Enhanced and effective parallel optical flow method for vehicle detection and tracking

Conference Paper · May 2015

DOI: 10.1109/ISMSC.2015.7594042

3 authors:

Prem Kumar Bhaskar

Suet Peng Yong
Universiti Teknologi PETRONAS

Tang Jung Low
Universiti Teknologi PETRONAS

Enhanced and Effective Parallel Optical Flow Method for Vehicle Detection and Tracking

Prem Kumar Bhaskar[1], Suet-Peng Yong[2], Low Tang Jung[3]
Department of Computers and Information Sciences
Universiti Teknologi PETRONAS
Seri Iskandar, 32610, Perak, Malaysia
Email: [1]pkumar_g01837@utp.edu.my, {[2]yongsuetpeng, [3]lowtanjung }@petronas.com.my

Abstract— **In the area of traffic flow monitoring, planning and controlling, a video based traffic detection and tracking plays an effective and significant role where effective traffic management and safety is the main concern. The goal of the project is to recognize moving vehicles and track them throughout their life spans. In this paper, we discuss and address the issue of detecting vehicle/traffic data from video frames with increased real time video processing. Although various researches have been done in this area and many methods have been implemented, still this area has room for improvements. With a view to do improvements, it is proposed to develop an unique algorithm for vehicle data recognition and tracking using Parallel Optical Flow method based on Lucas-Kanade algorithm. Here, Motion detection is determined by temporal differencing and template matching is done only on the locations as guided by the motion detection stage to provide a robust target-tracking method. The foreground optical flow detector detects the object and a binary computation is done to define rectangular regions around every detected object. To detect the moving object correctly and to remove the noise some morphological operations have been applied. Then the final counting is done by tracking the detected objects and their regions in a real time sequence. Results show no false object recognition in some tested frames, perfect tracking for the detected images and 98% tracked rate on the real video with an enhanced real time video processing.**

Keywords-Image Processing; Optical Flow Method; Vehicle Detection; Vehicle Tracking; Vehicle Counting

I. INTRODUCTION

Vehicle information system and intelligent traffic system widely uses automatic recognition of vehicle data to speed up the processing. It has acquired more attention of researchers from the last decade with the advancement of digital imaging technology and computational capacity. Automatic vehicle detection systems are keys to road traffic control nowadays; some applications of these systems are traffic response system, traffic signal controller, lane departure warning system, automatic vehicle accident detection and automatic traffic density estimation [1, 13].

An Automatic vehicle counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. It is just the ability of automatically extract and recognize the traffic data e.g. total number of vehicles. Counting vehicles gives us the information needed to obtain a basic understanding over the flow of traffic in any region under surveillance. So, the first data we have tried to gather is counting of vehicles from available traffic videos from various libraries. In each video frame, optical flow detector differentiates objects in motion from the background by tracking detected objects inside a specific region of the frame and then counting is carried out.

The goal of this current research is to develop an automatic vehicle counting system, which can process videos recorded from stationary cameras over roads e.g. CCTV cameras installed near traffic intersections/ junctions and counting the number of vehicles passing a spot in a particular time for further collection of vehicle/traffic data. A simple approach was carried out to tackle the problem by using Lukas-Kanade algorithm based parallel flow detection, a non-predictive regional tracking and a counting of tracked objects based on simple rules.

The remaining part of this paper is organized in various sections. Section II describes about the recent related works done in this area. Based on that related works we propose video-based vehicle detection and tracking algorithm for traffic data retrieval. Section III describes the image processing based method of detection and tracking of vehicles. Section IV shows the experimental results done on traffic videos. Finally, conclusions are drawn in section V.

II. RELATED WORK

In the recent works, various approaches have been applied in this particular area of detecting vehicle data but still the gap is there as it needs improvement in detection and tracking for accurate prediction with faster processing speed. As we know that real time processing of traffic videos gives accurate and fast results so the data can be used for various purposes quickly and easily. Lee et al. [3] proposed a method using optical flow that can detect vehicle light sources while they are in close range of the camera. The proposed algorithm performs better on the removal of non-vehicle lights sources from the image but method performance may decrease due to the effects of non-vehicle light sources, such as street lights or signals. Da Li et al. [4] presented the well discussed method of foreground and background subtraction [20] using virtual detector and blob tracking through a combination of Otsu's thresholding method and moving cast shadow detection method. However, it is not clear that how it helped in real-time processing of videos within a specific time-frame. Chen et al. used the Time-Spatial Images (TSI) from input video, removing the shadow portions in TSI through the use of Support Vector Machine (SVM) and Deterministic Non-Model based approach, detecting the Region of Interest (ROI) through a simple morphological process and finally using the ROI accumulative curve method to perform vehicle classification and counting. The proposed method is feasible but it is not accurate in small vehicle detection due to traffic conditions [5].

Wei Wang et al. presented a method of multi-vehicle tracking and counting using a fisheye camera that integrates the low level feature point tracking and the higher level affinity based association where the algorithm satisfies the approach but the average processing time is around 750ms which shows that it's not easy to achieve the real-time processing [6]. Mithun et al. [11] applied the technique of virtual line based detector which mainly uses multiple time-spatial images (TSIs), each one obtained from a multiple virtual detector (MVDL) line on the frames of a vehicle video. MVDL-based method may be highly effective in intelligent transportation system but accuracy may or may not be satisfactory in complex traffic situations. Daniel et al. [7] proposed a new scheme of detecting the overtaking vehicles using 1D optical flow algorithm which has been implemented and tested in real-time environment but the position estimation of vehicles against the ground truth needs attention without which the expected results can be achieved easily. Lin et al.[8] applied the technique of detecting possible vehicles in the specified blind-spot area by integrating the appearance-based features and edge-based features but the results are slightly unsatisfactory due to the complex background. Feed-forward neural network has been used to identify the vehicles by P. Rajesh [16] for solving problems such as classification, clustering, and function approximation but it needs clear video input to stop mis-detection of vehicles.

Rohit Sharma et al. [22] presented optical flow and Hough transform based approach for lane departure warning system where optical flow and lane detection both has been used but only one method is activated at any given time. Huang et al. [17] presented a feature-based method of vehicle analysis and counting for bi-directional roads in a real-time traffic surveillance system but it is not clear that how much it is perfect in the scenarios of increased traffic volume. Hashmi et al. [2] proposed a different approach based on statistical parameters to determine the traffic situation at heavily crowded junctions in urban areas and this method need optimization in parameters i.e. color, shape, size and classification of vehicles [21]. Liu et al. [24] presented a vehicle tracking method using conditional random field, which is a feasible method but if the difference between the background and the boundary of the vehicle is small, the segmentation will contain errors in the boundary, resulting from inaccurate optical flow and low image resolution. Nandyall and Patil [18] used automatic vehicle detection and classification based on pair wise geometrical histogram and edge features to represent the model of vehicle type. Then these features are trained with neural network which works fine but counting of vehicles is dependent on threshold value and may not be accurate in heavy traffic. A vision based detection and attribute-based search of vehicles in dense traffic monitoring has been presented by Feris et al. [9] using multiple detectors and can be extended to large scale adaptation. Method is effective in vehicle monitoring but the condition of heavy traffic flow is not clear. Kota and Rao [19] proposed the frame difference method to detect the moving regions with different time instances to classify and count the vehicles. However, the performance of this system is significantly affected by the selected thresholds. Huang et al. [12] proposed moving object detection algorithm from video for localization of vehicle by differentiating current image and background image and applying connectivity and relabeling technique to count vehicles. Although the approach has filtered background noise from video using opening operation, still it has some noise clustering which cannot be filtered easily. Zhao and Wang [15] have proposed a new approach to count vehicles in complex traffic scenarios by utilizing the information from semantic regions and counting vehicles on each path separately. The approach has some limitations as a semantic region could be detected if pedestrians frequently walk through a zebra crossing causing difficulty on trajectory clustering.

Bouvie et al. [14] presented an alternative using particle motion information but interrupted traffic flow and occlusion may downgrade the results. Very small vehicles can be missed, since the number of particles may be insufficient to generate a cluster. Siang Teoh and Thomas Bräunl [10] proposed a mechanism for vehicle tracking and controlling in consecutive video frames based on Kalman filter and a reliability point system. The most probable location of a detected vehicle in the subsequent video frame is predicted by Kalman filter and this data is used by the tracking function to narrow down the search area for re-detecting a vehicle. It also helps to smooth out the irregularities due to the measurement error. To monitor the quality of tracking for the vehicles in the tracking list, this system uses reliability points. Each vehicle is assigned with a reliability point, which can be increased or decreased at every tracking cycle depending on how consistent the vehicle is being re-detected. Automatic vehicle detection and counting algorithm proposed by Jang et al. [23] uses Gaussian mixture model (GMM), object histogram and Lucas-Kanade method to count number of vehicles passing through the scene. The proposed method has one limitation that the accuracy was reduced at a crowded driving or low-speed driving.

III. Methodology

The proposed automatic vehicle counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. In each frame, parallel optical flow detector differentiates the object in motion from the background by tracking detected objects inside a specific region and then counting is carried out in a real time sequence. Most of the optical flow algorithms are based on frequency information, correlation and gradient respectively such as Lucas-Kanade, Block Matching and Horn-Schunck, Buxton–Buxton method, Black–Jepson method, General variation methods. Lucas and Kanade method is consistent in producing accurate depth maps with good noise tolerance. Hence we will delve into Lucas-Kanade algorithm.

Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second. Optical flow algorithms aim to estimate this motion of pixels as they travel from frame to frame. The full optical flow equation is not solvable, since it is one equations with two unknowns (of x and y velocity). Instead, all optical flow algorithms introduce another constraint equation to make the problem solvable.

So from user point of view, idea is simple, we give some pixels to track; we receive the optical flow vectors of those points. But again there are some problems. Traditionally, we were dealing with small motions for optical flow estimation. This fails when there is large motion. In order to overcome this, we go for pyramids construction. When we go up in the pyramid, small motions are removed and a large motion becomes small motions. So applying Lucas-Kanade there, we get optical flow along with the scale.

A. Optical Flow based Computation

Consider a pixel $I(x, y, t)$ in first frame. Say, it moves by distance (dx, dy) in next frame taken after dt time. So since those pixels are the same and intensity does not change, we can say,

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{1}$$

Then take Taylor series approximation of right-hand side, remove common terms and divide by dt to get the following equation:

$$f_x u + f_y v + f_t = 0 \tag{2}$$

Where,

$f_x = \frac{\partial f}{\partial x}$, $\quad f_y = \frac{\partial f}{\partial x}$,

$u = \frac{dx}{dt}$, $\quad v = \frac{dy}{dt}$

Above equation is called the Optical Flow equation. However, this is valid only for small motions. In order to make it generic for all types of motion, small or large, we go for image pyramid construction. The concept of image pyramids is explained next.

B. Image Pyramid Representation

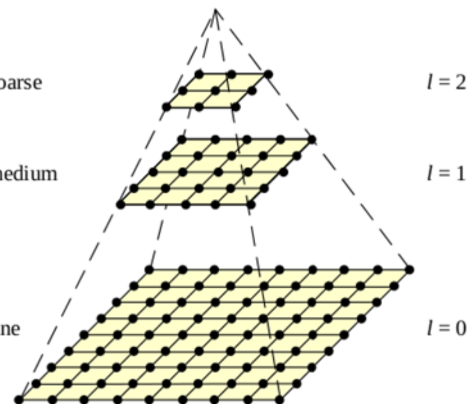

Figure1: Image Pyramid Representation

The above figure is an example of a Gaussian Pyramid. Depicted are three levels of the Gaussian pyramid, levels 0 to 2 presented from bottom to top. We perform optical flow on each of these levels, starting with the finest level and proceeding to the coarsest resolution. This ensures we do not miss any fine-grain motion vectors. Implementation wise, the image frame is sub-sampled by 2 when going up a level. We now calculate optical flow for the current level, using the equation given in the previous section.

Mathematically, for down sampling, this is equal to smoothing the image using a Gaussian kernel, followed by skipping each even numbered row and column. The Gaussian kernel used is of the form:

$$\frac{1}{16}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

The process of rows and columns removal can be,

$$G_0(x, y) = I \tag{3}$$

$$G_{i+1}(x, y) = REDUCE\big(G_i(x, y)\big) \tag{4}$$

C. Implementation

The overall block diagram of the implementation is as follows,

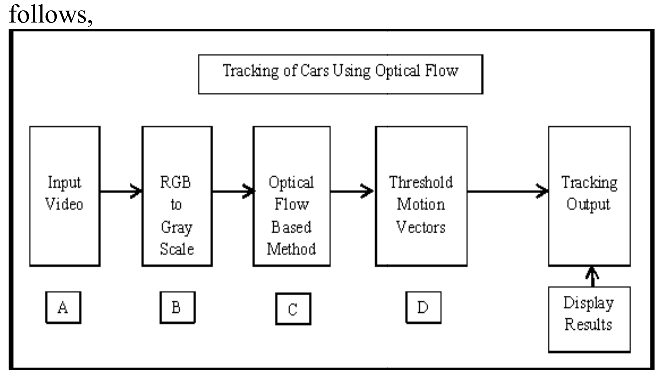

Figure2: Overall Implementation block diagram

The complete implementation is done in MATLAB. The block labeled 'C' is where the major computations are done. Thresholding is performed in the HSV space using the output of stage 'C'. Each colored region in this output forms a separate layer in the HSV space, and hence it is easy to threshold. An example of the output of stage 'C' is shown below:

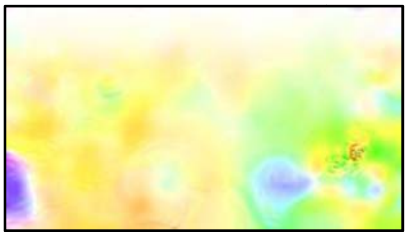

Figure3: Output of stage C

The colored regions above yellow (i.e., yellow / orange / red) constitute the motion vectors that encode the maximally observed pixel motion in a frame.

D. Thresholding in HSV space

In general thresholding algorithms, the input is a gray scale image which is mathematically a single matrix in the RGB (red, green and blue) space. Thresholding operation is mathematically represented as:

$$dst(x,y) = \begin{cases} maxVal, & if\ src(x,y) > threshold \\ 0, & Otherwise \end{cases} \tag{5}$$

Where 0.4 is used as detection threshold in computation and dst the destination and src is the source image. However, the input to the thresholding step here is the colored 'flow image' as shown above, which are mathematically 3 matrices (R, G and B) in the RGB space. Handling thresholding across color channels thus poses a challenge in the RGB space. However, if we are to transform this colored image into HSV (hue, saturation, and luminance) space, the H-component is almost the same as the colored image in RGB space. Each color forms a separate layer in the hue space, thus making color-based thresholding task much simpler.

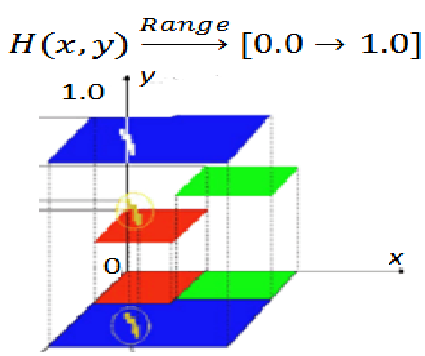

Figure4: Thresholding of Image

E. Morphological Operations

Once the threshold output is available, some amount of morphological operations are required in order to get the binary large objects (blobs) that actually correspond to the object. In general, the blob that has the maximum area will correspond to the object that has moved. However, due to noise in the image, there may be some small blobs that may exist in the threshold image.

This can be resolved using morphological erosion. The erosion of a binary image f by a structuring element s (denoted as, $s(f)$) produces a new binary image $g = s(f)$with ones in all locations (x, y) of a structuring element's origin at which that structuring element s fits the input image f, i.e. $g(x,y) = 1$ is s fits f and 0 otherwise, repeating for all pixel coordinates (x, y).

Thus, using erosion, we first remove off these noisy blobs followed by detection of the centroid of the main blob for tracking and counting.

F. Parallelizing the Optical Flow algorithm

As it can be observed from the optical flow algorithm equations in section A, the motion vector is dependent only upon its own pixel locations in the two frames being considered. This offers a major window into parallelizing the algorithm. Further, when we extract the frames from a video, each pair is independent of every other pair of frames. Thus, the parallel implementation involves two stages of parallelizing – pixel level and frame level.

a) Pixel level : The equation for optical flow indicates that the motion vectors are calculated for every pixel independently. This independence affords the possibility to parallelize the algorithm. Thus, we consider a fixed area around a pixel (say, 9x9), and calculate the motion vectors in each such block of pixels in parallel. This area around a pixel serves as a boundary to prevent any overlap of motion vectors.

b) Frame level: Since motion vector calculation depends only upon the current and the previous frame, we can easily parallelize at this level, i.e., take a pair of frames at a time and find the motion vectors on it. In parallel another pair can undergo the same set of operations for motion vector calculation without affecting the previous pair in any way.

IV. RESULT ANALYSIS

The experimental result consists of comparing the automatic vehicle counting from videos against the manual counting done by the researcher (ground truth). In this table, vehicle video1 having 745 frames denotes the comparatively low traffic flow, vehicle video2 which has 1402 frames denotes an average medium traffic flow and vehicle video 3 having 3500 frames denotes a high traffic scene. The evaluation results obtained by proposed algorithm were compared with a similar purpose algorithm proposed by Hashmi et al. [2] and Jang et al. [23] offering a satisfactory success rate for counting the vehicles in a traffic surveillance system. We thoroughly tested the algorithm using MATLAB and Parallel computing tool box with 16 parallel nodes and found comparatively faster than the previous other results. In the proposed method, over all the results are good and can be applied in real time scenario where we get more than 98% of average accurate counting. Due to variation in traffic scene there is a bit less counting appears in high traffic video sequence.

Table I shows the experimental results obtained by the proposed method and the comparison done with the similar purpose method [2, 23].

TABLE I. COUNTED VEHICLES FROM TRAFFIC VIDEOS USING OUR METHOD AND COMPARISON OF OUR METHOD WITH REFERENCE [2, 23]

Video	Similar Method [1]			Our Proposed Method		
No. of Frames and Avg. Process Time	Exact No. of Vehicl-es in Video	No. Of Vehicles Calcul-ated by Sys [2] and %	No. Of Vehicles Calcul-ated by Sys [23] and %	Processi-ng Time taken in ms	No. Of Vehicles Calcul-aed by our System	Success Rate in Percent
745 (236.81 ms)	20	16 80%	19 95%	184.84 ms	20	100 %
1402 (321.53 ms)	37	30 81.1%	35 94.6%	157.15 ms	37	100 %
3500 (865.41 ms)	97	78 80.5%	90 92.8 %	485.10 ms	91	94 %
Avg.	154	124 80.53 %	144 94.13%		148	98 %

V. CONCLUSION

A simple and effective system which solves the problem of real time video processing under study has been developed. The detection of vehicles in a mix traffic situation of low, medium and high traffic is precisely as expected and the counting algorithm is almost accurate. The limitation of the developed method is that some vehicles is missed by the optical flow detector due to heavy traffic flow condition where the frames has many images in the same defined area or the vehicle is partialy passing from the capture area of camera lenses. The proposed algorithm is accurate in the condition of front facing camera where the vehicles are passing in a streight lane on road.

ACKNOWLEDGMENT

We are thankful to Centre for Graduate Studies, Universiti Teknologi PETRONAS (UTP), Malaysia for providing Graduate Assistantship (GA) support for this research. We are also thankful to YouTube library and UTP library for the traffic videos used for this work.

REFERENCES

[1] Bhaskar, Prem Kumar; Yong, Suet-Peng, "Image processing based vehicle detection and tracking method," *Computer and Information Sciences (ICCOINS), 2014 International Conference on*, vol., no., pp.1,5, 3-5 June 2014.

[2] Hashmi, M.F.; , A.G., "Analysis and monitoring of a high density traffic flow at T-intersection using statistical computer vision based approach," *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, vol., no., pp.52,57, 27-29 Nov. 2012.

[3] Jiann-der Lee; Jun-Ting Wu; Chung-Hung Hsieh; Jong-Chih Chien, "Close range vehicle detection and tracking by vehicle lights," Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on , vol., no., pp.381,386, 26-29 Aug. 2014.

[4] Da Li; Bodong Liang; Weigang Zhang, "Real-time moving vehicle detection, tracking, and counting system implemented with Open CV," Information Science and Technology (ICIST), 2014 4th IEEE International Conference on, vol., no., pp.631,634, 26-28 April 2014.

[5] Chun-Yu Chen; Yu-Ming Liang; Sei-Wang Chen, "Vehicle classification and counting system," Audio, Language and Image Processing (ICALIP), 2014 International Conference on, vol., no., pp.485,490, 7-9 July 2014.

[6] Wei Wang; Gee, T.; Price, J.; Hairong Qi, "Real Time Multi-vehicle Tracking and Counting at Intersections from a Fisheye Camera," Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on , vol., no., pp.17,24, 5-9 Jan. 2015.

[7] Hultqvist, D.; Roll, J.; Svensson, F.; Dahlin, J.; Schon, T.B., "Detecting and positioning overtaking vehicles using 1D optical flow," Intelligent Vehicles Symposium Proceedings, 2014 IEEE , vol., no., pp.861,866, 8-11 June 2014.

[8] Bin-Feng Lin; Yi-Ming Chan; Li-Chen Fu; Pei-Yung Hsiao; Li-An Chuang; Shin-Shinh Huang; Min-Fang Lo; "Integrating Appearance and Edge Features for Sedan Vehicle Detection in the Blind-Spot Area," *Intelligent Transportation Systems, IEEE Transactions on* , vol.13, no.2, pp.737,747, June 2012.

[9] Feris, R.S.; Siddiquie, B.; Petterson, J.; Yun Zhai; Datta, A.; Brown, L.M.; Pankanti, S., "Large-Scale Vehicle Detection, Indexing, and Search in Urban Surveillance Videos," *Multimedia, IEEE Transactions on* , vol.14, no.1, pp.28,42, Feb. 2012.

[10] Soo Teoh and Thomas Bräunl, "A reliability point and Kalman filter-based vehicle tracking technique", Proceedings of the International Conference on Intelligent Systems (ICIS'2012), Penang, Malaysia, pp. 134-138, May 2012.

[11] Mithun, N.C.; Rashid, N.U.; Rahman, S.M.M., "Detection and Classification of Vehicles from Video Using Multiple Time-Spatial Images," *Intelligent Transportation Systems, IEEE Transactions on*, vol.13, no.3, pp.1215,1225, Sept. 2012.

[12] Chieh-Ling Huang; Heng-Ning Ma, "A Moving Object Detection Algorithm for Vehicle Localization," *Genetic and Evolutionary Computing (ICGEC), 2012 Sixth International Conference on* , vol., no., pp.376,379, 25-28 Aug. 2012.

[13] Morris, B.T.; Cuong Tran; Scora, G.; Trivedi, M.M.; Barth, M.J., "Real-Time Video-Based Traffic Measurement and Visualization System for Energy/Emissions," *Intelligent Transportation Systems, IEEE Transactions on* , vol.13, no.4, pp.1667,1678, Dec. 2012.

[14] Bouvie, C.; Scharcanski, J.; Barcellos, P.; Lopes Escouto, F., "Tracking and counting vehicles in traffic video sequences using particle filtering," *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International* , vol., no., pp.812,815, 6-9 May 2013.

[15] Zhao, R.; Wang, X., "Counting Vehicles from Semantic Regions," *Intelligent Transportation Systems, IEEE Transactions on*, vol.14, no.2, pp.1016,1022, June 2013.

[16] P. Rajesh, M. Kalaiselvi Geetha and R.Ramu; "Traffic density estimation, vehicle classification and stopped vehicle detection for traffic surveillance system using predefined traffic videos", Elixir Comp. Sci. & Engg. 56A (2013) 13671-13676, 2013.

[17] Deng-Yuan Huang, Chao-Ho Chen, Wu-Chih Hu, Shu-Chung Yi, and Yu-Feng Lin; "Feature-Based Vehicle Flow Analysis and Measurement for a Real-Time Traffic Surveillance System", Journal

of Information Hiding and Multimedia Signal Processing, Volume 3, Number 3, pp. 2073-4212, July 2012.

[18] Suvarna Nandyal1, Pushpalata Patil; "Vehicle Detection and Traffic Assessment Using Images", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 9, pg.8 – 17, September 2013.

[19] Ravi Kumar Kota, Chandra Sekhar Rao T; "Analysis Of Classification And Tracking In Vehicles Using Shape Based Features", International Journal of Innovative Research and Development, IJIRD, Vol 2. Issue 8, pp. 279-286, August 2013.

[20] T. Horprasert, D Harwood, L. S. Davis, "A statistical approach for real time robust background subtraction and shadow detection," in proceedings of IEEE ICCV'99 Frame rate workshop, pp. 1-19, 1999.

[21] S. Suzuki, K. Abe., "Topological Structural Analysis of Digital Binary Images by Border Following.", CVGIP, v.30, n.1, pp. 32-46, 1985.

[22] Sharma, R.; Taubel, G.; Jiann-Shiou Yang, "An optical flow and Hough transform based approach to a lane departure warning system," *Control & Automation (ICCA), 11th IEEE International Conference on* , vol., no., pp.688,693, 18-20 June 2014.

[23] Hyeok Jang; In-Su Won; Dong-Seok Jeong, "Automatic Vehicle Detection and Counting Algorithm," IJCSNS International Journal of Computer Science and Network Security, VOL.14 No.9, September 2014.

[24] Yuqiang Liu; Kunfeng Wang; Fei-Yue Wang, "Visual vehicle tracking based on conditional random fields," Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on , vol., no., pp.3106,3111, 8-11 Oct. 2014.