2 Comparison of travel time estimations using intelligent infrastructure and floating car data

2.1 Introduction

Travel time estimation plays a crucial role in traffic management. Having reliable, real-time measurements it is possible to inform the drivers, by Variable Message Signs or by other communication means, of expected delays. This allows drivers to choose less congested connections and results in the improvement of the overall flux in traffic networks. Travel time data could also be used to estimate traffic stream characteristics and could be transformed into reliable traffic information required for the proper functioning of modern Intelligent Transport Systems (ITS) [Hal97]; [Her+10]; [Wor+10].

There are several techniques for measuring the speed of any traffic flow. In this chapter the focus is on two of them: using floating car data and intelligent infrastructure. Those methods have not been analyzed as comprehensively for urban areas as they have been for rural areas [BLM05]. In this chapter a contribution is given comparing these methods by interpreting the measurements obtained from a traffic simulator.

The chapter is organized as follows: the next section presents the state of the art. Section 2.3 describes simulation environment used for performing tests. Then two sections explain how the reliability of the two methods has been compared. The final sections contain the results and future work.

2.2 State of the art

Travel time is defined [Tur98] as “the time necessary to traverse a route between any two points of interest.” Travel time is measured or estimated based on traffic models [LTL08]. Direct methods of measurements can be divided into two groups. The first group acquires data from sensors installed in-roadway or over-roadway like signpost systems using contactless technologies like Radio Frequency Identification (RFID), or from license

M. Markiewicz, *Reduction of CO2 Emissions from Road Transport in Cities*, DOI 10.1007/978-3-658-16319-8_2

plate recognition systems with video streams of traffic monitoring cameras, loop detectors, Global System for Mobile Communications (GSM) signaling analysis and others [Cha+11]; [Coi02]; [Kle+06]. The second group uses moving observer sensing. It includes FCD or volunteer drivers and probe vehicle methods [FRV08]; [LZT05]. Travel time obviously depends on the road curvature, congestion level and traffic signal timings and is affected by a range of unpredictable factors like changing weather conditions or traffic incidents [LZT05]. To increase the accuracy of measurements of travel time along road sections, readouts can be aggregated on turn level, to catch differences in travel time due to different turning directions [Bro+10].

In this work we measure travel times from both groups – intelligent infrastructure and moving observers – using a traffic simulator.

2.3 Simulation environment

To compare the reliability of techniques for measuring the travel time we are using *Naxos* Traffic Simulator. The developed simulator use cellular automata with Nagel-Schreckenberg traffic rules [NS92]. To speed up the simulation time it has been partially implemented using the Compute Unified Device Architecture (CUDA).

To be precise, we introduce some formalism which describes the initial condition of simulation and outcomes at each simulation step. Let $\mathcal{S} = (\rho, L, C, S, P, R, T)$ describe the initial simulation conditions, where:

- $\rho \in [0, 1]$ is a percentage of all cells occupied by vehicles (density),
- L is a type of traffic light controller used at intersections in the simulation. Possible values are: Random Cycle Length, Fixed Cycle Length and Self–Organizing Traffic Lights,
- $C \subset \mathbb{N}$ is a set of crossroads identifiers,
- $S \subset C \times C \times \mathbb{R} \times \mathbb{R}$ is a set of road segments. Any $s = (c_1, c_2, l, \alpha) \in S$ consists of two adjacent crossroads c_1 and c_2, l is the distance between c_1 and c_2 and α is the directed angle formed by the line connecting c_1c_2 and parallel. For example the value $l = (4, 6, 120, 90)$ means that there is a road segment l between crossroads with identifiers 4 and 6 of length 120 meters which lies on the meridian,
- $P \subset C \times C \times C$ contains all permitted passing possibilities. The meaning of the passing possibility will be described in detail later. Roughly speaking a passing possibility $p = (4, 1, 15)$ means that it is possible to reach the crossroads with identifier 15 by driving from crossroads 4 by crossroads 1,

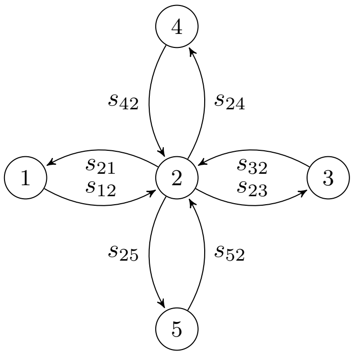
- R is a random seed value. Since all the randomness in the simulator comes from a pseudorandom generator (namely the linear congruential generator (LCG)), we need to choose a different seed each time we run the simulation to set it up properly. Simulations with the same value of seed behave identically,
- $T = \{t \in \mathbb{N} : t \leqslant k, k \in \mathbb{N}\}$ is the time of the simulation, for example $T = \{0, 1, \dots, 900\}$ indicates that the simulation is running for 900 time steps.

For a given $\mathcal{S} = (\rho, L, C, S, P, R, T)$, the *Naxos* Traffic Simulator generates as result the relation denoted by $\mathcal{R} = Sim(\mathcal{S}) \subset T \times H \times R^{n_1} \times C \times R^{n_2} \times S \times R^{n_3} \times P \times R^{n_4}$. The outcome are separate sequences of numbers for each simulation step containing numerical values of vehicle properties like current position, speed, acceleration, etc. Similarly, for crossroads and passing possibilities numerical values describe the number of vehicles passed in each direction and other statistical data. At first, we recall mathematical projection as a set operation. To avoid giving a formal definition, we describe it as following: for any two sets X and Y, any function $f : X \times Y \to X$, such as $f(x, y) = x$, $(x, y) \in X \times Y$ is called *projection* of $X \times Y$ onto X. Now, we can define sets of results obtained by the simulator more specifically:

- Projection of $\mathcal{R}$ on $T \times H \times R^{n_1}$ is the function $V_p : T \times H \to R^{n_1}$, which assigns to each vehicle identifier a n_1-tuple representing the vehicle properties. Here n_1 is equal to the number of vehicle properties. $H \subseteq \mathbb{N}$ is a set of vehicle identifiers.
- Projection of $\mathcal{R}$ on $T \times C \times R^{n_2}$ is the function of $C_p : T \times C \to R^{n_2}$, which assigns to each crossroads identifier a n_2-tuple representing the crossroads properties. Here n_2 is equal to the number of crossroads properties.
- Projection of $\mathcal{R}$ on $T \times S \times R^{n_3}$ is the function of $S_p : T \times S \to R^{n_3}$, which assigns to each segment a n_3-tuple representing segment properties. Here n_3 is equal to the number of segment properties.
- Projection of $\mathcal{R}$ on $T \times P \times R^{n_4}$ is the function of $P_p : T \times P \to R^{n_4}$, which assigns to each passing possibility a n_4-tuple representing passing possibility properties. Here n_4 is equal to the number of passing possibilities.

$\mathcal{R}$ is a database enabling the analysis of simulation results. It is formulated in mathematical terms for clarity.

The sets *SegmentProperties*, *CrossingProperties*, *PasssingPossibilityProperties* enumerate properties related to a traffic network that reflects the



	from	by	to
p_1	1	2	3
p_2	1	2	5
p_3	3	2	1
p_4	3	2	4
p_5	5	2	4
p_6	5	2	3
p_7	4	2	5
p_8	4	2	1

Figure 2.1: Passing possibilities at a crossroads with no left-turn option.

dynamics of traffic. The *VehicleProperties* set contains elements characterizing the vehicle state as: speed, acceleration, selected destination, next crossroads on route and so on.

The set P consists of all triples $(c_1, c_2, c_3) \in C \times C \times C$ such that there exist $l, l' \in \mathbb{N}$, $a, a' \in \mathbb{R}$, $s = (c_1, c_2, l, a) \in S$ and $s' = (c_2, c_3, l', a') \in S$. In other words if $p = (c_1, c_2, c_3) \in P$, then it is permitted to drive from the crossroads c_1 by crossroads c_2 to crossroads c_3 (through segments s and s' – s begins at crossroads c_1 and ends at crossroads c_2. The segment s' starts at crossroads c_2 and ends at crossroads c_3). Fig. 2.1 shows a part of a road network illustrating the concept of the passing possibility. Driving from crossroads number 1 it is possible to drive straight ahead or turn right. That is why there are only two passing possibilities starting from crossroads 1, namely $(1, 2, 3)$ and $(1, 2, 5)$. Passing possibilities describe driving rules in a road network and are used to attribute traveling times along road sections. The latter allows measurements to be taken in respect to driving directions at crossroads (compare Fig. 2.2 and Fig. 2.3).

Having explained the formalism related to simulation input and output, we can proceed to the description of methods used to measure travel times in a simulated environment.

2.4 Methods

The estimation of the travel time along road segments could be made in many ways. In this chapter we will focus on two methods:

- By collecting readouts from onboard GPS receivers which register in-

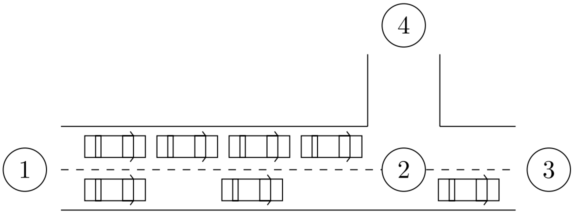


Figure 2.2: Differences in travel time along segment 1 → 2 depend on the next crossing on route. Vehicles turning left have to wait longer than those which are driving ahead.

formation of the current vehicle position. Travel time along a road segment is computed by the difference between the time when the vehicle entered and left a road segment (floating car data method – FCD).

- By identifying vehicles approaching crossroads using technologies like license plate recognition or RFID. The travel time between a pair of crossroads is computed as the difference between the time the vehicle has been unambiguously identified at the first and the second crossroads (intelligent infrastructure method – II).

The main difference between the methods is that the former gives information about travel time only between crossings which are on route of transmitting (FCD–enabled) vehicles. If the number of vehicles is too small, information from areas where the vehicles are not moving is missing. On the other hand, some vehicle identification technologies like license plate recognition are sensitive to factors like changing weather conditions, affecting the robustness of the latter method.

In the simulator the two methods of collecting data are implemented according to the following Algorithms 1 and 2.

In Algorithm 1 we are checking for each vehicle whenever it is FCD–enabled. If it is not then we proceed to the next simulated vehicle. In case it is FCD–enabled and just moving from one segment (s_{in}) to another (s_{out}), we find the passing possibility p that corresponds to this movement. Then we retrieve a property responsible for holding time when the vehicle entered s_{in} segment. If the entry time has been already set we use this value to compute the time of traveling along s_{in} by subtracting the entry time from the current simulation time. The computed value is assigned to the passing possibility p as the latest measured traveling time. Finally, the entry time property is set to the current time, as the vehicle is just entering segment s_{out}.

Algorithm 1 Floating Car Data travel time update

for each Vehicle v **do**
  if $Property(v, FCD\_Enabled) = 0$ **then**
    continue
  end if
  if not $isMovingBetweenSegments(v)$ **then**
    continue
  end if
  $s_{in} \leftarrow Property(v, InputSegment)$
  $s_{out} \leftarrow Property(v, OutputSegment)$
  $p \leftarrow GetPassingPossibility(s_{in}, s_{out})$
  $entryTime \leftarrow Property(v, EntryTimeFCD)$
  if $entryTime > 0$ **then**
    $d \leftarrow Now - entryTime$
    $Property(p, TravelTimeFCD) \leftarrow d$
  end if
  $Property(v, EntryTimeFCD) \leftarrow Now$
end for

Algorithm 2 Intelligent infrastructure travel time update

for each Vehicle v **do**
  if not $isMovingBetweenSegments(v)$ **then**
    continue
  end if
  $q \leftarrow Property(v, VehicleIdRecognized)$
  $r \leftarrow Random() < RecognitionTreshold$
  $Property(v, VehicleIdRecognized) \leftarrow r$
  if not $(q \wedge r)$ **then**
    continue
  end if
  $s_{in} \leftarrow Property(v, InputSegment)$
  $s_{out} \leftarrow Property(v, OutputSegment)$
  $p \leftarrow GetPassingPossibility(s_{in}, s_{out})$
  $entryTime \leftarrow Property(v, EntryTimeII)$
  if $entryTime > 0$ **then**
    $d \leftarrow Now - entryTime$
    $Property(p, TravelTimeII) \leftarrow d$
  end if
  $Property(v, EntryTimeII) \leftarrow Now$
end for

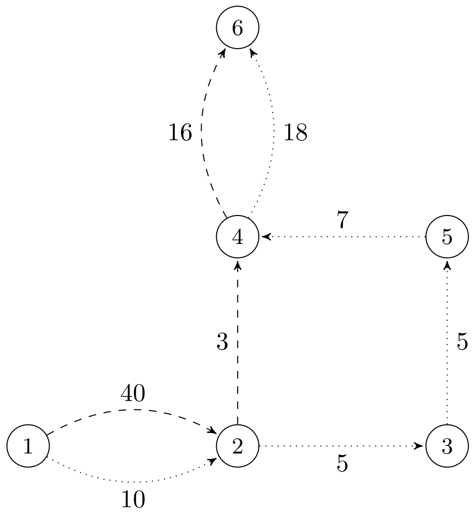


Figure 2.3: Readouts from two vehicles show differences in traveling time along the same segment depending on their routes. The numbers at the edges represent travel times (given in seconds).

In Algorithm 2 each time the vehicle is moving from one segment s_{in} to another s_{out}, it is verified if its identity was properly recognized on the previous crossroads. Test whenever the vehicle is properly identified on the current crossroads is performed by comparing a generated pseudorandom value with a given threshold corresponding to the identity detector reliability. If the vehicle has been correctly identified at two consecutive crossroads that belong to s_{in} segment, then the travel time along s_{in} segment is set to the difference between the time the vehicle has been identified at the first crossroads and the time it has been identified at the second crossroads. In this algorithm the travel time updates are performed for two kinds of passing possibilities: for all the vehicles by updating the $TravelTime$ property and for the vehicles with properly recognized identities on both crossroads by updating the $TravelTimeII$ property, the $TravelTime$ property is used as a reference comparing the robustness of both algorithms (see Fig. 2.4).

2.5 Measurements

To measure the reliability of the two methods of collecting information regarding travel time described in the previous section, two functions are

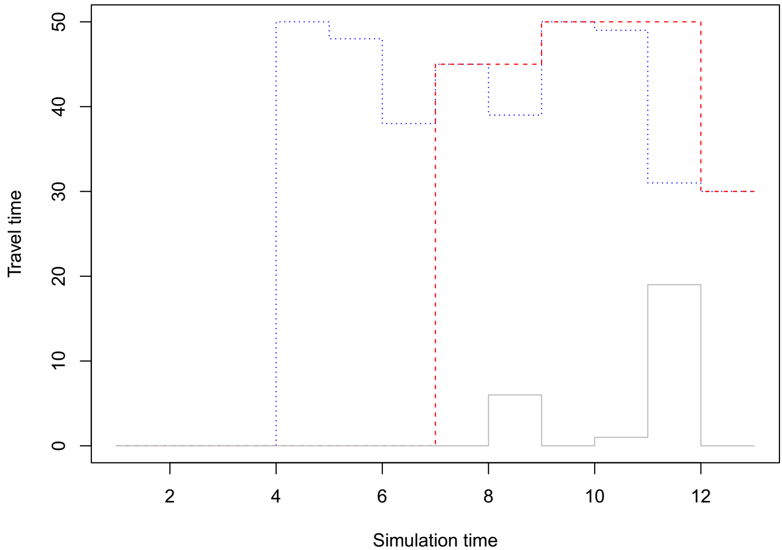


Figure 2.4: Differences in travel times along a road section. Dashed line represents travel times updates triggered by vehicles whose identities have been properly recognized. Dotted line – triggered by all vehicles. Solid line represents absolute error (values of $|f_{\mathrm{ref}} - f_{\mathrm{ii}}|$).

defined $f_{\mathrm{fcd}} : P \times T \to \mathbb{R}$ and $f_{\mathrm{ii}} : P \times T \to \mathbb{R}$. For $p \in P$ and $t_0 \in T$ the value of the function $f_{\mathrm{fcd}}(p, t_0)$ is not defined if in time period $[0, t_0]$ there was no FCD–enabled vehicle which drove along the whole s_{in} segment (we remind that $p = (s_{in}, s_{out})$). For the remaining arguments $f_{\mathrm{fcd}}(p, t)$ is equal to the last update of the travel time triggered by the FCD–enabled vehicle. The function f_{fcd} measures the absolute error of the travel time acquired from the transmitting vehicles as measured by algorithm 1.

Similarly for the function f_{ii}, for $p \in P$ and $t_0 \in T$ the value of $f_{\mathrm{ii}}(p, t_0)$ is not defined if there were no vehicles in the time period $[0, t_0]$ which have been properly recognized at crossroads at the end of the segment s_{in} ($p = (s_{in}, s_{out})$). For the remaining arguments $f_{\mathrm{ii}}(p, t)$ is equal to the last update of travel time triggered by the properly recognized vehicle at both crossroads.

To find out which function provides better estimations of travel times along road segments we need the reference function $f_{\mathrm{ref}} : S \times T \to \mathbb{R}$. It is the projection of P_p on the coordinate representing the value of the $TravelTime$ property. The value of this function is equal to the travel time of the last vehicle that passed a given segment. The $TravelTime$ property is updated each time a vehicle moves from one segment to another. This function is equal to f_{fcd} for which 100% of the vehicles are FCD–enabled and equal to f_{ii} function for vehicles with an identity recognition rated 100%.

To compare the reliability of the estimation of both functions we introduce the auxiliary function $F_{\mathrm{fcd}} : \mathcal{R} \to \mathbb{R}$ which allows for measuring both under- and overestimations of the traveling time averaged for the duration of the whole simulation:

$$F_{\mathrm{fcd}} = \frac{1}{\|T\|} \sum_{t \in T} \frac{1}{\|P_1^t\|} \sum_{p \in P_1^t} |f_{\mathrm{fcd}}(p, t) - f_{\mathrm{ref}}(p, t)| \tag{2.1}$$

the internal sum is taken of all $p \in P_1^t \subseteq P$ for which f_{fcd} and f_{ref} are defined at fixed time t. Analog it is $F_{\mathrm{ii}} : \mathcal{R} \to \mathbb{R}$ as

$$F_{\mathrm{ii}} = \frac{1}{\|T\|} \sum_{t \in T} \frac{1}{\|P_2^t\|} \sum_{p \in P_2^t} |f_{\mathrm{ii}}(p, t) - f_{\mathrm{ref}}(p, t)| \tag{2.2}$$

The internal sum is taken of all $p \in P_2^t \subseteq P$ for which f_{ii} and f_{ref} are defined at fixed time t. In case of F_{fcd} and F_{ii} the sum is taken of all passing possibilities. This means that every eligible passing possibility in the simulation environment is taken into account. Dividing this figure by the number of passing possibilities gives the average value.

2.6 Results

Simulations have been performed in a virtual city which has 100 crossroads on a grid of 10×10 connected by road segments with length of 100 cells each. All crossroads have traffic light controllers. Their cycle length is varying in range [30, 60]. It is determined by a pseudorandom numbers generator which seed is set once at the beginning of each simulation. The distribution of pseudorandom variables is uniform.

The initial positions of the vehicles and their destinations are selected randomly. To preserve a constant vehicle density during the whole simulation new vehicles are not added. The vehicles which reach their destination are programmed to return to their initial position in a cycle. The route for each vehicle is computed by the Floyd—Warshall algorithm, i.e. via the shortest path. The duration of a single simulation is 900 time steps.

For a given set of parameters 100 simulations are performed and the expected values (weighted mean) of functions F_{fcd} and F_{ii} are computed. The changing parameters are:

- ρ – vehicle density, $\rho \in \{0.1, 0.2, \ldots, 0.9\}$,
- θ – probability of detection of the vehicle identity $\theta \in \{0.0, 0.1, \ldots, 1\}$,
- η – percentage of FCD–enabled vehicles, $\eta \in \{0.0, 0.1, \ldots, 1\}$.

Simulations have shown that the fundamental difference in the methodologies of acquiring the travel time information have an impact on the distribution of the estimation accuracy. Average values of F_{fcd} and F_{ii} representing the accuracy of those methods are given in Fig. 2.5 and Fig. 2.6. The numerical results and the simulator itself can be downloaded from `https://github.com/naxos-simulator/NaxosSimulator`.

The comparison reveals different degradation ratios of robustness as parameters describing quality of readouts are decreasing. So, as the number of FCD–enabled vehicles decreases (θ the parameter for F_{fcd}) – the reliability is reduced. Similarly, as the probability of proper identification of a vehicle decreases (η parameter for F_{ii}) – the reliability of estimation also decreases. It is worth noting, that when the number of vehicles reaches a critical density, a transition from free flow to congested flow occurs, which results in the breakdown of estimations made by both methods. However, F_{fcd} seems to be better in that case, having a smaller error compared to F_{ii}. On the other hand, for probabilities of proper identification of a vehicle greater that 50% F_{ii} outperforms F_{fcd}.

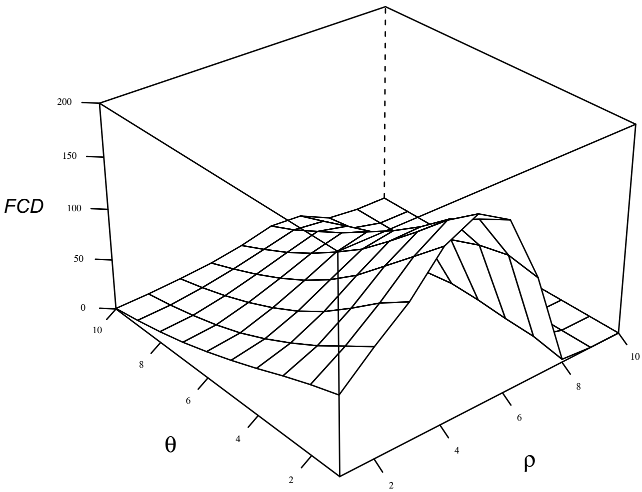


Figure 2.5: Average value of function F_{fcd}, (ρ – vehicle density, θ – percentage of FCD–enabled vehicles. 10 on the scale represents 100%).

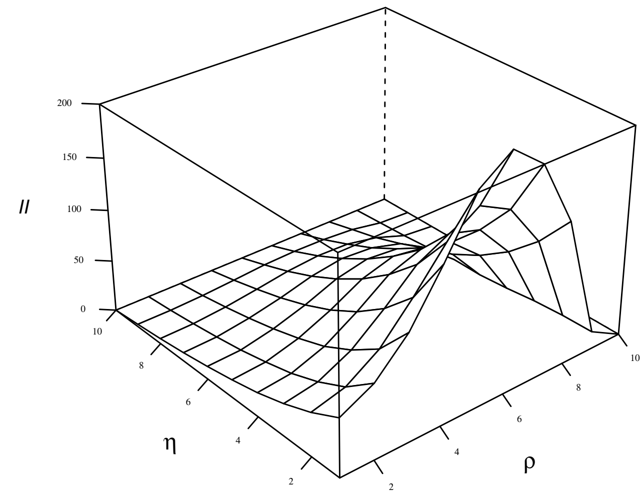


Figure 2.6: Average value of function F_{ii}, (ρ – vehicle density, η – percentage of properly recognized vehicles identities at crossings. 10 on the scale represents 100%).

2.7 Summary

The aim of this chapter was to compare the reliability of two selected methods of estimating travel time along road segments. We used a traffic simulator to compare two approaches in measuring traveling time: by tracking selected vehicles and by identifying vehicles at the end of road sections.

Simulation results show that even a small fraction of of FCD–enabled vehicles gives reasonable estimations. On the other hand, if the technology used for vehicle identification at crossroads guarantees correctness greater than 50% then the second method gives much better results.

In the real world, the selection of a method used for measuring the travel time in a city obviously depends not only on robustness of a method but also other factors like costs of installation, maintenance, and operation (mainly communication) which were not considered.

Combining both methods and investigating the usefulness of the acquired data for such applications like dynamic route guidance seems to be a good subject of future research.

http://www.springer.com/978-3-658-16318-1

Reduction of CO2 Emissions from Road Transport in Cities
Impact of Dynamic Route Guidance System on Greenhouse Gas Emission
Markiewicz, M.
2017, XXVI, 169 p. 93 illus., 21 illus. in color., Softcover
ISBN: 978-3-658-16318-1