Cyclist-aware traffic lights through distributed smartphone sensing

Theodoros Anagnostopoulos *, Denzil Ferreira, Alexander Samodelkin, Muzamil Ahmed, Vassilis Kostakos
Center for Ubiquitous Computing, University of Oulu, 90570 Oulu, Finland

A R T I C L E   I N F O

Article history:
Received 19 October 2015
Received in revised form 15 December 2015
Accepted 29 January 2016
Available online 17 February 2016

Keywords:
Smartphone sensing
Distributed model
Power consumption
Time-of-arrival estimation

A B S T R A C T

Cycling in smart cities can be safer if enhanced with a smart traffic lights infrastructure. A distributed smartphone-based sensing approach is a cost-effective infrastructure to enable cyclist-aware traffic lights system. In this article, we treat cyclist movement on a trajectory with a Boundary model able to reduce GPS sensor power consumption, while performing time-of-arrival estimation to the nearest light. A global quantitative metric of model efficiency is proposed for assessing the overall behavior of the model, and a false-positives rating qualitative metric is used to assess the recall of the model. We evaluated the model with confined yet realistic cycling experiments and verify the precision of our model using an Android application installed in participants’ smartphones. We compared our model with previous literature, achieving a promising model for in-the-wild cycling scenarios.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In early 2012, the city of Oulu in Finland successfully piloted a vehicular 3G-based communication system that monitors the position of emergency vehicles (i.e., ambulances and fire trucks) and controls in real-time the traffic light signals across the city to give priority to those vehicles. This was achieved without installing new hardware at road intersections, but solely through distributed sensing. This system has dramatically reduced the time that emergency vehicles require to reach their destination, and has significantly decreased the number of traffic accidents involving emergency vehicles. The system won the 2013 Innovation Award from the Finnish Fire Protection fund [1,2].

In this article, we describe our extension to this system to also sense cyclists and give them a green light priority. Such a system for cyclists will potentially improve enjoyment [3], reduce accidents due to cyclists crossing a red light [3], and encourage people to cycle more [4]. There exist several alternatives to sense cyclists, such as lane counters [5], cameras [6], and road radars [7] which incur costs. An economical approach is to reprogram all traffic lights to give priority to cyclists by default [8], but this method, besides being time consuming, only works well with sporadic vehicular traffic.

To sense cyclists without installing new hardware, we propose using cyclists’ own smartphones rather than specialized hardware currently used onboard safety vehicles. Evidently, this introduces a number of challenges related to power efficiency, modeling, and privacy in estimating the position, speed and direction of cyclists. If a cyclist’s mobile phone reports location, direction, and speed (obtained via GPS) to a traffic server, this is a power-hungry and privacy-sensitive operation that may demotivate cyclists from using our proposed method. To overcome both energy and privacy concerns, we

* Corresponding author.
E-mail address: tanagnos@ee.oulu.fi (T. Anagnostopoulos).
http://dx.doi.org/10.1016/j.pmcj.2016.01.012
1574-1192/© 2016 Elsevier B.V. All rights reserved.
experimentally evaluate a number of adaptive sensing strategies that attempt to reduce power needs while only revealing minimal information about an individual’s real-time location. In summary, cyclists’ smartphones compute the distance between their real-time and current location against a set of known traffic light positions (previously stored on the device), and only communicate with the server when they wish to request a “green light” at a particular intersection. The frequency of location checks increases as the cyclist approaches a traffic light, and gradually decreases when no traffic lights are nearby. Thus, a cyclist’s location is only revealed to the server when a green light priority is requested, and consequently the battery consumption is drastically reduced with this adaptive location polling mechanism.

We report on the iterative development, implementation, and assessment of an energy-efficient smartphone application that monitors a cyclist’s location and requests a “green light priority” at intersections where the cyclist is expected to be in the near future. Participants’ feedback has helped us design and improve the built-in algorithms into our application. Finally, our performance evaluation consists of both simulations and validation in the field with cyclists.

2. Related work

2.1. Cycling and traffic lights

Cycling has important positive outcomes on health [6,4] and the environment by reducing air pollution and greenhouse emissions [9]. For many cities, it is vital to promote cycling and improve urban center cycling experience. However, many studies have consistently found cyclists performing a red light infringement [10,3,11], leading to accidents and even death. For instance, 37% of Australian cyclists reported that they had ridden through a signalized intersection during a red light phase [3], while in Taiwan, the short red light duration at intersections is one of the main reasons for cyclists to exhibit risky behavior [10].

Overall, cyclists prefer cycling infrastructures that are segregated from vehicular traffic, regardless of their cycling confidence [12]. In Europe, for example, research has highlighted the need to enhance safety and mobility by minimizing the waiting time for pedestrians and cyclists at crossings; providing them with the same rights as motorized traffic; and prioritizing walking and cycling in urban areas [13]. One of the reported technical recommendations is the use of detectors that provide pedestrians and cyclists with green light priority. Of course, motorists may experience inconvenience and delays if cyclist priority is not well calibrated for the flow of cyclists.

2.2. Sensing mobility using smartphones

A number of projects have attempted to accurately reconstruct general mobility patterns by using people’s mobile devices. Mobile phone tracking has been used as an approach to measure the flows of passengers between parts of a city, and for estimating speeds and travel times [14,15]. The results typically have low spatial resolution and are most effective for long-distance segments such as highways. Lu [16] categorizes past research on geospatial analysis into three groups: (i) in a data-driven approach where spatiotemporal patterns are mined from trajectory data; (ii) research that aims to analyze and model dynamic interactions between people; and (iii) “urban studies” that focus on modeling human and vehicular flows in cities. The third category best describes the aim of our work, as we focus on techniques to establish comprehensive urban traffic mobility data from multi-modal sources.

In the past, a popular approach for sensing mobile phones has been the use of proximity-based technologies, such as Bluetooth or Wi-Fi traces [17–19]. However, this approach suffers from strong privacy concerns [20] and the need for a dense network or urban “scanners”. In our approach, we use modern smartphones’ Internet connectivity through 3G and EDGE. A minor drawback of this approach is the slight decrease battery life and additional network bandwidth costs. On the other hand, users have control over the data they provide to us, and there is no need for installation of additional sensing infrastructure.

Furthermore, the use of communication technology for capturing city-scale mobility is not new. Multiple smart-city initiatives have explored city-scale instrumentation of major infrastructure including energy networks (e.g., electricity, gas, water), building monitoring in terms of environmental and structural performance and transport including urban mobility, energy costs, and environmental impact. Some notable initiatives include IBM’s Smarter Planet (http://www.ibm.com/smarterplanet/), the European Union’s Smart Cities and Communities (http://www.eu-smartcities.eu) and the Smart Santander project in Spain (http://www.smartsantander.eu), and Korea’s Songdo city (http://www.songdo.com) to name a few. Particularly in the context of urban mobility and transport, a number of previous projects have considered ways to estimate road traffic [21], and travel patterns using cellular [22] or proximity [19] technologies, or a combination of these as in the case of the European Union’s Instant Mobility project (http://instant-mobility.com).

2.3. Estimating location using smartphones

Location prediction of cyclists is feasible using a variety of sensors, including GPS, available on smartphones. Typically, location prediction for outdoor settings is achieved either by analyzing the relative position of movement handovers within a cellular network [23], or by exploiting the recorded GPS position of a moving entity. In the latter case, analysis of GPS
position data may rely solely on GPS coordinates [24], GPS coordinates enhanced with time [25], or GPS coordinates along with velocity and direction [26]. In fact, predicting the future location of a cyclist is the first step towards predicting the time-of-arrival (ToA) to particular locations of interest, which in our case are the traffic lights of a city. The current literature on ToA estimation contains a variety of computational approaches based on: historical trajectories [27], real-time map matching [28], shared locations [29], and mobile phone participatory urban sensing [30,31]. In this paper, we focus on ToA estimation of cyclists and develop models that incorporate GPS coordinates, time, velocity and direction enhanced with bearing.

2.4. Limitations of previous work

Most urban mobility work has been demonstrated in the context of short-lived research projects and pilots. Our approach paves the way for the establishment of an open API for third-party applications who could provide a means to make cyclist sensing sustainable in the long run. Effectively, this API allows any third-party app (e.g., fitness apps, music apps, maps) to request green light priority from the local city. To the best of our knowledge, no previous work has considered the use of distributed smartphone sensing to directly influence traffic light priorities for cyclists in an urban setting.

Furthermore, while multiple strategies exist to sense cyclists, they are not without prohibitive drawbacks. For example, the city of Assen in the Netherlands has programmed traffic lights to give priority to cyclists by default, unless vehicles are present [8]. However, this approach only works well when vehicular traffic is low. The United States Department of Transportation has developed the IntelliDrive program, focused on the creation of interoperable connectivity among all types of vehicles, the traffic management infrastructure, and mobile devices [7]. However, this requires individual drivers, pedestrians and cyclists to carry specialized hardware, which is cost inefficient. Similarly, lane counters [5] and camera systems [6] have high hardware and installation costs.

3. System description

Our system consists of Android software running on cyclists’ own smartphones, and a server that receives requests from those smartphones. Every smartphone has a built-in list of traffic lights’ coordinates within the city, and the smartphone software’s objective is to predict whether the cyclist will attempt to cross a particular traffic light in the near future. When such a prediction is made, the smartphone sends a request for “green light” to the server. The server manages multiple competing requests for green lights from cyclists, pedestrians pressing the physical green light request button at the intersection, and vehicular traffic sensed using underground magnetic sensors on the driveway.

A system overview is presented in Fig. 1, showing a cyclist moving from point A to point B, and a traffic light located at point B. At point A, the GPS sensor is activated and the GPS coordinates are obtained. The system computes the average speed based on the current speed recording and the previous speed-readings, and makes a ToA estimation on when the cyclist is likely to reach a nearby traffic light (point B). Not all traffic lights are considered, but only those that lie “ahead” of the cyclist. Hence, the smartphone calculates the cyclist’s movement orientation (0°–359°), and considers only traffic lights within an arch defined by a specified angle of bearing b and radius r. By incorporating b and r the smartphones treat consequent lights as a traffic light chain, which can be turned green with only one request.

Given the estimated ToA, the smartphone decides whether to actually request a green light for the traffic light at point B, or to remain idle. A ToA estimation is successful if the cyclist arrives at point B and the traffic light is green. In this paper, we focus on three fundamental requirements for the smartphone software:

- **Privacy**: we require that the smartphone software reveals the minimum necessary location data to the server. Thus, while it would be feasible to upload mobility traces to the server, we require that the smartphones should only send
a request for “green light” to the server. Such a request does reveal some location information: by implication we can assume that the cyclist must be near the traffic light for which she requests a green light. However, this is the minimum information necessary that the cyclist can reveal to the system.

- **Power efficiency**: we wish to minimize the energy consumption of our software, thus it is important to optimize the sleeping time of the software. This will allow us to minimize the number of times the software requests the GPS location from the operating system, and the number of requests made via the network. In practice, our software only uses network connectivity to send the request for “green light” to the server.

- **Precision & recall**: we wish to maximize the probability that when a cyclist approaches a traffic light it will be green. At the same time, we want to minimize the chance that a cyclist requests a “green light” for a particular intersection that they never cross.

To operationalize our analysis, we must make two assumptions. First, we assume that the once the server obtains a request for a green light, it will be able to actually turn the light green after 15 s. Thus, a request is not fulfilled instantly but after a fixed threshold of 15 s. In practice, this value adapts based on the actual traffic conditions, provided by the server’s access to the current driveway traffic sensors. While our analysis used a fixed threshold, a dynamic threshold is easily calculated for each traffic light separately, and communicated to all smartphones in real-time. Second, we assume that once a traffic light turns green, it will remain green for the duration of 15 s. Once again, this duration may adapt to actual traffic conditions, and may differ for each traffic light separately. Again, the server could make real-time estimations of both these and pass them on to all smartphones in real-time. We would like to note that our model adapts to the threshold used. This covers cities where such driveway traffic sensors are not available.

### 3.1. Metrics

To evaluate the performance of the system, we propose several quantitative and qualitative metrics. Power consumption \( c \) is a quantitative metric and it is defined as the normalized number of GPS sensor invocations. Specifically, assume \( c_T \) to be the actual number of GPS sensor invocations for a certain experiment and \( c_{\text{max}} \) be the maximum number of GPS sensor invocations of all the experiments then \( c \) is defined as in Eq. (1):

\[
c = \frac{c_T}{c_{\text{max}}}.
\]  

Accordingly, ToA estimation \( p \) is another quantitative metric and it is defined as the normalized number of the accurate ToA estimation. Specifically, let \( p_T \) be the actual value of ToA estimation for a certain experiment and \( p_{\text{max}} \) be the maximum value of ToA estimations of all the experiments then \( p \) is defined as in Eq. (2):

\[
p = \frac{p_T}{p_{\text{max}}}.
\]

Let us give a numerical example for ToA estimation \( p \). Assume that for a certain experiment a cyclist comes across 20 traffic lights, and our ToA estimation is correct for 15 of them. Therefore, for this particular experiment we calculate that \( p_T = 15/20 = 0.75 \). Let us also assume that across all our experiments, the highest success rate we observed was \( p_{\text{max}} = 0.95 \), i.e. there was at least one experiment a ToA estimation success rate value equals to 0.95. Then the value of \( p \) is defined as \( p = \frac{0.75}{0.95} = 0.79 \), which is the normalized success rate value of that experiment.

Assume weight \( w \) be a normalization parameter used for weighted sum. We define the quantitative metric efficiency \( e \) as in Eq. (3):

\[
e = w \cdot p + (1 - w) \cdot (1 - c)
\]

to be the global metric of an optimum solution. We also assume \( w = 0.3 \) since we are more interested in low \( c \) and less in high \( p \).

A qualitative metric \( f \) measuring the false positives is incorporated. It measures how many lights, which turned green, should not have been turned green. Specifically, \( f \) is defined as the fraction of false positives \( f_p \) over the sum of true positives \( t_p \) and false positives \( f_p \), as in Eq. (4):

\[
f = \frac{f_p}{t_p + f_p}.
\]

Intuitively, high values of bearing and radius will result to more false positives since a larger number of traffic lights is considered for requesting green, but low values of bearing and radius may reduce system accuracy.

### 3.2. Time of arrival models

By considering the privacy, efficiency, and accuracy requirements of the system, we have developed 5 models assigned to 3 main categories whose primary purpose is to send green light requests to the server, and to put the system to sleep. Specifically, the first model has three variations. The models employ different strategies to optimize GPS sensing. All models were evaluated against the metrics previously presented. The models are executed constantly on the background.
Table 1
The boundary model.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Input</strong>: ToA, interval, threshold</td>
</tr>
<tr>
<td>2</td>
<td><strong>Output</strong>: sleep</td>
</tr>
<tr>
<td>3</td>
<td><strong>While</strong> (light NOT reached)** Do **First Condition</td>
</tr>
<tr>
<td>4</td>
<td>boundary = interval · threshold</td>
</tr>
<tr>
<td>5</td>
<td>If (ToA &gt; boundary)** Then</td>
</tr>
<tr>
<td>6</td>
<td>sleep = \ \frac{ToA}{interval}</td>
</tr>
<tr>
<td>7</td>
<td>Else If (ToA ≤ boundary &amp; ToA &gt; (threshold + interval²))** Then **Second Condition</td>
</tr>
<tr>
<td>8</td>
<td>sleep = interval²</td>
</tr>
<tr>
<td>9</td>
<td>Else</td>
</tr>
<tr>
<td>10</td>
<td>sleep = 1</td>
</tr>
<tr>
<td>11</td>
<td>End If</td>
</tr>
<tr>
<td>12</td>
<td>End While</td>
</tr>
</tbody>
</table>

**Constant sleep time (Naïve model)**

In this model, the smartphone is re-activated after a fixed number of seconds in order to determine ToA and/or to send data to the server. We use three different Naïve model variations, based on the constant sleep time of 1, 2 and 10 s. Thus, we have Naïve, Naïve-2 and Naïve-10 models respectively.

**Logarithm**

This model computes sleep time as a function of ToA estimation to the nearest light using a logarithmic curve. Specifically, for higher ToA estimations between the cyclist and the nearest light, the model calculates a higher sleep time value. However, sleep time is quickly reduced when ToA estimation to the traffic light is lower, and for low ToA estimations the sleep time has a constant value of 1 s. Because this model uses the base 2 logarithms, we called it the Log2 model.

**Boundary**

This model encapsulates variations in velocity (see Table 1, line 4). The model has inputs: (i) ToA, (ii) interval, and (iii) threshold and has output the sleep time of the GPS sensor. Effectively, the model considers three broad zones away from the traffic light, and adopts a different behavior for each zone. The zones are calculated by taking into account that a cyclist may exhibit variations in their velocity.

4. Evaluation

4.1. Simulation with cycling traces

The proposed models were evaluated by experiments applied on real datasets. We use four different datasets (Fig. 2) in order to run our simulations. A volunteer of the University of Oulu created the first dataset, which is located in the northern suburbs of the city of Oulu in Finland. The dataset covers a rural cycling area from the city center of Oulu to the university’s campus. The trace contains 970 GPS positions with minimum coordinates of latitude and longitude to be 65.05771 and 25.45637 while the maximum coordinates are 65.08508 and 25.47221, respectively. The coverage area of the trace is 14.87 square kilometers while the distance covered by the cyclist within this area is 4947 m. The average velocity of the cyclist is 18.4 km per hour (i.e., 5.1 m per second).

The second dataset covers a rural cycling area from the city of Dinkelhausen to the city of Hameln in Germany [32]. The length of the trace is 2256 GPS positions with minimum coordinates of latitude and longitude to be 51.66477 and 9.376016 while the maximum coordinates are 51.10197 and 9.678354, respectively. The coverage area of the trace is 385.7 square kilometers while the distance covered by the cyclist within this area is 12408 m. The average velocity of the cyclist is 19.8 km per hour (i.e., 5.5 m per second).

The third dataset covers an urban area of cycling within the city of Pluneret in France [33]. The length of the trace is 2738 GPS positions with minimum coordinates of latitude and longitude to be 47.66605 and —2.969214 while the maximum coordinates are 47.68369 and —2.950897, respectively. The coverage area of the trace is 11.05 square kilometers while the distance covered by the cyclist within this area is 11773 m. The average velocity of the cyclist is 15.5 km per hour (i.e., 4.3 m per second).

Finally, the fourth dataset covers an urban area of cycling within the city of Barrow in United Kingdom [34]. The length of the trace is 4189 GPS positions with minimum coordinates of latitude and longitude to be 54.1080 and —3.2349 while the maximum coordinates are 54.1635 and —3.1819, respectively. The coverage area of the trace is 102.8 square kilometers while the distance covered by the cyclist within this area is 18850 m. The average velocity of the cyclist is 16.3 km per hour (i.e., 4.5 m per second).

4.2. Field trial

We developed a mobile application that incorporates the best-performing model, and tested it in a field trial under realistic conditions. For safety reasons, we did not use real traffic lights in junctions with vehicular traffic, but used emulated
traffic lights. The application consists of several components (Fig. 3) that communicate with a server using JSON over HTTP POST/GET requests. The server is also used to create our experimental test setup by storing traffic lights' coordinates and test settings, and collecting the telemetry data from the cyclist phone for further analysis. Telemetry data is event-based and sent by the cyclist's phone when its state changes (e.g., GPS turned on or off, GPS coordinates changed, request for traffic light sent). The data gathered from the cyclist phone consist of the following fields:

- **Server timestamp**: time when data was received.
- **Smartphone timestamp**: device time when data was collected.
- **GPS status**: status of GPS receiver to be on or off, received coordinates, accuracy.
- **Cycling information**: estimated speed and bearing.
- **Traffic light information**: closest traffic light, distance to traffic light, traffic light status.
- **Algorithm information**: calculated sleep time, calculated time to light.

Participants were handed a phone with a pre-installed application (Fig. 4(a)) that allows them to start the service and begin the experiment. Data is also collected from the researcher’s phone, who cycles behind the cyclist for observation. The researcher application (Fig. 4(b)) provides a button for indicating the moment when the cyclist actually passes the traffic

---

**Fig. 2.** Datasets: (a) dataset 1, (b) dataset 2, (c) dataset 3, (d) dataset 4.

**Fig. 3.** Software application overview.
light, thus establishing ground truth. This is required to obtain exact timing and status of the lights. Those cannot be collected with just the cyclists’ phone, because of the GPS accuracy and power saving algorithms, which may result in GPS off state when passing the due light.

In our experiment the traffic lights were emulated by tablets in portable kiosks (Fig. 5) attached to existing road signs (Fig. 13). The tablets either glow red or green to represent different traffic light states. Every second, the traffic lights pull from the server the current traffic light state and update their color as necessary.

We also developed a web UI (Fig. 6) that allows a researcher to plan the experiment by setting up the positions of the traffic lights on a map. The traffic lights that are placed on the map are saved into the database and pushed to participants’ smartphones. Also, the user interface allows the researcher to define the values for interval and bearing for a particular experimental run.

When an experimental trial begins, the software on participants’ smartphones operates as shown in Fig. 7. This uses the previously described Boundary model to adapt sleeping behavior, but also takes into consideration the time required by the GPS sensor to acquire higher location accuracy, which can vary. This delay is known as Time-To-First-Fix (TTFF) and should be taken into consideration when scheduling the receiver wake-up time. The time required to obtain a fix depends on the required precision, as shown in Fig. 8, based on our own testing in the field. Prior to our field trial, we considered TTFF in
the range of 9 to 13 s, and were able to obtain the best results when TTFF is set to 11 s. For this reason, the model in Fig. 7 introduces a delay of 11 s to obtain a reliable GPS fix. This time is effectively subtracted from the sleep time.

5. Results

5.1. Simulation results

We evaluate the proposed models under several simulated conditions and across all 4 datasets. Specifically, we varied the number of traffic lights on the trace between 1 and 10 lights per kilometer, and the lights around the trace between 20 and 80 lights per square kilometer. The first set of lights was supposed to be turned green by our models, while the second set of lights were decoys and should not be turned green.

We also tested the effects of different bearing and radius values (that define the arc ahead of the cyclists), ranging between 1° and 15° bearing, and 61 and 85 m radius, for all datasets. The radius values were chosen to roughly equate to 15 s’ traveling time. Finally, for the Boundary model we tested different threshold values in the range of 15–25 s, and interval values in the range of 2–4 s.

To identify optimal values, we used the efficiency metric $e$ (Eq. (3)) which takes into account both accuracy and energy consumption. The results show that the optimal values of bearing for datasets 1 & 2 is 4°, while for datasets 3 & 4 is 5°. Similarly, the optimal radius value for dataset 1 is 77 m (arch length is 5.37 m, i.e. $\text{arch length} = \frac{77 \times \pi \times 360}{360}$), for dataset 2 is...
Fig. 8. GPS Time-To-First-Fix accuracy as a function of time. Different colors represent accuracy (in meters). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

83 m (arch length is 5.79 m), for dataset 3 is 65 m (arch length is 5.66 m) and dataset 4 is 68 m (arch length is 5.93 m). Furthermore, we found that the optimal threshold value for datasets 1 & 2 is 22 s, while for datasets 3 & 4 is 19 s. Finally, the results show that the optimal value for interval across all datasets is 2 s.

In Fig. 9 we show the expected power consumption of our models, across the 4 datasets and across the varying simulated conditions. In Fig. 10 we show the accuracy of each model across the same conditions. Since power consumption and accuracy are a tradeoff, in Fig. 11 we show the efficiency of each model, which incorporates both power consumption and accuracy (Eq. (3)). Finally, Fig. 12 shows the rate of false positives for each model (Eq. (4)).

Assessing the overall performance of the models across all simulated conditions, we find that the Boundary model outperforms others in terms of efficiency. While the naive models have high accuracy, they also use a lot of power by making frequent GPS requests. On the other hand, the Boundary model is able to sense variations in velocity due to the interval time quantity, and thus does not sacrifice accuracy in favor of precision.

The four different datasets in our tests represent very different contexts both in terms of speed, urban layout, and settings. Despite these differences, our results do not substantially change between datasets for each tested metric. This provides high confidence for our simulated results, since they seem to hold across different datasets.
Finally, the effect of traffic light density does seem to be important. As the number of traffic lights per kilometer increases, we find that the behavior of our models vary and/or converge. For instance, we find that as the number of traffic lights increases, then power consumption goes up and accuracy drops. We also find that as the number of traffic lights on the trace goes above 7 per kilometer (representing dense urban settings), then the efficiency of Boundary, Log2, and Naive10 converge. Hence, this appears to be an important density threshold for model performance. Similarly, we find that as the number of decoy traffic lights goes up, then false-positives increase.
5.2. Field trial

We evaluated our system in a field trial at the University of Oulu campus. Specifically, we defined a trajectory of 2500 m length and we inserted within it 8 kiosks emulating traffic lights, as shown in Fig. 13. To assess false-positives rate we also enter in the system the locations of 8 decoy lights outside of the expected trajectory.

We recruited 5 participants (3 male) from the university via mailing lists, aged between 20 and 30 years old. All participants were experienced cyclists and familiar with the area of the experiment. Each participant answered a questionnaire before the experimental task, and upon completion of the tasks. Each participant was compensated with a movie ticket.

The participants were first briefed about the purpose of the experiment, and given instructions about how to proceed. During the trials, a researcher was cycling behind the participants to observe the experiment and also collect ground truth data. In early pilot testing we realized that a cyclist’s behavior converged after completing our circuit more than 2 times, since they became aware of the trajectory and its peculiarities. Thus, participants were asked to complete at least 2 laps (5 km), and two participants chose to complete more laps on their own initiative.

Fig. 12. False positives rate: (a) dataset 1, (b) dataset 2, (c) dataset 3, (d) dataset 4.

Fig. 13. Kiosks emulating traffic lights.
Table 2
Field trial results.

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>c</th>
<th>p</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Lap 1)</td>
<td>0.40</td>
<td>0.875</td>
<td>0.68</td>
<td>0.250</td>
</tr>
<tr>
<td>1 (Lap 2)</td>
<td>0.45</td>
<td>0.875</td>
<td>0.65</td>
<td>0.000</td>
</tr>
<tr>
<td>1 (Lap 3)</td>
<td>0.50</td>
<td>0.875</td>
<td>0.61</td>
<td>0.000</td>
</tr>
<tr>
<td>1 (Lap 4)</td>
<td>0.39</td>
<td>0.875</td>
<td>0.69</td>
<td>0.250</td>
</tr>
<tr>
<td>1 (Lap 5)</td>
<td>0.33</td>
<td>0.875</td>
<td>0.73</td>
<td>0.125</td>
</tr>
<tr>
<td>2 (Lap 1)</td>
<td>0.50</td>
<td>0.625</td>
<td>0.54</td>
<td>0.000</td>
</tr>
<tr>
<td>2 (Lap 2)</td>
<td>0.52</td>
<td>0.750</td>
<td>0.56</td>
<td>0.000</td>
</tr>
<tr>
<td>2 (Lap 3)</td>
<td>0.53</td>
<td>0.750</td>
<td>0.55</td>
<td>0.000</td>
</tr>
<tr>
<td>3 (Lap 1)</td>
<td>0.34</td>
<td>0.750</td>
<td>0.69</td>
<td>0.125</td>
</tr>
<tr>
<td>3 (Lap 2)</td>
<td>0.39</td>
<td>0.750</td>
<td>0.65</td>
<td>0.125</td>
</tr>
<tr>
<td>4 (Lap 1)</td>
<td>0.36</td>
<td>0.750</td>
<td>0.67</td>
<td>0.000</td>
</tr>
<tr>
<td>4 (Lap 2)</td>
<td>0.39</td>
<td>0.875</td>
<td>0.69</td>
<td>0.125</td>
</tr>
<tr>
<td>5 (Lap 1)</td>
<td>0.35</td>
<td>0.750</td>
<td>0.68</td>
<td>0.125</td>
</tr>
<tr>
<td>5 (Lap 2)</td>
<td>0.29</td>
<td>0.750</td>
<td>0.72</td>
<td>0.125</td>
</tr>
</tbody>
</table>

In our experiment, bearing was experimentally set to 11° to account for the round nature of our circuit (Fig. 6). This value is double the optimal value reported in our simulations, because in our case the circuit was circular and relatively short. Also, the interval was set to 2 s according to our simulation results.

During the trials, we observed that participants’ cycling behavior varied according to their physical condition, as expected. Specifically, the time needed to complete one lap varied between 553 and 896 s. The lowest speed we observed was 9.94 m/s while the highest was 16.27 m/s, and as a result the model’s radius varied between 70.24 and 105.41 m and threshold varied in the range of 19 to 22. These variations are due to the varying cyclist behavior, which are captured ad hoc by our Boundary model.

In terms of the performance of our system (Table 2), we found that power consumption ranged between 0.33 and 0.53, accuracy (ToA estimation) varied between 0.625 and 0.875, efficiency varied between 0.54 and 0.73, while the false positives rate varied between 0 to 0.25. The discrepancy between these values and our simulations are largely attributed to the latency in GPS TFFF.

From the pre and post questionnaires we collected data about participants’ experience, attitude, and perception of the system. We use our questionnaire and interview findings to enrich our discussion. Overall, our participants did not encounter any significant issue with the system, and would consider using such a system.

6. Discussion

The main objective of our work is to demonstrate the feasibility of building traffic lights that can rely on smartphone sensing to give priority to cyclists. Cities are increasingly investing in intelligent traffic infrastructure, and a key motivation is to provide safety to the cyclists [35]. It is often the case that even legislation in such cities can be adapted to provide bicycle-friendly policies when an instrumented bicycle infrastructure is available [36]. While central to safety is the usage of helmets, it has also been reported that cyclist traffic lights decreases the crash risks of cyclists [37]. In addition, cycling attitude and traffic behavior of cyclists are important factors in gaining public acceptability of transport policy measures [38]. Hence, in order to provide advanced services to cyclists, our work seeks to provide the feature of ToA estimation to the nearest traffic light. By incorporating ToA, the system regulates the traffic in the road and gives priority to cyclists in the intersections.

To achieve this, our work first adopted a simulation approach to evaluate a number of alternative models that can help balance power consumption and accuracy, while also minimize privacy concerns by minimizing the information that clients reveal to the server. Our simulation analysis identified the Boundary model as the most efficient across a variety of settings and simulation conditions. This model was subsequently tested in a field trial. The results of the field trial are positive, and the system performed efficiently in these realistic conditions, and is comparable to other similar systems. As expected, the field trial also gave us a number of insights about the feasibility and performance of reacting to cyclists by relying on smartphone sensing.

6.1. Effectiveness of the model

There exist several techniques for providing efficient ToA estimation, which is the approach we have adopted. Certain models for ToA estimation of vehicles use shared traveler's GPS positions to collect traffic conditions, and can reach accuracy of 95% [29]. Similarly, using historical and real-time GPS data can lead to high accuracy levels (95%) [27]. It is also possible to use participatory sensing, for example to estimate ToA for buses [31]. This approach can incorporate more generally available and energy efficient sensing resources, such as cell tower signals, movement patterns and audio recording, but suffers in terms of accuracy by reaching levels between 82% and 94%. Compared to these models, our Boundary model has an accuracy of 95% on synthetic cycling traces (in our simulations) and 62.5% to 87.5% during our field trials.

Compared to the performance of those models, our model performs on par on synthetic data. In our field trial, we achieved lower accuracy since GPS sensor is not constantly active during the cycling. However, when we consider previous work
that also focuses on energy efficiency [31], our model performs better. This is because previous work has used context from multiple sources of high space complexity, while our approach incorporates sparse GPS locations due to the boundary model. In addition, much of the performance of previous work [31] can be attributed to the incorporation of bus routes’ historical, which is not available in our field trial.

6.2. Performance in the field trial

We observed that during the field trial our system was robust to short turns, hills and speed variations. However, there were some deviations from the simulation results, and this is attributed due to GPS TTIFF inefficiency. Specifically, there were two conditions that lead the model to perform poorly. First, high-speed variations affect the ToA estimation, especially when near a traffic lights. This results to a cyclist missing the traffic lights (i.e. getting a red light) due to high sleep values or to increased GPS usage due to low sleep values. Especially in cases where participants were cycling up hill, their speed tends to slow down and the speed system measurements during the up hill lead the model to sleep for long time, which then resulted to traffic lights loss.

Second, when the participants took a sharp turn while the GPS was offline, then the system measured their distance as distance between two points on the trajectory. In that case the computed distance by the model was shorter that the distance of the actual trajectory, which lead to traffic light loss due to late system signaling. However, traffic lights located within a certain arch were signaled. The arch had variable size due to speed variations. For a cyclist to get a green light, the request should be sent from the cyclist’s phone to the server sometime between 15 and 25 s in advance. In our experiment, successful signaling time was between 19 and 21 s, which is similar to the values observed in our simulations.

These results suggest that a way to further improve the performance of the system, without sacrificing privacy by profiling users [20], is to account for the environment. Specifically, since our model rapidly increases the GPS sensing frequency as the cyclists approaches the traffic light, it may be possible to infer whether the cyclist had to stop or not at the junction, thus indicating success or failure. This data can be aggregated per traffic light anonymously, and used to dynamically adapt the model for each individual traffic light. With such data, it is possible to adapt the model for traffic lights that may be placed in challenging locations, such as after a sharp turn or at the top of a hill. This approach is effectively the same as the one used by Bluetooth or Wi-Fi scanning projects [17–19] which passively model mobility at each individual hotspot location.

6.3. Cyclist–light interaction

In our experiment, we observed that despite the relatively small number of participants, they “interacted” with the traffic light in different ways, thus leading to changes to our model’s performance. For instance, some participants cycled confidently and would slow down only when they were relatively close to the lights, while others would slow down at a relatively far distance from the traffic light. This change of cycling pace affected the performance of the system, which effectively delayed the green light as long as participants were slowing down. In this sense, we observed that the “confident” participants ultimately had higher success in getting a green light.

In general, all participants consciously attempted to avoid a red light, and claimed to be annoyed if they had to stop cycling. As part of their strategy, they all noted that they wanted to know if the system request was sent from their phone. In other words, they wanted feedback from the system on whether it has sent a request to the traffic light. However, providing such feedback to cyclists can be challenging. While it is possible to inform them that a request has been sent to the traffic light, in many cases it is extremely hard to inform cyclists exactly when the light will turn green. Traffic junctions consider a large number of parameters and sensors (such as push-buttons, vehicular sensors, controller priority) when deciding to switch a light green or red. Thus, while our system can provide input to such a traffic junction controller, it is hard to get feedback from them early enough.

6.4. Smartphones as sensing platforms

There exist several techniques to sense cyclists, and potentially provide extremely accurate readings without concerns about power efficiency. Such systems include lane counters [5], cameras [6], and road radars [7]. Automatic detection of bicycle traffic flow has been achieved by inductive loop frequency detector systems based on pattern recognition and classification models [39], reaching a counting accuracy of 85%. Video object tracking and image processing has also been used for bicycle and pedestrian detection by performing feature-based reasoning on the visual data [40], performing bicycle detection with 94% accuracy. Computer vision systems embedded in road intersections are also able to detect and classify bicycles and vehicles based on monocular image sequences processing [41] with accuracy of 95.1%. CCTV cameras located in roadside are used to feed Gaussian Mixture Models (GMM) and Support Vector Machines (SVM) in order to perform bicycle and vehicle detection, tracking and classification of urban traffic flow [42] with 94.69% accuracy. Automated data collection models using video sensors at road intersections are used to classify bicycles and pedestrians based on Histogram of Oriented Gradients and Support Vector machines [43]. The results of the bicycle’s classification accuracy is 93.3%.

However, an important limitation of these systems is that they incur substantial installation and maintenance costs, because they require hardware to be purchased and installed at each junction, and possibly at multiple orientations
to account for traffic flow. Another limitation is that they only capture the location of the bicycles without any other information such as velocity, direction and time of arrival estimation to the nearest traffic light while our system is able to provide this information to the infrastructure. However, a key benefit is that they can account for all cyclists, and cyclists do not need to carry any equipment.

While a key limitation of our system is that it requires cyclists to carry a smartphone, it also has substantial benefits. An important benefit is that no physical installations are needed at traffic junctions, and much of the maintenance costs are borne by cyclists, who are already motivated to take care of their phone if it breaks down.

Thus, an important challenge in our approach to raise and maintain cyclists’ interest in using a system like ours. We have identified two strategies for achieving this. First, there exist popular deployment channels for reaching cyclists: existing application stores such as Google Play. In fact, cyclists already use a variety of software such as MapMyRide [44], the Move! Bike Computer [9], Endomondo Sports Tracker [9], MyTracks [9], Cyclometer GPS [45], Cycle Tracker Pro [46], BiCycle [47] and Cycle Watch [48]. This means that use of our system can simply piggy-back on top of the existing practices of cyclists, who are increasingly interested in collecting data about their cycling behavior, or simply listening to music or using maps to obtain directions.

Second, it is possible to consider publishing the API specifications for our servers, and allowing third-party applications to easily integrate with a city’s traffic-light infrastructure. Thus, instead of us publishing a standalone application that implements our model and sends request to the traffic light server, it is technically possible to build that functionality into popular applications that cyclists already use, such as the ones above. In fact, this is something that could readily be incorporated into navigation and map applications, such as Google Maps. Since those applications already monitor participants’ movement, they could bind to a predetermined API and make requests for a traffic light to become green, similar to how they offer recommendations for calling Taxis or booking Restaurants.

In fact, during our interviews, most participants claimed that the system functionality was extremely useful, and claimed that they would in fact actively try to share the results of the experiments with their friends through social networks. This approach is well-aligned with the increasing number of smart-city projects, such as IBM’s Smarter Planet (http://www.ibm.com/smarterplanet/) or the Smart Santander project in Spain (http://www.smartsantander.eu), in the sense that a smart-city can provide an API that third-party developers can utilize. Effectively, this approach can create a sustainable ecosystem whereby cities invest in infrastructure (smart traffic lights and API) that then creates an economy of products and services (third-party apps), which in turn generate revenue and tax income.

6.5. Limitations

One issue we have not touched upon in our work is whether it is “fair” to give priority to cyclists, and whether this would actually result in increased traffic congestion in any given city. In our particular case, the model and system we have developed are one of the many traffic management tools that the Traffic Control Center of our city has at its disposal to manage traffic. Other tools include magnetic-based vehicle detectors to infer the presence of cars, and pedestrian push-buttons to request green lights at an intersection. With our system, it is now possible to also account for cyclists in the city. However, the decision to priorities one traffic flow over another, e.g. cyclists over cars, rests with the operators at the Traffic Control Center who dynamically respond to traffic requirements by manipulating these priorities at traffic lights. These operators attempt to maximize traffic throughput by manipulating traffic priorities at intersections.

Another limitation is that our model does not handle multiple requests from different directions. In the current version we assume the case where we have one cyclist cycling in a certain direction. In addition, the model does not handle the case of multiple cyclists who request green in the same direction. We intend to extend the current system in order to incorporate these scenarios, taking into account the privacy issues that they give rise to. Specifically, dealing with these scenarios requires to keep historical records of cyclists’ movements, which in turn gives rise to privacy concerns.

7. Conclusions and future work

This paper presents an effective, power efficient, and privacy-preserving mechanisms for enabling traffic lights to sense and react to cyclists. We have developed a model that optimizes these concerns, and appears to be robust to both simulations and a field trial. Our model effectively considers various dynamic boundaries around a traffic light, which adjust to cyclists’ speed. The benefits of making traffic lights responds to cyclists are wide-ranging, and can potentially increase enjoyment [3], reduce accidents due to cyclists crossing a red light [3], and encourage people to cycle more [4].

There are two main directions that our future work is taking. First, from an HCI perspective, we have begun to further explore the impact of providing feedback to cyclists. Particularly, our field trial showed that cyclists form different mental models and strategies for getting a green light, and we expect that any form of feedback will further effect their strategy. Thus, we are considering ways to identify the optimal time and method for providing feedback to cyclists. Second, from a modeling perspective, we are considering ways in which our system can take into account groups of cyclists. To optimize traffic flows, it would be ideal if our system could nudge cyclists to form groups, which then cross a junction at once. Thus through a combination of modeling and feedback nudging of cyclists it may be possible for our system to decrease the gaps between cyclists and further optimize traffic light signaling.
Acknowledgments

This work is partially funded by the Academy of Finland (Grants 276786, 285062, 286386), and the European Commission (Grants PCIG11-GA-2012-322138 and 645706-GRAGE).

References


