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

Senior Design Project 2017: Cumulative Design Review

## Meet the Team

Cyril Caparanga (CSE)



Alex Dunyak (CSE)



Christopher Barbeau (CSE)
Matthew Shin (CSE)





# System Requirements



#### 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 - CDR



# Demonstration of Deliverables



# **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
  - imes Send image data to server via 3G

#### **Image Processing Requirements**

- Detecting Cars from a top down view
- Counting the amount of cars in the picture
- Distinguishing between different lanes of a highway
  - Determining spacing between cars in each lane

# **Eliminating Noise**

- The dimensions of the camera's view are both proportional to the height of the camera.
- At a height of 2m, the camera has a 2m x 1.33 m viewing window, so at 1m it has a 1m x .665m viewing window.
- We can then approximate the number of pixels occupied by one square meter with the following equation:

$$\frac{1}{h^2(0.665*1)}*720\times480$$

# Eliminating Noise (continued)

- The average area of a car is about 5mx10m, or 50 square meters.
- Given this approximation, we can threshold the detected contours by removing contours that are much smaller than cars should be.
- Currently, we prune contours if they are less than 70% of our calculated approximation.

# Eliminating Noise (demonstration)

#### No pruning: After pruning:



# Our motion detection algorithm - High level

- Background subtraction is very sensitive to camera motion
- Overcome this by considering the fact that the drone moves relatively slowly
  - We can find a mapping from each frame to another frame a set amount of time (1/3 or 1/6 of a second) later.
  - By making this mapping, we can find the homography matrix that shifts the earlier frame onto the new frame.
  - The shifted image does not account for movement outside of drone movement, so we can feed it into a standard background subtraction algorithm.

#### Our motion detection algorithm - Cont.

- Background subtracted image can be passed to contour detection, which gives point descriptions of the motion found.
- Using a few more assumptions about the heading of the drone and the compass orientation of the road, we can find the distance between contours

Road Cropping (original)

 Assume we know the heading of the drone and the road



# Road Cropping (revised)

 Similar to original method, but instead of cropping the road by rotating the mask, we rotate the drone so that the camera is oriented along the road lengthwise.





# Interval detection (previous)

- By assuming we know the heading of the drone from the internal compass and the compass direction of the road to a reasonably accurate degree before takeoff, we can find the vector between detected contours, and compare that angle to the expected road angle.
- If the angles are in a cone nearby and within two boundaries at a distance, then consider the number of pixels between the two the interval.

# Interval Detection Diagram (previous)



# Interval detection and density

- After pruning the roads, we can use similar methods to detect density and interval between cars
  - Since the Massachusetts department of transportation lists the width of a lane of a road as between ten and fifteen feet, by assuming the drone is over the road, simply scan across a horizontal line a half-lane width from the center of the screen.

Effective Lane Width 
$$\sim \frac{4.5}{h} \times 480$$



#### Density and interval

#### Density: Per lane, density is

pixels belonging to detected car contours 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,

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

# Database Client

- For any flight, a single pair of density and interval will be generated
- The image processing will have access to a client that will send this data to the database
- Formats the data into a JSON-style document and inserts timestamp
- Allows Python scripts on the Raspberry Pi to write to the database when internet connection exists

#### **Internet Connection**

- Take video at 30 frames per second
- Transmit processed data over 3G to data server



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
- New 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)

#### Just a T.A.D. Database

Density				
Filter by Interval: Interval				
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	Edil Delete	

#### Raspberry Pi- Initial Setup

- Unpacked and Assembled
- Acquired a micro SD card and downloaded raspbian
- Installed Raspbian
- Endured many hours of troubleshooting for a default keyboard layout issue



#### Raspberry Pi- Camera Setup

- Unpacked and Installed
- Enabled Camera in System Config
- Endured many hours of troubleshooting
- Took test photos in the terminal
- Used piCamera to code and automate picture taking
- Understood what this camera is capable of



# 3G Dongle

- Must Acquire 3G subscription
- Install Drivers
- Hot Plugging
- Device Switching
- Sending HTTP Packets
- Dealing with Firewalls
- Test Saving Video over Network(Streaming)



## 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	



# Team Responsibilities and Schedule



# **FDR 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)
- All
  - Assist with final report

# Team Schedule



#### 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?**



SDP17 - **TAD** 33