EE368/CS232 Digital Image Processing Project Proposal

Mobility Analysis of Vehicle Front View Camera Videos

Yaqi Zhang (yaqiz@stanford.edu)

Background

With increasing technology to improve driving security, surrounding camera is increasing popular among recent models of family using vehicles. With abundant information collected by these cameras, there are few existing techniques that can automatically analyze and understand the content of recording and providing valuable information about behavior of driver and driving conditions. This project explores the possibility of using combination of image processing and computer vision techniques to extract information about the driving behavior. The project is collaborated with a Phd in computer vision research group at Stanford. No Android device is needed for the project. The data used in this project is collected in urban street in China. Along with the videos, there are also sensor data for acceleration of x,y,z directions for training of the model.

Goal

Ultimate goal of the project including the followings:

- Understand driving condition of the car, including road types (urban/highway), pedestrian detection, surrounding vehicles detection, road sign recognition, etc.
- Infer vehicle state from the video, such as speed, distance, acceleration/deceleration, turning, breaking, etc.
- Construct causality relationship between the environment the vehicle is experiencing and the driver's behavior. For example, understand the reason for a decision, such as acceleration/turning is caused by presence of an obstacle.

The short term goal of this class project majorly focus on the image processing component. For speed analyze, due to pretense of many interfering objects moving at their own speeds, it is challenge to find still reference object to analyze absolute speed of the vehicle. With motion detection, we can acquire flow information of the video, which can be then used to analyze moving and still objects in the frame. The difference between flow of left half and right half scene can also be used to study the turning angle of the vehicle. With plate recognition, we can

pin point the location of the other cars in the frame, along with flow distribution adjacent to the plate (should be identical to flow of the plate if it's part of the same car), it is possible to identify and segment out the moving pixels of the interfering car and leave only the still references and forming a clean base for later machine learning on the video. Additionally, image processing can be used to detect road signs along the street to help understanding the environment of the vehicle. Due to the video is taken inside front window of the camera, there are reflection of the glass inconsistently appears in the video, which can be removed with image preprocessing.

Related Work

There are existing studies on using videos taken by highway surveillance camera for speed detection [1]. Our problem is more challenging as the video is taken inside a moving vehicle and many occasions, obstacles such as vehicle and pedestrian occupies a large region of the scene. There are many other existing techniques and libraries, such as road sign recognition of an image [2] [3] and license plate recognition [4], optical flow detection [5], TenserFlow[6] that are relevant for this project.

Reference

[1] Ibrahim, Osman, Hazem ElGendy, and Ahmed M. ElShafee. "Speed Detection Camera System using Image ProcessingTechniques on Video Streams." International journal of computer and electrical engineering 3.6 (2011): 771.

[2] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno and F. Lopez-Ferreras, "Road-Sign Detection and Recognition Based on Support Vector Machines," in IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 2, pp. 264-278, June 2007.
doi: 10.1109/TITS.2007.895311

[3] L. Estevez and N. Kehtarnavaz, "A real-time histographic approach to road sign recognition," Image Analysis and Interpretation, 1996., Proceedings of the IEEE Southwest Symposium on, San Antonio, TX, 1996, pp. 95-100.
doi: 10.1109/IAI.1996.493734

[4] https://github.com/openalpr/openalpr

[5] http://docs.opencv.org/trunk/d7/d8b/tutorial_py_lucas_kanade.html

[6] https://www.tensorflow.org/versions/r0.11/tutorials/index.html