Carnegie Mellon University
The Robotics Institute

Technologies for Safe and Efficient Transportation
A U.S. DOT UNIVERSITY TRANSPORTATION CENTER

Speed Gun App – Increasing Awareness of Urban Speeding

Final Report – November 28, 2018

Principal Investigator: Bernardo R. Pires
ORCID ID: https://orcid.org/0000-0003-0591-4250

Student Collaborators: Christopher Kaffine, Jian Gong, Kylee Santos

Contract No. DTRT12GUTG11

Problem Description

The objective of the Speed Gun App is to empower government officials and transportation advocates. It is well known that speeding, particularly in urban areas, is extremely dangerous to pedestrians and cyclists. However, concerned citizens are often powerless to tackle this problem. A Speed Gun App allows users to obtain the approximate speed of passing cars. Its use is not intended for enforcing, but instead for drawing awareness to localized speeding problems.

Note: The original concept for this project was born at an UTC organized event "Traffic21/BikePgh! Speed Monitoring Hack Night". This project benefited from a significant amount of development effort from both PI and students before receiving T-SET UTC support.

Approach

This project produced a method for calculating the speed of a moving vehicle from a video captured using a mobile device. We created an algorithm for tracking the vehicle's position relative to the camera by following the location of the license plate in the video. As an input to the algorithm, the user identifies the edges of the license plate in one or more frames of the video. The position of the license plate in three-dimensional space is determined by solving for the license plate pose based on point correspondences between the plate's corners and the known dimensions of the plate, using preexisting camera calibration data. Numerical tests were performed to measure the algorithm's sensitivity to noise in the locations of the license plate corners. Finally, this procedure was integrated into an iPhone app, as shown in Figure 1.

FIGURE 1. Screenshot of the developed iPhone app

Introduction

Increased vehicle speed has a disproportionate impact on pedestrian fatalities in crashes. Crashes involving cars moving at 30 mph have a 40% higher chance of causing pedestrian fatalities than cars moving at 20 mph, and at 40 mph the chances increase by another 40% [1]. In order to make intelligent decisions regarding traffic policies to reduce crash fatalities, it's critical for there to be reliable data on travel speeds along major roads. A recent work [2] applied simple computer vision principles to track the speed of traffic along a busy Pittsburgh road in a video taken from a very high vantage point. While this work highlighted the usefulness of such a system, the process is not applicable in more realistic scenarios. It requires user intervention to set distance scales, and only works on video with a top-down perspective, both of which are difficult to obtain in a generic setting.

To build on that work, we propose a more robust system which infers distance scales and camera perspective from a vehicle's license plate. The only user interaction required by the system is the identification of the location of the license plate, a visual task which requires no specialized knowledge. We target smartphones as a platform for the system to enable widespread use and to maximize data collection opportunities, since modern smart-phones have cameras with high enough resolution to make the task feasible. In principle, anyone with a smart-phone could download the app and with a small amount of practice be able to generate videos labeled with vehicle speeds.

Method

We take advantage of the fact that US regulations specify that license plates are all 12″ × 6″ to determine the speed of the vehicle (note that applying the method to foreign license plates, or even non-standard US plates, is straight- forward as long as the plate dimensions are known). Using this information, we can construct a license plate reference frame whose origin is located at the center of the license plate oriented such that the positive z-axis is orthogonal to the license plate, pointing away from the text. We know that the locations of the corners of the plate in this frame are (in inches):

$$L_l = \begin{bmatrix} -6 & -6 & 6 & 6 \\ 3 & -3 & -3 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where the columns of the matrix are the coordinates of the top-left, bottom-left, bottom-right, and top-right corners respectively. Our goal is to identify the position of the license frame in the camera's reference frame. To do this, we will assume we have the homogeneous coordinates of the corners in the image in a matrix L_c and the camera calibration matrix K.

Defining H_c^l as the homogenous transformation from the camera frame to the license frame, we have the following relationship:

$$L_c = K\, H_c^l\, L_l$$

H_c^l has 6 degrees of freedom and the detected license corners in L_c give us 8 constraints, so the problem is over-constrained and we can expect to find an approximate solution. This problem is a specific case of a more general problem in computer vision called pose estimation, for which standard solutions are well known. For this project, we use the pose estimation implementation provided by the OpenCV computer vision library [3].

We calculate the speed of the license plate by finding the pose in two frames separated by some time interval t. We extract the position of the license plate in the camera frame from the last column of the matrix H_c^l for each of these frames to obtain the license plate positions p_0 and p_1. If we assume that the camera is stationary throughout the video, then the average vehicle speed during the time interval t is:

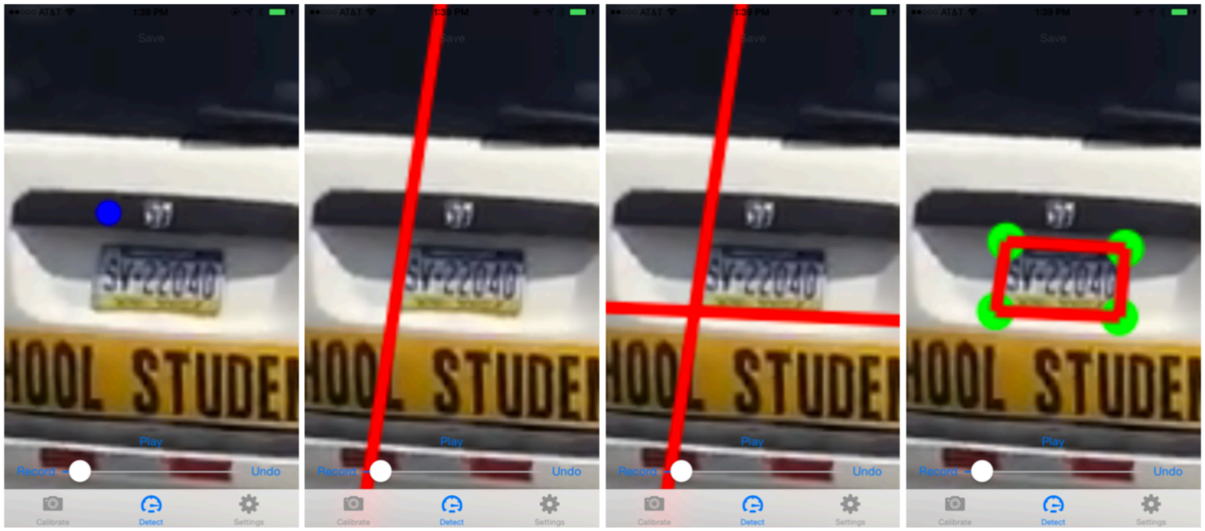
$$s = \frac{\|p_1 - p_0\|}{t}$$

This calculation can be expanded to obtain a time-varying estimate of the speed of a vehicle as it drives thru the scene by repeatedly finding the license plate pose in a series of frames throughout the video.

iPhone Application

The algorithm described above was implemented in an iPhone application, shown in Figure 1 above. The app allows users to take a video of a vehicle driving by, then assist the algorithm by identifying the corners of the vehicle's license plate in several frames. To do this, the user can scrub through the video to find an appropriate frame and zoom in on the license plate. They then touch two points along each edge of the license. The app intersects the lines defined by these pairs of points to find the corners, which are passed into the pose estimation algorithm. The process of selecting these points is shown in Figure 2. After at least two frames have been annotated in this way, the app displays a label with the current speed estimate centered on the license plate, as shown in Figure 3. This label is animated so that it tracks the current estimate of the license plate location at any point in the video, and can serve as an indicator of how good the estimate is.

More than 10 hours of video where collected in the vicinity of CMU's campus to assist in app development, quality control, UI refinement, and verification of vehicle speed estimation accuracy.



(a) User selects one point along an edge

(b) User selects another point, completing the edge

(c) User repeats the process for the next edge

(d) After four edges are specified, the corners are found

FIGURE 2. User interface used to specify license plate locations.

FIGURE 3. Frame of output video showing the vehicle's speed animated over its license plate.

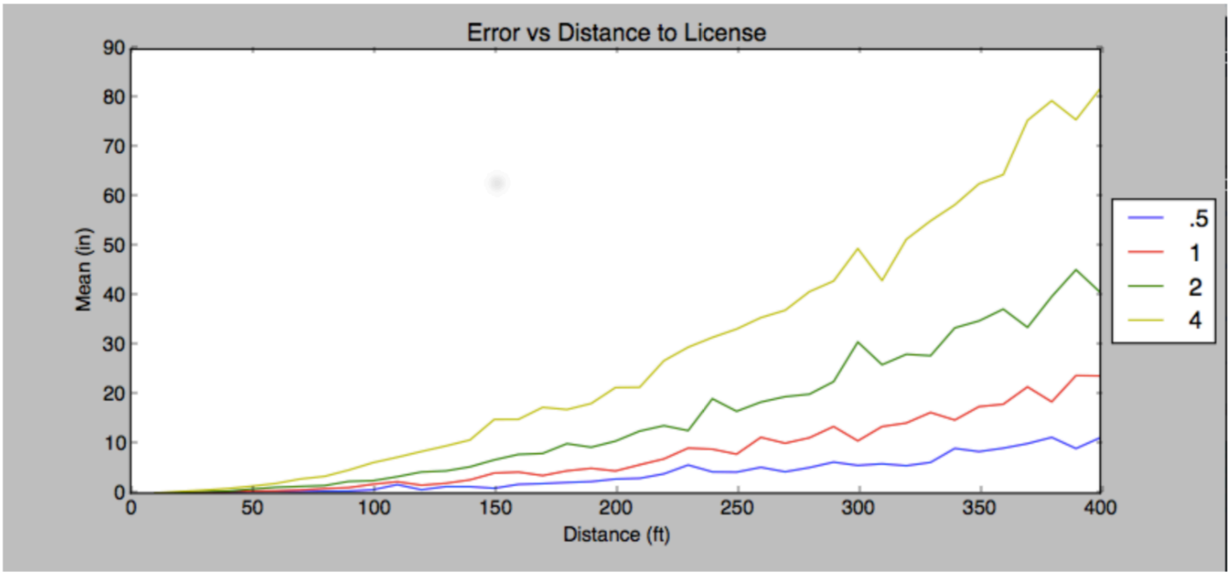


FIGURE 4. Computed mean error as a function of distance to the license plate.

Experimental Results

In order for the app to be useful, it needs to estimate speeds sufficiently reliably that it can differentiate between vehicles that are speeding and those that are under the speed limit. A major obstacle to this reliability in our approach is inaccuracy in the corner locations provided by the user. Specifically, it is difficult for users to identify the corners to within less than 2 pixels of accuracy. To see what effects this error has on the results, we ran simulations adding noise to correct corner estimates.

For each test, we constructed a rigid transformation H_c^l setting the license between 10″ and 400″ away from the camera with a random orientation. This specific distance range was chosen to be representative of distances expected in actual use. Similarly, we restricted the possible orientations to be composed of rotations in roll, pitch and yaw with magnitudes between $\pm\pi/4$ radians, since we expect users to attempt to align the camera with the license plate. We then performed the projection $K\ H_c^l\ L_l$ to get the simulated corner locations. We next added an offset to each of the corners generated from a Gaussian distribution with standard deviation of .5, 1, 2, or 4 pixels. Finally, we performed the pose estimation using the altered corner locations to get an imprecise estimate of the license location. The distance between this estimate and the known actual location based on H_c^l is the error for that experiment.

Our results are shown in Figure 4, plotted over distance to show how accuracy trails off as the vehicle gets further away. They show that when the corner estimates are accurate to within a single pixel, the error in license location remains under 2 ft even when the vehicle is as much as 400ft away. More realistically, if the accuracy is within 2 pixels the error stays under 3ft at a distance of 400ft. With even lower accuracy at 4 pixels, the error unsurprisingly increases dramatically, rising to almost 7ft. To put these numbers in context, if a user annotates two frames that are 1 second apart and there is an error of 3ft in the location estimate for both frames, the difference may be off by ±6ft. This would cause an error of ±4mph in the speed estimate. With frames that are 2 seconds apart this becomes ±2mph, and so on. Based on this analysis, our approach does seem to be sufficiently accurate to identify vehicles that are speeding by more than 5mph.

Conclusions and Future Work

This project serves as a proof of concept encouraging further work in this research area. There are a number of direction future research could take, most notably in better automation of the license detection process. A system that could track the license plate after detecting it in one frame could allow pose estimation to be done in the frames in between user-assisted detections, allowing a continuous estimate of speed throughout the video. Similarly, better detection algorithms could eliminate the need for user assistance entirely and allow the whole process to be automated. Another interesting goal would be to eliminate the assumption that the user is holding the camera still. The inclusion of accelerometer and gyroscope data available on mobile phones along with visual odometry through tracking optical features in the scene could provide a model of the movement of the camera frame in the global reference frame, which could be factored into velocity calculations to remove the contribution of camera movement.

References

[1] Effects of vehicle speed on pedestrian fatalities. https://github.com/mbauman/TrafficSpeed.

[2] M. Bauman. Guerrilla traffic speed monitoring to inform and push for change. https://github.com/mbauman/ TrafficSpeed, 2015.

[3] G. Bradski. The opencv library. Dr. Dobb's Journal of Software Tools, 2000.