

Framework for Illumination Invariant Vehicular Traffic Density Estimation

Pranam Janney and Glenn Geers

Dept of Computer Science and Engineering,
University of New South Wales, Australia
and

National ICT Australia (NICTA) **
Sydney, Australia

{pranam.janney, glenn.geers}@nicta.com.au

Abstract. CCTV cameras are becoming a common fixture at the roadside. Their use varies from traffic monitoring to security surveillance. In this paper a novel technique, using Invariant Features of Local Textures (IFLT) & Support Vector Machine (SVM), for estimating vehicular traffic density on a road segment is presented. The proposed approach is computationally efficient and robust to varying illumination. Experimental results have shown that the proposed framework can achieve high performance than extant state-of-the-art techniques in varying illumination conditions.

Key words: Intelligent Transport Systems (ITS), Invariant Features of Local Textures (IFLT), Support Vector Machines (SVM), density estimation, traffic information, parameters, illumination invariance

1 Introduction

Intelligent Transportation Systems (ITS) are used in many cities to provide information about traffic conditions on the road network and as an aid to streamlining vehicular traffic flow in an effort to reduce traffic congestion. An ITS typically uses various kinds of sensors, such as video cameras and inductive loop detectors to measure the significant properties of vehicular traffic flow. Traffic density is a useful property that an ITS can use to perform *higher level* functions such as traffic light sequencing.

Video monitoring systems promise many advantages over the now-dominant inductive loop detectors which are point detectors that sense the vehicles passing over them [1]. Cameras are cost effective and easier to maintain than other road-mounted sensors. They also offer the potential of providing a much richer data stream than the simple loop.

Vehicle tracking is a popular technique for traffic density estimation [5, 13, 4]. Vehicle tracking algorithms are generally flexible enough to determine almost any type

** NICTA is funded by the Australian Federal Government as represented by the Department of Broadband, Communications and the Digital Economy, the NSW Department of State and Regional Development, the ACT Government and the Australian Research Council through the ICT Centre of Excellence Program.

of traffic information. But in terms of just determining the traffic density, the performance of vehicle tracking algorithms tends to degrade in heavy traffic situations due to occlusion, cluttering and false background estimation.

Porikli and Li published two closely related papers [6, 10] that used the learning from low-level features technique. Their main idea is to extract low-level features from traffic video and learn the traffic density state *a priori*. In [12] it has been rightly pointed out that the drawback of this technique is that the motion information used in their implementation is produced by the video encoder, and this does not represent true motion but the best match with respect to some dissimilarity measure (e.g. Sum of Absolute Differences). Also, Gaussian Mixture Hidden Markov Models (GMHMM) are used in their implementation but the number of Gaussians used per state is unknown while the structure of the Hidden Markov Model (HMM) is determined empirically in [10] and *a priori* in [6].

In [12], a technique similar to that described above is proposed. Low-level features are directly extracted from the traffic video. The number of Gaussians and HMM structure is determined from information provided by an unsupervised learning scheme called AutoClass [2]. The features used are a combination of different types of texture statistics and Edge Histogram Descriptors (EHID) proposed for MPEG-7 [14]. Even though the framework uses very simple low-level features, classification using an HMM and Autoclass is computationally inefficient. Also, using edge features could prove problematic in varying illumination conditions as varying illumination changes the appearances of edges thereby leading to confusing feature sets. There is a need to develop systems that can analyze surveillance footage especially in the field of vehicular traffic analysis in varying illumination conditions.

In this paper, we propose a novel varying illumination resilient technique for estimating traffic density which uses Invariant Features of Local Textures' descriptors [3] and Support Vector Machines (SVM) [11]. For the purpose of this paper *traffic density* is defined as the percentage of the Region-of-Interest (ROI—usually a lane segment) occupied by vehicles. The system framework and detailed description is presented in Section 2 and preliminary experimental results and analysis are provided in Section 3.

2 Traffic lane density Estimation

We consider an ROI within which we need to estimate the traffic density. Invariant Features of Local Textures (IFLT) [3] are used to generate local texture descriptors to represent the ROI.

For every input frame, IFLT descriptors of the ROI are generated and assembled into a feature vector. The feature vectors are input to an SVM in order to, classify the traffic density of the particular traffic lane into one of four states:

1. *Empty*: less than 5% of the lane is occupied by vehicles.
2. *Low*: 5–30% of the lane is occupied by vehicles.
3. *High*: 30–90% of the lane is occupied by vehicles.
4. *Full*: more than 90% of the lane is occupied by vehicles.

Manually labelled video sequences were used for training the SVM.

The architecture of the traffic lane density estimation system described in this paper is shown schematically in Figure 1.

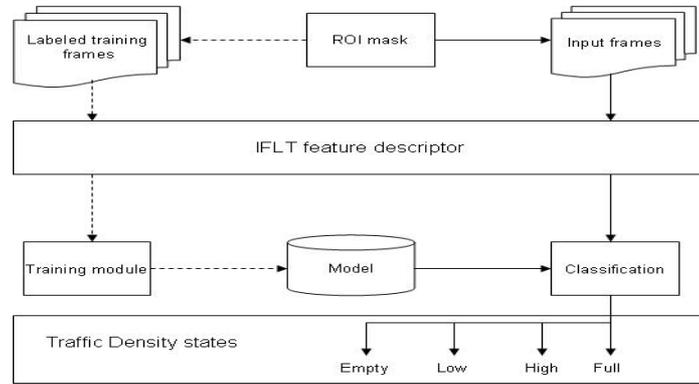


Fig. 1. Framework for Traffic lane density estimation using IFLT descriptors.

Sample frames from video sequence captured by traffic cameras on Anzac Parade/Barker Street junction with ROI lane which denotes the four different density states are shown in Figure 2.



Fig. 2. *Empty, Low, High and Full* states of a lane of traffic at the Anzac Parade/Barker street junction. (the circular reflection is that of the camera lens onto the window of the camera housing)

Traffic density is being estimated for a ROI covering one traffic lane. We use IFLT to generate feature descriptors. IFLT is a texture descriptor that is scale, rotation and (essentially) illumination invariant [3]. The density of local textures will provide an approximate of the coverage of traffic lane. Experimental results have demonstrated that the IFLT descriptors exceeds performance of previously published state of the art local texture algorithms at a significantly lower computational cost [3]. Traffic surveil-

lance cameras are subjected to varying illumination due to changes in sunlight . IFLT descriptors are very robust towards illumination variation and are, therefore, eminently suited for traffic density estimation.

A detailed explanation of IFLT descriptors [3], dimension selection process and a brief explanation regarding the usage of Support Vector Machine is provided in the sections below.

2.1 Invariant Features of Local Textures (IFLT)

Consider a 3×3 neighbourhood of pixels. True circular symmetry around X_c can be achieved by recalculating pixel intensities at the co-ordinates given by,

$$X_i = \left(R \cos \frac{2\pi i}{p}, R \sin \frac{2\pi i}{p} \right) \quad (1)$$

where X_i is the equivalent position of the i^{th} of $p - 1$ pixels in circular symmetry around X_c with radius R . In the work that follows R is set to unity.

The gray values of neighbors which do not fall exactly on integral pixels are estimated by interpolation. With I_c as the intensity of the centre pixel, the gradient of intensity in all directions with reference to the centre pixel are computed.

The gradient intensities around a centre pixel can be re-written as a one-dimensional vector, \mathbf{I} , as shown in Equation 2,

$$\mathbf{I} = [I_c - I_0, \dots, I_c - I_7] \quad (2)$$

where I_c is the intensity of the centre pixel and $I_{(0..7)}$ are the intensities of the neighbouring pixels.

This one-dimensional vector is further normalised.

$$\mathbf{I}_{norm} = \frac{\mathbf{I}}{\max(\mathbf{I})} \quad (3)$$

The discrete wavelet transform (DWT) of the signal I_{norm} is calculated by passing it through a series of filters [7]. In this work Haar wavelets were used because of their computational efficiency. The required filter coefficients are given in Equation 4.

$$h = \left[\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}} \right], g = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \quad (4)$$

The signal is decomposed simultaneously using a high-pass filter h and a low-pass filter g . The outputs of the high pass filter and low pass filter are known as the *detail* and *approximate* coefficients respectively.

Noting that the wavelet transform operation corresponds to a convolution followed by downsampling by 2 allows the filter outputs to be written more concisely as,

$$y_{low} = (I_{norm} * g) \downarrow 2, y_{high} = (I_{norm} * h) \downarrow 2 \quad (5)$$

where \downarrow is used to denote the Downsampling Operator.

The detail and approximate coefficients have shift invariant energy distributions. As shown above, rotations in image space have been transformed into linear shifts in transform space and so the energy distribution of the detail and approximate coefficients are also rotation invariant. In the experiments described below the mean and standard deviation of the high pass and the low pass filter outputs generated by one step of the wavelet transform of Equation 2 are used as the texture features. These features are inherently intensity and rotation invariant for a small 3×3 neighborhood of pixels.

A histogram is built from the extracted local texture features in the texture patch. This involves partitioning the 4-dimensions of texture features (mean and the standard deviation of the energy distributions of the high pass and the low pass wavelet bands) into a number of bins and calculating the number of occurrence of local texture feature values in those bins. In our current set-up we use 32-bin histogram.

2.2 Dimension Selection

We have used 32-bin histograms, thus our feature vector has minimum 32 dimensions. Figure 2.2 shows a representative plot of average of features of each state in each dimension for a video sequence (*Anzac Parade/Barker Street Jtm -1*). It is clearly evident that features from some of the dimensions were not distinctive enough to discriminate between the four density states.

Consider, $e_{N,D}$ where $e_{N,D}$ is a matrix of N samples representing *Empty* state with D dimensions. Similarly, we have, $l_{N,D}$, $h_{N,D}$ and $f_{N,D}$ corresponding to *Low*, *High* and *Full* states.

Considering $e_{N,D}$ we have,

$$\bar{e}_j = \frac{1}{N} \sum_{i=1}^N e_{i,j} \quad (6)$$

$$\sigma_{e_j} = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_{i,j} - \bar{e}_j)^2} \quad (7)$$

where $j = 1, \dots, D$; and \bar{e}_j and σ_{e_j} are the average and standard deviation models of the ‘empty’ state respectively. $[\bar{l}_j, \sigma_{l_j}]$, $[\bar{h}_j, \sigma_{h_j}]$ and $[\bar{f}_j, \sigma_{f_j}]$ corresponding to ‘low’, ‘high’ and ‘stop’ states are calculated similarly.

Next, the contribution of each dimension in one state towards distinguishing that particular state from the other three states is determined as follows:

Consider the ‘empty’ state model (i.e. $[\bar{e}, \sigma_e]$) and the ‘low’ state model (i.e. $[\bar{l}, \sigma_l]$),

1. The distance between the two states is given by

$$\text{dist}_{e \rightarrow l, j} = |(\bar{e}_j - \bar{l}_j)| + |(\sigma_{e_j} - \sigma_{l_j})| \quad (8)$$

where, $j = 1, \dots, D$ and $\text{dist}_{e \rightarrow l}$ is a vector in which the j^{th} element represents the distance between the corresponding states with respect to the j^{th} dimension.

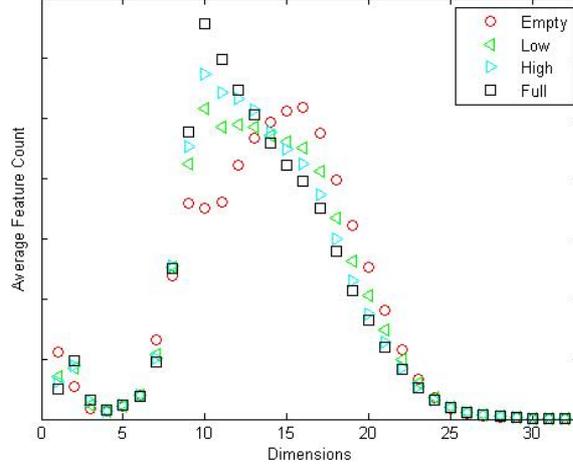


Fig. 3. Average of features for each dimension of each state clearly showing the dimensions that help in discriminating between four density states

2. Calculate a threshold using Equation 9, which can be used to eliminate dimensions that do not particularly help in distinguishing between the two states.

$$T = \frac{1}{D} \sum_{j=1}^D \text{dist}_{e \rightarrow l, j} \quad (9)$$

3. Select dimensions by comparing the distance generated by each dimension with the threshold, T .

$$p_{e \rightarrow l, j} = \{j | (\text{dist}_{e \rightarrow l, j} \geq T)\} \quad (10)$$

where, $j = 1, \dots, D$ and $p_{e \rightarrow l}$ is a set of dimensions.

Repeat steps 1 through to 3, for all combinations of states i.e $e \rightarrow h, e \rightarrow s, l \rightarrow h, l \rightarrow f, h \rightarrow f$.

Hence, the dimensions, P , which are best suited to distinguish states from one another are given by,

$$P = \{p_{e \rightarrow l} \cup p_{e \rightarrow h} \cup p_{e \rightarrow s} \cup p_{l \rightarrow h} \cup p_{l \rightarrow s} \cup p_{h \rightarrow s}\} - \{0\} \quad (11)$$

In our tests, features from 19 dimensions out of the available 32 could clearly discriminate between all the different states and these 19 dimensions are used as our feature descriptors.

2.3 Support Vector Machine

SVM has extraordinary generalisation capacity and is also one of the popular large margin classifiers [11]. An SVM constructs a binary classifier from a set of labeled

patterns called *training samples*. Let $(x_i, y_i) \in \mathbb{R}^N \times \{\pm 1\}, i = 1, \dots, l$ be such a set of *training samples*. The purpose is to select the function $f_\alpha : \mathbb{R}^N \rightarrow \{\pm 1\}$ from a given class of functions $\{f_\alpha : \alpha \in A\}$ such that f will correctly classify test examples (x, y) . The reader is referred to [11] for details.

We use three different types of kernels in our experiments.

1. Linear kernel: $K(x, y) = (x \cdot y)$
2. Polynomial kernel: $K(x, y) = (x \cdot y)^p$, where $p = 2$
3. Gaussian kernel: $K(x, y) = \exp(-\frac{1}{2\sigma^2}|x - y|^2)$, where $\sigma = 0.70$ (σ is optimised via cross-validation)

3 Experimental Setup

In our experiments we have used video sequences from a busy intersection located just outside the Sydney CBD (Anzac Parade/ Barker Street Junction). All video sequences used for this experiment were captured during daytime and exhibit the expected range of natural lighting variation. Each video sequence is of 6 minutes—9000 frames—duration. Resolution of each frame is 320×240 . In our experiments we have used only one lane as the ROI. Figure 2 shows four sample frames (with ROI shown in red) that are representative of the *Empty*, *Low*, *High* and *Full* traffic density states used in this paper. Table 1 details the illumination effects present in each of the video sequences.

Sequence	Effect of Illumination
Anzac Parade/Barker Street Jtn. - 1	Bright/Sunny
Anzac Parade/Barker Street Jtn. - 2	Raining
Anzac Parade/Barker Street Jtn. - 3	Cloudy
Anzac Parade/Barker Street Jtn. - 4	Sun-glare

Table 1. Illumination effects present in each of the video sequences used for testing. All these video sequences were captured by traffic cameras installed by the Roads and Traffic Authority of New South Wales, Australia

3.1 Experiment-I

For Experiment-I we implemented the method described in [12] and reproduced the reported results. We tested our algorithm on video sequences containing both cloudy and sunny conditions captured during the daytime. The test as described in [12] comprises of training a model using 70% of samples from a particular sequence and testing the model with the remaining 30% of samples from the same sequence.

Table 2 shows the comparison of performance results for the framework in [12] and the proposed framework (using SVM with gaussian kernel, $\sigma = 0.7$). Under conditions of uniform illumination both methods have comparable performance. However, the current technique achieves (as expected) better results under conditions of varying

Illumination	Framework [12],(AutoClass/Kmeans)		Proposed Framework	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
Sunny only	98.9 / 98.8	95.4 / 96.9	99	98.1
Cloudy only	99.5 / 99.3	97.8 / 98.3	99.2	97.9
Sunny and cloudy	99.8 / 99.2	94.4 / 83.3	99.8	97

Table 2. Comparison of performance results of framework in [12] and the proposed framework.

illumination. It is also interesting to note that the performance results of the proposed approach is very consistent for all tests. We also used SVM with linear/polynomial kernels but for this experimental setup, the results were inferior compared to that of SVM with gaussian kernel. Hence, those results have been omitted from this paper.

Computation time required for the framework of [12] is very high. Using the proposed framework, we were able to process 1-2 frames per second using unoptimized Matlab implementations of the methods, running on a PC with Pentium 3.2GHz processor. It seems likely that the proposed framework will be capable of running in real-time when implemented in C or C++.

We feel that experimental set-up in [12] is a very lenient test for illumination invariant traffic density estimation. Primary reason being that the illumination variations included only sunny and cloudy sequences. And the second but very important reason is that for a particular test sequence, 70% of frames are used for training and the remaining 30% is used for testing. So, considering "Sunny" sequence, model is trained using 70% of frames from this sequence and the rest 30% of the frames from the same sequence is used for testing. Trained model is being tested with samples/frames with illumination effect, which the model has already learnt. Many of the parameters required in [12] are *ad hoc*, which again is not feasible for real world applications.

Thus, we feel this experimental setup is not a valid test for illumination invariant traffic density estimation.

3.2 Experiment-II

This set of experiments were conducted on all combinations of sequences listed in Table 1. Sequences used to train the model and then to test it have different illumination effects.

The results presented are average of classification accuracies for a sequence, when tested on a model that is trained by a sequence other than itself.

As seen from Table 3, the proposed framework shows consistent classification accuracy for all sequences when the testing and training sequences have different illumination effects. Previously, The difference in performance between linear, polynomial and gaussian kernels is negligible.

The results for the proposed framework presented in Table 2 was achieved using SVM with gaussian kernel ($\sigma = 0.7$). Comparing performance of the proposed framework from Tables 2 and 3, it is evident that SVM with a gaussian kernel does not outperform SVM with a linear kernel when testing sequence has a different illumination effect than that of the training sequence. This could be an effect of noisy data or

Sequence	Linear kernel (%)	Polynomial kernel (%)	Gaussian kernel(%)
Anzac Parade/Barker Street Jtn. – 1	81.3	80.2	80.2
Anzac Parade/Barker Street Jtn. – 2	78.7	79.1	77.1
Anzac Parade/Barker Street Jtn. – 3	77.1	76.5	75.4
Anzac Parade/Barker Street Jtn. – 4	79.1	78.2	76.8

Table 3. 4-State classification accuracy for the proposed framework where testing sequence and training sequence have different illumination effect.(%) denoting the percentage of correct classifications.

may be a consequence of the illumination invariant nature of IFLT. During the course of our initial experiments we noticed that it is difficult to reduce the error rate of *High* states being misclassified mostly as *Full* or sometimes as *Low*. Looking at frames of our annotated training sequence, we realised that annotation of frames whose density state is in-between two states is very subjective to the person annotating the sequence. Especially if the density state of frames lie in-between *High* and *Full* then it is hard, even for the person annotating the sequence, to arrive at a decision as to whether the frame has to be annotated as *High* or *Full*. Hence, subjective annotation is a major contributor to noisy data.

Sequence	Texture Statistics,EHID [12]			LBP [9]			IFLT		
	l-SVM	p-SVM	g-SVM	l-SVM	p-SVM	g-SVM	l-SVM	p-SVM	g-SVM
AP/BS Jtn. – 1	48.2	51.8	51.9	73.1	71.2	72.8	81.3	80.2	80.2
AP/BS Jtn. – 2	40.1	41.1	41.1	73.5	73.2	69.1	78.7	79.1	77.1
AP/BS Jtn. – 3	48.2	48.2	47.9	72.9	71.1	70.9	77.1	76.5	75.4
AP/BS Jtn. – 4	45.1	42.9	42.8	70.4	71	70.1	79.1	78.2	76.8

Table 4. Comparison of average classification accuracies between IFLT,Texture statistics,EHID [12] and LBP [9] using different classification algorithms, when testing and training sequences have different illumination effect. *l – SVM*, *p – SVM*, *g – SVM* denotes SVM with Linear, Polynomial and Gaussian kernels respectively. AP/BS = Anzac Parade/Barker Street

Table 4 shows the performance comparison between IFLT feature descriptors , texture statistics features [12] and Linear Binary Patterns (LBP) [9] in the proposed framework. Texture statistics features used in [12] are a combination of first & second order texture statistics and Edge Histogram Descriptors (EHID) proposed for MPEG-7 [14]. Linear Binary Patterns are state-of-the-art local texture descriptors proposed by [9]. A detailed description of LBP can be found in [8]. IFLT features have very high classification accuracy when compared to that of texture statistics proposed in [12] and LBP [9] for illumination invariant traffic density estimation.

Results from Tables 3 and 4, indicate that a SVM with a linear kernel is the best option for classification because it provides good classification accuracy and is com-

putationally less intense. Hence, IFLT descriptors are the best descriptors for use in illumination invariant traffic density estimation.

Considering noisy data due to subjective annotation, we conducted tests where *High* and *Full* were merged as one state i.e. *High* state. We also divided the ROI into N regions, using IFLT descriptors for each region. Using N regions would provide us with more local information thus improving the performance. In our experiments we have used 8-regions.

Sequence	4-States			3-States		
	1-region	8-regions	Combination(1,8)	1-region	8-regions	Combination(1,8)
AP/BS Jtn. – 1	81.3	82.2	85.1	84.7	87.4	92.4
AP/BS Jtn. – 2	78.7	79.5	83.2	83.6	86.9	91.1
AP/BS Jtn. – 3	76.5	76.1	83	82.9	83.1	89.9
AP/BS Jtn. – 4	79.1	77.2	83.1	83.5	84.8	90.8

Table 5. Comparison of average classification accuracies between 4-state and 3-state classification with 1-region/8-regions using SVM linear kernel for classification. Testing and training sequence had different illumination effects. AP/BS = Anzac Parade/Barker Street.

Table 5 shows that merging two states, i.e. reducing a 4-state classification problem to 3-state classification problem, we can achieve higher performance. This also supports the fact that subjective annotation is a major contributor for noise in 4-state classification problem. It is interesting to note that dividing ROI into 8-regions did not improve performance by a big margin. However, by close observation of prediction outputs we realised that output predictions for 8-regions based approach were complimenting the errors of output predictions from the 1-region based approach. Table 5 presents results for *combination(1,8)* based approach where in we can achieve classification accuracies in excess of 90% irrespective of varying illuminations. Thus, we were able to achieve higher performance by combining global and local information available from the ROI.

4 Conclusion

In this paper, we have presented a new framework for vehicular traffic density estimation using IFLT descriptors and SVMs. The framework uses illumination invariant feature descriptors in conjunction with an SVM with a linear kernel, for classification. Both algorithms are computationally efficient. Although we were only able to process 1–2 frames per second in Matlab on a Pentium 3.2GHz class machine, our technique should be capable of working in real-time when coded in C or C++. We were able to achieve in excess of 90% correct classification of vehicle density states in varying illumination conditions.

The authors would like to take this opportunity to point out the need for an annotated traffic video database, which could be used as ground truth for research in this area. Such databases are widely available in other image processing specialities (such as face

recognition) but are sadly lacking in others, making performance comparison difficult. In due course, we hope to make our data publicly available.

5 Acknowledgments

The assistance of the Roads and Traffic Authority of New South Wales in obtaining the video sequences is gratefully appreciated.

The authors would also like to thank Dr Jing Chen and Mr Sakrapee Paisitkriangkrai for useful discussions during the preparation of this paper.

References

1. D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 495–501, 1997.
2. P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, pages 153–180, 1996.
3. P. Janney and Z. Yu. Invariant features of local textures - a rotation invariant local texture descriptor. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)(BP-Workshop)*, June 2007.
4. S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. In *IEEE Trans. on ITS*, volume 1, June 2000.
5. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the International Conference on Pattern Recognition*, Israel, 1994.
6. X. Li and F. M. Porikli. A hidden markov model framework for traffic event detection using video features. In *Proceedings of International Conference on Image Processing (ICIP)*, 2004.
7. S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, 1999.
8. T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(7):971–987, July 2002.
9. T. Ojala, M. Pietikinen, and T. Menp. Gray scale and rotation invariant texture classification with local binary patterns. In *In Proc. European Conference on Computer Vision (ECCV)*, pages 404–420, 2000.
10. F. Porikli and X. Li. Traffic congestion estimation using hmm models without vehicle tracking. In *IEEE Intelligent Vehicles Symposium*, pages 188–193, June 2004.
11. J. Shawe-Taylor and N. Cristianini. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.
12. E. Tan and J. Chen. Vehicular traffic density estimation via statistical methods with automated state learning. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, September 2007.
13. B. Tseng, C.-Y. Lin, and J. Smith. Real-time video surveillance for traffic monitoring using virtual line analysis. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pages 541–544, August 2002.
14. C. S. Won, D. K. Park, and S.-J. Park. Efficient use of mpeg-7 edge histogram descriptor. *ETRI Journal*, 24(1), February 2002.