LMC 6650: Seeing like a Bike

Surface Quality

Milestone 3

Team: Shruti Dalvi | Kristin Hare | Nene Igietseme | Jayanth Mohana Krishna

Overview

Working on three methods of gathering data

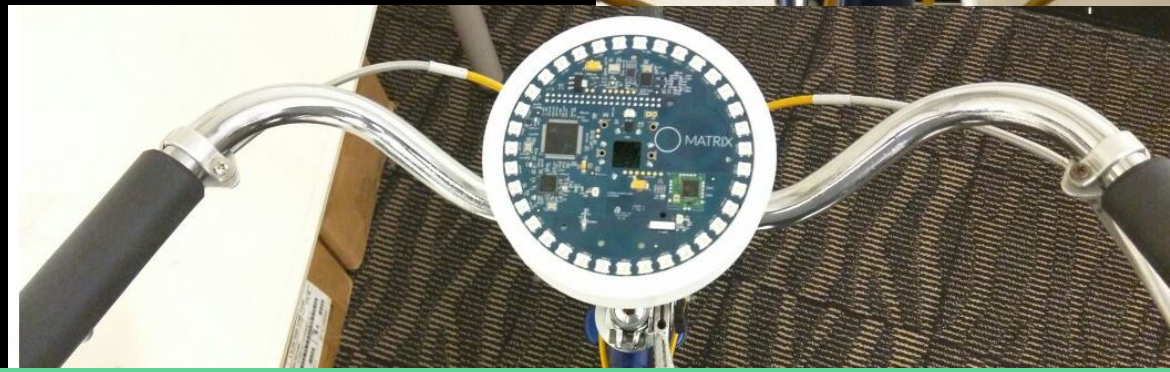
- Through the Matrix - environmental and inertial data
- Surface Mic - for picking up vibration from the road
- Camera, Crosshair laser and OpenCV (experimental) for determining surface quality

General

- Server implementation and processing
- Data gathering and visualization?

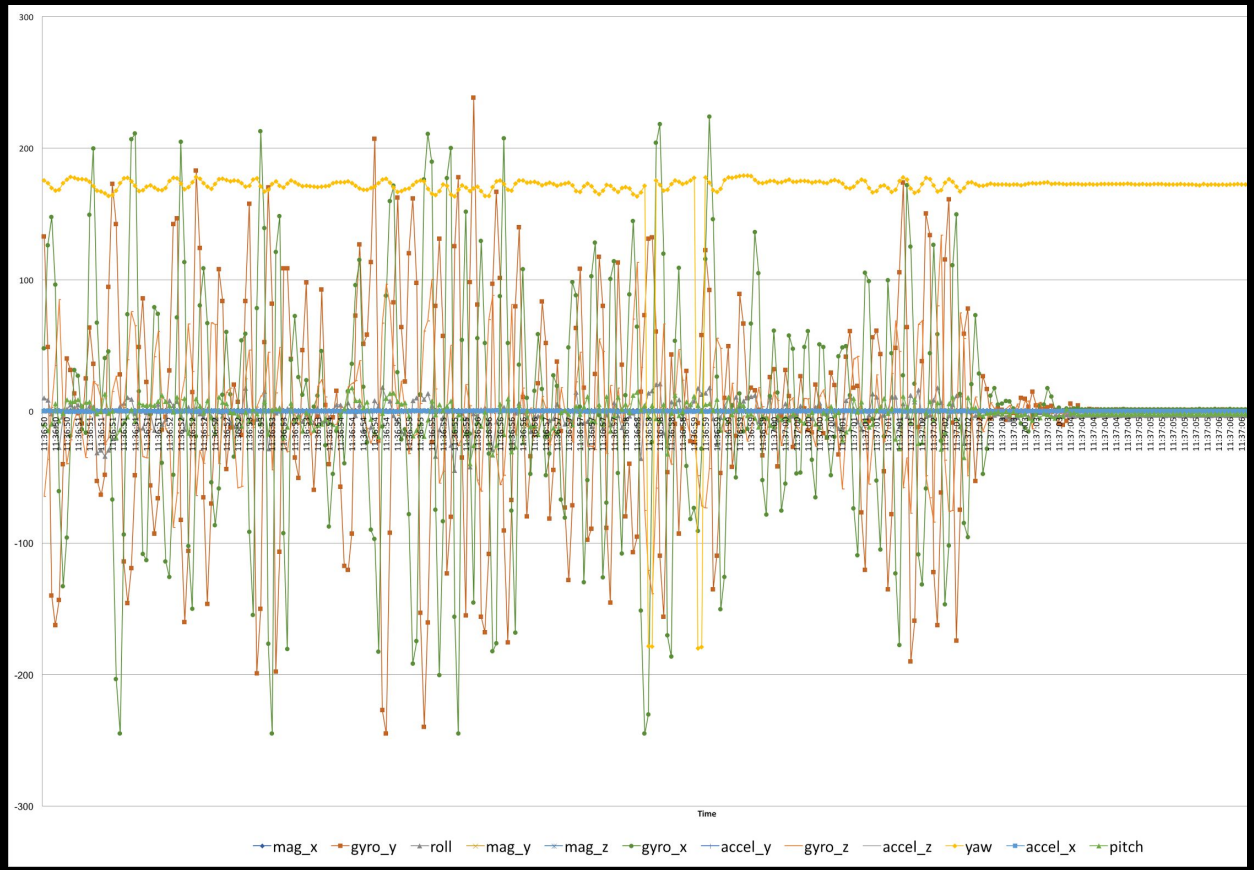
Raspberry Pi + Matrix One Server

- Collecting data from on-board sensors in JSON format
- Python servers implemented and tested
- Added code for ad-hoc processing for non-standard data formats



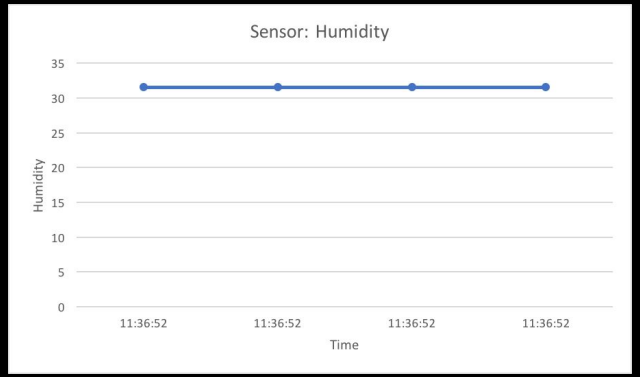
Test Data Gathered

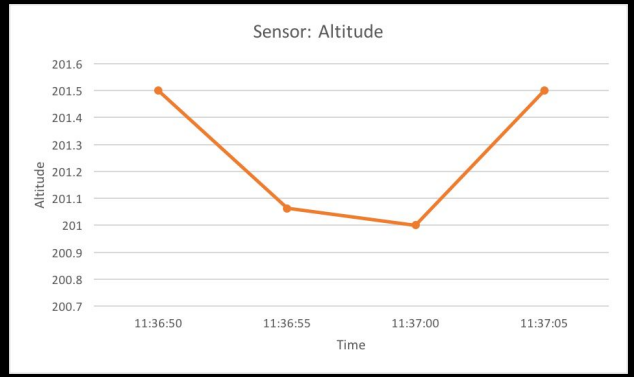
- Accelerometer, Gyroscope, Magnetometer measured
  - Direction of motion in the x, y, & z planes
  - Roll
  - Pitch
  - Yaw

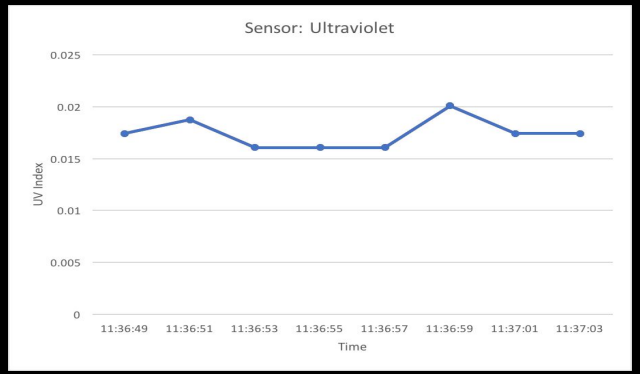


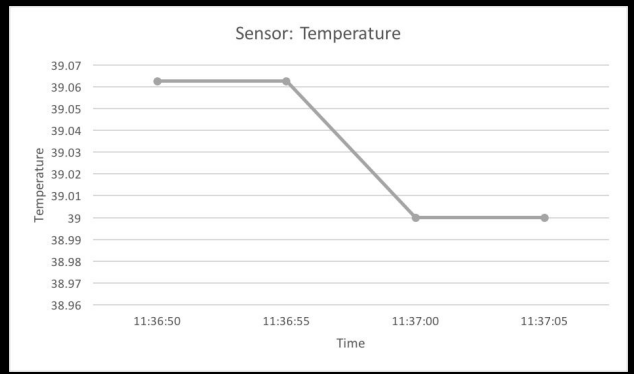
Test Data Gathered

- Humidity
- Ultraviolet
- Altitude
- Temperature
- Pressure









Progress : OpenCV

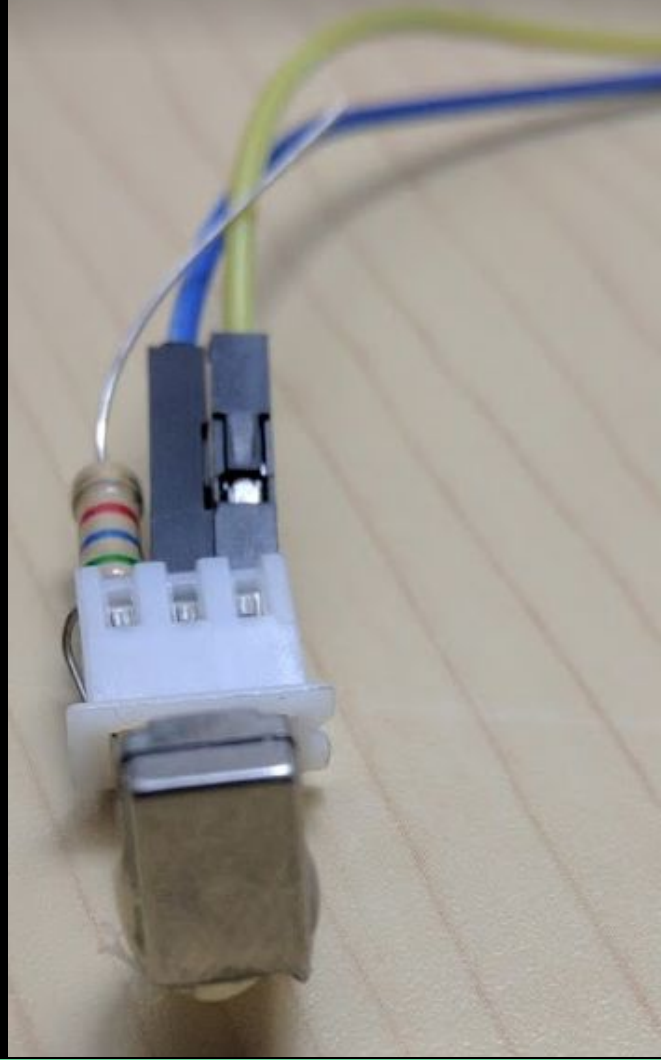
- Obtained the crosshair laser yesterday (3.12.17)
- 3D modeled a part to hold the camera and the laser pointer; will print today
- Tried example code to determine surface smoothness; OpenCV constants hard to understand
- Alternative: LIDAR / proximity

Physical Interfacing on the Bike

- GoPro mounts to go on the bike
- Mounted on the handlebar so it can measure pitch, roll and yaw
- Requires (initial) calibration - during installation / on the server
- Ordered a 26800 mAh battery with 3 USB ports (2A max each)
- Should mount the luggage rack on the bike, get USB cables for Pi(s)

Help / Action Needed

- A team to let us use a GPIO pin for Analog Read (for the surface microphone) - we ran out of pins on our Pi
- Provide sensor data to the central Raspberry Pi server in the prescribed format
- Move all codebase to one/two Raspberry Pis



Reminder - Data Format

```
{
‘sensor’: ‘sensor_name’,
‘timestamp’: ‘YYYY-MM-DD HH:MM:SS.ssssss’,
‘data’: {sensor_payload}
}
```

- Send *sensor_payload* in a JSON-like format (CSV, text is OK) - will process to valid JSON on the Python server
- Will generate timestamp and sensor name on the server - based on HTTP endpoint and server time

Next Steps

- Mounting all the sensors on the bike - does any team need help with 3D dimensioning, modeling and printing?
- Use Microsoft Azure or something similar to analyze and visualize data
- More centralized work?