City Project Summary

The traffic flow in a city is a rich source of information about the city. For instance, the real-time traffic conditions can be used for suggesting drivers the optimal routes, while retrospective data can be utilized for planning road construction and public transport routes.

We design a system to build a traffic map in a city via analyzing traffic flowing at a certain set of locations, where optical cameras are installed and image data is made publicly available. Though the scientific community definitely benefits from access to these data, the number of cameras as well as their location and the quality of image data is unfortunately out of control of researchers and as such has to be considered as problem parameters. In particular, we are currently using 450 cameras located in NYC area. Instead of video, image snapshots are available through a public web-interface of NYC Department of Transportation. The size of camera image varies from about 300x400 to 800x500 for different cameras. The image quality depends on the weather and light conditions, and is generally poor. Most importantly, images are updated on the server with the variable rate of 1 to 3 seconds. In many cameras a single car can rarely be seen in more than 3 frames due to such a small frame rate.

The system is designed to work with all cameras independently. For every camera we consider the problem of computing the total number of cars that drive past a camera in a given time interval. In order to avoid over counting, cars detected in multiple frames need to be identified as a single entity. Since the system is intended to work in the real-time, the processing speed for every frame is limited.

Optical system for analyzing traffic generally detect and track cars from a video input. However, in our case tracking-based approaches are not applicable because of very low framerate. Instead we combine car detections from different frames in a probabilistic model that uses car appearance information and hints from scene geometry to identify same cars. To be specific, our approach includes four separate components: the background subtraction, the scene geometry, the car detector, and the probabilistic counting model (fig. 1)

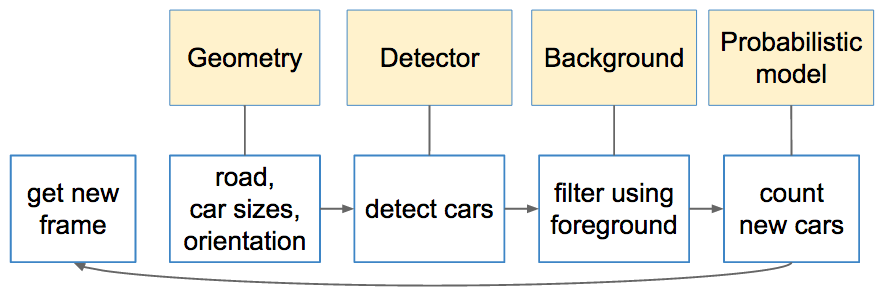


Fig. 1

First, we subtract the background based on a set of previous frames. Last several frames are used to build and update Gaussian Mixture Models separately for every pixel and thus separate moving objects from the background. The foreground mask is further refined to reduce noise and achieve a near-100% recall rate.

Understanding the geometry of the scene forms the second part of the pipeline. Since the position of every camera does not change over time, the scene geometry can be learned and exploited to improve the system robustness and accuracy. In particular, taking in account information about scene perspective helps to predict the expected size of a car at every particular point in the image and the mutual probability for a car to be detected in certain locations in two sequential frames. the scene geometry can be initially learned from clues like vanishing points and lane markings, and further refined using detected car trajectories.

The system further uses a Viola-Jones boosted cascade to detect cars in every frame. Information from background subtraction and scene geometry is used to filter improbable detections. The cascade model was initially trained from manually labelled data and existing car datasets, and can be supplemented with information from detected cars. Figure 2 depicts an example of filtered detections.

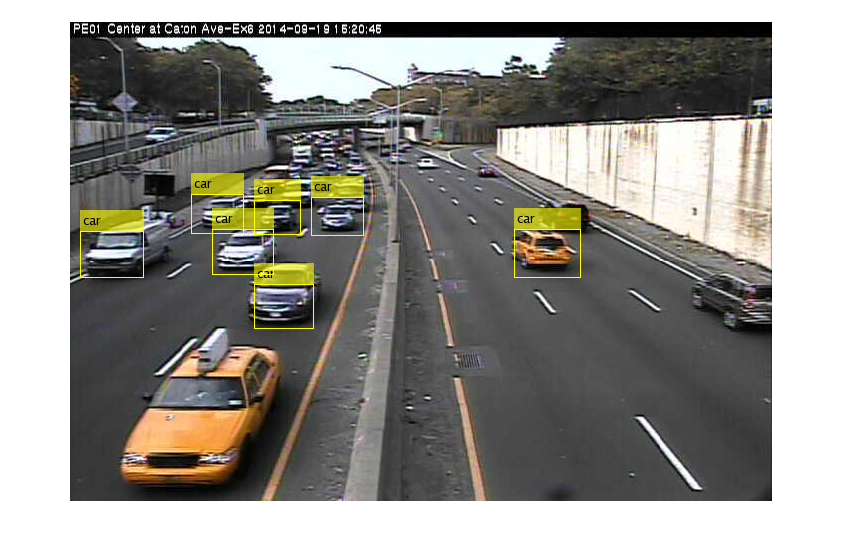


Fig. 2

Finally, based on the car patches detected in the above steps, we find identical cars in pairs of sequential frames. Geometric constraints on location of a car in two frames and generated appearance-based features are used to build the transition probability matrix for every pair of frames. Car correspondences are then implied from this matrix.

The future work includes unsupervised learning of scene geometry and the cascade detector from camera data, and building a more powerful probabilistic model.

1. 1. Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*. Vol. 2. IEEE, 1999.
2. D. Hoiem, A.A. Efros, and M. Hebert, "Geometric Context from a Single Image", *International Conference on Computer Vision,* 2005.
3. Viola, Paul and Michael J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001. Volume: 1, pp.511–518.