

# Traffic Flow Monitoring in Crowded Cities

John A. Quinn and Rose Nakibuule

Faculty of Computing & I.T.  
Makerere University  
P.O. Box 7062, Kampala, Uganda  
{jquinn,rnakibuule}@cit.mak.ac.ug

## Abstract

Traffic monitoring systems usually make assumptions about the movement of vehicles, such as that they drive in dedicated lanes, and that those lanes rarely include non-vehicle clutter. Urban settings within developing countries often present extremely chaotic traffic scenarios which make these assumptions unrealistic. We show how a standard approach to traffic monitoring can be made more robust by using probabilistic inference, and in such a way that we bypass the need for vehicle segmentation. Instead of tracking individual vehicles but treat a lane of traffic as a fluid and estimate the rate of flow. Our modelling of uncertainty allows us to accurately monitor traffic flow even in the presence of substantial clutter.

## Introduction

Traffic congestion is a problem in many developing world cities where road infrastructure is inadequate. The problem is compounded by a lack of real time information on traffic flows in a city, and a lack of facilities for optimising this flow (traffic lights and police manpower). In this paper we address the problem of gaining real-time data on congestion levels around a heavily congested city using computer vision. CCTV networks are in place in many such cities, motivated by security concerns, but under ordinary circumstances there is often a lack of sufficient staff to fully process data. Kampala, for example, has a network of around 50 CCTV cameras in the city centre.

Computer vision work on traffic monitoring usually assumes a constrained and clutter-free environment such as a highway, where cars move uniformly within separate lanes. In some situations, for example in many cities in the developing world, extremely chaotic traffic conditions make such assumptions unrealistic. Traffic monitoring is usually done in two stages: estimation of road geometry and tracking of vehicles. We can make both these processes robust by incorporating our uncertainty into a probabilistic dynamical model.

Figure 1 illustrates some typical, challenging scenarios for automated traffic monitoring – in particular extreme congestion and the presence of distracting visual features in the scene. Our approach to traffic monitoring differs from other systems in two main respects. First, we learn a model for



Figure 1: Typical, challenging traffic monitoring scenarios in Kampala, a developing-world city. The left panel shows an area near the entrance of a minibus taxi interchange, while the right panel shows an example of animals loose on the road.

classifying vehicles and visual clutter tailored to these conditions. Second, we do not attempt to track individual vehicles, which would be very difficult in the situations we consider. Instead we characterise a lane of traffic as a fluid and quantify the flow rate.

Our long term aim is to derive a real-time congestion map of the city. Such work helps police to know how to redirect traffic to optimise the available capacity, and which locations are priorities for dispatching traffic officers. This processing therefore helps to make the most of scarce infrastructure, using hardware which is already in place and currently under-utilised. For road users, such information would help in journey planning, thus reducing the economic penalty experienced by businesses associated with transport on inadequate roads.

In the next section we review existing work on traffic monitoring, generally designed for developed-world conditions. We then discuss techniques for assessing vehicle flow using feature transforms of the video frames. The flows are then used to calculate road geometry, and we describe a probabilistic dynamical model for inferring the overall rate of flow given observations.

## Related Work

Automated traffic monitoring usually breaks down into two tasks: estimation of road geometry and tracking of vehicles. Much of the previous work on this has been reviewed by (Kastrinaki, Zervakis, and Kalaitzakis 2003). What this

work has in common is the fact that individual vehicles are generally *tracked*. This necessarily involves segmenting individual vehicles in each frame, which brings up issues such as shadow detection and then need for sophisticated schemes to deal with occlusion. Previous work generally assumes a highway or other ideal environment where the only moving objects are the vehicles we wish to monitor, and the way in which they move is constrained, for example within specific lanes on the road.

The work in (Maurin, Masoud, and Papanikolopoulos 2002; 2005) is aimed at monitoring crowded urban scenes, including vehicles, pedestrians and cyclists. This work is again based on segmentation and tracking every object, without classifying the types of objects which are tracked. Vehicles are classified in the work of (Messelodi, Modena, and Zanin 2005) based on a 3D profile, and in the work of (Zhang, Fang, and Yang 2005) based on motion models to distinguish traffic from background movement, such as the motion of swaying trees in the background. A number of vehicle classification schemes are reviewed in (Sun, Bebis, and Miller 2006) from the perspective of an in-car monitoring system intended to alert the driver to obstacles. Again this work focuses in the segmentation of vehicles.

Our work attempts to make inferences about flow based on corresponding features across consecutive frames in a traffic video stream, without requiring segmentation or tracking. There has been some previous work along these lines (Porikli and Li 2004; Tan and Chen 2007) using hidden Markov model based classification of traffic, though in both cases assuming road conditions without clutter. For some background on analysing flow through feature extraction, see (Posta et al. 2003).

### Feature flow

Given a video stream of traffic, our first step is to take pairs of video frames and calculate the correspondence between the two, so that we can obtain flow vectors corresponding to every moving object. The approaches normally used for doing this are known as optic flow (calculating a flow vector for every pixel in the frame) and feature flow (calculating flow vectors only for points at which feature descriptors match between frames). We use the latter as it has advantages both in terms of speed and incorporation into object recognition, as explained in the next section. We use scale invariant feature transform (SIFT) keypoints (Lowe 2004) to identify the correspondences between frames. SIFT keypoints are vectors (usually 128-dimensional) which describe a visual feature in a representation which is invariant to rotation and scaling, and partially invariant to illumination changes and affine transformations. Keypoints are identified by applying difference of Gaussian convolutions to an image at different scales and looking for stable maxima and minima. The descriptors are then created based on local gradient data around the point of interest.

By calculating SIFT keypoints for two consecutive frames in a video sequence, we can match pairs of features by looking at Euclidean distance between them and then see which parts of the frame are moving and which are stationary.



(a)



(b)

Figure 2: Flow vectors between current and previous frames in different traffic video sequences. (a) All feature flow vectors in a scene, (b) flow vectors classified as vehicles only.

Figure 2(a) shows all flow vectors between the current and previous frames in part of a traffic sequence, where frames are  $\frac{1}{3}$  of a second apart. This includes the flows of pedestrians and other categories of object such as motorcycles. To determine where the road and vehicles are, we need to know which of these features are significant. We therefore have to filter out the flow vectors associated with clutter.

### Recognising vehicles

To distinguish different types of moving objects, we collected a database of training images, with 363 images of vehicles from different aspects, and 573 images of clutter on the road (e.g. pedestrians, motorcycles, background objects). Examples of training images from each class are shown in Figure 3. From these we obtained 55,764 vehicle keypoint descriptors and 49,671 clutter keypoint descriptors. For computational efficiency we reduced these keypoints down to a set 900 canonical descriptors using k-means clustering.

In Figure 5 we show keypoint matches from a training image of a vehicle to a test image in a similar perspective.

As well as looking at matching keypoint descriptors, it is also possible to use the length of a flow vector in classifying whether that flow is due to a vehicle or another type of object. We formulate a likelihood function  $p(l|t)$ , where  $l$  is the flow length in pixel units and  $t \in \{\text{vehicle, clutter}\}$ . We parameterise  $p(l|\text{vehicle})$  as a Gaussian distribution, whereas we set  $p(l|\text{clutter})$  to be an exponential decay plus a con-

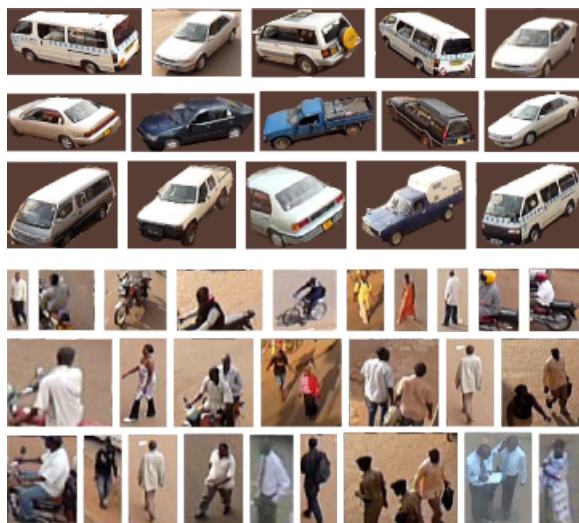


Figure 3: Examples of training image patches for vehicle and clutter classes.

stant, as shown in Figure 4. We set the parameters heuristically depending on the distribution of flow displacements found in each pair of frames. Clutter or background is most likely to be stationary or moving at low speed, though an object is also likely to be clutter if it is moving significantly faster than the vehicles in the scene (for example motorcycles moving quickly through gridlocked traffic). Vehicles may be stationary, but are more likely to move with some velocity.

Taking the matches from the feature flow, we cluster features together based on their position and velocity, again using k-means. Each cluster normally contains flow vectors from a single moving object, and we can take this group of keypoints and classify them. For each cluster of features, we can make a classification of whether it is a vehicle or clutter using Bayes rule with assumptions of conditional independence between the features (Naive Bayes), such that

$$p(t|k_1, \dots, k_n, v) \propto p(k_1|t) \times \dots \times p(k_n|t)p(v|t)p(t) \quad (1)$$

$$p(t = \text{vehicle}|k_{1:n}, v) + p(t = \text{clutter}|k_{1:n}, v) = 1 \quad (2)$$

where  $k_{1:n}$  are the keypoint descriptors in the cluster.

---

#### Algorithm 1 Calculation of vehicle flow vectors

---

1. Calculate feature flows for a consecutive pair of frames.
  2. Cluster the flows using k-means to approximately group the flows corresponding to the same object.
  3. Calculate the nearest neighbour to each feature descriptor from the training descriptors.
  4. Calculate the likelihood for the flow velocity.
  5. Use Naive Bayes to classify each group of flows.
  6. For the clusters classified as vehicles, return the mean position and velocity as a vehicle flow vector.
- 

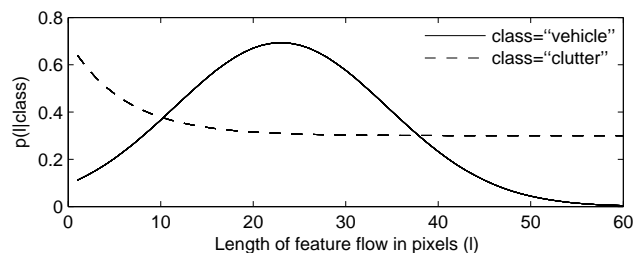


Figure 4: Likelihoods of inter-frame feature displacement lengths for different classes of flow.

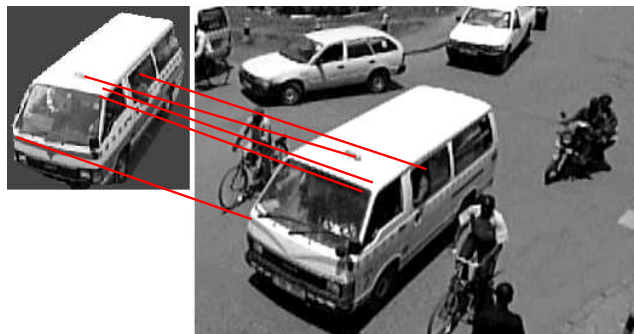


Figure 5: Keypoint matches from a vehicle training image to part of a video frame.

The likelihood terms  $p(k_i|t)$  are determined according to whether the nearest neighbour to  $k_i$  in the set of training descriptors belongs to the same class  $t$  or not. In this way we obtain a set of flow vectors corresponding to vehicles moving within the scene, for example by thresholding at 0.5. We do not try to segment or count individual vehicles. Figure 2(b) shows flow vectors in a scene which have been classified as vehicles (note that we regard motorcycles as clutter because their movements tend to be erratic).

Note that the form of the calculation in Eq. (1) makes it easy to add any number of other features for classification, such as colour histogram or morphological features, as long as we can in some way formulate the likelihood term  $p(\text{feature}|t)$ .

### Inferring observation geometry

Due to perspective and road layout, it does not make sense to simply average the velocity of the vehicle flow vectors. We assume that at each time frame there is a overall flow rate  $f_t$ , and that the velocity of each flow vector we observe is proportional to this flow rate, scaled by some constant. At any point  $\mathbf{o}$  in the observable field  $\mathcal{O}$  (such that  $\mathbf{o} \in \mathcal{O} \in \mathbb{R}^2$ ), we therefore need to know the scaling  $\mu(\mathbf{o}) \in \mathbb{R}_+$ .

For this task we use Gaussian process regression. We specify that the covariance between any two points  $\mathbf{o}$  and  $\mathbf{o}'$  is given by the function  $k(\mathbf{o}, \mathbf{o}')$ . In these experiments we parameterise  $k(\cdot, \cdot)$  as a squared exponential function with isotropic noise representing the variance in the observations.

From a sequence of training data, we calculate the vehicle flows as described in the previous section, then use the velocity and positions of these flows as training data. GP regression is then used to give the values of  $\mu$  and  $S$ , the mean and variance of the estimation at every observable pixel. For a location  $\mathbf{o} \in \mathcal{O}$ ,  $\mu(\mathbf{o})$  and  $S(\mathbf{o})$  are given by the GP predictive distribution (Rasmussen and Williams 2006, §2.2).

Figure 6 shows examples of this for two different traffic video sequences. In 6(a) we see one image from the sequence, in (b) we see the vehicle flows, in (c) we see the flow scaling  $\mu(\mathbf{o})$  for each pixel in the observed field, and in (d) we see the variance  $S(\mathbf{o})$ . This variance is higher where we have received fewer vehicle flow training vectors.

If required we could calibrate the scaling  $C(\mathbf{x})$  by manually selecting a point and giving its scale.

We monitor only the scale of the flow, and not the direction in this step. Learning the direction would entail having to study more complex dynamics such as vehicles crossing at an intersection, where the distribution of directions would be multimodal. An extension would be to look at correlations in the flow to calculate the number of lanes, and the directions in which vehicles move.

### Inferring flow rate

We model the dynamics of the flow rate  $f_t$  as being governed by

$$f_t = f_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, q^2). \quad (3)$$

That is, we assume that the flow follows a random walk with variance  $q^2$  at each step. A slightly more realistic model could be made by having this be mean-reverting around a nonzero value. The dynamics in Eq. (3) could in principle lead to a negative flow, which would be inconsistent. If this was a problem in practice we could model the square root or log of the flow, although this would make inference more difficult as we would then have nonlinear observations.

We now describe the observation model. At each time frame  $t$  we assume there are  $n_t$  different vehicle flow observations  $\mathbf{y}_t = y_{t,1}, \dots, y_{t,n_t}$ , each of which is dependent on the overall flow  $f_t$  and a noise component. Each observation  $y_{t,i}$  is associated with a location  $\mathbf{o}_{t,i} \in \mathcal{O}$ .

The flow observations are distributed according to the overall flow rate and the scaling field

$$y_{t,i} \sim \mathcal{N}(f_t(\mathbf{C}_t)_{i,1}, (\mathbf{R}_t)_{i,i}) \quad (4)$$

where we have to formulate the observation parameters  $\mathbf{C}_t$  and  $\mathbf{R}_t$  at each time step, depending on the locations of the observed flow vectors. We use  $(\mathbf{C}_t)_{i,1} = \mu(\mathbf{o}_{t,i})$  and  $(\mathbf{R}_t)_{i,i} = S(\mathbf{o}_{t,i})$ , as calculated in the previous section. The dynamical noise parameter  $q$  can be calculated with expectation-maximisation. The overall generative model is depicted in Figure 7.

Given this model we can use inhomogenous Kalman filter equations to infer a mean and variance ( $\hat{f}_t$  and  $P_t$ ) of the estimated overall flow rate through time. Assuming some initial estimates  $\hat{f}_0$  and  $P_0$ , we first predict the distribution on the flow rate having the same mean as in the previous

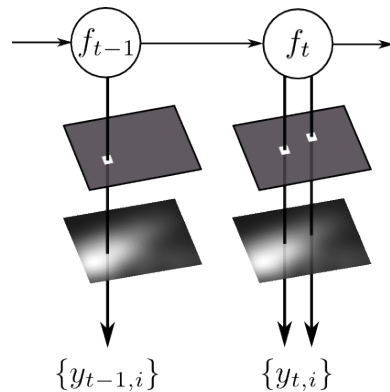


Figure 7: Dynamical model for vehicle flow observations generated by evolving global flow rate. As the flow rate  $f_t$  evolves, at each time frame  $n_t$  flows are generated from it (positions illustrated in the upper plane), and scaled according to the observation field (depicted in the lower plane) to give the observations.

time frame but with increased variance:

$$\hat{y}_t^- = \hat{y}_{t-1} \quad (5)$$

$$P_t^- = P_{t-1} + q^2. \quad (6)$$

The update equations then take into account the observed flows, and the associated measurement variance. The rest of the steps to complete the estimate are

$$\tilde{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{C}_t \hat{f}_t^- \quad (7)$$

$$\mathbf{K}_t = P_t^- \mathbf{C}_t^\top (\mathbf{C}_t P_t^- \mathbf{C}_t^\top + \mathbf{R}_t)^{-1} \quad (8)$$

$$\hat{f}_t = \hat{f}_t^- + \mathbf{K}_t \tilde{\mathbf{y}}_t \quad (9)$$

$$P_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) P_t^- \quad (10)$$

These expressions differ from the standard Kalman filter equations only in that  $\mathbf{C}_t$  and  $\mathbf{R}_t$  change at each time step.

### Inference on sample data

We show the results of inferring flow rate in Figure 8. Here we took two short video sequences (180 frames at 3 frames per second) in which traffic flow was visibly changing. After learning the scaling field, we applied the inference recursions in Eqs. (5-10). We can see these changes in the estimated flow. In the first example, traffic was regularly speeding up and slowing down at an intersection. In the second example, passenger carrying vehicles pulled into the side of the road periodically, interrupting the flow of the traffic.

### Discussion

In this paper we have discussed solutions to the problem of traffic congestion monitoring in a developing-world city where traffic scenes have a high degree of clutter. This is done by modelling an observed lane of traffic as a single moving entity with an overall rate of flow. By classifying feature descriptors we can incorporate the flows of vehicles without being affected by the flow of non-vehicle objects.

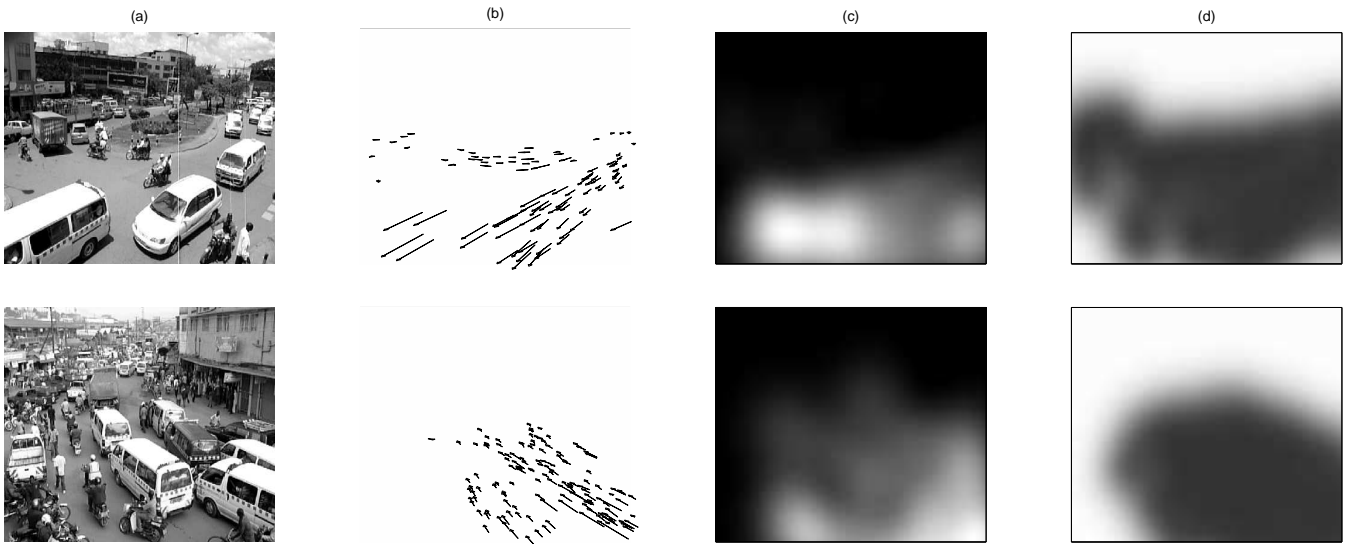


Figure 6: Inferred observed field for different video sequences. (a) Initial frame in sequence, (b) the first 100 vehicle flows found in the sequence, (c) the observation field, lighter shades indicating that vehicles travel faster within that region, (d) the corresponding variance, lighter shades indicating greater uncertainty. The more vehicle flows we observe, the greater the precision with which we can reconstruct the observation and road geometry.

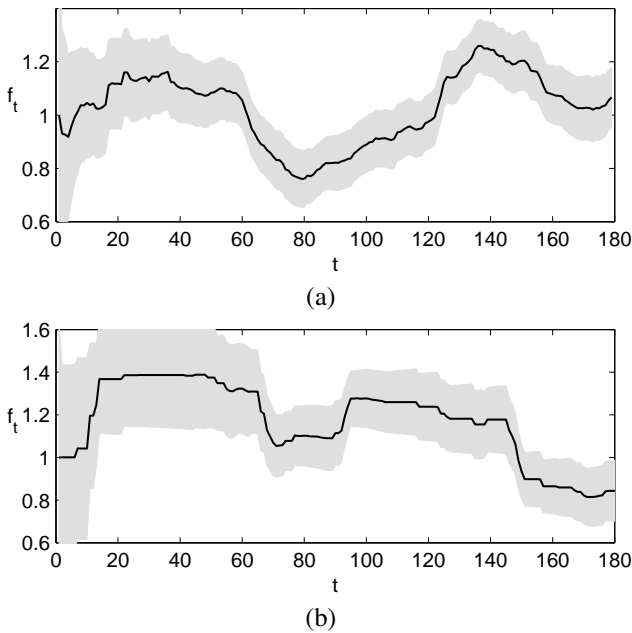


Figure 8: Estimated flow rates in two different video sequences. (a) Traffic speeds up and slows down periodically near an intersection; (b) traffic flow changes as buses and minibus taxis pull in to the side of the road to let passengers on and off. The shaded areas show two standard deviations of the estimated distribution.

This work takes us closer to our overall aim of dynamically mapping the congestion of Kampala’s city centre using its network of around 50 cameras. Our work on extracting information from traffic video streams could also fit into higher level work on traffic prediction and monitoring (see e.g. (Horvitz et al. 2005)).

Our Matlab implementation takes a few seconds to process each frame, whereas real time is three frames per second. Where processing power is limited, such as when dealing with many video streams simultaneously, we envisage periodically processing video streams (for example processing a 1-minute sequence every 15 minutes).

Future work includes calculating road geometry more thoroughly, to deal with separate lanes each with their own flow, and intersections where the direction of flow might change. Also, much recent work on traffic monitoring has been looking not at rates of flow but on detecting unusual occurrences on the roads. Uganda Police currently use CCTV for such a purpose, looking out for accidents or dangerous driving, so it would also be interesting to adapt our system to detect unusual behaviour on the road, by looking for vehicle flows which are unlikely under the model.

Finally, the different probabilistic elements in our current system are decoupled – for instance, our state of certainty about whether a flow is associated with a vehicle or not is lost when we make a hard decision before going on to the next stage of inferring the observed geometry. Better results could be expected if we could use one model to represent a joint distribution over all these variables.

## Acknowledgments

Thanks to Yusuf Ssewanyana and Uganda Police for providing advice and access to their CCTV systems.

## References

- Horvitz, E.; Apacible, J.; Sarin, R.; and Liao, L. 2005. Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service. In *Proceedings of the 21st Conference on Uncertainty and Artificial Intelligence*.
- Kastrinaki, V.; Zervakis, M.; and Kalaitzakis, K. 2003. A survey of video processing techniques for traffic applications. *Image and Vision Computing* 21:359–381.
- Lowe, D. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.
- Maurin, B.; Masoud, O.; and Papanikolopoulos, N. 2002. Monitoring Crowded Traffic Scenes. In *Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems*.
- Maurin, B.; Masoud, O.; and Papanikolopoulos, N. 2005. Tracking All Traffic: Computer Vision Algorithms for Monitoring Vehicles, Individuals, and Crowds. *IEEE Robotics and Automation Magazine* 12(1):29–36.
- Messelodi, S.; Modena, C. M.; and Zanin, M. 2005. A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Analysis and Applications* 8:17–31.
- Porikli, F., and Li, X. 2004. Traffic Congestion Estimation Using HMM Models Without Vehicle Tracking. In *Proceedings of the 2004 IEEE Intelligent Vehicles Symposium*.
- Posta, F.; Vrolijk, B.; Hauser, H.; Laramée, R.; and Doleisch, H. 2003. The State of the Art in Flow Visualisation: Feature Extraction and Tracking. *Computer Graphics Forum* 22(4):1–17.
- Rasmussen, C., and Williams, C. 2006. *Gaussian Processes for Machine Learning*. MIT Press.
- Sun, Z.; Bebis, G.; and Miller, R. 2006. On-Road Vehicle Detection: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(5):694–711.
- Tan, E., and Chen, J. 2007. Vehicular Traffic Density Estimation via Statistical Methods with Automated State Learning. In *Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*.
- Zhang, W.; Fang, X.; and Yang, X. 2005. Moving vehicles segmentation based on Bayesian framework for Gaussian motion model. *Pattern Recognition Letters* 27(9):956–967.