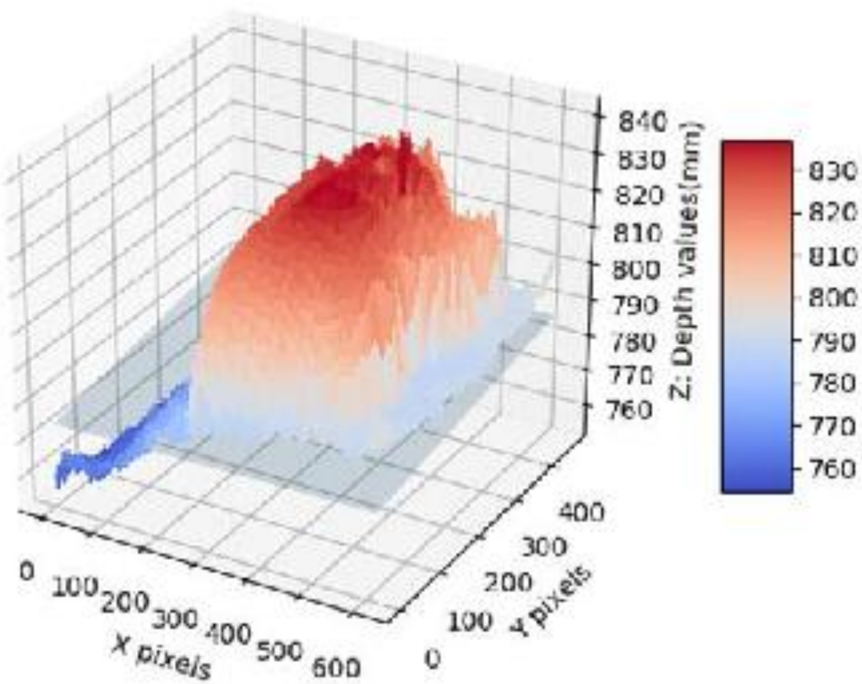
- Otsu's Method transforms the data into a usable form by "binarizing" the data - splitting into two parts

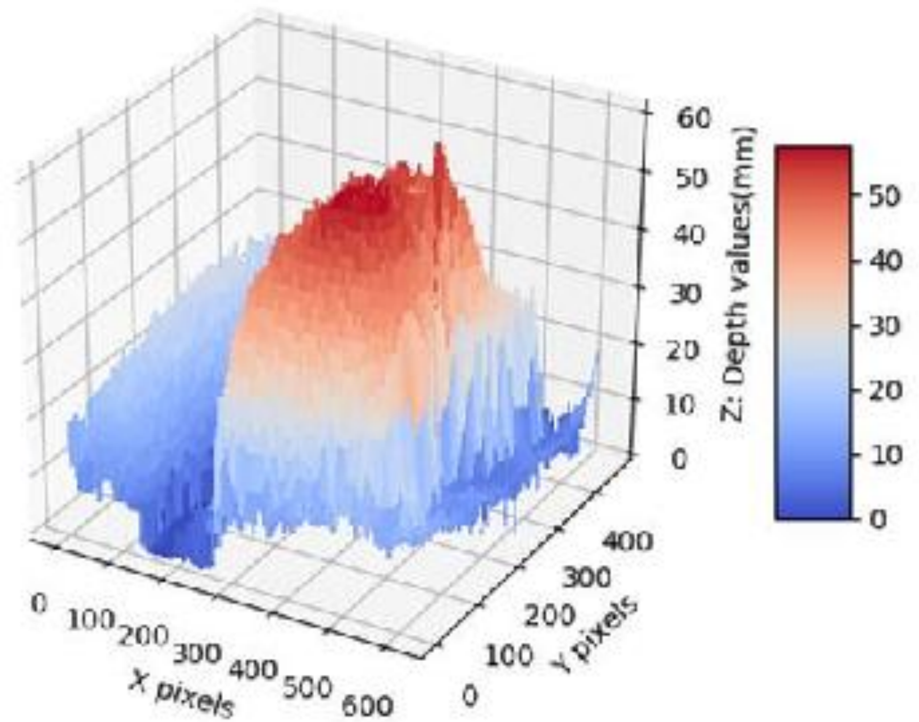


Sample Output 1

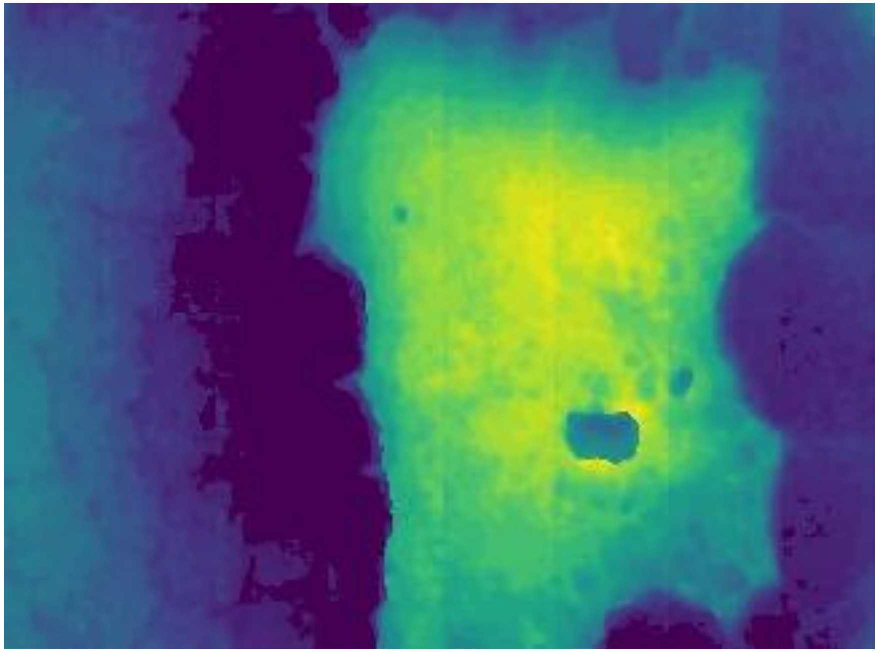


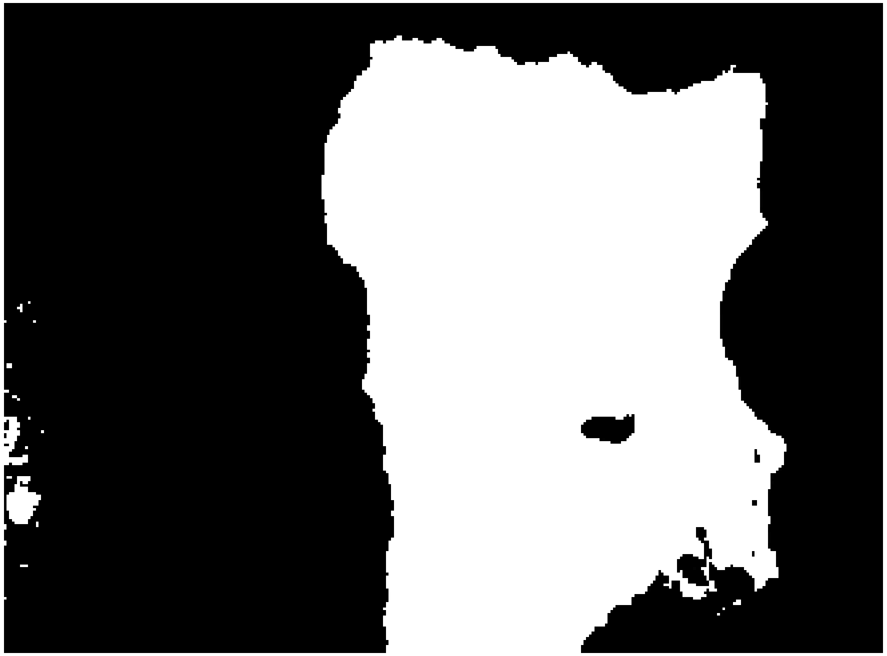
Sample Output 1



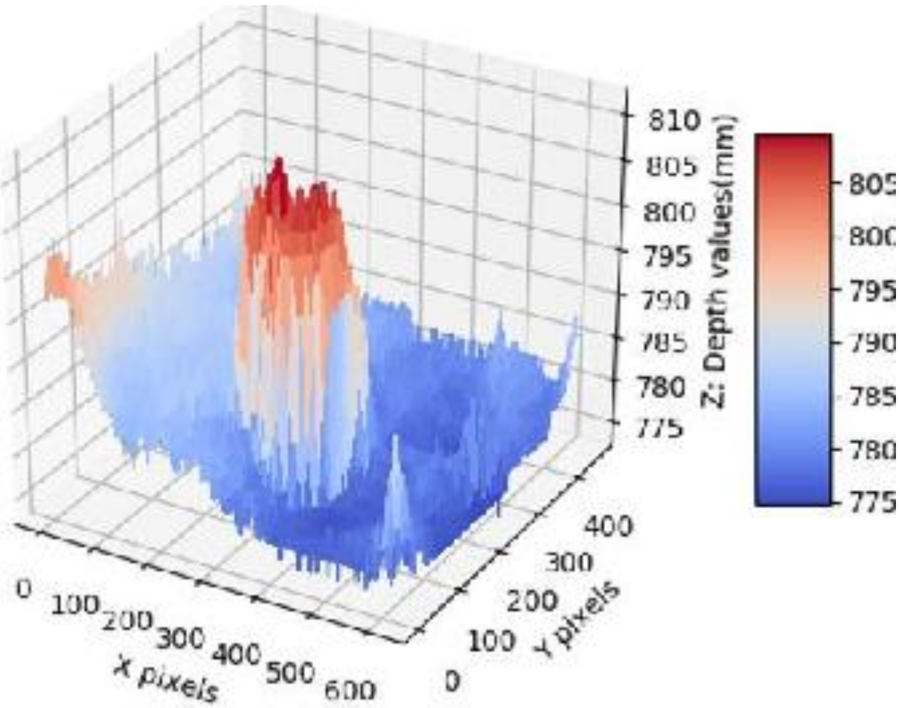


Sample Output 1

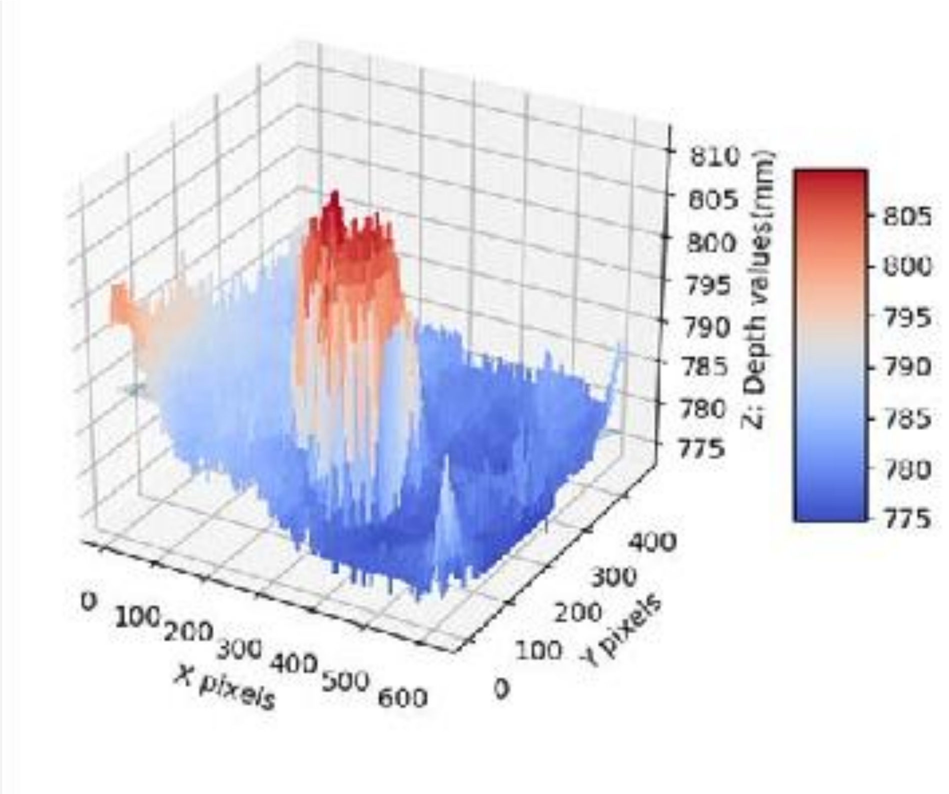


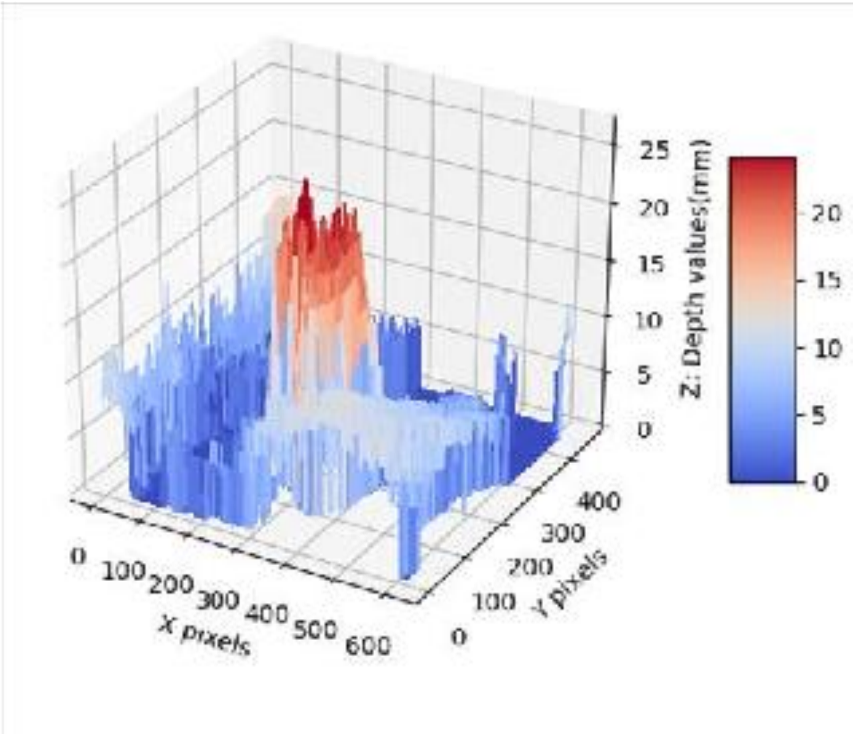


Sample Output 2

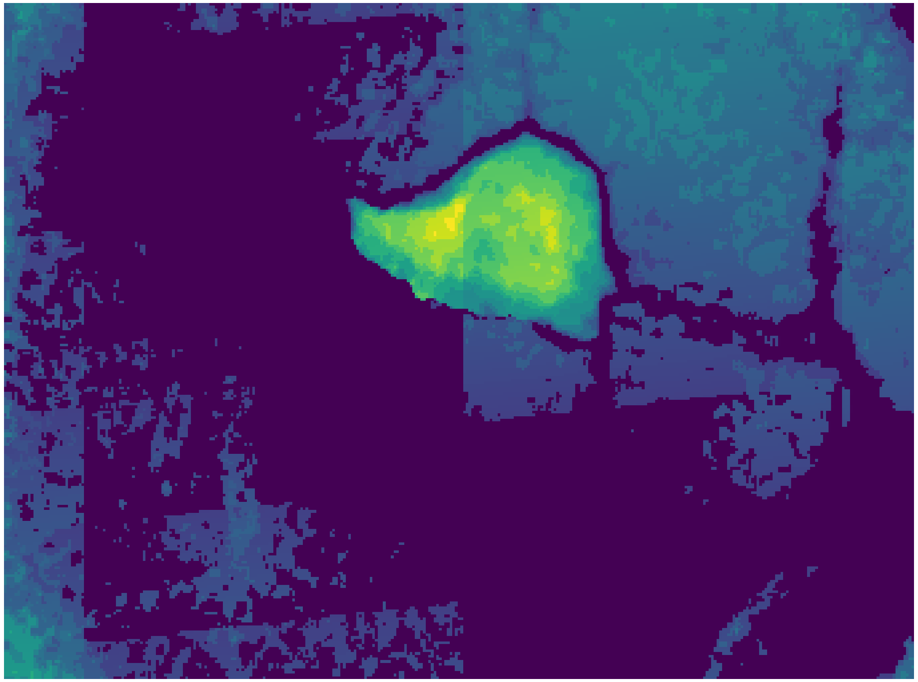


Sample Output 2

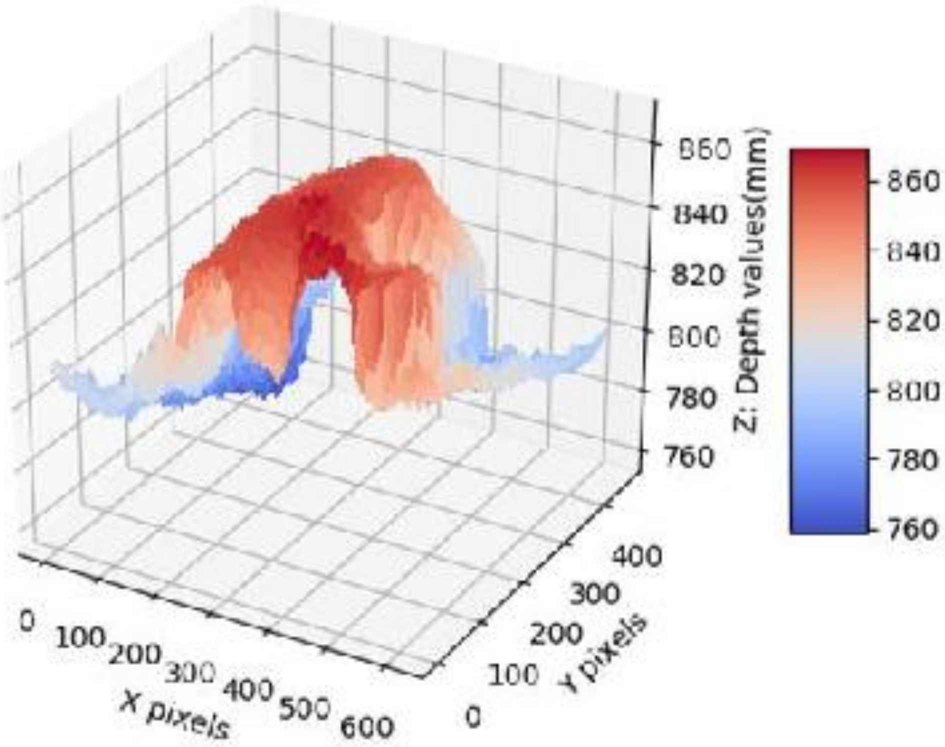




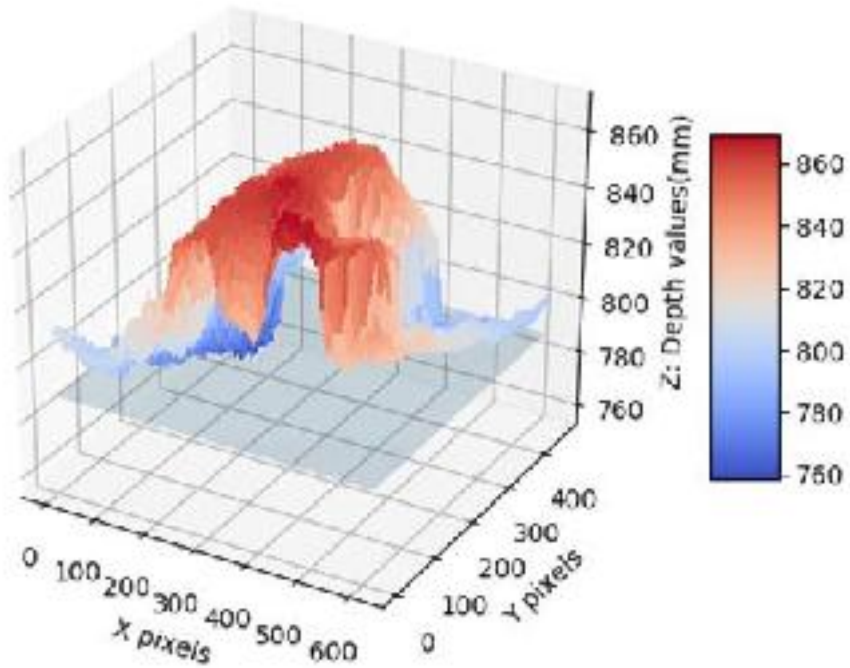
Sample Output 2

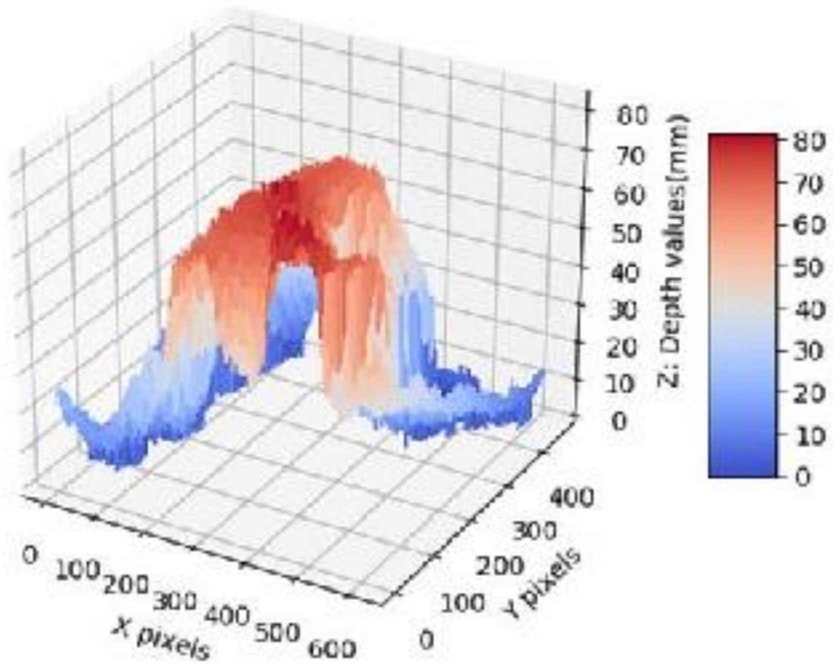


Sample Output 3

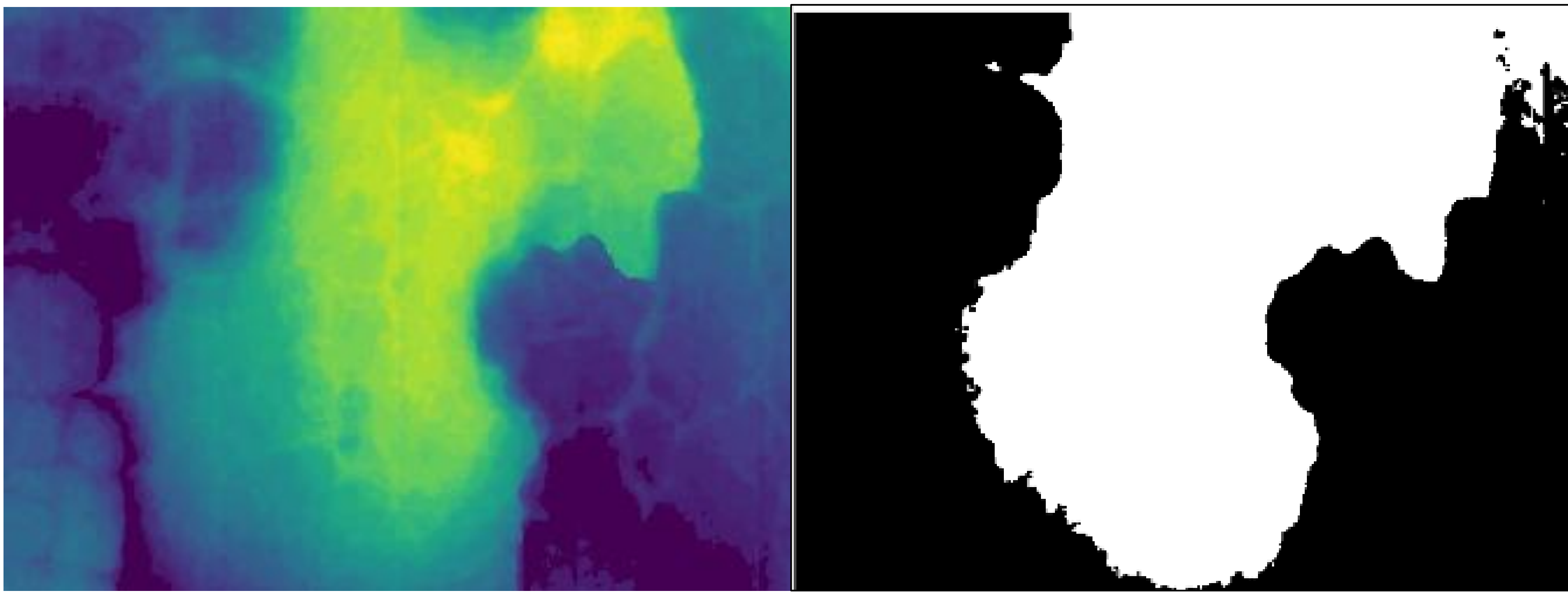


Sample Output 3





Sample Output 3



Severity

Design Question: Which severity model to utilize

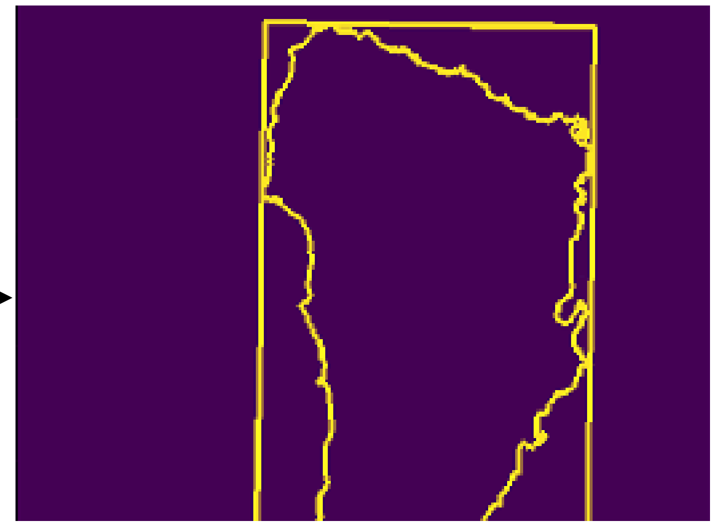
1. ASTM Standards
- Rectangular bounding
- Relevance - only potholes

2.) PASER Standards
- Histogram
- Not just potholes

Severity

ASTM and Rectangle bounding

After Otsu’s “Binarization”



Edge Detection

Pothole Severity Quantification

ASTM and Rectangle bounding

Maximum Pothole Depth	47 mm
Average Diameter	482 mm
Severity	High
Process Time	1.87 s

Maximum depth of pothole [mm]	Average diameter of pothole [mm]		
	100 - 200	200 - 450	450 - 750
13 - 25	Low	Low	Med
25 - 50	Low	Med	**High**
> 50	Med	High	High

System for running program

Design Questions: Where does the program run?

1. Send to Server
- After vehicle finishes route
- Requires large data transfer (2 - 4 TB)
- Less maintenance

2.) Process on Microcontroller
- Too slow

3.) Office Computer
- Hard disk attached to car
- More maintenance

System for running program

Design Questions: Where does the program run?

1. **Send to Server**
   - **After vehicle finishes route**
   - **Requires large data transfer (2 - 4 TB)**
   - **Less maintenance**

2.) Process on Microcontroller
- Too slow
- Incompatible with program

3.) Office Computer
- Hard disk attached to car
- More maintenance

System for running a program

Edge devices (related topic)

"Edge computing allows data produced by internet of things (IoT) devices to be processed closer to where it is created instead of sending it across long routes to data centers or clouds."

Mid-semester Issues

Learning

- Numpy and OpenCV
- Programming/ mathematics background

Speed

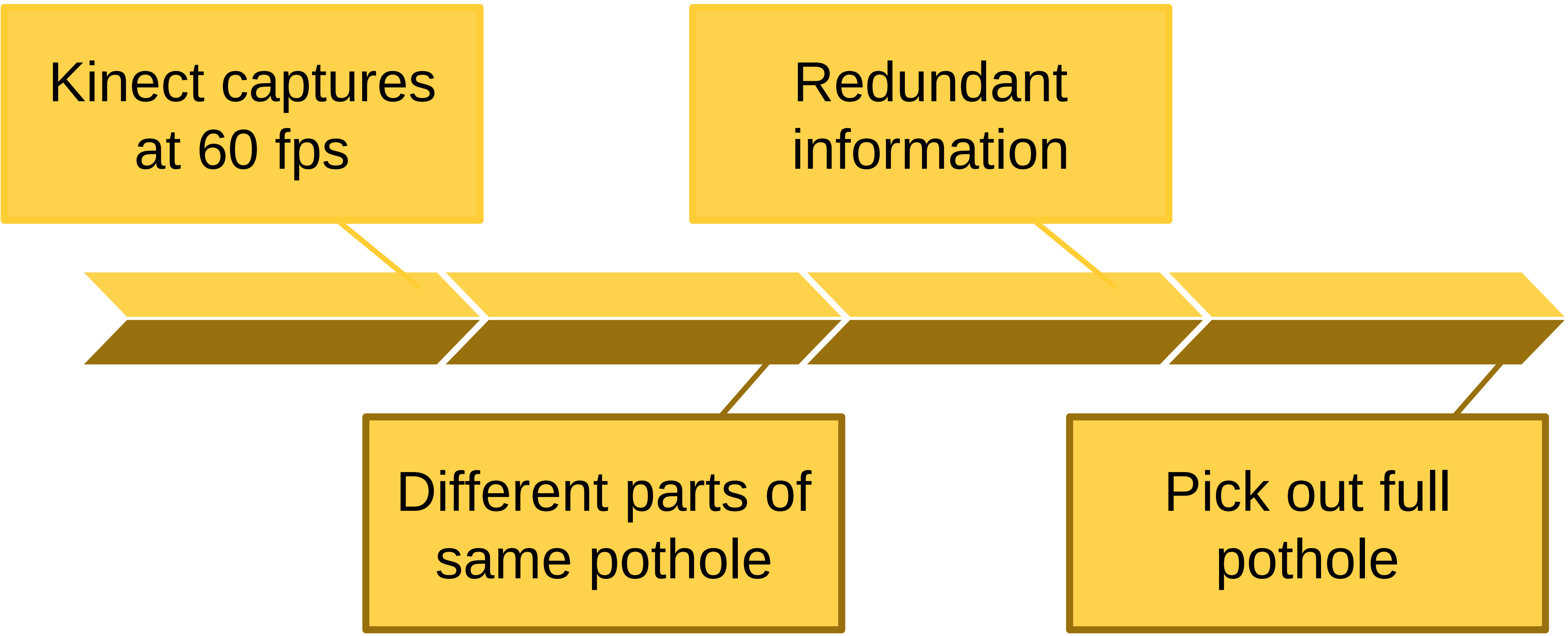
- Data processing slow

  1.87 s for one frame

Tentatives

- Overlapping frames
- Multiple potholes

Data Mapping / Image tracking



Mid-Semester Issues

Learning

- Numpy and OpenCV
- Programming/ mathematics background

Speed

- Data processing slow

Tentatives

- Overlapping frames
- Multiple potholes

Timing the Process

Plane Fitting	<u>0.50 seconds</u>
Otsu's Method	0.19 seconds
Rectangular Bounding	0.00056 seconds
Severity	2.50 e-05 seconds

1.17 seconds

Areas to Improve

Transition

- Clear transition document
- Faster development next semester

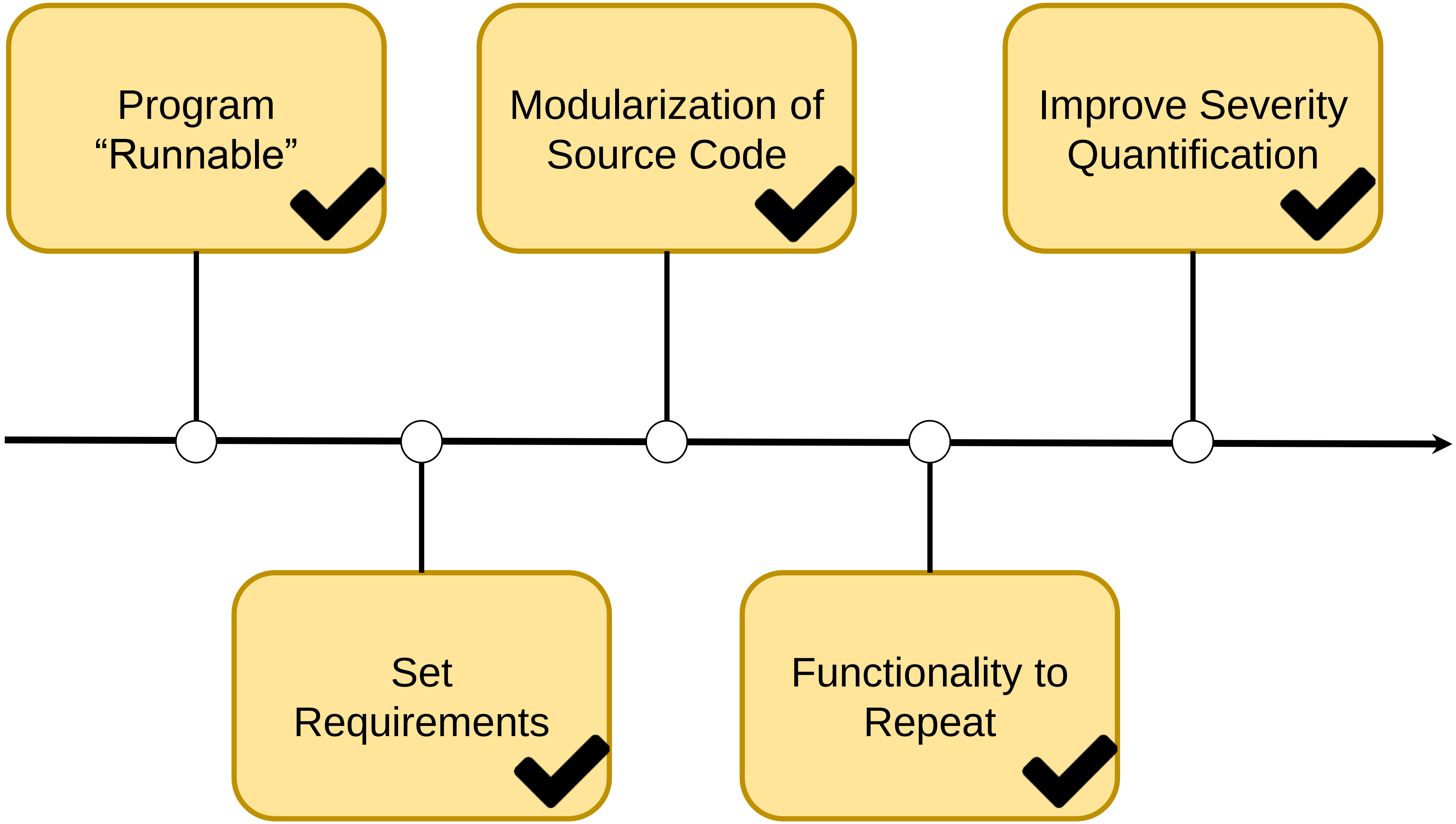
Speed

- Improve processing of bottlenecks
- 1.17 seconds

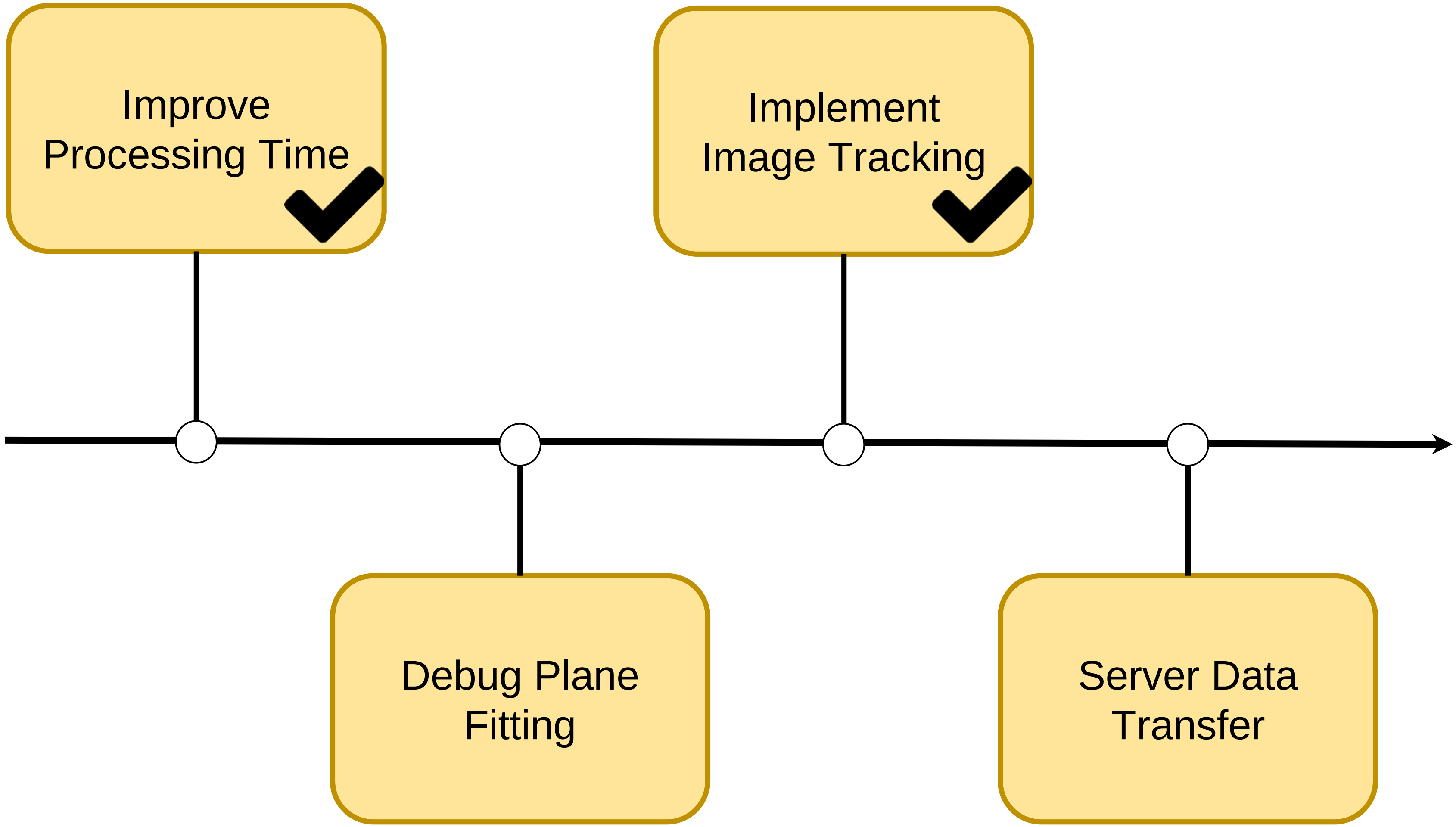
Integration

- Work with app team
- Data transfer

Semester Deliverables



Semester Deliverables



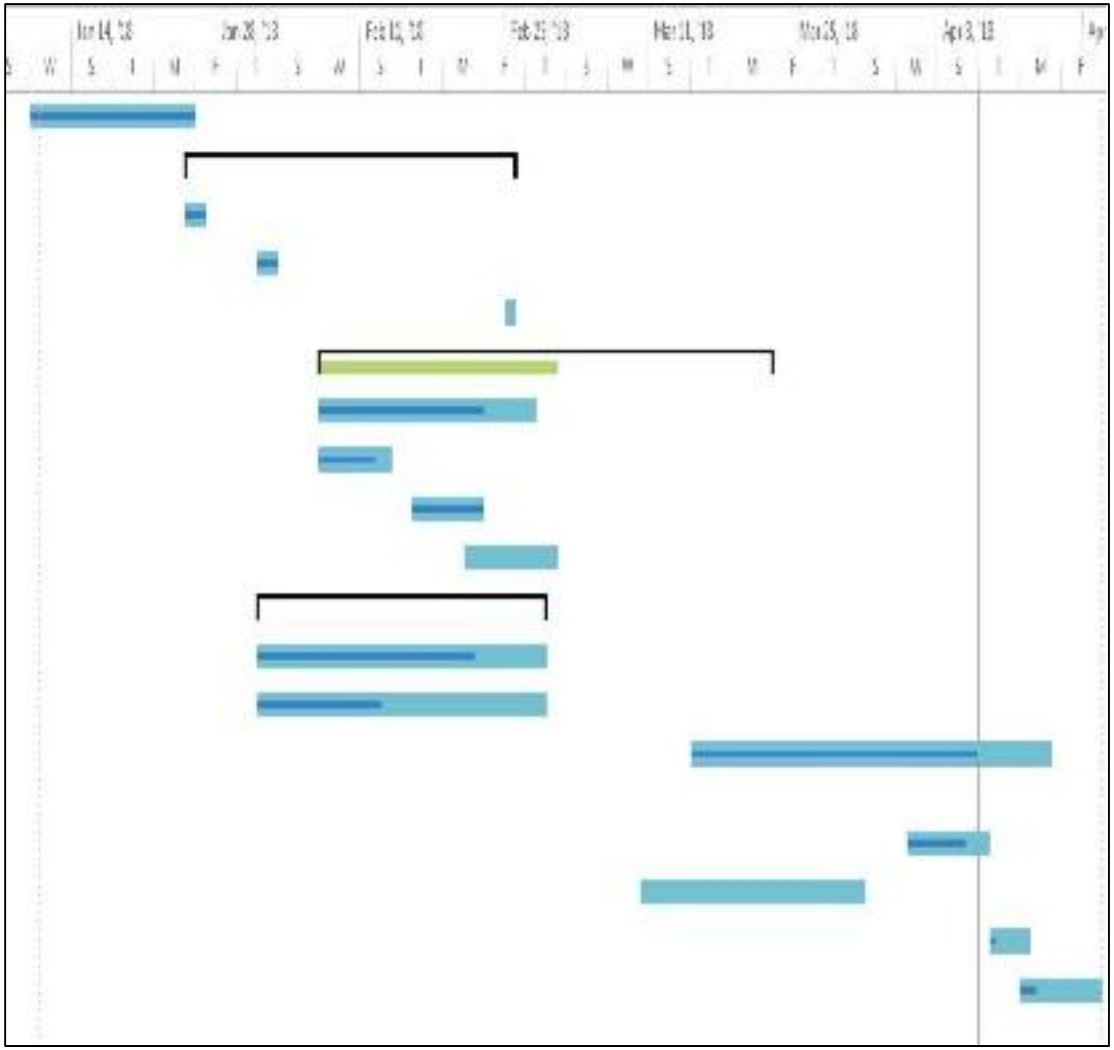
Data Analysis

Summary

1. Plane fitting
2. Otsu's binarization
3. Rectangle bounding and ASTM
4. Image tracking

Data Analysis Team Timeline

Task Name
Orientation
Learning past team's work
Analyzing the code
Troubleshooting the code
Meeting with advisor
Improving the code base
Analyzing the errors in the code
Modularizing the code
Creating breakpoints for better debugging
Have the code read different continously
Learning code base properly
Learning detection in Python
Learning severity quantification python
Image mapping (trimming multiple scans of single pothole)
Set up a server
Distortion Analysis
Pass output information to server
Transition preparation



| Jan | Feb | March | Apr

EPICS SMART CITY

Website & Application Development Team

EPICS PURDUE

Meet the Web & App Team

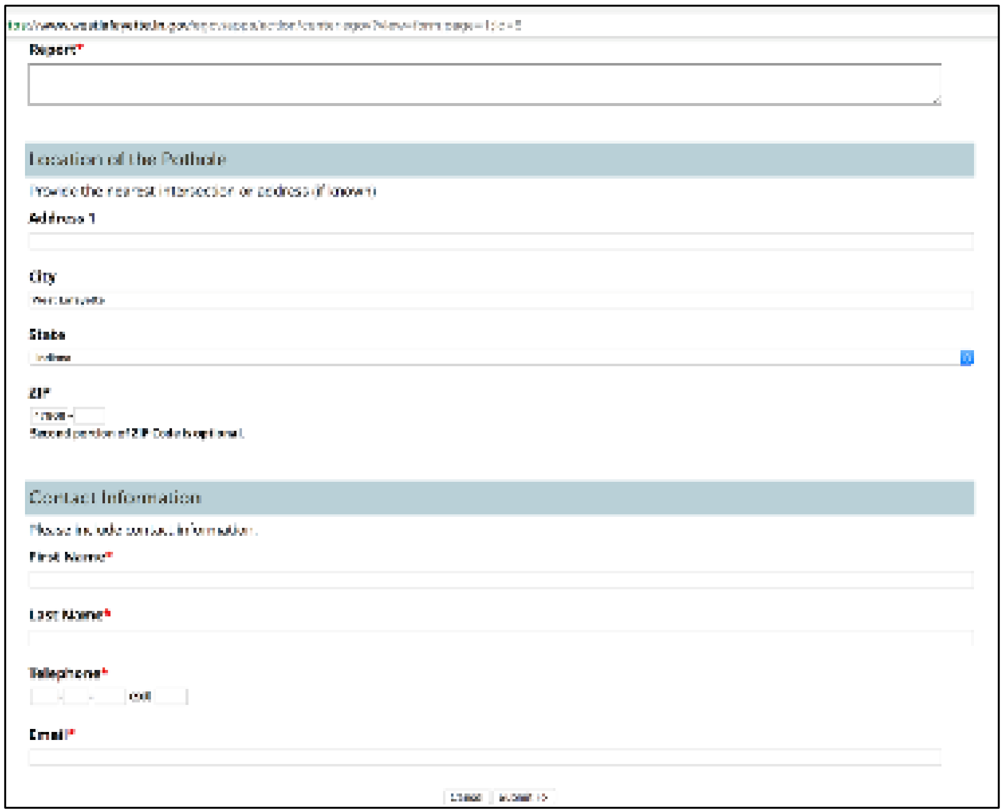
Name	Major/Year
Kartik Mittal	Computer Science / 1st
Muhammad Shorieri	Electrical Engineering / 3rd
Rachel Lee	Mechanical Engineering / 2nd
Khaing Zin	First Year Engineering / 1st

Needs and Specifications

Needs	Specifications
- Simple and easy design to use - Integrated data server - Efficiently displaying the information	- Google API implementation - Camera app from the device - Position of user using GPS - Autonomous updation of data

Existing Solutions

Current Reporting


Report*

Location of the Pothole

Provide the nearest intersection or address (if known)

Address 1

City

State

ZIP

Second portion of ZIP Code is optional.

Contact Information

Please include contact information.

First Name*

Last Name*

Telephone*

ext

Email*

Existing Apps

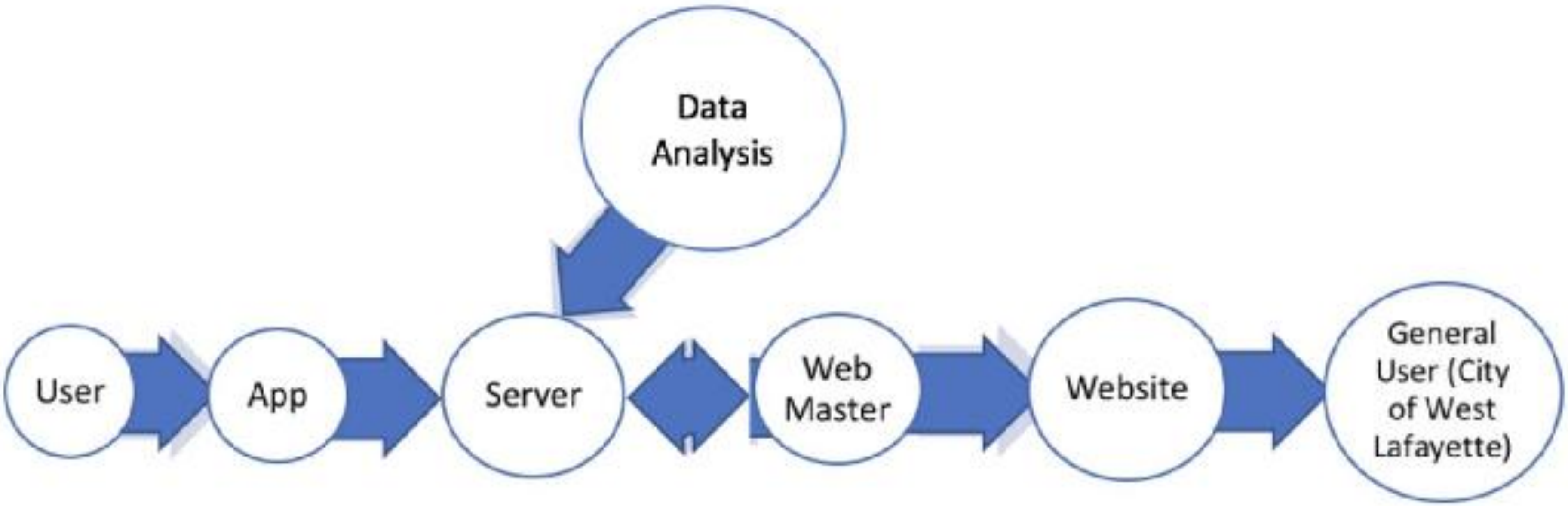
Initial Goals

Develop design for functional app

- Learn and understand Android Studio
- Fix the User Interface and make it robust
- Choose the appropriate server
- Link the website and application
- Try to find the bugs and fix it

Flow Chart

Proposed Plan of Design Process



Progress

Current Progress

- Set up proper workspace with GitHub
- Fixed the abrupt behaviour of the layout
- Improved on the user interface of the app
- Integrated Google's Firebase server for storage
- Added a refresh button
- Working on the website prototype

Decision Matrix on Choosing Server

Factors: (0-5 scale)	Cost	Quality	Reliability	Compatibility	Total
Google's Firebase	4	5	5	5	**19**
Amazon (AWS)	3	4	5	3	**15**
Microsoft's Azure	4	5	5	3	**17**

Cost - How much to run the server
Quality - Speed, bandwidth
Reliability - Security, how often server down
Compatibility - Its usability with application

Ideation

Map and options

Simple, few clicks

Single screen

Application Demo

DEMO

RealTime Database Storage

https://smart-city-app-epics.firebaseio.com/

smart-city-app-epics

- -L75UIfgMJHZymGnUnCa
  - encodedImage: "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQE..."
  - latitude: "40.42455005572096"
  - longitude: "-86.91887106746435"
  - severity: "Minor"
  - timeStamp: "2018-03-08T11:41:33"

Website

Current Features

- Visualize data collected by application and kinect
- Can be sorted by severity, date or street name
- Written in JavaScript with Node.js
- Currently running on free temporary server (Cloud9)

Current Prototypes

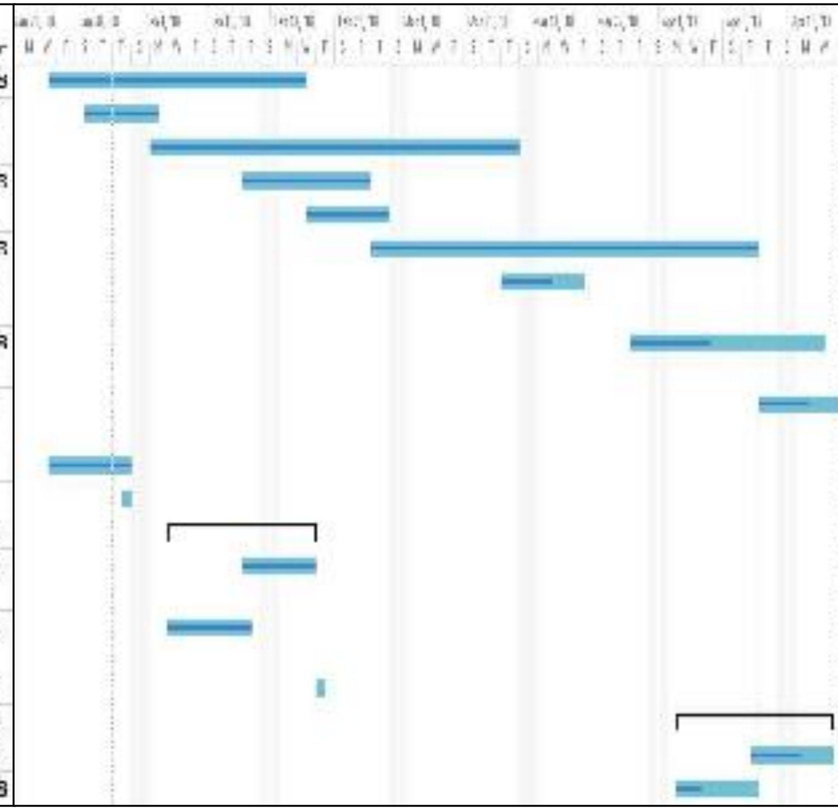
Goals for the Next Semester

Fully Functional Website and Application

- Final version of the user app
- Full website (local server) for the city to track the data
- Test for Human-Computer interaction
- Make both products visually appealing
- Add more functions - report an issue, alert for general issues

App Development Team Timeline

	Task Mode	Task Name	Duration	Start	Finish
✓		Learn how to use Android Studios	20 days	Thu 1/25/18	Wed 2/21/18
✓		Test last semester's App	6 days	Mon 1/29/18	Mon 2/5/18
✓		Work on developing App	30 days	Mon 2/5/18	Thu 3/15/18
✓		Set up server to receive data	10 days	Thu 2/15/18	Wed 2/28/18
✓		Finalize design	7 days	Thu 2/22/18	Fri 3/2/18
✓		Finalize the app	30 days	Thu 3/1/18	Wed 4/11/18
		Test the app on more than 2 device	7 days	Thu 3/15/18	Fri 3/23/18
		Take feedback from others and improve on it	15 days	Thu 3/29/18	Wed 4/18/18
		Finalize the transition document and design review	7 days	Thu 4/12/18	Fri 4/20/18
✓		Semester Plan/Gantt Chart	7 days	Thu 1/25/18	Fri 2/2/18
		Notebook Review 1	1 day	Fri 2/2/18	Fri 2/2/18
✓		**Mid Semester Design Review**	**12 days**	**Wed 2/7/18**	**Thu 2/22/18**
✓		Preparation for Mid-Semester Review	6 days	Thu 2/15/18	Thu 2/22/18
✓		Create slides for Mid-Semester Review	7 days	Wed 2/7/18	Thu 2/15/18
		Notebook Review 2	1 day	Fri 2/23/18	Fri 2/23/18
		Final Design Review	**13 days**	**Tue 4/3/18**	**Thu 4/19/18**
		Preparation for Final Review	7 days	Wed 4/11/18	Thu 4/19/18
		Create slides for Final Review	7 days	Tue 4/3/18	Wed 4/11/18



| Jan | Feb | March | Apr

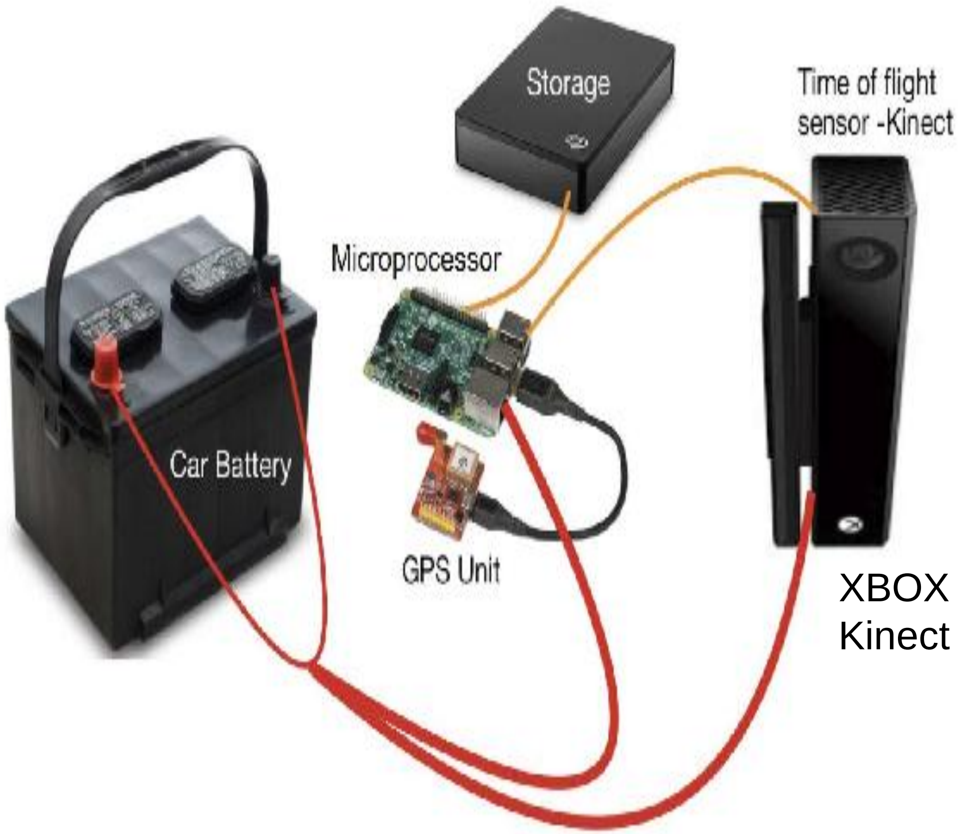
EPICS SMART CITY

SPRING 2018

EPICS PURDUE

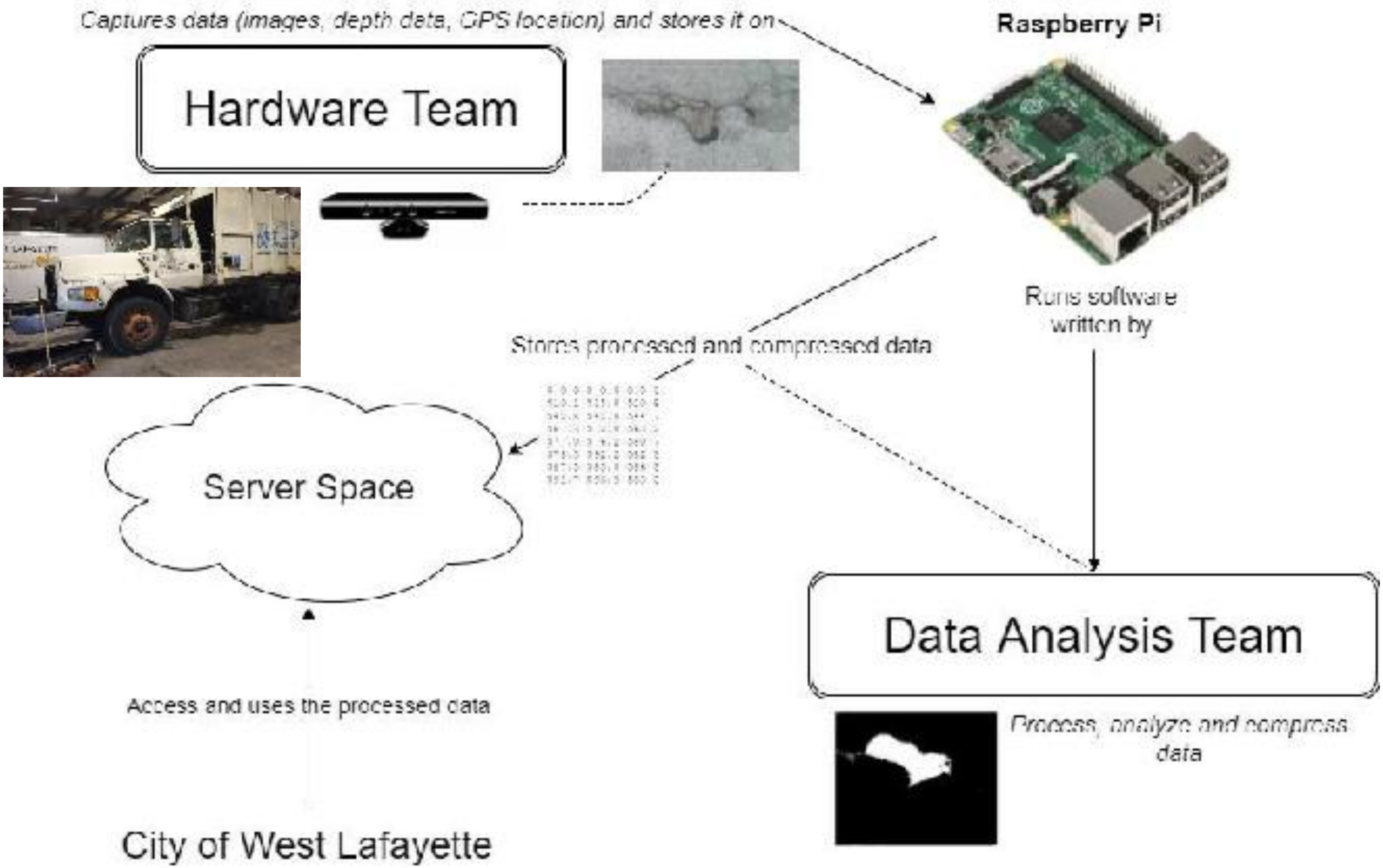
Hardware

- 3 Microsoft Kinects
- Microprocessor
- Hard Drive
- GPS

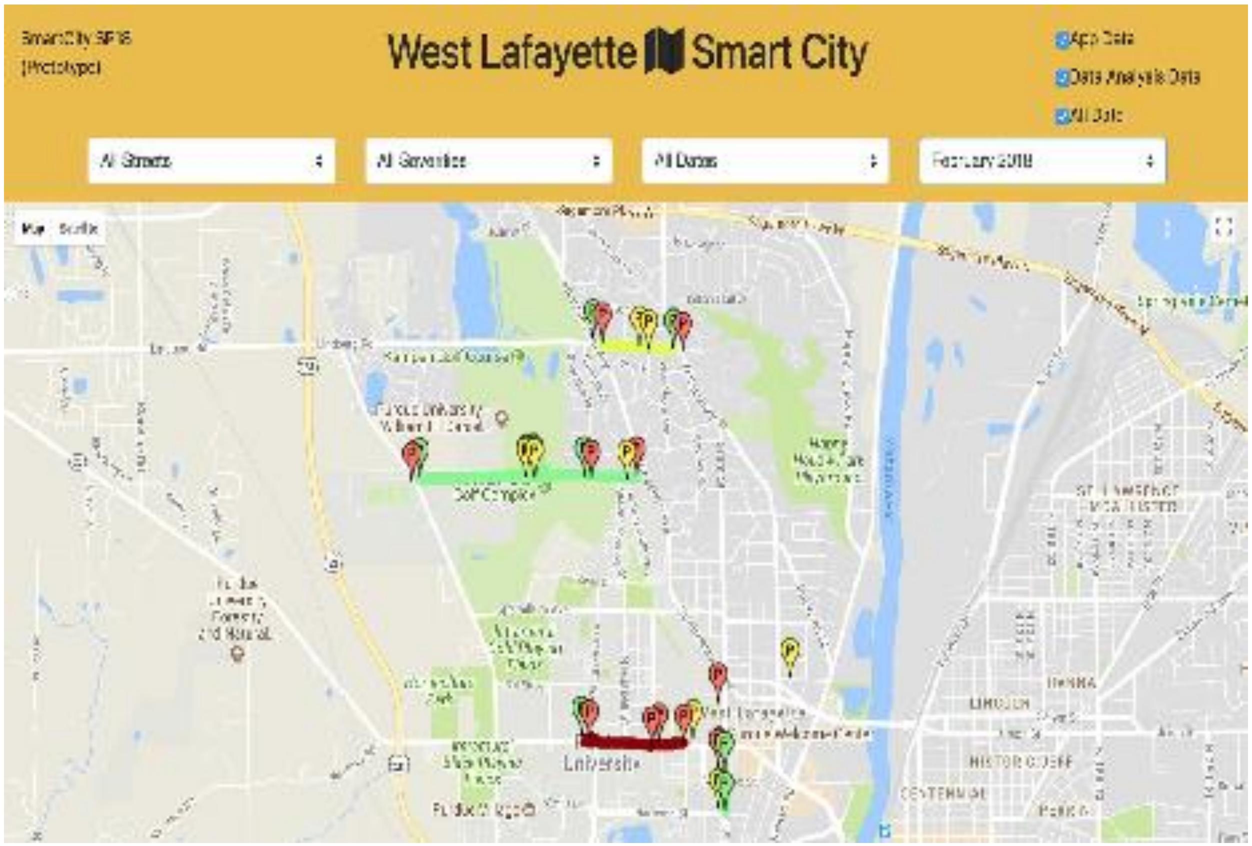


XBOX
Kinect

Data Analysis



Website and Application Development



Thank you!

Comments or Questions?