Video-based Self-positioning for Intelligent Transportation Systems Applications

Parag S. Chandakkar, Ragav Venkatesan and Baoxin Li, Senior Member, IEEE

Computer Science and Engineering, Arizona State University, Tempe, AZ, U.S.A.
{pchandak, ragav.venkatesan, baoxin.li}@asu.edu

Abstract. Many urban areas face traffic congestion. Automatic traffic management systems and congestion pricing are getting prominence in recent research. An important stage in such systems is lane prediction and on-road self-positioning. We introduce a novel problem of vehicle self-positioning which involves predicting the number of lanes on the road and localizing the vehicle within those lanes, using the video captured by a dashboard camera. To overcome the disadvantages of most existing low-level vision-based techniques while tackling this complex problem, we formulate a model in which the video is a key observation. The model consists of the number of lanes and vehicle position in those lanes as parameters, hence allowing the use of high-level semantic knowledge. Under this formulation, we employ a lane-width-based model and a maximum-likelihood-estimator making the method tolerant to slight viewing angle variation. The overall approach is tested on real-world videos and is found to be effective.

1 Introduction

Globally, per-capita vehicle ownership is on a constant rise. In the US[1] alone, in the year 2010, the per-capita vehicle ownership was 0.78 with an average individual covering over 15528 kilo-meters. There were 1.77 fatalities per-kilo-meter, in the year 2010. In the year 2011, there were 211.87 million drivers and 311.59 million people traveling on highways alone, with over 244.77 million vehicles, of which 130.89 million are cars. Travel on roads in the US, increased by over 1.5% from September 2012 to September 2013, with over 345.5 billion vehicle kilo-meters covered in this time-line.

This situation is not unique to the U.S. Similar trends were found in the UK[2] as well. Cars, vans and taxis traveled about 800 billion passenger kilo-meters. Distance traveled by cars observed a steep increase from 27% to 87% from 1952 to 2013. In the year 2013, 64% of all transportation was done by car, with over 34 million vehicles used by over 62 million people everyday. The national transport model center forecast in UK predicts a 43% increase in vehicle traffic from 2010 to 2040. With such drastic increase in global road-traffic, intelligent transport system (ITS) research has taken center-stage prominence in recent years.

A typical ITS involves collecting real-time vehicle and traffic information, estimating flow and congestion, and providing corrective feedback, usually through congestion

[1] US statistics were from http://www.fhwa.dot.gov.

[2] UK statistics were from http://www.gov.uk; Tables TSGB0101 to TSGB0102

pricing and toll. Today's state-of-the-art ITS stands at a level where they collect information at a road-level and provide congestion pricing for platoons of cars entering the road and exiting the road in a connected-pipes topology [1,2]. The envisioned future for this kind of systems is to predict and model vehicle patterns and traffic at the lane-level, along with modeling inter-lane and intra-lane traffic changes in addition to road level traffic changes. This enables the ITS to price different lanes differently and to provide routing assistance to drivers at a lane-level. Therefore self-positioning for cars is an important stage in such an ITS model. Although, many hardware dependent modalities such as lane marker detection using LIDAR and using those using GPS are popular in literature, they are cost in-efficient and are not usually preferred [3,4]. With the importance of the problem and the multi-modal data streams involved, ITS applications calls for development of new multi-media processing and analysis techniques.

Video-based lane detection and prediction is a popular modality as digital cameras have become ubiquitous and portable. Although in the past, many simple solutions such as the Hough transform, were found to give acceptable results in lane detection, extracting knowledge about lanes from them is a difficult task [5]. In this paper, we develop a lane modeling and prediction system from a wind-shield mounted front-facing camera. The aim of this system is not to detect lane markers, although we treat it as a subproblem, but to predict the number of lanes on the road and the lane at which the car is presently being driven. In this paper, we propose a Bayesian joint model with the number of lanes and lane number as parameters and we maximize the likelihood for every combination of these parameters. The likelihood function is developed in a bottom-up approach using guided filter, lane-width modeling and prediction confidence of a classifier. To facilitate the exploration of the vehicle self-positioning problem and to improve and verify the current techniques, we have made the database publicly available.

The rest of the paper is organized as follows. Section 2 discusses recent literature on lane detection. Section 3 formalizes the proposed method. Section 4 provides experiments and results. Section 5 provides concluding remarks.

2 Related Work

There have been extremely less attempts at predicting number of lanes and the position of the car on road in literature. There is a plethora of lane detection algorithms using lane-marker detection. Lane markers are usually detected in a multi-stage approach, typically involving an intra-frame prediction accompanied by an inter-frame adjustment. The first stage will usually comprise of a pre-processing to select some candidate regions and later stages will involve tracking-based methods to exploit the correlation between consecutive frames, thereby reducing false detections. Pre-processing stage followed by tracking using Kalman filter was performed for detecting lanes in [6]. Huang et al, proposed a method where, a simple set of filters were used to detect candidate regions and second-order derivatives were used for further filtering [3]. A similar idea was used in another lane-keeping paper, where the lane model was estimated row-by-row from the bottom of the image to the top. This way, the lower rows provided reference to detect lane markers on higher rows [7]. The supportive argument for this method was the fact that probabilities of finding lane-markers on lower end of the image in a front-view is more than the higher end. Many methods enforce road models using temporal integration. In one such tracking-based method the existing lane model was enforced on the new image and only newer or emerging lanes were considered [8]. Similarly, in another method, albeit not as strict, the existing lane model was used as a constraint to find the lanes in the current frame in a Hough transform based setup [9]. A simpler way of using a similar idea is to define a region of interest in the current frame using the the predictions of the previous frames. Such an idea was discussed in [10].

Since lanes are typically lines or curves, the use of line fitting or grid fitting is quite popular. This is done either in the original view or in a transformed view, usually using inverse perspective mapping (IPM). Parametric line-fitting is used in many methods [11,12,13]. In these methods, after an initial detection of lane-markers, lines are fit using various methods to each lane-lines. Although they work well, they are dependent on the initial prediction. Roads are not typically straight and sometimes require curves to be fit. Many methods in literature fit either splines or poly-lines to solve this issue [14,15]. The major problem in using active-contour models like b-snakes for lane detection is that, they are computationally in-efficient in convergence to a final contour. They also require seed points which are typically provided from edge maps or pre-assumed regions. Further more, even after the contouring of lane-markers, one still has to classify contours between lanes and non-lanes. Beyond curve and line fitting is a grid-fitting approach. One interesting method in the IPM view is the hierarchical bipartite graph (HBPG) [16]. The major advantage of this approach is that the lane model is accompanied by a confidence measure for each lane. Another method finds and tracks multiple lanes by using computer vision techniques and LIDAR [17]. These both are stable systems for lane-marker detection. Both of them have their disadvantages. The former method doesn't detect lanes beyond a limit and the latter depends on expensive components such as LIDAR. Both methods do not attempt at predicting number of lanes or lane number at which the car is currently being driven. An approach using Spatial RAY features has tried to predict the position of the vehicle on road given a video stream [18]. However, they predict at most 3 lanes either on the left or right hand side, whereas we have upto 6 lanes in our database. They also have low traffic density and use IPM for processing the data. We have normal-to-moderate traffic density and do not use IPM which makes our method tolerant to different positions of the camera. Therefore, direct comparison between these may not be appropriate.

Due to increase in computational capabilities and developments in fields such as computer vision, a few learning based approaches have also been developed recently. Patch-learning based approaches have also been developed to classify patches in images as containing lane-markers. While most methods uses image gradients as features [19], steerable filters based features have also proven reliable [20] for such patch based methods. Detection of lanes is further bettered by introducing learning of additional knowledge such as types of lanes by recognition of colors and structure of lane-lines [21]. Texture of the lane-markers were studied and along with a pre-defined lane-model ,and further adjustments using Hough [22].

While innumerable methods of lane-marker detection, lane detection and lane modeling methods exist in literature, and a survey of all of them are beyond the scope of this paper, a select few are surveyed to give the reader a general idea as to the directions that the research in this field have so far taken. Albeit the all the effort in lane detection, there are no solutions to predicting the number of lanes or self-positioning of vehicles given a front-facing view.

3 Proposed Method

Assume that we have labeled data $D = \begin{bmatrix} X_1 & \varTheta_1 \\ X_2 & \varTheta_2 \\ \vdots & \vdots \\ X_n & \varTheta_n \end{bmatrix}$ where X_i is the i^{th} video clip and $\varTheta_i$ is a $2D$ vector label $[\theta_1, \theta_2]$, where $\theta_1 \in [1, L]$ is the number of lanes on the observed video, L being the maximum number of lanes considered and $\theta_2 \in [1, \theta_1]$ is the lane at which the car is currently driving, and each video clip in X_i contains frames $[f_i^1, f_i^2, \dots, f_i^m]$. The joint probability density of the proposed model for any frame f_r^j such that $X_r \notin D$ is $P(f_r^j, \varTheta | D, f_r^{j-1}, f_r^{j-2}, \dots, f_r^{j-b})$, where $[f_r^{j-1}, f_r^{j-2}, \dots, f_r^{j-b}]$ is a buffered set of b frames immediately before f_r^j.

The posterior density for the prediction of the label $\hat{\varTheta}$ at frame f_r^j given the previous b frames and the training data D is, $P(\varTheta | f_r^j, f_r^{j-1}, f_r^{j-2}, \dots, f_r^{j-b}, D)$. By Bayes rule,

$$P(\varTheta | f_r, f_r^{j-1}, f_r^{j-2}, \dots, f_r^{j-b}, D) \propto \\ P(f_r, f_r^{j-1}, f_r^{j-2}, \dots, f_r^{j-b} | \varTheta, D) P(\varTheta | D). \quad (1)$$

Assuming a uniform prior $P(\varTheta | D)$, the maximum-a-posteriori problem becomes a maximum-likelihood estimation problem as

$$\hat{\varTheta} = \arg\max_{\varTheta} P(f_r^j, f_r^{j-1}, f_r^{j-2}, \dots, f_r^{j-b} | \varTheta, D). \quad (2)$$

The weighted likelihood for each frame to be maximized is approximated as,

$$P(f_r^j, f_r^{j-1}, \dots, f_r^{j-b} | \varTheta, D) \approx \\ \varPhi^T [P(f_r^j | \varTheta, D), P(f_r^{j-1} | \varTheta, D), \dots P(f_r^{j-b} | \varTheta, D)], \quad (3)$$

where $\varPhi$ is a weighting kernel.

The advantage of modeling this formulation as a Bayesian framework is that it can be easily expanded to include other parameters in the future, For example car dynamics and higher-level knowledge about the road structure such as vehicle traveling directions in various parts of the road. As the problem complexity grows and more data becomes available, more classifiers/detectors/other small frameworks can be included as a part of this bigger framework. This will help us in the future to create more accurate lane assignments for vehicles. This formulation also helps us to update this likelihood for every incoming frame as shown in equation 4. Although $\varPhi$ can be any weighting kernel, we consider an averaging kernel:

$$\varPhi = \frac{1}{b+1} \cdot \mathbf{1}, \quad (4)$$

where $\mathbf{1}$ is a column vector of length $b+1$. Averaging gives uniform weighting to the current frame and all the frames in the buffer, thus helping us to compensate for false predictions. This requires us to create a likelihood function $P(f|\Theta, D)$ for every frame f and every label Θ.

3.1 The likelihood function.

There are several stages in the estimation of the likelihood function, $P(f|\Theta, D)$ for each frame f. These are:

1. Pre-processing using guided filter.
2. Lane-width modeling.
3. Feature extraction.
4. Estimation of $P(f|\Theta, D)$.

Guided filtering. Guided filter is a linear-time, edge-preserving, smoothing filter [23]. It works with a pair of images. One of these images is a reference image, with the guidance of which, the other is filtered. When both images are the same, the guided filter smooths the entire image while preserving the edges. For the special case where both the images are identical, the filtering is performed as follows:

$$\mathcal{G}(f) = \bar{\alpha} f + \bar{\beta}, \tag{5}$$

where f is a gray-scale video frame. $\bar{\alpha}$ and $\bar{\beta}$ are block-wise averaged values generated from $\alpha = \dfrac{\sigma^2}{\sigma^2 + \epsilon}$ and $\beta = (1-\alpha)\mu$, where μ is the mean, σ is the standard deviation for any block in the image and ϵ is a small constant. If there is a flat patch in the image then $\sigma = 0 \implies \alpha = 0$ and $\beta = \mu$. Each pixel in the flat region is thus averaged with its neighbors. For a high variance region such as lane edges, $\sigma \gg \epsilon$ and hence $\alpha \approx 1$ and $\beta \approx 0$. This leaves the edges unchanged. A post-processing operation such as over-subtracting and then saturating returns an image similar to the one shown in the middle in Fig. 1. The post-processing operation can be represented as:

$$Binarized_f = [(f - \delta\mathcal{G}(f)) * 255] \geq 0; \tag{6}$$

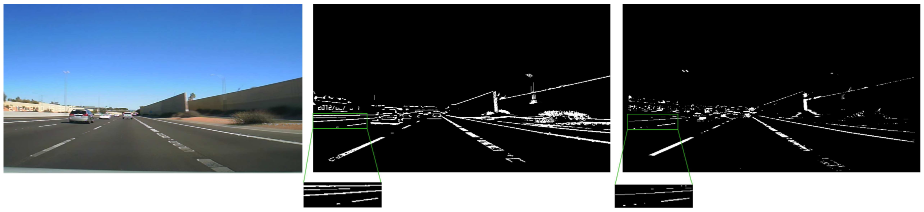

Fig. 1: From left to right: Captured frame, pre-processing using guided filter, processing using Marcos' filter.

where $\delta > 1$ (in this case, $\delta = 1.06$) due to which the original image gets over-subtracted from itself. Since only sharp edges have retained their value in the filtered image, other regions become negative after over-subtraction. This provides us a much better frame to work with. Fig. 1 shows the application of the guided filter and compares it with Marcos' filter, specially designed for lane detection [24]. The Marcos' filtering process can be defined as follows:

$$y_i = 2x_i - (x_{i-\tau} + x_{i+\tau}) - |x_{i-\tau} - x_{i+\tau}| \tag{7}$$

τ is the width parameter which controls the filtering process. The filtering process will produce high response if the current pixel has high value and pixels at a distance of τ on both sides have equal value. The lane pixels satisfy these criteria. However, due to the width parameter, Marcos' filter is not as reliable as guided filter in detecting the lane markers in extreme left/right. Fig. 1 shows the effect of guided and Marcos' filter on a frame. The zoomed-in region shows that Marcos' filter fails to detect the extreme left lane. Guided filter also works in many weather conditions, for example, rainy day, low-sunlight conditions, low-contrast images etc. Section 4 compares the results obtained with guided and Marcos' filter.

Lane-width modeling. The pre-processing using guided filter provides us a frame that contains all the lane markers and some spurious objects. Before extracting features, these spurious regions may be filtered. Assuming that the lanes on the free-ways and such are of a fixed width, the width of lanes, and thereby the probable regions for finding lane markers in the video, can be modeled as a function of camera parameters and the ordinates in the image plane with respect to an initial lane (center-lane in this case).

We first detect the lane-markers of the lane in which the car is present. The center lane markers are detected from the pre-processed video using heuristics about its possible locations and low-level color and gradient features. A line is fit for left ($c_l[x_l, y_l]$) and right ($c_r[x_r, y_r]$) center-lane-lines individually along the detected lane-markers. (x_l, y_l) and (x_r, y_r) are the points lying on the left and right lane markers of the center lane respectively. Three additional lane-lines on the left and right of the center-lane are modeled as:

$$l_i[x_l, y_l] = d_{l_i}(c_l[x_l, y_l]) = \psi_i^l y_l + \tau_i^l, \forall i \in [1, 2, 3], \tag{8}$$

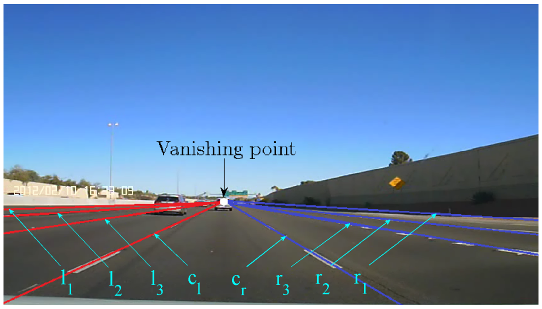

Fig. 2: Lane width modeling

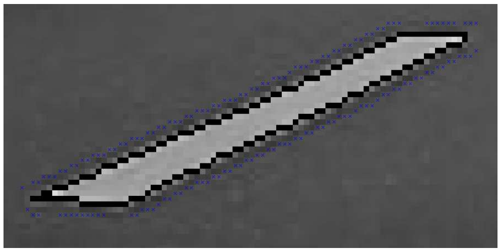

Fig. 3: Pixel types: The lane pixels in white are type-1. The pixels with blue cross mark are type-2. Black pixels indicate the boundary of lane pixels. The rest are type-3 pixels. Please zoom in on a computer monitor for better viewing.

and,

$$r_i[x_r, y_r] = d_{r_i}(c_r[x_r, y_r]) = \psi_i^r y_r + \tau_i^r, \forall i \in [1, 2, 3]. \tag{9}$$

where $l_i[x_l, y_l]$ and $r_i[x_r, y_r]$ are the x and y coordinates lying on the i^{th} left and right of the center lane markers respectively (also shown in Fig. 2). $d_{l_i}(c_l[x_l, y_l])$ and $d_{r_i}(c_r[x_r, y_r])$ are the horizontal distances of the l_i and r_i lane markers from c_l and c_r respectively. y_l and y_r are the y-coordinates of the left and right lane markers. $\{\psi_i^l, \psi_i^r, \tau_i^l, \tau_i^r\}$ are the slope and intercept parameters of the left and right lane markers.

Feature extraction. Fig. 2 shows the lane-lines that are modeled for $L = 7$. For simplicity, we assume a constant camera position. Equation 8 and 9 model the distance between all the left (right) lane markers and the center-left (center-right) lane marker as a function of only the y-coordinate in the image. Other parameters such as the camera parameters can also be included. Similarly, although it was sufficient for us to model the displacement as a linear function, other kernel representations may also be used. The intersection of these regions is the vanishing point, which can act as an additional parameter in this formulation. The vanishing point is the intersection of the lines $\{l_i\}_{i=1}^3$, $\{r_j\}_{j=1}^3$, c_l and c_r as shown in Fig. 2.

For every region chosen by the lane-width model, we classify every pixel in the region into type-1, type-2 or type-3 pixels. The pixels given by the guided filter are type-1 pixels. The pixels that are just outside the type-1 pixels and along the direction of the gradient of the nearest type-1 pixel are the type-2 pixels. Every other pixel in the region is classified as type-3. Type-1 and type-2 pixels are mean centered. We empirically observed that subtracting the mean from the pixels provides robustness against varying lighting conditions. Fig. 3 shows the various types of pixels.

For every region now we extract the following 40 dimensional feature:

1. First and second moments of the type-1 pixels ($2 - D$).
2. First and second moments of the type-2 pixels ($2 - D$).
3. 36-bin histogram of gradients of type-1 pixels ($36 - D$).

We neglect the type-3 pixels as they usually constitute anomalies with the road, like shadows, repair and vehicles. This gives us a $40 - D$ vector for each lane. First four

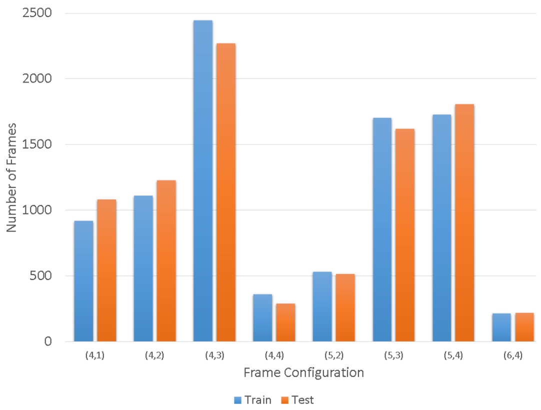

Fig. 4: Distribution of Video Frames in Train and Test Set

features act as low-level color features by encoding mean and standard deviation of lane and road pixels respectively. The remaining features describe the shape and the color contrast of the contour under consideration. A lane line would have most of the gradients pointing only in one direction (treating two opposite directions as one) and with high contrast and thus high magnitudes. We found these low-level statistics enough to describe lane-lines well. The maximum number of lanes we consider are $L = 7$. We model all the lanes except the lane in which the car is present. Therefore our feature dimensionality is 240.

Estimation of $P(f|\Theta, D)$ A linear SVM is constructed using the features extracted from D. For any test frame, the same set of extracted features are projected onto the SVM feature space. The likelihood estimate $P(f|\Theta, D)$ is the confidence of prediction for each Θ. It is measured as the distance of the data-point to the hyper-plane. The likelihood extracted is then fed into the MLE system for prediction.

4 Experiments and Results.

The videos for this experiment have been captured using a camera mounted on the inside of the wind-shield. Our data set contains 53 videos, captured at 30 frames/second [3]. The average length of each clip is 11.8 seconds. Though there are many possible configurations with $L = 7$, we consider only a subset of frequently occurring configurations. The following configurations exist in our database: $\Theta = \{[4,1], [4,2], [4,3], [4,4], [5,2], [5,3], [5,4], [6,4]\}$. Uniform prior is assumed among all these configurations and 0 prior for configurations that are not considered. We use 27 out of 53 videos in our database for training purposes and the rest for testing purposes. The total number of frames in training and testing set are 9018 and 9036. The distribution of video frames over all

[3] The dataset is available at http://www.public.asu.edu/~bli24/CodeSoftwareDatasets.html

Table 1: Per-class accuracy.

	Guided filter			
Class = $[\theta_1, \theta_2]$	RF	Lin. SVM	**Model + Lin. SVM** (Proposed method)	Marcos' filter+ +Model+RF (Best combination)
$[4, 1]$	30.22%	46.67%	44.27%	30.48%
$[4, 2]$	76.94%	73.27%	80.03%	38.47%
$[4, 3]$	55.22%	68.75%	71.66%	57.29%
$[4, 4]$	98.98%	79.52%	82.59%	21.84%
$[5, 2]$	36.94%	56.48%	63.25%	47.40%
$[5, 3]$	63.99%	53.30%	51.70%	59.32%
$[5, 4]$	46.46%	39.06%	39.94%	59.91%
$[6, 4]$	96.80%	96.80%	100%	98.17%
Overall Accuracy	56.37%	58.33%	**60.15%**	51.76%

configurations in training and testing set can be seen in Fig. 4. Each configuration contains videos of varying lighting, weather and surface conditions. In this problem, the training data has noise embedded in it. Many frames contain vehicles which occlude the full view of all the lanes. These external factors increase the difficulty.

The results are presented in two-fold. Table 1 compares the per-class accuracy obtained using guided and Marcos' filter. Two classifiers, Linear SVM and Random Forest (RF) have been used. Due to the uneven distribution of data in each class, weights in RF have been adjusted accordingly. For Marcos' filter, best performance has been reported. Due to Marcos' filter's inability to detect extreme lanes, it has poor accuracy when compared to guided filter. Table 1 also shows the per-class accuracy of detection for both the prediction from the SVM stage and the improvements seen using the MLE model. Table 2 shows the confusion matrix for the initial prediction and Table 3 shows the confusion matrix for the final prediction by MLE. From the confusion matrices, it can be seen that the MLE estimator betters the initial prediction in all but two configurations. The decrease in the performance occurs when SVM wrongly predicts for at least $b/2$ consecutive frames with high probability. Increasing the value of b increases the accuracy but it also delays the determination of the current frame configuration. We input

Table 2: Confusion matrix for SVM prediction.

Class = $[\theta_1, \theta_2]$	$[4, 1]$	$[4, 2]$	$[4, 3]$	$[4, 4]$	$[5, 2]$	$[5, 3]$	$[5, 4]$	$[6, 4]$
$[4, 1]$	505	399	96	0	58	24	0	0
$[4, 2]$	6	899	52	0	231	14	2	23
$[4, 3]$	112	181	1560	8	1	308	99	0
$[4, 4]$	0	39	13	233	3	0	4	1
$[5, 2]$	11	127	69	10	292	0	8	0
$[5, 3]$	0	569	166	0	0	863	2	19
$[5, 4]$	11	448	268	14	6	4	707	352
$[6, 4]$	0	3	0	0	0	0	4	212

Table 3: Final Confusion Matrix

Class = $[\theta_1, \theta_2]$	$[4,1]$	$[4,2]$	$[4,3]$	$[4,4]$	$[5,2]$	$[5,3]$	$[5,4]$	$[6,4]$
$[4,1]$	479	391	134	0	63	15	0	0
$[4,2]$	3	982	27	0	215	0	0	0
$[4,3]$	112	170	1626	0	0	266	95	0
$[4,4]$	0	31	13	242	0	7	0	0
$[5,2]$	9	119	45	8	327	0	9	0
$[5,3]$	0	614	168	0	0	837	0	0
$[5,4]$	14	494	232	0	0	0	723	347
$[6,4]$	0	0	0	0	0	0	0	219

previous 10 frames ($b = 10$) to the MLE estimator. Also due to averaging kernel, there is a smooth transition in the likelihood when the configuration changes. Therefore, first $b/2$ frames may be predicted wrongly which may decrease the final accuracy. While the configurations $[4, 1]$, $[5, 4]$ have low accuracy, it should be considered that there are 3 lanes on one of the sides of the current lane. The presence of the third lane is difficult to detect given that the lane is not more than 4 pixels wide in the frame. The approach also has to deal with broken and faint lane lines on the road in addition to occlusion. Interestingly, the configuration $[6, 4]$ achieves 100% accuracy despite having the most number of lanes. We would like to point out the fact that the $[6, 4]$ configuration has the best weather and relatively less occlusion from vehicles. It shows that our approach can achieve excellent results if the road conditions, weather and the traffic density permits.

The final confusion matrix in Table 3 shows that two lowest performing configurations are $[5, 4]$ and $[4, 1]$. Bad weather conditions and the presence of extreme lane in both the configurations, as shown in Fig. 5, reduces the classification performance. This is an eight-class classification problem and the variable weather conditions, road

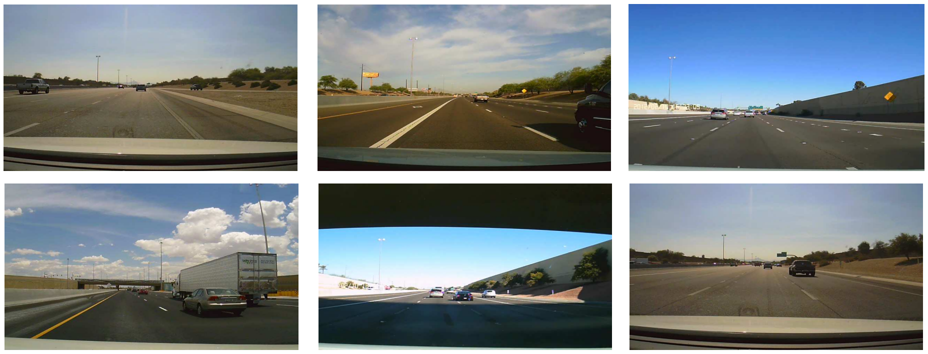

Fig. 5: From left to right, row 1: Correct predictions for configurations $[4, 4]$, $[5, 2]$ and $[6, 4]$. Row 2: Wrong predictions for configurations $[4, 1]$, $[5, 3]$ and $[5, 4]$. Occlusion, bad lighting conditions and bad road conditions are the plausible causes for wrong detections.

conditions and inherent noise in the training data makes the problem challenging. As shown in Fig. 5, the proposed approach is able to handle partial occlusions and bad road conditions. Still many improvements can be made to the current approach. Apart from the lane-width modeling, it can incorporate other semantic knowledge such as the vehicle movement, vehicle detection to assist lane detection etc. The current approach does not perform so well in bad road conditions such as broken lanes or disappeared lane markers. More sophisticated tracking along with accelerometer and a better lane-model can be used to overcome these problems.

5 Conclusion and Future Scope

A novel problem of predicting the number of lanes from a video and self-positioning of the car in those lanes was introduced and a high-level model based approach is developed. This video-based self-positioning approach provides the system an ability to perform dynamic traffic routing on a lane-level. This would possibly contribute to greater reduction of congestion and thus result in faster travel times. This was formulated as a top-down maximum-likelihood estimator. The likelihood of the two parameters were modeled using the confidence of a low-level predictor and guided-filtering. Lane-width modeling and pixel-level statistics were used to choose candidate-lane regions before the prediction. Testing this framework on real-world videos yielded satisfactory results.

The system presently accommodates just the video data to perform prediction. Our aim is to expand this model to include vehicle dynamics and GPS information. Vehicle dynamics which includes (but not limited to) accelerometer data, speed information etc. can be used as a high-level knowledge to filter out obvious false detections. To increase reliability of prediction on the roads with large number of lanes, we may obtain the number of lanes as an input. Our ultimate aim remains to integrate all these systems to build a lane-level dynamic traffic routing system.

Acknowledgement: The work was supported in part by a grant from the National Science Foundation. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

1. Zheng, H., Chiu, Y.C., Mirchandani, P.B.: On the system optimum dynamic traffic assignment and earliest arrival flow problems. Transportation Science (2013)
2. Angel, A., Hickman, M., Mirchandani, P., Chandnani, D.: Methods of analyzing traffic imagery collected from aerial platforms. Intelligent Transportation Systems, IEEE Transactions on **4** (2003) 99–107
3. Kammel, S., Pitzer, B.: Lidar-based lane marker detection and mapping. In: Intelligent Vehicles Symposium, 2008 IEEE, IEEE (2008)
4. Huang, A.S., Moore, D., Antone, M., Olson, E., Teller, S.: Finding multiple lanes in urban road networks with vision and lidar. Autonomous Robots **26** (2009) 103–122
5. Hillel, A.B., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. Machine Vision and Applications (2012) 1–19

6. Nieto, M., Laborda, J.A., Salgado, L.: Road environment modeling using robust perspective analysis and recursive bayesian segmentation. Machine Vision and Applications **22** (2011) 927–945
7. Wu, S.J., Chiang, H.H., Perng, J.W., Chen, C.J., Wu, B.F., Lee, T.T.: The heterogeneous systems integration design and implementation for lane keeping on a vehicle. Intelligent Transportation Systems, IEEE Transactions on **9** (2008) 246–263
8. Lipski, C., Scholz, B., Berger, K., Linz, C., Stich, T., Magnor, M.: A fast and robust approach to lane marking detection and lane tracking. In: Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on, IEEE (2008) 57–60
9. Cheng, H.Y., Jeng, B.S., Tseng, P.T., Fan, K.C.: Lane detection with moving vehicles in the traffic scenes. Intelligent Transportation Systems, IEEE Transactions on **7** (2006) 571–582
10. Labayrade, R., Douret, J., Laneurit, J., Chapuis, R.: A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance. IEICE transactions on information and systems **89** (2006) 2092–2100
11. Kong, H., Audibert, J.Y., Ponce, J.: Vanishing point detection for road detection. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 96–103
12. Alon, Y., Ferencz, A., Shashua, A.: Off-road path following using region classification and geometric projection constraints. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 689–696
13. Zhang, G., Zheng, N., Cui, C., Yan, Y., Yuan, Z.: An efficient road detection method in noisy urban environment. In: Intelligent Vehicles Symposium, 2009 IEEE, IEEE (2009) 556–561
14. Sawano, H., Okada, M.: A road extraction method by an active contour model with inertia and differential features. IEICE transactions on information and systems **89** (2006) 2257–2267
15. Wang, Y., Teoh, E.K., Shen, D.: Lane detection and tracking using b-snake. Image and Vision computing **22** (2004)
16. Nieto, M., Salgado, L., Jaureguizar, F.: Robust road modeling based on a hierarchical bipartite graph. In: Intelligent Vehicles Symposium, 2008 IEEE, IEEE (2008) 61–66
17. Huang, A.S., Moore, D., Antone, M., Olson, E., Teller, S.: Finding multiple lanes in urban road networks with vision and lidar. Autonomous Robots **26** (2009) 103–122
18. Kuhnl, T., Kummert, F., Fritsch, J.: Visual ego-vehicle lane assignment using spatial ray features. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE (2013) 1101–1106
19. Samadzadegan, F., Sarafraz, A., Tabibi, M.: Automatic lane detection in image sequences for vision-based navigation purposes. In: Proceedings of the ISPRS Commission V Symposium'Image Engineering and Vision Metrology. (2006)
20. McCall, J.C., Trivedi, M.M.: Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. Intelligent Transportation Systems, IEEE Transactions on **7** (2006)
21. Collado, J.M., Hilario, C., De La Escalera, A., Armingol, J.M.: Adaptative road lanes detection and classification. In: Advanced Concepts for Intelligent Vision Systems, Springer (2006)
22. Zhou, S., Jiang, Y., Xi, J., Gong, J., Xiong, G., Chen, H.: A novel lane detection based on geometrical model and gabor filter. In: Intelligent Vehicles Symposium (IV), 2010 IEEE, IEEE (2010) 59–64
23. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell **35** (2013) 1397–1409
24. Nieto, M., Laborda, J.A., Salgado, L.: Road environment modeling using robust perspective analysis and recursive bayesian segmentation. Machine Vision and Applications **22** (2011)