March 22, 2017
6.S062
The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring

Key idea: use accelerometers to map road surface, identify locations of potholes

Why? Potholes damage hundreds of thousands of cars a year, and cause many injuries.

Hardware: linux-based PC, with a 3 axis accelerometer, mounted in a fixed location & known orientation in glove box. Powered via cigarette lighter.

Software: continuously record acceleration data, upload "opportunistically" (when wifi connectivity is present).

(Aside: show open wifi / cabernet slides)

Abstraction: dpipe

server:

```
dpipe listen localhost pipeName
```

client:

```
dpipe write file serverIP pipeName
```

Guarantees that every file written is eventually delivered, in file order. Handles fragmentation of large files into small chunks & reassembly, acknowledgement, deletion, etc. Important when connectivity is very intermittent.

Not totally clear from paper, but system had a training phase, where we collected everything, and then a "live" phase where we only uploaded possible pothole detections.

Live mode diagram

```
Car                                   Server
                           dpipe
Collection -> FIltering  ---------->   Clustering -> Display
```

Detection algorithm:
Check if speed > x

```
s = window of 256 3axis accel samples
f = high pass (s)  --- show
peak = max(f[z])
peak_loc = index(max(f[z]),f[z])
if (peak > t1)  // peak filter
    x_filt = f[z][peak_loc-40, peak_loc+40]
    x_max = max(x_filt)
    if (x_max / z_max > t2)  //xz ratio
        if (zmax / speed > t3) //speed ratio
            if more than k detections at this location, emit pothole
```

Intuition behind stages:

high-pass -- to remove bias, and large accelerations due to dynamics of car -- potholes are short, sharp bumps lasting a few 10s of ms

z peak -- potholes mostly show in the axis

x/z ratio -- potholes should have significant x axis energy (a tilt to one side), unlike expansion joints and others

speed ratio -- lots of little bumps at higher speeds -- real potholes will be even bigger at high speeds

clustering (more than k detections) -- to remove spurious events

(Show code examples -- see pothole-demo.zip)

Note that these filters depend on various thresholds. Hand tuned the values of these thresholds (why not machine learning?), by sweeping parameters to find the most effective values.

Data collection:

Drove zipcars over a bunch of potholes (and other road anomalies), labelled by having a passenger mark each pothole / manhole / etc

"loosely labeled data", marked segments as "none/few/many" for each road anomaly

"wild" data -- just taxis driving to see if we can find new potholes

Results :
- slide 23 (page 48): showing overall performance with different steps on different classes of hand labeled data
    - ideally only pothole traces would find any potholes
    - note that expansion joints vs potholes are often confused
    - adding z/x filter and speed filter helps a lot

<u>How well does it do?</u>

About 7% of non-pothole labelled data samples are marked as potholes, and 93% of potholes are properly labelled

(No smooth road is marked as potholes)

We can't really know how many false positives there are on real roads, because training data is a biased to have way more road anomalies than non-anomalies

We also can't really know how many potholes we are missing, since training data isn't exhaustive

We can estimate upper bound on false positives on real roads by looking at loosely labeled data where we think there are no potholes (show slide)

Real world data -- found a bunch of potholes, few confusions