Just a T.A.D. (Traffic Analysis Drone)

Senior Design Project 2017: Final Project Review

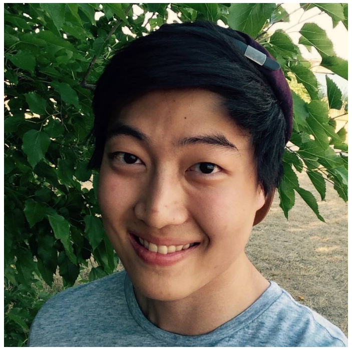
Meet the Team

- Cyril Caparanga (CSE)

- Alex Dunyak (CSE)

- Christopher Barbeau (CSE)

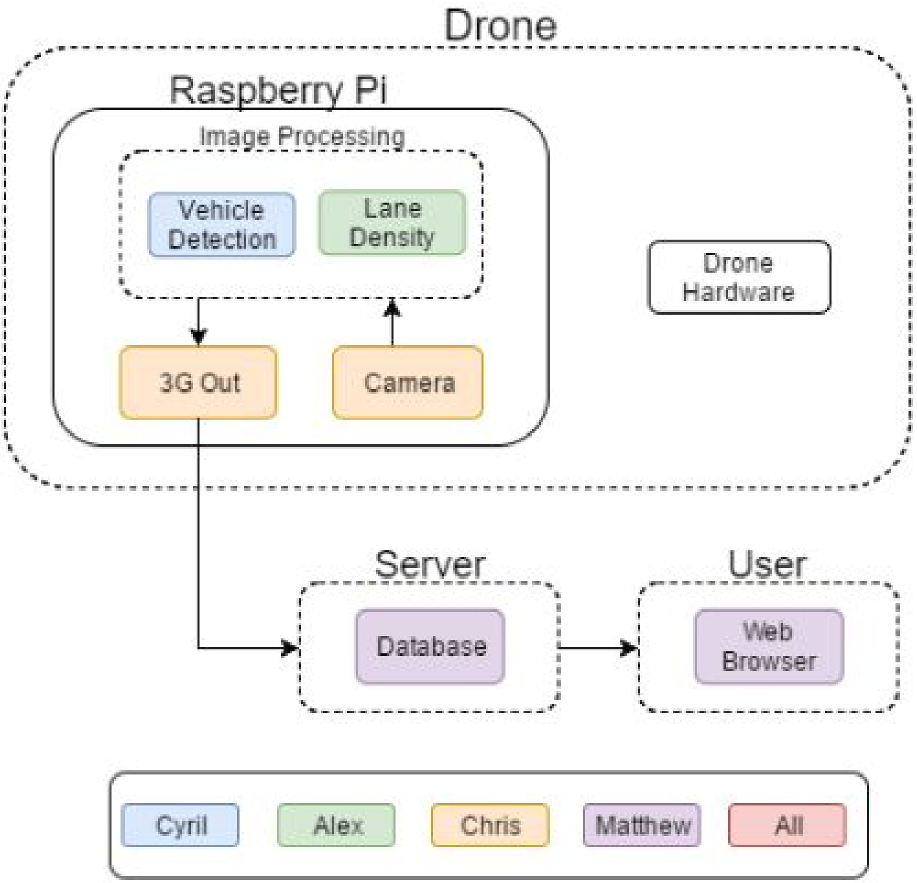
- Matthew Shin (CSE)



Recap

- Traffic data needs to be more comprehensive as more and more cars are on the roads
- Current traffic data collection methods are expensive and/or insufficient
- An Unmanned Aerial Vehicle (UAV) can be used to provide aerial image and video
- Image processing will analyze the image/video for car density and spacing on the drone
- This data is sent to a server in the cloud for display

Block Diagram - FDR



CDR Deliverables

- Alex/Cyril: Image processing
  - ✓ Identifies visible cars in own test images
  - ✓ Calculates density of cars as cars per distance or as car to road ratio
  - ✓ Integrate server
  - ✓ Integrate camera
- Matt: Data Server
  - ✓ Automatically update web app to display most recent database content
  - ✓ Assemble and test the 3DR Iris+
- Chris: Raspberry Pi
  - ✓ Camera interfaces with image processing software
  - ✗ Send image data to server via 3G

FPR Deliverables

- Alex/Cyril
  - ✓ Alex: Completion and debugging of software
  - ✓ Cyril: Finalize integration of software with camera and sending to server
- Matt
  - ✓ Become proficient in piloting the drone for demo
  - ✓ Test and develop web application
- Chris
  - ✗ DroneKit integration
  - ✗ 3G integration (or WiFi)

DroneKit and Autopilot

- Difficult to interface with drone’s flight controller due to preexisting hardware
- Has easy to interface autopilot via Android or Windows app
- App contains necessary metrics
  - Latitude
  - Longitude
  - Yaw
  - Altitude

New algorithm

Previous algorithm far too performance intensive to run on Raspberry Pi.

Simpler idea: find the lane, and iterate over a line through the lane, finding both the mean and the variance.

If a sequence of pixels (with some tolerance for error) are further than a standard deviation from the mean, then declare that to be a car.

Works with stationary vehicles, as it is memoryless.

Visual results

Density and interval

Density: Per lane, density is

$$\frac{\text{pixels belonging to detected car contours}}{\text{720 pixels per scan}}$$

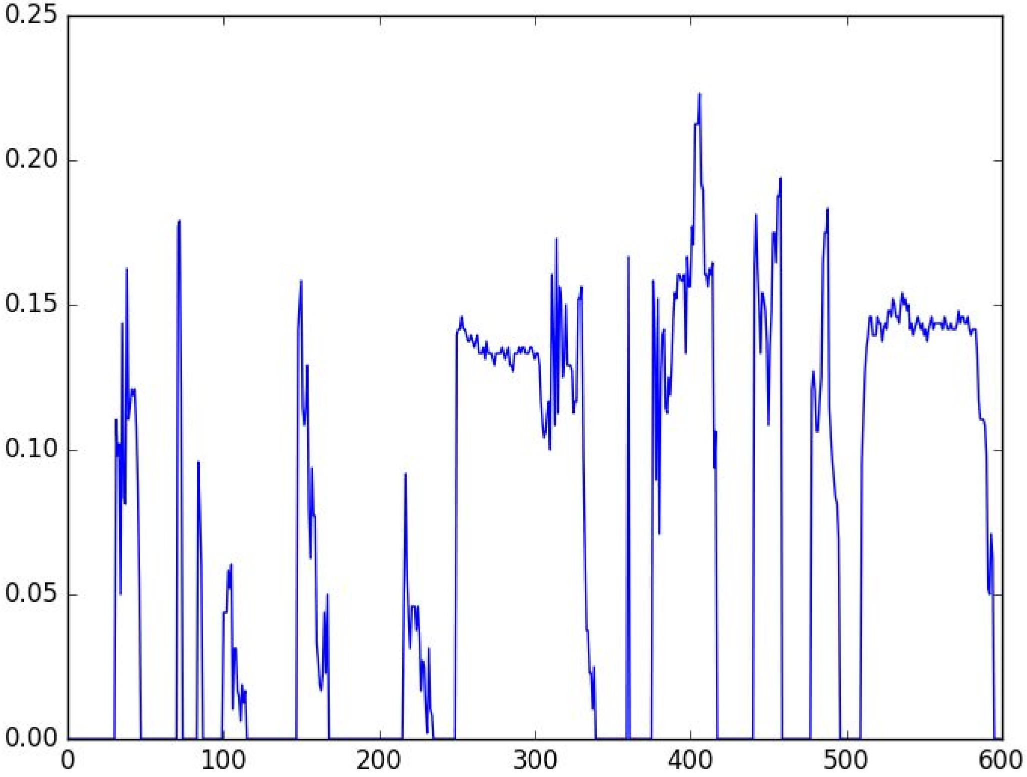
Interval spacing: Interval spacing can be described by

Given two adjacent cars, c_x and c_y, where T_x and T_y are the adjacent thresholds of those cars,

$$I = (T_x - T_y) \times \frac{h * 0.665}{480}$$

Density Graphs

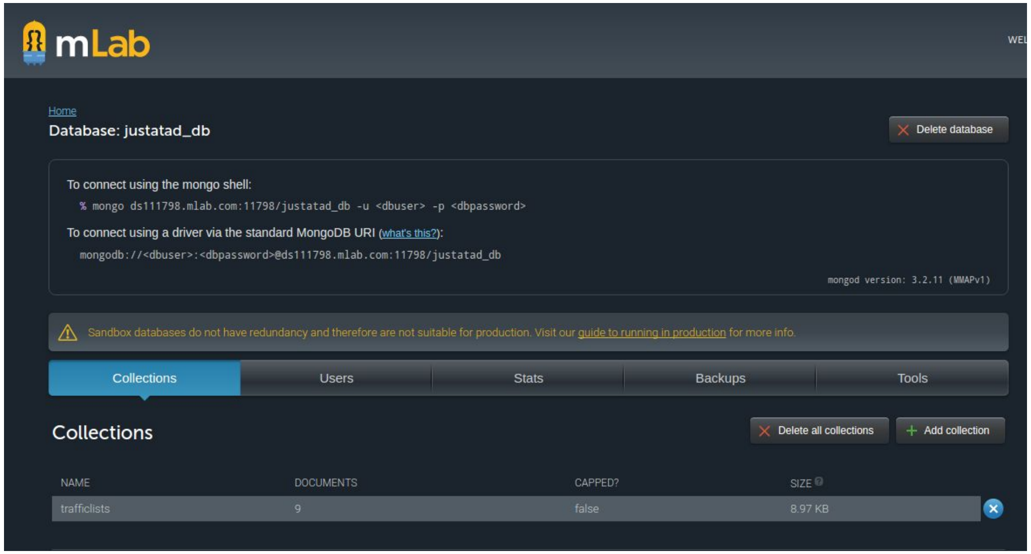
Density (Pixels declared cars over total pixels)



Frame number

Data Server

- Server receives processed data (Density, Interval)
- Displays data on web page



Database: justatad_db

To connect using the mongo shell:

```
% mongo ds111798.mlab.com:11798/justatad_db -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds111798.mlab.com:11798/justatad_db
```

```
mongod version: 3.2.11 (MMAPv1)
```

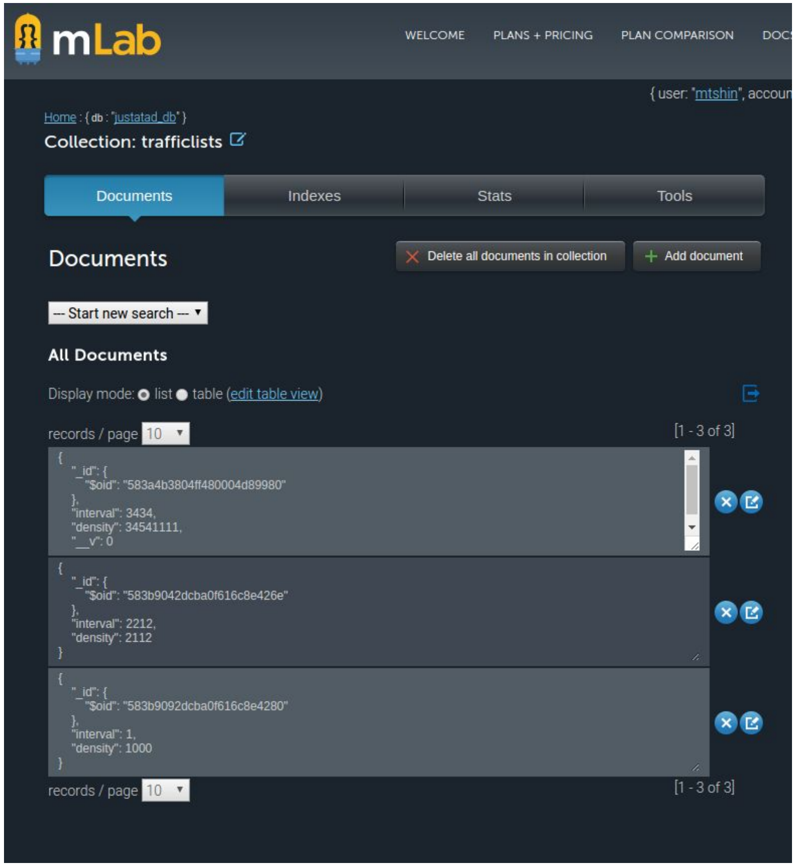
Sandbox databases do not have redundancy and therefore are not suitable for production. Visit our guide to running in production for more info.

Collections

NAME	DOCUMENTS	CAPPED?	SIZE
trafficlists	9	false	8.97 KB

Data Server Implementation

- Remove as much load from Raspberry Pi as possible for image processing
- Database hosted on cloud (mongolab)
  - Deployed on AWS (Reliable, free up to 500 MB)
  - Database visualization
- mongoDB
  - JSON documents allow for varying structure
  - Flexible (dynamic schemas)



{ user: "mtshin", accoun

Home : { db : "justatad_db" }

Collection: trafficlists

Documents

All Documents

Display mode: ● list ○ table (edit table view)

records / page 10 ▼ [1 - 3 of 3]

```
{
    "_id": {
        "$oid": "583a4b3804ff480004d89980"
    },
    "interval": 3434,
    "density": 34541111,
    "__v": 0
```

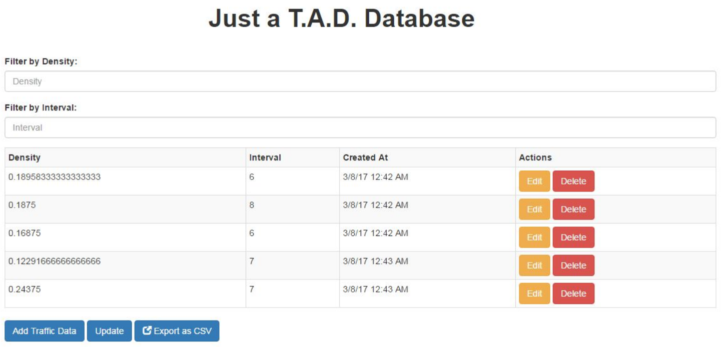
```
{
    "_id": {
        "$oid": "583b9042dcba0f616c8e426e"
    },
    "interval": 2212,
    "density": 2112
}
```

```
{
    "_id": {
        "$oid": "583b9092dcba0f616c8e4280"
    },
    "interval": 1,
    "density": 1000
}
```

records / page 10 ▼ [1 - 3 of 3]

Web App UI

- MEAN (MongoDB, Express, Angular, NodeJS) Stack web application to query database
- Features:
  - Search for specified density/interval
  - Sort by increasing/decreasing order by clicking density, interva or time headers
  - Export data as CSV
  - Automatically update (poll every X seconds, currently 5)
  - Infinite table scrolling



Just a T.A.D. Database

Filter by Density:

Density

Filter by Interval:

Interval

Density	Interval	Created At	Actions
0.18958333333333333	6	3/8/17 12:42 AM	Edit Delete
0.1875	8	3/8/17 12:42 AM	Edit Delete
0.16875	6	3/8/17 12:42 AM	Edit Delete
0.12291666666666666	7	3/8/17 12:43 AM	Edit Delete
0.24375	7	3/8/17 12:43 AM	Edit Delete

3G Dongle

- Acquired 3G subscription
- Installed Drivers
- Fixed Hot Plugging
- Allowed for Device Switching

Current Pricing - 3DR Iris+

Drone	$598
Camera	$15
Raspberry Pi	$50
3G Dongle	$34
3G Subscription	$25
FAA Registration	$5
Raspberry Pi Battery	$20
USB Cord	$5
Total (with drone)	$752
Total (without drone)	$154

Demo

- Image processing on own test images provided by test flight
- Sending of data to server via WiFi
- Server refreshing to display new data within 10 seconds

Thank You!

Questions?