Traffic surveillance: Anomaly detection using Image Processing

Progress Report

In Fulfilment of the requirements of

NU 302 R&D Project

NIIT UNIVERSITY

Submitted By

Siddharth Shashikar(U101114FCS186)

S.Shakthi(U101114FCS196)

Madarapu Srikar(U101114FCS085)

Hrithik Puri(U101114FCS069)

Certificate

This is to certify that the present research work entitled "Anomaly Detection Using Image Processing" being submitted to NIIT University, Neemrana, Rajasthan, in the fulfilment of the requirements for the course at NIIT University, Neemrana, embodies authentic and faithful record of original research carried out by Madarapu Srikar, S.Shakthi, Siddharth Shashikar and Hrithik Puri, students of B Tech (CSE) at NIIT University, Neemrana. They have worked under our supervision and that the matter embodied in this project work has not been submitted, in part or full, as a project report for any course of NIIT University, Neemrana or any other university.

Prof. Vikas Upadhyaya

ACKNOWLEDGEMENT

We would like to extend our gratitude towards our mentor Mr. Vikas Upadhyaya under whom it was a pleasure working with and a great learning experience.

We would like to add few heartfelt words for the people who were the part of the project in various ways, people who gave unending support, right from the beginning.

Contents

List of Figures

List of Tables

Abstract

There is a huge increase in the use of automobiles on roads and highways which has in turn increased the concern to monitor, regulate the traffic on roads and highways. This has led to proposal of new and efficient traffic surveillance systems.

This research completely focusses on detecting direction anomalies in the traffic surveillance videos and the challenge was to achieve all this using basic Image processing techniques and algorithms, and not compromising on the desired output. The entire processing of direction detection and direction anomaly detection of vehicles is further broken down to six steps which are as follows,

1. Frame acquisition
2. Background subtraction
3. Morphological operations
4. Vehicle detection and tracking
5. Vehicle direction detection
6. Direction anomaly detection

All the algorithms used are highly region based and do the necessary processing only on the region of interest and only as and when required thus saving processing power and time.

Rationale of Work

Traffic management system are a necessity in today's world where the number of vehicles on the road is exponentially increasing every day. Traffic management has always been the area of focus for all the countries. The reason to come up with such ideas is to equip the human traffic policing with systems that are capable of detecting any anomaly that occur during the normal day traffic surveillance. With these systems traffic policing would become uncomplicated and would in-turn increase the efficiency, accuracy and would also save manpower.

Even now in many places around the world, traffic policing is done manually in which the police patrol the highways which is laborious. This is where these traffic surveillance systems have an edge over the manual methods. Using these systems one can determine the direction of various number of vehicles in the frame and also can notify when any anomaly occurs. These systems prove to be efficient and can overcome many drawbacks that were in the manual method.

1. Wastage of manpower
    a. For determining direction of vehicle
    b. To check for direction anomaly
2. Inefficient traffic patrolling due to human error

All these drawbacks led to the search of an alternative which would be cost and performance efficient. Image processing has been used for the past decade to come up with solutions to various issues related with traffic surveillance. This research delivers a solution that how on using a live feed from traffic cameras we can determine the direction as well as anomaly in direction of a vehicle.

Literature Review

Following terms have been used in this report and hence defined,

Frame Differencing: Frame differencing is a technique in which a computer checks the difference between two video frames. Any change in the pixels indicates a change in the image.

Background Subtraction: Background subtraction is a type of frame differencing in which the one frame,that does not contain object of interest is saved and is subtracted from all subsequent video frames. The difference between them indicates any change in the frame.

Thresholding: The resultantimage obtained from techniques like background subtraction are converted into binary images by this process. The intensity value is of every pixel of the difference frame is compared to a threshold value, it is defined as part of object if greater than the threshold value else it is considered as the part of the background.

Convex Hull: It is one of the morphological operations that is performed on the output binary image after background subtraction and thresholding. The convex hull of a binary shape can be visualized quite easily by imagining stretching an elastic band around the shape. The elastic band will follow the convex contours of the shape, but will "bridge" the concave contours. The resulting shape will have no concavities and contains the original shape.

Erosion: Erosion is a basic morphological operations on which all other morphological operations are based. The basic effect of the operator on binary image is to erode away the boundaries of regions of the foreground pixels. Thus the area of foreground pixels shrink in size.

Dilation: Dilation is another basic morphological operation. The basic effect on a binary image is to gradually enlarge the boundaries of regions of foreground pixels thus increasing the size of the foreground pixels.

Opening: Opening is a morphological operation which involves *Erosion* followed by *Dilation*. It is less destructive than Erosion alone, though it also tends to removes some foreground pixels from the edges of regions of foreground pixels. The exact operation is determined by the *Structuring element*. The effect of the operator is to preserve foreground regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels.

Closing: Closing is the process of performing *Dilation* followed by *Erosion*. It is less destructive than Dilation alone though it also tends to enlarge the boundaries of foreground regions in the image. The effect of this operation is to preserve background regions that have a similar shape to the *Structuring Element* or that can completely contain the structuring element, while eliminating all other regions of background pixels.

Hole Filling: A hole may be defined as a background region surrounded by a connected border of foreground pixels. The algorithm is based on set dilation, complementation and intersection for filling holes in an image.

Objectives

The main objective of the research was to detect any anomaly occurrence in the video feed from traffic surveillance cameras. This section discusses the methodology involved in the research which was further subdivided into four stages:

1. Vehicle detection
2. Vehicle tracking
3. Vehicle direction detection
4. Direction anomaly detection

Vehicle Detection

Object detection has been a field of research for a decade and many algorithms have been proposed. Vehicle detection in real time video is a challenging task. As any detection of a false object would lead to false alarm. We have used *Background Subtraction* in our research work for vehicle detection.

Vehicle Tracking

Object tracking is applied over the detected object in the previous stage. This process consists of *Object Segmentatio*n and apply a *bounding box* around the object and then tracking the object of interest and finding its centroid.

Vehicle Direction Detection

Direction of the vehicle is determined by subtracting the centroid value of the object from the previous frame to the centroid value in the present frame.

Direction anomaly detection

Anomaly in direction can be detected by determining the lane in which the vehicle is traveling and accordingly checking its centroid value.

Methodology

Detailed explanation of the process can be understood with the help of flow chart

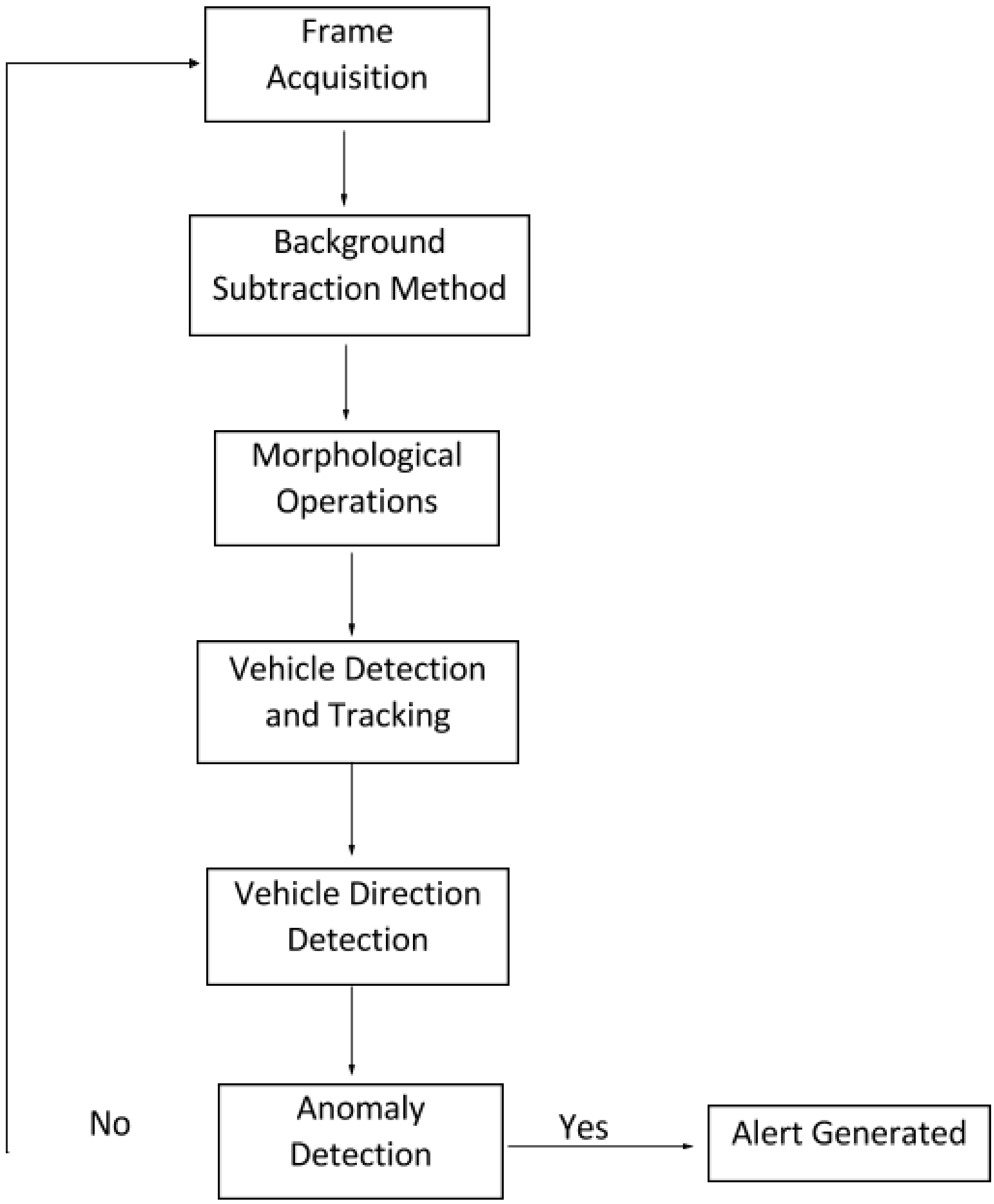

Figure 1 - Methodology Flow Chart

Object Detection

Background Subtraction

Background subtraction also known as the foreground detection. It is the core idea for various object detection algorithms. In background subtraction an image is fixed as the background image. Every frame of the video is then subtracted from this background image and the resulting image is then subjected to the thresholding process to get the final output as a logical image.

Mathematically,

If $|B(x,y) - F(x,y)| \geq Threshold$

$R(x,y) = 1$

else

$R(x,y) = 0$

Where, $B(x,y)$ = pixels of background frame

$F(x,y)$ = pixels of frame containing foreground image

$R(x,y)$ = resultant matrix

Drawbacks

One of the major drawbacks of background subtraction is that it fails to accommodate Illumination changes. As ambient intensity keeps on changing throughout the day.

Object Tracking

Vehicle Tracking can be divided into three sub-parts:

1. Zone Classification
2. Object Segmentation
3. Bounding Box and Centroid Extraction

Zone Classification

The entire frame is divided into three sub regions: **Entrance Zone**, **Main Area** and the **Exit Zone**. The **Entrance Zone** would be the region from where we would start tracking the blobs after applying the morphological operations on the object. The entrance zone is generally demarcated at such location from where the vehicles are having a considerable blob size, which will allow us to track them efficiently.

Exit Zone is the region where we stop tracking the vehicle as the blob size of the vehicle will be too small, which would lead to false detection. As we will not be able to distinguish between a vehicle and a cat.

Main Area is the region between the entrance zone and the exit zone. Here the vehicle blob is tracked and its subsequent centroid position in the next frame is known. Main area is the place where we analyse the blob thoroughly so as to confirm that they are indeed vehicles and not some other object which is not of our interest, and if found that it is not a vehicle according to our definition then it is not tracked any further.

Once these zones are classified the tracking of vehicle starts from the Entrance zone and tracked through the Main area and then tracking stops at Exit zone.

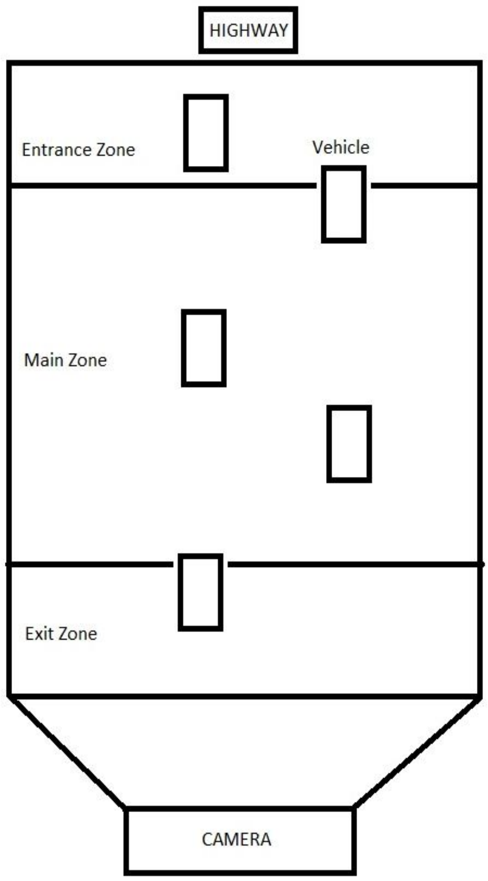

Figure 4 - Zone Classification

Object Segmentation

Object Segmentation is done on the resultant image that we obtained after Background subtraction. This method helps in keeping the parts of the same object, connected. If this process is not carried out, parts of the same object would be detected as multiple objects which is not desired. This process involves applying morphological operations on the resultant binary image.

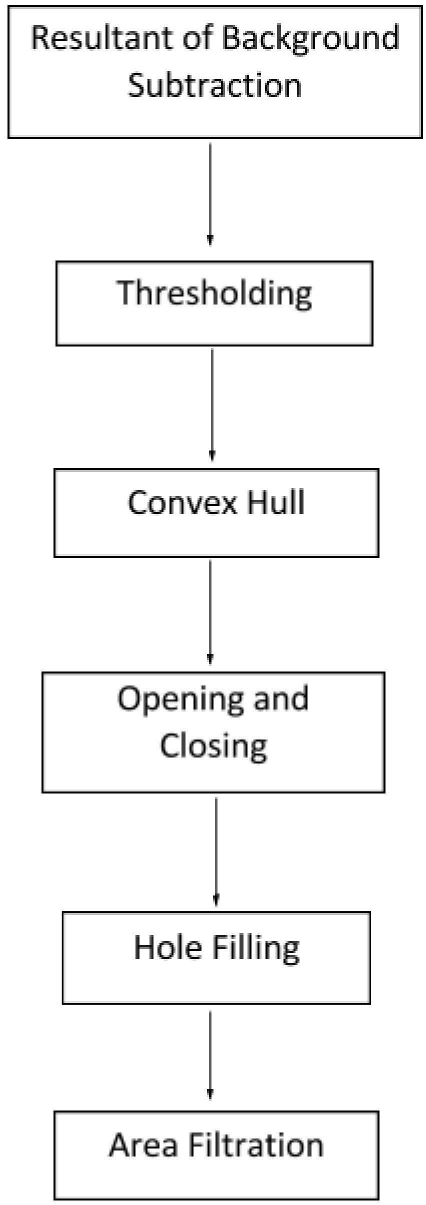

Figure 2 - Flow of Morphological Operations

In the project after *thresholding*, *Convex Hull* algorithm has been applied using the '*Disk'* structuring element. In convex hull, any two extreme object pixels are connected with a line and all the pixels on the line are turned into the object.

After *Convex Hull*, *Opening* and *Closing* operations are performed on the resultant image using the same structuring element. Subsequently it is followed by *Hole Filling*. The final blob obtained is a polygon shaped object. Area filtration helps in filtering out those small unwanted objects such as animals humans etc.

Bounding Box and Centroid Extraction

Once the object blob is obtained a bounding box is placed around the vehicle to highlight the object in the video and its centroid value is obtained which further helps in direction detection of the vehicles.

Vehicle Direction Detection

Direction detection can be done by tracking the coordinate values. The process can be explained with an example.

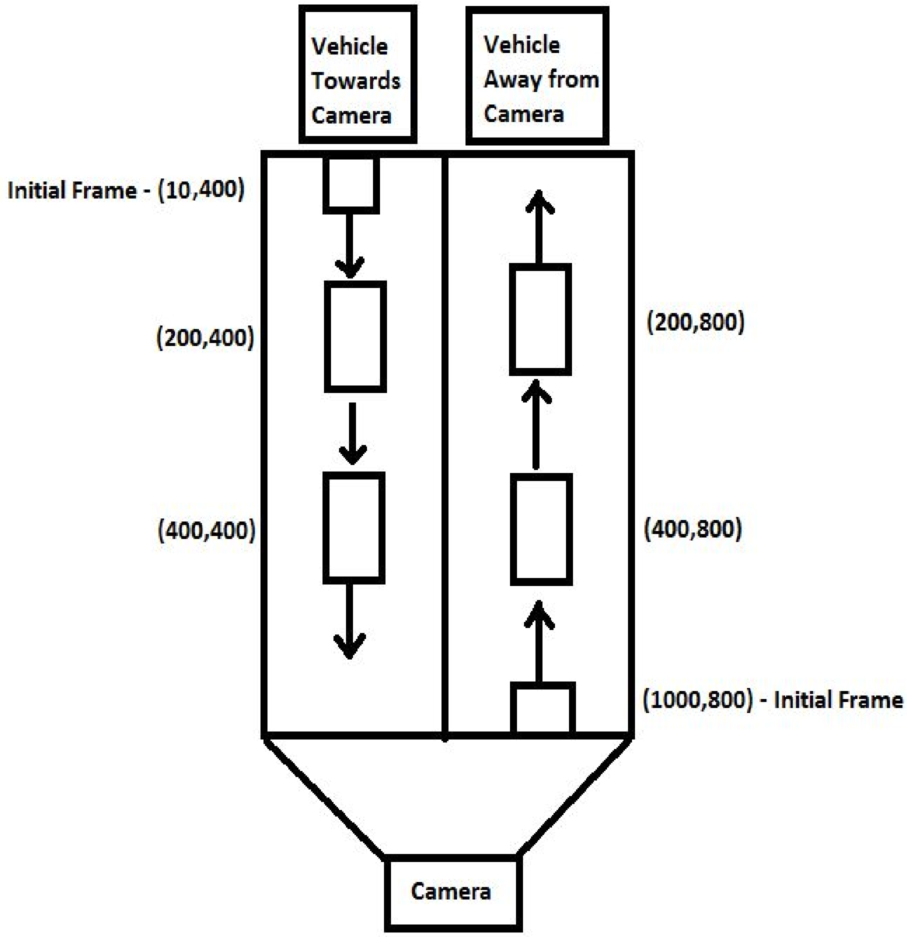

Figure 3 - Vehicle Direction Detection with Coordinates

In the project we **assumed** that vehicle on the left lane will move towards the camera and the vehicles on the right lane move away from the camera.

This table gives Centroid Coordinate from the above example –

Vehicle Direction	Background Frame Centroid Coordinates	50th frame Centroid Coordinates	Distance b/w BG frame & 50th Frame	100th frame Centroid Coordinates	Distance b/w 50th frame & 100th frame	Direction
Case-1	(10,400)	(200,400)	190	(400,400)	200	**Towards the Camera**
Case-2	(1000,800)	(400,800)	600	(200,800)	200	**Away from the Camera**

Table 1 - Direction Detection using Coordinates of Centroid

For case 1 let the object centroid be $(10, 400)$ as the object just enters the scene and for case-2 let the object centroid be $(1000, 800)$.

***NOTE - Please refer figure -3 for reference*

Case-1:

In 50th frame let the centroid of the detected object is $(200, 400)$ and in 100th frame centroid of the detected vehicle is $(400, 400)$.

Then the direction of vehicle can be determined by subtracting the x-coordinate of the initial centroid from the 50th frame centroid and subtracting x-coordinate of the 50th frame from the 100th frame centroid:

For 50th frame:

$$x_2 - x_1 = |200 - 10|$$

$$= 190$$

Where x_1 is x coordinate of centroid in background frame

x_2 is x coordinate of centroid in 50th frame

For 100th frame:

$$x_3 - x_2 = |400 - 200|$$

$$= 200$$

Where x_3 is x coordinate of centroid in 100th frame

x_2 is x coordinate of centroid in 50th frame

When the results are compared, it can be inferred that the **vehicle is moving towards the camera,** as the centroid distance is increasing frame by frame.

Case-2:

In 50th frame let the centroid value of the detected object be $(400, 800)$ and in 100th frame the centroid is $(200, 800)$.

Then the direction of vehicle can be determined by subtracting the x-coordinate of the initial centroid from the 50th frame centroid and subtracting x-coordinate of the 50th frame from the 100th frame centroid:

For 50th frame

$$x_2 - x_1 = |400 - 1000|$$

$$= 600$$

Where x_1 is x coordinate of centroid in background frame

x_2 is x coordinate of centroid in 50th frame

For 50th frame 00:

$$x_3 - x_2 = |200 - 400|$$

$$= 200$$

Where x_3 is x coordinate of centroid in 100th frame

x_2 is x coordinate of centroid in 50th frame

When the results are compared, it can be inferred that the **vehicle is moving away the camera** as centroid distance is decreasing frame by frame.

Anomaly Detection

Any deviation from the normal standards will cause an anomaly. Anomaly detection is a two step process which are as follows,

1. Lane Detection
2. Direction anomaly detection

Lane Detection

Lane Detection plays an important role while detecting the anomaly in vehicle direction. A basic preliminary method has been used in this project for lane detection. The Lane has been manually classified by dividing the road using the coordinate systems.

Anomaly in Direction of Vehicle

Any deviation from the normal standards fixed by the government would result in an anomaly. As per our assumption, the vehicles on the left lane must move towards the camera while in these cases the vehicle is moving away from camera and vice versa for the right lane. Anomaly in direction of the vehicle will be detected in following ways

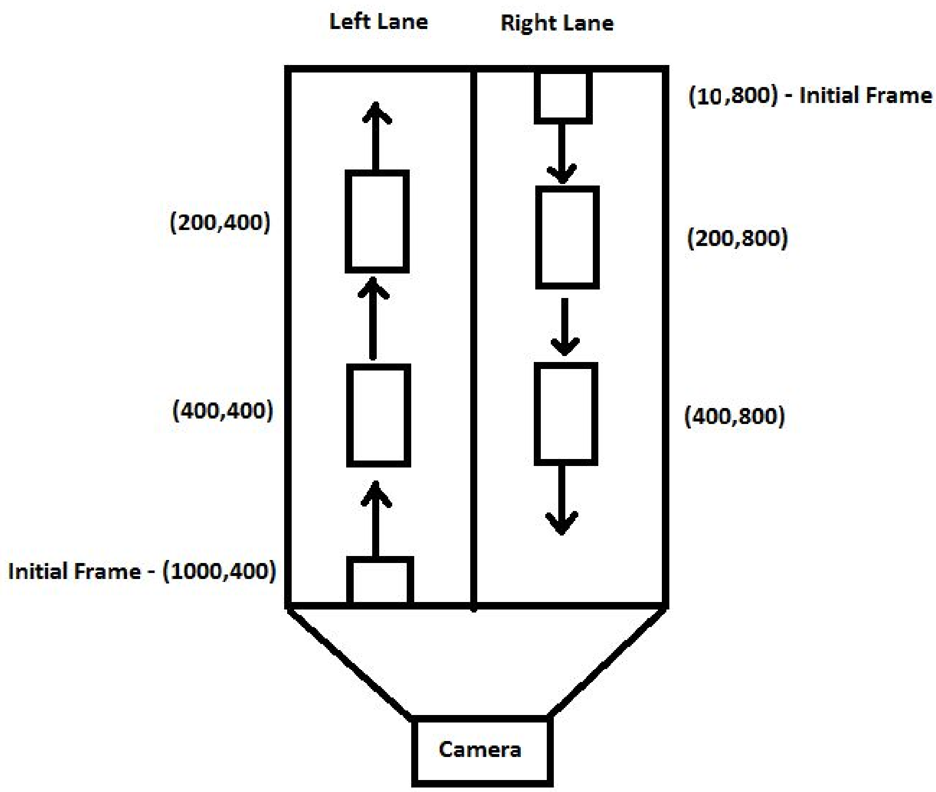

Figure 5 - Direction Anomaly Detection

This table gives Centroid Coordinate from the above example –

Vehicle Direction	Background Frame Centroid Coordinates	50th frame Centroid Coordinates	Distance b/w BG frame & 50th frame	100th frame Centroid Coordinates	Distance b/w 50th frame & 100th frame	Direction
Case-1	(1000,400)	(400,400)	600	(200,400)	200	**Away from Camera**
Case-2	(10,800)	(200,800)	190	(400,800)	200	**Towards the Camera**

Table 2 - Anomaly Detection in Direction using Coordinates of Centroid

The Background Image Centroids is $(1000, 400)$ in Case-1 and $(10, 800)$ for Case-2.

In **Case-1**:

In the 50th frame let the centroid value be $(400, 400)$ and in 100th frame let the centroid value be $(200, 400)$.

Then the direction of vehicle can be determined by subtracting the x-coordinate of the initial centroid from the 50th frame centroid and subtracting x-coordinate of the 50th frame from the 100th frame centroid:

For 50th frame:

$$x_2 - x_1 = |400 - 1000|$$

$$= 600$$

Where x_1 is x coordinate of centroid of background frame

x_2 is x coordinate of centroid of 50th frame

For 100th frame:

$$x_3 - x_2 = |200 - 400|$$

$$= 200$$

Where x_2 is x coordinate of centroid of 50th frame

x_3 is x coordinate of centroid of 100th frame

When the results are compared, **the vehicle is moving in the left lane but away from the camera .** In this case the distance obtained by subtracting the centroids is decreasing rather than increasing, which would be the case for a vehicle moving towards the camera in the left lane. So we can infer that there is a direction anomaly an alert message can be generated.

In **Case-2**:

In 50th frame let the centroid value is $(200, 800)$ and in 100th frame let the centroid value is $(400, 800)$ .Then the direction of vehicle can be determined by subtracting the x-coordinate of the initial centroid from the 50th frame centroid and subtracting x-coordinate of the 50th frame from the 100th frame centroid:

For 50th frame :

$$x_2 - x_1 = |200 - 10|$$

$$= 190$$

Where x_1 is x coordinate of centroid of background frame

x_2 is x coordinate of centroid of 50th frame

For 100th frame :

$$x_3 - x_2 = |400 - 200|$$

$$= 200$$

Where x_2 is x coordinate of centroid of 50th frame

x_3 is x coordinate of centroid of 100th frame

When the results are compared, **the vehicle is moving in the right lane but towards the camera .**In this case the distance obtained by subtracting the centroids is increasing rather than decreasing, which would be the case for a vehicle moving away from the camera in the right lane. So we can infer that there is a direction anomaly and an alert message can be generated.

.

Results

Object detection and tracking

The first step is extracting the reference frame from the video. Frame extraction is followed by *Background Subtraction*. For this we subtract the background frame from all the subsequent frames.

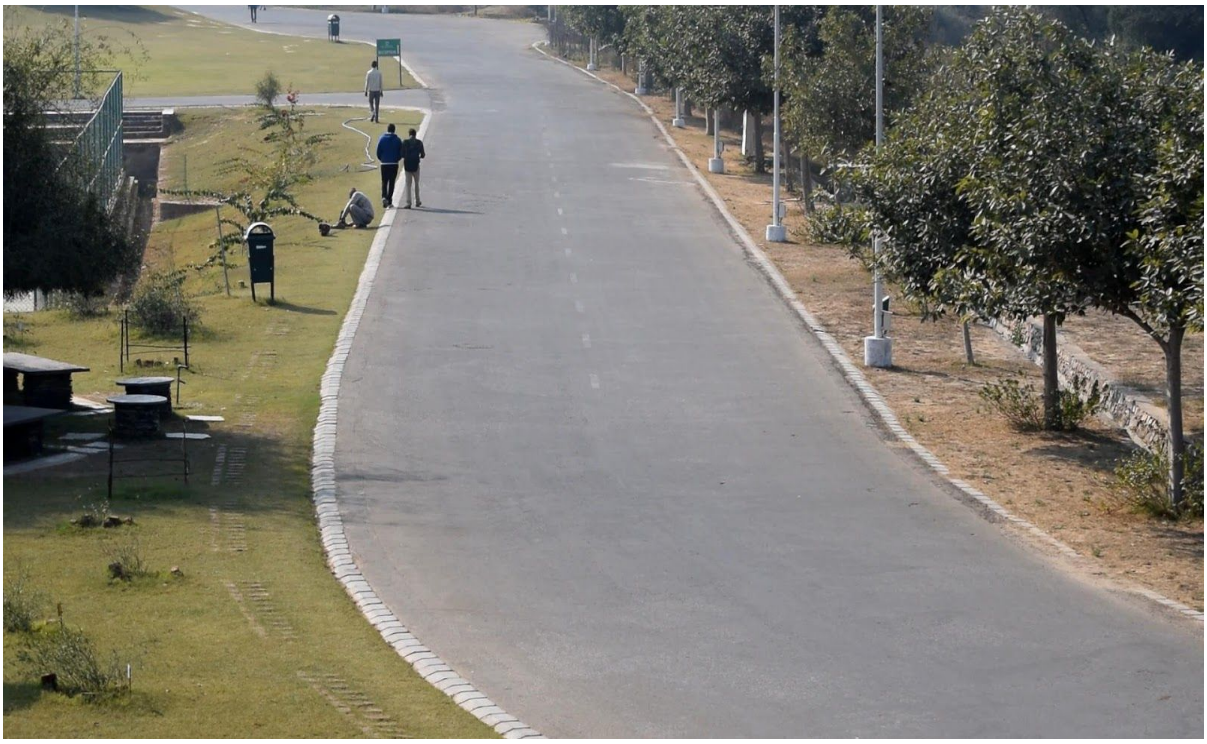

Figure 6 - Background Frame

Consecutive frames were subtracted from the above background frame. The next figure consists of the frame where we have our desired object i.e the Vehicle.

Figure 7- Frame with Vehicle

After subtracting the above frame from the background image we ended up with the following image.

Figure 8 - Result of Background Subtraction Method

After background subtraction, we did region based separation so that only the road is detected and the objects on the either side of the road is not be detected.

Figure 9 - Region based separation of Vehicle

After region based separation of the vehicle from area based filtration was done in order to remove those objects that do not qualify as an object of interest. We also performed some morphological operations on the resultant to have a complete structure of the vehicle. These are few of the morphological operations that we applied *Convex Hull, Opening, Closing, Hole-Filling.*

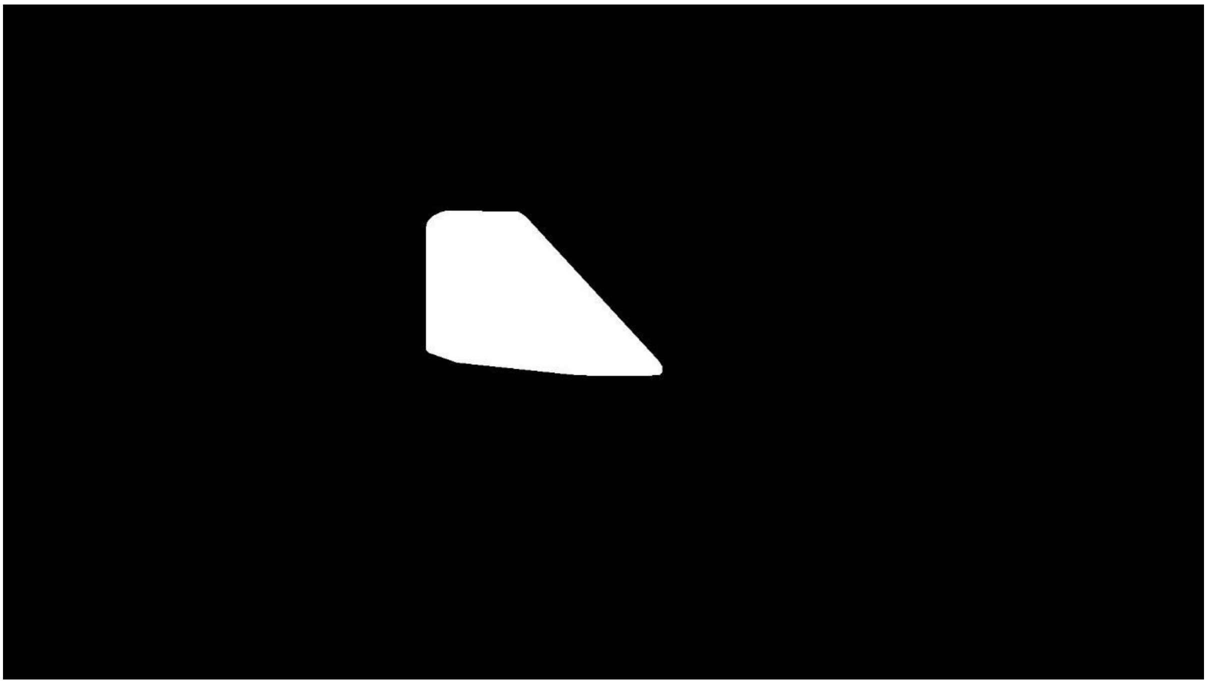

Figure 10 - Resultant after applying the Morphological Operations

After applying morphological operations the resultant looks like Figure-5. The next step is creating a *bounding box* around the vehicle for highlight the object of interest in the video and creating zones for the object to be tracked.

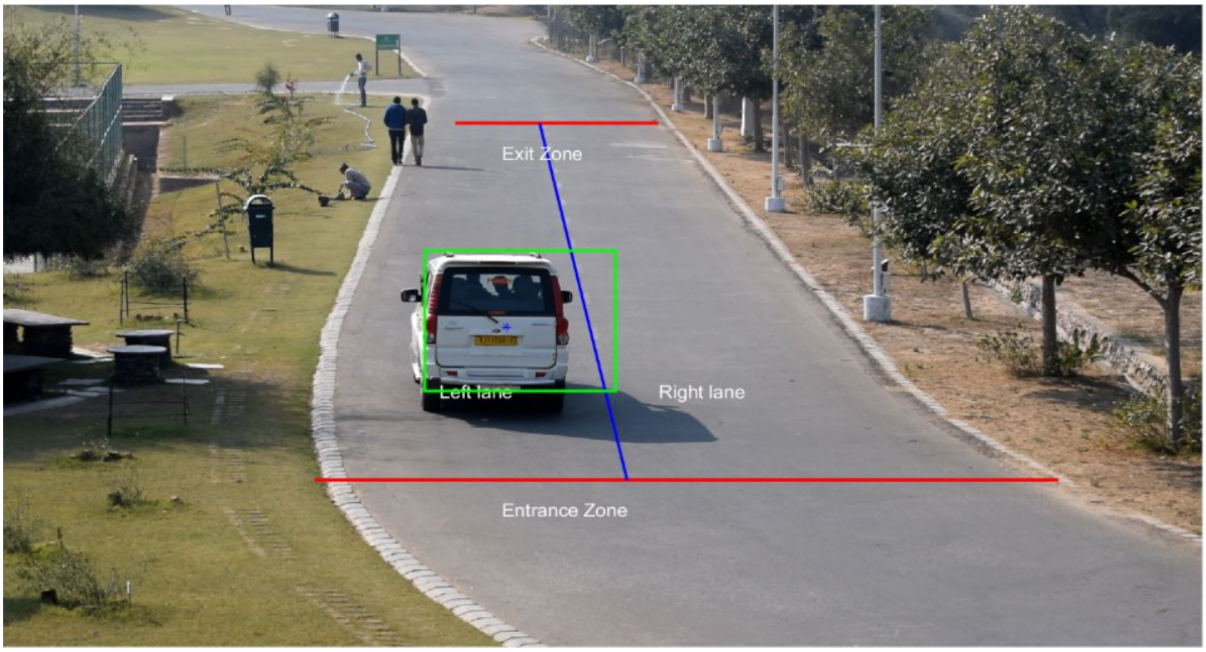

Figure 11 - Bounding Box around the Vehicle

Direction Detection

So in order to find the direction of the object the centroid values of the object were monitored, depending on the increasing and decreasing centroid distance values the direction of the vehicle was determined.

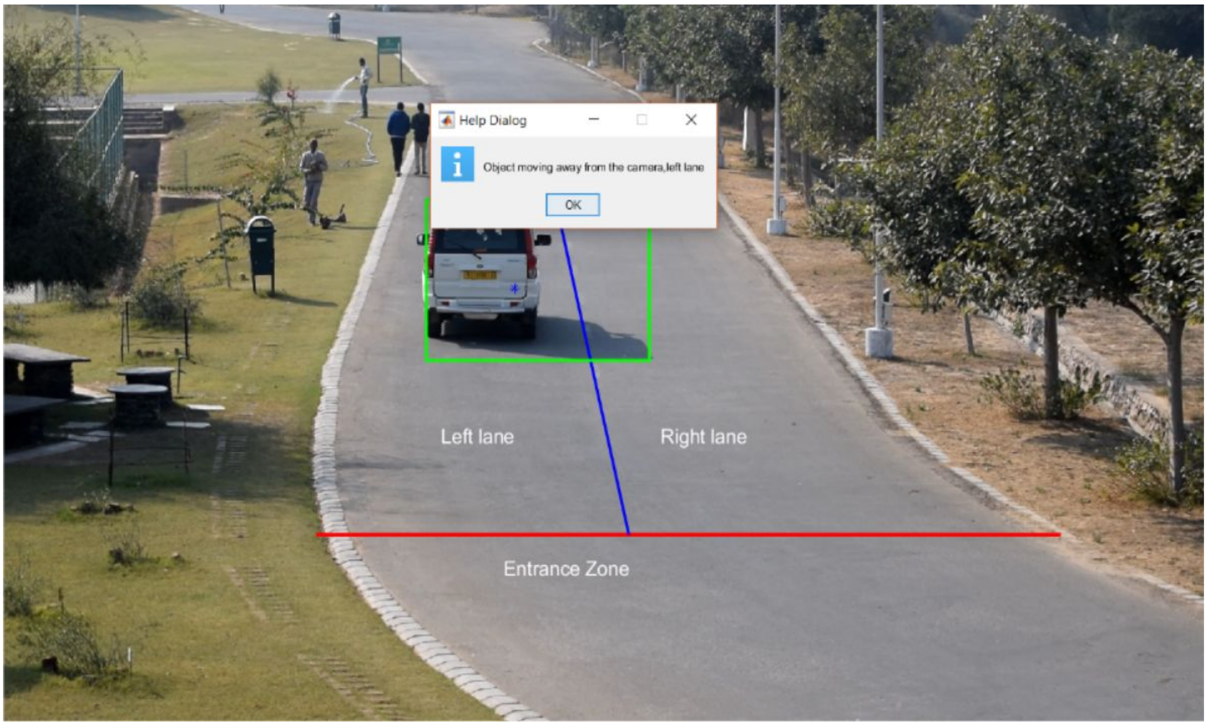

Figure 12 - Direction Detected for the Vehicle

Apart from that the lane in which the vehicle was moving was also determined by looking at the coordinate values of the detected object.

Direction Anomaly Detection

After direction detection, we check for direction anomaly. Depending on the behaviour of the centroid distances we can easily determine the direction, once we have the direction of the object. Anomaly detection is a easy process of matching the obtained direction from the standard assumption.

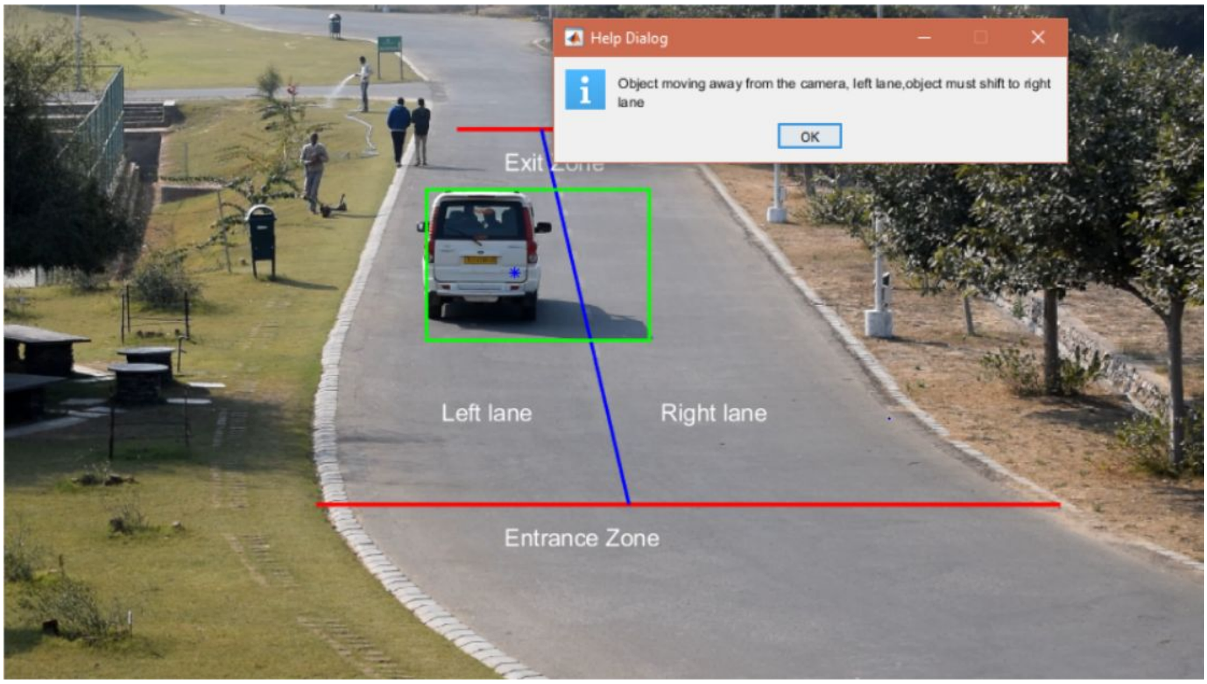

Figure 13 - Anomaly Detection in Direction of the Vehicle

Conclusion and Future Work

This research is a combination of various fields in the domain of vehicle detection and classification and occlusion handling, hence each every stage was dependent on the one before it hence one could only proceed to the next process by completing the previous one satisfactorily. This has also caused the research to strive for perfection in every stage so as to finally have a good result.

In future –

1. The focus shall be on implementing *Trajectory analysis and Support Vector Machines* for direction detection of vehicle and direction anomaly detection.

2. Currently the lane detection is done manually, it can further be detected using some advanced algorithms such as *Hough transform* which automates the entire process of lane detection.

3. In our work we have implemented Background Subtraction Method for vehicle detection. In future, the advancement would be applying Adaptive Background Subtraction Method to remove the drawbacks of Background Subtraction Method.

4. The current system fails to discriminate between the object and the shadow, in near future focus will be on implementing shadow removal techniques. So that the shadow doesn't play any role in the area detection of the vehicle.

References

[1] Speed Detection using Image Processing Pranith Kumar Thadagoppula , Vikas Upadhyaya Dept. of Electronics and Communication NIIT University.

[2] Detection and Classification of Highway Lanes Using Vehicle Motion Trajectories José Melo, Andrew Naftel, Member, IEEE, Alexandre Bernardino, Member, IEEE, and José Santos-Victor, Member, IEEE, accepted June 2, 2006.

[3] Video Based Vehicle Detection and Its Application in Intelligent Transportation Systems Naveen Chintalacheruvu, Venkatesan Muthukumar Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA , Received July 30, 2012; revised August 28, 2012; accepted September 7, 2012.

[4] Trajectory-Based Anomalous Event Detection Claudio Piciarelli, Member, IEEE, Christian Micheloni, Member, IEEE, and Gian Luca Foresti, Senior Member, IEEE. TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 18, NO. 11, accepted November 11 , 2008.

[5] A real-time computer vision system for vehicle tracking and traffic surveillance Benjamin Coifman Institute of Transportation Studies, University of California, Berkeley, CA, 94720, USA David Beymer Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1776, USA. , Philip McLauchlan , Jitendra Malik. accepted December 3,1998.

[6] A Survey on Object Detection and Tracking Methods Himani S. Parekh P.G.Student, Department of Computer Engineering, B.V.M. Engineering College, V. V. Nagar, India , Darshak G. Thakore Associate Professor, Department of Computer Engineering, B.V.M. Engineering College, V. V. Nagar, India , Udesang K. Jaliya .Assistant Professor, Department of Information Technology, B.V.M. Engineering College, V. V. Nagar, India .ISO 3297 accepted Feb 2 , 2014.

Appendix – Matlab Code

Direction Detection –

```
%  Steps to be followed
% 1) Read an input video and variable initialization.
% 2) Extract the first frame as the background frame.
% 3) Apply background subtraction on the subsequent video frame and some morphological operations.
% 4) Calculate the area of the detected object.
% 5) If object is of interest,track the motion of the object.
% 6) Find the centroid,add bounding box on the detected object.
% 7) Find the direction of the detected vehicle.
% 8) Once you have the direction of the vehicle check for the direction anomaly.
% 9) If there is a case of direction anomaly generate an alert message.

% Read an input video and variable initialization
source = VideoReader('Away-L20.mp4');
% source = VideoReader('Away-R20.mp4');
nFrames = source.NumberOfFrames;
threshold = 30;
n_centroid = zeros(1,2);
o_centroid = zeros(1,2);
o_cd = 0;
n_cd = 0;
divider_pos = 915;
```

```
% Extract the first frame.
mov(1).cdata = read(source,1);
ref_img = mov(1).cdata;
ref_img = rgb2gray(ref_img);
ref_img = double(ref_img);
[h,w] = size(ref_img);

% Getting the 75th frame,when the vehicle enters in the view of camera.
x = 75;
mov(x).cdata = read(source,x);
frame = mov(x).cdata;
frame = rgb2gray(frame);
frame_bw = double(frame);
frame_diff = frame_bw - ref_img;

for i = 1:h
        for j = 1:w
                if(frame_diff(i,j) < 0)
                        frame_diff(i,j) = -(frame_diff(i,j));
                end
        end
end

track_old = regionprops(frame_diff,'basic');
o_centroid = track_old.Centroid;
time = 1;
```

```
for x = 50:50:nFrames
%   Background subtraction and other morphological operations.
        mov(x).cdata = read(source,x);
        frame = mov(x).cdata;
        op = frame;
        frame = rgb2gray(frame);
        frame_bw = double(frame);
        frame_diff = frame_bw - ref_img;

        for i = 1:h
                for j = 1:w
                        if(frame_diff(i,j) < 0)
                                frame_diff(i,j) = -(frame_diff(i,j));
                        end
                end
        end
%   Thresholding
        for i = 1:h
                for j = 1:w
                        if(frame_diff(i,j) >= threshold)
                                if ((i >=150 && i <= 650) && (j <= 1350 && j >= 677))
                                        frame_diff(i,j) = 255;
                                else
                                        frame_diff(i,j) = 0;
                                end
                        else
                                frame_diff(i,j) = 0;
                        end
                  end
        end
```

```
% Morphological operations
        frame_diff = bwareaopen(frame_diff,2000);
        se = strel('disk',10);
        frame_diff = bwconvhull(frame_diff,'objects');
        frame_diff = imopen(frame_diff,se);
        frame_diff = imclose(frame_diff,se);
        frame_diff = imfill(frame_diff,'holes');
        frame_diff = bwareaopen(frame_diff,2000);
        track_objects = regionprops(frame_diff,'basic');
        n_centroid = cat(1,track_objects.Centroid);

% Direction detection and direction anomaly detection
        if(isempty(n_centroid) ~= 1)
                n_cd = n_centroid(2) - o_centroid(2);
                if (n_cd < o_cd && n_centroid(1) <= divider_pos)
                        message = sprintf ('Object moving away from the camera,left lane');
                elseif (n_cd < o_cd && n_centroid(1) > divider_pos)
                        message = sprintf ('Object moving away from the camera, right
                        lane,object must shift to left lane');
                elseif (n_cd > o_cd && n_centroid(1) <= divider_pos)
                        message = sprintf ('Object moving towards the camera,right
                        lane,object must shift to left lane');
                else
                        message = sprintf('Object moving towards the camera, left lane');
                end

                helpdlg(message);
        end
```

```
        v3 = n_centroid;
        imshow(op);

%  Marking different Zones and lanes on the road.
hold on
        plot([500,1690],[800,800],'LineWidth',2.5,'Color','red');
        plot([725,1050],[200,200],'LineWidth',2.5,'Color','red');
        p1 = [200,860];p2 = [800,1000];
        plot([p1(2),p2(2)],[p1(1),p2(1)],'LineWidth',2,'Color','blue');
        text(700,650,'Left lane','Color','white','FontSize',14)
        text(1050,650,'Right lane','Color','white','FontSize',14)
        text(800,250,'Exit Zone','Color','white','FontSize',14)
        text(800,850,'Entrance Zone','Color','white','FontSize',14)
        fontsize = 14;

% Applying bounding box around the vehicle.
hold on
        for index = 1:length(track_objects)
                box = track_objects(index).BoundingBox;
                plot(track_objects(index).Centroid(1),track_objects(index).Centroid(2),'b*');
                rectangle('position',box,'Edgecolor','green','LineWidth',2);
        end
        movegui(gcf);
        F(x) = getframe(figure(1));
hold off
o_cd = n_cd;
end
```