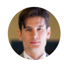
Vision-based vehicle queue analysis at junctions

Conference Paper · August 2015

DOI: 10.1109/AVSS.2015.7301732

2 authors:

Mohammad Shokrolah Shirazi
University of Indianapolis



Brendan Morris
University of Nevada, Las Vegas

Vision-based Vehicle Queue Analysis at Junctions

Mohammad Shokrolah Shirazi[1] and Brendan Morris[1*]

Abstract

This paper presents a tracking method for vision-based queue analysis at junctions including queue length and waiting time estimation of vehicles. The tracking method works based on improving the optical flow method to track small and low quality vehicles with overlong waiting time. The improvement process is performed to keep track of overlong stopped vehicles with high level of robustness against occlusion by crossing pedestrians. The results of experiments are presented by estimating queue length, waiting time distribution and number of waiting vehicles for the highly cluttered video of a Las Vegas junction. The accuracy of the system is evaluated by comparing the queue analysis results with the ground truth.

1. Introduction

Vehicular traffic data such as flow, speed and density is an important criterion used to design intersections and boost safety. Another important parameter is the vehicle queue analysis used in control models to improve the passing capacity. Moreover, queue length estimation and associated delay are useful for devising traffic management strategies that would help to optimize traffic signals and improve the performance of a traffic network.

Data collection is the first step towards queue length estimation, which is usually performed by the loop detectors [5] and video cameras [6][1] at junctions. Manual methods require human observation to collect data which is difficult for long time evaluation. Vision-based methods are the one of the most preferred automatic methods due to their low cost and high support of different real time applications (e.g. vehicle count, speed and classification). Long time data collection of traffic flow with high speed is an essential requirement in Intelligent Transportation Systems (ITS).

Vision-based queue analysis can be performed by two major groups of tracking and non-tracking methods. Non-tracking methods determine the existence of vehicles based on different introduced features on the road. Local Binary Pattern (LBP) [6], spatial edges [8], FFT [8], and image gradients [21] are the important features used to detect stopped vehicles in the literature. As an example of other features, the entropy method is proposed [20] to detect stopped vehicles and Harris corner features [1] are useful in detecting stopped vehicles when they build a queue.

Although queue analysis by non-tracking methods is simple, its application is limited to queue length estimation. However, tracking methods can provide other important vehicular traffic data such as speed, count, waiting time and time headway. A tracking method is appropriate for queue analysis if it addresses the two below problems.

1. Stopped vehicle detection is difficult since motion is mostly used as a cue in video surveillance [14]. Moreover, detection by motion leads to occlusion for slow moving vehicles in the line.

2. Tracking of stopped vehicles is a difficult task since motion-based tracking methods like optical flow work poorly for stationary objects.

Tracking methods rarely address queue analysis. For example, tracking of vehicles on the head and tail position of the line is estimated in [22] and other techniques aim to detect and track stopped vehicles [18][4] like parked vehicles [2]. Tracking methods based on optical flow are presented in [19] and [13] to provide robust vehicle tracking at intersections. However, there is no effort towards estimating queue length and waiting time of separate vehicles in a line.

This paper presents a tracking method for queue analysis by estimating queue length, vehicle numbers in the queue and their associated delays (i.e. waiting times). The proposed method solves formerly mentioned problems since detection by motion is only performed over the small areas which initializes tracks. Then, optical flow tracking handles the partial occlusion between slow moving vehicles in the line. When the waiting state of a vehicle is determined, an appropriate bounding box estimation is performed to handle the possible failure of tracking by optical flow due to reduction of feature points. The tracking method is explained in more details in the rest of the paper.

The paper is organized as follows. Section 2 explains the detection and tracking system and Section 3 shows the

*[1]M. S. Shirazi and B. Morris are with the Electrical and Computer Engineering Department, University of Nevada, Las Vegas, NV 89154, USA shirazi@unlv.nevada.edu, brendan.morris@unlv.edu

Figure 1: Motion area and typical paths

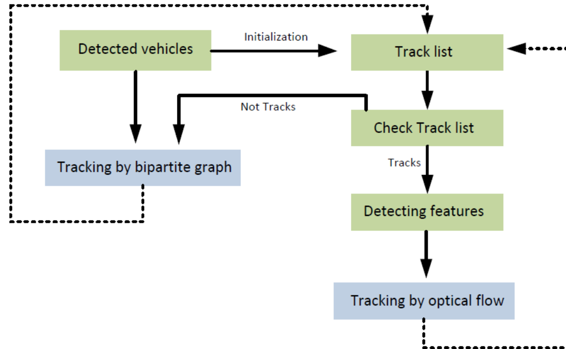


Figure 2: Vehicle Tracking System

essential improvements needed to be addressed for queue analysis. Section 4 presents the queue length estimation process and experimental results are presented in Section 5. Finally, Section 6 concludes the paper.

2. Vehicle Detection & Tracking System

2.1. Vehicle detection

Vehicle detection by motion is a common method used in video surveillance [14] and it is usually conducted using optical flow and background subtraction. Optical flow finds the displacement of moving objects between each two frames [10] while background subtraction estimates adaptive background from the sequence of images. Background subtraction is fast but optical flow computation of features for whole frame is computationally expensive.

This vehicle detection system uses Gaussian mixture model (GMM) [17] to create the adaptive background model for background subtraction method. Moving vehicles are detected in the motion area to initialize the tracks (See Figure 1). Since detection by motion is prone to occlusion or blob merging [14], the motion area is placed in a location in which vehicles do not usually stop. In addition, motion area should be close to the camera as it ensures stable moving objects (i.e. blobs) obtained by background subtraction.

2.2. Vehicle tracking

Vehicle tracking system benefits cooperation of bipartite graph with optical flow to track detected vehicles. The vehicle tracking system is shown in Figure 2. Detected vehicles in the motion area are given to the tracking system which uses bipartite graph at first to track detected vehicles for 3 frames and initialize the tracks. The initialized tracks use optical flow during the tracking phase.

The optical flow methods rely on updating and grouping features based on direction and displacement of motion vectors. Grouping features of nearby moving vehicles with the same speed and direction is difficult. The initialized tracks provided by bipartite graph helps to group features and solve this problem.

2.2.1 Optical flow

Optical flow is a default tracking method used for initialized tracks since it is robust toward partial occlusion. Optical flow is successful when brightness constancy and small motion exists between a same object of two frames. The detected features inside the bounding box area of tracks find their matches on the next frame using optical flow tracker. Since optical flow is limited to features of tracks, the computational cost is reduced a lot.

2.2.2 Bipartite graph

A greedy approach is used to find a nearest detection using bipartite graph matching [7]. The nodes of the graph are tracks and frame's detections and cost of each two nodes' edge is the difference between blob measurements such as location, area, perimeter, eccentricity and orientation [15]. When a detection does not find any match for the existing tracks in the track list , a new track is created. If an existing track does not find a detection, it is marked for deletion.

3. Improvements

3.1. Enriching features

The problem of feature points reduction is prevalent for small size objects with low quality. This worsens during the optical flow tracking for stopped or slow moving vehicles. The idea is to sample each vehicle with more feature points, called enriching feature points, to tackle this problem.

Although Harris corner features have shown good performance [1] for high quality images, their performance is reduced for small and low quality image samples. Harris [9]

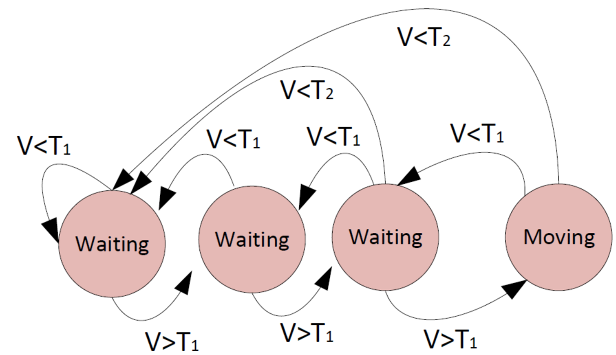


Figure 3: The diagram of waiting/moving state estimation

only detects robust corners but it suffers from a lack of connectivity for feature points to describe surfaces and objects. FAST [12] features are also accumulated since detection of corners is prioritized over edges leading to complementary feature points. FAST actually modified the Harris detector to decrease the computational time of the algorithm without compromising the results.

Since FAST and Harris rely on corners and nearby edges, a texture based feature detection can also be used to enrich the detected features of the vehicle. As a result, Speeded-Up Robust Features (SURF) [3] are accumulated as representative of texture based features. SURF is used in this work since it is the fast version of scale invariant feature transform (SIFT) [11] without sacrificing the quality of the detected points. So, its robustness against scale and orientation will result in successful tracking of vehicles under perspective scenario.

3.2. Filtering features

Filtering is a process of rectifying feature points by removing falsely detected features inside the track. Since, texture based features are also used, there is higher possibility of detecting static features around the moving features regarding to vehicles. These noisy features, lead to false bounding box estimation which causes a track to drift and move away from the vehicle gradually.

The filtering process is performed by determining the state of the vehicle and filtering the opposing features [4]. For example, when a vehicle state is waiting, the static features are predominant and moving features are removed. The way of determining the state of the vehicle is based on the average of the displacement vector (V). The state based diagram is considered regarding each track to handle the noisy speed measurements which are unavoidable in tracking context.

The moving/waiting state detection [16] is shown in Figure 3. Two displacement thresholds T_1 and T_2 are considered as soft and hard thresholds to determine waiting state of a track ($T_1 > T_2 > 0$). When the sequence of "waiting" state happens for a track, its state doesn't change to "moving" by one displacement measurement (e.g. $V > T_1$), which could be noisy. The very small value of the displacement vector ($V < T_2$) implies the waiting state of the vehicle that only can accept transition to moving state by sequence of displacement vectors higher than the soft threshold ($V > T_1$).

3.3. Bounding box estimation

The likelihood of successful tracking is determined based on the quality of the detected matches and the estimated bounding box around the features. The bounding box is estimated based on spatial distribution of feature points that have found their match and passed the filtering process.

Bounding box estimation is a crucial part of the tracking process when vehicles are creating the queue line. The incorrect bounding box estimation for slow moving or stopped vehicles lead to drift of bounding box and tracking confusion. As a result, the fixed bounding box around the fixed position is leveraged when the waiting state is determined. This helps significantly in maintaining a track since feature points are reduced by optical flow when the vehicle is stationary.

The smaller bounding box for vehicles closer to the stop bar is preferred since there is large possibility of occlusion with crossing pedestrians. Pedestrians introduce feature points that lead to false bounding box estimation and losing the track.

4. Queue Analysis

Queue analysis includes estimating the number of vehicles in the queue, their waiting time and queue length estimation. Since a tracking method is used, vehicles' waiting time are readily estimated. Three steps are needed to perform, explained below.

4.1. Path recognition

Vehicles paths are recognized using temporal alignment techniques for similarity measures of trajectories with typical paths which have been collected for each lane (see Figure 1). Longest common subsequence (LCSS) distance is a popular technique for comparing unequal length trajectories [15][16]. The LCSS distance is utilized in this work since it is robust against noise and it has shown good performance for turning recognition [15]. The LCSS distance is computed as

$$D_{LCSS}(F_i^{T_i}, F_j^{T_j}) = 1 - \frac{LCSS(F_i^{T_i}, F_j^{T_j})}{min(T_i, T_j)} \qquad (2)$$

$$LCSS(F_i, F_j) = \begin{cases} 0 & T_i = 0 | T_j = 0 \\ 1 + LCSS(F_i^{T_i - 1}, F_j^{T_j - 1}) & d_E(f_{T_i}, f_{T_j} < \epsilon) \& |T_i - T_j| < \delta \\ \max(LCSS(F_i^{T_i - 1}, F_j^{T_j}), LCSS(F_i^{T}, F_j^{T_j - 1})) & \text{otherwise} \end{cases} \quad (1)$$

where T_i is the length of trajectory F_i. The LCSS is defined in (1) and provides the number of matching points between two trajectories. $F^t = \{f_1, ..., f_t\}$ denotes the trajectory centroid up to time t. ϵ and δ are two major parameters that are empirically chosen based on an intersection settings. ϵ is used to compare and find matching points within a small Euclidean distance and δ is a temporal constraint to ensure that lengths are comparable and meaningful.

The path recognition is performed by comparing the observed trajectory with all the stored typical paths of vehicles at the intersection. The path with the smallest D_{LCSS} value is considered the best match and its label is applied to the vehicle trajectory.

4.2. Waiting state detection

Tracks are labeled regarding each lane and their moving or waiting state is determined at each frame based on the state diagram shown in Figure 3. When the absolute waiting state of vehicles is determined, they become candidates for queue length estimation if there is spatial proximity between them. The track candidates are saved into separate lists regarding each lane for queue length estimation.

4.3. Queue length estimation

Queue length is gauged for waiting vehicles of each lane using feature points of the tracks. Texture and corner feature points of stopped vehicles are used to estimate the queue length. Since vehicles' queue lines might have different orientations, the line is projected into each lane by selecting (x, y) coordinates of feature points according to highest and least y values. Two selected feature points find the nearest neighbor coordinate from typical paths and the Euclidean distance is used to measure the distance.

Figure 4 shows an example of the queue length estimation using feature points. The estimated lines (red line) are shown for two queues with different orientations. As shown in Figure 4, one vehicle has missed the track since it has not come into the motion area to get initialized from the beginning. Note that, the bounding box of the vehicles closer to the stop bar (second and third lanes) are smaller since there is possibility of occlusion with crossing pedestrians, as discussed in Section 3.3.

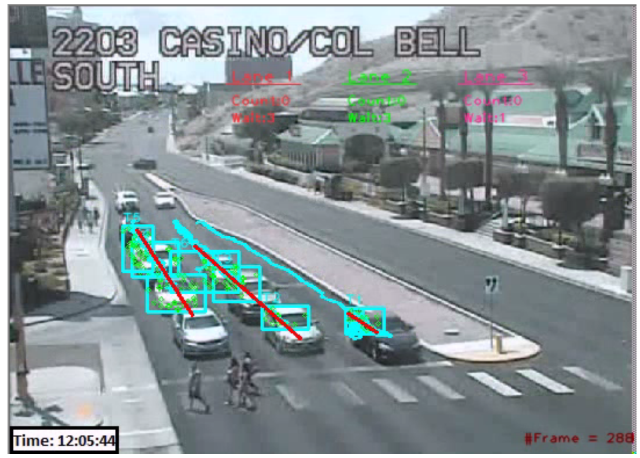


Figure 4: The queue length estimation (red line), detected stopped vehicles (aqua bounding box) and feature points (green points)

5. Experimental Results

The proposed system was implemented in C++ using OpenCV 2.3 and it was run on quad core Intel i7 processor with 6 GB RAM. The proposed system was evaluated for a highly cluttered video of the Las Vegas junction. The experiments included evaluation of the system for queue length estimation, count of waiting vehicles in queue and waiting time estimation of vehicles in regards to different lanes.

The queue length and number of waiting vehicles are estimated and compared against the ground truth for 800 frames (i.e. 8 frames per second) of the junction video (See Figures 5, 6). The ground truth is created based on manual observation of each frame. Figure 5 shows that the estimated queue length follows the ground truth with small offset due to projection error. The average of absolute error (Ae) is used in our evaluations and it is defined in (3)

$$E(Ae) = \sum_{i=1}^{N} \frac{\left| H_i - \hat{H}_i \right|}{N} \quad (3)$$

where H_i is the manually extracted queue length, $\hat{H}_i$ is the estimated queue length and N refers to total time (seconds). The average of absolute error was 0.42 meter for $N = 100$ which indicates the effectiveness of the proposed method for estimating queue length. The only offset shown in Figure 6, manifested as big gap in Figure 5 (seconds 42-48) due

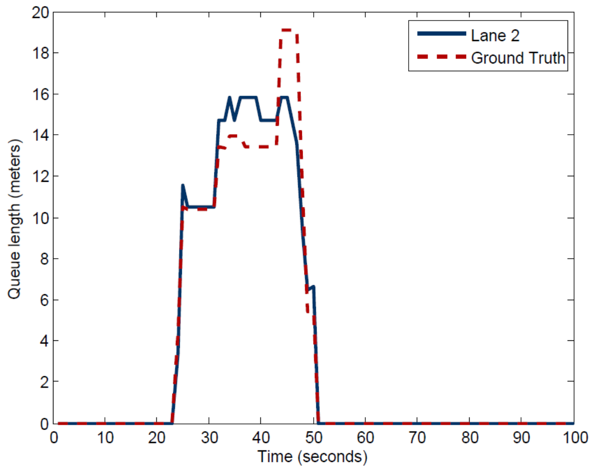


Figure 5: Queue length estimation versus the ground truth (lane 2)

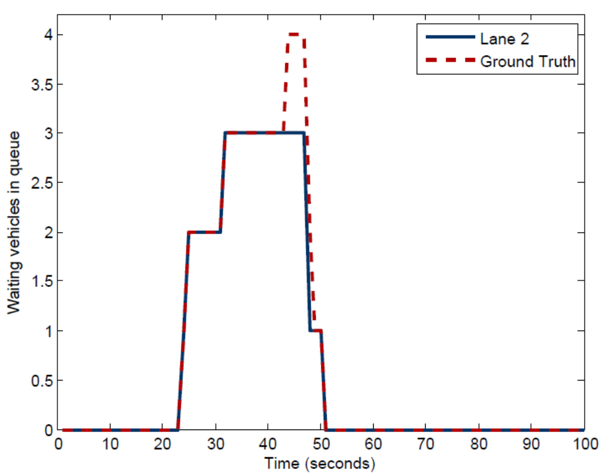


Figure 6: Waiting vehicles in queue versus the ground truth (lane 2)

to losing the track of the last vehicle.

Figure 7 shows the queue length estimation and Figure 8 depicts the corresponding number of waiting vehicles regarding two different lanes. These figures show a similar pattern and traffic signal phases can be inferred from them. Different queue lengths for the same number of waiting vehicles are shown (e.g. 24-50 seconds) due to tracking error and different sizes of waiting vehicles in lane 1.

Figure 9 shows the waiting time distribution of the vehicles regarding two different lanes. There is a higher average of waiting time for lane 1 since it is a more congested path. As shown in Figure 9, there are some vehicles with waiting times of more than 40 seconds since they reach to junction at the beginning of the red phase signal. Moreover, more vehicles lead to higher queue length which results in more

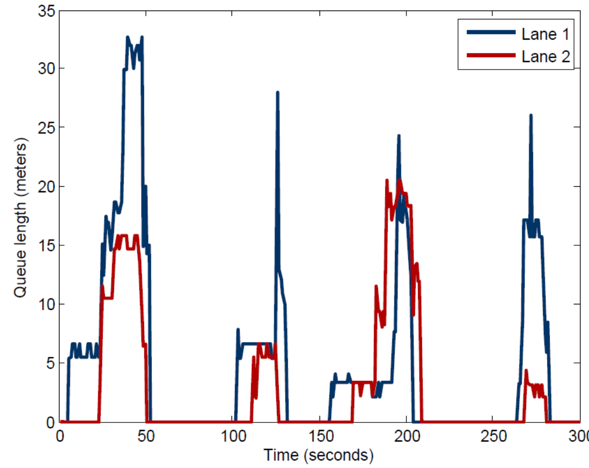


Figure 7: Queue length estimation

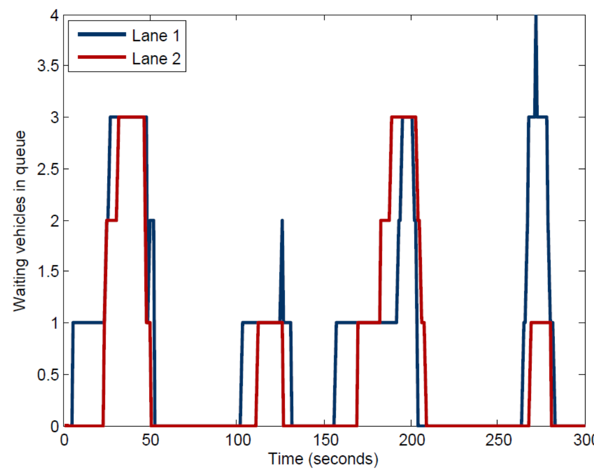


Figure 8: Waiting vehicles in queue

waiting time of the last vehicles.

6. Conclusion & Future Work

This paper presents a vision-based tracking method for queue length and waiting time estimation of vehicles. The tracking method works based on optical flow and undergoes improvements to handle some difficult tracking scenarios such as low number of feature points for small, low quality vehicles and losing track of long time stationary vehicles. The proposed tracking method can still be improved if there is prior knowledge about the bounding box estimation of individual vehicles as complementary data. Since track might drift over the stopped vehicle, the nearby corner features of vehicles outside tracks could help to estimate the queue length more accurately.

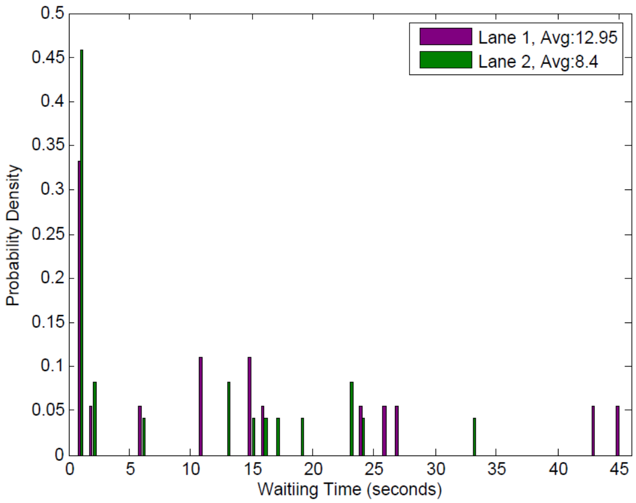


Figure 9: Waitining time of vehicles in lanes 1 and 2

References

[1] A. Albiol, A. Albiol, and J. Mossi. Video-based traffic queue length estimation. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1928–1932, Nov 2011.

[2] A. Albiol, L. Sanchis, A. Albiol, and J. Mossi. Detection of parked vehicles using spatiotemporal maps. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1277–1291, Dec 2011.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[4] A. Bevilacqua and S. Vaccari. Real time detection of stopped vehicles in traffic scenes. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 266–270, Sept 2007.

[5] J. J. Bezuidenhout, P. Ranjitkar, and R. Dunn. Estimating queue length at signalized intersections from single loop detector data. In *Proceedings of the Eastern Asia Society for Transportation Studies*, 2013.

[6] Q. Chai, C. Cheng, C. Liu, and H. Chen. Vehicle queue length measurement based on a modified local variance and lbp. In *9th International Conference, ICIC 2013*, pages 123–128, Nanning, China, July 2013.

[7] W. Fang, Y. Zhao, Y. Yuan, and K. Liu. Real-time multiple vehicles tracking with occlusion handling. In *Proc. International Conference on Image and Graphics*, pages 667–672, San Juan, Puerto Rico, August 2011.

[8] M. Fathy and M. Siyal. Real-time image processing approach to measure traffic queue parameters. *Vision, Image and Signal Processing, IEE Proceedings -*, 142(5):297–303, Oct 1995.

[9] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[10] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17(1-3).

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2).

[12] M. H. Miroslav Trajkovic. Fast corner detection. *Image and Vision Computing*, 16.

[13] N. Saunier and T. Sayed. A feature-based tracking algorithm for vehicles in intersections. In *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, pages 59–59, June 2006.

[14] M. S. Shirazi and B. Morris. Contextual combination of appearance and motion for intersection videos with vehicles and pedestrians. In *10th International Symposium on Visual Computing, ISVC 2014*, pages 708–717, Las Vegas, USA, December 2014.

[15] M. S. Shirazi and B. Morris. Vision-based turning movement counting at intersections by cooperating zone and trajectory comparison modules. In *Proceeding 17th International IEEE Conference on Intelligent Transportation Systems*, pages 3100–3105, Qingdao, China, October 2014.

[16] M. S. Shirazi and B. Morris. A typical video-based framework for counting, behavior and safety analysis at intersections. In *to be appeared in In Proc. of IEEE Intelligent Vehicles Symposium*, Seoul, Korea, June 2015.

[17] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 246–252, June 1999.

[18] P. Venetianer, Z. Zhang, W. Yin, and A. Lipton. Stationary target detection using the objectvideo surveillance system. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 242–247, Sept 2007.

[19] W. Wang, T. Gee, J. Price, and H. Qi. Real time multi-vehicle tracking and counting at intersections from a fisheye camera. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 17–24, Jan 2015.

[20] C. Yingfeng, Z. Weigong, and W. Hai. Measurement of vehicle queue length based on video processing in intelligent traffic signal control system. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 2, pages 615–618, March 2010.

[21] M. Zanin, S. Messelodi, and C. Modena. An efficient vehicle queue detection system based on image processing. In *Image Analysis and Processing, 2003.Proceedings. 12th International Conference on*, pages 232–237, Sept 2003.

[22] J. Zhou and L. Cheng. *An Efficient Vehicle Queue and Dissipation Detection Algorithm Based on Spatial-Temporal Markov Random Field*, chapter 49, pages 468–474.