Vehicle Detection & Speed Tracking

Problem statement

Vehicle tracking is the process of locating a moving vehicle using a camera. Capture vehicle in video sequence from surveillance camera is demanding application to improve tracking performance. This technology is increasing the number of applications such as traffic control, traffic monitoring, traffic flow, security etc. The estimated cost using this technology will be very less. Video and image processing has been used for traffic surveillance, analysis and monitoring of traffic conditions in many cities and urban areas. Various methods for speed estimation are proposed in recent years. All approaches attempt to increase accuracy and decrease cost of hardware implementation. *The aim is to build an automatic system that can accurately localise and track the speed of any vehicles that appear in aerial video frames.*

Background

In recent years many researchers have worked on video cameras which are considered as a sensor device for capturing and recognizing moving vehicle. Video based systems can capture a large variety of information which is less expensive to install and maintain cameras. It is not easy for a single moving camera to quickly capture information. The main difficulty in vehicle video is the variation of environment ,and the headlights are important features for initializing and tracking vehicle at night time. It is capable of detecting vehicles and measures traffic parameters like speed, count, incidents etc. Cascade classification in Computer Vision is typically accomplished using AdaBoost. During the learning step in the AdaBoost algorithm, a weak classifier or feature value with the lowest classification error is selected in any particular iteration from positive and negative samples. A classification error is the probability of a non-vehicle to be classified as a vehicle and a vehicle to be classified as a non-vehicle, while a positive sample is an image that contains a vehicle and a negative sample is a non-vehicle image.

Methodology

The speed of the vehicle in each frame is calculated using the position of the vehicle in each frame, so the next step is to find the spots Bounding, and the centre of gravity. Bubble centroid distance is important to understand the moving vehicle in consecutive frames and therefore is known as the frame rate for motion capture, the speed calculation becomes possible. This information must be recorded in a continuous array cell in the same size as the camera image captured because the distance travelled by the centroid is needed. Specific coordinate will be used to determine the vehicle estimated speed, tracking, and distance travelled by the vehicle: Architecture of the Vehicle Detection & Speed Tracking is shown in Figure 1.

The tracking and speed estimation consists of the following steps:

Step 1: Use the binary image and segment it into groups of moving objects using the aforementioned shrinking algorithm to creates over region.

Step 2: Track each in consecutive frames and find its spatial bounding box coordinates, i.e., upper left side coordinate of the spatial bounding box (Yt, Xt) at time instant t.

Step 3: Trigger the timing ti when the object passes the first imaginary line located at Y_1, i.e., Yti < Y_1 and record its upper left side coordinate of the spatial bounding box, i.e., (Yti, Xti).

Step: 4 Trigger the timing ti when the object passes the first imaginary line located at Y_2, i.e. Yte < Y_2, and record its upper left side coordinate of the spatial bounding box, i.e. (Yte, Xte),

Step 5: Calculate the speed V as follows:

$$V = \frac{\Delta y}{\Delta x} = \frac{y_{te} - y_{ti}}{t_e - t_i}$$

Step 6: If the speed V is lower than the speed limit, then discard the object and go to Step 1.

Step 7: Extract the license plate using colour information.

Step 8: Transmit the extracted license plate image to authorized station.

Step 9: Go to Step 1.

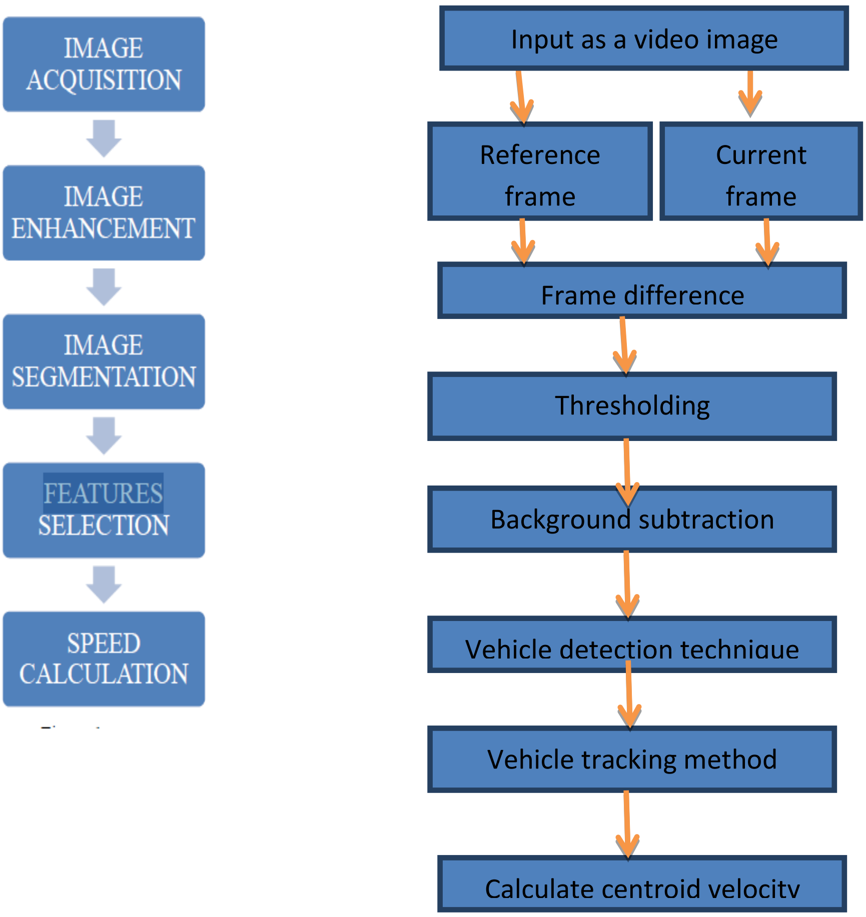

Figure 1: Architecture of the Vehicle Detection & Speed Tracking system

Experimental Design:

Dataset:
Video dataset of moving vehicles at highway/street road captured from the camera will used for experimentation and evaluation.

Evaluation measure:
Measures such as Area Under Curve, Recall Precision Curve, false positive, false negative, true positive, and true negative will be computed using datasets.

Hardware & Software Requirements:
Python based Computer Vision and Deep Learning libraries will be exploited for the development and experimentation of the project. Tools such as Anaconda Python, and libraries such as OpenCV etc. will be utilized for this process. Training will be conducted on NVIDIA GPUs using some object detection model and Raspberry Pi and Raspberry Pi Camera Module will be required for collected the real time data.