Real Time Traffic Surveillance Using Canny Operator And Fuzzy Hybrid Detection

Sai Smruthi Rajagopal[1], Saravanan Nakkeeran[2], Varsha Jayaraman[3]
[1, 2, 3] Dept of Computer Science
[1, 2, 3] College Of engineering guindy

Abstract- *Every day growth of vehicles has become an increased problem of urbanism especially in major cities. Traffic control and monitoring using video cameras has recently drawn increasing attention, due to the significant advances in the field of computer vision. Many commercial and research systems use video processing, aiming to solve specific problems in road traffic monitoring. Vision based traffic Surveillance forms an integral part of any intelligent traffic system. Although many surveillance methods have been presented, two problems still need to be solved in congested situations. First is the low angle of the camera which makes the vehicles appear as connected visually in images. Therefore, occluded vehicles need to be separated to achieve detection accuracy. Second is the inaccuracy in background updating. The proposed system uses Fuzzy Hybrid detection Mechanism to provide an efficient way to handle the shortcomings in congested scenarios. The proposed method involves three stages. First is the background extraction and updating to separate the intended object from the clutter. During background modelling, an efficient shadow-removal technique is adopted for efficient retrieval of the image under interest alone. Next, we have the vehicle detection with moving edge feature extraction. This approach detects the vehicle candidates from the foreground image, and resolves problems such as headlight effects. The tracking technique is then employed to track the displacement in position of the vehicles in consecutive frames which is followed by a tracking verification using Fuzzy Hybrid Information Inference Mechanism. Finally, a method that compensates for error cases under congested conditions is applied to refine the tracking qualities.*

Keywords- Congested condition, traffic surveillance, vehicle detection

I. INTRODUCTION

Traffic congestion has been a significantly challenging problem. It has been widely realized that increase of preliminary transportation infrastructure has not been able to relieve city of congestion. Increasing reliance on traffic surveillance is needed to better understand traffic flow. Traffic light scheduling is the traditional approach, where vehicles proceed in a stop-and-go style as per the occurrence of green light. These signals work on a predefined hardware, where the red, orange and green lights are turned on and off after fixed time periods as dictated by the stipulated values in the software. These signals operate irrespective of traffic conditions on the road, without any real-time intelligence. This leads to heavy congestion at intersection joints, disabling the smooth flow of traffic and leading to long vehicular queue length. The time intervals in which the signals should be changed is configured by the administrator, but expecting the administrator to monitor the live traffic feed and change the timer accordingly isn't pragmatic. Hence there is a requirement of a system which offers flexibility to control traffic, based on real time vehicle situation on roads. So, there is a pressing need for an Intelligent Transportation system(ITS) to predict the traffic flow based on monitoring traffic congestions.

Recent efforts on traffic light control focus on adaptive and smart traffic light scheduling, mainly by making use of computational intelligence , including evolutionary computation algorithm , fuzzy logic, neural network and machine learning .The ever-increasing traffic congestion, accompanied by unpredicted emergencies and accidents have motivated the development of intelligent transportation systems (ITS). Intelligent Transportation Systems (ITS) represent a major transition in transportation on many dimensions. ITS has various applications, ranging from traffic surveillance, collision avoidance, to automatic transportation pricing. Among others, traffic control at intersections has been always a key issue in the research and development of ITS. Vision based Intelligent Transportation System helps to extract useful and precise traffic information for traffic image analysis and traffic flow control applications like vehicle count, vehicle velocity, traffic lane changes, license plate recognition, etc. Traffic control and monitoring using video cameras has recently drawn increasing attention, due to the significant advances in the field of computer vision. Many commercial and research systems use video processing, aiming to solve specific problems in road traffic monitoring. Vision based traffic Surveillance forms an integral part of any intelligent traffic system. Increasing reliance on traffic surveillance is needed to better understand traffic flow.

II. RELATED WORKS

Kanhere and Birchfield presented a taxonomy for roadside camera This method implements segmenting and tracking vehicles on highways using a camera that was of relatively low-level approach that takes into account all the available measurements, resulting in reduced error as well as overcoming the inherent ambiguity in the single-vanishing-point solutions. Tsaiss Vehicle Detection Using Normalized Colour and Edge Map is another novel vehicle detection approach for detecting vehicles from static images using colour and edges. It introduces a new colour transform model to find important vehicle colour for quickly locating possible vehicle candidates. After finding possible vehicle candidates, three important features, including corners, edge maps, and coefficients of wavelet transforms, are used for constructing a cascade multichannel classifier followed by scanning. Houben et al. extracted maximum phase congruency and edges from stereo images and matched together with local matching algorithm, then processed by maximum spanning tree clustering algorithm to group the points into vehicle objects. An Automatic Traffic Surveillance System for Vehicle Tracking and Classification method is used to estimate important traffic parameters from video sequences using only one camera. Categorizes vehicles into more specific classes using linearity feature in vehicle representation. It uses line-based shadow algorithm which uses a set of lines to eliminate all unwanted shadows. Detecting and Tracking Moving Objects for Video Surveillance relies on a graph representation of moving objects which allows to derive and maintain a dynamic template of each moving object by enforcing their temporal coherence. This inferred template along with the graph representation allows us to characterize objects trajectories as an optimal path in a graph.

III.SYSTEM DESIGN

The proposed method involves three phases for improved video based traffic surveillance: The first phase deals with background extraction and modelling for improvised foreground detection. Background model is one of the most useful methods for change detection. Most commonly used Background Modelling methods are sigma delta estimation (SDE) and the Gaussian Mixture Model. Gaussian mixture model works well when the load is free. A shadow removal technique is adopted for efficient retrieval of image under interest alone. Under congested conditions, many vehicles cover the background pixels for a long period. The updating problems become difficult because the background image needs to be accurately updated and it also needs to simultaneously detect moving objects. Therefore, the occupancy (K) in (1) is computed to distinguish whether the flow is free or forced. If the traffic flow is free, a normal updating flow is performed. Else, small-range updating is adopted.

$$K=(Ni \times 100\%)/N$$

where N is the number of frames in a period, and Ni is the number of frames where the lane is occupied by vehicles.

The second phase is to detect vehicles. During vehicle detection, our approach detects the vehicle candidates from the foreground image (i.e.) moving objects are separated from input image using Foreground block segmentation. First, foreground segmentation and moving-edge extraction are used to obtain the moving features of vehicles. Then, connected component labeling is used to identify the shapes of objects for false merged object separation. The third phase is that of Vehicle tracking and verification. The tracking technique is employed to track vehicles in consecutive frames. First, scanning method detects edge features and boundaries of tracking vehicles. Next, Fuzzy Hybrid Detection Mechanism (FHDM) is employed to verify tracked vehicles. Two features are utilized in FHDM, i.e., the color similarity and the area consistency. Higher scores will be obtained if the search results match the previous tracking results. Three linguistic variables, i.e., low, medium and high, are defined as the degrees of the color and the area comparisons, which were used as the inputs of the fuzzy inference system. The output degree is formulated using the fuzzy sets and fuzzy rules. Finally, a method that compensates for error cases under congested conditions is applied to refine the tracking qualities. The vehicle counts of tracked vehicles is displayed to make the surveillance system more efficient.

IV. BACKGROUND PROCESSING

Moving object detection from subsequent frames in video streams is one of the most vibrant areas of research in computer vision. It is the rudimentary step for extracting interested aspects or objects in various vision based applications such as traffic monitoring (pedestrian detection, vehicle detection), automated video surveillance (human detection, anomaly detection), and control applications (automated robot bodies, human computer interaction), etc.

The objective of a background modelling algorithm is to distinguish moving objects or foreground from static parts of the scene or background. In most of the cases, background is not already known and needs to be generated automatically by the background subtraction algorithm. When background image is available, moving objects can be obtained by subtracting background image from the current frame.

Background Modelling is achieved here by adopting Gaussian Mixture Models. GMM assumes that background is already known in the scene, and hence it can be processed using Gaussian distributions. This method can deal with illumination changes in real time video sequences. This method however fails when the traffic is congested as foreground objects cover most of the background image pixels. The second approach proposed is the Small Range Updating which helps model the background even under congestion. This method comprises of three fundamental steps, namely

- **Background Modelling:**

A) Gaussian Mixture Model (GMM)

In this phase, a referential model is taken, and a background image is generated using this model. In GMM based approach, each pixel is modelled as a mixture of K Gaussian distributions. The recent history of a pixel at any time instant t can be written as {X1, X2 …. Xt}. The probability of observing the current pixel value in next frame can be written as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta_i (X_t, \mu_{i,t}, \Sigma_{i,t})$$

K is the number of Gaussian distributions
Wi,t is an estimate of the weight of the ith Gaussian
µ is the mean
∑i,t is the covariance matrix of the ith Gaussian

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1}(X_t-\mu_t)}$$

- **Background Subtraction:** In this phase, a foreground mask is generated for every frame in the stream. Foreground mask is a binary mask where background = 0 and foreground =1.

For classifying a pixel as foreground or background, Gaussian distributions of every pixel are ordered by ratio in descending order ώ/σ. if a pixel matches any of the first B distributions it is classified as background pixel otherwise foreground pixel,

$$B = \arg\min_b \left(\sum_{i=1}^{b} \omega_{i,t} > T \right)$$

If a pixel is classified as background pixel then its colour value will be used in next frame. If a pixel is classified as foreground pixel then mean of the Gaussian distribution with lowest variance and largest weight is chosen as background pixel value.

- **Background Model Update for Future Purposes:** In this phase, parameters of the background model are updated for the generation of next background image. The Gaussian distribution is declared matched if Mahalanobi's distance,

$$\sqrt{((X_{t+1} - \mu_{i,t})^T . \Sigma_{i,t}^{-1} . (X_{t+1} - \mu_{i,t}))} < k\sigma_{i,t}$$

During updating of parameters, two cases can occur,

Case 1: Xt is matched with one of the K Gaussians.
For matched components, weight is increased, mean is brought closer to current pixel value and variance is decreased to make distribution more relevant.

$$\omega_{i,t+1} = (1-\alpha)\omega_{i,t} + \alpha$$

where α is a constant learning rate.

$$\mu_{i,t+1} = (1-\rho)\mu_{i,t} + \rho . X_{t+1}$$

$$\sigma_{i,t+1}^2 = (1-\rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t+1}).(X_{t+1} - \mu_{i,t+1})^T$$

where $\rho = \alpha . \eta(X_{t+1}, \mu_i, \Sigma_i)$.

Case 2: Xt doesn't match with any of the K Gaussians.
The parameters of least probable distribution k are updated as follows:

$$\omega_{k,t+1} = \text{Low Prior Weight}$$

$$\mu_{k,t+1} = X_{t+1}$$

$$\sigma_{k,t+1}^2 = \text{Large Initial Variance}$$

After updating the parameters, foreground detection can be made in subsequent frames.

- **Adaptive Mechanism:** The Gaussian Mixture Model will be met with more challenges in real life scenarios, such as, varying colour intensities depending upon the time of the day, introduction of new objects and removal of existing ones, and so on. We choose a time period T and at time t, we have history {X1, X2…Xt}. After each period of T, the Gaussian probability distribution function given by equation (2) is recomputed for every pixel. Thus, proposed approach can adapt to illumination changes and scene dynamics.

- **Shadow Detection and Processing:**

GMM based background subtraction is susceptible to formation of shadows which are often termed as foreground. These shadows distort the shape of moving objects and lead to misleading calculations for vehicular extraction in subsequent frames. Two methods of shadow removal have been used for the purpose of increasing efficiency, one for light traffic and the latter which has proven the most effective for scenarios comprising heavy traffic load.

Horprasert's computational color mode

The discrimination between expected RGB color of a pixel i,

$$E_i = [\mu_R(i), \mu_G(i), \mu_B(i)]$$

in the background image, and present RGB color value in current image,

$$X_i = [X_R(i), X_G(i), X_B(i)]$$

is measured. This discrimination is done by decomposing in two parts: brightness distortion and color distortion. Brightness distortion of a pixel i, denoted by ƛi , represents the fraction of remaining brightness with respect to expected value. Color distortion is defined as the orthogonal distance between observed color value and the expected color value. The color distortion of i th pixel is given by:

$$CD_i = \| X_i - \lambda_i E_i \|$$

Camera devices have different sensitivities for different color. So, in order to balance weights on the three RGB color channels, the pixel values are scaled and normalized by standard deviation. Normalized color value of a pixel i, denoted as siis given by:

$$s_i = [\mu_R(i)/\sigma_R(i), \mu_G(i)/\sigma_G(i), \mu_B(i)/\sigma_B(i)]$$

Hence, the brightness distortion and color distortion is given by:

$$\lambda_i = \min\left[\left(\frac{X_R(i)-\lambda_i\mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{X_G(i)-\lambda_i\mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{X_B(i)-\lambda_i\mu_B(i)}{\sigma_B(i)}\right)^2\right]$$

$$CD_i = \sqrt{\left(\frac{X_R(i)-\lambda_i\mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{X_G(i)-\lambda_i\mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{X_B(i)-\lambda_i\mu_B(i)}{\sigma_B(i)}\right)^2}$$

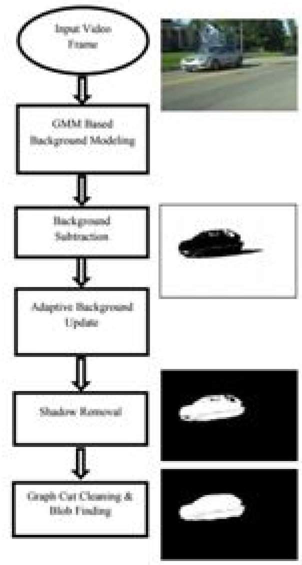
A pixel in the below image may be classified as follows:

$$X_i = \begin{cases} Shadow \quad CD_i < \beta_1 \ and \ \lambda_i < 1 \\ Highlight \quad CD_i < \beta_1 \ and \ \lambda_i > \beta_2 \end{cases}$$

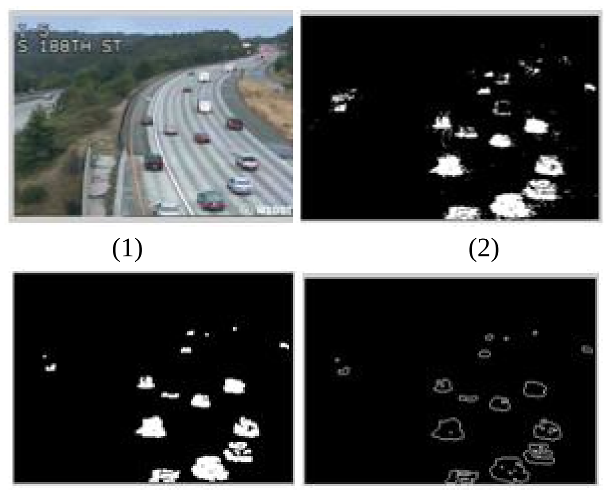
$\beta 1$ is a threshold to determine chromaticity difference between expected and current value. In order to avoid misclassification of a very dark background pixel as shadow, a normalized threshold is used.

$$\beta_2 = 1/(1-\varepsilon)$$

ε is a lower bound for the normalized brightness distortion.



Implementation screenshot for non-a congested scenario. Since the load is free, Gaussian Mixture Model is applied. Fig (1) shows the Input video frame, fig (2) is the result after applying GMM. The output of application of morphological gradient is shown in fig (3) and fig (4) is the result of moving edge extraction using Canny operator.



(1) (2)

(3) (4)

B)Small Range Updating

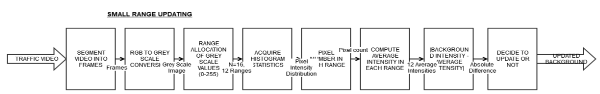
In this method, the grey scale values (0-255) of the background intensities are equally divided into N small ranges. The count of each range denotes the statistics of the histogram. The statistic value Sn(m) of the mth pixel in the nth small range is calculated using,

$$S_n(m) = \begin{cases} S_n(m)+1, & \text{if } I(m) \in \left[\frac{n\times 256}{N}, \frac{(n+1)\times 256}{N}\right) \\ S_n(m), & \text{otherwise} \end{cases}$$

whereas the average grey value µn(m) in the nth small range is calculated using

$$\mu_n(m) = \frac{1}{S_n(m)} \sum_{i=0}^{S_n(m)-1} I_i(m)$$

where the intensity of the mth pixel in the current frame is I(m), and Ii(m) is the ith intensity in the nth range. This value determines whether the background model is updated or not. Small ranges are taken to ensure the foreground is not updated instead of background. Now, if the absolute difference between the background intensity and the average value, in each range, is lesser than the range itself, then the pixel is updated. Otherwise no update is done. Experimental results suggest the value of N=16 and the nth frame be 150. Thus, this approach is undertaken if the frame number is more than 90. The side effects are found to be low in this approach.



Implementation screenshot for a more congested scenario.since the load is heavy, small range updating is applied.

(1)Input video frame (2)After Small range updating

ALGORITHM:

1. Let N denotes the number of ranges in which the grey scale Values (0-255) are divided.
2. Initialize N=16
3. Compute the number of pixels in each range
4. Compute the average intensity value for each range
5. diff= abs—average value in each range - background intensity—
6. if diff is smaller than N
7. update background intensity to average intensity value
8. else
9. no update is performed

V. VEHICLE EXTRACTION USING BLOCK SEGMENTATION AND CANNY EDGE DETECTION

In this section, we propose a method to extract moving objects from the input image. We follow two steps- a foreground block wise segmentation and Canny edge detection. In Foreground block segmentation Sum of absolute difference between input and background image is used to confirm whether each block is in the foreground or not where Input image size is M x N and designated block size is D x D.

Algorithm: Foreground block segmentation

1. Image is divided into blocks of fixed size.
2. Sum of absolute difference between input and background image is used to confirm whether each block is in foreground or not.

Connected component labelling is then applied to identify the shapes of the moving blocks. the problem of visually connected vehicles is resolved by obtaining information from lanes and using them to mask the segmentation images, thus false-merged objects can be successfully separated. However, even after the block-based segmentation is performed, some tiny objects generated by noises should be removed. Thus, moving edge becomes an important approach in the proposed system.

In Moving Edge Detection, canny edge detection operator is applied to the image to obtain the edge features. Canny edge detection operator is a popular edge detection technique. This algorithm can be chosen for

- optimal detection without spuriousness in responses by optimal smoothing
- localization with minimal edge disposition by non-maximum suppression giving an output with thin lines of edge points at proper places
- single response to avoid confusion

Canny is a two step algorithm

1. Extract gradient using a Gaussian derivative.
2. Suppress non-maxima using two thresholds.

For obtaining gradient, we differentiate unit vectors Ux = [1,0] and Uy = [0,1]

$$\nabla g(x,y) = \frac{\partial g(x,y,\sigma)}{\partial x} U_x + \frac{\partial g(x,y,\sigma)}{\partial y} U_y$$
$$= -\frac{x}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} U_x - \frac{y}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} U_y$$

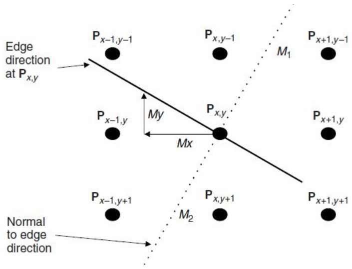
The maximum value of Gn, which is a first derivative of a Gaussian function g in the direction of the normal, n⊥, should be derived to obtain the peak of the edge data, where the gradient in the original image is sharpest (the location of the edge).

$$\frac{\partial^2 (G * \mathbf{P})}{\partial \mathbf{n}_\perp{}^2} = 0$$

Where

$$\mathbf{n}_{\perp} = \frac{\nabla(\mathbf{P} * g)}{|\nabla(\mathbf{P} * g)|}$$

This implies that we need values of gradient along a line that is normal to the edge at a point. The neighbouring points to the point of interest, Px,y, the edge direction at Px,y and the normal to the edge direction at Px,y .



Since we have a discrete neighbourhood, M1 and M2, maximum gradient, need to be interpolated, First order interpolation using Mxand My at Px,y are used to obtain this as

$$M_1 = \frac{My}{Mx} M(x+1,\ y-1) + \frac{Mx - My}{Mx} M(x,\ y-1)$$

$$M_2 = \frac{My}{Mx} M(x-1,\ y+1) + \frac{Mx - My}{Mx} M(x,\ y+1)$$

The implementation of non-maximum suppression first requires a function that generates the coordinates of the points between which the edge magnitude is interpolated.

```
get_coords(angle):= | δ←0.000000000000001
                    | x1←ceil[(cos(angle+π/8)·√2)−0.5−δ]
                    | y1←ceil[(−sin(angle−π/8)·√2)−0.5−δ]
                    | x2←ceil[(cos(angle−π/8)·√2)−0.5−δ]
                    | y2←ceil[(−sin(angle−π/8)·√2)−0.5−δ]
                    | (x1 y1 x2 y2)
```

The non-maximum suppression operator, then interpolates the edge magnitude at the two points either side of the normal to the edge direction. If the edge magnitude at the point of interest exceeds these two then it is retained, otherwise it is discarded.

```
non_max(edges):= | for i∈1..cols(edges_0,0)−2
                 |   for j∈1..rows(edges_0,0)−2
                 |     | Mx←(edges_0,0)_j,i
                 |     | My←(edges_0,1)_j,i
                 |     | o←atan(Mx/My) if My ≠ 0
                 |     | (o←π/2) if (My=0)·(Mx>0)
                 |     | o←−π/2 otherwise
                 |     | adds←get_coords(o)
                 |     | M1←[My·(edges_0,2)_j+adds_0,1, i+adds_0,0 ...
                 |     |      +(Mx−My)·(edges_0,2)_j+adds_0,3, i+adds_0,2]
                 |     | adds←get_coords(o+π)
                 |     | M2←[My·(edges_0,2)_j+adds_0,1, i+adds_0,0 ...
                 |     |      +(Mx−My)·(edges_0,2)_j+adds_0,3, i+adds_0,2]
                 |     | isbigger←[[Mx·(edges_0,2)_j,i>M1]·[Mx.(edges_0,2)_j,i≥M2]] ...
                 |     |          +[[Mx·(edges_0,2)_j,i<M1]·[Mx.(edges_0,2)_j,i≤M2]]
                 |     | new_edge_j,i←(edges_0,2)_j,i if isbigger
                 |     | new_edge_j,i←0 otherwise
                 |
                 | new_edge
```

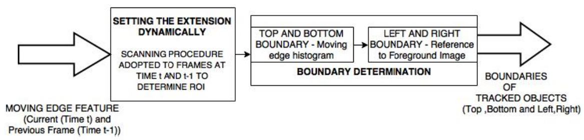
VI.VEHICLE TRACKING USING FUZZY LOGIC

The surveillance system requires a tracking procedure to identify the same vehicle in consecutive frames.
This module again consists of 3 parts

- Searching for the tracked vehicles
- Tracking Verification by Fuzzy Hybrid Detection Mechanism
- DE fuzzification and Error Compensation.
- Under congested conditions, incorrect vehicle boundaries may be detected due to the visually connected vehicles in the foreground image. Thus, the moving edge is used to determine the top and bottom boundaries of vehicles.
- In feature searching, we set the extension and determine the boundaries. Setting the extension will Dynamically determine the extension of moving-edge image in the region of interest namely the top left and bottom right positions and Boundary Determination uses Scanning procedures to determine the overall boundary of the tracked objects. The left and right boundaries are searched by reference to the foreground image. The height is determined based on the top and bottom boundaries. This information helps us identify the boundary of vehicles to be tracked. The steps involved are

1. Input Current Frame at Time t and previous frame at Time t-1 along with moving edge features.
2. Scan the movement across the two frames (current frame and previous frame) to determine the ROI.
3. Top and Bottom boundary determination: Vertically Scan the Moving edge histogram to find extension ().
4. Based on Top and Bottom Boundary, Height of rectangle is determined.
5. Left and Right Boundaries are determined by scanning about the Foreground Image.

6. Boundaries of Tracked vehicles are determined.



To verify the tracking results in the current and previous frame, we used an FHDM. Two features are utilized in FHDM, i.e., the colour similarity and the area consistency.

To obtain Colour Similarity, Colour intensities histogram distribution for all the colours in tracked regions are compared. This is followed by Colour similarity measurement (d) is computed. Colour similarity measurement (d) is computed using below formula where HA(i) is the histogram value of search result at time t and HB(i) is the histogram value of tracking value at time t-1.

$$\delta = \frac{\sum_{i=0}^{32767} Min\left(\hat{H}_A(i), \hat{H}_B(i)\right)}{\sum_{i=0}^{32767} Max\left(\hat{H}_A(i), \hat{H}_B(i)\right)}, \ 0 \le \delta \le 1.$$

To obtain Area Consistency, correct tracking is achieved when there is a relationship between time t, t-1, t-2. Search and tracking region will overlap if search is successful. Compute Overlap Ratio R which is the ratio of Search and Tracking results.

A1 - Searched region at time t
A2 - Tracking region at time t-1
A3 - Tracking region at time t-2

$$R = \frac{A_{A_1 \cap A_3}}{A_{A_1 \cap A_2}}.$$

Fuzzy logic inference engine uses the above values of Colour similarity and area consistency to verify the tracking. Three linguistic input variables (Low, Medium, High) in terms of degrees of colour and area comparisons are given as input to fuzzy inference system. Set of Membership functions and inference rules produce the fuzzy output. Five linguistic output variables (Different, Unlike, Normal, Like, Same) which will be the output of Fuzzy system represent the correctness of tracking. The fuzzy inference rule table is

	High	Medium	Low
High	Same	Like	Unlike
Medium	Like	Normal	Unlike
Low	Unlike	Unlike	Different

The output degree is formulated using the fuzzy sets and fuzzy rules. The process of Defuzzification helps convert fuzzy information to human inferable results. We use the Centre-of-sum method for Defuzzification. Center of sums is the fastest Defuzzification method. This process involves the algebraic sum of individual outputfuzzy sets. The algebraic equation for Defuzzified value is given by the equation:

$$z^{\chi} = \frac{\int_z z \sum_{k=1}^{n} \mu_{c_k} z \, dz}{\int_z \sum_{k=1}^{n} \mu_{c_k} z \, dz}$$

µCz represents areas of the respective membership functions

The tracking procedure does not easily lose its target under non-congested conditions. But some unpredictable conditions still occur, where the inference results may differ under congested conditions.

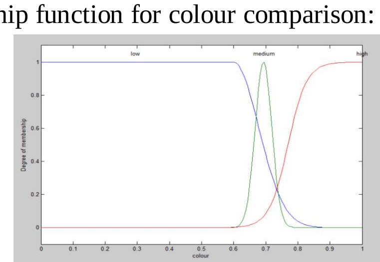
We propose a mechanism wherein a Standard absolute difference is computed for each position in ROI to refine the tracking. The mask is used to eliminate the tracked objects from the foreground image. Now the untracked objects will be correctly labelled by connected component labelling.
The FHDM algorithm has the following steps

1. Accept input in precise form.
2. Fuzzification: Input value is converted into Fuzzy input value with help of suitable membership function.
3. detection Mechanism (Fuzzy Reasoning): Defines different type of Fuzzy Rules in the form of If-Then for performance evaluation devised based on the criteria.
4. Fuzzy Output: Clips the output variable Fuzzy Set for each active rule invoked from the rule base there by generating the Clipped Fuzzy Sets.
5. De fuzzification (Performances): Compute the final output (Performance Value) with the help suitable defuzzification method.

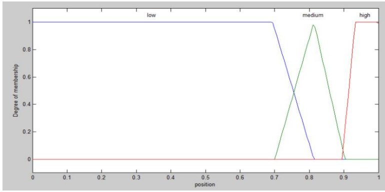
VII.EXPERIMENTAL EVALUATION

EXPERIMENTAL RESULT:

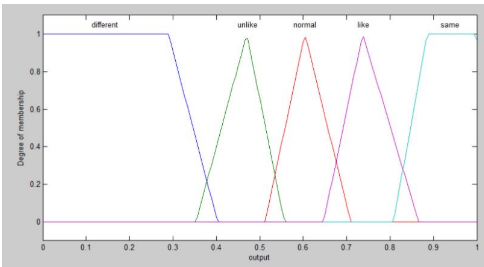
1. Membership function for colour comparison:



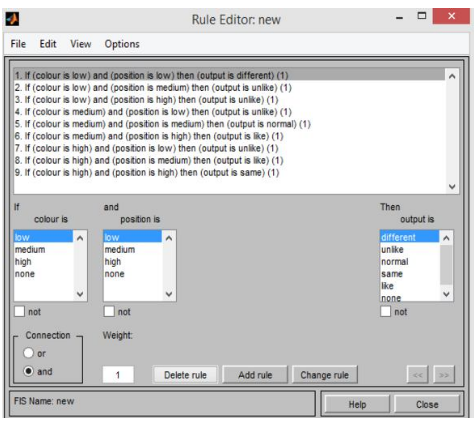
2. Membership function for position comparison:



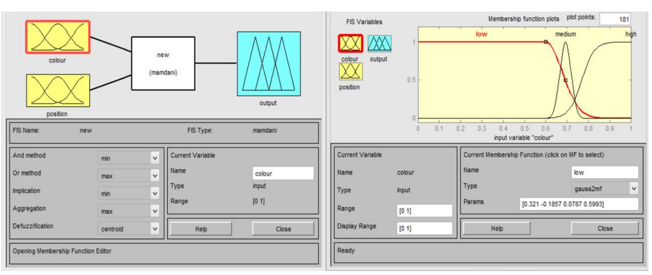
3. Membership function for output:



4. Fuzzy rules:



5. Fuzzy engine:



6. Vehicle tracking in Day time scenario with fewer no of vehicles along with vehicle count:

7. Vehicle tracking in day time scenario with increased no of vehicles:

8. Vehicle tracking in rainy scenario and vehicle count Generation:

9. Vehicle tracking in late evening scenario:

B) PERFORMANCE EVALUATION:

The performance of proposed system can be evaluated using traffic surveillance videos. Four scenarios are used to evaluate the system -daytime, rainy, congested and late evening traffic. In the quantitative evaluations of the proposed system, the Recall and Precision are considered.

Recall =TP/ (TP + FN)

Precision =TP/ (TP + FP)

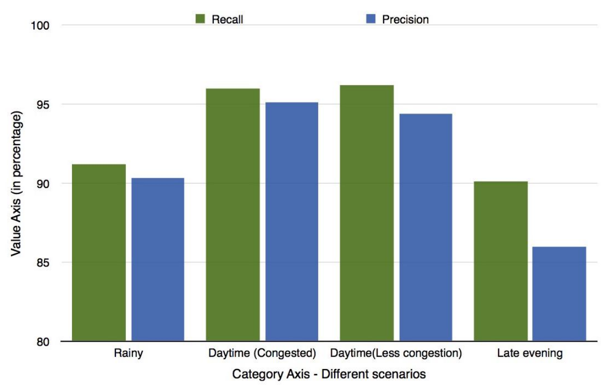
Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant. Recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. where TP (true positives) - the number of correctly detected vehicles FP (false positives) -the number of falsely detected vehicles FN (false negatives)- the number of missing vehicles.

These are used to evaluate the information retrieval performance. The accuracy of tracking can be calculated using the formula
Accuracy = TP / (TP + FP + FN)

In the situational evaluation, analysis in different road conditions is carried out.

Table displaying the results of experimental analysis under various scenarios:

Scenario	True Positive (TP)	False Positive (FP)	False Negative (FN)	Precision	Recall
Rainy	94	9	10	90.3	91.2
Daytime (Congested)	98	4	5	95.1	96
Daytime (less congested)	51	2	4	94.4	96.2
Late evening	110	12	18	86	90.1



(Fig)Graph displaying results of recall,precision values in different scenarios

On analysing the graph and cumulative results for around ten cases under different scenarios the following can be observed.

- The precision value of tracking is lower in the case of late evening than when the sky was overcast and there was moisture in the air.
- The precision and recall value of daytime vehicle tracking is high compared to the other two scenarios be it under congested or less congested conditions.
- The recall value of tracking in late evening and rainy scenario are almost the same. However, by a small margin we can say that system works better under rainy scenario.

VIII. CONCLUSION

The proposed system is a new approach to a video based traffic surveillance that overcame the limitations of congested traffic scenarios by using FHIIM. Firstly, the background has been updated smoothly using Gaussian mixture model along with shadow removal. For congested conditions, we have made use of small range updating. Next the block based segmentation has been used in place of traditional pixel based method. Then an edge feature method which makes use of Canny operator has been used to eliminate effects of headlights. FHDM is then used to compare the final decisions of tracked vehicles. Finally, error compensation has been used in case of any errors and thus used to improve the quality of tracking. Based on the experimental results, our proposed method works well under normal as well as congested conditions and gives reasonable accuracy for rainy days and late evening conditions. In the previous approaches, the quality of the captured images was very poor, and the setting height of the camera was quite low that it resulted in bad viewing angles and poorer detection. Thus, the experimental results have shown that the proposed approach had many outstanding performances in terms of detection and tracking.

REFERENCES

[1] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, "Data driven intelligent transportation systems: A survey," IEEE Trans. Intell.
Transp. Syst., vol. 12, no. 4, pp. 1624–1639, Dec. 2011.
[2] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 3, pp. 920–939, Sep. 2011.
[3] N. K. Kanhere and S. T. Birchfield, "A taxonomy and analysis of camera calibration methods for traffic monitoring applications," IEEE Trans. Intell. Transp. Syst., vol. 11, no. 2, pp. 441–452, Jun. 2010.
[4] F. Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," IEEE Trans. Intell. Transp. Syst., vol. 11, no. 3, pp. 630–638, Sep. 2010.

[5] L. W. Tsai, J. W. Hsieh, and K. C. Fan, “Vehicle detection using normalized color and edge map,” IEEE Trans. Image Process., vol. 16, no. 3,pp. 850–864, Mar. 2007.
[6] N. K. Kanhere and S. T. Birchfield, “Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features,” IEEE Trans. Intell. Transp. Syst., vol. 9, no. 1, pp. 148–160, Mar. 2008.
[7] Y. Wang, “Joint random field model for all-weather moving vehicle detection,” IEEE Trans. Image Process., vol. 19, no. 9, pp. 2491–2501, Sep. 2010.
[8] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, “Detection and classification of highway lanes using vehicle motion trajectories,” IEEE Trans. Intell.Transp. Syst., vol. 7, no. 2, pp. 188–200, June. 2006.
[9] E. A. Mosabbeb, M. Sadeghi, M. Fathy, and M. Bahekmat, “A lowcost strong shadow-based segmentation approach for vehicle tracking in congested traffic scenes,” in Proc. IEEE Int. Conf. Mach. Vis., 2007, pp. 147–152.
[10] W. Zhang, Q. M. J. Wu, G. Wang, and X.You, “Tracking and pairing vehicle headlight in night scenes,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1, pp. 140–152, Mar. 2012.
[11] Y. L. Chen, B. F. Wu, H. Y. Huang, and C. J. Fan, “A real-time vision system for nighttime vehicle detection and traffic surveillance,” IEEE Trans. Ind. Electron., vol. 58, no. 5, pp. 2030–2044, May 2011.
[12] H. Y. Cheng and S. H. Hsu, “Intelligent highway traffic surveillance with self-diagnosis abilities,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1462–1472, Dec. 2011.
[13] S. C. Chen, M. L. Shyu, S. Peeta, and C. Zhang, “Learning-based spatiotemporal vehicle tracking and indexing for transportation multimedia database systems,” IEEE Trans. Intell. Transp. Syst., vol. 4, no. 3, pp. 154–167, Sep. 2003.
[14] P. Kumar, S. Ranganath, W. Huang, and K. Sengupta, “Framework for real-time behavior interpretation from traffic video,” IEEE Trans. Intell. Transp. Syst., vol. 6, no. 1, pp. 43–53, Mar. 2005.
[15] J. W. Hsieh, S. H. Yu, Y. S. Chen, and W. F. Hu, “Automatic traffic surveillance for vehicle tracking and classification,” IEEE Trans. Intell. Transp. Syst., vol. 7, no. 2, pp. 175–187, June. 2006.
[16] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung, “A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images,” IEEE Trans. Intell. Transp. Syst., vol. 8, no. 3, pp. 441–459, Sep. 2007.
[17] A. Manzanera and J. C. Richefeu, “A new motion detection algorithm based on $\Sigma - \Delta$ background estimation,” Pattern Recognit. Lett., vol. 28, no. 3, pp. 320–328, Feb. 2007.
[18] I. Cohen and G. Medioni, “Detecting and tracking moving objects for video surveillance,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog., 1999, vol. 1, pp. 319–325.