Vehicle Counting And Classification Using Kalman Filter And Pixel Scanner Technique And Its Verification With Optical Flow Estimation

GJCST Classification
I.3.8,1.3.m

H.S. Mohana[1], Aswatha Kumar. M [2] and G. Shivakumar[3]
[1,3]Malnad College of Engineering, Hassan, Karnataka
[2]MS Ramaiah Institute of Technology, Bangalore, Karnataka
hsm@mcehassan.ac.in, maswatha@yahoo.com,gs@mcehassan.ac.in

***Abstract*-Vehicle tracking is important in traffic monitoring systems. The behaviors of regions of moving vehicles are complicated, since the regions may combine or break during the tracking due to mistakes in vehicle detection and tracking or vehicles' overlapping with each other, and as a result, region matching simply according to similarities between successive frames is not enough to achieve reliable results. This paper proposes a novel tracking strategy that can robustly track and classify the objects within a fixed environment. We define a robust model-based tracker and classifier using kalman filtering combined with pixel scanner. The tracking is done by fitting successively more elaborate models on the tracked region and the segmentation is done by extracting the regions of the image that are consistent with the computed model of the target. We adopt a competitive and efficient dynamic Kalman filtering to adaptively update the object model by adding new stable features as well as deleting inactive features. In the next stage we need to check each and every frame for object recognition. This work introduce a diagonal pixel scanner to identify the objects. The result is verified further by implementing optical flow analysis. The tracking, counting and classification of object/vehicle have produced very consistent result. The average accuracy with short length video clipping is greater than 98%.**

Keywords-Kalman filter, pixel scanner, object classification and object counting.

I. INTRODUCTION

Traffic on roads may consist of pedestrians, ridden or herded animals, vehicles, streetcars and other conveyances, either singly or together, while using the public way for purposes of travel. Traffic is often classified by type: heavy motor vehicle (e.g., car, truck); other vehicle (e.g., moped, bicycle); and pedestrian. Computer vision is concerned with the theory for building artificial systems that obtain information from images. The image data can take many forms, such as a video sequence and /or views from multiple cameras.

The Kalman filter produces estimates of the true values of measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value, and computing a weighted average of the predicted value and the measured value. The most weight is given to the value with the least uncertainty. The estimates produced by the method tend to be closer to the true values than the original measurements because the weighted average has a better estimated uncertainty than either of the values that went into the weighted average.

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow can arise from relative motion of objects and the viewer. Consequently the optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement. Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects. Attempts have been made to perform such segmentation using differences between successive image frames.

Several papers address the problem of recovering the motions of objects relative to the viewer from the optical flow. Some recent papers provide a clear exposition of this enterprise. The mathematics can be made rather difficult, by the way, by choosing an inconvenient coordinate system. In some cases information about the shape of an object may also be recovered. It is assumed that the optical flow has already been determined. Although some reference has been made to schemes for computing the flow from successive views of a scene, the specifics of a scheme for determining the flow from the image have not been described. Related work has been done in an attempt to formulate a model for the short range motion detection processes in human vision .The pixel recursive equations of Netravali and Robbins designed a method for coding motion in television signals. This bear some similarity to the iterative equations developed in this paper. A recent review of computational techniques for the analysis of image sequences contains over 150 references. It suggests that the optical flow cannot be computed at a point in the image independently of neighboring points without introducing additional constraints, because the velocity field at each image point has two components while the change in image brightness at a point in the image plane due to motion yields only one constraint.

Velocity is a vector quantity which refers to "the rate at which an object changes its position." Imagine a person moving rapidly - one step forward and one step back - always returning to the original starting position. While this might result in a frenzy of activity, it would result in a zero velocity. Because the person always returns to the original position, the motion would never result in a change in position. Since velocity is defined as the rate at which the position changes, this motion results in zero velocity. If a person in motion wishes to maximize their velocity, then that person must make every effort to maximize the amount that they are displaced from their original position. Every step must go into moving that person further from where he or she started. For certain, the person should never change directions and begin to return to the starting position.

Considering a patch of pattern where brightness varies as a function of one image coordinate but not the other. Movement of the pattern in one direction alters the brightness at a particular point, but motion in the other direction yields no change. Thus components of movement in the latter direction cannot be determined locally.

II. RELATED WORK

After referring some of the technical papers under Traffic analysis resulted in novel idea to reach the objective. Some of the papers referred are presented here. Discussion on referred paper provides the limitations of those methods and how our approach seems to be advantageous over them. Reference [1] presents algorithms for vision-based detection and classification of vehicles in monocular image sequences of traffic scenes recorded by a stationary camera. Processing is done at three levels: raw images, region level and vehicle level. It is observed that data acquisition of monocular image sequence is a very tedious task. The information gathered is more than required as it covers the region level information as well. the present method incorporates .avi standard sequences which can be easily manipulated and worked upon. The computational time and memory requirement is much less than compared to the above method. In [2] a study on a stand-alone image tracking system for automatic traffic monitoring is presented. The proposed image tracker consists of three parts: an edge detection module, an image tracking module and a traffic monitoring module. The above paper uses a tracking system which automatically does the monitoring system with no manual interference in real time. It consists of edge detection, image tracking and monitoring the traffic. A novel tracking strategy is proposed in [3] that can robustly track an object within a fixed environment. Authors define a robust model-based tracker using Kalman filtering combined with recursive least squares. It uses a tracking done by fitting successively more elaborate models on the tracked region and the segmentation is done by extracting the regions of the image that are consistent with the computed model of the target. But the present work adopts a competitive and efficient dynamic Kalman filtering to adaptively update the object model by adding new stable features as well as deleting inactive features. Reference [4] reads real

time monitoring video from communications department and converts it into images. After that, we change them into corresponding gray images and carry out image binarization with dynamic multiple thresholds method which selects thresholds depending on pixel, grayscale and pixel position. Here the system updates the background periodically background refreshing method. We also put forward an adaptive background subtraction method, which can remove noise, to identify the moving objects and get total movement in a given time.

A new approach is developed in [5], in order to track the vehicles, which is known as region processing. The regions may combine or break during the tracking due to mistakes in vehicle detection and tracking or vehicles overlapping with each other, so a method to overcome this effect is developed and accomplished. In this paper, a vehicle tracking method is proposed to reduce mistake in spatial segmentation. "Temporary vehicle", "confirmed vehicle" and the corresponding judging rules are presented. A fuzzy judgment is proposed to determine whether vehicle overlapping occurs or not. Reference [6] presents a practical real-time traffic monitoring system based on object detection and tracking for measuring traffic parameters such as speed and volume. In the proposed system, background is modeled by using edge information and this model is used for extracting foreground moving objects. The advantage of using edge information to model the background is that it is more robust to the lighting variation. The extracted moving objects are then tracked by using Lukas-Kanade (Pyramid) optical flow algorithm. Only the successfully tracked vehicle will be considered for retrieving traffic information. In [7] a real time video surveillance is presented for traffic monitoring of vehicle volume on major highways. This paper deals with the determination of traffic volume automatically in real time to dynamically plan their trips more efficiently. A new method known as the virtual line analyzer detects vehicles as they cross the virtual boundary. The goal of this paper is to provide real time and accurate vehicle counter when using stationary web cams, fixed highways and lanes, and deterministic vehicle characteristics.

A real time system for pedestrian tracking in sequences of grayscale images acquired by a stationary camera is presented in [8]. The proposed scheme is also useful for the detection of several diverse tracking objects of interest. Blob tracking is modeled as a graph optimization problem. Pedestrians are modeled as dynamic rectangular patches. Kalman filtering is used in order to estimate pedestrian parameters. Disadvantage is that this system assumes that all objects in the scene are pedestrians. This means that if another object /vehicle appear into the scene, it will be treated as group of pedestrians.

A computer vision based approach for event detection and data collection at traffic intersections is proposed in [9]. It implements a robust tracking algorithm for targets through combination of multiple uses and multiple motion models. Also, event detection system using results of a switching Kalman filter in combination with some simple rules is implemented. The estimation is that the detected events are very simple based on simple rules for detection. The system makes no distinction between a target moving and stopped vehicle in the scene.

III. IMPLEMENTATION

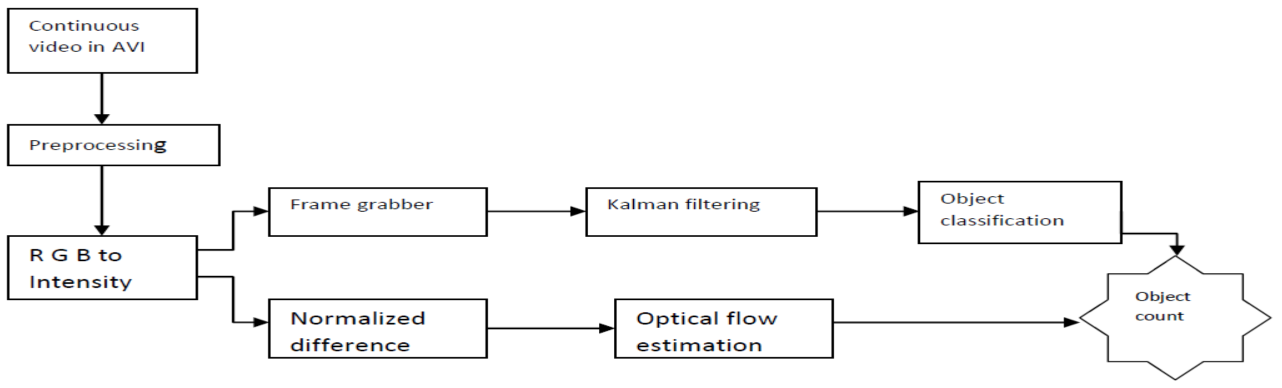


Fig.1: Block diagram representation of the developed system.

The methodology described above is followed and the computation is done .The whole procedure is represented by a block diagram as shown in Fig.1. The input to the system is video sequence which is in MPEG2 format.The video sequences are then converted into AVI format with the help of software. Then this is converted into frame vise using frame grabber software. Then a frame with no vehicle is taken and this is said to be a reference image which is then used for background subtraction. Continuous frames are sent so that the subtraction takes place within the reference image and the current image. Thus we accomplish the background subtraction via which we determine the object segmentation which is useful in classification of vehicles. Next this is passed through the Kalman filter. The state of the object is upgraded and thus it helps in the classification of vehicles. Pixel scanner is just acting like asensor. Initially while fixing pixel scanner, each and every pixels having different RGB values for different frames. Basically a pixel scanner line consisting of more than 100 pixels, once an object enters into the frame and when it passes through pixel scanner, there will be drastic changes in RGB values of the pixel until the object is in contact with the pixel scanner.

IV. OPTICAL FLOW

First the video is selected which is in avi format. then it is subjected to intensity form by converting its RGB into its respective intensity. Then this intensity is next converted into single image format where white pixel representation is obtained. To this the optical flow technique is implemented with which we get the velocity. The optical flow method used here is the Horn-schunk[10] method where the horizontal and vertical components are taken into consideration and the difference between them are calculated so that the velocity is determined. This velocity is then subjected to thresholding and median filtering where morphological features are extracted. This results with the motion vector. Thus the determination of optical flow is implemented. Further, the result is used for counting and classification of vehicles.

V. OBJECT RECOGNITION BY USING KALMAN FILTER

Here we need to maintain separate data base for each and every object, and it is totally depending upon your camera position that's related to your road. Object data base will vary according to distance between camera and main road, so accordingly we need to maintain an object data base.

Take an example: Two-wheeler

Distance between camera and road ==5m

Overall road width==10m

Total from camera position==15m

Suppose bike travelling in 10m distance from camera,

Assume that height==5cm

Width==7cm (by using keen observation)

Find area now,

Area A== (1/2)*height*width==17.5cm/sq

Now construct a rectangle of area between 15-20cm/sq by using kalman filter.

It means in between 15-20 we should suppose to maintain 10 or more than 10 rectangle areas in our background database. Then apply Kalman filter to each and every frames, find object areas and compare with backgrounds present in our data base, when it matches (not 100 percent) almost, then you can easily recognize a given object. Similarly you can do it for four-wheeler and heavy objects also. For object display operation we have taken a frame which is having much change in their almost all pixels RGB values which are all present in the pixel scanner line, and then algorithm can read that frame easily.

By using velocity vectors in optical flow, we can easily find out vehicle count by using pixel scanner line. Green value will increase once velocity vector reaches pixel scanner line.

VI. DIFFERENCE FILTER

1. Compute I_x and I_y using the kernel $[-1 \quad 8 \quad 0 \quad -8 \quad 1]/12$ and its transposed form.
   If you are working with fixed-point data types, the kernel values are signed fixed-point values with word length equal to 16 and fraction length equal to 15.
2. Compute I_t between images 1 and 2 using the $[-1 \quad 1]$ kernel.
3. Smooth the gradient components, I_x, I_y, and I_t, using a separable and isotropic 5-by-5 element kernel whose effective 1-D coefficients are $[1 \quad 4 \quad 6 \quad 4 \quad 1]/16$. If you are working with fixed-point data types, the kernel values are unsigned fixed-point values with word length equal to 8 and fraction length equal to 7.
4. Solve the 2-by-2 linear equations for each pixel using the following method:

- If $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix}$

  Then the eigen values of A are $\lambda_i = \frac{a+c}{2} \pm \frac{\sqrt{4b^2 + (a-c)^2}}{2}; i = 1,2$

In the fixed-point diagrams, $\frac{a+c}{2}, Q = \frac{\sqrt{4b^2 + (a-c)^2}}{2}$

When the block finds the eigen values,it compares them to the threshold,that corresponds to the value you enter for the Threshold for noise reduction parameter. The results fall into one of the following cases: The Compute optical flow between, N, and Velocity output parameters are described in Horn-Schunck Method. Use the Threshold for noise reduction parameter to eliminate the effect of small movements between frames. The higher the number, the less small movements impact the optical flow calculation.

Case 1: $\lambda_1 \geq \tau$ and $\lambda_2 \geq \tau$

A is nonsingular, so the block solves the system of equations using Cramer's rule.

Case 2: $\lambda_1 \geq \tau$ and $\lambda_2 < \tau$

A is singular (noninvertible), so the block normalizes the gradient flow to calculate *u* and *v*.

Case 3: $\lambda_1 < \tau$ and $\lambda_2 < \tau$

The optical flow, *u* and *v*, is 0.

The Compute optical flow between, N, and Velocity output parameters are described in Horn-Schunck Method. Use the Threshold for noise reduction parameter to eliminate the effect of small movements between frames. The higher the number, the less small movements impact the optical flow calculation.

VII. DERIVATIVE OF GAUSSIAN

If you set the Temporal gradient filter parameter to Derivative of Gaussian, the block solves for *u* and *v* using the following steps. You can see the flow chart for this process at the end of this section:

i. Compute I_x and I_y using the following steps:
   a. Use a Gaussian filter to perform temporal filtering. Specify the temporal filter characteristics such as the standard deviation and number of filter coefficients using the Number of frames to buffer for temporal smoothing parameter.
   b. Use a Gaussian filter and the derivative of a Gaussian filter to smooth the image using spatial filtering. Specify the standard deviation and length of the image smoothing filter using the Standard deviation for image smoothing filter parameter.

ii. Compute I_t between images 1 and 2 using the following steps:
   a. Use the derivative of a Gaussian filter to perform temporal filtering. Specify the temporal filter characteristics such as the standard deviation and number of filter coefficients using the Number of frames to buffer for temporal smoothing parameter.
   b. Use the filter described in step 1b to perform spatial filtering on the output of the temporal filter.

iii. Smooth the gradient components, I_x, I_y, and $I_t$using a gradient smoothing filter. Us
Standard deviation for gradient smoothing filter parameter to specify the standard de
the number of filter coefficients for the gradient smoothing filter.

iv. Solve the 2-by-2 linear equations for each pixel using the following method:

If $$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix}$$

Then the eigenvalues of A are

$$\lambda_i = \frac{a+c}{2} \pm \frac{\sqrt{4b^2 + (a-c)^2}}{2}; i = 1,2$$

When the block finds the eigenvalues, it compares them to the threshold, that corresponds to the value you enter for the Threshold for noise reduction parameter. The results fall into one of the following cases:

Case 1: $\lambda_1 \geq \tau$ and $\lambda_2 \geq \tau$

A is nonsingular, so the block solves the system of equations using Cramer's rule.

Case 2: $\lambda_1 \geq \tau$ and $\lambda_2 < \tau$

A is singular (noninvertible), so the block normalizes the gradient flow to calculate u and v.

Case 3: $\lambda_1 < \tau$ and $\lambda_2 < \tau$

The optical flow, u and v, is 0.

Select the Discard normal flow estimates when constraint equation is ill-conditioned check box if it is required that the block to set the motion vector to zero when the optical flow constraint equation is ill-conditioned. The block calculates these motion vectors on a pixel-by-pixel basis.
Select the Output image corresponding to motion vectors (accounts for block delay) check box if required that the block to output the image that corresponds to the motion vector being output by the block.

VIII. Algorithm

Initially algorithm will read continuous movie in AVI format by using MATLAB, then we will separate out the frames by using frame grabber. In the next stage we need to check each and every frame for object recognition. So we can introduce a diagonal pixel scanner to identify the objects. Pixel scanner is just acting like a sensor. Initially while fixing pixel scanner, each and every pixels having different RGB values for different frames. Basically a pixel scanner line consisting of more than 100 pixels, once an object enters into the frame and when it passes through pixel scanner, there will be drastic changes in RGB values of the pixel until the object is in contact with the pixel scanner. Now we can consider output of pixel scanner lines, I mean RGB values of each and every pixels present in the pixel scanner.

Following are the steps involved:

i. Pixel 1: initial RGB value=30, 20, 10….when there is no object present in the scene
ii. Changed R1G1B1 values= 220, 110, 80….presence of object in the scene.
iii. Subtract RGB from R1G1B1
iv. Initialize counter……c==0
v. If (R1G1B1-RGB) is greater than or equal to (10,10,10)====increment count
Count ==c+1:

Else

Count==c:

End

vi. Display the result of object counting.

Object Recognition By Using Kalman Filter

At this level of analysis it is needed to maintain separate data base for each and every object, and it's totally depending upon your camera position that's related to your road. Object data base will vary according to distance between camera and main road, so accordingly we need to maintain an object data base.

Specific Identifier

Two-wheeler
Distance between camera and road ==5m
Overall road width==10m
Total from camera position==15m
Suppose bike travelling in 10m distance from camera,
Assume that height==5cm
Width==7cm (by using keen observation)
Find area now,
Area A== (1/2)*height*width==17.5cm/sq
Now construct a rectangle of area between 15-20cm/sq by using Kalman filter.
It means in between 15-20 we should suppose to maintain 10 or more than 10 rectangle areas in our background database. Then apply Kalman filter to each andevery frames, find object areas and compare with backgrounds present in our data base, when it matches (not 100 percent) almost, then you can easily recognize a given object. Similarly you can do it for four-wheeler and heavy objects also.
For object display operation we have taken a frame which is having much changes in their almost all pixels RGB values which are all present in the pixel scanner line, then you can read that frame easily.
By using velocity vectors in optical flow, we can easily find out vehicle count by using pixel scanner line. Green value will increase once velocity vector reaches pixel scanner line.

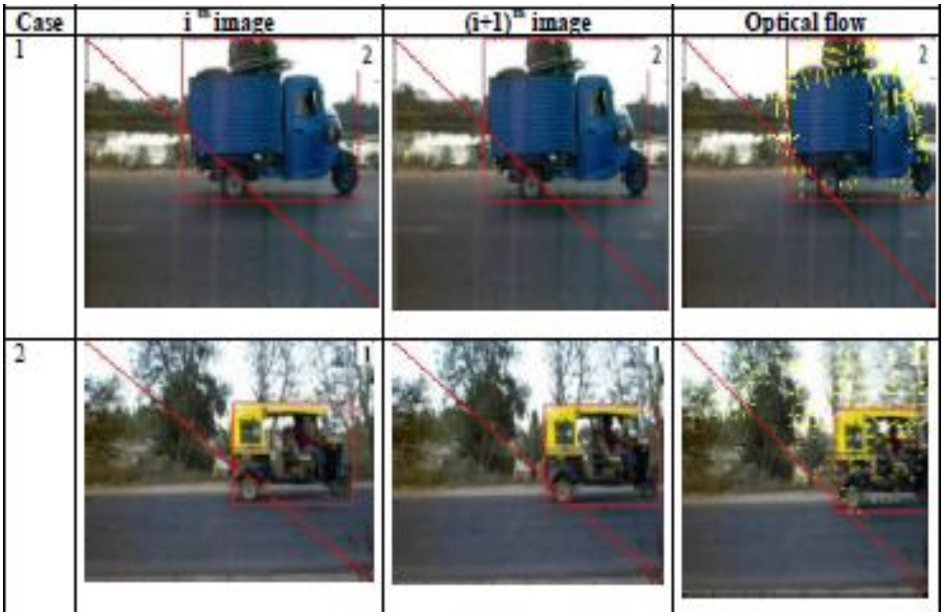
Table 1: Tabulates results achieved for different natural video steams

	Actual Classification	Count	Estimated Classification	Count	Error	Accuracy
Case1	Two wheeler	1	Two wheeler	1	0	100%
	Four wheeler	4	Four wheeler	4	0	100%
	Heavy vehicles	3	Heavy vehicles	3	0	100%
	Total	8	Total	8	0	100%
Case2	Two wheeler	4	Two wheeler	1	0	100%
	Four wheeler	4	Four wheeler	1	0	100%
	Heavy vehicles	1	Heavy vehicles	1	0	100%
	Total	9	Total	9	0	100%
Case3	Two wheeler	7	Two wheeler	6	0	85.71%
	Four wheeler	5	Four wheeler	5	0	100%
	Heavy vehicles	2	Heavy vehicles	2	0	100%
	Total	14	Total	13	0	95.12%
Case 4	Two wheeler	4	Two wheeler	0	0	100%
	Four wheeler	3	Four wheeler	3	1	100%
	Heavy vehicles	2	Heavy vehicles	6	0	100%
	Total	9	Total	9	1	100%
	Average				**1.6**	**98.78%**

IX. Presentation of Results

In order to test the proposed algorithm, several sets of natural image sequences have been used. Real image sequences, recorded in MPEG2 format have been used with camera, in fixed position to capture the aerial view of the road. Different natural traffic videos are taken in situations where obstacles are found in the line of view, vehicle shadows, building shadows in the path and oblique view of the traffic. The first set of images is taken in order to establish the reference images under different illumination condition from morning till evening. Four such reference frames have been identified for experimentation.

In the present work, a platform has been created so that complete automation of dynamic and intelligent traffic control devoid the human intervention. Monocular camera with fixed resolution of 1024X1024 with a frame rate of 30 is used to acquire the data. The present algorithm translates image size 1024X1024 to 200X200 in order to reduce the computational complexity. Table.1 showcases the result established through the implementation of the present algorithm.



Optical Flow Estimation

Fig.2-Display of results of optical flow needle diagram for different vehicles

Display of results of optical flow needle diagram for different vehicles is as shown in Fig.2. The needle diagram clearly indicates the direction of movement of the vehicle. Hence, bidirectional vehicular movement analysis is achieved apart from counting and classification of vehicles..

Reslts Presentation: Object Counting By Using Kalman Filter

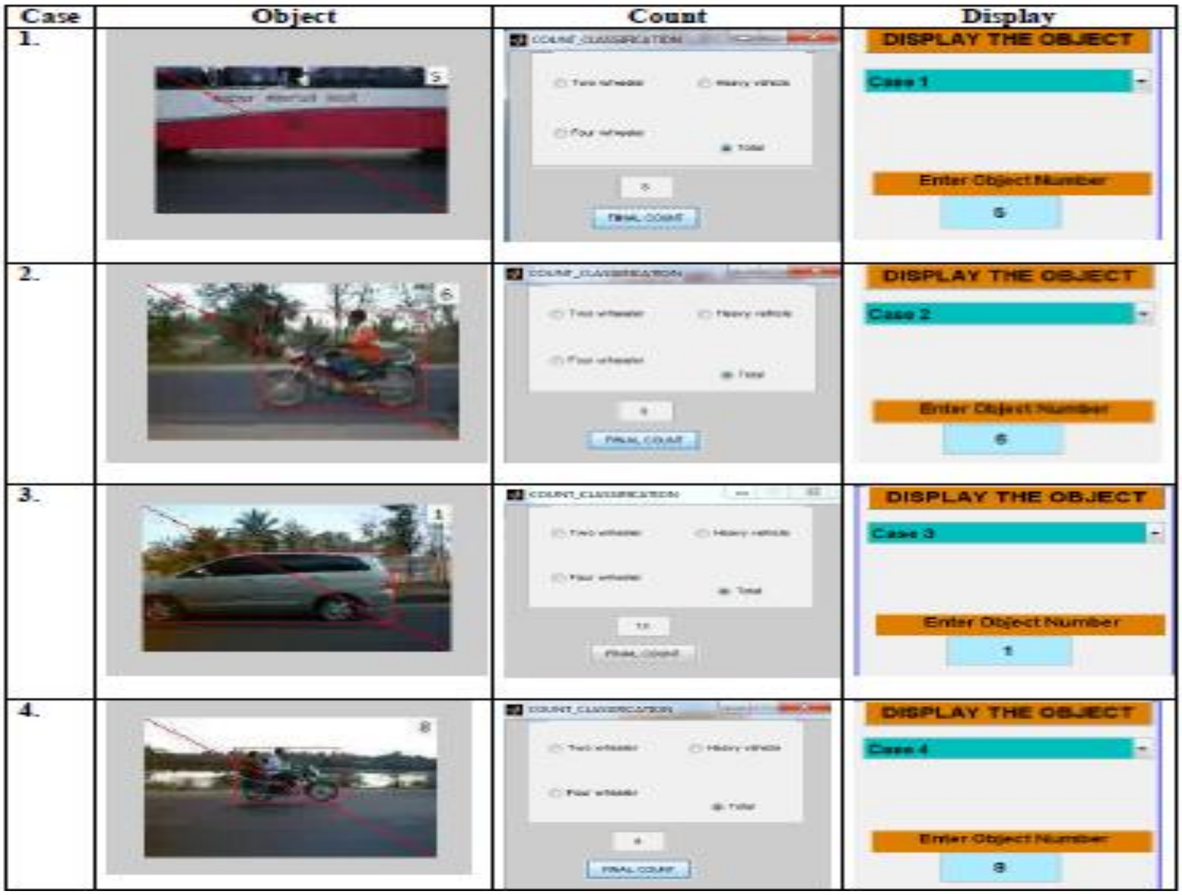


Fig.3

Display of results of Kalman filter for different vehicles.

The classification of vehicles based on Kalman filter is showcased in Fig.3. The result presentation consists of the vehicle count, its sequence of appearance and type. It also provides the insight to the frame number at which each vehicle passed though the geometric center of the frame.

X. Conclusion

The result established is consistent with good repeatability. The system developed consider output of pixel scanner lines, it means that RGB values of each and every pixels present in the pixel scanner. We also notice that the proposed method fails if the traffic is too congested, because in this case, vehicles may overlap from the beginning to the end, or more than two vehicles are overlapped with each other, so it is difficult to distinguish each of the vehicles. The Optical Flow block estimates the direction and speed of object motion from one image to another or from one video frame to another using either the Horn-Schunck or the Lucas-Kanade method in order to verify the result.

XI. References

1) Ching-Po Lin, Jen-.Chao Tai and Kai-Tai Song, “Traffic Monitoring Based On Real Time Image Tracking”, IEEE International Conference on Robotics and automation,pp2091-96,2003.
2) Feng Yi-wei, Guo Ge ,Zhu Chao-qun, “Object Tracking By Kalman Filtering And ecursive Least Squares based On 2D Image Motion”, IEEE International Conference on Computational Intelligence and Design,pp106-109,2008.
3) Gang-Yi Jiang, Mei Yu, Sheng-Nan Wang, Rang-Ding Wang, “New Approach To Vehicle Tracking Based On Region Processing”, IEEE International Conference on Machine learning and Cybernetics, pp5028-33,2005.
4) Kiratiratanapruk K. and Supakorn Siddhichai, “Practical Application for Vision-Based Traffic Monitoring System”, 978-1-4244-3388-9,2009.
5) Fei Zhu, “ A Video-Based Traffic Congestion Monitoring System Using Adaptive Background

Subtraction", Second International Symposium on Electronic Commerce and Security, pp73-77,2009.

6) Belle L. Tseng, Ching-Yung Lin and John R. Smith, "Real-Time Video Surveillance For Traffic Monitoring Using Virtual Line Analysis", IEEE 0-7803-7309-9, pp541-544,2002.
7) R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," IEEE Transactions on Intelligent Transportation System., vol. 1, no. 2, pp. 119–130, Jun. 2000.
8) Osama Masoud and Nikolas P.Papanikolopoulos, "A Novel method for tracking and counting pedestrians in Real time using a single camera", IEEE Transactions on Vehicular Technology ,Volume 50, no.5,sept 2001.
9) Harini Veeraraghavan, Paul Schrater and Nikolas Papanikolopoulos, "Switching Kalman filter-Based approach for Tracking and Event Detection at Traffic Intersection", IEEEInternational conference on control and Automation , pp. 1167-1172, 2005.
10) B.K.P. Horn and B.G. Schunck, " Determining optical flow", Artificial Intelligence, 17; 185-203,1981.