Full-Body Occlusion Handling and Density Analysis in Traffic Video-Surveillance Systems

Authors:

Evangelos Palinginis, M.Sc\(^1\)*
Graduate Research Assistant, e_palinginis@gatech.edu

Keitaro Kamiya, M.Sc\(^1\)
Graduate Research Assistant, keitarokamiya@gatech.edu

Jorge Laval, Ph.D.\(^1\)
Associate Professor, jorge.laval@ce.gatech.edu

Ioannis Brilakis, Ph.D.\(^2\)
Laing O’Rourke Lecturer, ib340@cam.ac.uk

Randall Guensler, Ph.D.\(^1\)
Professor, randall.guensler@ce.gatech.edu

\(^1\)Phone: 404-894-2201 / Fax: 404-894-2278
School of Civil and Environmental Engineering
Georgia Institute of Technology
790 Atlantic Drive, Atlanta, GA 30332

\(^2\)Phone: +44-(0)-1223 332718
Department of Engineering
University of Cambridge
Trumpington Street, Cambridge,
Cambridgeshire CB2 1PZ

* Corresponding author

Word Count: 7365 (text: 6865; tables: 1 x 250: 250; figures: 1 x 250: 250)
Submission Date: August 01, 2013
ABSTRACT

Vision-based traffic surveillance systems are amidst the most reliable, inexpensive and highly applicable methodologies for surveying traffic conditions. The implementation of these strategies, however, is limited under certain conditions, such as the presence of vehicle occlusions or poor illumination conditions that lead to either over-counted or undercounted traffic data. The proposed motion-based methodology aims at overcoming these limitations by employing a new technique for full-body occlusion handling of vehicle cars. The methodology is based on five main steps and three main methods: Background Subtraction, Histogram of Oriented Gradients (HOG) trained by linear Support Vector Machine (SVM), Haar-Like features (HL) trained by Adaboost and Vehicle Counting. The proposed methodology is tested with various 30-minute videos and 452 pre-identified cases of occlusion. Preliminary results indicate that the proposed methodology is reliability and robust in providing traffic density analysis. Future work may rely on the extension of the proposed methodology to deal with the detection of vehicles moving towards multiple directions.
INTRODUCTION

Vision-based traffic surveillance systems have been widely implemented as one of the most efficient and reliable alternatives to the traditional mean of surveying the traffic conditions (1). Precise detection of every vehicle, however, is not possible with current technology (1-2) using pre-existing algorithms given varying environmental conditions, such as illumination, and given traffic conditions that result in occlusion of vehicles. The vast majority of the existing research-oriented or commercialized traffic counting systems still suffer from over-counting by detecting multiple smaller parts of larger vehicles, or under-counting due to the implementation of non-robust detectors. In the special case of heavily congested traffic, incorrect detection can be severely exaggerated because most existing systems rely on motion-based detection; thus, many overlapped vehicles may be counted as one. Additionally, motion-based detection may cause alignment errors as an effect of the occluded object. In particular, when the detected object is occluded - regardless of the pre-occluded and the post-occluded state - the centroid at the center of the vehicle is shifted, altering and transposing the trajectory of the vehicle. Hence, the detected vehicle is disassociated from its movement trajectory.

The proposed full-body occlusion handling framework allows for a wider area of detection zones to be pre-set to match only the reliable data with the total number of counted vehicles. This way, the algorithms can detect a vehicle that would otherwise be missed in the traditional traffic counting methods, because a clear view of the fully occluded vehicle can be recovered before and immediately after the state of occlusion. The five-step methodology is implemented to automatically detect the vehicle element as truthful to the ground truth as possible.

Contrary to motion-based detection, one of the advantages of feature-based detection is its robustness against alignment error. More accurate counting per lane is facilitated with this method. The detected vehicle triggers the tracking algorithm with eigenimages and estimates vehicles’ location through particle filtering. Then, the tracking data are converted into real-world road coordinates enabled through calibration of the video field of view. In this particular framework, the detected vehicle may then be completely covered by an adjacent vehicle (e.g. heavy-duty truck), depending on the angle of the stationary camera regarding the road surface, leading to tracking failure and a required redetection of the vehicle.

In this paper, two mitigation strategies are employed to remove the previously detected tracking data and to solely consider the re-detected object so as to avoid over-counting. Also, a methodology to re-detect the occluded vehicle by the time it re-appears is presented for use with a spectrum of traffic state scenarios. Initially, various cases where full-vehicle occlusions occur in the sample video are examined to enable the extraction of workable samples. The extraction process is achieved by recognizing the irregular movement pattern of the labeled counts provided by the speed and acceleration data obtained through the object's trajectory in the calibrated field of view. Accordingly, the data corresponding to entirely occluded objects are stored automatically into databases and trained using feature-based detection to create learning classifiers iteratively applied to each occurrence. Finally, the occlusion handling from speed and acceleration data, as well as several tracking parameters’ tweaking are compared each other to assess which method provides the most reliable tracking occlusion handling.

The processing was conducted on surveillance video data provided by the Georgia Navigator System in the State of Georgia, USA, which covers most of the highway corridor in metropolitan Atlanta. The benchmarking is conducted between the performance results of the classifier from the automatic and manual data collection to assess the feasibility of the automated
image-collection method. The proposed methodology is tested using various thirty-minute-long videos recorded from different locations along the Georgia Highway Corridor. Because the credibility of the methodology relies greatly on the precision of the results, manual counts are employed as ground truth data for comparison with post-process analysis. The detection rate of feature-based detection as well as the removal rate of occluded vehicles is weighted individually. The results indicate the potential of the proposed occlusion-handling technique to provide reliable traffic flow results. Though it relies on either the instantaneous or average speed, the method is effective because of its versatility in adapting various environmental variables such as camera view-angle.

**LITERATURE REVIEW**

Motion-based detection considers a descriptive pattern of moving objects separated from the background model to locate objects of interest. In the application of vision-based traffic surveillance tools, many methods have been implemented, including adaptive median filters (3), color median (4-5), frame differentiation (6), mixture of Gaussian (MoG) (7-8), wavelet differentiation (9), kernel-based density estimation (10-11) and sigma-delta filters (12). Also, MoG is implemented with shadow elimination based on color reflectance and gradient feature (8). Recently, a sigma-delta filter was proposed (12) to continuously recognize a stopped vehicle as the foreground object and reduce the computational cost. Traditionally, background subtraction methods cannot cope with the major intrinsic problems of vision-based systems, such as occlusion, shadow, illumination, camera distortion and weather conditions (13); therefore, they are not considered reliable while standing alone. Also, contemporary background subtraction itself is ineffective detector of the vehicle objects at their center, which is critical for initiating accurate tracking records.

Many researchers have employed feature-based detection on top of computationally inexpensive background subtraction models to constrain the region of their interest (ROI). This strategy works well in dealing with shadow effects that cause over-counting, because the detected window fits to the vehicles’ appearance. In particular, various features of vehicle objects are targeted for appearance-based approach including - but not limited to - shape (14), points (15), edge (16) and symmetry (17) that are uniquely distinguishable. In approaches where machine learning is implemented to classify the object, detection capability relies on visual features that are selected for training. Furthermore, the extraction of point and edge-related information in low-quality images contains heavy noise and it is difficult to provide precision as correlation among neighboring pixels is not clearly defined.

More descriptive pattern-based object features such as Haar-wavelet features, HoG and Local Binary Patterns (LBP) are proven to be effective for vehicle detection applications. Haar-wavelet features classify each object by comparing the intensity of every adjacent pixel values and differentiating them depending on the presence of intensity gradients (18-19). Haar features are known to be robust for low-resolution images (18, 20) and yield fast processing times. Recently, Haar-Like features (HL) without gradient information was trained with SVM to attain satisfactory results (21). Color-based Haar detection was also demonstrated to be more effective than traditional gray-scale Haar feature detection (22). Another famous feature, the HoG, is widely used to consider gradient information and orientation around key point location of all pixels in vehicle image (15, 23). This methodology (24) was further investigated upon implementing unsupervised learning of images collected by background subtraction. Benchmarking of HL and HoG features for vehicle detection was conducted (23) and both
demonstrated to be effective. In a method described by Avery et al. (15), point-based features called Multi-Block LBP are considered to detect flat areas, corner points and edges in the trained images using AdaBoost along with integral image for faster computation. In addition to the benefit of appearance-based features described above, the precise acquisition of centroid of vehicle object is now possible because the detected window is appropriately fit to the vehicles’ appearance.

Many of the existing detection algorithms used in traffic surveillance applications recognize the importance of identifying the presence of occlusion, however, they do not address how the blocked view causes problems. Also numerous tracking algorithms have been implemented and improved; notably the Kalman filter (25), the particle filter (7, 26) and mean-shift (27). Within one important research initiative (25), the corner points of the detected vehicle are recognized by specific user-defined eigenvalue thresholds, and tracked using Kalman filter. Some known hybrid models as seen in a method described by Prevost et al. (23) use projective Kalman filter combined with mean shift tracker. Others simply linked independent detection in consecutive frames for tracking vehicles, assuming constant acceleration (24).

The majority of the existing tracking strategies involve tracking of local feature points instead of capturing the object in its entirety. This process is widely known as feature cluster tracking. In particular, Canny edge detection algorithms and Hough transformation were used to obtain the spatiotemporal information of the moving vehicles (16). Those clustering Hough feature points are grouped together to represent the vehicle's travel path, however, by assuming that the vehicle always travels linearly, which does not necessarily represent the real-world. Vehicles pass through curves and change lanes during periods of occlusion.

Similar concepts are implemented by considering clusters of point-based feature that are tracked using the Lucas-Kanade feature tracker provided as an open source (OpenCV) (28). It is critical to have a precise spatiotemporal travel path, because even a minor flickering movement can cause great variability in instantaneous speed and acceleration. Although not transportation related, some researchers have successfully demonstrated that the localized centroid update method can sufficiently alleviate small fluctuation in object movement (29). This method however requires additional post-processing work depending on the measurements between target model and detected candidate. More descriptive information regarding some of the tracking algorithms presented in this document can be found in the document created by Yilmaz et al. (30).

Recently, state-of-the-art commercialized products work similarly by localizing the detection zone to detect different types of vehicles (31). The structure of these systems is computationally inexpensive and can run on a real-time basis while retaining very respectable detection rates; however, they contain some inevitable problems caused by the object occlusion where larger vehicle with partially occluded smaller vehicle are typically considered as one object. This happens because foreground detection methods are not intrinsically designed to segregate multiple occluded vehicles (16). In another case, the appearance of larger vehicle or vehicle's shadow occluding the adjacent lanes also triggers false detection in commercial systems (28). Some detection-and-tracking systems are known to perform in real time as seen in a method proposed by Rodrigue et al. (32), however the traffic speed is assumed to be constant for each vehicle. Given the aforementioned, the merit of using computer vision as a surveillance tool had been limited; focusing strictly on building reliable systems that can perform in real-time.

Last but not least, most of the existing methods have been employed at either the ground level, observed from an overpass bridge, or on high-ground, where the detected vehicles’ size
remains large with less noise relative to the video frame size (6, 23, 26, 28, 33). However, most
CCTV cameras in surveillance networks are located very high above the ground, thus each
vehicle “occupies” a small area within the camera field of view. The reduced number of pixels
representing the vehicle in the field of view may not affect the detection capability of the system
when state-of-the-art background subtraction algorithms are applied that cope well with the
motion variables. However, degraded quality and smaller pixel-wise objects certainly can affect
the tracking accuracy of vehicle objects because single pixel fluctuation in tracking quality on 2-
D image coordinates results in greater difference in real-world vehicle trajectory.

Considering the CCVT cameras’ field of view on highways, severe full-body occlusion
has a significant impact on the tracking accuracy of the software. The proposed framework is
able to detect and track each vehicle independently, even if it is occluded. The only major
constraint is based on the requirement to have the vehicle-to-be-occluded at least partially shown
at the beginning. Moreover, if the scaling factor of each tracker diminishes significantly per
frame, it is more probable that the detected element is occluded. On the contrary, full-body
occlusion is sometimes imminent when the clear view of a vehicle is completely obscured by
another vehicle in the adjacent lane. In particular, the feasibility of capturing the ground truth
movement of the vehicle is examined using its trajectory estimation given the initial speed and
acceleration to estimate the position and speed of the vehicle when the state of occlusion is
passed. Finally, the averaged acceleration data are used on top of the scaling factor to distinguish
between false detections and true occlusions.

The proposed framework can benefit transportation researchers and practitioners by
proving more accurate results, with no capital cost, associated with the traffic density, evaluation
of drivers’ behavior in respect to the existing real-time condition (i.e., maintenance projects,
weather, incidents) and evaluation of the causes of crashes. The accurately extracted data can be
a stepping-stone for further assessment of the systems’ performance.

METHODOLOGY
This paper drives new technology information and achieves what has never even been attempted
before; a method for full-body occlusion handling and density analysis in traffic video-
surveillance systems where the object to be occluded is a moving vehicle. The proposed method
consists of five main stages: camera calibration, detection of the vehicle-to-be-occluded,
tracking of the vehicle-to-be-occluded, and the occlusion phase which includes the sub-phases of
vehicle counting and full-occlusion. Figure 1 illustrates the main steps and the appearance of the
output per step.
FIGURE 1: The main steps of our framework

Camera Calibration
Camera calibration provides the relationship between the 2-D image coordinates and the ground-truth coordinates. The CCTV cameras used in the Georgia Highway Transportation System provide a unique configuration where the field of observation is static; thus the calibration is robust for surveillance applications. During calibration, each video sample has been calibrated to its unique characteristics using the Vanishing Point, Length and Width calibration strategy (34). Considering that the width of each traffic lane is 12 ft and the transverse length between each end of the tick mark of dotted lane 40 ft, as they are both depicted in figure 1, the intrinsic camera properties including focal length \((f)\) or external variables such as tilt angle \((\phi)\) and pan angle \((\theta)\) can be computed.

The real world three-dimensional coordinates \((x_r, y_r, z_r)\) are converted into an image coordinate \((\mu_i, v_i)\) in following format:

\[
\begin{bmatrix}
\mu_i \\
v_i \\
1
\end{bmatrix} = \begin{bmatrix}
f & 0 & 0 \\
0 & -f\sin(\phi) & f\cos(\phi) \\
0 & \cos(\phi) & h\sin(\phi)
\end{bmatrix} \begin{bmatrix}
x_r \\
y_r \\
1
\end{bmatrix}, \quad \text{(Eq. 1)}
\]

assuming every point on the road is a planer object, which is not the case for vehicle objects. To incorporate this change, the transformation matrix is modified (eq.2) such to include the height of each vehicle:

\[
\begin{bmatrix}
\mu_i \\
v_i \\
1
\end{bmatrix} = \begin{bmatrix}
f & 0 & 0 \\
0 & -f\sin(\phi) & f(h-h_z)\cos(\phi) \\
0 & \cos(\phi) & (h-h_z)\sin(\phi)
\end{bmatrix} \begin{bmatrix}
x_i \\
y_i \\
1
\end{bmatrix}, \quad \text{(Eq. 2)},
\]

where the height is approximated as (eq. 3):
Moreover, the user specifies the total number of lanes and identifies an area in the video called entry zone (EZ) in the video frame where the detection phase will occur. The explicit selection of the EZ is critical for the steps that follow in our methodology. The details of the selection in the vehicle-counting chapter are described later in this paper.

Detection
The detection phase is subdivided into two steps; Background Subtraction and Haar-Feature detection. Our method implements two types of cues to characterize the appearance of the vehicles; motion and shape. These cues are exploited separately in a sequence of two steps. In the first step, the foreground blobs of moving objects are obtained. Then, the vehicles are separated from the foreground blobs based on their shape. The shape of the vehicles is defined by patterns of HoG features. Since the CCVT cameras are fixed and the field of view is static, the background can be estimated, therefore, the regions of the vehicles can be detected with lower processing time and fewer errors. To classify the areas of our interest, the AdaBoost classifier (35) is used.

To construct a classifier of vehicles’ shapes, which determines whether a sample of HoG feature represents vehicles or not, a large number of training images is required. Through the training, a classifier learns the patterns of HoG descriptors that can discriminate vehicles from other objects. The AdaBoost classifier is used to train Haar-Cascade by grouping the HL from the truly detected vehicles and segregating them from the false-detected images. During the process, each HL is applied separately to the same vehicle within the region of interest. The Haar-Cascade implementation in its default inserts a parameter called closer neighbor (37). The corresponding value of the closest neighbor specifies the minimum number of feature points that the overlapping detection needed for the object to be detected. Whenever the number of feature points decreases, more objects are detected because it reduces the number of overlapping regions to be detected. The output of detection is a number of rectangles surrounding each vehicle separately.

Tracking
The detection and matching of each image pixel in consequent frames to corresponding pixels allows the determination of the position of each vehicle for each time interval, or “timestamp”. To minimize computational effort and provide a real-time tracking algorithm, the matched and marked entities are used as input to the existing vision-based tracking algorithms in every frame. Given the automated input of these entities, the tracking methods can be used for real-time tracking without manual re-identification of the vehicle.

The appearance of the vehicle is modeled with principal components of its eigenimages, which is stored in temporal subspace. The principal components are repeatedly updated during tracking. The implementation of this tracking algorithm works smoothly with the Haar-Cascade detection because it establishes a good starting point for initial eigenimage capturing without any subsequent processing. This method is also fundamentally advantageous in detection-tracking systems as it is tracking a memorized object rather than the cluster of identical characteristics denoted by point or line features. In previous work (1), tracking precision was highest when kernel-based methods were implemented; however, occlusion and changes in illumination and
scale were considered independently. It is critical, though, to determine not only the number of
occluded vehicles but also their behavior throughout the phase of occlusion.

**Vehicle Counting and False Detection**

The first step is to specify the Entry Region (ER); an area where the detection and
tracking begin. The selection of the ER is the most important step because the precision of each
further step is dependent upon this selection. Thus, the ER needs to include all the areas where
detection and tracking works the best and simultaneously exclude all the stationary objects that
can hinder the tracking capability.

For every detected vehicle, the tracking algorithm seeks to capture the \( x \) and \( y \)
coordinates of the center of the box that surrounds the vehicle as long as it remains in the ER.
Simultaneously, the \( x-y \) pairs “draw” the trace of its trajectory; thus, information about the speed
and location of the vehicle can be obtained. The field of view calibration then promotes the
conversion of the 2-D image planes to 2-D road coordinates. Whenever the vehicle departs from
the ER, a counter placed over the specific lane increments the count by one. This process is
repeated for all the lanes.

One of the predominant constraints of counting in the ER is that various interfering
traffic-related and visual conditions may occur in the ER. These include partial or full-body
occlusion, lane changes, closely following cars and visibility issues. The first and most important
is the recognition of false detected vehicles.

For the purpose of false detection, the relationship between average and individual speed
versus density is examined. In general, traffic density is inversely proportional to traffic speed
(Eq. 36); meaning that if the flow speed is high, then density is generally low, and vice versa. If the
relationship is not met, the detected object is more likely to be a non-vehicle. However, the
variability of speed versus density is not linear and cannot be expressed as a function, therefore a
generalized relationship cannot serve as a precise reference. Also, each lane behaves differently
(forthcoming research results). The authors decided to store the local behavior per lane at
specific timestamps to deal with this issue. Equation 1 describes the average local speed at
timestamp \( t \) as the following:

\[
v_{\text{min},t,l} = v_{r,t} = \frac{\sum_{k=t-1}^{T} v_{k,l} \Delta t_{k,l} - v_{th}}{N} \quad \text{(Eq. 4)}
\]

Additionally, the scale factor serves as a criterion of false detection. One contour box
circumscribes each vehicle and thus is assumed to decrease at a specific rate as it moves away
from the camera. If the decreasing rate fluctuates, then it is likely that the box slipped, indicating
that the detected object is a non-vehicle. The formula in this case is presented in Equation 2.

\[
S_{t} > \frac{s_{\text{avg}}}{4} \quad \text{(Eq. 5)}
\]

Finally, both the instantaneous and averaged speed data are affected by the varying frame rate of
the video. For this reason, there is a slight probability that the vehicle can be falsely identified as
a non-vehicle. To resolve this problem, the tracking of each vehicle contains a method where for
each time the speed falls within the filter range, as specified by Equation 1 or 2, its value is
incremented by 1. If this value exceeds or equals the pre-determined threshold (in this
framework, a constant threshold of 2 is used), then such tracking can finally be considered as
non-vehicle.
Occlusion Handling

Occlusion handling is more complicated than the identification of false detection because:

1. The high variance in speed fluctuation at the instances when the clear view of a vehicle is hindered; as an example, in queue dissipation conditions,
2. Different types of occlusion occur, including occlusion of one car by another in free flow conditions, multiple occlusion from one car under stop-and-go conditions and more, and
3. The relationship between the acceleration and the occluding vehicle just before, during and after the occlusion.

Therefore, pattern recognition regarding different types of occlusion scenarios as well as their relation to the general traffic movement can be hard to generalize. The clear view of a single vehicle moving at constant speed can be hindered by traffic and vehicle types moving various speeds in the neighboring lane, anywhere from 0 mph to design speed (e.g. 75mph on highway) depending on congestion level.

The tracking algorithm presented in (5) offers a unique solution to the phenomenon of a vehicle image “disappearing” from the scene when such vehicle is completely occluded by an adjacent larger vehicle (such as a large truck). The precision, though, of this algorithm depends on the presence of clear trained images from the vehicle that is initiated in the detection phase. The algorithm continuously recaptures and updates the view of this vehicle element throughout the vehicles’ movement and stores the spatial data into the temporary subspace, while managing to retain accuracy even as the vehicle image is rotated or skewed. However, when the vehicle's clear view is disrupted by the occlusion, tracking that is based on the past-stored subspace becomes unreliable over time as the change in the view departs from the stored spatial view. This essentially causes the tracking of the same vehicle to become unstable as the tracked element "wobbles" around the region where the object was successfully tracked.

The proposed method can recognize this wobble movement as reflected by the abrupt change in the scaling of the tracked area where the algorithm is trying to maintain the subspace data. The following criteria will first detect the occluded candidates.

\[
\frac{S_t}{S_{t-1}} < S_{th} \quad \text{(Eq. 6)}
\]

\[
S_t > \frac{S_{avg}}{4} \quad \text{(Eq. 7)}
\]

Similar to the identification of false detection, the occlusion-handling framework recognizes each possible occlusion, and each successful tracking event is incremented by 1. If the value exceeds the preset threshold, the system finally recognizes the vehicle as being occluded. Empirical study indicates that the method segregate the occluded data from the rest fairly well, but also contain extraneous data from the occluding vehicle or object. The extraneous data that are misclassified as the occluded vehicle typically reduces in scale value have extremely high, nonrealistic fluctuation in average speed. The true occluded vehicles typically seem to reduce in scaling factor gradually and steadily while retaining their average speed fairly constant. Thus, additional processing was added to sort out true occlusion by reinforcing the algorithm with acceleration data.
1) \( \frac{a_{\text{avg},i}}{a_{\text{th}}} < 1 \) \hspace{1cm} (Eq. 8)

The acceleration data are highly unreliable without additional smoothing due to varying frame rates, however, they serve effectively in comparing the result produced from the first occlusion handling based on the scale factor. This low-pass filter compares the current vehicle acceleration to the threshold. If the filter exceeds the threshold more than twice, the label is officially registered as an occluded vehicle.

There are four possible occlusion scenarios that can be observed in highway traffic and numerous subcases for each of them. The main scenarios include:

1. The vehicle is initially visible, then becomes occluded (V-O),
2. The vehicles’ view is clear, then becomes occluded and then reappears once the occlusion is finished (V-O-V),
3. The vehicle is initially occluded and then appears somewhere in the ER (O-V),
4. The vehicle is initially occluded, becomes visible, and is occluded again (O-V-O), which may not be detected if the vehicle first becomes visible outside of the EZ, and
5. The vehicle is occluded for its’ entire movement (O), which cannot be detected, tracked or counted.

For scenario 3, occlusion-handling techniques are unnecessary; as the vehicle count algorithm can added to the counter exactly by the time the vehicle enters the counting zone. On the contrary, in scenarios 1 and 2, the output may cause undercounting or over-counting. As an example, if the detected vehicle is to be tracked with the same configuration as before, it often results in undercounting because of the wobble movement that increasingly affects the tracking accuracy. In this scenario a rapid increase in scaling is observed.

By applying conservation of vehicles, the total number of cars that enter the system equals the total number of cars that exists the system, even under the conditions of occlusion. The following paragraphs explain how the neighboring traffic environment, detection training, and lane changes affect the outcome of occlusion for the cases 1 (V-O) and 2 (V-O-V).

In scenario 1, the algorithm retains the vehicles’ motion trace for use in subsequent frames given the trained pre-occluded images that memorize its appearance. It is also assumed that the vehicle moves parallel-to-traffic, its speed remains constant and that the perpendicular-to-the-traffic speed vector remains within specific range during the occlusion, which is against what it was assumed in the previous literature. The range of the perpendicular to the traffic speed varies depending on the actual speed of the car and is calculated based on the 95% precision that needs to be achieved, plus the consideration that the level of resulting alteration in the x-y speed does not exceed 2mph. Given the aforementioned, the formula takes the following form (equation 6).

\[
\sum V_F \text{oc. car} = \sum V_l \pm 2, \text{ and } V_F - 1.96 \frac{2}{\sqrt{n}} \leq \sum V_F \text{oc. car} \leq V_F + 1.96 \frac{2}{\sqrt{n}}, \text{where } V_F \text{ is the avg speed.} \text{ (Eq.9)}
\]

The average speed, \( V_F \), is based on the cumulative averaged speed of the cars that surround the occluded vehicle based on the minimum neighbor value (MNV) discussed earlier. The lower the value of minimum-neighbor, the better the precision will be; because it is directly related with the occlusion and the possibility of lane changing.
RESULTS AND VALIDATION

The proposed method was implemented using a prototype created by Microsoft Visual C#. This prototype uses OpenCV as its main image processing library and EmguCV as a .Net wrapper to OpenCV. The team recorded 350 vehicle pre-occluded images (640 by 480 pixels) from forty-five videos recorded from the CCVT cameras of the Georgia Department of Transportation on I-85 to “train” the algorithm. To be able to check the accuracy of the method in different situations, pre-occluded images are obtained under different weather and lighting conditions as well as under various neighboring and traffic conditions.

Using these images, the uniqueness threshold was varied to find the value that detects and reliably matches the HoG of the vehicle to be occluded in eight or more points. The threshold value of 0.0625 was the smallest distance value that achieved the required number of matching points. Therefore, the threshold was set to 0.0625. Under this threshold, the average error is 1 pixel. After calibrating the uniqueness threshold, the accuracy of the method was tested according to changes in the aforementioned various conditions, however, since the objects have been captured using the CCVT auto-focus cameras, the method was not tested for poorly focused images. The camera set was calibrated using MAPSAC algorithm (32). After camera calibration, information regarding detection, tracking, vehicle counting and occlusion handling was extracted. The output data include direct and indirect information regarding:

1. The number of vehicles per lane,
2. The number of cars per lane,
3. The number of lane changes by lane of origin and the exact location where this occurs
4. Speed Data for all the vehicles, and
5. The number of occluded vehicles and location of their occlusion.

To verify the robustness of the proposed methodology, the output counting data were compared to the manual counting data of the same video for verification. The performance of system for each video is evaluated by the following testing parameters:

1) Correct Counting Rate (CCR) = $\frac{\text{Counting data that are vehicle}}{\text{Manual counting data}}$

2) False Counting Rate (FCR) = $\frac{\text{Counting data that are non-vehicles}}{\text{Total counting data}}$

3) Removal Rate RR (RR) = $\frac{\text{Total number of falsely detected objects removed from counting}}{\text{Total number of falsely detected objects}}$

4) Occlusion Handling Rate (OHR) = $\frac{\text{True positive occlusion}}{\text{Total detected occlusion}}$

The aggregated results of the comparison from the forty-five 30-minute long videos as well as the independent variables and measures of effectiveness are presented in Table 1 for MNV=2 and inside parenthesis for MNV=5. The results from each video sample are also compared in terms of observed environmental variables such as illumination variance, congestion level and the difference in camera angles ($\theta$, $\phi$, horizontal and vertical angles respectively). The most significant results are illustrated in the total number of automated counts and the correct versus false occlusion counts, which validates the performance of the algorithm.
**TABLE 1: Counting Data for Minimum Neighbor = 2 (In parenthesis MN =5)**

<table>
<thead>
<tr>
<th>True (manual) Counts</th>
<th>380</th>
<th>562</th>
<th>421</th>
<th>512</th>
<th>301</th>
<th>2176</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane Number</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>Total</td>
</tr>
<tr>
<td>Automated Counts</td>
<td>377 (352)</td>
<td>554 (511)</td>
<td>413 (408)</td>
<td>500 (469)</td>
<td>299 (289)</td>
<td>2142 (2028)</td>
</tr>
</tbody>
</table>

**Over-counts**

<table>
<thead>
<tr>
<th></th>
<th>Double Count</th>
<th>False Count</th>
<th>Falsely added to the counter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 (0)</td>
<td>4 (4)</td>
<td>2 (2)</td>
</tr>
<tr>
<td></td>
<td>0 (0)</td>
<td>2 (2)</td>
<td>4 (4)</td>
</tr>
<tr>
<td></td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (3)</td>
</tr>
<tr>
<td></td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>

**Under-counts**

| False Occlusion (Correct Counts) | 0 (0) | 2 (3) | 1 (3) | 0 (1) | 0 (0) | 3 (7) |
| Correct Occlusion Counts        | 14 (14) | 21 (20) | 8 (8) | 1 (0) | 0 (0) | 44 (42) |

Total Correct Vehicle

| 377 (352) | 548 (505) | 412 (402) | 500 (464) | 297 (286) | 2134 (2009) |

Total False Count

| 0 (0) | 6 (6) | 6 (6) | 1 (5) | 1 (3) | 14 (20) |

Total Missed Vehicle

| 3 (26) | 14 (57) | 9 (19) | 12 (48) | 4 (15) | 42 (165) |

Total Removed

| 8 (10) | 25 (31) | 44 (41) | 46 (39) | 10 (12) | 133 (133) |

Total Correctly Added by OH

| 14 (14) | 23 (23) | 9 (11) | 1 (1) | 0 (0) | 44 (46) |

CCR (%)

| 98.95 (92.63) | 97.69 (89.86) | 97.86 (95.49) | 97.66 (90.63) | 98.67 (95.02) | 98.29 (92.73) |

FCR (%)

| 0.0 (0.0) | 1.1 (1.2) | 1.46 (1.49) | 0.2 (1.08) | 0.3 (1.05) | 0.612 (0.964) |

RR(%)

| 0.0 (0.0) | 96.2 (91.4) | 99.6 (96.2) | 99.7 (93.4) | 99.6 (93.3) | 98.9 (93.6) |

OHR(%)

| 100 (100) | 100 (100) | 97.1 (92.3) | 100 (100) | NA (NA) | 99.42 (98.46) |

**LIMITATIONS**

There are three main limitations to the occlusion handling methods developed in this work: 1) speed variability under various congestion levels causes errors, 2) the spillover effect, and 3) the propagation of errors within vehicle platoons.

Regarding speed variability, the initial assumption is that the drivers’ characteristics are approaching the normal distribution, however, this assumption does not adequately account for
timid and aggressive drivers. Also, whenever the sight distance is greater (i.e. in free flow), the error is greater because the potential acceleration or deceleration of the vehicle cannot be estimated precisely while it is occluded. On the other hand, the average difference in speed in the aforementioned case shouldn’t exceed two miles per hour, as the speed defined by the geometric position of the occluded car with respect to the camera given that the assumed average length of a car and truck is 18ft and 40ft respectively. Finally the same issue is observed when series of vehicles were not successfully detected in the system: the stored reference speed fails to cope with the abrupt change in the speed differentials.

Most issues are defined when the detection fails to represent the vehicles’ entire view; thus the occlusion does not match any of the major categories described above. In this case, vehicles’ counts are often misplaced into the adjacent lanes. This is defined as the spillover effect and results from:

1. Partial detection of the moving object: In this case, the centroid – which corresponds to the x-y pair of vehicles’ location at every timestamp – is shifted from the center of the lane and appears on a different case. In that other case lane changes cannot be observed.

2. If counting zones are placed in the same transition zone, false vehicle counts are added on the adjacent lane; therefore the calculation of density is not precise.

Finally, when vehicle platoon is observed, the probability of error in occlusion increases because of the speed fluctuation (stop-and-go). In particular, a potential error in the detection of occlusion in one vehicle expands to the vehicles that follow behind on the same lane causing a series of errors.

CONCLUSION AND RECOMMENDATIONS
The purpose of this paper is to delve into new fields for research by introducing a reliable framework for full-body occlusion handling on vehicles using monocular CCVT cameras mounted on the sides of the Georgia Interstate System. Also, a novel and completely automated version for vehicle detection is introduced as part of the process. The benefits of the proposed framework are summarized below:

1. When the trained classifier is relatively weak, an increase in the area that the ER covers leads to greater detection rates

2. Recognition of false positives is effortless, thus the vehicle counting output is precise, and the performance of a classifier based on the region of interest curve can be loosened to allow for higher detection rates,

3. Occlusion detection can also be tested and can be utilized in order to add missing vehicle to the overall counting.

Recommendations for future work include, but are not limited, to the definition of the sub-components of the algorithm to deal with more complicated movements. For example, the occlusion of one vehicle by another in a transit hub or a crossing area may require adaptation.

With respect to image processing, further assessment of the potential use of the proposed methods in cheaper and broadband cameras with lower resolution or poor video quality must be examined. The reliability of proposed framework may be further improved by reinforcing the filtering operation with more advanced microscopic traffic theory in the aforementioned challenging environments. These propositions will serve as a significant component in the dissemination of this research.
ACKNOWLEDGEMENTS

This material is based upon work supported by the Georgia Department of Transportation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Georgia Department of Transportation.

REFERENCES


