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

Senior Design Project 2017: 
Final Project Review
Meet the Team

▪ Cyril Caparanga (CSE)

▪ Christopher Barbeau (CSE)

▪ Alex Dunyak (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

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

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,

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

Density
(Pixels declared cars over total pixels)

Frame number
Data Server

- Server receives processed data (Density, Interval)
- Displays data on web page
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)
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, interval, or time headers
  - Export data as CSV
  - Automatically update (poll every X seconds, currently 5)
  - Infinite table scrolling
3G Dongle

- Acquired 3G subscription
- Installed Drivers
- Fixed Hot Plugging
- Allowed for Device Switching
### Current Pricing - 3DR Iris+

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drone</td>
<td>$598</td>
</tr>
<tr>
<td>Camera</td>
<td>$15</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>$50</td>
</tr>
<tr>
<td>3G Dongle</td>
<td>$34</td>
</tr>
<tr>
<td>3G Subscription</td>
<td>$25</td>
</tr>
<tr>
<td>FAA Registration</td>
<td>$5</td>
</tr>
<tr>
<td>Raspberry Pi Battery</td>
<td>$20</td>
</tr>
<tr>
<td>USB Cord</td>
<td>$5</td>
</tr>
<tr>
<td><strong>Total (with drone)</strong></td>
<td><strong>$752</strong></td>
</tr>
<tr>
<td><strong>Total (without drone)</strong></td>
<td><strong>$154</strong></td>
</tr>
</tbody>
</table>
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?