

## Research Article

# Jam Eyes: A Traffic Jam Awareness and Observation System Using Mobile Phones

**Xing Zhang, Haigang Gong, Zongyi Xu, Jinchuan Tang, and Bang Liu**

*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

Correspondence should be addressed to Haigang Gong, hggong@uestc.edu.cn

Received 31 October 2012; Accepted 3 December 2012

Academic Editor: Ming Liu

Copyright © 2012 Xing Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic jam is a very common and very annoying thing in urban traffic. The most annoying part in traffic jams is not that you have to wait for a long time but that you do not even know how long you have to wait and what causes the traffic jam. However, the pain of being trapped in traffic jams seems to be neglected by existing research works; they put their focuses on either mathematical modeling or optimal routing for those not trapped in traffic jams. In this paper, we propose a traffic jam awareness and observation system using mobile phones. It can tell a driver how many vehicles ahead are trapped in traffic jam and how much time the driver would probably wait. Moreover, it can provide real-time video streams from the head vehicles of the traffic queue so that the driver can see what causes the traffic jam and the progress of handling the traffic jam. The system is environment independent; it can even work when the traffic jam happens in a tunnel. Experiments show that our system can find the head vehicles of the traffic queue and give the queue length accurately, and the video streams coming from the head vehicles reflect the actual situation of the traffic jam basically.

## 1. Introduction

Traffic jam is already a daily routine of modern urban traffic. The sources of traffic jam can be categorized into three ways: (1) a temporary obstruction, (2) a permanent capacity constraint in the network itself, and (3) a stochastic fluctuation in the demand within a particular sector of the network [1]. Obviously, the second way is the fundamental reason why traffic jam happens so frequently. Researchers have been trying their best to reduce the frequency of traffic jam; however, their works are basically a kind of optimization, as long as the network capacity is far from handling the actual increasing traffic flows, traffic jams will be inevitable and be getting worse. Now that traffic jam is inevitable, we should at least pay some attention to relieving sufferings of people from trapping in traffic jams.

Almost everyone living in the city has experienced traffic jam; the most annoying thing in traffic jam is not that people have to wait for a long time but that people even do not know how long they have to wait. When people are trapped in a traffic jam, unless they are the head of traffic queue,

they hardly know what causes the traffic jam, how long is the traffic queue, and how is the progress of handling the traffic jam. In psychology, lines of evidence shows that people have strong fear of unknown [2]. Although the mentioned information cannot handle the traffic jam, it can handle the fear of people trapped in traffic jam.

Traffic density is a way to distinguish traffic jam from free flow. Traffic jam will cause a much higher traffic density than the density in free flow, and if we can obtain the locations of all vehicles in an area and depict a density map, traffic jam can be recognized. However, this method will cause big communication and computation cost; every vehicle will have to broadcast its location constantly and analyze the gathered locations to judge whether there is a traffic jam. Moreover, existing localization methods including GPS and WiFi Fingerprint [3] are environment dependent, the satellite signal and WiFi signal are easily blocked by buildings, trees, and so forth, and if traffic jam happens in a tunnel, GPS and wifi will fail to work. In fact, traffic density is a kind of relationship among each vehicle, and relative locations are enough to represent the relationship.

When a vehicle is trapped in a traffic jam, it has no speed or drive in a very low speed for at least a period of time, also it has neighbor vehicles surrounding it, and the neighbor vehicle also has no speed or low speed. The observations can help a vehicle to aware whether it is trapped in a traffic jam. The problem is how to distinguish parking from trapping in a traffic jam, because when a vehicle is parking at somewhere, it has neighbors surrounding it and it has no speed. Actually, we cannot distinguish the two cases, but with the help of a reasonable assumption, we can guarantee that the parking case will not interfere the judgment of traffic jam. In this research, we use mobile phones to implement our system; mobile phones have various sensors which can provide enough information of vehicle conditions, and they have networking ability to construct networks and communicate with each other which is not available on modern vehicles (although vehicular ad hoc network (VANET) has been researched for years, it has not equipped on modern vehicles). The assumption is that mobile phone is kind of a belonging to people; drivers will carry their mobile phones with them once they parked and left their vehicles. When a vehicle parks at a place, although it will sense a low speed of itself, it cannot sense any neighbors surround it because there are no surrounding mobile phones available for traffic jam judgment, and the vehicle therefore will not consider itself as trapping in a traffic jam.

Although vehicles are able to aware whether they are trapped in a traffic jam, they do not know their surrounding vehicles' relative locations: whether a surrounding vehicle is before or behind itself. Because the detecting signal is omitted omnidirectionally, a vehicle can know whether there is any surrounding vehicle but it cannot know the relative location of the surrounding vehicle. Without the information of relative location, a vehicle will not be able to count how many vehicles trapped in the traffic jam are ahead and estimate how long it has to wait. Fortunately, the time sequence of traffic jam can be mapped to the location sequence; if vehicle A joins a traffic queue earlier than vehicle B, vehicle A is ahead of vehicle B. If each vehicle records a time stamp when it is aware that it is trapping in a traffic jam, we can get the relative locations of each vehicle according to the time sequence. Therefore, the head vehicles of traffic queue are easily identified. If the head vehicles are willing to share their cameras, the drivers in the behind will be able to see what causes the traffic jam and how is the progress of handling the traffic jam.

The rest of the paper is organized as follows: Section 2 explains the system in detail; Section 3 shows the evaluation results. Related works and conclusions will be given in Sections 4 and 5.

## 2. System Overview

In this research, we do not need any extra hardware; the system is implemented on carry-on mobile phones. As mentioned before, to be aware of the traffic-jam state for a vehicle, two conditions must be satisfied: (1) the vehicle

drives in no speed or low speed for at least a period of time; (2) the vehicle has neighbors which are also trapping in the traffic jam. We can use a triple to describe the traffic jam state:  $Jam_i(Neighbors, Speed, Elapse)$ ,  $i$  is vehicle's ID,  $Neighbors$  is the set of neighbor vehicles' IDs,  $Speed$  is the driving speed of vehicle, and  $Elapse$  is the elapse time of staying in the  $Speed$ . As (1) shows

$$Jam_i = \begin{cases} \text{level1 :} & 0 \leq Speed \leq 20 \text{ km/h,} \\ \text{level2 :} & \text{level1 and } Elapse \geq 5 \text{ min,} \\ & (\text{level2 and } \exists j \in Neighbors \\ \text{level3 :} & \text{that } Jam_j = \text{level2 or level3}) \text{ or} \\ & (\text{level1 and } \exists j \in Neighbors \\ & \text{that } Jam_j = \text{level3}), \\ \vdots & \\ \text{normal :} & \text{otherwise,} \end{cases} \quad (1)$$

there are four different states of  $Jam_i$ : level1 is set when vehicle  $i$  drives in a speed which is lower than 20 km/h, and it has some possibility that vehicle  $i$  is trapping in a traffic jam; level2 is set when vehicle  $i$  stays in level1 for more than 5 minutes, and there is a big possibility that vehicle  $i$  is trapping in a traffic jam; however, it has to check whether there is any neighbor which is also trapping in the traffic jam, otherwise it is hard to distinguish it from mechanical failure; level3 is set when vehicle  $i$  is already in level2 and it at least has one neighbor which is also in level2, or if vehicle  $i$  is in level1 and it has at least one neighbor which is in level3. If vehicle  $i$  is in level3, it is trapping in a traffic jam. Therefore, if vehicle  $i$  is in level1 and it has at least one neighbor in level3, traffic jam is the most possible reason which slows it down, and  $Jam_i$  is set to level3 directly.

Once  $Jam_i$  is set to level1, the system will record the time  $t_i$ , and  $t_i$  will be used to judge the position of vehicle  $i$  in the traffic queue. Vehicle  $i$  or the driver can initiate a request for acquiring information of vehicles in which  $t_i > t_j$ ,  $j \neq i$ , through mobile ad hoc network (MANET) constructed by mobile phones. When the information is returned, the head vehicle of traffic queue is the one that has the smallest time stamp; the length of traffic queue is the number of qualified vehicles; video stream will send from the head vehicle through multihops in MANET.

**2.1. Traffic Jam Awareness.** The triple  $Jam_i(Neighbors, Speed, Elapse)$  indicates that the system needs three dimensional information. *Elapse* simply uses timer, and *Speed* and *Neighbors* use both sensors and wireless signals.

**2.1.1. Speed.** The readings of sensors are three dimensional, and they all obey the coordinate system of mobile phones. Take android system for example; if the mobile phone is placed as Figure 1 shows, the  $x$  axis is horizontal and points to the right, the  $y$  axis is vertical and points up, and the  $z$  axis points towards the outside of the front face of the screen [5].

Accelerometer gives three dimensional acceleration values, we use  $A_z$  to represent acceleration at  $z$  axis,  $A_x$  to

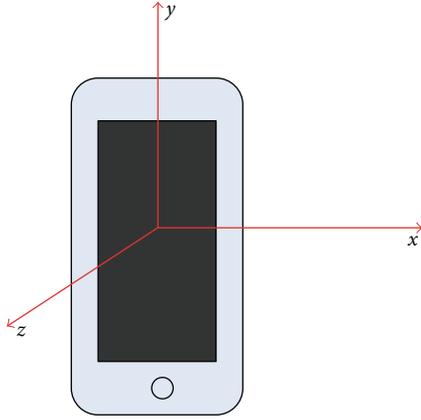


FIGURE 1: Coordinate system of mobile phones.

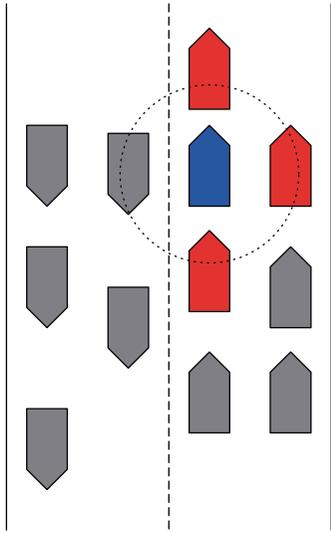


FIGURE 2: Neighbors detection. Blue block is a vehicle which initiates a detection of neighbors, red blocks are qualified neighbors of the blue block, and grey blocks are not the neighbors of the blue block.

represent acceleration at  $x$  axis,  $A_y$  to represent acceleration at  $y$  axis. If we fix the mobile phone as Figure 5(a) shows,  $A_z$  and  $A_y$  will be used to calculate vehicle speed as (2) shows, and  $A_x$  will never be used unless vehicles can fly. The sample rate of accelerometer we set is 20 ms; therefore  $\Delta t = 20$  ms:

$$a_i = \sqrt{A_y^2(i) + A_z^2(i)}, \quad (2)$$

$$v_{i+1} = v_i + a_{i+1} \Delta t. \quad (3)$$

**2.1.2. Neighbors.** Normally, vehicles are closed to each other in traffic jam. In this research, direct neighbors will be detected as *Neighbors*. As Figure 2 shows, the blue block is the vehicle detecting its neighbors, the red blocks are the neighbors of the blue block, and the grey blocks are vehicles not qualified as neighbors of the blue block. We can see that there is a grey block which is in the neighbor range of the

blue block, the reason why this block is not the neighbor of the blue block is that it is heading a different direction from the blue block, and traffic jam happens among vehicles in the same direction. Therefore, distance and direction are two factors for detecting neighbors.

Mobile phones have orientation sensor which returns three-dimensional values, Azimuth (degrees of rotation around the  $z$  axis), Pitch (degrees of rotation around the  $x$  axis), and Roll (degrees of rotation around the  $y$  axis) [6]. If the mobile phone is fixed as Figure 5(a) shows, then Pitch is the degrees we need. Suppose a vehicle  $i$  is detecting its neighbors, vehicle  $j$  is closed to vehicle  $i$ , and  $Pitch_i$  is the Pitch value of vehicle  $i$ , then vehicles  $i$  and  $j$  have the same direction when  $|Pitch_i - Pitch_j| \leq 30^\circ$ .

In this research, each vehicle needs to find its direct neighbors, and direct neighbors means they are very close to the detecting vehicle that there is no other vehicles between direct neighbors and detecting vehicle. If a neighbor is not a direct neighbor, for example, it is 100 meters away from the detecting vehicle, even the neighbor does be trapped in a traffic jam, it cannot be used as a reference to indicate a traffic jam because it is hard to tell whether the traffic jam is the same as the traffic jam the detecting vehicle is trapped in. Modern mobile phones are equipped with WiFi network cards, and the network cards can be used to construct MANET. It is easy to communicate in MANET, but it's hard to detect direct neighbors. If we use distance to represent the relationship of the direct neighbors, as [7] says the average car length is around 4 meters, the average distance between two vehicles is around 2 meters, then 6 meters will be a proper distance. RSSI-based localization [8] is a good way to measure distance, but it is difficult to acquire each node's RSSI value in a MANET with existing tools. In this research, we propose a new and simple detection method called ECHO. Common WiFi signal with 20 dbm initial power can cover an area with at least 100 meters radius, and the area is too large for the neighbor detection. If the initial transmit power is reduced to a much lower level that the signal can only reach up to 6 meters distance, when the detecting vehicle sends one detecting packet, only vehicles within 6 meters from the detecting vehicle can receive the packet and echo back their IDs. *Neighbors* will contain the vehicles which are within 6 meters from detecting vehicle and they have the same direction as the detecting vehicle.

Now that the three parameters of Jam, (*Neighbors*, *Speed*, *Elapse*) are all collected, vehicles are able to judge whether they are trapped in a traffic jam according to (1). Figure 3 shows the states transition of (1). When a vehicle's state changes to level1, it will record a time stamp, and the time stamp will be used to decide the relative location of each vehicle in the same traffic jam.

**2.2. Basic Information of Traffic Jam.** It is easy to calculate the length of traffic queue and find the head of the traffic queue as long as we have all the time stamps of vehicles in level3. To obtain the time stamps, the detecting vehicle needs to broadcast a detection packet, and due to the limitation of

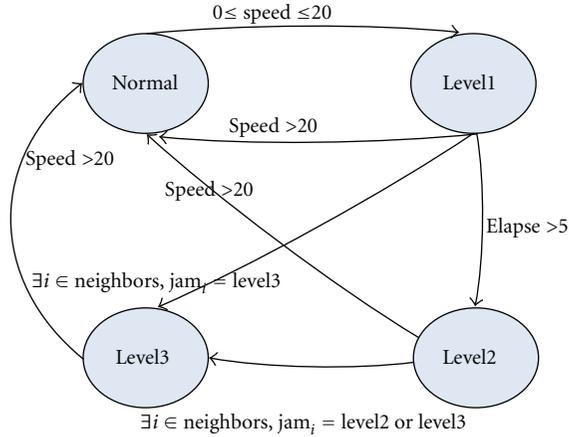


FIGURE 3: State transition of traffic jam. Vehicle starts in normal state; if it drives in a speed lower that 20 km/h, it will change to level1; if level1 stays for at least 5 minutes, it will change to level2; if level1 has one neighbor in level3, it will change to level3; if level2 has one neighbor in level2 or level3, it will change to level3. All levels will change to normal when speed is higher than 20 km/h. We consider a vehicle is trapping in a traffic jam when it is in level3.

```

while 1 do
  if receive one detection packet then
    if already has received the detection packet with same
      source vehicle ID in one minute then
      continue;
    end if
    if Jami = level3 then
      reply a packet with its vehicle ID and time stamp in
      it;
      broadcast the received detection packet;
    end if
  end if
end while
  
```

ALGORITHM 1: Reply and relay actions for received detection packet on vehicle  $i$ .

communication range of WiFi signal, multihop is needed. Algorithm 1 shows the actions of a vehicle when it receives a detection packet.

When detection finished, the system will sort all vehicle IDs according to time stamps. The vehicle with smallest time stamp is the head of the traffic queue, and we use  $V_{\text{head}}$  to represent it; the number of vehicles which have smaller time stamps than the detecting vehicle is the length of the traffic queue, and we use  $Q_i$  to represent the number of vehicles ahead of vehicle  $i$  in the traffic jam.

To estimate the time vehicle has to wait, we have to monitor  $V_{\text{head}}$ 's movement. When  $V_{\text{head}}$  changes its state to *normal*, and the time interval between the latest zero speed and state change to *normal* is recorded as  $T_{\text{head}}$ , the time between a vehicle moves and its following vehicle moves

```

while 1 do
  if it is  $V_{\text{head}}$  then
    broadcast the  $T_{\text{head}}$ ;
  else
    if received the broadcast packet then
      if hasn't received the broadcast packet within a
        minute then
        get the  $T_{\text{head}}$  from the packet;
        broadcast the packet;
      end if
    end if
  end if
end while
  
```

ALGORITHM 2:  $T_{\text{head}}$  broadcast.

is  $\Delta t$ , the time vehicle  $i$  will wait for will be calculated as  $Q_i \times \Delta t - T_{\text{head}}$ , then the calculation result could be negative because when  $V_{\text{head}}$  changes its state to *normal*, some following vehicles already moves. If  $V_{\text{head}}$  has not changed its state to *normal*,  $T_{\text{head}}$  will be  $\infty$ . The problem is how to get the  $T_{\text{head}}$  from  $V_{\text{head}}$ . One way is that each vehicle periodically requests the information from  $V_{\text{head}}$ , and the other way is that  $V_{\text{head}}$  broadcasts the  $T_{\text{head}}$  which it changes its state to *normal*. Obviously, the second way costs less, but it requires a vehicle that can be aware whether it is the  $V_{\text{head}}$ . With the help of the information of *Neighbors*, a vehicle that can be aware it is the  $V_{\text{head}}$  through comparing its time stamp with the time stamps of *Neighbors*; if its time stamp is the smallest, the vehicle is the  $V_{\text{head}}$ . Once a vehicle is aware that it is the  $V_{\text{head}}$ , it will broadcast its  $T_{\text{head}}$  using Algorithm 2.

**2.3. Traffic Jam Observation.** The basic information can tell drivers how long the traffic queue is and how long they probably have to wait; however, there are still a lot of useful information which sensors cannot sense and process; people trapped in the traffic jam usually want to know what causes the traffic jam and how is the progress of handling the traffic jam. It is better to let people see the traffic scene from the position of  $V_{\text{head}}$  and make their own judgment. However, camera is a privacy-sensitive device on mobile phone; people usually would not be willing to share their cameras, and therefore incentive schemes are needed.

Our system is crowdsourcing based, and there are some incentive researches on crowdsourcing systems, like [9–11]. Moreover, there are some researches on leveraging phone cameras to capture traffic lights [1], some researches on taking advantage of phone sensors to monitor driving behaviors [4, 12, 13], and some applications like driving navigation; if our system can be combined with these works, it is possible to make people wish to share their cameras.

Figure 4 shows two figures from [1, 4]; we can see the way the mobile phone fixed is convenient for cameras to capture outside scene. Therefore we use the same way to fix the mobile phone and, as Figure 5(b) shows, the camera



FIGURE 4: Existing mobile phone placement. (a) is from [1], and (b) is from [4].

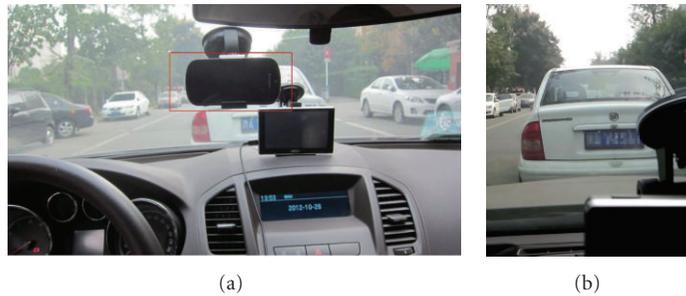


FIGURE 5: Mobile phone placement of our system. (a) is the placement of the mobile phone, and (b) is the camera view.

captures wide enough scene which can give remote drivers the ability to know what causes the traffic jam.

Video stream is large, if  $V_{\text{head}}$  sends each vehicle the video stream using a separate link, such as http, it will cause high overhead on  $V_{\text{head}}$  and network congestion when there are many vehicle requests for the stream. Multicast is a good way to solve the problem. In an ad hoc network, like MANET, every node will act like a router, once a node requests to receive a multicast channel, one other node in the range of this node will act as a multicast router to deliver the multicast packets to the node. As long as we can find a route from the node of request to  $V_{\text{head}}$ , and each node on the route will act as a multicast router, the node of request will receive the multicast packets. Algorithm 3 shows how to achieve a video stream multicast in MANET. In this algorithm, we assume that the relative location of each vehicle is stable and they are closed to each other; therefore the multi hop packet forward process is simply to choose the vehicle with the smallest time stamp.

### 3. Implementation and Evaluation

We rent ten automobiles to simulate a traffic jam in an open space of campus, each automobile is equipped with an android mobile phone fixed as Figure 5 shows. We set up three scenes of traffic jam as Figure 6 shows: (a) a single lane with an obstacle in the middle, and automobile has to slow

down and drive around the obstacle; (b) two lanes change to one lane, and it causes automobiles distributed on two lanes to compete for one lane; (c) automobiles are stopped due to traffic control. We test the accuracy of  $V_{\text{head}}$  judgment, the accuracy of waiting time estimation, the accuracy of traffic queue length, and the frames dropped rate of video stream multicast in the three scenes, and we repeat the tests for 20 times to get the statistics.

Before scene testing, we test the mapping relationship between RSSI and distance, we find that 2 mw is a good initial power to detect direct neighbors.

**3.1. Scene 1.** In scene 1, ten automobiles cannot cause more than 5-minute traffic jam; therefore we parked the ten automobiles in front of the obstacle for 5 minute, and the first automobile starts to drive around the obstacle in a speed of 5 km/h; once the first automobile moves, the second automobile will move in a delay time around one to two seconds due to human response time and driving operation time. The first automobile will spend 3 seconds driving around the obstacle and 1 second speeding up to 20 km/h. The detection of  $V_{\text{head}}$  is doing in a period time of 0.5 second.

Figure 7 shows that the accuracy of  $V_{\text{head}}$  is averagely 63%; the reason is that when a vehicle starts to speed up after driving around the obstacle, the vehicle is no longer the  $V_{\text{head}}$ , but our system will wait until its speed reaches

```

while 1 do
  if received a request packet for video stream then
    if it is  $V_{head}$  then
      if it has not initiate the video stream multicast then
        initiate the video stream multicast;
      end if
      continue;
    end if
    record the vehicle ID of the packet sender as
    a multicast client;
    collect the time stamp and ID information from its
    neighbors;
    choose a neighbor with the smallest time stamp and
    send it the request;
    continue;
  end if
  if receive a reply of request then
    collect the time stamp and ID information from its
    neighbors;
    if the request vehicle in among its neighbor then
      send the reply to the request vehicle;
    else
      choose a neighbor with the largest time stamp and
      send it the reply;
    end if
    continue;
  end if
  if received a video stream packet then
    if it has one or more multicast client records then
      deliver the packet;
    end if
    continue;
  end if
  if receive a quit packet of multicast then
    if it has one or more multicast client records and the
    quitter is one of these client then
      delete the record accordingly;
    end if
    continue;
  end if
end while

```

ALGORITHM 3: Video stream multicast.

20 km/h; the accuracy of waiting time estimation is within 1 second averagely,  $-1$  means the timer says there is still one second to wait while the vehicle is already moving,  $+1$  means the timer already changes to zero while the vehicle moves after one second; the misjudgment of traffic queue length is 37% according to the accuracy of  $V_{head}$ . Frames dropped rate is somehow a little high and unstable, and the reason is probably that the  $V_{head}$  is moving all the time and it may affect the signal propagation.

3.2. *Scene 2.* Scene 2 is more complicated than scene 1 because vehicles on difference lanes will compete for the only one lane; vehicles may change their lanes to occupy a better position, and it will cause that a vehicle with a bigger time stamp is in front of a vehicle with a smaller time stamp;

therefore the accuracy of  $V_{head}$  judgment may be affected. Figure 8 shows the accuracy of  $V_{head}$  judgement is dropped to 48% compared to scene 1, and the view from wrong  $V_{head}$  somehow does not reflect the progress of handling the traffic jam; the accuracy of waiting time estimation is bigger than scene 1 because the the competition will interrupt the moves of vehicle and prolong the  $\Delta t$ ; the misjudgment of traffic queues length is 52% according to the accuracy of  $V_{head}$ ; frames dropped rate is still high due to the same reason.

3.3. *Scene 3.* Scene 3 is another simple case, because vehicles will not compete for better positions and vehicles are not moving until the traffic control released; therefore the frames dropped rate is relatively lower than the other two cases. Figure 9 shows the result, and the accuracy of  $V_{head}$  judgment is 100%, because no vehicle moves in this scene; therefore the time estimation and queue length have perfect results which are not necessary to present here.

## 4. Related Works

Research works on traffic jam mainly focus on analyzing traffic jam through mathematical modeling. Reference [14] shows a new perspective of the origin of traffic jam, and it presents experimental evidence that the emergence of a traffic jam is a collective phenomenon like “dynamical” phase transitions and pattern formation in a nonequilibrium system. Reference [15] gives modeling and simulation of traffic jam in a macroscopic level, and [16, 17] give modeling of traffic jam in a microscopic level; they both give some interesting perspectives. Mathematic modeling can reveal some facts of traffic jam and help reduce the frequency of traffic jam, but we believe that the principal contradiction of traffic jam is the capacity of network that is unable to handle the increasing popularity of vehicles; therefore math can do some optimizations to traffic jam, it can never solve the problem. Now that traffic jam is unbeatable, we should put more energy on relieving suffering of people trapped in the traffic jam.

Sensors on vehicles and mobile phones are able to reflect the behaviors of vehicles: how fast they are, which direction they are heading to. Some researches leveraged mobile and vehicular sensors to monitor and correct drivers’ behaviors to enhance driving safety [4, 18], and some works capture the surrounding information to enhance driving comfort [1]. However, they all neglect the pain of trapping in the traffic jam.

## 5. Conclusions

In this paper, we propose a novel traffic jam awareness and observation system; it gives some basic information of traffic jam like the length of traffic queue, the time waiting in the traffic jam, and so forth, and also it provides video streams from the head of traffic queue to present the cause of the traffic jam and the progress of handling the traffic jam to people who are not in the front of the traffic queue. The system is environment independent, and it need not

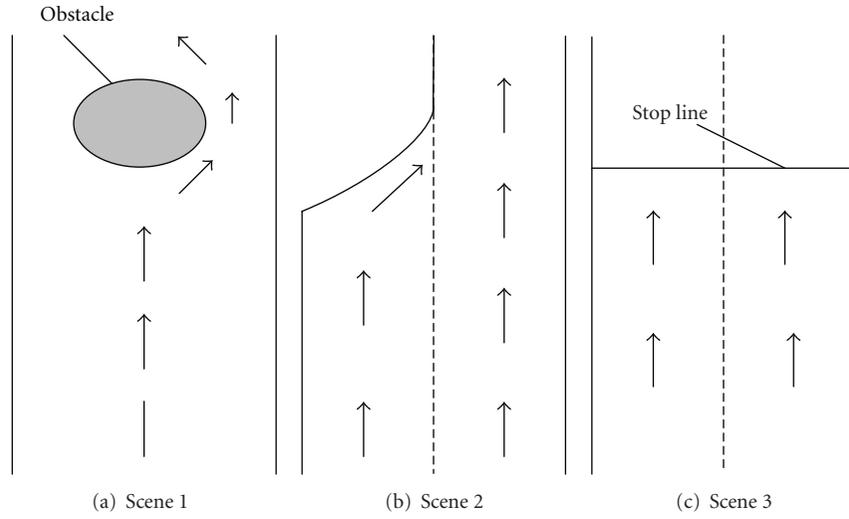
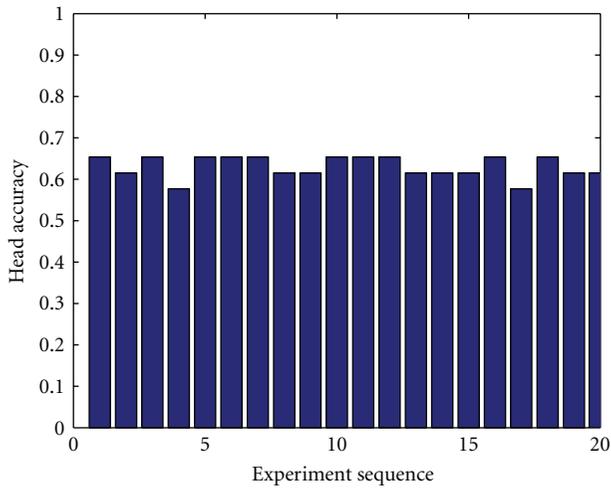
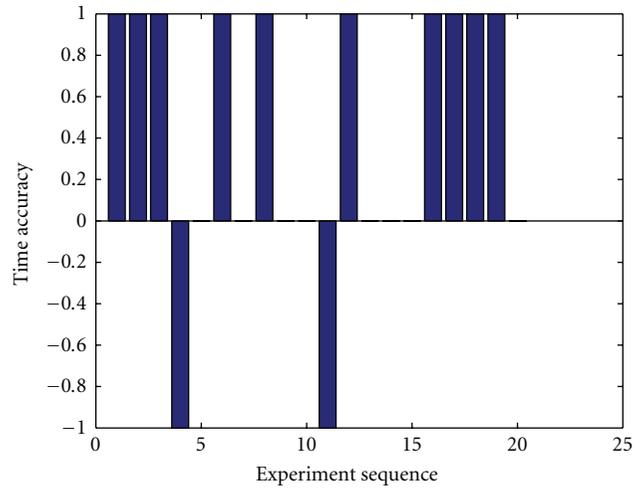


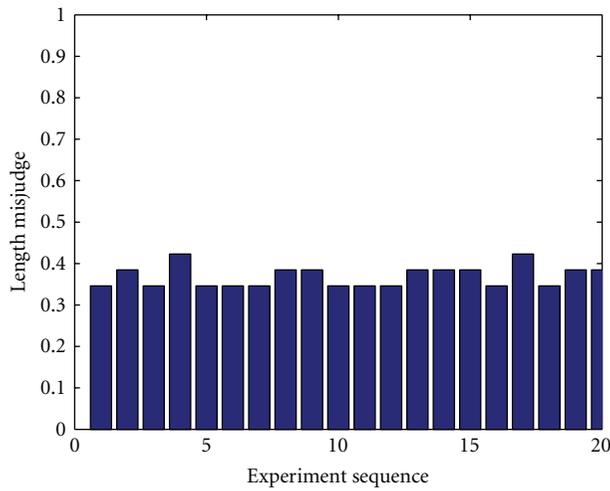
FIGURE 6: Three scenes of traffic jam.



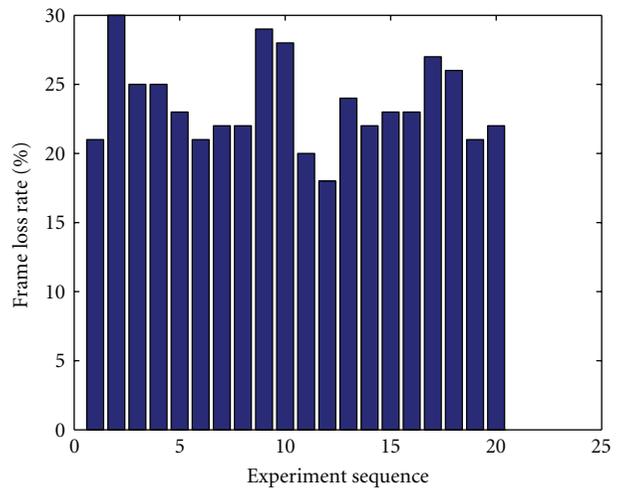
(a)



(b)



(c)



(d)

FIGURE 7: Experiment results of scene 1. (a) is the accuracy of  $V_{head}$  judgment, (b) is the accuracy of time estimation, (c) is the queue length misjudgment rate, and (d) is the frame loss rate.

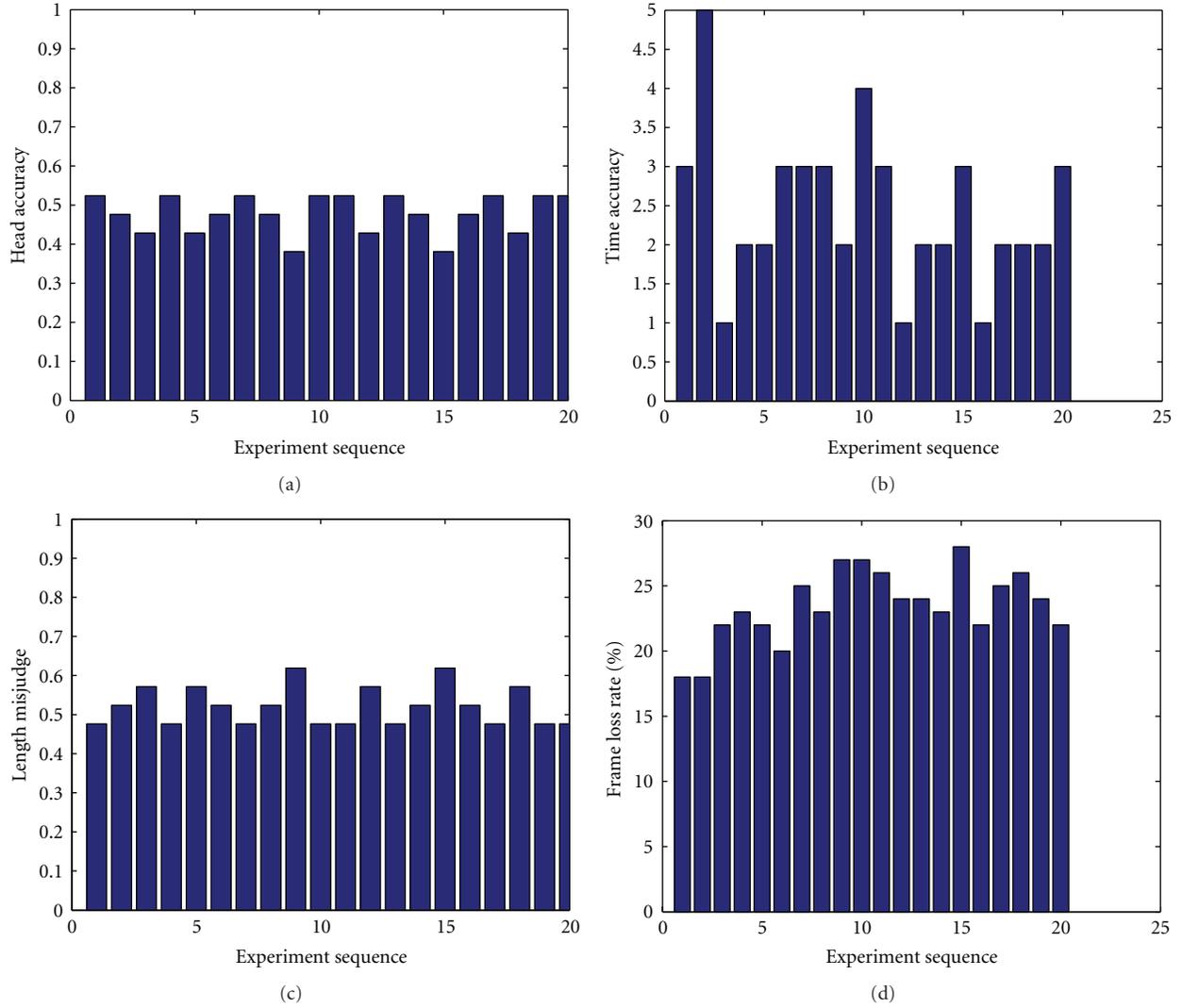


FIGURE 8: Experiment results of scene 2. (a) is the accuracy of  $V_{\text{head}}$  judgment, (b) is the accuracy of time estimation, (c) is the queue length misjudgment rate, and (d) is the frame loss rate.

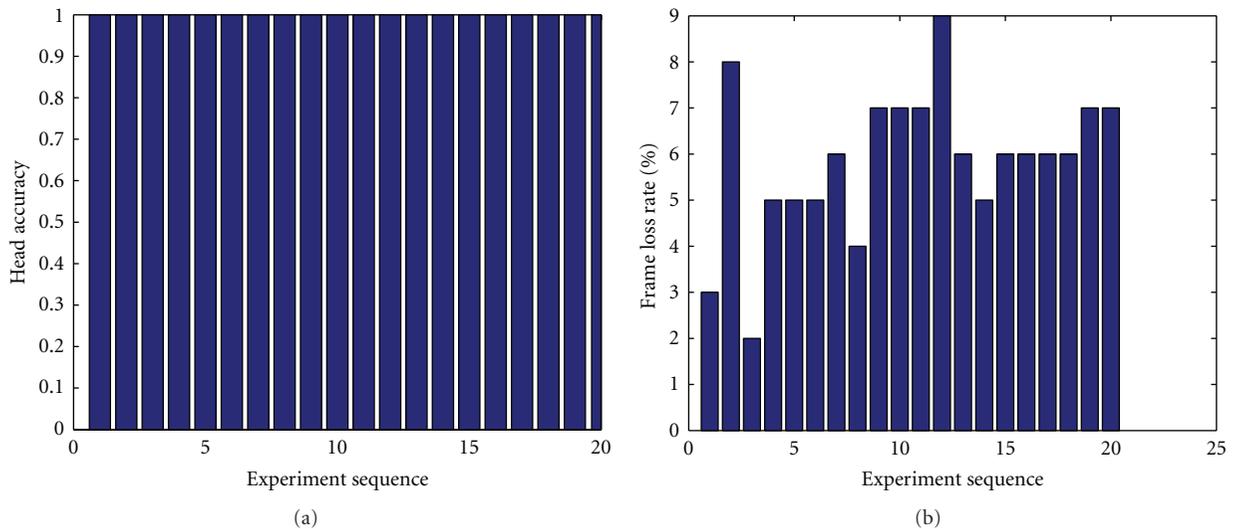


FIGURE 9: Experiment results of scene 3. (a) is the accuracy of  $V_{\text{head}}$  judgment, and (b) is the frame loss rate.

GPS device and any extra hardware, but the mobile phones carried by drivers are good enough.

## References

- [1] E. Koukoumidis, L. S. Peh, and M. R. Martonosi, "SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 127–140, ACM, Bethesda, Md, USA, July 2011.
- [2] H. H. Cao, B. Han, D. Hirshleifer, and H. H. Zhang, "Fear of the unknown: familiarity and economic decisions," *Review of Finance*, vol. 15, no. 1, pp. 173–206, 2011.
- [3] Y. C. Cheng, Y. Chawathe, A. Lamarca, and J. Krumm, "Accuracy characterization for metropolitan-scale Wi-Fi localization," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 233–245, ACM, June 2005.
- [4] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 1609–1615, Washington, DC, USA, 2011.
- [5] Android sensor coordinate system, <http://developer.android.com/reference/android/hardware/SensorEvent.html>.
- [6] Orientate sensor, [http://developer.android.com/guide/topics/sensors/sensors\\_position.html](http://developer.android.com/guide/topics/sensors/sensors_position.html).
- [7] "Vehicles keep inching up and putting on pounds," [http://usatoday30.usatoday.com/money/autos/2007-07-15-big-cars\\_N.htm](http://usatoday30.usatoday.com/money/autos/2007-07-15-big-cars_N.htm).
- [8] J. Arias, A. Zuloaga, J. Lázaro, J. Andreu, and A. Astarloa, "Malguki: an RSSI based ad hoc location algorithm," *Microprocessors and Microsystems*, vol. 28, no. 8, pp. 403–409, 2004.
- [9] P. Trompette, V. Chanal, and C. Pelissier, "Crowdsourcing as a way to access external knowledge for innovation," in *Proceedings of the 24th EGOS Colloquium*, Amsterdam, The Netherlands, July 2008.
- [10] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proceedings of the IEEE INFOCOM*, pp. 2140–2148, IEEE, Orlando, Fla, USA, March 2012.
- [11] D. Yang, G. Xue, and X. Fang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 173–184, Istanbul, Turkey, August 2012.
- [12] G. A. ten Holt, M. J. Reinders, and E. A. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Proceedings of the 13th Annual Conference of the Advanced School for Computing and Imaging*, June 2007.
- [13] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D'Alessio, "Classification of motor activities through derivative dynamic time warping applied on accelerometer data," in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4930–4933, 2007.
- [14] Y. Sugiyamal, M. Fukui, M. Kikuchi et al., "Traffic jams without bottlenecks-experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, Article ID 033001, 2008.
- [15] P. Degond and M. Delitala, "Modelling and simulation of vehicular traffic jam formation," *Kinetic and Related Models*, vol. 1, no. 2, pp. 279–293, 2008.
- [16] G. Orosz, R. E. Wilson, R. Szalai, and G. Stépán, "Exciting traffic jams: nonlinear phenomena behind traffic jam formation on highways," *Physical Review E*, vol. 80, no. 4, Article ID 046205, 2009.
- [17] J. Long, Z. Gao, X. Zhao, A. Lian, and P. Orenstein, "Urban traffic jam simulation based on the cell transmission model," *Networks and Spatial Economics*, vol. 11, no. 1, pp. 43–64, 2011.
- [18] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in *Proceedings of the 4th International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health '10)*, March 2010.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

