isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6 Issue: 3 Apr,2017,ISSN_NO: 2321-3337

Transforming Obsolete Data into Contemporary Data (TOC) to provide terrain information

,Akhila Sai Velaga[1] , Dr.L.Jaba Sheela[2]

Dept. of Computer Science and Engineering, Panimalar Engineering College, India[1].
Professor, Dept.of Computer Science and Engineering, Panimalar Engineering College, India[2]

ABSTRACT— *It is conceivable to determine the state of the road from the driving experience of the vehicle as it covers an extent of it. This data when communicated can be made accessible to the future users of that road, so all around educated choices can be made about changing the speed or maintaining a strategic distance from deceptive streets even though this data is pointless to the vehicle that produced it. This paper displays the TOC framework which plans to record a vehicle's experience while at the same time driving that specific extent of the road. The information gathered from the vehicle is broke down and made accessible to the future clients of the street consequently effectively Transforming Obsolete information into Contemporary information. The primary subject of this paper will be one of the two calculations utilized by the TOC framework to examine the information produced by the vehicles. This algorithm is utilized to collect useful vehicular information from the rest.*

Keywords— vibration sensor, rash driving behaviour, terrain information

1, INTRODUCTION

As of late with innovation introducing the Internet of Vehicles (a branch of the Internet of Things) the advantages of an interconnected group of drivers has become eminent. In the event that each traveler shares his or her experience, different travelers would without a doubt benefit from it. Outfitting the driver with data about the forthcoming way (unpleasant landscapes, reasonable streets, elusive surfaces) will permit better choices about slowing down, accelerating or picking an optional way by and large. Such a facility would avert mishaps as well as protect the suspension arrangement of the vehicle. On the boondocks of completely mechanized cars, this sort of machine to machine sharing can cross over any barrier between human instinct and machine learning. This paper displays the TOC framework which evaluates the driving experience and communicates this data.

2, PROPOSED SOLUTION

A moving vehicle experiences numerous conditions, for example, movement, uneven streets, smooth streets or slippery surfaces. On each event the responses of the vehicle and that of the driver are one of a kind. For instance, a smooth street may urge the driver to increase speed while a dangerous street may trigger the ABS to be conveyed. By dissecting the speed and driving conduct it is conceivable to decide the state of the street. The framework proposed here goes for utilizing the information produced by the vehicle and its sensors to convincingly decide the state of the street.

1) The information generated by the vehicle is sent to a server.
2) The server utilizes exceptional calculations to decide the street's condition
3) The street's condition, dictated by the calculation is refreshed to a guide which the future clients of the street can see on their smartphone.

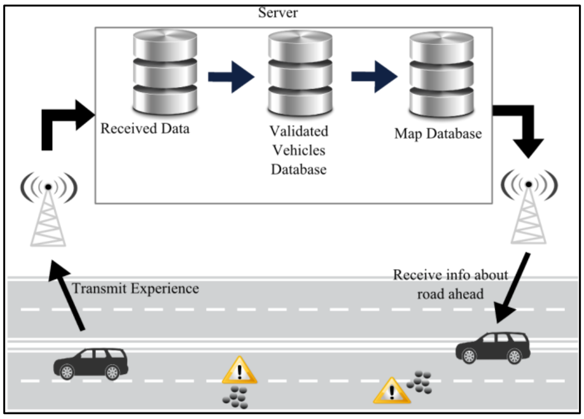

Figure.1 General outline of the system

3, EXPLANATION OF THE SYSTEM

At the heart of this framework is an vibration sensor that measures the experience of the driver. Since driving examples are capricious a solitary sensor can't envision each situation. Accordingly we utilize other supporting sensors. To be specific-

1) A rash driving sensor: This sensor intermittently decides whether the driving conduct is good or rash.
2) A sensor to decide whether the ABS was deployed.
3) The data received from the vehicle is-
    a. *Speed of the vehicle – Speed in km/hr
    b. *Vibration sensor data – Experience of the vehicle (1 = Good, 2 = Fair, 3 = Bad)
    c. Geographical co-ordinates of the vehicle – Map Co-ordinates
    d. *Rash driving sensor data – Binary value (1 = Driving is rash, 0 = Driving is normal)
    e. ABS deployment sensor data – Binary value (1 = Deployed, 0 = Not Deployed)

* implies that this data will be used in the following algorithms.

4, PROBLEM STATEMENT

The information is gotten from each vehicle on the road. This is favorable on the grounds that our calculation will have sufficient information to get conclusions from. This is additionally a drawback on the grounds that distinguishing substantial information from spurious information ends up noticeably tricky. Without a doubt, a vehicle that drives impulsively will have a terrible driving experience. Presuming that the street surface is terrible from such information would prompt incorrect outcomes. Likewise, slowly moving vehicles will dependably have a smooth driving knowledge. It is false to presume that the streets are great from such information. Consequently the framework needs a calculation which can dispose of suspicious information and recognize the solid information. On the off chance that the calculation is excessively prohibitive, the framework will acknowledge information just from the most know-it-all drivers and the outcomes would be of no down to earth utilize. Then again, if the calculation is excessively liberal it might incorporate the information from problematic sources too .

While recognizing the substantial information is a test in itself, the second portion of the issue is to decide the state of the street. Each street is assigned a level: Good, Fair or Bad. This review is finished up from the information got from the Vibration Sensor. The calculation needs to recognize the grade that a greater part of the vehicles have transmitted. Since the need of this framework is to caution the driver of unpleasant landscapes ahead, the calculation needs a critical viewpoint to decide the grade of the street.

Thus the problem statement can be split into two parts, the first is *Identifying the Valid Data* and the second is *Determining the Grade of the Road.*

Note: Grade 1 = Good; Grade 2 = Fair; Grade 3 = Bad. The terms 'Grade' and 'Experience Value' have been used interchangeably.

5, SAMPLE DATA

The data received from the vehicles is stored in the *Received_Data* database shown in Fig.1. The simulated data from 50 cars was used to illustrate the algorithms. A sample is shown in Table 1.

Car	Speed	Rash	Grade
1	37	0	1
2	38	0	2
3	39	1	1
4	50	1	1
5	42	1	2
6	48	1	3
7	50	0	2
8	55	0	2
9	60	1	3

Figure 2. Sample Data

6, ALGORITHM TO IDENTIFY THE VALID DATA

The server acknowledges the information from each vehicle. It dispenses with the inconsistent information by utilizing filters. These filters measure some parameter, for example, the speed or driving conduct to judge if the information is dependable or not. After each filter, some information is sifted through and the rest of the information alone is passed on to the following filter. The filters used are-

1) The Rash Driving filter
2) The Traffic Speed Filter

6.1 The Rash Driving Filter

This filter relies on upon the value got from the rash driving sensor. The conceivable values got from this sensor are 0 and 1. On the off chance that the driving is careless then the value is set to 1, else it is 0. The meaning of "rash driving" incorporates not slowing down at speed breakers, unpredictable utilization of horn and not utilizing pointers at turns. This bit will be intermittently set and reset.

This channel disposes of two kinds of vehicles-

1) The vehicles which are driving rashly with an experience value of 3
2) The vehicles which are driving carelessly with an experience value which shows a lower street condition than the current street condition. (The current street condition is the value that was concluded up from past iterations).

Without a doubt, the vehicles which are driving rashly are driving at high speeds. At high speeds the impact of each little bump will be amplified. So the information from these vehicles can't be utilized. Then again, vehicles which are driving rashly with an experience value demonstrating a street condition more terrible than the current values are additionally wiped out on the grounds that it is difficult to observe if the street condition has really weakened or not. The vehicles which are not rashly driven are permitted to pass. Moreover, the vehicles which are rashly driven however whose experience value indicates great streets are permitted to pass. This condition infers that the street surface has probably improved.

The impact of this filter is delineated with the accompanying representations. The first unfiltered information is represented in Figure 2. Subsequent to going through the rash driving filter the information is more refined. This is represented in Figure 3. By contrasting the two diagrams there is an obvious reduction in the general number of vehicles. The faster vehicles are the most altogether reduced.

6.2 The Traffic Speed Filter

For each street, the normal speed of the traffic changes. On a high way, vehicles move with a normal speed of 60km/hr though on city streets vehicles move at a slow pace of 30km/hr.

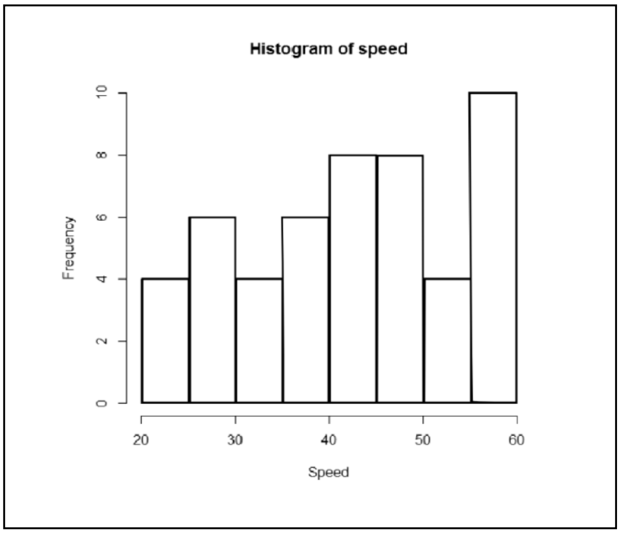

Figure. 3 The Histogram of speed for the Unfiltered Data

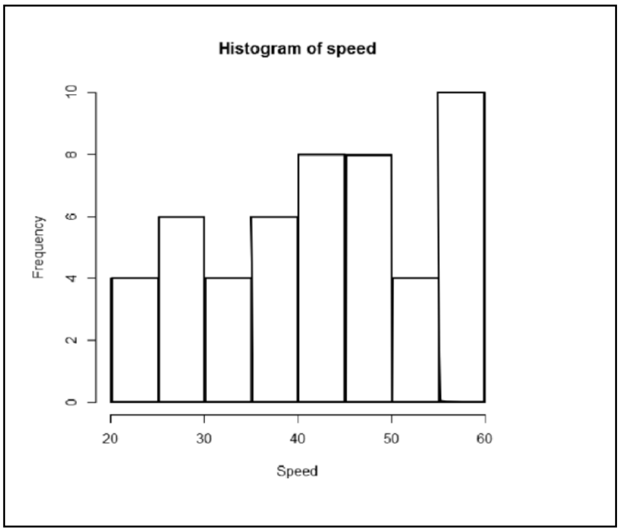

Figure. 4 The Histogram of speed for the data that has passed through the rash driving filter

Subsequently it is difficult to set up a general threshold which can separate the quick and moderate vehicles. Subsequently this filter holds the vehicles moving inside an adequate speed range. This adequate speed range is one of a kind to every street. The algorithm registers the value of the normal traffic speed-TS. This is a weighted mean of the speed of the considerable number of vehicles on a specific extent of street. The normal traffic speed is processed once in each 3 minute interim tokeep up to date. Using the value of TS, the filter works by retaining the vehicles which satisfy the following condition-

$$TS - \Delta TS \leq VS \leq TS + \Delta TS \qquad (1)$$

where TS- Average Traffic Speed, VS- Individual Vehicle's Speed

The value of the differential, Δ is chosen to be 0.15, therefore vehicles with a 15% deviation from TS will be retained by this filter and stored in a separate database. This separate database will henceforth be called the 'Validated_Vehicles_DB'.
The values which can pass this filter must not satisfy either of the following conditions-
If $VS < TS - \Delta TS$,
The vehicle is moving slowly and the experience is overly optimistic.
If $VS > TS + \Delta TS$,
The vehicle is moving too fast and the experience is overly pessimistic
Since,

$$\Delta TS \ \alpha \ TS \qquad (2)$$

, the filter will be very sensitive to change at low speeds. But it will accept a wider range of values at high speeds. This allows the filter to be more flexible.

In order to keep the value of TS up to date, it will be calculated once in every three minutes. Each 3 minute duration is called a slot. Cars 1, 2, 3 will be used to calculate the TS for the second slot. Since there are only two cars in the next slot, cars 4 and 5, the TS will be calculated with the cars of the previous slot as well. Therefore there must be a minimum of 3 cars to calculate a reliable TS value.

7, ALGORITHM TO IDENTIFY THE GRADE OF THE ROAD

The input for this algorithm is the output of the algorithm that identifies the valid data. This data is stored in the Validated Vehicles Database (shown in Fig. 1). This database contains only the grade value that was broadcast by the vehicle. All other information is stripped away. At this stage identifying an agreeable value for the Grade of the road is done by a majority vote. The number of vehicles that have broadcast a grade will be known as the *vote* of the grade.

The votes for Grade 1, Grade 2 and Grade 3 are determined. The grade with the highest number of votes is chosen. But it isn't always this straightforward. To impart some pessimism to the algorithm the number of votes for the highest voted grade and the second highest voted grade are compared. If the difference is less than three, and if the grade with the second highest vote is greater than the grade with the highest vote, then the second highest voted grade is chosen. The chosen value is assigned to the variable *maj_grade.* (As mentioned before, the higher the numeric value of the grade – highest being 3 – lower is the condition of the road). This is what the code snippet in Figure 6 performs.

We call the mean of all the grades of the Validated_Vehicles_DB as *average_grade*. Now the mean of the *average_grade* and *maj_grade* is determined and rounded up to the nearest value and stored in the variable *final*. This

value is almost always equal to that of *maj_grade*. But this step is necessary to ensure that false negatives are not introduced. The *final* value is the grade which will be updated to the map.

```
if(
((majority[order(majority, sorted$majority)[2]] -
majority[order(majority, sorted$majority)[3]]) >= -3) ## checking if
the votes of the highest and second highest voted grade are less than
or equal to 3
 &
(grades[order(majority, sorted$majority)[2]] > grades[order(majority,
sorted$majority)[3]])
## checking if the second highest voted grade is greater than the
highest voted grade) {
                    ## the second highest voted grade is chosen, since it
is more pessimistic
                    maj_grade <- grades[order(majority,
sorted$majority)[2]]}
          else{     ## the highest voted grade is chosen
                    maj_grade <- grades[order(majority,
sorted$majority)[3]]}
```

Figure.5 R Implementation to identify the grade of the road

8, CONCULSION AND FUTUREWORK

This paper displays a technique for approving large amounts of information got from the vehicle and its sensors. It utilizes two filters to dispose of temperamental information. The information that effectively goes through both the filters is utilized to refresh the Validated_DB. The Validated_DB is utilized by the TOC framework to gauge and break down the present and past grades of the street. Utilizing a calculation, the TOC framework decides the value of the grade to be updated. Once this value has been determined, the changes are updated to the Map_DB. This paper exhibits a straightforward framework to make an interconnected group of drivers utilizing crowd sourced information to improve the entire voyage experience.

REFERENCES

#1. Shilpa Garg, Pushpendra Singh, Parameswaran Ramanathan, Rijurekha Sen, "Vividhavahana: smartphone based vehicle classification and its applications in developing region" , Mobiquitous, London, Great Britain, 2014.

#2. Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden , Hari Balakrishnan, "The pothole patrol: using mobile sensor network for road surface monitoring", MobiSys'08, Breckenridge, Colorado, USA, 2008.

#3. Pushpendra Singh, Nikita Juneja, Shruti Kapoor, "Using mobile phone sensors to detect driving behaviour", Dev'13, Bangalore, India, 2013

#4. Ki-Yong Lee,, Jeong-Min Park,, Joon-Woong Lee “Estimation of longitudinal profile of road surface from stereo disparity using Dijkstra algorithm” Robotics and Automation, International Journal of Control, Automation and Systems August 2014, Volume 12, Issue 4, pp 895-903.

#5. Andreas Wedel, Hernán Badino, Clemens Rabe, Heidi Loose, Uwe Franke, and Daniel Cremers, Member, IEEE “B-Spline Modeling of Road Surfaces With an Application to Free-Space Estimation” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 10, NO. 4, DECEMBER 2009

#6. R. Labayrade, D. Aubert, and J. P. Tarel, “Real time obstacle detection in stereovision on non flat road geometry through “V-disparity” representation,” Proc. IEEE Intelligent Vehicles Symposium, pp. 646– 651, 2002H.

#7. J. Weber, D. Koller, Q.-T. Luong, and J. Malik, “An integrated stereobased approach to automatic vehicle guidance,” in Proc. 5th Int. Conf.Comput. Vis. , 1995, pp. 52–57.

#8. R. Labayrade, D. Aubert, and J.-P. Tarel, “Real time obstacle detection on non flat road geometry through ‘v-disparity’ representation,” inProc. IEEE Intell. Veh. Symp. , Versailles, France, 2002, pp. 646–651.[Online]. Available: http://perso.lcpc.fr/tarel.jean-philippe/iv02.html

#9. R. Labayrade and D. Aubert, “A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision,” in Proc. IEEE Intell. Veh. Symp. , Columbus, OH, Jun. 2003, pp. 31–36.

#10. N. Suganuma and N. Fujiwara, “An obstacle extraction method using virtual disparity image,” in Proc. IEEE Intell. Veh. Symp. , Istanbul, Turkey, Jun. 2007, pp. 456–461.

#11. F. Oniga, S. Nedevschi, M. Meinecke, and T. Binh, “Road surface and obstacle detection based on elevation maps from dense stereo,” in Proc. IEEE Intell. Trans. Syst. , Seattle, WA, 2007, pp. 859–865.