Traffic Saturation Detection using Hough Transform and VANET

Dekpeltakié Augustin METOUALE SOMDA [1], Abdoulaye SERE [2]

{somdaaugustin21@gmail.com[1], abdoulaye.sere@recifaso.org[2] }

Laboratoire d'Algèbre de Mathématiques Discrètes et Informatique (LAMDI), Université Nazi BONI, Burkina Faso

Equipe Signal, Image et Communications (SIC)[1,2]

Abstract. The main objective is to solve the problem of traffic congestion, using GPS (Global Positioning System) on board vehicles and servers in a control station, reducing the number of cameras to be deployed or additional hardware on the roads. This paper discusses the application of the Hough transformation method in VANET(Vehicular Ad-Hoc Network) to facilitate traffic congestion detection and monitoring. The VANET network allows sharing of safety information between vehicles to ensure the safety of road users. In response to the problem of traffic congestion and the increase in road accidents by vehicles, we designed and tested various traffic scenarios. We simulated the placement of RSUs (Road Side Units) in each scenario and analyzed the delay and packet delivery ratios (PDRs) in each scenario.

Keywords: Smart City, Hough Transform, Traffic Saturation, VANET

1 Introduction

Land transport plays a major role in our society, especially in cities where rush hour delays can have a significant impact on the organization of activities, the economy or the ecology. Traffic signals are an integral part of the traffic environment. They are designed to regulate the number of vehicles, provide specific information to road users or to warn off unforeseen road circumstances. The perception and rapid interpretation of traffic signs is crucial for the safety of the crucial to driver safety. Utilities responsible for maintaining traffic infrastructure install signs on posts at the side of roads, on highway lanes and in other locations so that they are easy to see without distracting the driver's attention from vehicle operation [1]. Sign pictograms are also designed and standardized according to a rule to maximize simplicity and distinctiveness. However, certain circumstances such as heavy visual clutter, unfavourable lighting or poor weather conditions can significantly impair the perception of signage. Purely physiological factors such as excitement, irritation or fatigue are known to further reduce a person's visual (human visual) concentration and can therefore be life-threatening, especially when driving at high speeds. Several research works have been dedicated to image processing and different techniques have been developed. We can cite the Hough Transform as an example which goal is to recognize an object in an image [2]. Much used in computer vision, in research laboratories and even in industries, the Hough transform is a little expensive in terms of execution time and memory during a detection [3]. Several efforts have been made in the literature to optimize it. In addition, its application to a collection of images is done sequentially, which results in a fairly high execution time. Traffic congestion is a problem in most large cities[2]. In this article, we study the Hough transform and the network protocols to be used in order to apply it to the solution of such a problem of traffic saturation by vehicles in order to optimize the resources dedicated to the management of road traffic. The main objective of this paper is to solve the problem of road traffic and server saturation in a control station. The challenge is to reduce the saturation of road traffic and the number of devices deployed in vehicles and on roads [4]. This paper proposes a method to detect traffic congestion, traffic signs, white lines on the roadways using Hough's transformation method and the communication protocol between vehicles in order to alert the drivers about the road condition. This paper is organized as follows: section 2 representing the preliminaries which describes the basic concepts related to the detection of straight lines, circles and rectangles using the Hough transform method and then VANET (Vehicular Ad-Hoc Network), followed by section 3 which presents the description of our method. Finally, section 4 concerns the illustrations.

2 Preliminaries

2.1 Hough Transform

Hough Transform converts lines in the two-dimensional image plane into points in the parameter plane that defines these lines, thus transforming the line detection problem into a simpler point detection problem. The principle is to take all pairs of pixels that belong to the image contour to build a two-dimensional histogram also called an "accumulator table", which will be used to record the lines found for each pair of pixels extracted. Any point in the image that is part of one of these lines adds a credit point to the histogram. The work of D. H. BALLARD [5] extends it to the detection of arbitrarily complex geometric shapes. In [6] and [7], we found a complete study of the Hough

transform. Indeed, in [7], which is more comprehensive, it is more generally defined as transform that allows to detect in images the presence of parametric curves belonging to a known family from a set of selected points called characteristic points. Several works related to the application methods of the Hough Transform have been published. Indeed, in [8] we find a summary of these methods. The Standard Hough Transform (SHT) [9, 10, 11, 12, 13, 14] is a classical method for detecting straight lines in an image. It has been used in several applications : road detection in satellite images by Geman and Jedynak [15], robot localization by Hoppenot et al. [16], robust barcode reading by Muniz et al. [17] and ship wake detection by Rey et al. [18] and Magli et al. [13]; there are also other applications of the method proposed by Kasturi et al. [9]. The Hough transform, also known as the standard Hough transform, can be summarized in three steps [8] : the calculation of the parameter values and their accumulation in the cells of the parameter space; the search for local maxima and the detection of the geometric shape using the position of the maximum of the accumulators. THS uses normal parameterization and the "1 to m" variant of TH in the context of right-handed detection. To a point in the image space of coordinates (x_i, y_i), the THS associates a sine curve in the parameter space defined by : $\rho = x_i\ cos\theta + y_i\ sin\theta; \forall(\rho, \theta)\varepsilon\omega$.
The search process can thus be optimized by prepending $\theta\varepsilon[0, \pi]$ and $-R \leq \rho \leq R$ with R the retinal radius. The disadvantage of the approach lies in the following facts : the process may take time and memory for storing the accumulators; the number of calculations depends on the number of input minutiae [8]. In addition the Standard Hough Transform cannot behave like a machine; applied directly to discrete line detection [19]. Several works have been carried out either to improve it or to extend it, including that of SERE et al. [19] which extends it to the detection of discrete lines. The extended standard Hough transform was introduced by SERE et al. [19] in 2013, it allows the detection of discrete lines which are seen as a sequence of pixels. To facilitate the reconstruction of discrete line into continuous line, they introduced two essential notions Dual and Preimage. The Dual is here equivalent to the Standard Hough Transform. Thus the dual of a point P in the image space is a sinusoid in the parameter space. This definition has been extended to objects as follows: for any object O, $dual(O) = \bigcup_{p_i \in O} dual(p_i)$ the relationship emerges from mathematical proofs which show that the computation of the dual of some regular objects (segment, triangle and square) can be reduced to the computation of the dual of some characteristic points. For example : The dual of a segment is the area bounded by the dual of its extremities. The dual of a triangle is the union of the dual of two of its adjoining sides. The dual of a square is the union of the dual of its diagonals. The notion of Preimage refers to the accumulator. Let S = $\{ P_1, P_2, ..., P_n \}$ be a set of n pixels. We define the Preimage of S as : $preimage(S) = \bigcap_{p_i \in S} Dual(p_i)$ In addition, Belaroussi et al. [20] proposed a geometric method using gradient orientation for vertical signage detection in

still images, regardless of their position and orientation. This method is fast and does not distinguish between circles and polygons with 4 or more sides. In [3], Garc à-Garrido and Al. studied how the Hough transform can be used for real-time detection of road signs. In cities, the number of roads and the distance register with the number of vehicles. The Hough transform is a tool for detecting parametric curves in images. It was proposed by Paul.V.C Hough in a patent filed in 1959 [21]. Since the 1980s, this transform has been used in many industrial fields, such as computer vision and image processing. It has become a more suitable solution to the problem of detecting straight lines, circles or any other parametric shape in images. However, the computation of TH requires a lot of processing time and storage space[8]. Therefore, several new algorithms have been proposed, such as probabilistic TH, random TH, hierarchical TH and incremental TH. Sere et al. in [22] established the double of a rectangle to detect numerical straight lines and then they applied the method for the detection of the saturation of road traffic [4].

2.2 Vehicular Ad-Hoc Networks

Vehicular ad hoc networking is an emerging technology that enables intelligent vehicle-to-vehicle communications and seamless Internet connectivity, which improves road safety, critical alerts, and access to comforts and entertainment. This technology integrates WLAN/cellular and Ad-Hoc networks to provide seamless connectivity. The VANET network turns each participating car into a wireless router or node, allowing cars located approximately 100 to 300 meters apart to connect and thus create a wide-range network. As cars move out of signal range and leave the network, other cars can join them, connecting vehicles to each other to create a mobile Internet. The goal of VANET is to ensure the safety of drivers and other road users, save space, reduce the total cost of parking, be environmentally friendly and provide higher throughput with faster operations. VANET is a broad topic of study that is used to implement many components of ITS. VANETs are a mixture of inter-vehicular communication (IVC) and road-vehicle communication (RVC). Communication in VANET can be facilitated in three ways : vehicle-to-vehicle (V2V) ; vehicle to infrastructure (V2I) and infrastructure-to-infrastructure (I2I). In this case, vehicles act as mobile nodes, while roadside units and infrastructure act as fixed nodes. Vehicular networks can be classified into three types of architectures [23], depending on the way users are able to share information: centralized, decentralized, or distributed and hybrid. Proactive routing conventions use standard separation vector steering methodologies (destination-sequenced distance vector steering (DSDV)) or connection-state steering techniques (optimized link-state routing convention (OLSR) and topology-biased reverse path routing (TBRPF)). They maintain and revise routing data to all hubs, including when the path is not in use. Route revisions are occasionally eected with little regard to system load, data transmission requirements and system size [24]. Reactive routing conventions, Dynamic Source Routing (DSR) and Ad hoc On-Demand Distance Vector (AODV), determine courses based on interest or need and retain only those courses that are currently in use, thereby reducing the system load when a subset of accessible courses is used and limiting the waste of data transfer capacity [21].

3 Method Description

3.1 Detecting a straight line on roads

The method of detecting lines on roads consists in locating white lines, the roads are marked by a solid white line (right) and by short line segments alternating with points (left). Figure 1 represents the different steps of image processing until the detection of lines on a road.

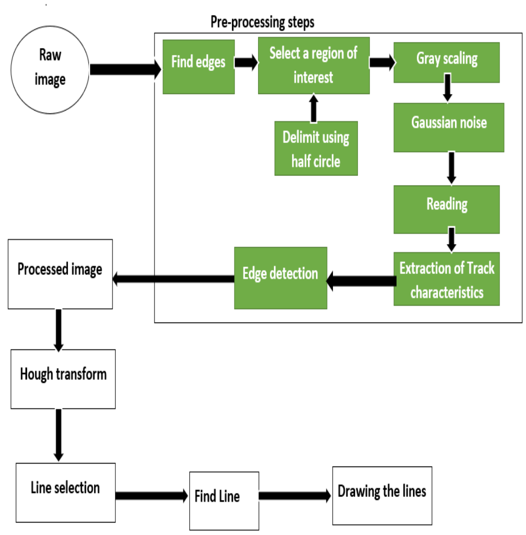

Fig. 1. Detection of lines

Algorithm 1 describes the process of detecting the lines on a given road. The different steps of line detection will be detailed in the next section.

3.1.1 Canny Edge Detection

It is important to note 2 steps before using Canny's method :
- noise reduction : The edges of an image are the regions where the intensity of the pixels changes. Edge detection is mainly based on gradient calculations, which is why the results are very sensitive

Algorithm 1: Algorithm for detecting lines on roads

```
Data : R_1,R_2,...,R_n a set of n routes containing pre-selected white
lines.
Data : I The raw image.
Data : B(I) Represents the edge detection function of the I image.
Data : m(J) represents the function of hiding the areas of the image
that do not contain lines of the road.
Data : θ definition of the angle.
Data : M = 0 initialization of the accumulator matrix
for all R_i(x,y) do do
  for all (x,y) ∈ M do do
    for all θ do do
      p = x * cos(θ) + y * sin(θ)
      M(p,θ) = M(p,θ) + 1
    end for
  end for
end for
ML = avg(I,M) Represents the average of the hough lines as left lane and
right lane.
C = DrawLine(I,ML) combine the average lines on the original image by
drawing the detected lines on the image.
```

to image noise. Thus, the first step is to apply a Gaussian blur on the image to smooth the rough edges. A 5x5 Gaussian kernel was used in our case. The size of the kernel can be modified to detect the best values (it must be positive and odd). Larger kernels imply more blur in the image and also require more time for processing. Therefore, smaller values are preferred if the effects are similar.
- grayscale conversion : The edge detection algorithm works on grayscale images because we are only interested in intensity gradients, so the input image must be converted to grayscale.
- Edge detection : Canny's edge detection function takes 3 main arguments, the input image and the edge intensity gradient thresholds, the threshold values decide which edges should be kept and which should be discarded. Edges with an intensity gradient greater than the maximum threshold are kept while those below the minimum threshold are discarded. Edges with values halfway are decided based on their connectivity, if they are related to a concermed edge, they are considered part of the edge, otherwise they are rejected. If the maximum threshold is very high, no edges will be found, while if it is too low, a large number of edges will be detected. In our case, we chose the values with minimum threshold equal to 50 and maximum threshold equal to 150.

3.1.2 Region of interest

All edges of the image are not useful for the task, which is to identify the traffic lanes on the road, as edges corresponding to the sky and trees, are not relevant, they should be removed. The region of interest should cover entirely and mainly the lane lines. A half circle is defined in the image to delimit the lane containing the lines in order to separate the lane from the part that are not part of the area of interest. Thus, the half circle must be cropped in the original image. All other parts of the image are excluded by applying a mask.

3.1.3 Hough transform for line detection

Using the Hough transform technique for extracting features to identify lines in an image. The problem of route detection requires identifying lines that cross all nearby edge pixels, from the edges detected in the region of interest. A transformation of the lines in Hough space allows to simply solve their intersections. We used the probabilistic Hough transform to generate Hough lines from an image with edge pixels.

3.1.4 Line averaging and extrapolation

The generated Hough lines indicate multiple lines on the same track. Thus, the average of these lines is used to represent a single line. In addition, some lines are partially detected. The average of the lines is extrapolated to cover the total length of the lanes. The process of using the average is done on the basis of the slopes of the multiple lines, which must be grouped together and belong to either the left or the right lane. In the image, the y-coordinate is reversed (since the origin is in the upper left corner), the value is highest when y is lower in the image. By this convention, the left corridor has a negative slope and the right corridor a positive slope. All lines with a positive slope are grouped together and averaged to get the right corridor, and vice versa for negative slopes to get the left corridor.

3.2 Traffic sign detection

The recognition of traffic signs is an aid to driving by indicating a speed limit or a prohibition to overtake. Figure 2 shows the various image processing steps for traffic sign detection and algorithm 2 describes the process of roadside sign detection.

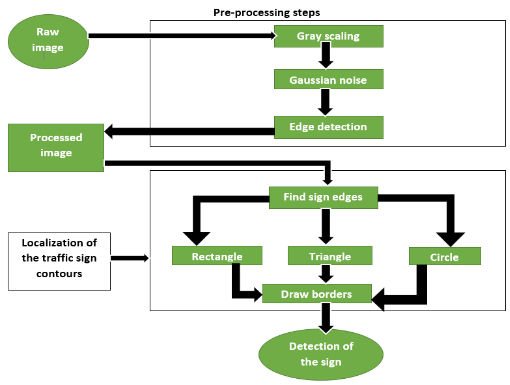

Fig. 2. The steps for detecting traffic signs

Algorithm 2: Traffic sign detection algorithm

Data:$R_1, R_2, \ldots, R_n$ a set of n roads containing preselected traffic signs.
Data:I, α and β
$contours = C(I)$
$contoursList = []$
$Maxlength = 0$
for all $R_i(x,y)$ **do**
  for all contour in contours **do**
    $perimeter = calculatesPerimeter(contour)$
    $approximately = ApproximateLength(contour, 0,015 * perimeter)$
    $Area = calculatesArea(contour)$
    if $Area > 100$ **then**
      $drawPanelsContour(I, contour, color)$
      if $approximately > \alpha$ AND $approximately < \beta$ **then**
        if $Area > \alpha$ Maxlength **then**
          $Maxlength = Area$
          $contourList[contour]$
        else if $Area == Maxlength$ **then**
          $contourList[contour]$
        end if
      end if
    end if
  end for
end for
$(x,y,w,h) = contourList[0]$
$drawPanels(I, (x,y), (x+w, y+h), color)$

I represents the raw image then α and β denote the approximate smallest and largest values respectively not to be exceeded for better traffic sign edge detection. C(I) is the edge detection function on image I, contoursList is used to store the list of detected traffic sign edges. the perimeter represents the length of the arc of the contour. Approximate is the approximate length of the contour with 10% of length and the Area which is the function for calculating the area of the traffic sign contour.

3.3 Saturation detection

In this section, we aim at solving the saturation problem on roads by monitoring vehicle traffic and road conditions, providing useful information about lane saturation. Saturation can be addressed either by increasing the capacity of the road network or by decreasing the demand in congested areas, especially during peak hours. One of the major challenges to reducing congestion on the road network is that it is difficult to change the mindset of drivers. Driving becomes an automatic and habitual process, and travelers often ignore alternative ways to get from A to B. The basic idea of the project will be to use real-time traffic updates for a period of time. Algorithm 3 describes the process of alerting about saturation on a given road proposed by SERE et al. [4].

Algorithm 3: Algorithm for saturation detection

```
Data :  R_1,R_2,...,R_n a set of n roads containing pre-selected vehicles.
Data :  V_1,V_2,...,V_m a set of m vehicles circulating on the roads.
Data :  u represents the number of lanes for the considered road.
Data :  S_i(u) is the saturation function.
Data :  γ_i is the maximum number of vehicles allowed on a given road.
Data :  β_i and α_i represent the thresholds for the saturation function
Data :  alert contains the alert message to the driver
for all R_i(x,y)  do
  while V_i(x_i,y_i)  do
    u = u+1
    S_i(u) = u ⟼ u/γ_i
    if S_i(u) > β_i THEN then
      allerte = ' to saturation'
    else if α_i < S_i(u) ≤ β_i THEN then
      allerte = 'near saturation '
    else
      allerte = ' not at saturation '
    end if
  end while
  u = u+1
end for
```

3.4 VANET Methodology

In this section we will show how to establish communication between vehicles and infrastructure and how to share information (about traffic signs, saturation and deviation of white lines on the lane) about road conditions between vehicles. In this methodology, we first initialized the VANET scenario by defining the number of vehicles in the reverse direction of their mobility, then we established the communication between the different vehicles and the roadside unit using the routing protocol for the communication process, finally we used the bandwidth of the cognitive radio for the transmission of packets from vehicle to vehicle, vehicle to RSU and RSU to vehicle through a sensing channel. The channel is free and can be allocated for communication.

3.5 Proposed System Model

A model of the VANET communication architecture is shown in figure 3.

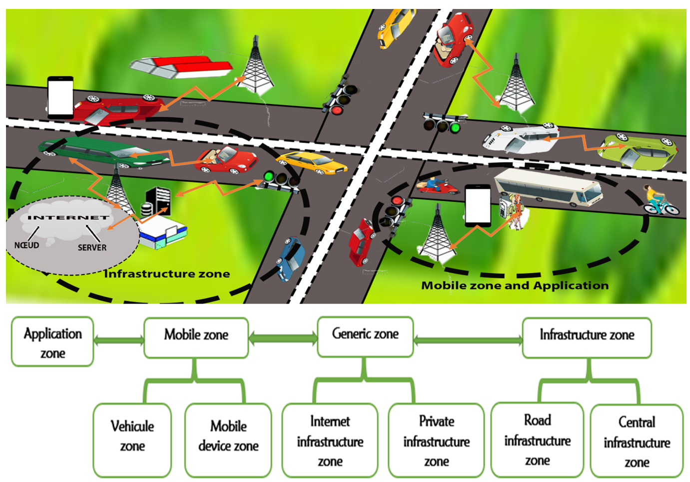

Fig. 3. VANET network communication architecture model

A complete model of the VANET system is established with vehicles as mobile nodes and RSUs as stationary nodes. RSUs are base stations that broadcast alert messages to all nodes within their range. It is assumed that RSUs are always connected to each other by a gateway. Furthermore, all RSUs are maintained in a central data center. The RSUs receive data packets: from the traffic management center; from other RSUs and from moving vehicles. The RSU maintains a database of the number of vehicles in its area, the emergency message alerts it has received from TMC/Vehicles, and the data to be broadcast. The RSU then broadcasts the data packets to nearby vehicles that reach their destination directly or through an ad hoc network. For example, in case of saturation on a main road, the RSU receives the alert message from the vehicles in that area and transfers it to the neighboring RSUs which, in turn, broadcast the message to the vehicles that are far from the saturation area and propose to these vehicles another non-saturation route, for example a secondary road. A second example is when there is an accident in an area, the RSU receives the alert message from a vehicle in the area and transfers it to the neighboring RSUs, which in turn broadcast the message to vehicles that are far from the area where the accident occurred. This gives these vehicles the opportunity to take an alternate route to avoid the accident area where roads may have been blocked or heavy traffic congestion may have occurred. In addition, information about the accident can be transmitted to an ambulance or police vehicle via other vehicles or emergency units, or even via the traffic management center if there is no emergency vehicle nearby. This way, a normal traffic scenario can be quickly restored and a large amount of time can be avoided. Furthermore, in our model, each vehicle communicates with the others by broadcasting messages. This could be very useful in case of extreme weather conditions, where even the visibility of nearby vehicles could be difficult, also in case of sharp turns or road damage. Thus, three communication scenarios are used in our road congestion detection system: V2V - Vehicle to Vehicle; V2I - Vehicle to Infrastructure and I2I - Infrastructure to Infrastructure. By using this communication combination, maximum interconnectivity can be achieved in the VANET scenario and data packets can be transferred to each vehicle in the network.

4 Experimental Results and Discussions

4.1 Simulation on the Hough transform

In this section, we will represent the different results for detecting white lines on a road.

4.1.1 Detection of lines on the road

Image "A" represents the original image without processing. The results of the simulation for edge detection by Canny are shown in image "B". It is important to mask the areas that are not part of the road to get the best results. Image "C and D" represents the result of masking the regions of interest. The mask facilitates the detection of lines on the road and avoids detecting lines of the road. The mask facilitates the detection of lines on the road and avoids the detection of lines off the road. The result of detecting lines on the road using the TH method is shown in image "E". This method is mainly applied to images where the roads have straight lines.

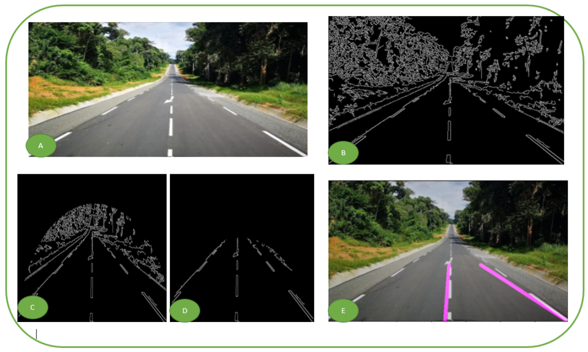

Fig. 4. Lines detected on the track

Some limitations of the method: the line average considers only 2 lanes, while there may be several lanes. Also roads with interchanges as long as two roads are superimposed it is difficult to detect the lines on these lanes. Also, this method assumes one lane on each side, and would fail if both lanes were on one side only. If the road is on a slope, the region of interest must be modified to detect the first horizon.

4.1.2 Traffic sign detection

When the sign is recognized by the system, it starts displaying a symbol representing it on the vehicle's dashboard. This graphic representation will remain on the dashboard until the driver enters a new area, subject to other traffic signs different from the one detected. Figure 5 is the result of the roadside sign detection.

Fig. 5. Traffic signs detected

4.1.3 Saturation detection with video surveillance

In this part, we will detect vehicles on the road using video surveillance, then count the number of vehicles occupying this road, when the number of vehicles is between 0 and 4 the road is at low saturation, then when it is between 5 and 9 the road is near saturation and when the number of vehicles is greater than 9 then the road is at saturation. In Figure 6: (a) and (b) represent low saturation roads with 3 vehicles on the road; Figure (c) and (d) represent near saturation roads with 6 and 9 vehicles on the road; (e), (f), (g) and (h) represent saturation roads containing 26 to 27 vehicles.

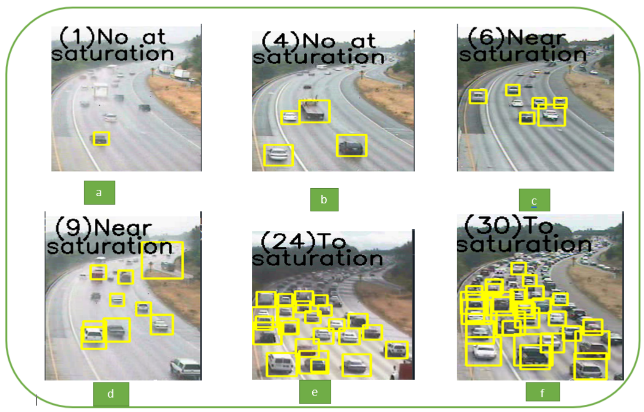

Fig. 6. Saturation detection with video surveillance

4.2 Simulation of Vehicle Saturation with SUMO and NS2

Traffic saturation detection and management using an Ns2 vehicular Ad-Hoc network, SUMO. In this section, we present the design and simulation of an intelligent transportation system allowing V2V and V2I communication for traffic congestion management. In figure 7: (a) represents a road map on which we performed the tests; (b), (c),(d),(e) and (f) represent the simulation of saturation, the blue vehicle being our test vehicle, we notice on (c) that there are more vehicles in front of the blue test vehicle on the road, so the algorithm proposes another alternative to the blue vehicle thanks to a V2V and V2I communication, to avoid the vehicle to enter a saturation zone.

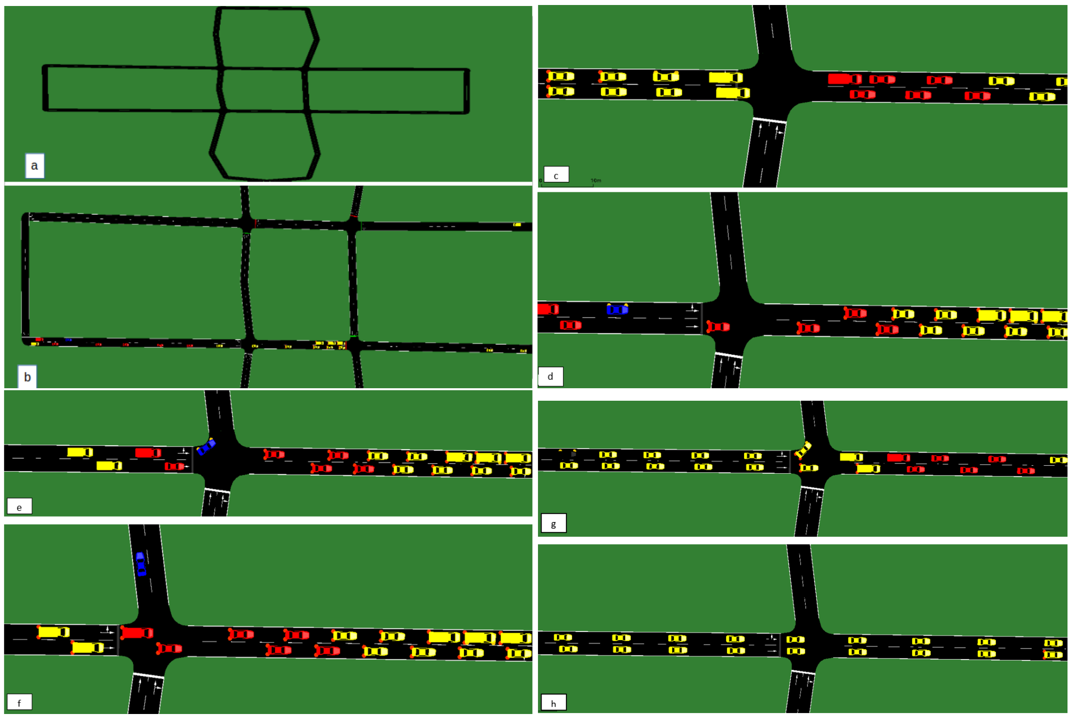

Fig. 7. Simulation of saturation on the road with SUMO

4.3 VANET Simulation

4.3.1 Comparison between AODV and DSDV

In this section we will perform an analysis of the two protocols - AODV and DSDV

- both protocols use the concept of sequence number to update the latest routing information
- bandwidth is wasted in the case of DSDV, due to the periodic broadcast of updated information. In AODV, nodes only propagate hello messages to their neighbors;
- in DSDV, the route information that is maintained in the routing table may be outdated because DSDV cannot handle the movement of nodes at high speeds due to the lack of alternative routes. But in AODV, routes are found on demand only, so route information cannot become stale;
- In DSDV, the throughput decreases due to periodic updates of routing information and if the mobility of nodes is high. In AODV, the throughput is stable because it does not broadcast routing information.

The end-to-end delay is also calculated for the DSDV and AODV routing protocols. The end-to-end delay represents the average time between the start of transmission of a packet by a source node and the delivery of the packet to a destination. It includes delays caused by buffering of data packets during route discovery, queuing at the interface, retransmission delays at the MAC, and propagation and transfer times. When calculating the average end-to-end delay (E2E) for DSDV and AODV, we see that AODV outperforms DSDV, with a lower delay. Since DSDV dynamically checks for the shortest path at each instant, even when the established route exists, the delay is higher and the throughput is therefore lower. While AODV protocol uses an already established route until the nodes move out of range. Therefore, the delay is lower and the throughput is high. Table 1 represents the obtained results of end-to-end delay as a function of the number of nodes present.

Table 1: AODV end-to-end delay

Number of Nodes	Average End-to-End Time (ms)
10 nodes	293.693
15 nodes	336.195
20 nodes	320.826
25 nodes	340.157

The table 2 shows the end-to-end delay results as a function of the number of nodes present.
The analysis shows that the AODV protocol provides lower delay regardless of the number of nodes

Table 2: DSDV end-to-end delay

Number of Nodes	Average End-to-End Time (ms)
10 nodes	671.866
15 nodes	724.623
20 nodes	869.972
25 nodes	1549.39

chosen. Therefore, in our project, which requires a demanding routing protocol with high throughput and low latency, as the vehicles move at high speed, AODV is more suitable for vehicular communication.
The parameters used for the realization of this comparative study of the AODV and DSDV protocol are presented in Table 3 .

Table 3: The ns2 parameters used for the protocols

Parameters	Values
Channel type	WirelessChannel
Number of nodes	10, 15, 20 and 25
Simulation time	150
Simulation area	500*400
Routing protocol	AODV and DSDV
Type of Mac	802.11
Type of data	TCP
Interface queue type	DropTail/PriQueue
Propagation	TwoRayGround

4.3.2 Communication model

NS2 is used to simulate communication models. Before trying the scenario in real time, it is necessary to check the performance of different communication models such as I2I, V2V, V2I in order to get optimal results. In addition, the wireless scenario using AODV protocol needs to be analyzed and confirmed that it could give successful results in real time.

Analysis of AODV The Ad hoc On Demand Distance Vector (AODV) protocol is analyzed for different scenarios. For example, let us consider 5 nodes:

- The communication is first established when a single node acts as a sender while the other nodes act as receivers/routers. Then the scenario is extended with two nodes sending data, then three nodes and so on;
- A scenario is also checked in which the duration of presence of the nodes in range differs. For example, when the sender and receiver are in range for the full duration, and when they are in range for a short duration.

The packet delivery rate and average end-to-end delay are calculated for each model:

- packet delivery rate = (received data/sent data)*100;
- end-to-end delay = (Downtime - Uptime).

RSU-TMC-RSU communication: In the VANET scenario, it is assumed that the RSUs are always connected via a gateway. They are interconnected with each other and also with the central TMC. The RSU-TMC-RSU simulation is first executed and the communication is activated, the result is shown in figure 8. Source - Central server; Receivers 1,2,3,4 - RSU. The central server maintains the

Fig. 8. RSU-TMC-RSU communication

complete database of traffic conditions and broadcasts messages to the RSUs at regular intervals. The RSUs also report traffic information in their communication area to the TMC, which then updates its database.

Vehicle-to-Vehicle Communication The V2V scenario is simulated individually using the AODV protocol and analyzed. The wireless scenarios are designed for a variable number of mobile nodes where each node acts as a transmitter, receiver and router. The nodes are set up manually and the movements occur at random times.

Scenario 1: V2V communication is first verified for different numbers of nodes, but with only 5 nodes actually communicating, the other nodes acting as routers. The result is shown in Figure 9.

The table 4 is a representation of the results obtained concerning the packets sent, received, lost and the delivery rate according to the number of nodes present.

Table 4: PDR pour V2V - Scénario 1

Number of nodes	Packages sent	Packages received	Package delivery rate	Lost packages
5 nodes	14473	14306	98.85	167
10 nodes	13831	13684	98.94	147
25 nodes	15206	14945	98.28	261
50 nodes	12085	11889	98.38	196
100 nodes	12779	12533	98.07	246

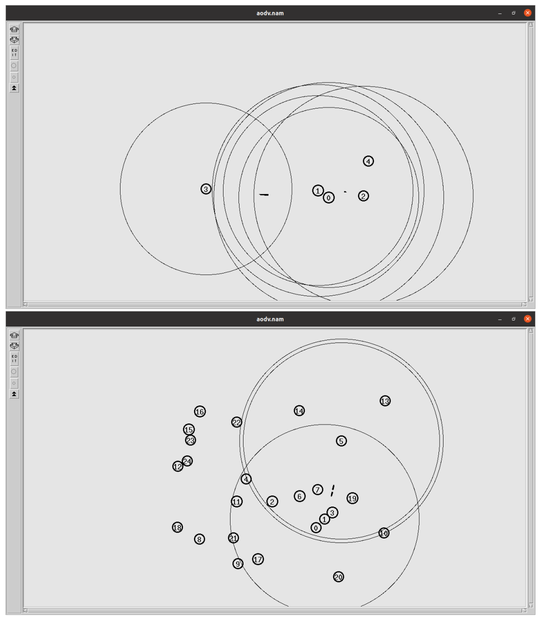

Fig. 9. V2V Communication - Scenario 1

The table 5 shows the end-to-end delay results as a function of the number of nodes present in scenario 1.

Table 5: PDR for E2E - Scenario 1

Number of Nodes	Average End-to-End Time (ms)
5 nodes	399.318
10 nodes	520.068
25 nodes	489.059
50 nodes	600.775
100 nodes	550.753

The results are tabulated and analyzed, and it turns out that the results are optimal in terms of parameters - high PDR and low E2E (a delay of less than 1 second is very good, so that the message can reach its destination quickly and the driver is alerted).

Scenario 2: The idea is now extended where each node communicates with all other nodes by broadcasting messages. In order to avoid broadcast storms, the communication is limited in such a way that at a given moment, a node transmits data to a destination, directly or through an Ad-Hoc Network, and once this communication is over, the next one is started. Thus, even though nodes are broadcasting packets, extreme congestion is avoided and packet loss is reduced. For example, if 100 nodes are in an area, node 1 will communicate with the other 99 nodes, node 2 with the other 99 nodes, and so on, so that a link is established between all nodes. But when node 1 needs to send data to node 25, it first checks the available paths to node 25 and selects the shortest available path. Thus, the data packet reaches node 25 directly or through the ad hoc. But when the communication between node 1 and node 25 takes place, node 1 does not send data to the other nodes, because if node 1 sends data to all nodes at the same time, heavy congestion may occur. In addition, the other nodes are sending data at this time in a similar manner. Thus, maximum interconnection is established and congestion is also avoided. In this part the table 6 is an illustration of the results obtained on the number of packets sent, received, lost and the delivery rate according to the number of nodes present.

Table 6: PDR for V2V - Scenario 2

Number of nodes	Packets sent	Packets received	Packet delivery rate()	Packets lost
5 nodes	15520	15384	99.12	136
10 nodes	14545	14196	97.60	349
25 nodes	14340	13586	94.74	754
50 nodes	12484	11386	91.20	1098
100 nodes	10459	9164	87.62	1295

Table 7: PDR for E2E - Scenario 2

Number of Nodes	Average End-to-End Time (ms)
5 nodes	330.808
10 nodes	824.376
25 nodes	1053.62
50 nodes	1134
100 nodes	1247.84

The table 7 represents the end-to-end delay results as a function of the number of nodes present in

scenario 2.
Thus, we can see that the results are satisfactory even for the diffusion case.

5 Conclusion and Perspectives

In this paper, we have successfully detected the white lines on the roads, followed by the detection of roadside signs, we were also able to detect the saturation with a video surveillance. In addition we established the V2V, V2I and I2I communications and ran the simulation of two (02) different scenarios and analyzed the traffic and communication for each model through an ad-hoc network using the AODV protocol. We also analyzed the packet delivery ratio and average delay and eliminated the errors that occurred during this project, in order to establish a reliable communication that can be used in real life situation. In perspective, the saturation detection method for crowd presence, as part of infectious disease prevention and to help emergency vehicles to move quickly without entering an area that is already saturated. The method of detecting lines on roads could be extended by combining with the method of detecting traffic signs could help drivers not to deviate from the road (an alert is sent to drivers as soon as they deviate from the line). Also the method could guide the drivers on stops, overtaking, speed limit, this helps to keep the drivers on the alert and reduce the number of accidents. We will see to what extent we can adapt our saturation detection method to help ambulances (emergency vehicles) to move quickly without entering a saturation zone, by applying V2V and V2I communication between vehicles, to allow other vehicles to leave the way for emergency vehicles and also to propose a short path to emergency vehicles.

References

[1] Moizumi H, Sugaya Y, Omachi M, Omachi S. Traffic Light Detection Considering Color Saturation Using In-Vehicle Stereo Camera. Journal of Information Processing. 2016;24(2):349-57.

[2] Huang S, Gao C, Meng S, Li Q, Chen C, Zhang C. Circular road sign detection and recognition based on hough transform. In: 2012 5th International Congress on Image and Signal Processing; 2012. p. 1214-8.

[3] Garcia-Garrido MA, Sotelo MA, Martin-Gorostiza E. Fast Road Sign Detection Using Hough Transform for Assisted Driving of Road Vehicles. In: Moreno Diaz R, Pichler F, Quesada Arencibia A, editors. Computer Aided Systems Theory – EUROCAST 2005. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005. p. 543-8.

[4] Sere A, Traore CADG, Traore Y, Sie O. Towards Traffic Saturation Detection Based on the Hough Transform Method. In: Arai K, Kapoor S, Bhatia R, editors. Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2. Cham : Springer International Publishing; 2021. p. 263-70.

[5] Ballard DH. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition. 1981;13(2):111-22. Available from: https://www.sciencedirect.com/science/article/pii/0031320381900091.

[6] Ballard DH. Parameter networks: Towards a theory of low-level vision. ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE; 1981.

[7] Maitre H. Un panorama de la transformation de Hough. Traitement du signal. 1985;2(4):305-17.

[8] Kalviainen H, Hirvonen P, Xu L, Oja E. Probabilistic and non-probabilistic Hough transforms: overview and comparisons. Image and Vision Computing. 1995;13(4):239-52. Available from: https://www.sciencedirect.com/science/article/pii/026288569599713B.

[9] Kasturi R, Bow ST, El-Masri W, Shah J, Gattiker JR, Mokate UB. A system for interpretation of line drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1990;12(10):978-92.

[10] Duda RO, Hart PE. Use of the Hough Transformation to Detect Lines and Curves in Pictures. Commun ACM. 1972 jan;15(1):11–15. Available from: https://doi.org/10.1145/361237.361242.

[11] Huang S, Gao C, Meng S, Li Q, Chen C, Zhang C. Circular road sign detection and recognition based on hough transform. In: 2012 5th International Congress on Image and Signal Processing; 2012. p. 1214-8.

[12] Leavers V, Boyce J. The Radon transform and its application to shape parametrization in machine vision. Image and Vision Computing. 1987;5(2):161-6. 2nd Alvey vision meeting. Available from: https://www.sciencedirect.com/science/article/pii/0262885687900448.

[13] Magli E, Olmo G, Presti LL. Pattern recognition by means of the Radon transform and the continuous wavelet transform. Signal Processing. 1999;73(3):277-89. Available from: https://www.sciencedirect.com/science/article/pii/S0165168498001984.

[14] D Ben-Tzvi MS. Analogue implementation of the Hough transform. IEE Proceedings G (Circuits, Devices and Systems). 1991 August;138:457-62(5). Available from: https://digital-library.theiet.org/content/journals/10.1049/ip-g-2.1991.0078.

[15] Quan L. Conic reconstruction and correspondence from two views. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1996;18(2):151-60.

[16] Hoppenot P, Colle E, Barat C. Off-line localisation of a mobile robot using ultrasonic measurements. Robotica. 2000;18(3):315–323.

[17] Muniz R, Junco L, Otero A. A robust software barcode reader using the Hough transform. In: Proceedings 1999 International Conference on Information Intelligence and Systems (Cat. No.PR00446); 1999. p. 313-9.

[18] Rey MT, Tunaley JK, Folinsbee JT, Jahans PA, Dixon JA, Vant MR. Application Of Radon Transform Techniques To Wake Detection In Seasat-A SAR Images. IEEE Transactions on Geoscience and Remote Sensing. 1990;28(4):553-60.

[19] Sere A, Sie O, Andres E. Extended Standard Hough Transform for analytical line recognition. In: 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT); 2012. p. 412-22.

[20] Tarel RBP. Détection des panneaux de signalisation routière par accumulation bivariée. Traitement du signal. 2010;27(3).

[21] Aissani M, Senouci MR, Demigna W, Mellouk A. Optimizations and Performance Study of the Dynamic Source Routing Protocol. In: International Conference on Networking and Services (ICNS '07); 2007. p. 107-7.

[22] Sere A, Ouedraogo FT, Zerbo B. An improvement of the standard hough transform method based on geometric shapes. In: Future of Information and Communication Conference. Springer; 2018. p. 369-84.

[23] Mezghani F, Dhaou R, Nogueira M, Beylot AL. Content dissemination in vehicular social networks: taxonomy and user satisfaction. IEEE Communications Magazine. 2014;52(12):34-40.

[24] Khan KUR, Zaman RU, Reddy AV, Reddy KA, Harsha TS. An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison. In: 2008 Second UKSIM European Symposium on Computer Modeling and Simulation; 2008. p. 506-11.