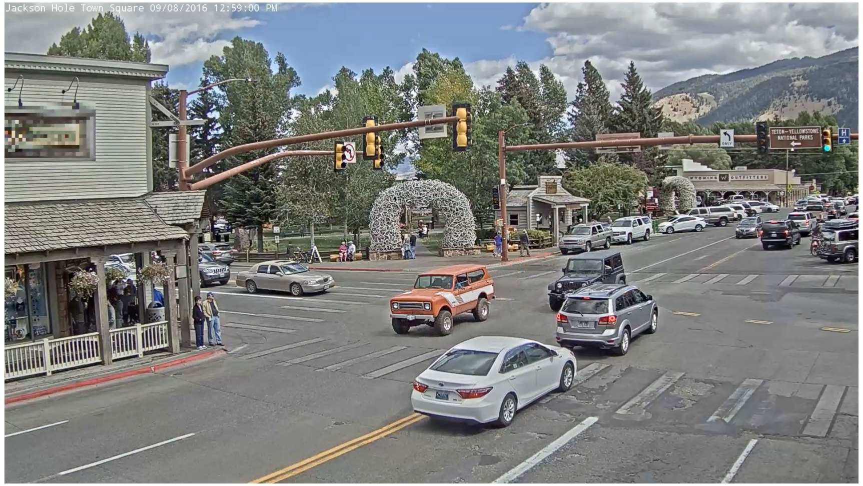
Traffic Camera Perspective Correction

CE-UY 4643-A Robot Vision

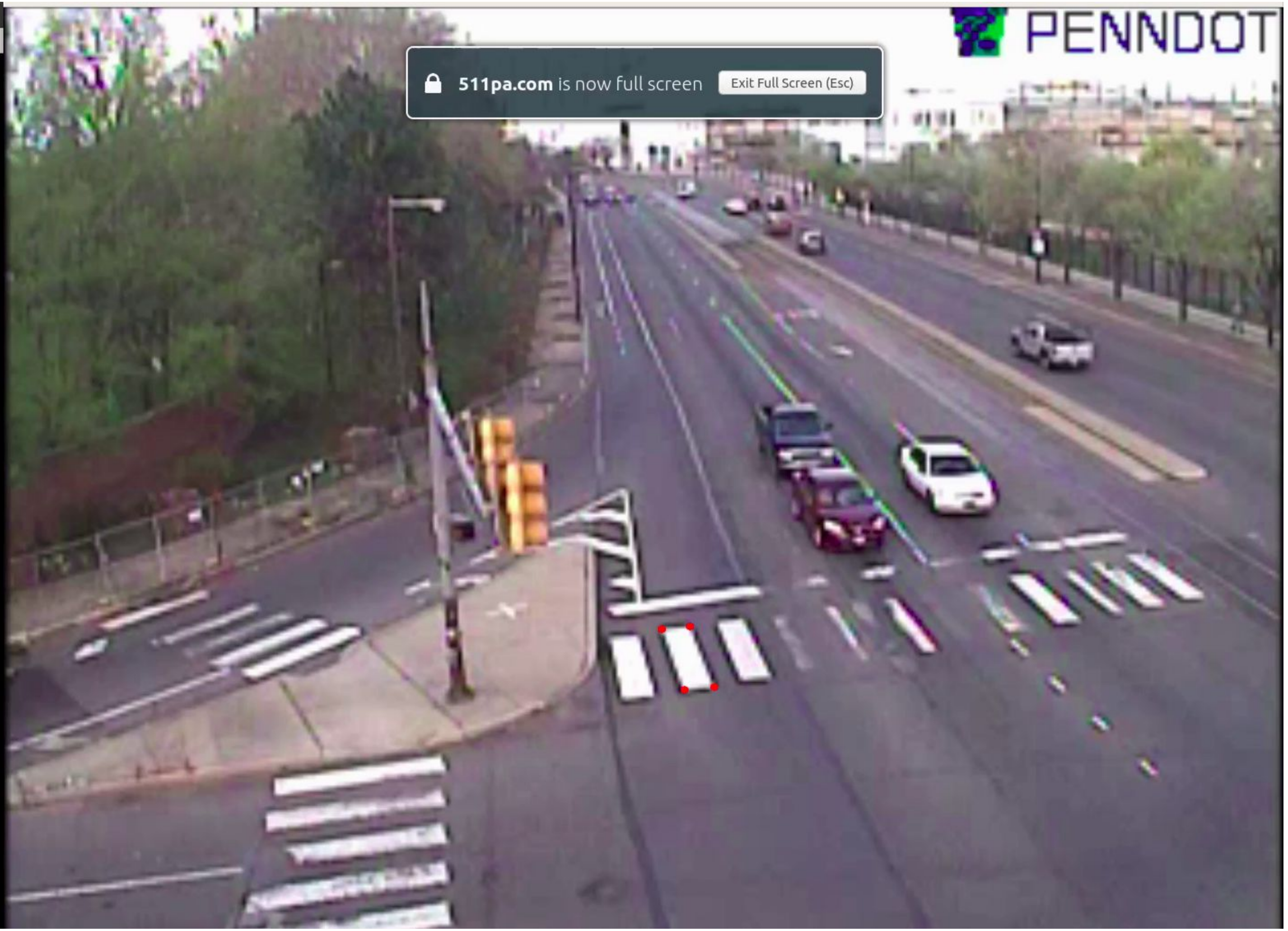
Irvin Tang

Introduction

- Normal traffic camera perspectives don’t provide optimal angles for computers to perform real-time analysis
- Since video streams provide a 2D image, it is hard to make assumptions about the 3D world these videos are representing
- This project aims to take these streams and apply perspective correction such that the new perspective is a top-down view of the current environment being observed


Jackson Hole Town Square 09/08/2016 12:59:00 PM
TETON–YELLOWSTONE NATIONAL PARKS

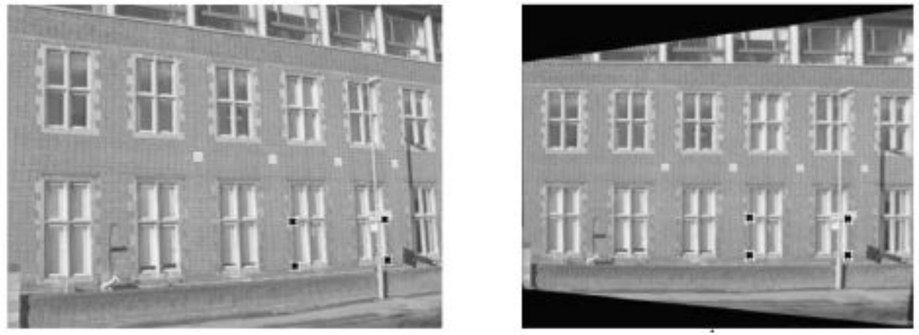
Before we get into the details, demo!


PENNDOT
511pa.com is now full screen
Exit Full Screen (Esc)

Techniques & Methods

Perspective correction:

- Homography transformation is used to perform the perspective correction
- For the specific demo, I used Philadelphia DoT standards(1) to get the aspect ratio of the cross walk markings in the video



Hartley & Zisserman 2011

Object Detection:

- YOLOv3: Real time object detection(2)
- In order to map objects onto the 2D plane, needed to be able to first detect the objects in the video first

Novelty & Improvements

Many projects do a lot of analysis on traffic cameras, but the perspective is never actually changed. The key difference here is that, instead of having just humans analyze the 2D plane, machine learning models can be trained on the movements of the dots in this graph and detect some kind of pattern.

Improvements for the future:

- More robust parallel line detection for homography purposes
- Introducing road markings in the planar representation of the video stream
- Increasing accuracy of object detection

Conclusions/Learnings

I learned:

- What a homography matrix is and what it is used for
- How to use a homography to perform perspective correction
- How to use OpenCV/NumPy for video and image processing
- How to address the issue of OpenCV’s warpPerspective function cropping the warped image

References + Resources Used

References:

1. http://www.dot.state.pa.us/public/pubsforms/publications/pub%20111.pdf
2. https://pjreddie.com/darknet/yolo/

Resources:

1. https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/
2. https://www.learnopencv.com/homography-examples-using-opencv-python-c/