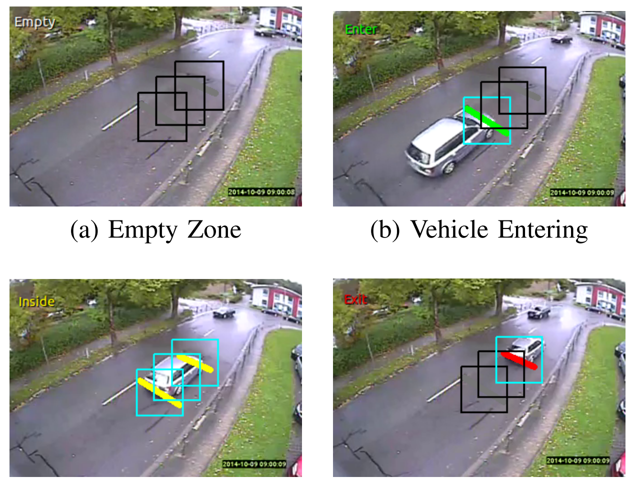
A Hidden Markov Model for Vehicle Detection and Counting

Nicholas Miller, Mohan A. Thomas, Justin A. Eichel, Akshaya Mishra
Miovision Technologies Inc.
148 Manitou Drive, Suite 101
Kitchener , Ontario, N2C 1L3
Email: {nmiller, mthomas, jeichel, amishra}@miovision.com

***Abstract*—To reduce roadway congestion and improve traffic safety, accurate traffic metrics, such as number of vehicles travelling through lane-ways, are required. Unfortunately most existing infrastructure, such as loop-detectors and many video detectors, do not feasibly provide accurate vehicle counts. Consequently, a novel method is proposed which models vehicle motion using hidden Markov models (HMM). The proposed method represents a specified small region of the roadway as 'empty', 'vehicle entering', 'vehicle inside', and 'vehicle exiting', and then applies a modified Viterbi algorithm to the HMM sequential estimation framework to initialize and track vehicles. Vehicle observations are obtained using an Adaboost trained Haar-like feature detector. When tested on 88 hours of video, from three distinct locations, the proposed method proved to be robust to changes in lighting conditions, moving shadows, and camera motion, and consistently out-performed Multiple Target Tracking (MTT) and Virtual Detection Line (VDL) implementations. The median vehicle count error of the proposed method is lower than MTT and VDL by 28%, and 70% respectively. As future work, this algorithm will be implemented to provide the traffic industry with improved automated vehicle counting, with the intent to eventually provide real-time counts.**

***Keywords*-hidden Markov models, vehicle detection, vehicle tracking, machine learning, Haar-like features, Intelligent Transportation Systems**



(a) Empty Zone (b) Vehicle Entering

(c) Vehicle Fully Inside (d) Vehicle Exiting

Figure 1. Example sequence showing an overview of vehicle detection and tracking in a "zone" (denoted by two bars perpendicular to vehicle motion). A state machine is used to track when the zone is empty, when a vehicle enters, passes inside, and exits the zone. The states are determined in sequence using a series of three object classifier boxes for observations. A trained Haar like feature object detector [4] provides vehicle observations to the boxes. Boxes appearing light blue indicate positive vehicle classification. Sequential state estimation is done using a hidden Markov model.

I. Introduction

Intelligent transportation systems (ITS) are increasingly reliant on sophisticated applications of sensing and control, including live real-time video analytics. Reliable real-time vehicle traffic information is essential for an effective implementation of ITS to ease congestion and reduce delays. A desirable measurement for predicting demand and efficiently controlling traffic light timing and routing is to count the total volume of vehicles over time through each intersection leg [1], [2]. For full integration into an effective intelligent transportation system, reliable volume counts should be provided online in real-time.

Traditional traffic systems merely adapt to traffic demand using simple traffic light actuation based on sensed vehicle presence. While inductive metal loops are a commonly used technology for presence detection, video has more recently provided a non-invasive technique for traffic sensing. Existing actuated traffic light infrastructure typically provides local presence detection only. Rudimentary post-processing is used for deriving volume counts, if volume counts are provided at all. Video technology has the potential to provide both verifiable online volume counts and local actuation presence information, and could be deployed en masse to integrate the ITS network of an entire city.

A. Application context

This paper addresses the application of video analysis to provide reliable online vehicle volume counts through user specified regions of interest, often referred to as "zones". See Fig. 1 for an example of a configured zone. The goal is to measure real world accuracy and reliability to assess suitability for deployment in an ITS system with realistic challenges [3], including changing lighting conditions, partial occlusions, low bitrate ($\approx$ 100 kbps), and low resolution ($\approx$ 320 $\times$ 240) video from uncalibrated cameras with a variety of perspectives and road geometries.

978-1-4799-1986-4/15 $31.00 © 2015 IEEE
DOI 10.1109/CRV.2015.42

B. Outline

This paper is organized as follows: in Section II, a brief overview of current approaches in video-based volume counting is described. This is followed by a detailed description of the problem in Section III. Section IV outlines contributions and innovations. The proposed approach has been tested on an experimental dataset and is compared with alternate approaches in Section V. Finally, Section VI contains a discussion of limitations and avenues for future work concluding in Section VII.

II. Background

A. Inductive loop detectors as real time counters

A very common vehicle detection technology consists of a loop of conductive material under the road used to measure inductance. Large metal-framed vehicles stationed over the loop are easy to identify in the inductance signal. Such vehicle detection is easily readable in real-time with minimal delay, and with a high signal to noise ratio. Unfortunately, inductive loops have several major drawbacks, the most significant of which is the high cost of installation and maintenance. It is costly and disruptive to dig up pavement to install a loop for every lane that requires detection. Loops are highly sensitive to configuration and placement, and there is a high cost associated with adjustments. In regions with freezing winters, the freeze-thaw cycle can substantially damage in-ground detectors: according to [1], in 1991, most cities in the U.S. reported 25% to 30% of their detectors were not functional at any given time.

Loop detectors are usually directly wired for presence detection and responsive actuation. The data is not human verifiable. Their limited sensitivity and fixed configuration means loops are unsuitable to adapt to sensing pedestrians, bicycles and some smaller vehicles such as motorcycles. Since only inductance is measured, traffic parameters including the vehicle volume count are not directly observable, and various ad-hoc inference methods must be used. Despite their limitations, loops are still the most prominent vehicle sensor.

B. Video-based vehicle detection

Video-based vehicle detection is attractive for its low relative cost and ease of installation and maintenance [3]. Modern connected video detection systems offer the ability to capture, store, and process video surveillance and analytics for various applications. Vehicles, cyclists, pedestrians, and other road users appear in the image and can be incorporated into detection and scene understanding algorithms for ITS applications. Video processing has been used for various traffic sensing applications, including vehicle counting and presence detection. Despite advances in research, real world accuracy and reliability have been found lacking, especially in adverse environmental conditions [5].

1) Background subtraction and blob tracking: A common approach to vehicle detection in video is simple moving object detection using background subtraction [1], [6]. The background of the traffic scene is learned and maintained using a background modeling process, such as mixture of Gaussians [7]. Moving objects in the current frame are identified by subtracting the background and applying morphology operations for smoothing [8]. Connected components [9, p. 317] is used to segment individual moving blobs, which can then be tracked with a Kalman Filter [10, p. 584]. Background subtraction is subject to failures due to camera motion and moving shadows. In stop and go traffic, stopped vehicles may get absorbed into the background image, resulting in detection misses, as well as spurious ghosting [3] when the vehicle leaves.

In order to perform multiple target tracking on vehicle blobs, the data association problem [9, p. 388] must be solved, and long term tracks must be correctly initialized and deleted as vehicles enter and exit the scene. The most effective approaches to data association for vehicle tracking - including multiple hypothesis tracking [11], particle filtering, and Monte Carlo Markov chains (MCMC) [8] - have a high computational cost, and are not suitable for real-time online implementation. Nearest neighbour data association is often used in real-time multiple target tracking despite high potential for data association failures [11] and is subject to error due to failures in background subtraction.

2) Virtual Detection Line: A more straightforward approach for vehicle counting is the Virtual Detection Line (VDL) [2]. In this approach, a virtual line is drawn on the image perpendicular to the flow of traffic. The pixel intensities from video are captured along the line and extruded over time to generate a time-spatial image (TSI). Vehicles passing over the line in the video appear in the TSI at an offset corresponding to the time they passed over the VDL. Counting vehicle volume is a matter of segmenting the appearances of individual vehicles in the TSI; in [2] edge finding and morphology operations are used to generate the segmented mask for counting. Generally, this segmentation is sensitive to shadows, variability in lighting conditions, occlusions and variability in vehicle speed, especially when vehicles are stopped. Multiple detection lines combined with correspondence has been employed to overcome these limitations and increase accuracy [2]. TSI based counting tends to be rule-based with no prescribed way to use machine learning to improve accuracy given previous data. Further, the authors are not aware of any work implementing this multi-VDL solution in an online fashion, nor have solutions been posed for simultaneously estimating other reliable real-time traffic information such as vehicle presence.

C. Adaboost and Haar features

Supervised machine learning for object detectors based on image appearance has shown promising results [12].

Learning processes such as neural networks and boosted classifiers [12] have a large computational cost and memory footprint but real-time detection and classification becomes feasible given a properly trained classifier.

Detecting and classifying vehicles based on Adaboost trained Haar-like feature sets is now a commonplace method for detecting vehicles in images [13], [4]. Such a classifier is robust to lighting changes and camera movement, but image artifacts from low bitrate videos may result in many false positives and false negatives. This should be mitigated using scene information and temporal motion information [3], [4]. Detections must be properly associated in time to count vehicle volumes.

III. Problem Formulation

The problem being explored can be stated as follows: develop an algorithm to accurately count vehicles moving through a configured video zone in real time.

The system must be robust to:

1) noise and artifacts from video encoding compression: this compression enables video data to be quickly transferred and easily stored;
2) use in stop-and-go traffic: at intersections, vehicles may be completely stationary for extended periods;
3) vehicles from adjacent lanes travelling in the opposite direction: depending on the camera perspective, vehicles from adjacent lanes may partially or fully occlude the lane of interest.

Occlusions and bumper-to-bumper traffic are not directly considered for this work outside of Section VI where further enhancements are suggested.

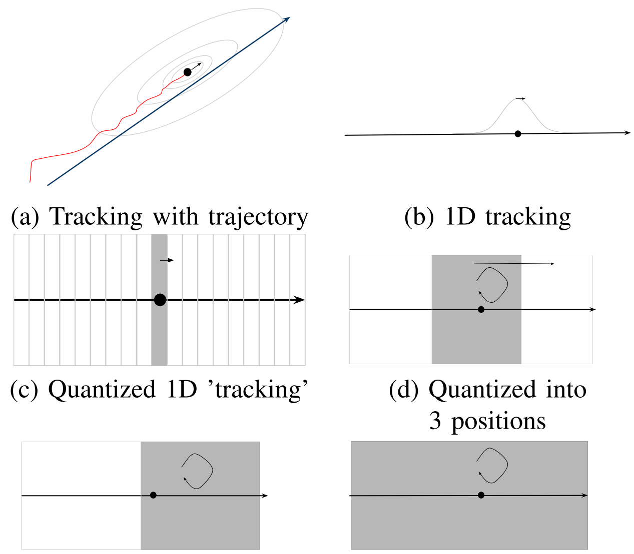
IV. Contribution: Solution and Innovations

The contribution is the development of a model and method for coarsely tracking vehicle movement using discrete states in a hidden Markov model framework [14] to reliably generate vehicle count data.

Tracking can be used to count vehicles by temporally associating candidate vehicle detections with unique vehicle tracks. However, tracking approaches generally have superfluous degrees of freedom which allow them to precisely localize objects. In addition, initialization and termination of object tracks may necessitate the use of ad-hoc rules and heuristics. In the counting application, a much simpler model is sufficient. The objective is only to keep track of how many unique vehicles have passed through a specific detection zone. Precise vehicle trajectories are neither relevant nor necessary. Instead, simple discrete values can be used to model the passage of individual vehicles.

A. Model development: quantization of tracker state space

Kalman filters are a popular way to filter noise in object tracks [9, p. 380]. This technique models the passage of vehicles using a continuous observation space corresponding to the positions of individual vehicles with a continuous linear motion model. Noisy observations may lie anywhere in the image and a meaningful trajectory is filtered out (Fig. 2a). However, as vehicles being tracked are all moving in the same direction, the most meaningful information from tracking is the position of the vehicle along the axis of the lane (Fig. 2b).



(a) Tracking with trajectory (b) 1D tracking (c) Quantized 1D 'tracking' (d) Quantized into 3 positions (e) 2 positions (f) Single position

Figure 2. (a) Tracking can locate and predict object trajectories through continuous space. Constrained motion for vehicles reduces the problem to 1D tracking (b). (c) Quantizing the space may reduce precise trajectory information but reduces the complexity and parameters necessary to track, particularly for multiple targets. (c-f) There is a trade off between the number of quantized values and model parameters. Too few values and failure to capture desirable motion parameters results. (f) One position value hides directionality, where observing only 2 quantized positions fails to capture accelerations (e). This work uses 3 quantized positions (d).

Consider now a discrete representation of one dimension of travel. If a vehicle's position can be coarsely estimated to quantized positions along the axis of travel, it can be represented using a simple state space. For the purpose of this paper, we will use the three value position quantization shown in Fig. 2d. The following states are defined:

1) s_{empty}: no vehicle is present in the zone,
2) s_{enter}: a vehicle has appeared at the beginning of the zone but is not entirely inside,
3) s_{inside}: a vehicle is centered entirely inside the zone
4) s_{exit}: a vehicle still appears partially inside the zone but is vacating.

Fig. 3 illustrates a learned model of how the states transition between each other to capture the passage of a vehicle in a particular direction. This state machine extends [1] and is akin to a motion model for tracking. Note that when it is assumed that at most one vehicle appears in a selected region of interest at a time, the empty state is allowed and provides

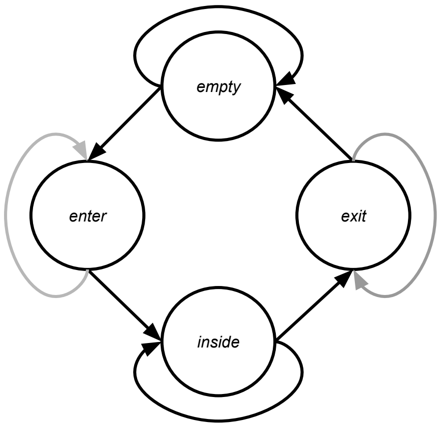


Figure 3. State machine describing vehicle passage through a zone in video. These transitions were learned as described in Section V-B4.

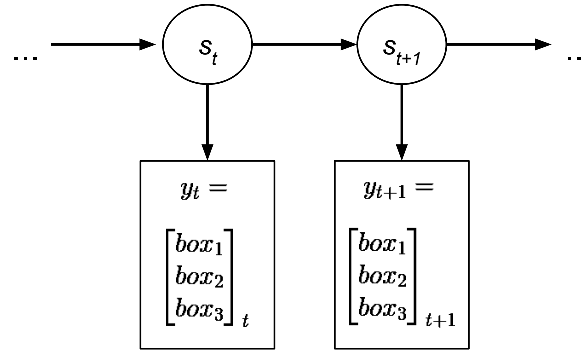


Figure 4. A hidden Markov model process describing the progression of vehicles and detector observations in a zone: t is the current time, $s_t \in \{s_{\text{empty}}, s_{\text{enter}}, s_{\text{inside}}, s_{\text{exit}}\}$, y_t are observations, $\text{box}_i \in \{0, 1\}$. The arrows indicate statistical dependence, so s_{t+1} depends only on s_t (Markov assumption) and y_t depends only on s_t and y_{t+1} depends only on s_{t+1}.

a means to systematically incorporate vehicle initialization and deletion in the same model as vehicle tracking.

B. Observations

Vehicle observation inputs for tracking algorithms are generated in a variety of ways including background subtraction [7] and object detection [4]. These observations, which lie in a continuous space, may be noisy. Given the coarse state model in Fig. 3, a continuous observation space is excessive. Instead of observations lying in a continuous space, 'boxes' along the zone are used as discrete measurement points (Fig. 1). In this work, three measurement boxes denoted box_1, box_2, and box_3 are used. Measurements are the binary result of the trained classifier [3] — they either contain a candidate vehicle, or they don't. These measurements may be noisy due to spurious detections and misses similar to tracker observation noise.

If the zones are small enough, the possibility of two vehicles being present at once becomes unlikely. However, they must extend enough for three boxes to clearly see an activation trend moving from the first box through the middle one to the last one.

Given a vehicle passing through a zone and perfect detections, the following state and observation sequences would likely occur:

$$\begin{aligned} &\{s_{\text{empty}}, s_{\text{enter}}, s_{\text{inside}}, s_{\text{inside}}, s_{\text{exit}}, s_{\text{empty}}\} \\ &\Rightarrow \{y_t, y_{t+1}, y_{t+2}, y_{t+3}, y_{t+4}, y_{t+5}\} \\ &\quad = \{\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T, \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T, \begin{bmatrix}1 & 1 & 0\end{bmatrix}^T, \\ &\qquad \begin{bmatrix}0 & 1 & 1\end{bmatrix}^T, \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T, \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T\} \end{aligned} \tag{1}$$

C. State estimation using hidden Markov models

Transition probabilities maybe learned as seen in Fig. 3, and each state has associated probabilities of emitting each of the 8 possible observation values. See Fig. 4. These provide discrete analogs to the continuous tracking process: transitions are analogous to the tracker motion model, and emissions are analogous to detections with noise.

$$S = \{s_{\text{empty}}, s_{\text{enter}}, s_{\text{inside}}, s_{\text{exit}}\} \tag{2}$$

$$y = \begin{bmatrix}\text{box}_1, \text{box}_2, \text{box}_3\end{bmatrix}^T \tag{3}$$

$$a_{ij} = P(s_{t+1} = j | s_t = i) \text{ for } i \in S, j \in S \tag{4}$$

$$b_j(y) = P(y_t = y | s_t = j) \text{ for } j \in S \tag{5}$$

A hidden Markov model is fully specified by a state transition matrix, and emission matrix [14]. These values may trained or estimated from real data [10].

D. Modified Viterbi for vehicle counting

The Viterbi algorithm [14] is often used to estimate a sequence of hidden true states given observations only. Building on [15] we use a modified Viterbi algorithm to estimate vehicle counts directly from box observations.

The Viterbi algorithm employs dynamic programming to estimate the most probable state sequence, given an observation sequence. It is a discrete analogy to multiple hypothesis tracking but with a tractable recursive definition [11], [14]. The algorithm defines the standard recursive probabilities, [14]

$$\delta_t(j) = \max_{i \in S} \delta_{t-1}(i) a_{ij} b_j(y_t) \tag{6}$$

$$\Psi_t(j) = \arg\max_{i \in S} \delta_{t-1}(i) a_{ij}. \tag{7}$$

Now vehicle count, C^* may be found using a count increment τ (i.e., a transition from s_{empty} to s_{enter}),

$$\tau(i,j) = \begin{cases} 1 & : (i,j) = (s_{\text{empty}}, s_{\text{enter}}) \\ 0 & : otherwise \end{cases} \tag{8}$$

$$C_t(j) = C_{t-1}(\Psi_t(j)) + \tau(\Psi_t(j), j) \tag{9}$$

$$s^* = \arg\max_{j \in S} \delta_t(j) \tag{10}$$

$$C^* = C_t(s^*). \tag{11}$$

This provides vehicle counts similar to a multiple hypothesis tracking approach, but with a single systematic model for handling noisy, spurious, and missing observations, as well as initializing and terminating vehicles.

V. Experimental Results

A. Experiment settings

The proposed method has been compared to multiple target tracking and VDL on a corpus of 88 hours of real world traffic video over a variety of changing environmental conditions such as wet, snow, day, night, dusk and dawn. This corpus is meant to be challenging and includes five different lane and view geometries including partial occlusions in addition to camera motion. All videos are low resolution (342×228 @ 30 fps) encoded at 95 kbps in h.264 to reflect the encoding necessary to maintain a constant video flow through a realistic ITS deployment.

For each lane in the dataset a lane mask is selected and a zone is configured by two lines perpendicular to vehicle flow, one indicating where vehicles enter the zone and one where they leave; see Fig. 5. This zone is manually entered so that it represents the size of a vehicle and typically only one vehicle is inside the zone at a time.

B. Implementation

1) Multiple target tracking: The MATLAB Computer Vision Toolbox's multiple target Kalman tracker [16] was employed within a lane mask in each video. Moving vehicles are detected and segmented from the background using a mixture of Gaussians background model, background subtraction, open and close morphology operations and connected components blob identification. The multiple target tracking uses Kalman filters with a linear motion model and constant width and height blob properties. The Munkres algorithm and gating [9, p. 390] provides data association and thresholding rules on associations, and missing detections are used to initialize and delete vehicle tracks. The resulting tracks provide vehicle counts. This method is not overly sensitive to threshold parameters so the default parameters were sufficient [7].

2) VDL: The line specifying the end of the zone served as a single virtual detection line, and online segmentation provided counts for the VDL algorithm.

3) HMM: The configured zone immediately specifies the location of the three observation boxes in the proposed HMM model as well as the specifying the zone where the state tracking applies. The counts were found using the aforementioned modified Viterbi algorithm.

4) Training: A training set of 50 minutes from additional videos not present in the testing corpus with various different geometries and conditions was used to perform supervised training for the HMM. Human annotation was used to precisely label the true state as s_{empty} s_{enter}, s_{inside}, or s_{exit} in each frame of the training set. The detector observations and true states were used to estimate the maximum likelihood transition and emission matrices [10]. Fig. 5 shows the tool used to label the training video. The decoded observations are overlayed in the three boxes for the given zone. The classifier training was described in [3].

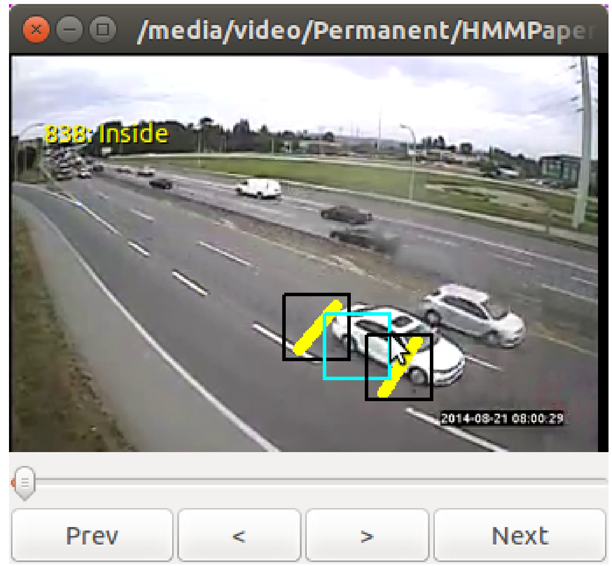


Figure 5. A tool developed for users to enter the state of a zone on every frame of video in a training set. On this particular image, it is the user who manually indicates that the vehicle is inside the zone for training purposes. This helps correspond the box observation emission (seen here as $[0\ 1\ 0]^T$ in light blue) with the s_{inside} state. The transition probabilities are learned by state changes between successive frames.

C. Ground truth

Ground truth vehicle counts were provided by a verified manual count process. For each lane, ground truth is collected by humans and aggregated into five minute bins, for a total of 1056 bins. The human counters play through each test video and indicate precisely when a new vehicle enters each lane. The count of the corresponding five minute bin for the lane is incremented. A second person counts an additional 12% of each hour to ensure agreement within 5% relative error and within 5 vehicles of absolute error.

D. Error calculation

Vehicle counts for each algorithm are compared to the ground truth bin counts. For each bin, the absolute relative error is measured:

$$error(bin_i) = \frac{|true\_count_i - alg\_count_i|}{true\_count_i}$$

Where bin_i is an index into the i^{th} 5 minute bin in the corpus; $true\_count_i$ is the vehicle count provided by humans for bin_i; alg_count_i is the count provided by the algorithm being measured. Since the focus is on a complete picture of the performance level of vehicle counting when deployed

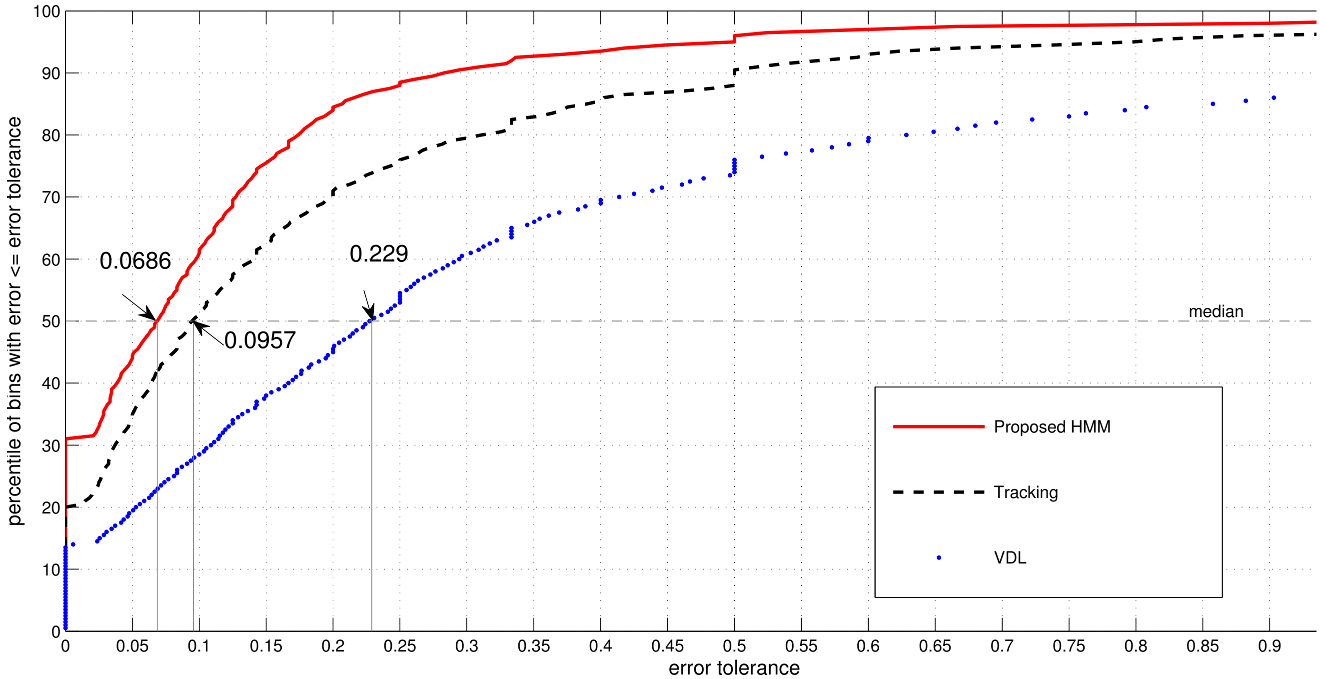


Figure 6. Error rates for three different counting algorithms. Error percentiles are used to show the complete picture of performance for each algorithm. The horizontal axis is the error tolerance and the vertical axis shows the percentile of five-minute-bins which have error less than or equal to the tolerance. An ideal algorithm would jump to the top left corner. The **median error rate** is given at the 50th percentile, so the median relative five-minute-bin-error for **multiple target tracking (black) is 0.0957** (comparable to previous results on challenging video [5], [6]) while the median achieved by the **proposed method (red) is 0.0686**. The VDL has many difficulties due to occlusion, camera motion, shadows, and changing lighting conditions so the median error for **VDL (blue) is 0.2290**.

in a real ITS system, percentile curves of five-minute-bin-errors, in addition to median error rates, are reported for each algorithm. An error percentile is the proportion of bins in the entire corpus with a measured relative error less than or equal to a given error tolerance:

$$\frac{\#\{bin_i : error(bin_i) \leq error\_tolerance\}}{1056} \times 100\%$$

E. Counting results comparison

Fig. 6 shows the results of the VDL, multiple target tracking, and HMM on the dataset described above. The error percentiles are plotted for each algorithm. The error percentile (vertical axis) is a measure of reliability: it gives the probability that a five minute bin count is less than or equal to a specified error tolerance (horizontal axis). For example, a perfect vehicle counter would have 0 error in all bins, so the percentile of bins with error rate less than or equal to 0 is 100% and the curve would appear as a step immediately from the bottom left corner to the top left corner and flat across.

On this 88 hour video dataset, the proposed method achieved a median five minute bin error of 0.0686 for this counting task while the multiple target tracking and VDL implementations had median errors of 0.0957 and 0.2290 respectively.

VI. Discussion

The advantages of vehicle counting with a simple state machine for object detection and counting compared to a multiple target motion tracker are evident in the results. Although multiple target tracking attempts to provide more precise information on vehicle location and motion, which is useful for classifying trajectories and events, track initialization and deletion, data association, and long term tracking failure can result in erroneous volume counts. The proposed HMM takes advantage of constrained vehicle motion to perform the counting task in selected zones. Since a trained object detector is used for measurement, there is less sensitivity to camera motion, shadows, and lighting changes than with VDL or the background subtraction used in multiple target tracking. The constrained vehicle motion captured in the proposed HMM transition model provides enough temporal information to be robust to detection errors. See Fig. 8.

A. Limitations

This HMM approach fails to model a few aspects of the scene leading to some failures, particularly due to large vehicles out of the scale of the configured zone, and bumper to bumper vehicle traffic placing two vehicles in a zone at the same time. Removing the Markov assumption for the state machine would allow it to take advantage of more sophisticated temporal information such as recent vehicle arrival and velocity rates albeit with a higher computational load.

Note that detections due to occluding vehicles in an opposing lane are addressed in the current model only by the emission probabilities. The HMM training establishes that measurements of occluding vehicles could appear in all possible states, particularly s_{empty}, and the transition model of the state machine is relied upon to correctly track only vehicles travelling along the constrained motion path through the zone. A better approach is to explicitly model occluding vehicles with states travelling in the opposite direction as demonstrated in Fig. 7.

B. Future work

The above problems will be addressed in future work. Scale selection should be used in the object detection process to provide proper presence measurements to the HMM for large or small vehicles of any kind. Image segmentation could be incorporated to indicate the presence of one vehicle or two bumper-to-bumper vehicle measurements. This can be incorporated into a larger state model with joint states for 2 or more vehicles with corresponding joint vehicle transitions, e.g. for vehicles V_1 and V_2,

$$\left(s_{\text{enter}}^{V_1} \times s_{\text{exit}}^{V_2}\right) \rightarrow \left(s_{\text{inside}}^{V_1} \times s_{\text{empty}}^{V_2}\right).$$

Simultaneous occluding vehicles could be incorporated similarly to Fig. 7.

Finally, presence detection is another important task for real-time ITS. Fixed lag-smoothing [10, p. 580] provides a method to derive a smooth presence signal from the proposed HMM. A time delay window allows the model to smooth object detections, with a trade off of higher accuracy but larger memory usage vs. delay length.

VII. Conclusions

A state tracking method has been proposed to take advantage of constrained vehicle motion to detect and count vehicles using a hidden Markov model. Observations for the model are provided by a trained Haar feature vehicle detector. These detections are robust in stop and go traffic and changes in lighting and camera motion, which would normally interfere with motion-based vehicle detection and counting. This method has been shown to give significantly better vehicle volume counts than both multiple target moving object tracking and VDL on a dataset of over 88 hours of video. On this testing set, the proposed method achieved a median 5-minute-bin error of 0.0686 for this counting task while the multiple target motion tracking and VDL implementations had median errors of 0.0957 and 0.2290 respectively. The proposed method was also more reliable having fewer and less severe occurrences of 5-minute-bin errors throughout the testing set.

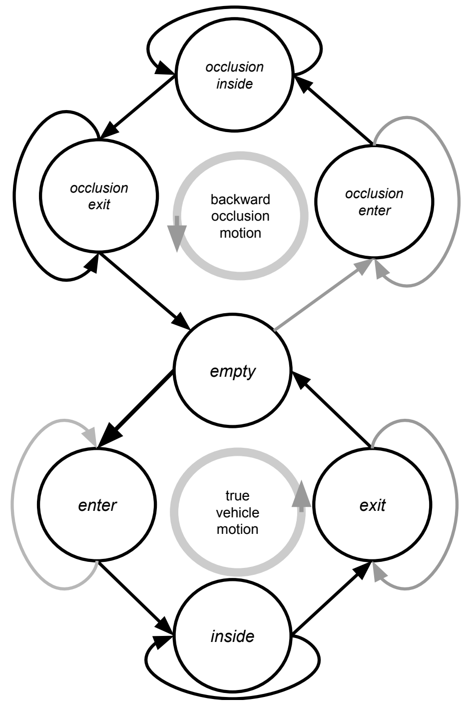


Figure 7. An extended state machine which explicitly models occluding vehicles travelling backwards through the zone. The box observations for occluding vehicles may resemble the measurements of the 'complementary' forward state. The parameters of this model could be also be trained by explicitly labelling occluding vehicles.

Acknowledgment

The authors would like to thank the entire Miovision team for all their hard work and help with data, implementations, and suggestions. Special thanks to David Swart, Joel Keller, and Tyler Abbott for comments and editing.

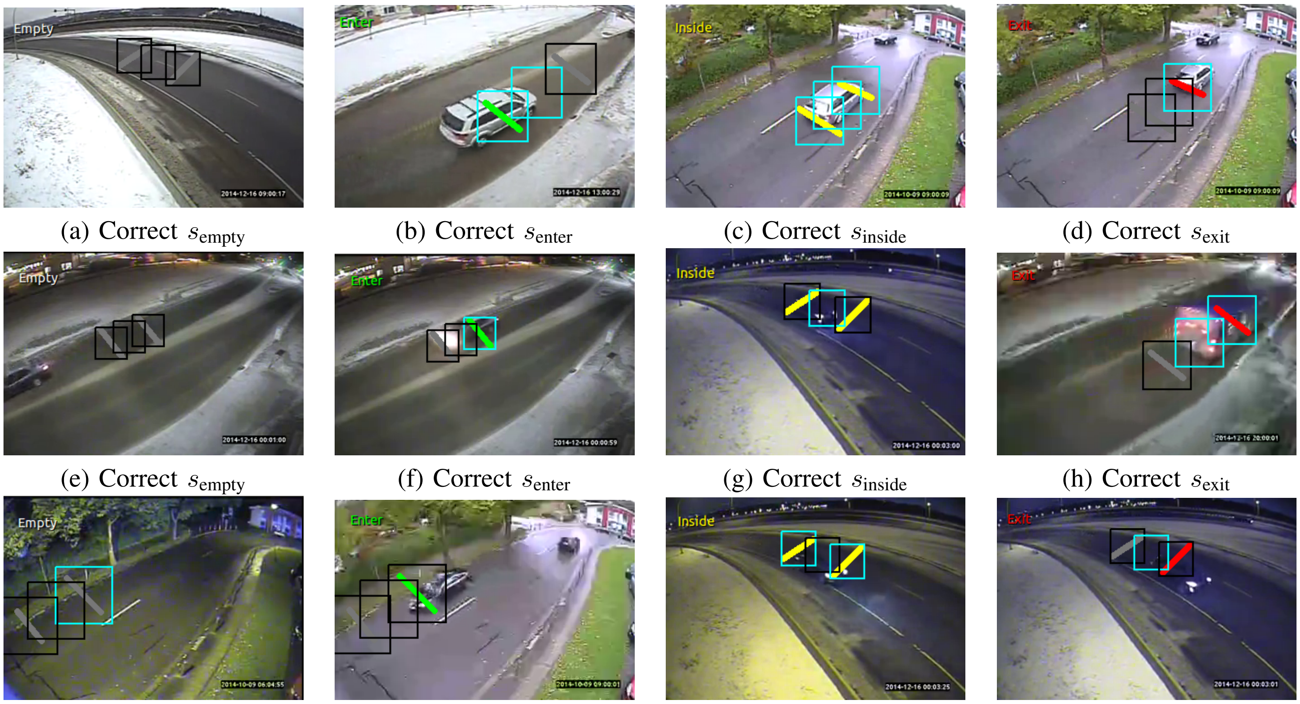
References

[1] P. Michalopoulos, "Vehicle detection video through image processing: the autoscope system," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 21–29, Feb 1991.

[2] N. Mithun, N. Rashid, and S. Rahman, "Detection and classification of vehicles from video using multiple time-spatial images," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1215–1225, Sept 2012.

[3] J. Eichel, A. Mishra, N. Miller, N. Jankovic, and K. McBride, "Decoding the spatiotemporal scene of a road-traffic intersection for real time event detection," in *IEEE Conf. Comput. Vis. Pattern Recog. Scene Understanding Workshop*, Columbus, OH, June 2014.

[4] S. Sivaraman and M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 906–917, June 2013.



(a) Correct s_{empty} (b) Correct s_{enter} (c) Correct s_{inside} (d) Correct s_{exit}

(e) Correct s_{empty} (f) Correct s_{enter} (g) Correct s_{inside} (h) Correct s_{exit}

(i) Correct s_{empty} (j) Correct s_{enter} (k) Correct s_{inside} (l) Correct s_{exit}

Figure 8. Example images showing results of HMM sequence labelling from the system. Images (a-d) demonstrate correct labels with typical detection box observations (light blue) for their respective states; (d-h) show alternative but still common box detections with state labels correctly identified at night. In (i) the correct state is obtained despite the feature detector giving a false positive due to edge artifacts; (j) contains a video compression artifact leading to detection misses; (k) and (l) demonstrate false negatives for dark patches at night. Figures (i-l) are interesting because they demonstrate how the state machine and transitions from the HMM model lead to correct state labels in the presence of detector failures.

[5] J. C. Medina, R. F. Benekohal, and M. V. Chitturi, "Evaluation of video detection systems volume 2-effects of illumination conditions in the performance of video detection systems," Illinois Center for Transportation, Tech. Rep. ICT-09-046, 2009.

[6] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar 2002.

[7] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," vol. 2, 1999, pp. –252 Vol. 2.

[8] K. McBride, "Vehicle tracking in occlusion and clutter," M.A.Sc. Thesis, University of Waterloo, Waterloo, Ontario, Canada, 2007.

[9] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[10] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, ser. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[11] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan 2004.

[12] S. Sivaraman and M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec 2013.

[13] A. Haselhoff and A. Kummert, "An evolutionary optimized vehicle tracker in collaboration with a detection system," in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, Oct 2009, pp. 1–6.

[14] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[15] G. Polatkan and O. Tuzel, "Compressed inference for probabilistic sequential models," *CoRR*, vol. abs/1202.3759, 2012. [Online]. Available: http://arxiv.org/abs/1202.3759

[16] MATLAB, *version 8.2.0.701 (R2013b)*. Natick, Massachusetts: The MathWorks Inc., 2013.