Automatic Daytime Road Traffic Control and Monitoring System

P.F. Alcantarilla, M.A. Sotelo, L.M. Bergasa
Department of Electronics
University of Alcalá
CAMPUS. 28805 Alcalá de Henares (Madrid), Spain.
E-Mail: pablo.alcantarilla, sotelo, bergasa@depeca.uah.es

Abstract— **This paper presents an automatic road traffic control and monitoring system for daytime sequences using a B&W camera. Important road traffic information such as mean speed, dimension and vehicles counting are obtained using computer vision methods. Firstly, moving objects are extracted from the scene by means of a frame-differencing algorithm and texture information based on grey scale intensity. However, shadows of moving objects belong also to the foreground. Shadows are removed from the foreground objects using top hat transformations and morphological operators. Finally, objects are tracked in a Kalman filtering process, and parameters such as position, dimensions, distance and speed of moving objects are measured. Then, according to these parameters moving objects are classified as vehicles (trucks or cars) or nuisance artifacts. For results visualization, a 3D model is projected onto vehicles in the image plane. Some experimental results using real outdoor sequences of images are shown. These results demonstrate the accuracy of the proposed system under daytime interurban traffic conditions.**

I. Introduction

Detection and tracking of moving vehicles is one of the most challenging problems in computer vision applications. Nowadays, traffic congestion is one of the biggest problems in many cities. For this purpose some researchers have studied how artificial intelligence techniques can be applied to traffic data measurements [1]. Traditionally, road traffic data was collected using different kinds of sensors such as loop detectors or pneumatic sensors. These sensors are limited, since they can only quantify traffic flow or solve specific subproblems [2]. Vision-based cameras are more powerful and sophisticated than loop detectors or pneumatic sensors and also they are easier to install and maintain. In addition, the variety of measurements that can be achieved with cameras are higher than the ones obtained with other sensors for road traffic applications. Vehicles can be tracked and classified precisely and information such as traffic flow, distance between vehicles, vehicles speed and other traffic events like incident detection can be computed and transmitted to drivers using different methods such as radio or GPRS [3].

There are several studies in the literature about automatic vehicle detection and tracking. These kind of systems must work under variable lighting and weather conditions, and also during day and nightime. Cucchiara *et al.*[2] proposed to extract moving vehicles during daytime by means of motion extraction using frame-differencing algorithms and morphological operators, while at nightime vehicles are identified by their headlights.

For counting vehicles, moving objects must be extracted from images. The most common method is known as background subtraction, which is normally a computationally efficient algorithm. Background must be updated in a dynamic way since background in road traffic images is variable. Subtracting this background image from the original image, moving objects can be extracted. However, this method generates erroneous ghosts during the background evolution period, which affects clustering and tracking processes [4]. Optical flow algorithms can be used too, but the computational burden is sometimes overwhelming for real time applications. One challenging problem in these applications is shadows detection, since shadows move along with the moving objects in the image. Shadows are detected as foreground pixels, since the difference with background is significantly. Shadows can cause object merging and object losses, which implies that shadows identification plays a key role in road traffic applications. Prati *et al.* [5] studied different algorithms for shadows detection. Normally shadows detection algorithms use colour information or some probabilistic shadows model. After moving objects segmentation and shadows removal, vehicles are tracked using a Kalman filter in a tracking process [3][6].

This work proposes an effective system for road traffic monitoring under daytime conditions. Moving objects are extracted by means of frame-differencing algorithm and the information from texture unit number. For shadows detection, only grey scale information is used. The effects of shadows are minimized by applying top hat transformations and morphological operators. Finally, moving objects are tracked using a Kalman filter and parameters such as vehicles counting, mean speed and dimensions are measured. The ratio aspect, speed, size and other objects parameters are used to classify each object between vehicles(trucks or cars) and nuisance artifacts (caused by moving vegetation or other artifacts). For results visualization, a 3D scaled prismatic model is projected onto the vehicle in the image plane using camera model equations. In the rest of the paper, the main parts of the algorithm and experimental results are provided.

II. Initialization Setup

The system has an initialization setup. In this setup the user defines an area of analysis in order to detect only closer vehicles to the camera, since working with the whole image is unnecessary waste of computational time. The user can define the road bounds selecting two lines. This process and also the detection of road lines can be done in an automatic way using image processing algorithms [7]. The system has a transient time of 200 frames, in which enough statistical data is computed in order to begin classifying pixels as foreground or background. Figure 1 shows this initialization process and the area of analysis.

Fig. 1.   System's initialization setup

III. Moving Objects Extraction

The extraction of moving objects is carried out by means of a frame-differencing algorithm and texture unit number [8]. During the transitory time (200 frames) some statistics are computed per pixel. These statistics are the mean value and the quadratic mean value of the grey scale intensity $f(x, y)$:

$$\mu_{x,y} = \frac{\sum_{i=1}^{N} f(x,y)}{N} \qquad (1)$$

$$P_{x,y} = \frac{\sum_{i=1}^{N} f(x,y)^2}{N} \qquad (2)$$

Information of the texture unit number difference between two consecutive frames is used. All the different values of texture unit number correspond to the relative gray level relationship between a pixel and its neighbors in all possible directions. For a certain pixel, the expression of the texture unit number is as follows:

$$TUN_{x,y} = \sum_{i=0}^{7} f(x,y)_i \cdot 2^i \qquad (3)$$

In last equation, index i indicates pixel position in the mask of Figure 2: After the transitory time, the system is able to classify each pixel between foreground or background. In order to obtain an adaptative background, a parameter called α is introduced in the system. This parameter has upper and lower bounds:

$$0.25 \leq \alpha \leq 0.6 \qquad (4)$$

By default, α is set to 0.5. If one pixel is classified as a foreground pixel, the parameter α is increased by 0.01,

0	1	2
7	X	3
6	5	4

Fig. 2.   Mask for texture unit number computation

which means that is very probably that this pixel will be classified as a foreground pixel in the next frame. On the contrary, if one pixel is classified as a background pixel, the parameter α is decreased by 0.05, which means that is very probably that this pixel will be classified as a background pixel in the next frame.

One pixel will be classified as a background pixel if the two conditions of equations 5 and 6 are fulfilled. Otherwise, the pixel will be classified as a foreground pixel, and it is statistics are not considered for the background updating.

$$f(x,y) - \mu_{x,y} < K \cdot \alpha \cdot \sqrt{P_{x,y}} \qquad (5)$$

$$\Delta_{TUN} = TUN_{i+1} - TUN_i < T \qquad (6)$$

In the last equations K and T are adaptive thresholds. These thresholds are used to increase or decrease the number of foreground pixels depending on lighting and weather conditions. In Figure 3 some results of moving objects extraction are shown.

IV. Shadows Detection

Shadows are composed by two different parts. The umbra is the darkest part of a shadow, from within the umbra, the source of light is completely blocked by the object causing the shadow. On the contrary, the penumbra is the part of the shadow where the light source is only partially blocked and there is only a partial shadow. Figure 4 shows a typical shadow casted by a car. In Figure 5 isolines of the analyzed shadow are depicted. Two main different areas can be distinguished in the isolines plot, the umbra with very low intensity and the penumbra with higher intensity.
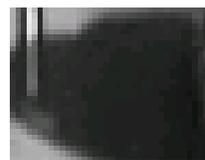
Fig. 4.   A common shadow casted by a car

The top-hat transformation is a powerful operator which permits the detection of contrasted objects on non-uniform background [9]. There are two different types of top-hat transformations: white hat and black hat. The white hat transformation is defined as the residue between the original image and its opening. The black hat transformation is defined as the residue between the

Fig. 3.   Results of moving objects extraction
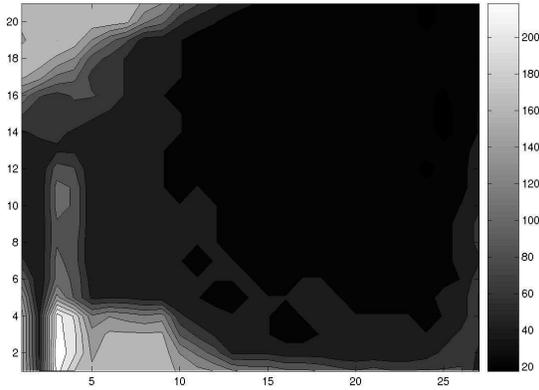
Fig. 5.   Shadows isolines

closing and the original image. The operations white and black hat transformations are defined as follows respectively:

$$\mathbf{WH_T}(\mathbf{x}, \mathbf{y}) = (f - f \circ B) \tag{7}$$

$$\mathbf{BH_T}(\mathbf{x}, \mathbf{y}) = (f \bullet B - f) \tag{8}$$

In the equations 7 and 8 $f(x,y)$ is a grey scale image and B is the structuring element. Both operators, white and black hat, can be used in order to modify the contrast of the image or enhancing contrast in some regions of the image. Normally, in grey scale images, the local contrast is ruled by two kinds of features: bright and dark features. The white hat image contains local peaks of the intensity and the black hat image contains local valleys of the intensity. Combining both hat transformations with the original image, shadows pixels can be detected. Umbra is characterized by dark features whereas penumbra is characterized by local peaks of intensity. The shadows detection algorithm comprises of the following steps:

1) The algorithm starts with the foreground image $f(x,y)$
2) Black hat and top hat transformations of the foreground image are computed using cross kernels

of 3x3 size. Let denote the grey intensity for the black hat transformation $BH(x,y)$ and the white hat transformation $WH(x,y)$

3) Create a new image $r(x,y)$ for storing the results of the Shadows Detection algorithm. This one is a binary image, and is used in the rest of the algorithm
4) For all the (x,y) pixels in the foreground image pixels are classified according to the following decision procedure:

IF $WH(x,y) > T_{WH}$ (Local peaks of intensity that don't belong to the penumbra)
⇒ The pixel value $r(x,y)$ is set to 255

ELSE IF $BH(x,y) > T_{BH}$ (Local valleys of intensity that don't belong to the umbra)
⇒ The pixel value $r(x,y)$ is set to 255

ELSE IF $f(x,y) > T_{BrightObjects}$ (Bright vehicle pixels)
⇒ The pixel value $r(x,y)$ is set to 255

ELSE IF $f(x,y) < T_{DarkObjects}$ (Dark vehicle pixels)
⇒ The pixel value $r(x,y)$ is set to 255

ELSE (Shadows pixels)
⇒ The pixel value $r(x,y)$ is set to 0

5) Apply *closing* operator to $r(x,y)$ image using a 5x5 elliptical kernel
6) Apply *opening* operator to $r(x,y)$ image using a 3x3 cross kernel

In Figure 6 and 7, two examples of shadows detection algorithm results are shown.

V. Perspective Camera Model

In order to estimate different objects parameters, a perspective camera model called pinhole is applied. The equations for computing the image coordinates u and v are shown:

$$u = \frac{F_U \cdot (X + CAMx)}{sen(\alpha) \cdot Y + cos(\alpha) \cdot Z} - u_0$$

$$v = \frac{F_V \cdot (cos(\alpha) \cdot Y - sen(\alpha) \cdot Z + CAMy)}{sen(\alpha) \cdot Y + cos(\alpha) \cdot Z} - v_0 \tag{9}$$

where:

| (a) Foreground image | (b) Black hat image | (c) White hat image | (d) Result image |

Fig. 6. Results of shadows detection: Example 1

| (a) Foreground image | (b) Black hat image | (c) White hat image | (d) Result image |
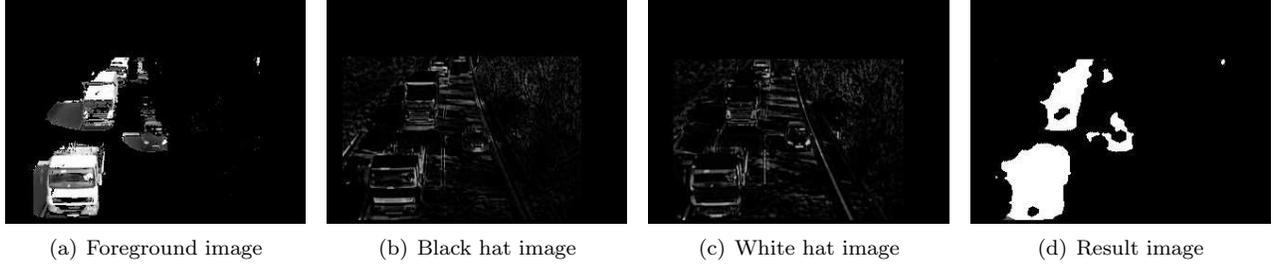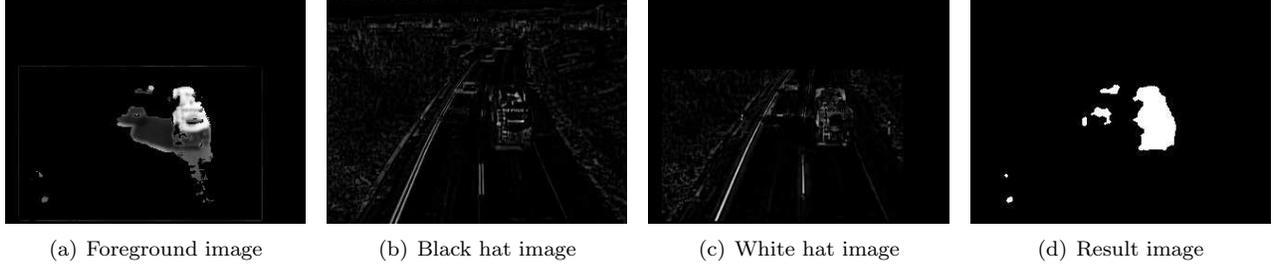
Fig. 7. Results of shadows detection: Example 2

- u: horizontal image coordinate (pixels)
- v: vertical image coordinate (pixels)
- F_u: horizontal focal length (pixels)
- F_v: vertical focal length (pixels)
- α: camera pitch angle (radians)
- u_0: image plane horizontal offset (pixels)
- v_0: image plane vertical offset (pixels)
- $CAMx$: horizontal displacement between real world and image plane center of coordinates (m)
- $CAMy$: vertical displacement between real world and image plane center of coordinates (m)
- X: real world horizontal coordinate (m)
- Y: real world vertical coordinate (m)
- Z: look ahead distance for planar ground (m)

Camera calibration parameters, are computed in a setup process. A vehicle at a look-ahead distance Z from the camera will be projected into the image plane at a vertical and horizontal coordinates (u,v) respectively. The expressions for computing the real world horizontal coordinate X and the look ahead distance Z are shown in 10. Using both equations, vehicle's length and width can be measured.

$$X = \frac{(sen(\alpha) \cdot Y + cos(\alpha) \cdot Z) \cdot (u + u_0) - F_U \cdot CAMx}{F_U}$$

$$Z = \frac{F_V \cdot (cos(\alpha) \cdot Y + CAMy) - Y \cdot (v + v_0) \cdot sen(\alpha)}{(v + v_0) \cdot cos(\alpha) + F_V \cdot sen(\alpha)}$$

$$(10)$$

VI. Tracking

After shadows removal, moving objects are tracked using a Kalman filter, based on the estimated position of object's centroid (coordinates u,v in the image plane). In the tracking process some important parameters of the objects, using perspective camera model equations, are computed. These parameters are mainly the width in meters, the length in meters and speed in km/h. Other parameters such as area and ratio aspect are computed too.

Tracked objects are classified between vehicles (trucks, cars) and nuisance artifacts (mainly caused by moving vegetation). Nuisance artifacts are detected in an easy way by means of speed. If speed of one object is lower than 5 km/h (in absolute value) and the average speed of the rest of the objects is higher, the object is considered as a nuisance artifact. For classifying between different types of vehicles, the ratio between the vertical image coordinate and the area of the object $v/Area$ is computed per object. This ratio is indicative of the relationship between the coordinate v in the image and the area of the object. As long as an object gets closer to the camera (higher v coordinate), the size of the object will be higher. On the contrary, as long as an object moves away from the camera (lower v coordinate), the size of the object will be lower too. Furthermore, two different values of vertical real world coordinate Y are used, depending on the vehicles classfication result.

Real length and width of vehicles (ΔZ and ΔX respectively) can be used for detecting possible occlusions. Occlusions are still a great problem in road traffic applications [7][10]. If one object is found to have length and/or width dimensions greater than an average value $\pm$ a standard deviation for each dimension, the object is considered to be suspicious of being an occlusion of different vehicles. However, occlusions treatment is not studied in depth in this work.

For estimating the speed of vehicles, speed is computed by means of the difference between two consecutive

frames of the look ahead distance Z of the vehicle. This look ahead distance, was computed using the filtered position of vehicle's centroid (u,v) as explained in equation 10. Then, a smoothing process is applied in speed equation as it can be seen in equation 11. In the last equation, the parameter F_R is the capture frame rate, and the constant 3.6 is just for describing speed units in km/h.

$$Speed_i = 0.6 \cdot Speed_{i-1} + 0.4 \cdot (Z_i - Z_{i-1}) \cdot F_R \cdot 3.6 \quad (11)$$

VII. 3D Modelling

A 3D prismatic model is computed per vehicle. Filtered centroid's image coordinates (u,v) are projected into real world coordinates (X,Y,Z) using equation 10. From the centroid's position in real world coordinates, an eight points prismatic model is built considering dimensions of table I. These values represent approximate average distances of analyzed vehicles in a setup process.

TABLE I

3D Model Parameters

Vehicle Type	$\Delta X(m)$	$\Delta Y(m)$	$\Delta Z(m)$
Cars	1.6	0.8	3
Trucks	2.5	1.6	7

VIII. Experimental Results

The proposed method was tested on different real monocular road traffic sequences at daytime. A B&W camera was located in a bridge of N II road (Madrid-Barcelona) at the location of Alcalá de Henares. This road is one of the roads in Madrid with more traffic congestion, and the traffic flow of big trucks is quite important. Vegetation is present in some sequences, which is problematic, since vegetation is also a moving object and cast shadows on the road. The algorithm is implemented with C/C++ on a Pentium IV 3 Ghz platform and the size of recorded image sequences is 320 pixels by 240 pixels per frame. The algorithm runs at 20 fps, which is enough for real time demands. Figure 8 shows some samples of the final result of the algorithm. 3D models for trucks and cars can be observed with different colours (blue and green respectively).

A. Estimated Dimensions of Real World

As it was explained, real length and real width are computed per object. These dimensions are computed projecting image plane points into real world coordinates. In Table II estimated dimensions of cars are shown, considering the mean and the standard deviation of all the processed cars in the sequence. In Table III, results for trucks are shown too. It can be observed, that estimated dimensions are acceptable compared to real dimensions of the vehicles. However, some errors are introduced due to inaccuracy in camera calibration parameters and moving objects segmentation.

TABLE II

Estimated Dimensions of cars

Sequence	# of Cars	$\Delta X(\mu \pm \sigma)(m)$	$\Delta Z(\mu \pm \sigma)(m)$
1	28	1.2728 ± 0.5773	3.7798 ± 0.9993
2	39	1.4960 ± 0.8216	3.6516 ± 1.3926
3	39	1.4623 ± 0.6138	3.4524 ± 1.3156
4	29	1.4101 ± 0.4850	2.0801 ± 1.8629

TABLE III

Estimated Dimensions of Trucks

Sequence	# of Trucks	$\Delta X(\mu \pm \sigma)(m)$	$\Delta Z(\mu \pm \sigma)(m)$
1	14	3.1349 ± 1.3026	10.7660 ± 5.0969
2	23	3.6595 ± 0.8214	12.4139 ± 5.3167
3	19	3.0267 ± 1.2334	10.0448 ± 4.6570
4	10	3.0251 ± 1.2145	10.5768 ± 4.4473

B. Estimation of Vehicles Speed

For estimating the speed of vehicles equation 11 is used. In figure 9 the vehicles speed histogram of a sequence is shown. The speed of all the analyzed vehicles was measured in each frame, and then the histogram was computed. It can be observed, that, in average, the main speed of vehicles is between 90 and 100 km/h. According to real situation, the recommended speed limit for that road section is 90 km/h.
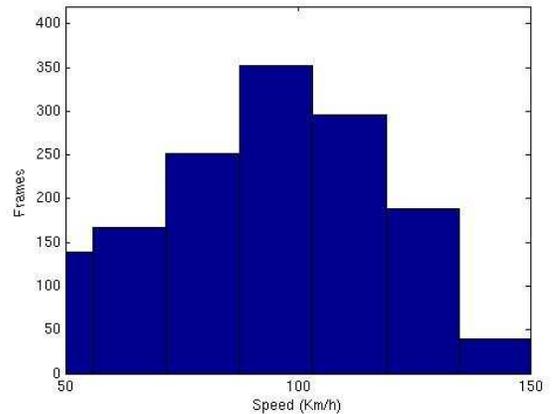

Fig. 9.   Histogram of vehicles speed

C. Estimated Vehicles Trajectories

In Figure 10, an histogram of vehicles moving at different lanes is shown. By means of vehicles trajectories histogram, traffic flow patterns can be studied. Besides, an automatic lane-dividing-line detection algorithm can be computed by means of this histogram [7].

D. Probability of Detection

The probability of detection (P_D) was computed for different video sequences. It can be observed in Table IV that the number of lost vehicles (vehicles that are not detected) is very small. Lost vehicles are not detected mainly due to occlusions problems.
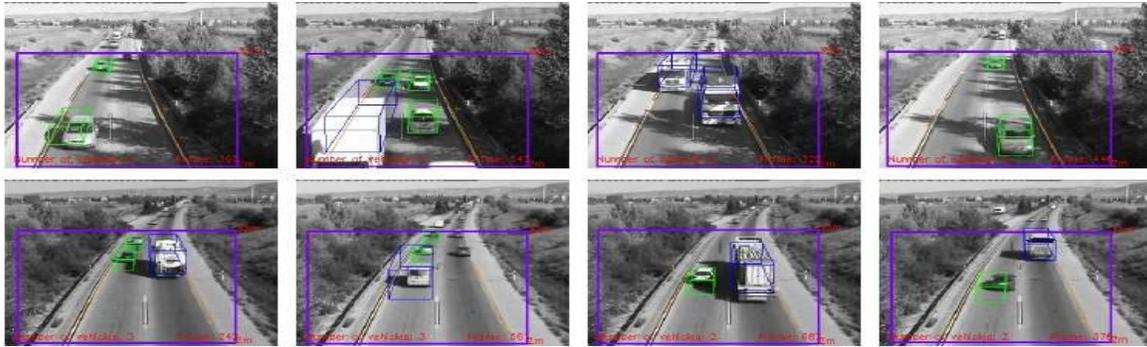
Fig. 8. Vehicles detection in daylight conditions

TABLE IV

DETECTION PROBABILITIES

| Nr. of Sequence	Frame Count	Vehicles Count	# of Detected Vehicles	P_D
1	1324	42	38	90.47 %
2	1483	62	55	88.71 %
3	1350	58	56	96.55 %
4	1303	39	38	97.43 %
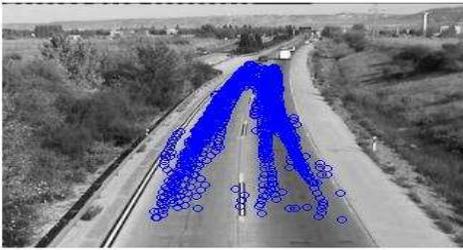
Fig. 10. Histogram of vehicles moving at different lanes

IX. CONCLUSIONS AND FUTURE WORK

This paper proposes a method for accurately counting the number of vehicles on a road at daytime. The most important novelty of the proposed method is the shadows treatment using only intensity of B&W images and top hat transformations. The algorithm works good under difficult road traffic conditions such as shadows, vegetation and big trucks. A good estimation of road traffic density can be achieved with the proposed system, but on the contrary the speed estimation of each vehicle is not completely reliable since it is influenced by errors in the segmentation and tracking processes. Information such as traffic flow patterns, traffic events and vehicles speed are a valuable information both for drivers and city planners in order to increase road safety and develop new capabilities for intelligent vehicles.

Although results are encouraging, several improvements must be included into the system. At the moment, we are working on the nighttime vehicles detection algorithm and studying different methods for occlusions treatment.

REFERENCES

[1] Remy Chang, Tarak Gandhi and Mohan M. Trivedi, "Computer vision for multi-sensory structural health monitoring system," *IEEE International Transportation Systems Conference*, October 2004.

[2] Rita Cucchiara, Massimo Piccardi and Paola Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Transactions on Intelligent Transportation Systems*, June 2000.

[3] Erhan Bas, A. Murat Tekalp, F. Sibel Salman, "Automatic vehicle counting from video for traffic flow analysis," *IEEE Transactions on Intelligent Transportation Systems*, June 2007.

[4] Y. Liu and P. Payeur, "Vision-based detection of activity for traffic control," *Canadian Conference on Electrical and Computer Engineering*, 2003.

[5] Andrea Prati, Ivana Mikic, Mohan M. Trivedi and Rita Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 2003.

[6] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC, Tech. Rep., 2001.

[7] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen and Wen-Fong Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Transactions on Intelligent Transportation Systems*, June 2006.

[8] S. Karkanis, K. Galousi and D. Maroulis, "Classification of endoscopic images based on texture spectrum," *ECCAI Advanced Course on Artificial Intelligence*, 1999.

[9] L. D. Ye Derong, Zhao Yuanyuan, "Fast computation of multiscale morphological operations for local contrast enhancement," in *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005.

[10] Clement Chun Cheong Pang, William Wai Leung Lam and Nelson Hon Ching Yung, "A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images," *IEEE Transactions on Intelligent Transportation Systems*, September 2007.