Slot Cars: 3D Modelling for Improved Visual Traffic Analytics

Eduardo R. Corral-Soto
Centre for Vision Research
York University
ecorral@cse.yorku.ca

James H. Elder
Centre for Vision Research
York University
jelder@yorku.ca

Abstract

A major challenge in visual highway traffic analytics is to disaggregate individual vehicles from clusters formed in dense traffic conditions. Here we introduce a data driven 3D generative reasoning method to tackle this segmentation problem. The method is comprised of offline (learning) and online (inference) stages. In the offline stage, we fit a mixture model for the prior distribution of vehicle dimensions to labelled data. Given camera intrinsic parameters and height, we use a parallelism method to estimate highway lane structure and camera tilt to project 3D models to the image. In the online stage, foreground vehicle cluster segments are extracted using motion and background subtraction. For each segment, we use a data-driven MCMC method to estimate the vehicles configuration and dimensions that provide the most likely account of the observed foreground pixels. We evaluate the method on two highway datasets and demonstrate a substantial improvement on the state of the art.

1. Introduction and Prior Work

In many traffic surveillance installations, camera placement is oblique. As a consequence, vehicles project to the image in clusters and are often only partially visible due to occlusion. Disaggregating these clusters into individual vehicles is central to attaining accurate vehicle counts. Tracking and appearance cues are highly fallible, as vehicles may move at similar speeds and have similar colours. Size cues are also tricky, as vehicle size can vary by an order of magnitude, from motorcycles to tractor-trailers.

Attempts have been made to solve this problem using 2D spatiotemporal constraints. One idea is to use variations in velocity within a segment or variations in the shape of an image segment over time to identify the multiple vehicles within a cluster [2, 1, 9]. Unfortunately, since highway speeds are highly regulated, velocity is not a very effective segmentation cue for highway traffic: it is very common for neighbouring vehicles to be travelling at the same speed and in the same direction. An image segment projecting from a single vehicle may also vary in shape over time due to shadows, specularities and projective distortations, making this a weak cue as well. Bouvié *et al*. [2, 1] attempt to strengthen this method with the constraint that local image features projecting from each vehicle form a convex group in the image. However, this approach is also limited, since an individual vehicle could project a non-convex group, while a cluster of vehicles could project a convex group.

To overcome these limitations we propose a 3D approach. 3D model-based methods for object verification (motorcycles, horses) [14] and for make/model vehicle recognition [8, 10] have recently proven effective on datasets where the objects are fairly isolated and already localized in the image. However, these papers do not address the thorny problem of detecting and individuating multiple mutually occluding objects in highly cluttered scenes, which is the problem we must solve to achieve accurate traffic analytics in rush-hour conditions.

To address this problem, we take as inspiration the earlier work of Song & Nevatia [13]. Their insight was that 3D models of common vehicles combined with knowledge of intrinsic and extrinsic camera parameters could be used to reason about the 3D configuration of vehicles most likely to account for observed clusters in the image. This is a powerful approach, and has the advantage that vehicle categorization and traffic volume measurement can potentially be solved simultaneously.

There are two main limitations of this 3D model approach that we address in this paper. The first is how the 3D models are defined. The Song & Nevatia algorithm employed three categories of vehicle (sedan, SUV, truck), and assumed they occur with equal probability. This is clearly unrealistic for highway traffic, where diverse vehicle categories are possible. For example, in our datasets we have enumerated 13 different semantic vehicle categories, and find that the prior distribution is far from uniform (Fig. 1). To address this challenge, we propose an automatic clustering approach, optimizing the number of clusters to maximize the accuracy of traffic volume measurements.

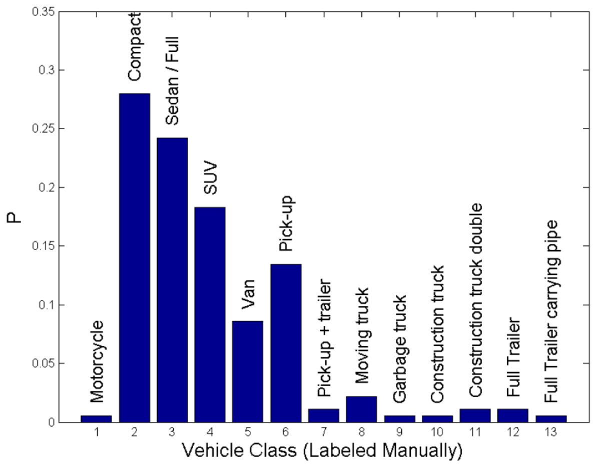


Figure 1: Distribution of vehicle classes for our training dataset.

The second limitation is the complexity of the configuration space. In the Song & Nevatia approach, the unknowns included the number, category, location and orientation of the vehicles in the scene. The combinatorial complexity of this configuration space led them to propose a rather complex coarse-to-fine search, terminating in a fine-grained MCMC stage. To address this problem, we take advantage of recent methods for automatically recovering the lane structure of the highway [4]. This allows us to fix the pose of each 3D vehicle proposal, and to limit the search over location to a 1D space. As a result, a single MCMC search stage is sufficient to recover optimal configurations. We refer to our method as 'Slot Cars' because the MCMC algorithm slides the 3D models one-dimensionally along each lane to identify the most probable configuration.

While Song & Nevatia employed complex 3D CAD models for their vehicles, we elect to employ simpler cuboid models (Fig. 2) that have been used effectively in recent work on camera calibration [5], vehicle detection [3] and fined-grained vehicle recognition [12]. On the other hand, while Song & Nevatia relied on an orthographic projection approximation, we assume full perspective projection, as it adds negligable complexity and should provide more accurate results.

2. Datasets & Geometry

We recorded two highway traffic datasets at different highways and on different days. Both datasets were recorded with a Sony Nexus 6 camera at 1440×1080 pixel resolution and 30fps. For Dataset 1 we labeled 1,072 frames ($\sim$ 36 sec) and for Dataset 2 we labeled 494 frames ($\sim$ 16 sec). We employed the first 566 labelled frames of Dataset 1 as training data, and used the last 506 labelled frames for evaluation. Dataset 2 was used solely for evaluation, serving to assess the ability of the algorithm to generalize to different conditions.

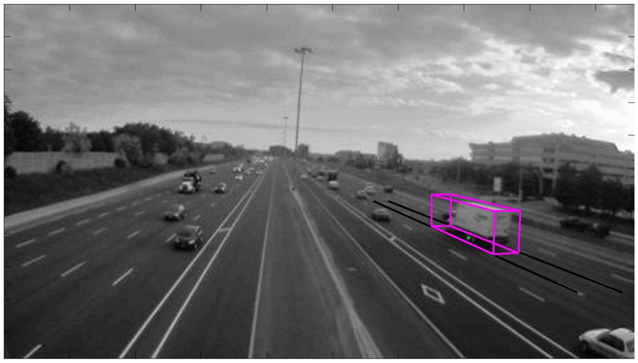


Figure 2: Example cuboid model

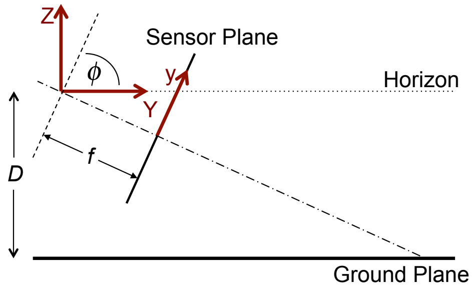


Figure 3: Camera geometry. Both the X-axis of the world frame and the x-axis of the image frame point out of the page.

The camera was calibrated in the lab using standard procedures [15]: The focal length was estimated to be $f = 1,142$ pixels and the principal point (p_x, p_y) was found to be centred vertically and displaced by only 3.5 pixels to the right horizontally. Skew was assumed to be zero, and pixel aspect ratio was assumed to be unity. For both datasets the camera was mounted on a tripod on an overpass overlooking a highway; Fig. 3 shows the geometry in profile. We measured the camera height above the ground plane to be approximately $D = 8.01$ meters for Dataset 1, and $D = 8.36$ meters for Dataset 2. The roll angle of the camera was minimized using the camera's internal electronic levelling gauge and we assume it to be zero in the following. We downsampled the video to 360×270 pixel resolution prior to processing to reduce computation time. Both datasets were hand-labelled to identify a bounding box and semantic category for each vehicle in each frame. A unique ID was assigned to each unique vehicle, tracked across frames.

We assume a planar horizontal ground surface and adopt a right-hand world coordinate system $[X, Y, Z]$ centred at the camera, where the Z-axis is in the upward normal direction (Fig. 3). Without loss of generality, we align the

x-axis of the image coordinate system with the X axis of the world coordinate system (both out of the page in Fig. 3). For notational simplicity we locate the centre of the image coordinate system at the principal point.

Under these conditions, a point $[X, Y]^T$ on the ground plane projects to a point $[x, y]^T$ on the image plane according to

$$\lambda[x, y, 1]^T = H[X, Y, 1]^T, \quad (1)$$

where λ is a scaling factor and the homography H is given by ([7], Page 328, Eqn. 15.16):

$$H = \begin{bmatrix} f & 0 & 0 \\ 0 & f\cos\phi & -fD\sin\phi \\ 0 & \sin\phi & D\cos\phi \end{bmatrix} \quad (2)$$

where ϕ is the tilt angle of the camera relative to the ground plane: $\phi = 0$ when the camera points straight down at the ground surface and increases to $\pi/2$ as the camera tilts up toward the horizon.

Conversely, points in the image can be back-projected to the ground plane using the inverse of this homography, $[X, Y, 1]^T = \lambda H^{-1}[x, y, 1]^T$, where

$$H^{-1} = (fD\cos 2\phi)^{-1} \begin{bmatrix} D & 0 & 0 \\ 0 & D\cos\phi & fD\sin\phi \\ 0 & -\sin\phi & f\cos\phi \end{bmatrix} \quad (3)$$

In Euclidean coordinates this backprojection can be written as:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{D}{f\cos\phi - y\sin\phi} \begin{bmatrix} x \\ y\cos\phi + f\sin\phi \end{bmatrix} \quad (4)$$

This inverse homography will be used to back-project the lane boundaries detected and grouped in the image back to the ground plane.

Our method will also involve projection of 3D cuboid vehicle models resting on the ground plane to the image for comparison with detected foreground segments, and for this we employ the 3×4 homogeneous camera projection matrix P: $\mathbf{x} = P\mathbf{X}$, where $P = KR[I \mid \mathbf{0}]$, and K and R are the 3×3 instrinsic parameter and rotation matrices respectively. Given our assumptions, the intrinsic matrix K reduces to:

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

and the rotation matrix R reduces to:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & \sin(\phi) & -\cos(\phi) \end{bmatrix} \quad (6)$$

We measured ground truth values for the tilt angle ϕ using an SPI digital protractor: $\phi = 81.2$ deg For Dataset 1, $\phi = 73.4$ deg For Dataset 2. These ground truth values will be used to validate the camera tilt estimates made automatically from the imagery (see below).

3. Algorithm

Our method consists of a 3D modelling stage and an inference stage (Fig. 4).

3.1. 3D Modelling Stage

In the 3D Modelling Stage, the geometry of the ground plane lane structure of the highway is first established. This is then used to learn the image appearance of 3D cuboid vehicle models populating these lanes.

3.1.1 3D Roadway Geometry Estimation

Our goal is to facilitate traffic analytics by automatically estimating the projective relationship between the highway ground plane and the image, and to automatically recover the lane structure of the roadway. To achieve this, we use a ‘preview video’ consisting of the 200 frames immediately preceding each labelled dataset to burn in an online mixture model background subtraction algorithm [6] in order to estimate a reliable background image that clearly shows the lane structure, without occlusions from vehicles (Fig. 5(a)). We then employ the parallelism method of Corral-Soto & Elder [4] for automatic single-view calibration and rectification. This algorithm first detects and groups local oriented structure into longer curve segments (Fig. 5(b)). (Although the highway shown here is straight, the method can handle curved highways.) These segments are then used together with knowledge of the camera height and intrinsic parameters to automatically estimate the camera tilt angle ϕ that maximizes the parallelism of the curve segments when they are back-projecrted to ground plane coordinates using the inverse homograph H^{-1}. (Fig. 5(c)).

Note that the extracted curve segments include the lane boundaries but also other parallel curves generated by the meridian, parallel lane markings for the HOV lanes, shoulder etc. In order to distinguish the traffic lanes from these other structures, we first identify as candidate lanes the curvilinear strips between all adjacent pairs of parallel curves. We then use the foreground segments detected in the preview video to identify which of these candidate lanes is active. Due to the oblique pose of the camera, the lowest point in each of these segments tends to lie close to the ground plane. We therefore identify the ground plane location of each segment by the back-projection of its lowest point in the image and then accumulate these points over time for each of the candidate lanes (Fig. 5(d)). Since some of these foreground segments will actually correspond to multiple vehicles spanning multiple lanes, the frequency distribution of these points (Fig. 5(e)) cannot be taken as an accurate estimate of traffic volume, but it is sufficient to identify the active lanes. In our system, we treat as active any lane containing more than 10 points (Fig. 5(f)).

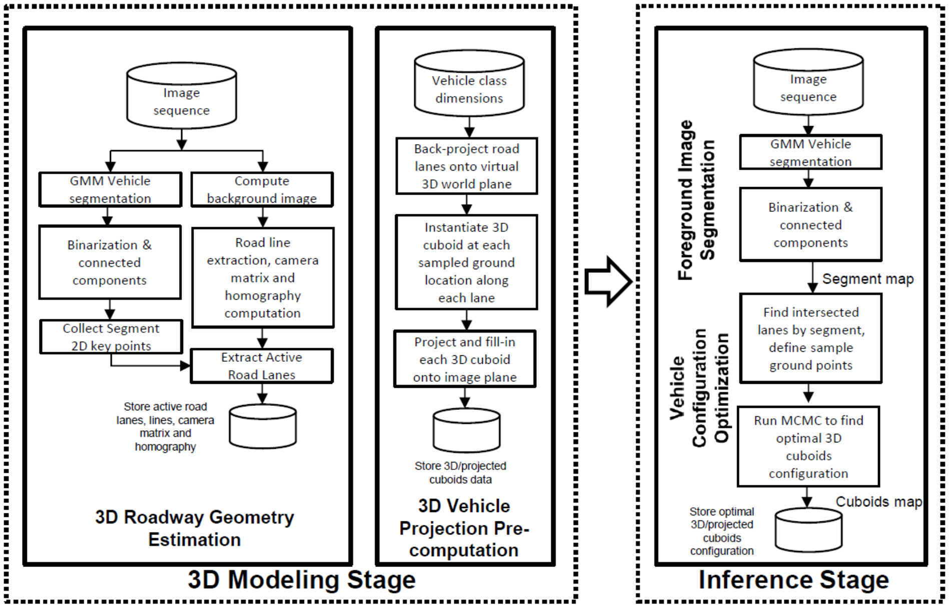


Image sequence
GMM Vehicle segmentation
Compute background image
Binarization & connected components
Road line extraction, camera matrix and homography computation
Collect Segment 2D key points
Extract Active Road Lanes
Store active road lanes, lines, camera matrix and homography
3D Roadway Geometry Estimation
Vehicle class dimensions
Back-project road lanes onto virtual 3D world plane
Instantiate 3D cuboid at each sampled ground location along each lane
Project and fill-in each 3D cuboid onto image plane
Store 3D/projected cuboids data
3D Vehicle Projection Pre-computation
3D Modeling Stage
Foreground Image Segmentation
Image sequence
GMM Vehicle segmentation
Binarization & connected components
Segment map
Vehicle Configuration Optimization
Find intersected lanes by segment, define sample ground points
Run MCMC to find optimal 3D cuboids configuration
Cuboids map
Store optimal 3D/projected cuboids configuration
Inference Stage

Figure 4: Algorithm Overview

3.1.2 3D Vehicle Projection Pre-Computation

The estimation of 3D roadway geometry gives us the potential to transfer observations and hypotheses between the 3D scene and the 2D image . We will use this to implement an analysis-by-synthesis approach in which 3D hypotheses of vehicle configurations on the roadway are evaluated in terms of how well their image projections align with detected image foreground segments. The first step is to determine the exact 3D vehicle models to employ.

3D Vehicles Classes
In their 3D modelling approach, Song & Nevatia employed CAD models for three vehicle categories (sedan, SUV, truck). For highways, the distribution of vehicles types is more diverse (Fig. 1) and there is considerable variation in dimensions and shape within each class. For these reasons, we elected to use simpler 3D cuboid models and to learn optimal dimensions for these models from training data. Specifically, we first manually identified the subset of segments in the training dataset that involve only one vehicle, and determined the lane for each by the lowest point in the segment. We then defined a ground plane origin for the modelling of the segment as the back-projection of the projection of the centroid of the image segment onto the midline of the lane. Next, we instantiated a set of 3D cuboid models uniformly sampling a range of plausible dimensions and locations, centred at the origin (Table 1). Finally, we identified the the set of parameters that maximized the intersection-over-union (IOU) of the image projection of the cuboid with the observed image segment.

Table 1: 3D cuboid model sampling on training data. Locations are relative to ground plane origin (see text).

Parameter	Min (m)	Max (m)	Resolution (m)
Location	-3.0	3.0	1.0
Length	2.2	23.0	0.2
Width	0.8	2.6	0.2
Height	1.1	5.0	0.2

Fig. 6 shows the resulting distribution of cuboid dimensions over the training dataset. Note the broadness of the distribution, particularly in the length dimension. To partition the distribution into more compact categories, we ran the k-means algorithm for $k = [1 \ldots 10]$, repeating $n = 1000$ times with random initial conditions for each value of k and selecting the solution for each value of k that minimizes the average intra-cluster variance. The number k of vehicle categories was selected to optimize the accuracy of traffic volume estimates on our training dataset: we found that $k = 4$ yields optimal performance (Fig. 6). We discuss this optimization in more detail in Section 4.

Table 2 shows the vehicle dimensions for the four cluster centres. Although there is not a 1:1 mapping between these clusters and semantic vehicle categories, Class 1 corresponds roughly to a compact car, Class 2 to an SUV, passenger van or pickup truck, Class 3 to a cube van and Class 4 to a semi-trailer.

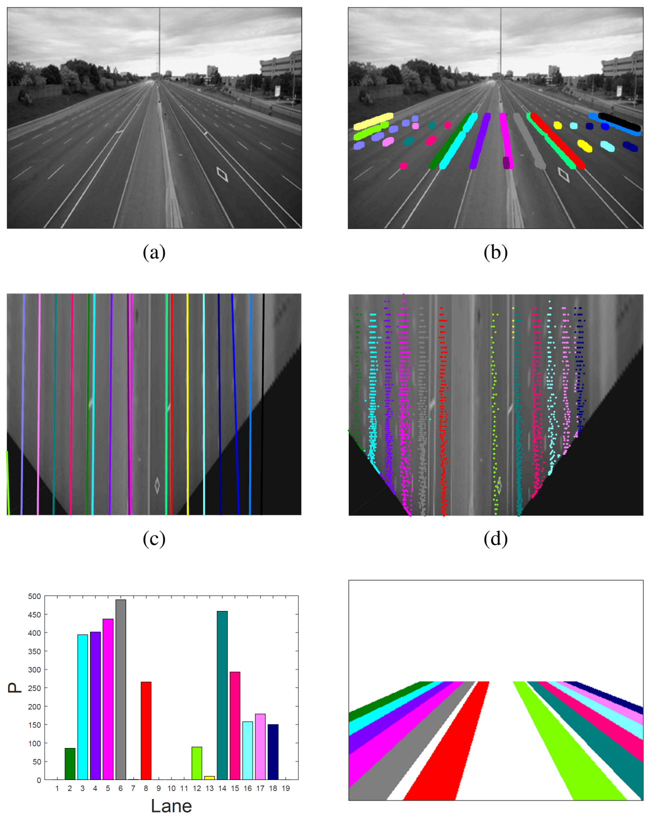


Figure 5: 3D roadway geometry estimation for Dataset 1. (a) Background image recovered from initial 200 frames. (b) Line segments detected and grouped in the background image. (c) Rectified background image with initial lane boundary estimates overlaid. (d) Rectified background image with vehicle locations used to identify active lanes. (e) Distribution of traffic over lanes. (f) Labelling of active lanes in the image.

Table 2: Dimensions for the four vehicle classes automatically learned from our training data.

Class	Length (m)	Width (m)	Height (m)
1	4.2	1.7	1.5
2	5.7	1.8	1.7
3	9.2	2.6	4.6
4	23.0	2.6	4.0

Projection of 3D Models to the Image

Having estimated the homography H relating the ground plane to the image, the lane structure in ground plane coordinates, and the 3D cuboid model classes, we can now pre-compute an estimate of the expected image appearance (silhouette) of each vehicle class for each traffic lane and for each location along each lane. For the sake of efficiency we assume that each vehicle will be centred within a lane. We sample lane locations at 1m resolution.

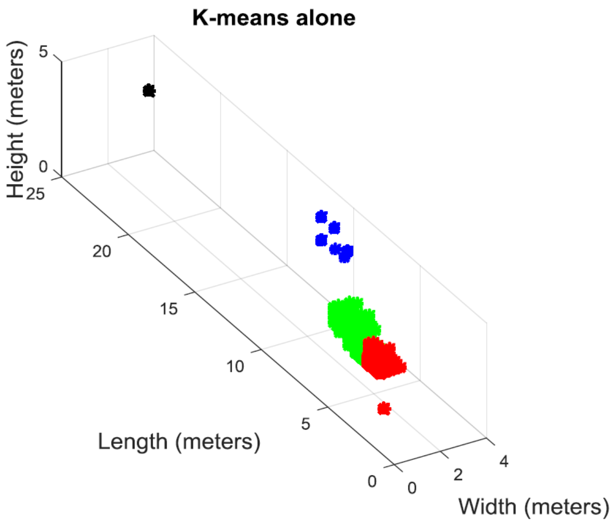


Figure 6: K-means clustering of vehicles in training dataset.

3.2. Online Inference Stage

3.2.1 Foreground Image Segmentation

We restrict our analysis to vehicles lying within or at least intersecting region of interest in the lower portion of the video frame, to avoid very distant vehicles near the horizon (Fig. 7(a)).

We employ a foreground segmentation method similar to the first stage of the object detection method of Elder *et al*. [6]. While the original method, intended for face detection, employs a probabilistic combination of background subtraction, motion and colour cues, colour cues have little discriminative value in our application due to the broad diversity in the colour of vehicles. We therefore omit the colour cues in our system.

The goal is to independently label each pixel in the image as foreground or background. The background subtraction component of the algorithm is based on a 2D adaptive Gaussian mixture model for pixel colour that ignores the luma channel to minimize responses to shadows. The motion component is a simple two-frame difference. The probabilistic combination of these two cues imbues the algorithm with a degree of invariance to traffic speed, since the background subtraction works well for slower speeds and the motion detection works well for higher speeds. Note, however, that for stalled traffic the background subtraction algorithm will eventually begin to interpret the stopped vehicles as background.

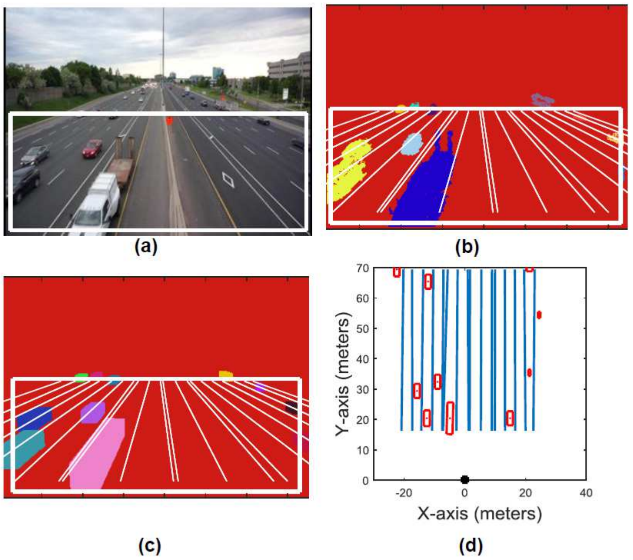


Figure 7: Online inference. (a) Example frame from the training dataset. The white box indicates the ROI. (b) Foreground image segments computed using the GMM algorithm. (c-d) Maximum probability configuration of cuboids returned by our MCMC algorithm.

Marginal conditional likelihoods for the two cues were learned from the labelled training data and combined under a naïve Bayes assumption, and priors were learned from the proportion of the image occupied by ground truth bounding boxes. An initial segmentation is then determined by applying a threshold $p_0 = 0.08$ to the posterior ratio, and the resulting connected foreground components that exceed a criterion area $A_0 = 60$ pixels are identified as foreground segments. These thresholds were optimized to maximize the IOU of detected segments with ground-truth bounding boxes on the training dataset. Fig. 7(b) shows the detected foreground segments for an example video frame. In the following we will use the label GMM (Gaussian Mixture & Motion) to identify this foreground segmentation method.

3.2.2 3D Vehicle Configuration Optimization

Given a foreground image segment we wish to estimate the number of vehicles most consistent with the shape and size of the image segment, as well as the lane, location and class of each of these vehicles. As a measure of consistency we employ the intersection over union (IOU) of the image segment with the union of projected 3D cuboid vehicle models.

We first identify the lanes overlapped by the image segment and compute the centroid of each lane's portion of the segment. These centroids are then back-projected to the ground plane using our inverse homography H^{-1} (Eqn. 4) and the closest sample point on each lane's ground plane midline is identified as the origin for the search. The search space consists of seven locations, centred on the origin and spaced at 1m intervals along the midline.

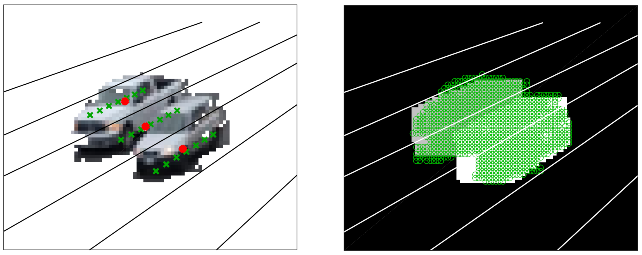
If we knew that a certain image segment was created by k vehicles in k specific lanes, there would still be a total of $(7 \times 4)^k = 28^k$ possibly configurations, given 7 locations per lane and 4 vehicle classes. Thus, given a foreground image segment that overlaps n lanes, the total number N of possible configurations is given by

$$N = \sum_{k=1}^{n} \frac{28^k n!}{(n-k)!k!} \tag{7}$$

We find that foreground image segments can overlap up to five active lanes, resulting in a total of more than 20 million possible configurations: too many to explore exhaustively, especially online. Instead, we employ a Markov Chain Monte Carlo (MCMC) method to explore the more probable regions of the configuration space within a reasonable amount of time.

We initialize the chain with a configuration computed using a simple greedy algorithm (Fig. 8). We first identify the lane with the largest overlap with the foreground image segment. We then exhaustively search the 28 possible location/class solutions within this lane, committing to the solution that maximizes the IOU with the whole segment. We then proceed to the lane intersecting the largest remaining unexplained portion of the foreground image segment and determine the solution in this lane that maximizes the IOU of the foreground image segment with the union of the two selected model projections. Note that it may be the case that for a particular lane no solution increases the IOU; in this case we assume no vehicle exists in this lane. This process continues until all intersected lanes have been considered.

Given this initial solution, we run MCMC, using the IOU as a model for the probability of each proposed configuration. Possible moves include: adding a vehicle in an unoccupied lane overlapped by the image segment, removing a vehicle, moving a vehicle to an adjacent location within a lane, and incrementing or decrementing the class of a vehicle by 1. Given this list of possible moves it is clear that any possible configuration can be reached. We imposed a time budget of 1 second per segment, which we found can accommodate 506 iterations of MCMC. We take as our solution the maximum probability configuration in the chain. Fig. 8(c-d) show the configurations selected for the segments in an example frame of the training video, in image and ground plane coordinates. We label our algorithm GMM3D to capture the combination of our GMM foreground segmentation algorithm with our 3D analysis-by-synthesis optimization of the vehicle configuration for each foreground segment.



(a)

(b)

Figure 8: Greedy initialization of our MCMC algorithm. (a) Green x show sampled traffic lane midline points and red dots indicate the centroids of the intersection of the foreground image segment with each lane. (b) The initial configuration selected by the greedy algorithm - Green area corresponds to the segment's area. Note that the furthest lane remains unoccupied since adding a vehicle there reduces the total IOU.

4. Results

We evaluate our proposed GMM3D method for traffic counting and compare against the state of the art on two test video clips. Camera tilt angles were automatically estimated to be $\phi = 78.3$ deg. (error = -2.9 deg.) for Dataset 1, and $\phi = 69.5$ deg. (error = -3.9 deg) for Dataset 2. We use three measures of performance: 1) Mean absolute error (MAE) per frame, 2) Accuracy of the total traffic flow over the duration of the clip (unique vehicle count), and 3) Accuracy of estimated vehicle dimensions.

4.1. Per-Frame Traffic Volume

To assess the value of 3D modelling for traffic analytics, we compare the performance of two variations of a 2D method against two variations of our 3D method. In the 2D methods, the number of foreground segments is used as an estimate of the number of vehicles in the frame. We consider two foreground segmentation algorithms: our GMM background subtraction + motion algorithm, adapted from [6], and the Principal Component Pursuit (PCP) method [11] that has been reported in previous work to outperform Gaussian mixture background subtraction algorithms - Code obtained from sites.google.com/a/istec.net/prodrig/Home/en/pubs/incpcp. Against these we compare two versions of our 3D algorithm: one using the GMM foreground segmentation as input, labelled GMM3D, and the second using the PCP foreground segmentation as input, labelled PCP3D.

The results are shown in Table 3. The two 2D methods perform comparably on both datasets. The GMM3D performs better than both 2D methods on both datasets, while the PCP3D method performs comparably on Dataset 1 but far better on Dataset 2.

Table 3: Per-frame volume estimation results. $\bar{n}$ is the mean ground-truth number of vehicles per frame.

Test Set 1			
	$\bar{n}$	MAE	MAE%
Test Set 1			
GMM	7.3	1.08	14.77
PCP	7.3	1.03	14.17
GMM3D	7.3	0.74	10.13
PCP3D	7.3	1.07	14.71
Test Set 2			
	$\bar{n}$	MAE	MAE%
GMM	9.4	2.91	30.93
PCP	9.4	2.50	26.56
GMM3D	9.4	1.90	20.22
PCP3D	9.4	1.37	14.58

4.2. Total Traffic Volume

We employ our GMM3D system to estimate total traffic volume (unique vehicle count) over our two test datasets, and compare against the particle method of Barcellos *et al.* [1], as the code is available online (obtained from https://www.researchgate.net/publication/278714978_Matlab_code_A_Novel_Video_Based_System_for_Detecting_and_Counting_Vehicles_at_User).

In the Barcellos method, a vehicle is detected and tracked as a group of particles, approximated by its convex hull. For each lane, a 'virtual loop' on the image is identified by hand. Any intersections between the convex hull representation of a vehicle and one or more virtual loops are identified. While one vehicle may intersect with more than one virtual loop, only the lane with the greatest intersection has its counter incremented. Barcellos *et al.* do not provide detailed rules for determining the location and width of these virtual loops. We therefore defined them to be qualitatively similar to those shown in their paper ([1], Figs. 8-9).

Since our GMM3D model assigns each vehicle to a specific lane, traffic counting is more straightforward. For each lane we identify a virtual gate in the image that is the projection of a ground plane line orthogonal to the lane boundaries. For each identified 3D vehicle model we extract its footprint, i.e., the contact surface between the model and the ground plane, which lies entirely within one lane (see Fig. 9). A lane counter is then incremented whenever a frame with a footprint on the gate follows a frame with no footprint on the gate. (This method works as long as two different vehicles never cross a line in two consecutive frames,

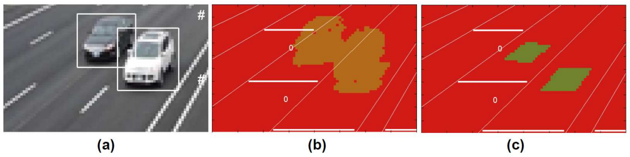


Figure 9: Measuring traffic volume. (a) Crop of example video frame. (b-c) GMM Foreground segment and GMM3D cuboid footprints, with virtual gates shown in white.

which would require vehicles speeds that exceed speeds we observe in our datasets.). We optimized the placement of these gates by maximizing their overlap with the ground truth boxes over the test datasets.

Quantitative results are summarized in Table 4. Our 3D method generates an average error of 12%, much lower than the mean error of the 2D method of Barcellos *et al.* (31%). Note that the 2D method consistently underestimates the traffic volume, due to a failure to disaggregate multiple vehicles that appear as a single image segment. This highlights the importance of 3D modelling for accurate traffic analytics.

Table 4: Total traffic volume estimation results.

	Ground-truth	Barcellos et al. [1]	GMM3D
Test Set 1	75	53	85
Test Set 2	59	39	53

4.3. Vehicle Classification

In addition to improving the accuracy of traffic volume estimates, the proposed GMM3D method produces a rough estimate of the vehicle dimensions, using the four size classes shown in Table 2. To evaluate the accuracy of these estimates, we analyze the classes assigned to vehicles when they cross a virtual gate (see previous section). We compare against ground truth estimates of vehicle dimensions, estimated by hand, and employing two measures of accuracy.

First, we consider categorical accuracy. Here we identify the ground truth category as the cluster whose centre lies closest to the ground truth dimensions in a Euclidean sense. Table 5 shows the confusion matrices for our two test sets. Results are fairly good, and most errors involve assignment to an adjacent category. We note that confusions between Categories 1 and 2 are not surprising, given that that the two corresponding training data clusters appear to be contiguous (Fig. 6). This motivates our second measure of performance. Here we measure the average error in estimated vehicle dimensions, compared with a baseline estimate that uses the mean dimensions over the training set for every test vehicle (Table 6). Our GMM3D method can be seen to generally produce surprisingly accurate estimates, within 24cm in all cases except for the length estimation in Test Set 1, which may be due to the misclassification of a few larger vehicles (Table 5).

Table 5: Confusion matrix for vehicle classification.

Test Set 1	GMM3D 1	GMM3D 2	GMM3D 3	GMM3D 4
GT 1	0.73	0.26	0	0
GT 2	0.24	0.72	0.04	0
GT 3	0.09	0.09	0.63	0.18
GT 4	0.09	0.09	0.27	0.54
Test Set 2	GMM3D 1	GMM3D 2	GMM3D 3	GMM3D 4
GT 1	0.88	0.11	0	0
GT 2	0.22	0.77	0	0
GT 3	0	0	1	0
GT 4	0	0	0	1

Table 6: Mean absolute error in vehicle dimension estimates.

Test Set 1			
	Length (m)	Width (m)	Height (m)
Baseline	2.72	0.23	0.67
GMM3D	1.66	0.08	0.24
Test Set 2			
	Length (m)	Width (m)	Height (m)
Baseline	1.37	0.15	0.33
GMM3D	0.23	0.02	0.03

5. Conclusions & Future Work

In this paper we have demonstrated that a 3D analysis-by-synthesis approach can be used effectively to disaggregate clusters of vehicles in highway traffic video, leading to improved estimates of traffic volume and vehicle dimensions. Future improvements may derive from the incorporation of learned likelihoods and priors into the MCMC search, tracking over time, accommodating lane changes, training and evaluation on larger and more diverse datasets, and efficient implementation to allow real-time deployment.

References

[1] P. Barcellos, C. Bouvié, F. L. Escouto, and J. Scharcanski. A novel video based system for detecting and counting vehicles at user-defined virtual loops. *Expert Systems with Applications*, 42(4):1845–1856, 2015. 1, 7, 8

[2] C. Bouvie, J. Scharcanski, P. Barcellos, and F. L. Escouto. Tracking and counting vehicles in traffic video sequences using particle filtering. *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 812–815, 2013. 1

[3] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 2

[4] E. R. Corral-Soto and J. H. Elder. Automatic Single-View Calibration and Rectification from Parallel Planar Curves. *Proceedings of the European Conference on Computer Vision*, pages 813–827, 2014. 2, 3

[5] M. Dubská, A. Herout, and J. Sochor. Automatic camera calibration for traffic understanding. *British Machine Vision Conference*, 2014. 2

[6] J. Elder, S. Prince, Y. Hou, M. Sizintsev, and E. Olevskiy. Pre-Attentive and Attentive Detection of Humans in Wide-Field Scenes. *International Journal of Computer Vision*, pages 47–66, 2007. 3, 5, 7

[7] S. Prince. *Computer Vision: Models, Learning and Inference*. Cambridge University Press, Cambridge, UK, 2012. 3

[8] J. Prokaj and G. Medioni. 3-d model based vehicle recognition. *Winter Conference on Applications of Computer Vision*, pages 1–7, 2009. 1

[9] J. Quesada and P. Rodriguez. Automatic vehicle counting method based on principal component pursuit background modeling. *IEEE International Conference onImage Processing*, pages 3822–3826, 2016. 1

[10] K. Ramnath, S. N. Sinha, R. Szeliski, and E. Hsiao. Car make and model recognition using 3d curve alignment. *IEEE Winter Conference on Applications of Computer Vision*, pages 285–292, 2014. 1

[11] P. Rodriguez and B. Wohlberg. A Matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling. *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3414–3416, 2014. 7

[12] J. Sochor, A. Herout, and J. Havel. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3006–3015, 2016. 2

[13] X. Song and R. Nevatia. A model-based vehicle segmentation method for tracking. *IEEE International Conference on Computer Vision*, 2:1124–1131, 2005. 1

[14] P. Yan, S. M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. *IEEE International Conference on Computer Vision*, pages 1–6, 2007. 1

[15] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. *IEEE International Conference on Computer Vision*, 1:666–673, 1999. 2