Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 88 (2015) 35867-35872

ISSN: 2229-712X

Traffic density estimation, vehicle classification and stopped vehicle detection for traffic surveillance system using predefined traffic videos

P. Rajesh[1], M. Kalaiselvi Geetha[2] and R.Ramu[3]

[1]Mathematics Section, Faculty of Engineering and Technology, Annamalai University, Annamalainagar-608002,

[2]Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Annamalainagar-608002,

[3]Department of Computer Science, Thiru A. Govindasamy Government Arts College, Tindivanam- 604 002,

ARTICLE INFO

Article history:

Received: 3 August 2012;

Received in revised form: 15 October 2015;

Accepted: 31 October 2015;

Keywords

Vehicle Identification,
Motion Detection,
Traffic Density Estimation and Stopped Vehicle Detection.

ABSTRACT

In this paper we present vehicle density estimation, vehicle classification and stopped vehicle detection system for outdoor traffic surveillance is presented. It is important to know the road traffic density in predefined traffic videos especially in mega cities for signal control and effective traffic management. In recent years, video monitoring and surveillance systems have been widely used in traffic surveillance system. Hence, traffic density estimation and vehicle classification can be achieved using video monitoring systems. In vehicle detection methods for several review of literature, only the detection of vehicles in frames of the given video. The stopped vehicle detection is based on the pixel history. This methodology has proved to be quite robust in terms of different weather conditions, lighting and image quality. Some experiments carried out on some highway scenarios demonstrate the robustness of the proposed solution.

Introduction

The traffic video monitoring and surveillance systems have been widely used in traffic management. Most of the companies have started to use several cameras for the use of traffic surveillance system. The surveillance system extracting useful information such as traffic density, vehicle types from these camera systems has become a hassle due to the high number of cameras in use. Manual analysis of these camera systems is now unapplicable. Development of intellegent systems that extract traffic density and vehicle classification information from traffic surveillance systems is crucial in traffic management.

It is important to know the traffic density of the roads real time especially in mega cities for signal control and effective traffic management. Time estimation of reaching from one location to another and recommendation of different route alternatives using real time traffic density information are very valuable for mega city residents. In addition, vehicle classification (big: truck, middle: van, or small: car) is also important for traffic control centers. For example, the effects of banning big vehicles from a road can be analyzed using vehicle classification information in a simulation program. This paper presents an automatic traffic density estimation and vehicle classification method for traffic surveillence system using neural networks.

Several other vehicle detectors such as loop, radar, infrared, ultrasonic, and microwave detectors exist in the literature. These sensors are expensive with limited capacity and involve installation, maintenance, and implementation difficulties. For example, loop sensor might need maintenance due to road ground deformation or metal barrier near the road might prevent effective detection using radar sensors [1]. In resent years, video processing techniques have attracted researchers for vehicle detection [2-7].

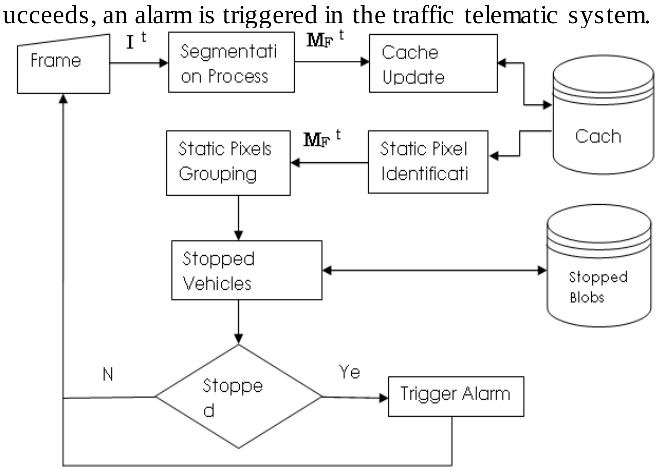
Detection of moving objects including vehicle, human, etc. in video can be achieved in three main approaches: Temporal difference, optical flow, and background substraction. In temporal difference, the image difference of two consecutive image frames are obtained [12-18]. However, this approach has some limitations such as visual homogeneity requirement and its effectiveness depends on the speeds of moving objects [2]. Optical flow method was developed to obtain effective background modification, which bases on the detection of intensity changes [2]. However, illumunation change due to weather or sun-light reflections decreases its effectiveness. It is also computationally inefficient [2]. The third method, background subtraction, is the mostly seen method in the literature for effective motion tracking and moving object identification [2, 4, 6, 9, 10, 11]. In background subtraction, background can be static, in which a fixed background is obtained beforehand and used in the entire process; or dynamic, in which background is dynamically updated with changing external effects like weather. Static background may not be effective in most applications, many methods include dynamic background subtraction. In [19], the background is detected dynamically by using dynamic threshold selection method. In [22], land mark based method and BS&Edge method are used to remove the shadow from the scene.

Different classification techniques have been employed after the moving objects are detected in order to identify the

Tele:
E-mail addresses: p_rajesh_1981@yahoo.com

moving object. In [4], support vector machines is used to idenfity if the detected moving object is a vehicle or not. Support vector machine is a two class classification method and requires modification for multi class classification. The vehicles are detected using mathematical modeling in [21]. The expected parameters of a moving vehicle is matematically modeled using the position of the camera, vechile, and sun; it is compared with the values obtained from the video. However, this model requires very sensitive calibration of the camera and it works for cases with short distance between camera and vehicles. The traffic videos used in Istanbul do not satisfy these needs. In [20], rule based reasoning is used for vehicle detection, in which the results highly depend on the rules decided by humans.

A vehicle stopped on the road or on the hard shoulder can represent a serious threat. An immediate detection of a stopped vehicle could help prevent serious accidents by warning the oncoming vehicles and highway assistance services or, ultimately, by warning the police. This paper focus on the detection of vehicles stopped on highways. For that purpose a segmentation process was used to identify vehicles in the image. The result of the segmentation process was used as input in a stopped vehicles detection system. This stopped vehicles detection system has three main phases. Firstly, it is verified if there are any static pixels segmented during a certain period of time. Secondly, those static pixels are then grouped into blobs. The static pixels identification is based on a pixel history cache analysis. Finally, a blob temporal validation is applied to discard false positives in traffic jam situations. If the validation succeeds, an alarm is triggered in the traffic telematic system.



(a)

In most vehicle detection methods in the literature, only the detection of vehicles in frames of the given video is emphesized. However, further analysis is needed in order to obtain the useful information for traffic management such as real time intensity of roads and number of vehicle types passing these roads. This paper presents vehicle classification and road intensity calculation methods using neural networks. Section 2 discusses the modeling background, section 3 gives the traffic density estimation algorithm and results obtained from traffic videos. The Section 4 gives the stopped vehicle detection in traffic surveillance system and finally Section 5 concludes the paper.

Modeling Background

Our model consists of three submodels: Moving Object Detector (MOD), Vehicle Identifier (VI), Traffic Density Calculator (TDC) as shown in Fig.1. In MOD, moving objects are detected using background estimation method. In VI, the vehicles are detected and identified as small, middle, and big using neural network model. In TDC, the traffic density is calculated using the identified vehicle information in successive frames. The details of these submodels are explained in following subsections.

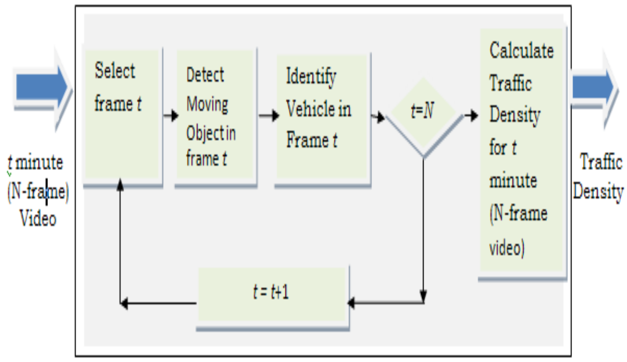


Figure 1: Model Flow

Moving Object Detector (MOD)

Moving object detection is applied to each frame distinctly and performed as shown in Fig. 3. In the first step, the background of the video is subtracted from the current image and the threshold is applied to the difference matrix. Then, the moving object is detected by analyzing the difference matrix between background and the current frame image. Lastly, the background is updated with the current frame for the following frames.

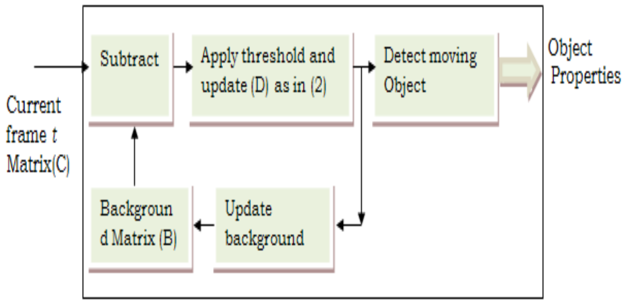


Figure 2: Moving Object Detector Flow

In the first step, the background is subtracted from the current frame. The difference matrix is applied to a threshold. The gray levels greater and lower than the threshold is updated as 1 and 0, respectively, which leads moving objects to be represented as white pixels as shown in Fig. 3.b and formulated as in (1) and (2).

$D_{i,j} = C_{i,j} - B_{i,j}$

D : Difference matrix with n rows and m colums

C : Current frame matrix with n rows and m colums

B : Background matrix with n rows and m colums

$$D_{i,j} = \begin{cases} 1 \text{ if } D_{i,j} > th \\ 0 \text{ if } D_{i,j} \le th \end{cases}$$

Once the binary matrix showing the difference between current frame and background is obtained, this matrix is analyzed in order to detect the moving object. The biggest area of ones within this matrix is detected. For example there are three objects represented with ones in the matrix given in Fig.4. In order to classify the moving objects, as much information as possible should be extracted. For this purpose, smallest ellipse and rectangle that covers the given object is considered and 14 properties are identified for each object. These properties are given in Fig. 5.

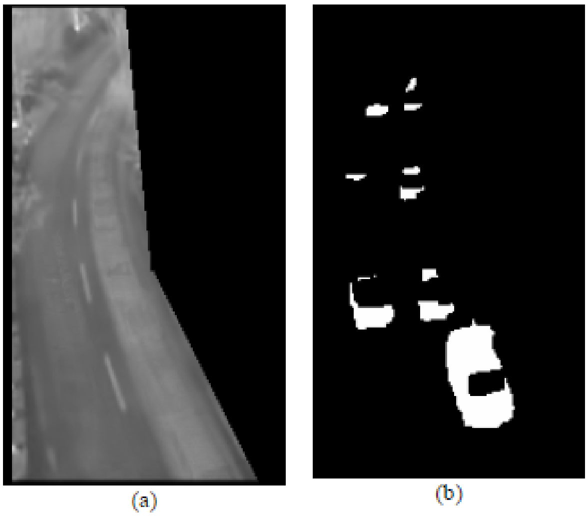


Figure 3: a) Dynamic background b) Difference between the current frame and background

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**
0	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**
0	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**
0	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**1**
0	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**

Figure 4. Example of Difference Matrix

As mentioned earlier, the static background (e.g., the image of the road without any vehicle is passing on the road) may be used. However, this approach works only for cases where external factors such as weather, illumnuation (day, night or time within a day) effects are the same with the static background. Thus, dynamic background calculation methods are common in the literature. In the dynamic background calculation, the background is updated for each frame. In our algorithm, the pixel gray levels are stored as time series data and the median value of the time series data is updated based upon the new gray level in the current frame.

Vehicle Identificator (VI)

In this work feed-forward Neural Network is used. Neural Networks (NN) have been used for decades for solving problems such as classification, clustering, and function approximation. NNs inspired from biological brains consisting of millions of interconnected neurons. NN calculates an output by processing the inputs through neurons in input, hidden, and output layers. There exist several types of NN methods.

Our Neural Network model consists of 14 input and 4 output layers. Input layers are the 14 object properties identified in MOD and shown in Fig. 5. Output layers are binary valued nodes and each represents a vehicle type (i.e., big, small, and medium vehicle and not a vehicle). Only one output node can be one at a time. In other words, these output nodes compete each other in order to represent the given input. Only one hidden layer is used, which includes 25 nodes. It is optimized using trial-error method. Fig. 6 represents the structure of the neural network model.

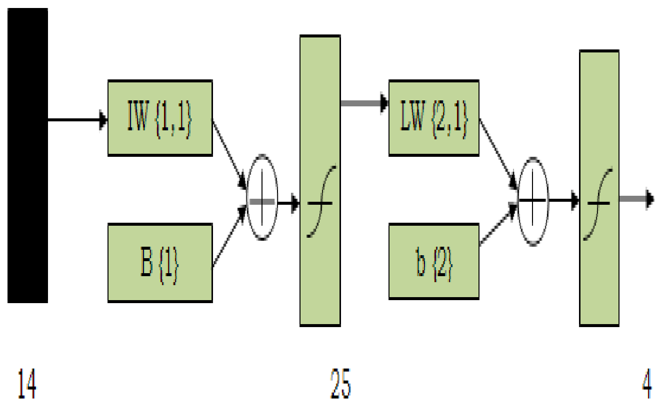


Figure 6 Neural Network Model

Figure 5: Properties of classify a vehicle

1	P1n	Number of pixels covered for the objects. Number of ones and zeros surrounded with these ones
2	P2n	Number of covering ones of the objects
3,4	cx,cy	Center of the area
5,6,7,8	E1,E2,E3,E4	The coordinates of four edges of the smallest rectangle that can cover the object
9	dx	The distance between smallest indices on x-axis
10	dy	The distance between smallest indices on y-axis
11	O	Angle between the object axis with the x-axis
12	R	Ratio of number of ones to the number of pixels within the smallest rectangle that cover the object
13	Ec	Eccentricity of the smallest ellipse that cover the object
14	di	Diameter of the smallest ellipse that cover the object

```
Vi is calculated based on
AllVech = 0
for (k=1 to N) //N is the last frame
for (j=1 to Detk) //Detk is the number vehicles detected in frame k
AllVech=AllVech+1
Center[AllVech]=c[ j ]
Vi = 0
isclose = false:
while (is AllVech empty)
  select vehicle from AllVech
  [x , y]=get the coordinates of selected vehicle
  remove the selected vehicle from AllVech
  for (go back 1 to H number of frames)
    [tempx, tempy]=get the coordinates of vehicles in each frame
   if (is tempx and tempy close enough to x and y)
     isclose = true
  if (not isclose)
    isclose = false
    Vi = Vi+1
Store the selected vehicle for future comparisons
```

Figure 7 : Vehicle counting algorithm

Traffic Density Calculator (TDC)

In the first two submodels (MOD and VI), the frames are processed individually. In this submodel, all video frames for a given time period is processed together in order to calculate the number of vehicles that passed through the road for the given time period. Succusive frames represent the scene of the road miliseconds after each other. Thus, the same vehicles will be seen in successive frames. In this submodel, number of vehicles that passed the road for the given time period is counted using the location of the vehicle in successive frames. The traffic density is calculated as the number of vehicles over the time as shown in (3). The number of vehicles is calculated according to the algorithm given in Fig. 7.

$$Tensity_i = \frac{V_i}{t}$$

Densit y_i : Traffic density of vehicle type i

V_i: Number of vehicle type i that passed the road in time period T

T : Time period

Model application to real videos

The presented method is applied to video obtained from one of the traffic cameras used Istanbul traffic management company. One scene of the video is given in Fig. 8. In the selected video, there are three road parallel to each other. Each road consists of two lanes. The left road, which is the most crowded one, is selected in our application. Our model is applied to 1000 frames, which lasts 100 seconds (10 frames in a second).

Figure 8 : One scene of the traffic video used in model

Since we are interested in only one road, the rest of the matrix is cut out and the background is calculated only for the selected road as in Fig. 9.

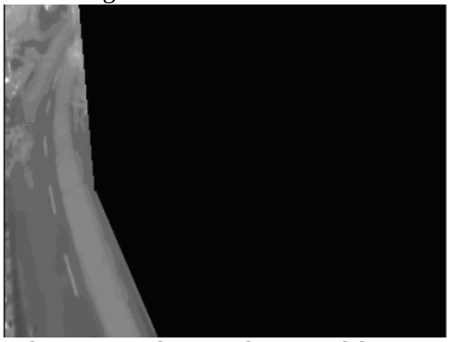


Figure 9 : The Background image

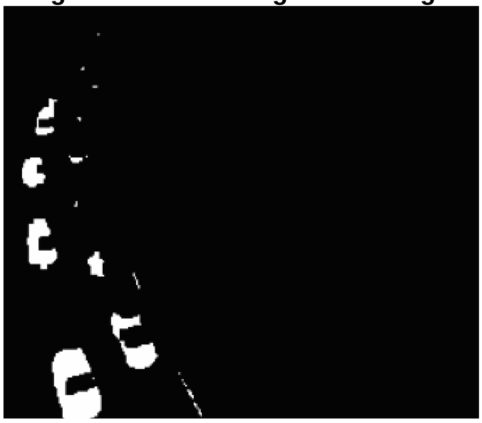


Figure 10 : The difference matrix

The difference matrix is used to identify the vehicle nominees. 14 properties shown in Fig. 5. is used as input for NN model. NN model identifies the vehicles with classification accuracy of 98.9878. After the vehicles in frames are identified, traffic intensity is calculated. In 100 seconds, 68 vehicles has passed from the road. Number of vehicles counted by our algorithm is also 68. In other words, traffic intensity is calculated without any error. However, the classification of vehicle types is performed with some error. Some of the vehicles have been misclassified. The results are given in Table 1. The video is uploaded to the our website at

Table 1: Classification Results

	Total Vehicles	Small (Cars)	Medium (Van)	Big (Bus)	Unclassified
Real	68	58	9	1	0
Found	68	48	15	1	4
Correctly Identified	68	64			4
Accuracy	100%	94%			6%

Robust Vehicle Segmentation

The core of an incident detection system has to be an accurate and robust segmentation process.

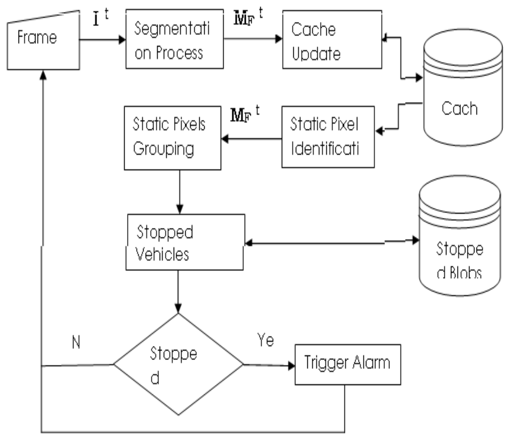


Figure 11. Overview of the stopped vehicles detection process here proposed.

This system is based on background subtraction, and uses three backgrounds to model the background variations. Two distinct thresholds are used: a per-pixel threshold for a robust adaptation to the scene variations, and a global threshold, that is the minimum value, to eliminate camera noise scenario features. It is also used a shadow/highlight detection algorithm based on cross-correlation to discard the blobs generated by lighting variation. This segmentation process proved to be a good basis for an incident detection system.

Stopped Vehicles Detection

The main assumption for the stopped vehicles detection method is that some foreground pixels with the same color that appears during a large period, are probably part of a stopped vehicle. After a blob has been formed by these pixels grouping, it will only be considered as stopped if it holds approximately the same position and dimensions for a predefined validation time (see Fig. 11).

Stopped Pixels Recognition

The stopped pixels are identified by the analysis of the segmented pixels. A pixel is classified as *static* if it is labeled as foreground with the same color with a certain frequency. A pixel history cache [27] is used to analyze the pixel color frequency.

For each pixel it is analyzed a set of colors that appears at least one time in the last *Th* frames. Cache is an array of the same size of the image. Each entry corresponds to a pixel and it has a list of Codewords (called Codebook). A Codeword saves the RGB color components and a validation buffer that keeps occurrence history of a color in the pixel in the last ® frames. Occlusions, vibrations or lighting changes that can temporarily hide or change a vehicle's pixel color are taken into account with the pixel history cache.

For each foreground pixel in the present frame it is checked if the color matches any Codeword present in the Codebook. A match event exists if the euclidean distance between two colors, in RGB space, is below ² = 30. If there is a match, with a Codeword in Cache, RGB components are updated by a weighted average, and the validation buffer is updated with '1'. For all the other Codewords present in the Codebook the validation buffer is updated with '0'. If there is no Codeword matching, a new Codeword is created in the pixel Codebook. The validation buffer of the Codewords of non segmented pixels are also updated with '0'. Finally, all the Cache Codewords that not appear at least one time in the last *Th* frames are deleted (*Th* = 25). A pixel is validated as static if exists a Codeword on its Codebook with ¯ occurrences in the last ® frames, where ® is the validation buffer size (namely ¯ = 40 and ® = 64). Thisstrategy handles the occlusion problem mentioned above. Fig. 2 b) shows chromatically the number of occurrences of the most frequent Codeword.

Stopped Vehicle Validation

Once identified the static pixels, they are grouped into a blob. The validation process of a possible stopped vehicle starts if the blob has a significant area. Not all the blobs that result from static pixels grouping are really stopped vehicles. They can be vehicles moving slowly in the image or wrong segmented regions. The main idea of the validation process is to track the blob in validation and verify if it maintains the same position with the same dimensions during a certain validation period. In this situation, the blob is considered as a stopped vehicle, and an alarm is triggered.

Experimental Results for stopped vehicle detection

The system was tested with a real set of image sequences from highways traffic surveillance cameras with different weather conditions, lighting, image quality and fields of view. The system was also tested in real time in some Portuguese high traffic density highway scenarios and in a carpark at Coimbra University. Table 2 shows the experimental results obtained in some different scenarios. In Fig. 12, it is illustrated the stopped vehicles detection. In the experiments performed to the system it was verified that the mis-detection of stopped vehicles was due to the lack of image contrast. Most of the false positives are related to

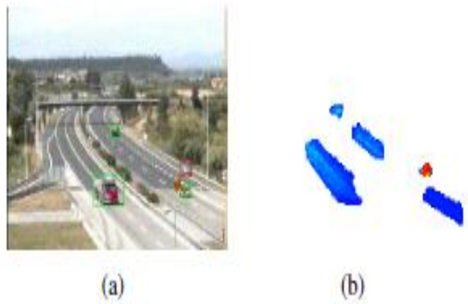


Figure 12. Stopped vehicles detection system experiments
a) stopped vehicle detection (red box);
b) chromatic representation of the number of occurrences of the most frequent Codeword. a wrong segmentation result.

Table 2. Experiments conducted on the stopped vehicle detection system.

	Period	Stopped	Hits	FP
Tunnel	8 minutes	4	4	1
Car-Park	24 hours	85	82	18
Highway 1	24 hours	0	0	7
Highway 2	24 hours	2	2	4

Conclusion

Automatic traffic density estimation and vehicle classification through video processing and artificial systems are important for traffic management companies especially in mega cities. Traditional traffic density estimation methods such as radars, loop sensors, ultrasonic waves etc. have some limitations. A stopped vehicles detection system based on a pixel history cache was presented. The experiments conducted on a large number of scenes demonstrate that this system is able to robustly detect stopped vehicles under different weather conditions, lighting, image quality and image compression variation. In this paper, automatic traffic density estimation, vehicle classification method and stopped vehicle detection is presented.

References

[1] M. Alyuruk, Information Technologies Chief at Istanbul Traffic Management Company (ISBAK), Inteview on June 2008

[2] X. Ji, Z. Wei, Y. Feng, Effective vehicle detection techniques for traffic surveillence systems, J. Vis. Commun. Image R. 17, 647-658, 2006
[3] A. Lozano, G. Manfredi, L. Nieddu, An algorithm for the recognition of levels of congestion in road traffic problems, Math. Comput. Simul., doi:10.1016/j.matcom.2007.06.08
[4] J. Zhou, D. Gao, D. Zhang, Moving vehicle detection for automatic traffic monitoring, IEEE Trans. Vehic. Tech., 56(1), 51-59, 2007
[5] X. Niu, A semi-automatic framework for highway extraction and vehicle detection based on a geometric deformable model, ISPRS J. Photog. Remot. Sens. 61 170-186, 2006
[6] W. Zhang, X. Z. Fang, X. Yang, Moving vehicles segmentation based on Bayesian framework fr Gaussian motion model, Patt. Recog. Let. 27, 956-967,2006
[7] X. Li, Z. Q. Liu, K. M. Leung, Detection of vehicles from traffic scenes usingfuzzy integrals, Patt. Recog. 35, 967-980, 2002
[8] V. Kastrinaki, M. Zervakis, K. Kalaitzakis, A survey of video processingtechniques for traffic applications, Image Vis. Comput. 21, 359–381, 2003
[9] J. B. Kim, H. J. Kim, Efficient region-based motion segmentation for a video monitoring systems. Pattern Recognit. Lett. 24, 113-128, 2003
[10] J. K. Cai Q.i Aggarwal, Tracking human motion in structured environments using a distributed camera system, IEEE Trans. Pattern Anal. Mach. Intell. (12), 1241-1247, 1999
[11] D. Murray, A. Basu, Motion tracking with tracking with an active camera, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5), 449-459, 1994
[12] M. T.Lo´pez, Ferna´ndez-Caballero, A., Ferna´ndez, M. A., Mira, J., & Delgado, A. E. (in press). Dynamic visual attention model in image sequences. Image and Vision Computing.
[13] M. T Lo´pez, Ferna´ndez-Caballero, A., Ferna´ndez, M. A., Mira, J., & Delgado, A. E. Visual surveillance by dynamic visual attention method. Pattern Recognition, 39(11), 2194–2211, 2006
[14] M. T. Lo´pez, Ferna´ndez-Caballero, A., Mira, J., Delgado, A. E., & Ferna´ndez, M. A. Algorithmic lateral inhibition method in dynamic and selective visual attention task: Application to moving objects detection and labelling. Expert Systems with Applications, 31(3), 570–594, 2006
[15] M. Dubuisson, & A. Jain, Contour extraction of moving objects in complex outdoor scenes. International Journal of Computer Vision, 14, 83–105, 1995
[16] S. Gil& T. Punt, Non-linear multiresolution relaxation for alerting. In Proceedings of the European conference on circuit theory and design, 1639–1644, 1993
[17] M. Teal, & T. Ellis, Spatial-temporal reasoning on object motion. In Proceedings of the British machine vision conference, 465–474, BMVA Press, 1996
[18] R. Waterfall, & K. Dickinson, Image processing applied to traffic. Traffic Engineering Control, 25, 60–67, 1984
[19] M. Fathy and M. Y. Siyal, A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters, IEEE Trans. Veh. Technol., 47, (4),1342–1349, 1998.
[20] R. Cucchiara, M. Piccardi, and P. Mello, Image analysis and rule-based reasoning for a traffic monitoring system, IEEE Trans. Intell. Transp. Syst., 1,(2), 119–130,2000
[21] Z. Zhu and G. Xu, VISATRAM: A real-time vision system for automatic traffic monitoring, Image Vis. Comput., 18(10), 781– 794, 2000
[22] M. Yu and Y. D. Kim, Vision based vehicle detection and traffic parameter extraction," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., E84A,(6),1461–1470, 2001.
[23] Nadhir Messai, Philippe Thomas, Dimitri Lefebvre and Abdellah El Moudni, Neural networks for local monitoring of traffic magnetic sensors Control Engineering Practice, Volume 13, Issue 1, January 2005, Pages 67-80
[24] Srinivasan, D., Choy, M.C. and Cheu, R.L., Neural Networks for Real-Time Traffic Signal Control,ITS(7), No. 3, September 2006, pp. 261-272.
[25] Baher Abdulhai, Himanshu Porwal and Will Recker Short Term Freeway Traffic Flow Prediction Using Genetically-Optimized Time-Delay-Based Neural Networks
[26] J. Batista, P. Peixoto, C. Fernandes, and M. Ribeiro. A dualstage robust vehicle detection and tracking for real-time traffic monitoring. In *IEEE Int. Conference on Intelligent Transportation Systems*, 2006.
[27] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. In *Real-time Imaging*, pages 167–256, 2005.
[28] G. Monteiro, J. Marcos, M. Ribeiro, and J. Batista. Robust segmentation for outdoor traffic surveillance. In *IEEE ICIP 08*, 2008.
[29] G. Monteiro, M. Ribeiro, J. Marcos, and J. Batista. Wrong way drivers detection based on optical flow. In *IEEE ICIP 07*, 2007.