# Vehicle Behavior Analysis for Uneven Road Surface Detection

Shubhranshu Barnwal
Research and Development Center, Hitachi India Pvt. Ltd.

***Abstract*— Although on-road vehicle detection and tracking is a well-researched area, vehicle behavior analysis is less addressed and still in a nascent stage. In our study, after successful vehicle detection and tracking, vehicle behavior is modeled using two dimensional spatio-temporal trajectories for uneven road surface detection. Uneven road surfaces are identified from specific motion patterns which are described using a Hidden Markov Model based trajectory classification model. We claim that capturing the sudden change in vehicle's vertical motion pattern can suggest an anomaly, like a speed bump or a pothole on road.**

***Keywords—In-Vehicle Camera Applications, Uneven Road Surface Detection, Speed Bump Detection, Pot Hole Detection***

## I. INTRODUCTION

According to the World Health Organization, road traffic injuries caused an estimated 1.24 million deaths worldwide in the year 2013. 238,562 fatalities were recorded in India for 2013 by Ministry of Road Transport, Govt. of India, making India with the highest number of road fatalities in the world [1], [2]. Over the past decade, there has been significant research effort dedicated to the development of intelligent driver assistance systems and autonomous vehicles, which is intended to enhance safety by monitoring the on-road environment

From more than a decade of research work, a variety of sensing technologies has become available for driver assistance systems and autonomous vehicles, including radar, lidar and computer vision. At the same time camera sensing and computational technologies have also greatly advanced making on-road vehicle identification and tracking more accurate and robust. It is now commonplace for research studies to report the ability to reliably detect and track on-road vehicles in real time, over extended periods [3], [4]. Theoretical, practical and algorithmic advances have opened up research opportunities that seek higher level of semantic interpretation on on-road vehicle behavior. The aggregate of this spatiotemporal information from vehicle detection and tracking can be used to identify maneuvers and to learn, model and classify on-road behavior. Examples of work in this nascent area include prediction of turning behavior [3], prediction of lane changes [5], modeling typical on-road behavior [6], detection of overtaking and receding vehicle with respect to ego vehicle [7], etc.

As it is evident the technology involved in in-vehicle camera applications have reached a maturity level where we are now able to identify and track target cars with fairly good accuracy, but understanding their behavior on road is still in a very nascent stage. One such less explored problem is that of identification of speed bumps and potholes on road by observing target vehicle's on-road motion behavior. Speed limit enforcement is almost non-existent in India, and authorities rely on using speed-bumps to enforce vehicle speed within city limits. As a result, over time there has been excessive use of speed-bumps in India. Also lack of maintenance of road infrastructure has left many roads with potholes and lack of street lights, thus making driving without caution for speed bumps and potholes dangerous for driver and passenger safety. Not slowing down to appropriate speeds at potholes and speed bumps also reduce vehicle life because of damages to vehicle parts like suspension, thus demanding more vehicle maintenance. When driving on road during night time with insufficient lighting and some a-priori knowledge that road are not well maintained in a particular locality, it is common behavior in India, to drive behind a car and take cues about road structure from its behavior of when it breaks, maneuvers around or undergoes a sudden jerked motion. Translating this human behavior to a computer driven model, we thus make an effort to find anomalies in road surfaces by modeling vehicle behavior using two dimensional spatio-temporal trajectories. To our best knowledge, this automated analysis of such events to identify uneven road surface is the first of its kind in literature.

The research target is to demonstrate that the proposed idea to analyze vehicle behavior is viable to identify events wherein the target vehicle passes over a speed bump or a pothole.

## II. VEHICLE DETECTION AND TRACKING

### A. Vehicle Detection

Robust on-road vehicle detection is a challenging computer vision problem. Changing backgrounds and illuminations make roads a dynamic environment. Target vehicles on roads are always in a relative motion w.r.t. ego vehicle, as a result their size and location continually changes. Videos captured from camera mounted on the ego vehicle may be jittery from vehicle's vibration and may have to be stabilized for further tasks. In addition, there is high variability in the shape, size, and appearance of vehicles found on a given road, and they differ even more across different geographical locations.

For the vehicle detection task, we have chosen to use boosted cascade classifiers of Haar-like rectangular features, which was introduced by Viola and Jones [8] in the context of face detection. Since then, many papers have used this object detection framework in on-road vehicle detection systems such as [9]-[12]. The set of Haar-like features is well-suited to the detection of the shape of vehicles, because the rectangular features are sensitive to edges, bars, vertical and horizontal details, and symmetric structures.

In addition, the algorithm allows for rapid object detection that can be exploited in building a real-time system. Part of this is due to the fact that feature extraction is extremely fast and efficient, due to the use of the integral image, an intermediate representation for the image. Using the integral image

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.279

representation, the feature extraction of a Haar-like rectangular feature can be computed in just four array references. The resulting extracted values are effective weak learners [8],[11], which are then classified by Adaboost.

Adaboost is a discriminative learning algorithm, which performs classification based on a weighted majority vote of weak learners [13]. We use Adaboost learning to construct a cascade of several binary classifier stages $S_1 ... S_N$. The earlier stages in the cascade eliminate many non-vehicle regions with very little processing [8]. The decision rule at each stage is made based on a threshold of scores computed from feature extraction. Each stage of the cascade reduces the number of vehicle candidates, and if a candidate image region survives until it is output from the final stage, it is classified as a positive detection [11].

Upon testing the trained classifier, false positive rates were and high and we found its performance inadequate. We used Adaboost cascade classifier on consecutive frames to detect rear end of vehicles and used the threshold to improve performance detection. Overlapping multiple positive detections in the consecutive frames were unlikely to be false positives and hence considered as true positives, while others were dropped considering them as false positives. We used a threshold of 5 overlapping detection on 8 consecutive frames.

### B. *Vehicle Tracking*

Since tracking based on local optimization of Lucas-Kanade trackers [14][15] may fail due to occlusion or rapid illumination change e.g. when passing under a bridge or entering a tunnel, the need to maintain a temporally consistent model of the environment requires the ability to re-detect a temporarily lost vehicle which in turn requires unsupervised on-line learning of detectors of specific vehicles. Such learning and re-detection capability is provided by the TLD algorithm [16][17].

In TLD, the detector uses the sliding window approach and the object is represented by on-line learnt Randomized Forest (RF). The RF [18]is a set of a restricted class of decision trees called ferns [19] with Haar-like features associated with internal nodes. Observations at internal nodes define a single leaf node in every fern, where an estimate of object vs background likelihood is stored. Initially, the estimates are based on a single example provided by the Adaboost detector.

For each vehicle a new RF is learnt. The RFs consist of 10 ferns each with depth 7 [20], which is a compromise between the speed of evaluation and the discriminative power of the model. Initially, we populate an RF with the positive examples generated by warping the validated object image patch and then negative examples learnt incrementally, as in the TLD, by considering the positive responses of the sliding window detector which are far from object position as the negative examples. Detection is far from the object if the overlap with the object position is less than 0.7. The current object position (provided by the Adaboost detector) is learnt as a positive example. The learning takes place only if at the current position the similarity to the collection of object patches is higher than 0.75, where similarity is measured by the maximum (over patches) of the normalized cross correlation.

## III. Event Detection using 2D Trajectory Classification

After successfully tracking vehicles, the next task is to detect events where target vehicles go over a speed bump or a pot hole. We used a general trajectory classification method [21] which detects the events by comparing the test trajectory to representative trajectories of vehicles traveling over a smooth road. Considering 2D trajectories is attractive since they form computable image features which capture elaborate spatio-temporal information on the tracked vehicles. These trajectories are given as a set of consecutive positions in the image plane $(x,y)$ over time.

We aim at designing a general trajectory classification method which takes into account the trajectory shape, i.e. the geometric information related to the type of motion and to variations in the motion direction, and the speed change of the moving vehicle on its trajectory, i.e. the dynamics related information. It should not be affected by the location of the trajectory in the image plane, i.e. invariance to translation, and by the distance of the tracked vehicle to the camera, i.e. invariance to scale. It should also be robust enough, since local differential features computed on the extracted trajectories are prone to be noise corrupted.

### A. *Trajectory Features*

A feature that represents both trajectory-shape and object acceleration is required to capture the full intrinsic properties of the trajectory. A trajectory $T_k$ is defined by a set of $n_k$ points $\{(x_1,y_1),\ldots,(x_{n_k},y_{n_k})\}$ corresponding to the successive image positions of the tracked point of interest in the image sequence. The term "point of interest" must be understood in a broad sense. For our preliminary study, we considered gravity center of red lights found in the bounding boxes of the tracked vehicle. Alternate points of interest may also be considered. To reliably compute the trajectory features, we need a continuous representation of the curve formed by the trajectory. We perform a kernel approximation of $T_k$ defined by

$$u_t = \frac{\sum_{j=1}^{n_k} e^{-\left(\frac{t-j}{h}\right)^2} x_j}{\sum_{j=1}^{n_k} e^{-\left(\frac{t-j}{h}\right)^2}}, \; v_t = \frac{\sum_{j=1}^{n_k} e^{-\left(\frac{t-j}{h}\right)^2} y_j}{\sum_{j=1}^{n_k} e^{-\left(\frac{t-j}{h}\right)^2}} \tag{1}$$

where $(x_t,y_t)$ designates the coordinates of the tracked object at $t$ and $(u_t,v_t)$ its smoothed representation. $h$ is a smoothing parameter to be set according to the observed noise magnitude. Explicit expressions can then be derived for the first and second order temporal derivatives of the trajectory positions respectively $\dot{u}_t$, $\dot{v}_t$, $\ddot{u}_t$, and $\ddot{v}_t$.

The local orientation of the curve is given by

$$\gamma_t = arctan\left(\frac{\dot{v}_t}{\dot{u}_t}\right) \tag{2}$$

By construction, it is invariant to 2D translation and scale transformation. It can very easily be shown

$$\dot{\gamma}_t = \frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{\dot{u}_t^2 + \dot{v}_t^2} = K_t . \|w_t\| \tag{3}$$

where

$$K_t = \frac{\dot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{(\dot{u}_t^2+\dot{v}_t^2)^{\frac{3}{2}}}, \text{ and } w_t = (\dot{u}_t^2 + \dot{v}_t^2)^2 \quad (4)$$

This local feature well captures both the trajectory shape and the object speed since it is the product of the local curvature ($K_t$) and the instantaneous velocity magnitude ($w_t$)

*B. Trajectory modeling and similarity measure*

We resort to a hidden Markov Model (HMM) to build the statistical framework we need since HMM naturally expresses temporal causality. The feature vector representing a trajectory $T_k$ extracted in a video shot is the vector containing the $n_k$ successive values of$\dot{\gamma}(t)$: $V_k = (\dot{\gamma}_1, \dot{\gamma}_2, \ldots, \dot{\gamma}_{n_k-1}, \dot{\gamma}_{n_k})$.

To determine the HMMs state values, we first study the distribution of $\dot{\gamma}(t)$ on representative trajectories. We define an interval [-*S*, *S*] containing a given percentage $P_v$ of computed $\dot{\gamma}$ values in order to discard "outliers" and to control the number *N* of state values. Hence, a quantization is performed on [-*S*, *S*] into a fixed number *N* of bins

The HMM which models the trajectory $T_k$ is now characterized by:

- the state transition matrix $A = \{a_{ij}\}$ with

$a_{ij} = P[\, q_{t+1} = S_j \mid q_t = S_i \,]$, $1 \le i, j \le N$,

where $q_t$ is the state variable at instant *t* and $S_i$ is its value (i.e., the $i^{th}$ bin of the quantized histogram);

- the initial state distribution $\pi = \{\pi_i\}$, with

$\pi_i = P[\, q_1 = S_i \,]$, $1 \le i \le N$;

- the conditional observation probabilities $B = \{\, b_i(\dot{\gamma}_t) \,\}$, where

$b_i(\dot{\gamma}_t) = P[\, \dot{\gamma}_t \mid q_t = S_i]$,

since the computed $\dot{\gamma}_t$are the observed values.

The conditional observation probability is defined as a Gaussian distribution of mean $\mu_i$ (i.e., the median value of the histogram bin $S_i$). Its standard deviations does not depend on the state and is specified so that the interval $[\mu_i - \sigma,\ \mu_i + \sigma]$ corresponds to the bin width. This conditional observation model can reasonably account for measurement uncertainty. It also prevents from having zero values when estimating matrix *A* in the training stage by lack of measures (especially in case of short trajectories).

To estimate *A* and $\pi$, we have adapted the least squares technique proposed by Ford and Moore [22] where the HMM is assimilated to a count process. If $H_t^{(i)} = P(\, \dot{\gamma}_t \mid q_t = i)$ (corresponding to a weight for the count process), empirical estimates of *A* and $\pi$, for a trajectory *k* of size $n_k$ are given by

$$a_{ij} = \frac{\sum_{t=1}^{n_k-1} H_t^{(i)} H_{t+1}^{(j)}}{\sum_{t=1}^{n_k-1} H_t^{(i)}} \quad \text{and} \quad \pi_i = \frac{\sum_{t=1}^{n_k} H_t^{i}}{n_k} \quad (5)$$

To compare two trajectories, we have to define a similarity measure. We adopt the distance between HMMs proposed by Rabiner [23]. Given two HMMs represented by their parameter sets $\lambda_1$ and $\lambda_1$ ( $\lambda_i = (A_i, B_i, \pi_i)$, $i$ =1,2), the distance *D* is defined by

$$D(\lambda_1, \lambda_2) = \frac{1}{t}\left[\log P\left(O^{(2)} \middle| \lambda_2\right) - \log P\left(O^{(2)} \middle| \lambda_1\right)\right] \quad (6)$$

where $O^{(2)} = \{\, \dot{\gamma}_1, \dot{\gamma}_2, \ldots, \dot{\gamma}_{n_k} \}$ is the sequence of measures used to train the model $\lambda_j$ and P( $O^{(j)}|\lambda_i$ ) expresses the probability of observing $O^{(j)}$ with model $\lambda_i$ (computed with Viterbi algorithm). To be used as a similarity measure, a symmetrized version is required:

$$D_s(\lambda_1, \lambda_2) = \frac{1}{2}\left[D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)\right] \quad (7)$$

*C. Event Detecion*

To detect events of vehicles going over a speed bump or a pot hole, we use supervised learning. We consider two set of predefined classes - {*smooth*, *speed-bump*}, where each class is modeled by a set of HMMs corresponding to representative trajectories. Recognition is then performed by assigning the processed trajectory to the nearest class. The distance between two groups of trajectories $G_i$ and $G_j$ is defined using an average link method, e.g. calculation the mean of the distance between all pairs of trajectories.

$$D_{avg\ link}(G_i, G_j) = \frac{\sum_{T_k \epsilon G_i, T_l \epsilon G_j} D_s(T_k, T_l)}{\#G_i \#G_j} \quad (8)$$

## IV. EXPERIMENTS

To test our detector we ran our experiments on LISA [24] data set. As we were concerned about detection only in first few frames (in our case 7), we have chosen only two performance metrics for the vehicle detector – True positive rate (TPR) and False Detection rate (FDR)

$$TPR = \frac{Number\ of\ Detected\ Vehicles}{Total\ Number\ of\ Vehicles} \quad (9)$$

$$FDR = \frac{Number\ of\ False\ Positives}{Number\ of\ False\ Positives + Number\ Detected} \quad (10)$$

The initial Adaboost cascaded classifier was trained using 2,000 positive training images, and 20,000 negative training images. The classifier was trained for 10 cascade stages. We had a 96% true positive rate, while 22% false detection rate. The 5 threshold detection on 7 consecutive frames as described in section 2.A. was able to bring false detection rate down to near 3%

For uneven road surface detection, authors collected data from few hours of real driving on roads in and around World Trade Center situated at Bangalore, India, using a monocular in-vehicle camera. The data was collected during late morning and late afternoon sessions in month of April. The rear end of vehicles (only hatchback and sedan cars) were extracted using manual annotation from the above mentioned data sets to serve as positive training samples, while negative training set consisted of random non-vehicle images collected from the data set and various other sources including road videos available on Youtube. For uneven road surface detection, real trajectories have been extracted from videos where the ego vehicle is travelling on a smooth road with occasional speed bumps. Videos were manually annotated for ground truth, and trajectories were only computed on 4-6 second duration comprising both before and after the actual event taking place. Trajectories were also computed on 4-6 seconds videos when target vehicles were traveling on a smooth road. The entire video set is 2 hrs in durations with around 26 target vehicles moving over a speed bump at different times and locations.

Tests performed on the above described segmented data gave promising results, with a near perfect classification for most parameter configurations ($N$, $h$ and $P_v$). However, this needs to be further investigated in terms of evaluation on continuous video sets, where captured videos are also influenced by motions of sudden acceleration and braking of ego vehicles, or ego vehicle themselves going over rough road. However to deal such cases, a robust video stabilization hardware and/or algorithm would be required. Many commercial and academic solutions exist like arcadia by SRI [25], Liang et al [26], Matsushita et al [27].

## V. CONCLUSIONS

We have proposed a trajectory-based HMM framework for uneven road surface detection. We have introduced appropriate local trajectory features invariant to translation, and scale transformations. Though the proposed solution has its own limits to finding uneven road surfaces and will not work when no target vehicle is ahead of ego vehicle, we have conducted experiments on real life video examples recorded from an in-vehicle monocular camera and have shown that our method supplies promising results for detection of speed bumps. The proposed solution should also be tested on speed bumps found in first world countries as their shape and size will differ significantly from those found in India.

Extensions of this work will investigate introducing multiple classes for uneven road surfaces like potholes (and possibly with their locations), and discard events detected when the ego vehicle themselves are moving over a speed bump or a pothole. We believe that the algorithm should be general and flexible enough for detection of the above mentioned events. A more comparative study with other vision and non-vision based methods will also be required to evaluate the robustness of proposed solution.

## REFERENCES


[1] "Global Status Report on Road Safety 2013: supporting a decade of action" (PDF) (official report). Geneva, Switzerland: World Health Organisation (WHO). pp. vii, 1–8, 53ff (countries), 244–251 (table A2), 296–303 (table A10). ISBN 978 92 4 156456 4. Retrieved 2014-05-30. Tables A2 & A10, data from 2010.

[2] "Road Accidents in India 2013" (PDF). New Dehli: Ministry of Road Transport and Highways Transport Research Wing, Government of India. August 2014. pp. 2, 5–7. Retrieved 2015-01-14.

[3] A. Barth and U. Franke, "Tracking oncoming and turning vehicles at intersections," in *Proc. 13th Int. IEEE ITSC*, Sep. 2010, pp. 861–868.

[4] S. Sivaraman and M. M. Trivedi, "Combining monocular and stereovision for real-time vehicle ranging and tracking on multilane highways," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2011, pp. 1249–1254.

[5] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, and W. Rosenstiel, "Object-oriented Bayesian networks for detection of lane change maneuvers," in *Proc. IEEE IV*, Jun. 2011, pp. 673–678.

[6] S. Sivaraman, B. T. Morris, and M. M. Trivedi, "Learning multi-lane trajectories using vehicle-based vision," in *Proc. IEEE Int. Conf. Computer. Vision Workshop*, 2011, pp. 2070–2076.

[7] Satzoda, Ravi Kumar, and Mohan M. Trivedi. "Overtaking & receding vehicle detection for driver assistance and naturalistic driving studies." In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pp. 697-702. IEEE, 2014.

[8] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I-511. IEEE, 2001.

[9] Haselhoff, Anselm, Sam Schauland, and Anton Kummert. "A signal theoretic approach to measure the influence of image resolution for appearance-based vehicle detection." In *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 822-827. IEEE, 2008.

[10] Ponsa, Daniel, Antonio López, Felipe Lumbreras, Joan Serrat, and Thorsten Graf. "3D vehicle sensor based on monocular vision." In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 1096-1101. IEEE, 2005

[11] Wender, Stefan, and Klaus Dietmayer. "3D vehicle detection using a laser scanner and a video camera." *IET Intelligent Transport Systems* 2, no. 2 (2008): 105-112.

[12] Sivaraman, S, and Mohan M. Trivedi. "Active learning based robust monocular vehicle detection for on-road safety systems." In *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 399-404. IEEE, 2009.

[13] Freund, Yoav, Robert Schapire, and N. Abe. "A short introduction to boosting."*Journal-Japanese Society For Artificial Intelligence* 14, no. 771-780 (1999): 1612

[14] Vojır, Tomaš, and Jirı Matas. "The Enhanced Flock of Trackers." *Registration and Recognition in Images and Videos* 532 (2014): 113.

[15] Lucas, Bruce D., and Takeo Kanade. "An iterative image registration technique with an application to stereo vision." In *IJCAI*, vol. 81, pp. 674-679. 1981

[16] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*34, no. 7 (2012): 1409-1422.

[17] Kalal, Zdenek, Jiri Matas, and Krystian Mikolajczyk. "Pn learning: Bootstrapping binary classifiers by structural constraints." In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 49-56. IEEE, 2010.

[18] Breiman, L. "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.

[19] Ozuysal, Mustafa, Michael Calonder, Vincent Lepetit, and Pascal Fua. "Fast keypoint recognition using random ferns." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, no. 3 (2010): 448-461.

[20] Caraffi, Claudio, Tomas Vojir, Jura Trefny, Jan Sochman, and Jirı Matas. "A system for real-time detection and tracking of vehicles from a single car-mounted camera." In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 975-982. IEEE, 2012.

[21] Hervieu, Alexandre, Patrick Bouthemy, and Jean-Pierre Le Cadre. "Video Event Classification and Detection Using 2D Trajectories." In *VISAPP (2)*, pp. 158-166. 2008.

[22] Ford, Jason J., and John B. Moore. "On adaptive HMM state estimation."*Signal Processing, IEEE Transactions on* 46, no. 2 (1998): 475-486.

[23] Rabiner, Lawrence. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77, no. 2 (1989): 257-286.

[24] Sivaraman, S, and M. M. Trivedi. "A general active-learning framework for on-road vehicle recognition and tracking." *Intelligent Transportation Systems, IEEE Transactions on* 11, no. 2 (2010): 267-276.

[25] http://www.sri.com/engage/products-solutions/acadia-video-processors

[26] Liang, Yu-Ming, Hsiao-Rong Tyan, Shyang-Lih Chang, Hong-Yuan Mark Liao, and Sei-Wang Chen. "Video stabilization for a camcorder mounted on a moving vehicle." *Vehicular Technology, IEEE Transactions on* 53, no. 6 (2004): 1636-1648.

[27] Matsushita, Yasuyuki, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. "Full-frame video stabilization with motion inpainting." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 7 (2006): 1150-1163.