Lane Discovery in Traffic Video

Nicholas Miller
David M. Swart
Akshaya Mishra
Andrew Achkar

Miovision
Kitchener, ON
Canada

Abstract

Video sensing has become very important in Intelligent Transportation Systems (ITS) due to its relative low cost and non-invasive deployment. An effective ITS requires detailed traffic information, including vehicle volume counts for each lane in surveillance video of a highway or an intersection. The multiple-target, vehicle-tracking and counting problem is most reliably solved in a reduced space defined by the constraints of the vehicles driving within lanes. This requires lanes to be pre-specified. An off-line pre-processing method is presented which automatically discovers traffic lanes from vehicle motion in uncalibrated video from a stationary camera. A moving vehicle density map is constructed, then multiple lane curves are fitted. Traffic lanes are found without relying on possibly noisy tracked vehicle trajectories.

Fig. 1: Example traffic video. Three lanes (blue) are discovered from moving vehicles. Vehicle density is also overlaid (red).

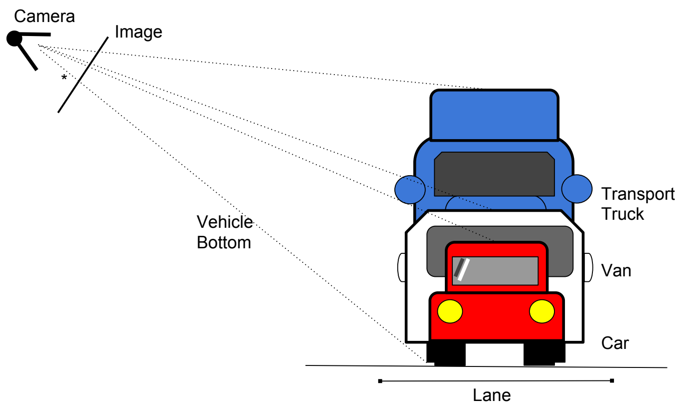

Fig. 2: A demonstration of the appearance of various sized vehicles in the image. The fronts of several vehicles are shown, all from the same lane. The vehicle tops have various image projections but the bottoms all converge near one image point. This may be used to define the location of the lane in the image.

1 Introduction

In computer vision, the heaviest focus on traffic lane detection is in the autonomous driving application. Only the driver's own lane boundaries and adjacent lanes need to be detected. Lanes must be localized in real world space from video image space. Hence the camera for this set-up is typically a calibrated front facing camera, relatively close to the road. The standard approaches often rely on line fitting to edges from road markings which are clearly visible to the driver from this vantage point [1, 2]. Ego motion in the image is present and initial lane detections are tracked and updated.

Another area where lane detection is required is for vehicle counting from a statically mounted traffic surveillance camera. Often, in the computer vision literature, traffic lanes are specified manually as a region of interest [4, 5, 6, 7, 8, 9]. This may be sufficient in some situations where only total volumes are required, but in many ITS applications a precise breakdown of counts for each lane may be required. Yet manual lane annotation can prove to be error prone, especially when lane marking are unclear and only a small segment of video or a still frame is presented to the user which does not fully demonstrate vehicle paths for every possible lane.

For the vehicle counting application, it is sufficient to fit the traffic lanes in the image space only. This avoids the need for calibration and extrinsic camera parameter estimation. Furthermore, if the lane detection may be performed off-line as a pre-processing step, lanes may be discovered and traced from the motion of detected vehicles. This results in more reliable lane placement than relying on possibly noisy, degraded, or invisible edge markings. Lane detection from moving vehicles was performed in [3] by tracking trajectories and simple clustering of fitted lines. This assumes that lanes are straight on a planar road and does not address noise due to vehicle tracking errors. Most other automatic lane detection work assumes straight planar lanes and employs manual annotation, camera calibration, or homographies [9].

2 Methodology

The objective of this article is to use the positions of vehicles as they move through the image in order to discover all of the traffic lanes. The general approach is to tightly localize vehicles in every frame and register their positions over time using a density image. A random search is used to fit multiple curve models to the density image and thereby identify each separate lane as an individual curve. See Fig. 1.

Since traffic cameras are mounted upright, vehicles are vertically aligned and their tops and bottoms appear up and down respectively. There may be a variety of vehicles of different heights so the location of vehicle tops and centroids may vary in the image. Nonetheless, the defining characteristic of all vehicles traveling inside a particular lane is the curve traced out by bottom point of their apparent boundaries as they move through the image, see Fig. 2.

A greedy random search is employed to fit multiple quadratic spline models to the density image. It is similar to sequentially applying Random Sample Consensus (RANSAC) to fit multiple models by removing fitted samples. There are a few key differences to address the well-known pitfalls when using the greedy sequential RANSAC strategy [11, 12].

2.1 Vehicle Density Image

Vehicle trajectories provide the locations of the lanes in which they travel. Unfortunately, solving the multiple target tracking and data association problem reliably enough to discover lanes from vehicles trajectories proves difficult in practice. Furthermore, especially for vehicle counting, reliable identification and tracking is best done in a reduced space where the traffic lanes are already identified [6, 7, 10].

In order to discover lanes from vehicle motion without explicitly tracking them, each vehicle is independently localized in every frame of video. Vehicles are detected using a deep learning vehicle contour detector which is fused with background subtraction [8]. This provides a precise closed boundary contour around moving vehicles from which the bottom point of each vehicle can be located in each frame. These points are added as samples in a vehicle motion density image. The resulting density image has

Fig. 3: Example vehicle motion density images. (a) and (c) show original video frames while (b) and (d) show the corresponding motion density. Note the false positive blobs in the parking lot of (b).

strong signal along lane paths. See Fig. 3. The log-sample-density image is taken to avoid saturation by stationary false positive vehicle detections. Robust model fitting is then used to eliminate noise and find the consistent lane paths.

2.2 Quadratic Spline Fitting

The curve models that are fitted to the samples are essentially specified by quadratic splines. The models and the fitting process are slightly enhanced to work more reliably within the sequential RANSAC framework.

Each curve is specified by 7 parameters: A spline with 3 $\mathbb{R}^2$ control points and a thickness parameter. As specified in RANSAC, 10000 random curves are proposed from random points in the density image and the best is selected. Unlike RANSAC, instead of using a threshold to count inliers for suitability measurement, a kernel is constructed for each proposed curve.

Fig. 4: An example image kernel for a proposed quadratic spline. It has the same dimensions as the vehicle density image. It is applied to the density image by element-wise multiplication of each pixel and summing. The white area near the curve is positive to encourage inlying points. Its thickness is a parameter to be optimized. The black area is negative and serves to penalize curves which don't fit along an isolated lane away from other density points. The gray area is 0.

The kernel is inspired by the Gabor Filter [13] and is applied to the density image to obtain a data fidelity score for a given model. An example kernel for one spline model can be seen in Fig. 4. Essentially the kernel positively scores close inliers, while penalizing conflating nearby points, and ignores distant outliers from other lanes and noise. This encourages models that have a strong agreement with inlier signal, yet are isolated from other signal. It also discourages models that cross multiple lanes and have inliers from each lane since the conflating lanes will have relatively high density inside the penalized area.

Instead of explicit parameter fitting, a hill climbing approach proves to be quite effective. It is much slower but is acceptable for this off-line pre-processing task. The 7 parameters for the selected model are optimized with respect to this data fidelity score. This finds the closest appropriate lane.

2.3 Multiple Lane Models

After a model has been fit to a lane, its thickness parameter is used to remove inlying points from the density image. It is added to a list of proposed final lanes and the process is resumed to find additional models.

This is continued until no random model can be found with sufficient data fidelity. Then a final step is employed to validate all proposed lanes globally. Each proposed lane model is further optimized on the original log-density image (where all previously removed model inliers reappear) to compute its global data fidelity. Models which now have a low score are likely to be too close to better existing lane models or were spuriously constructed from coincidental noise in the active density image with removed samples. They can safely be eliminated leaving true vehicle lanes.

3 Discussion

This method has been run on 135 different traffic videos and is showing some great promise. See Fig. 5. By assumption, the lanes must be distinct and separate and the kernel enforces this. The effect is that any lane which appears too close to another lane might be suppressed in the result.

Some measures have been taken to reduce the sensitivity of sequential RANSAC. In particular the assumption of isolated lane inliers may prevent a lane from being correctly detected if it is too close to noise from false positives. A true globally optimized multiple curve model estimator may prove more reliable than the

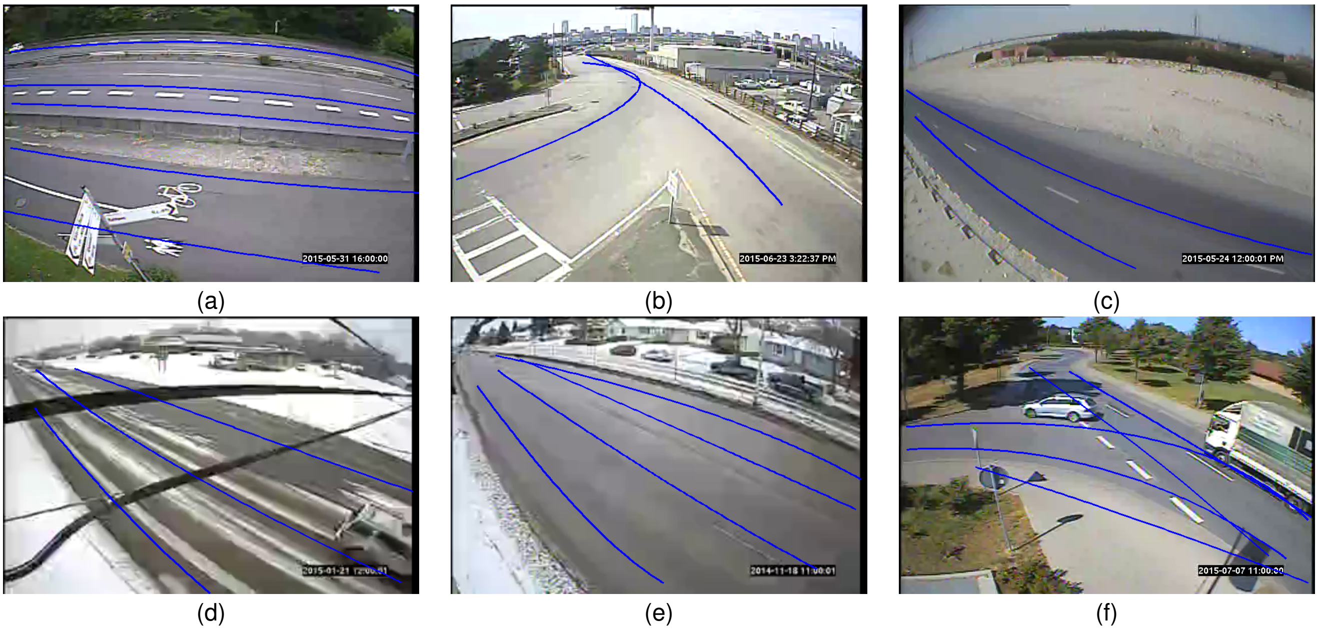

Fig. 5: Results running the presented method on various videos. Resulting lane curves are shown in blue. Note some distant lanes in (a) are merged together. (b) shows the limits of quadratic splines for turning movements. All lanes are successfully found in (d) despite snow covering lane markings. Pedestrians and cyclists sometimes generate coherent false positives in the density map resulting in spurious lanes in (f) and (a).

heuristics used here. To put the presented method into practise, a human-in-the-loop should be effective to remove spurious curves, make slight corrections, and to validate and correctly label resulting lanes.

Quadratic splines were chosen because of their suitability to represent curved lanes, straight lanes through curved camera lenses, and some turning movement lanes. More advanced models may be required in other traffic setups such as roundabouts. This could expose the method to over-fitting and may require a stiffness or regularization term in the optimization.

This is a costly non-deterministic off-line pre-processing step for analyzing traffic videos. Therefore some aspects of this lane finding approach are not suitable for real-time online deployment. Importantly, vehicles must be detected and segmented in every frame before lanes are even specified. Lanes are discovered solely by vehicle motion, so lanes where no representative vehicle is present in the entire video sample are necessarily missed. What is more, the resulting lane models are initialized only once. In the presence of camera motion and shifting it may be necessary to iteratively collect additional vehicle density and update the curve models.

References

[1] Sivaraman, S. and Trivedi, M. Integrated Lane and Vehicle Detection, Localization, and Tracking: A Synergistic Approach. *IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2 pp. 906-917.* (2013).

[2] Aly, M. Real Time Detection of Lane Markers in Urban Streets. *IEEE Intell. Vehicles Symposium* (2008).

[3] Sochos, J. Fully Automated Real-Time Vehicles Detection and Tracking with Lanes Analysis. *Central European Seminar on Computer Graphics*. (2014).

[4] Mithun, N. and Rashid, N. and Rahman, S. Detection and Classification of Vehicles from Video Using Multiple Time-Spatial Images. *IEEE Trans. Intell. Transp. Syst., vol. 13, no. 3, pp. 1215–1225* (2012).

[5] Michalopoulos, P.G. Vehicle Detection Video Through Image Processing: the Autoscope System. *IEEE Trans. Veh. Technol. vol. 40 no. 1 pp. 21-25*. (1991).

[6] Miller, N. and Thomas, M.A. and Eichel, J.A. and Mishra, A. A Hidden Markov Model for Vehicle Detection and Counting. *Conf. Computer and Robot Vision, pp. 269-276*. (2015).

[7] Yin, M. and Zhang, H. and Meng, H. and Wang, X. An HMM-Based Algorithm for Vehicle Detection in Congested Traffic Situations. *Proc. IEEE Intell. Transp. Syst*. (2007).

[8] McBride, K. Vehicle Tracking in Occlusion and Clutter. *M.A.Sc. Thesis, University of Waterloo, Waterloo, Ontario*. (2007).

[9] Coifman, B. and Beymer, D. and McLauchlan, P. and Malik, J. A Real-Time Computer Vision System for Vehicle Tracking and Traffc Surveillance. *Transportation Research Part C. vol. 6, no. 4, pp .271-288* (1998).

[10] Miller, N. and Swart, David M. Straightening Curved Traffic Lanes in Video. *Vision Letters, vol. 1, no. 1*. (2015).

[11] Zuliani, M. and Kenney, C. S. and Manjunath, B. S. The Multiransac Algorithm and its Application to Detect Planar Homographies *Proc. IEEE International Conf. on Image Processing*. (2005).

[12] Isack, H. and Boykov, Y. Energy-based Geometric Multi-Model Fitting. *International Journal of Computer Vision, vol. 97, no. 2, pp. 123-147*. (2012).

[13] Bovik, A. C. and Clark, M. and Geisler, W. S. Multichannel Texture Analysis using Localized Spatial Filters. *IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 1, pp. 57-73*. (1990).