Enhancing in Vehicle Digital Maps via GPS Crowdsourcing

Abstract—In this paper, we propose a simple and effective method to extend digital maps with the location and timing of stop-signs and traffic lights in a city, given GPS traces collected by on-road vehicles. Our system finds the location and timing of traffic lights and stop-signs using a small number of traces per road segment. We developed and evaluated the system by applying it to two sets of GPS traces —open street map, a multicity, publicly available database of GPS traces, and a set of traces collected in west Los Angeles. Evaluation results show that our system can estimate the location/type of stop-signs with as little as 5 traversal per road segment and the location/timing of traffic lights with as little as 7 traversals, with more than 90% accuracy. The core of our system can be summarized as the sequence of the following basic tasks:

Map Information Extraction and processing: the raw files containing the digital maps are parsed and the information is homogenized and loaded into efficient data structures. Trace Selection: the GPS traces are preprocessed discarding the ones that present too many discontinuities and changes in sample frequency, and location errors thus reporting as moving a vehicle lined-up at the traffic light. We cope with such artifacts by performing a time/location correlation analysis on the GPS traces to identify unique patterns on each street-segment and intersection that indicate the likely location and timing of stop-signs and traffic lights. Furthermore we show that the proposed algorithms yet work with missing data, correctly identifying the location of traffic lights and stop signs using only the data relative to a subset of the roads forming the intersection. For evaluation, we applied our method to two different trace collections; the Open Street Maps public database that contains more than 300,000 GPS traces world wide, and a set of traces we collected in the greater Los Angeles, area.

I. INTRODUCTION

Digital Maps are part of our daily activities—they convey vital information for a plethora of location-enhanced applications such as Yelp, Flicker, Facebook, Google Maps or, simply, location-aware searches. PDA and smartphone users are progressively more dependent on GPS and location based services (LBS) for both in vehicle, and pedestrian navigation [1]. Map services are also evolving in nature from uni-directional to bi-directional —the street topology is completed with dynamic information, including traffic and user-created contents. Current market trends show an increasing need of up-to date geo-referenced information as a key component in the design of mobile systems such as networked vehicles, participatory sensing, mobile social networks, or personal fitness [2][3]. The Android based Google Maps-Navigation, for instance, is one of the first prosumer based navigation services for smartphones. Participating users upload GPS probes onto google servers. User-provided content is used in the traffic estimation together with more traditional data sources such as roadway sensors, cellular, video, etc. [4][5]. The current business model, based on few companies that pay up-front for the hefty costs to build the world geographical database is no longer able to sustain the increasing demand for new data attributes such as realtime traffic, points of interest, user reviews etc. Navteq and Teleatlas, the two major map providers for both in-dash and on line navigation services are forced to rely gradually more on user provided data [6][7]. Similarly, on-line and in-dash navigation providers such as Google and TomTom are using customer provided data to increase the data accuracy and definition while cutting data-licensing costs and reducing the dependency from cartography providers [8]. Finally, consumers gained an essential role in the process of building open-source digital maps as demonstrated by the Open Street Map project (OSM), an on line-site designed to build a free editable map of the whole planet derived from GPS traces uploaded by users worldwide [9]. Updatable digital maps are poised to become an enormous enabler of highly dynamic and interactive mobile systems that piggyback on few user-provided information to convey a plethora of advanced applications to a large base of consumers [10]. Future navigation systems, for example, may employ dynamic extended attributes on stops signs, traffic lights and traffic status, to perform a multi-commodity optimization on carbon footprint, commuting time, and traffic flow at the same time.

In this paper we show how GPS data collected from personal and in-vehicle devices can be used to extend digital maps with the location of stop signs, traffic lights and relative timing —these information are today rare and local to each municipality. We present an algorithm that can extract useful topology information from such collective GPS traces, and show that even a tiny fraction of vehicles so equipped can enable the extension of digital maps with new attributes. GPS devices introduce inaccuracies and discontinuities in the data, making the identification of map features more difficult. For instance, the sampling may suffer from sudden discontinuities, changes in sample frequency, and location errors thus reporting as moving a vehicle lined-up at the traffic light. We cope with such artifacts by performing a time/location correlation analysis on the GPS traces to identify unique patterns on each street-segment and intersection that indicate the likely location and timing of stop-signs and traffic lights. Furthermore we show that the proposed algorithms yet work with missing data, correctly identifying the location of traffic lights and stop signs using only the data relative to a subset of the roads forming the intersection. For evaluation, we applied our method to two different trace collections; the Open Street Maps public database that contains more than 300,000 GPS traces world wide, and a set of traces we collected in the greater Los Angeles, area.

II. SYSTEM DESCRIPTION

The core of our system can be summarized as the sequence of the following basic tasks:

Map Information Extraction and processing: the raw files containing the digital maps are parsed and the information is homogenized and loaded into efficient data structures. Trace Selection: the GPS traces are preprocessed discarding the ones that present too many discontinuities and changes in the sampling frequency. Trace Mapping: the GPS traces are mapped onto the basic geographic maps through an advanced reverse geocoding process applied on a dynamic time-based window instead of a single GPS point. Stop-sign and traffic light information extraction: a time-space pattern analysis on the resulting dataset extrapolates the position of stop signs and traffic lights.

In the following we describe in detail each part of our system.
A. Map Preprocessing

The best representation of a digital map is a directed graph, that consists of a set of nodes (i.e. the intersections) and oriented edges (i.e. the road segments). This representation is indeed one of the most used by most map databases built to represent the physical road layout for navigation and display purposes. However, the mere physical representation of the topology is not always suitable for our feature extraction algorithms. We need a logical representation of the road layout; in particular, we need to redefine the intersections and roads as atomic entities that can be directly used in the feature extraction. For example, intersections of separated road[1] represented by two separated edges, would be stored in a digital map database such as the census TIGER database[11] as two different nodes. However, urban planners and traffic engineers consider these intersections as a single intersection and it are indeed regulated by a single traffic regulator (stop sign or traffic light). We store the digital maps in dictionaries using a logical representation in which the road system is seen as a set of intersections and a set of oriented ways that interconnect them; intersections and ways are abstracted as atomic and may include several physical nodes and edges respectively. While this representation can be obtained starting from any map database, for the sake of this study we considered two very well known databases: TIGER Census[11] and OpenStreetMap[9].

B. Trace Selection

OpenStreetMap provides a large amount of user generated GPS traces. However, the wide variety of consumer GPS devices results in a large number of traces that do not meet the minimum requirements for our algorithm to extract useful information from them. The most impacting feature is the sampling rate. Many of the traces that we analyzed are affected by a highly variable sampling interval. A variable sampling interval is most likely due to a poor GPS signal, as many commercial GPS devices do not output any position information when it is not possible to obtain a fixing for the current point. For each trace in the dataset we compute the most common sampling interval (SI); we then define a sampling band as shown in equation 1

$$\frac{1}{2} SI < SI < \frac{3}{2} SI$$

We discard the traces that do not have at least 80% of the samples contained within the sampling-band. In essence we only consider traces that have a fairly constant sampling rate. Furthermore, in order to extract user-behavior such as slow-down and standstill vehicles we need a relatively high frequency sampling. Longer sampling periods result in an inaccurate estimation of the speed of the vehicle. Therefore we discard all traces that have $SI > 4$. Finally we also discard traces that are not recorded by vehicles moving in their normal environment, like traces recorded by pedestrians, easily recognizable analyzing the maximum speed.

C. Trace Mapping

Once the traces dataset is cleaned from unstable and unreliable traces as described above, it is possible to map each GPS trace onto the underlying road system. Our approach involves an efficient reverse geocoding algorithm for the mapping of each single point of the trace, and an advanced correction algorithm that takes into account the actual space-temporal relationship among subsequent points of the same trace.

The reverse geocoding algorithm maps each GPS point to the closest road segment. This process performs a lookup over the entire street database to find the road segment that is closest to the current GPS coordinate. We optimized the lookup pre-ordering all the streets and intersections by longitude/latitude. There are several drawbacks in evaluating each point in a GPS trace as independent entity. In particular:

- GPS traces are affected by bursty positioning errors. In fact, as the car moves, the surrounding environment (buildings, vegetation etc.) might shade some of the satellites, causing a local positioning error[12]. These local errors might result in a wrong reverse geocoding location.
- In digital maps, roads are represented as segments, thus with no information about their width. Therefore, in proximity of an intersection, the position of a vehicle traveling on a straight road may be geocoded into the crossing road[13][14]. In fact, the car would be moving aside of the segment representing the road is traveling on. At intersections, the segment representing the crossing road will be geometrically closer than the actual traveled road, as shown in figure 1.

![Common reverse geocoding failure approaching an intersection.](image)

For the above reasons it is crucial to exploit the temporal-space relationship between GPS samples belonging to the same trace. We implemented a reverse geocoding correction algorithm that considers both spatial and temporal information. Our algorithm scans each trace two times, first forwards and then backwards. In both scans we take advantage of the previous reverse-geocoding results, thus the first scan propagates forward information about the past and the second scan propagates backward information about the future. For the sake of simplicity, we here detail the forward scan, as the backward scan just operates inverting the indexes. The scanning algorithm takes different actions if the vehicle is moving or if it is standstill, we will discuss how to distinguish between these two cases in the next section.
For each GPS sample our mapping procedure considers a set of candidate roads ranked by distance from the point coordinates. If the vehicle is standstill, the algorithm maps the current position on the same road of the previous position, if such road is in the current candidate road-set. Instead, if the vehicle is moving, the algorithm will try to map the vehicle, among the candidate road-set, on a street oriented as the vehicle movement, giving highest priority to the road selected by the previous match, and decreasing priority, according to the orientation, to the streets connected to the previous match. If none of these is a possible i.e. the candidate road-set does not contain either the previous match nor a road connected to it, the algorithm returns null and enters the seek for temporary match mode. In this mode, the algorithm compares the closest road to the road most closely oriented as the vehicle movement. If they are the same (i.e. same road) and remain the same for 3 consecutive samples, the last match becomes effective and the seek for temporary match mode is over. The remaining null mapping points will then be fixed by the backward scan if possible.

After the two scans we further process the trace to identify temporal and path holes. Indeed the traces might still have some missing samples or errors. In such cases, if the two sides of the temporal hole are matched on roads that are not connected to each other (indicating a path hole) we split the trace into two new ones. Finally, we proceed removing all points that could not be matched on the road system and the points that are matched onto a freeway. In this study we are seeking stop signs and traffic lights that are not present on freeways. In particular, if a trace contains freeway and non freeway components (i.e. a trace moving on surface road, going on a freeway and then back on surface roads), we identify each component and split the trace accordingly, removing the freeway points.

D. Trace Processing

The ultimate goal of this work is extracting the location and timing of traffic regulators such as stop signs and traffic lights in a urban road system. In order to achieve this goal we need to initially classify the drivers behavior to build a set of abstractions that can be used in the identification of road regulators. In particular, we initially focus on the following basic behaviors:

- Slow downs: the vehicle speed decreases.
- Standstills
- Turning choices at each intersection.

Slow Downs: For the purpose of this study we only consider slow downs that happen in proximity of an intersection. In fact a vehicle can slow down for any number of reasons, but it is only at intersections that is forced to slow down due to road or traffic signals. We estimate the speed of the vehicle at each trace point as the derivative of the movement between the point itself and the next one. We consider a vehicle as slowing down if its speed falls under 5m/s. If this slow down happens within 50 meters of an intersection (that the vehicle will traverse) we save it as related to the way the vehicle is traveling on and the upcoming intersection. 

Standstills: Standstill situations recognition might seem a very simple operation, however the GPS error turns it in a very hard task. As shown in figure 2 a vehicle that is standing still will record a trace that is in continuous movement. For this reason a simple evaluation of the speed is not sufficient, as the speed will always be greater than zero. Through a set of experiments we measured that a fixed GPS device returns geographical coordinates that are within 10 meters of its actual position. We designed a heuristic to determine standstill situation that takes into account GPS artifacts. The algorithm is divided into two steps. The first step recognizes all “candidates” for standstill selecting the points of the trace characterized by a speed lower than 4m/s. In the second step we analyze these selected dataset using a time window of 10 seconds. If possible the time window is centered on the point we are analyzing. This is not possible at the beginning and at the end of the sequence of “candidate” points, in which case we use only the future or the past points respectively. If the distance between the two extremes of the time windows is smaller than 20 meters we consider the point as a standstill.

Turning Choices: The traces at this point are mapped on to the road network. However, there might still be some errors in the sequence of ways that the trace is mapped on. To correct these errors we scan the trace considering each way and intersection they are going through. Considering sets of two intersections at a time we are able to remove wrong mappings. In fact, a vehicle can move from one intersection to another using only the way that is connecting them. Then if any other mapping is present in between the traversal of two intersections we correct it and map those points onto the way that is connecting the two intersections. By performing this operation we can also extract statistics about the turning choices at intersections (left, right turn or straight).

E. Feature Extraction

Slow-downs and a standstills, defined in section [1,3] pave the road for a further level of abstraction; slow-downs and standstill events are indeed the foundation on which we can build the traffic regulators identification algorithm. The traffic regulators that we focus on are traffic lights and stop signs. We designed a set of heuristics derived from on-the-road observations and traffic regulations. These heuristics take into consideration the fact that the GPS traces may contain artifacts,
are affected by inaccuracy and have a finite sampling rate. We separated the two procedures for the extrapolation of stop signs and traffic lights. In particular, for each intersection we first look for a stop sign regulator, and, only if the probability of being a stop sign is below threshold, we check if it is regulated by a traffic light. Due to missing or incomplete data both heuristics might not be able to provide a clear result, in such cases we provide our “best guess” through heuristics that consider some basic traffic rules. In the following we describe each procedure in detail.

**Stop Signs:** Field observations, and the traffic law, suggest that all cars approaching an intersection with a stop sign will slow down, stop for a few seconds and then start moving again. This assumption, though is not confirmed by the GPS traces dataset. A first explanation is the actual behavior of drivers: most drivers do not exactly stop, but they just slow down in proximity of the intersection. Secondly, as discussed in the previous section, even when the vehicle is actually standstill the GPS output may be misleading. Finally, the trace sample rate implies a lower bound to the length of the period that we can recognize as a standstill. For instance, a sampling period of 1 second (that is actually the best we could find in the dataset) only allows to detect stopping times of at least 2 seconds. For these reasons we use the slow downs informations as defined in section II-D as indication that a vehicle is approaching an intersection with a stop. For each intersection we analyze the GPS traces on its incoming ways. We mark a way as regulated by a potential stop if at least SST of the traces slows down. We define SST as the Stop Sign Threshold and we found its optimal value to be 80% (see section II-B). The obtained per-way information is aggregated to include all the ways connected to the same intersection. In a crossroads regulated by stop signs, at most 2 ways could have the right of way, while all the others have to yield, regardless of how many they are. Therefore if all or all but one ways belonging to a given intersection are marked as potential stops, the intersection is identified as stop-sign regulated, and all the ways marked with a potential stops become actual stop signs. If all but two ways are marked as potential stops the algorithms is not able to make an immediate decision and refers the intersection to the traffic light recognition algorithm. On the road observations show that at the intersections between main roads and secondary roads the traffic lights are usually enhanced with induction loops that only switch the light to green on demand for the secondary road. As a result, most vehicles approaching this kind of intersection from the secondary road will have to stop. To make sure we are not confusing this kind of intersection with a 2-way stop case, the decision is deferred.

**Traffic Lights:** The behavior of vehicles approaching a traffic light depends on if the light is actually red or green. Analyzing the traces approaching a traffic light, as expected, we could observe that only a fraction of vehicles stops. This fraction depends on the ratio between the green and the red light duration; an information very hard to have a priori. We mark a way as a potential traffic light if at least TLT of the vehicles approaching the intersection comes to a stop\(^2\). We define TLT as the Traffic Light Threshold and we found its optimal value to be 15% (see section II-B). If an intersection is regulated by a traffic light all its incoming ways should comply with this requirement. However, due to the limited amount of traces that we have per each intersection, and to the randomness of the driver behavior, we consider an intersection regulated by a traffic light if half plus one of the incoming ways are marked as potential traffic light.

A special case is a potential two-way stop intersection, detailed in the previous paragraph. The intersection is marked as regulated by a traffic light only if all the ways not classified by the stop-sign identification mechanism satisfy the requirements to be potential traffic light. In all other cases we mark it as a two-way stop intersection.

Once the intersection is marked as a traffic light we then proceed estimating the red light duration on each way. For this purpose, we select the 95\(^{th}\) percentile of the standstill times as the duration of the red light, thus discarding peculiar cases such as left turns, parkings etc.; as also suggested in [15].

**Dealing with Missing Information:** If an intersection could not be classified either as regulated by a stop sign or regulated by a traffic light it might be because some information is missing. If actually one or two ways coming into the intersection do not have traces traveling on them we reprocess the intersection. Both the heuristics described above are executed one more time but this time the ways with missing values are not considered, i.e. we evaluate the intersection as if the ways with missing traces would not exist. The results of this further processing are marked as best guesses, as we do not have sufficient information to provide a certain estimation. However, as shown in the evaluation section, these guesses still perform fairly well.

### III. Evaluation

#### A. Characteristics of The Dataset

To assess the performance of our system we tested it on two sets of traces: traces extracted from OpenStreetMap and a set of traces that we collected driving vehicles equipped with GPS for several days in the greater Los Angeles Area.

**OpenStreetMap:** OpenStreetMap is a free editable map database of the whole world. It is built by the contribution of users from all over the world. Conceptually, users upload their GPS traces and as an option they can edit the roads they have traveled on. Afterwards, system administrators check the result of the editing and validate it. This approach guarantees an always up-to-date map at virtually no cost. In addition, all the GPS traces that are uploaded are available to other users for download. At the time of writing there were 307,000 traces on the database, but this number keeps growing as more and more users join the community. In Table IIare reported the number of traces that traverse some of the regions where OpenStreetMap users are most active, in USA and Germany. Indeed, Germany is the most active country, and its digital map is for the most part created by users. In USA, instead, the map is a derivation of the TIGER database with updates performed by the users.

\(^2\)For Coming to a Stop we refer to the standstill as defined in section II-D.
Although the number of traces is fairly large, their geographical coverage is far from global. In figure 3, we show the cumulative distribution of intersections as a function of the number of traces that traverses them. We can observe that 95% of the intersections are traversed by less than 10 traces. These numbers get worse if we consider how the traces traverse each intersection. In figure 4, we show the number of intersections that are traversed by a minimum number of traces on all (N) incoming ways, all but one (N−1) and all but two (N−2) of the incoming ways for the region of Oberbayern, Germany. We can observe that the number of intersections that have at least 5 traces on all ways is close to 0. Furthermore, the data show that most of the intersections are largely traversed through the same way, that is usually a primary road. To partially cope with this our algorithms provide a best guess performing an analysis with missing data.

![Fig. 3. Cumulative distribution of intersections as a function of the number of traces traversing them.](image1)

![Fig. 4. Distribution of intersections as a function of the number of traces traversing them on varying number of ways for the Oberbayern region.](image2)

### TABLE I

DISTRIBUTION OF OPENSTREETMAP TRACES OVER THE MOST ACTIVE REGIONS OF USA AND GERMANY.

<table>
<thead>
<tr>
<th>Region</th>
<th>Number of Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oberbayern, Germany</td>
<td>37,795</td>
</tr>
<tr>
<td>Los Angeles County, California</td>
<td>25,64</td>
</tr>
<tr>
<td>Marion County, Indiana</td>
<td>11,33</td>
</tr>
<tr>
<td>Orange County, California</td>
<td>8,76</td>
</tr>
<tr>
<td>Santa Clara County, California</td>
<td>8,48</td>
</tr>
<tr>
<td>Travis County, Texas</td>
<td>8,45</td>
</tr>
</tbody>
</table>

The CALI Dataset: As discussed in the previous paragraph, using only the OpenStreetMap traces, there are very few intersections that are covered by a sufficient amount of traces to be considered. Moreover, on these few intersections the traces are unevenly distributed, as most traces traverse the main roads and just a few traverse secondary roads. For evaluation, we had to know the ground truth of GPS traces on traffic regulators. Thus, we decided to collect a new, more complete dataset of traces to find what is the lowest amount of information that our system needs to provide confident results. We collected GPS traces in two different areas of California. The first area consists of 28 intersections of which 3 regulated by a traffic light and 25 by stop signs—we label this dataset as CALI-I. The second area consists of 4 subsequent intersections all regulated by traffic lights — this dataset is labeled as CALI-II. We designed our mobility pattern in such a way that the traces would be evenly distributed over all the intersections in the two areas. In order to minimize potential interference in the data acquisition we asked some of our students to drive the instrumented vehicles in three different week days on October 26-29, 2009 and November 26, 2009. To capture an extensive time window the vehicles have been driven from 7am to 12.30pm and from 2pm to 6pm in all the experiments.

### B. Experimental Results

In this section we present the results of the experimental evaluation of the proposed system. We first present an analysis on the sensitivity of the proposed system to the setup parameters. We then evaluate the robustness of the system against missing data. Finally, we test the overall performance of the system on the OpenStreetMap database for both traffic regulator recognition and estimation of the red light duration. We present the performance of our system only for the greater Los Angeles Area. Nevertheless, we tested our algorithms on other geographical areas, but we were not able to verify our results and therefore to present them.

**Sensitivity to Setup Parameters:** The proposed system involves the use of a set of parameters that determine the overall performance. In this section we present an analysis of the impact of these parameters on the performance, proving that the values chosen for these parameters are actually the optimal ones. In the following we focus on the two most impacting parameters:

- Stop Sign Threshold (SST): the threshold of minimum portion of slow downs needed to mark a way as a way regulated by a potential stop (see section II-E).
- Traffic Light Threshold (TLT): the threshold of minimum portion of standstills needed to mark a way as a potential traffic light (see section II-E).

We selected a set of 61 intersections in the Los Angeles county. Among these intersections, 25 are regulated by stop signs and 36 are regulated by traffic lights. We first investigate the impact of SST by running our algorithm varying SST in the interval [0.5, 1] and fixing TLT = 0.15. Figure 5 shows the portion of correctly recognized traffic regulators using this setup. In particular, we show the correctly recognized stop signs, traffic lights and the total aggregate success ratio for all intersections together with the portion of intersection for

---

3 Centered at: 34°2’48.15”N, 118°26’2.92”W
4 Centered at: 34°1’4.65”N, 118°29’35.48”W
which the system could not determine the traffic regulator. We can observe that for low values of SST the system correctly recognizes all stop signs, but at the same time performs poorly on traffic lights. This is due to the fact that the system processes as traffic light candidates only the intersections that are not marked as regulated by stop signs. Then for low SST many intersections will be erroneously marked as regulated by stop signs and not considered as traffic lights. For higher values of SST, the system starts failing to recognize stop signs. Consequently, a higher portion of traffic light is correctly extrapolated, but at the same time, the portion of intersections that can not be assigned any regulator grows as well. We find the best trade-off at SST = 0.8.

We investigate also the impact of TLT by running our algorithm varying TLT int the interval [0.05, 0.5] and fixing SST = 0.8. Figure (c) shows the portion of correctly identified regulators as a function of TLT. In particular, we present the portion of correctly identified stop signs and traffic lights, the resulting overall performance on all regulators and the portion of intersections that could not be assigned either a stop sign or a traffic light. We can observe a trend that is inverse to the one related to SST. Indeed for a growing TLT the system identifies correctly more and more stop signs and always less traffic lights. At the same time, the portion of intersections that could not be assigned any regulator increases. We can find the best trade-off at TLT = 0.15.

**Fig. 5.** Sensitivity to Setup Parameters: Portion of correctly recognized regulators for varying SST with fixed TLT = 15%.

**Fig. 6.** Sensitivity to Setup Parameters: Portion of correctly recognized regulators for varying TLT with fixed SST = 80%.

**Fig. 7.** Robustness to missing data results on the CALI data set: Average number of successful extrapolations for intersections regulated by (a) stop signs, (b) by traffic lights and (c) for all intersections, as a function of the number of traces used on each incoming way, for all (N), all but one (N - 1), all but two (N - 2) ways: (d) Value of the threshold that is used in the discrete case for both traffic lights and stop signs, as a function of the number of traces used on each incoming way.

**Robustness to missing data:** We tested the robustness of the system against missing data on all the intersections in the CALI dataset. We consider only intersections that are traversed by at least 10 traces on all incoming ways. All 32 intersections in the CALI dataset fulfill this requirement. We
performed the tests using information on all, but one and all but two ways coming into the intersection. In all three cases we run the algorithm 100,000 times randomly selecting a subset of incoming ways. In each run we use a fixed number of randomly chosen traces on each selected way. Figure 7 shows the results obtained on the CALI dataset. In particular figure 7(a) shows the average number of successful extrapolations for intersections regulated by stop signs. The system successfully recognizes the traffic regulator in more than 90% of the cases if we use all or all but one incoming ways. If we use all but two ways the performance drops but it is still around 80% that is a very good result as we have information on only half of the incoming ways (all intersections are 4-way). It is remarkable that the performance is better in the case of all but one ways than the performance obtained using all ways. This is easily explainable if we consider that when one way is missing information that way is not considered by the algorithm, thus the traces that do not slow down on that way are not taken into account. Figure 7(b) shows the average number of successful extrapolations for intersections regulated by traffic lights. We can observe a drop in the success ratio with 3 and 8 traces. This drop is due to the discretization of SST. As shown in figure 7(d) the actual value of SST in the discrete case, is different from the optimal 80%. In particular, for 3 traces, SST discretized is equal to 0.67. Such a low SST results in a very good performance for the stop signs, but a poor performance for traffic lights, as shown in figure 5. The same conclusion can be drawn for the case of 8 traces. Figure 7(c) shows the aggregated result of the success ratio over all the intersections. The system can correctly extrapolate the traffic regulator in more than 90% of the cases with as low as 5 traces on each way.

**Overall Performance Evaluation:** Having proved that the proposed system can cope with missing data we evaluate its performance over the complete dataset. In the area relative to the CALI dataset we can achieve a 100% accuracy. In fact, the system can correctly recognize the 25 intersections regulated by stop signs and the 7 intersections regulated by traffic lights. In addition, the system provides “best guesses” on 17 intersections surrounding the considered area, that actually match the real traffic regulators of such intersections. Figure 8 shows a snapshot of the map of area 1. An additional layer shows the type of traffic regulator assigned to each intersection: white hexagons represent stop signs and purple circles represent traffic lights.

As discussed in section III-A in the OpenStreetMap dataset there are very few intersection that offer a good set of traces. For this reason we have been forced to select the 65 intersections for which we have at least 5 traces on all but one incoming ways. Out of these 65 intersections the system recognizes correctly 51 of them leading to a success rate of 78%. The lower performance is due to the missing data in one way and the overall quality of the GPS traces on the remaining ways.

**Red Light Duration Estimation:** Finally we present the results of the evaluation of the red light duration. In table 11 we report the measured duration of the red light and its standard deviation together with the estimate provided by our algorithm. Our algorithm estimates the red light duration as the 95th percentile of the stop time of all traces incoming from each particular way. For this reason the estimate reflects a higher bound to the red light duration. In addition we can observe that the red light duration is always overestimated by about 5 seconds. This overestimation is due to the way our algorithm recognizes the vehicles stop times (see section II-D). Since we are using a 10 seconds window, the vehicle could still be slowing down for the first part of the time window and stop in the second part. Therefore the stop times are likely to be overestimated and consequently the red light duration. This is easily fixable introducing a simple correction offset to account for the error introduced by the time-window estimator.

<table>
<thead>
<tr>
<th>Intersection (Direction)</th>
<th>Average Time (σ)</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (EW / WE)</td>
<td>17.75(3.2)</td>
<td>24</td>
</tr>
<tr>
<td>A (NS / SN)</td>
<td>42.48(5.98)</td>
<td>51</td>
</tr>
<tr>
<td>B (EW / WE)</td>
<td>24.36(0.44)</td>
<td>29</td>
</tr>
<tr>
<td>B (NS / SN)</td>
<td>33.52(0.29)</td>
<td>36</td>
</tr>
<tr>
<td>C (EW / WE)</td>
<td>22.49(5.26)</td>
<td>31</td>
</tr>
<tr>
<td>C (NS / SN)</td>
<td>22.52(7.48)</td>
<td>27</td>
</tr>
<tr>
<td>D (EW / WE)</td>
<td>22.35(3.7)</td>
<td>29</td>
</tr>
<tr>
<td>D (NS / SN)</td>
<td>23.50(1.64)</td>
<td>28</td>
</tr>
</tbody>
</table>

Table II
Average Red Light Duration Measured 10 Times Compared to the Estimate of Our Algorithm. σ is the Standard Deviation.

**IV. Related Work**

Feature extraction from GPS traces has been researched since the the first GPS devices hit the markets aiming at the extraction of information on user habits as well as environmental features; most the research contributions, however, focus on Traffic status and lane extraction techniques.

For example, in [15] and [12] the authors present data mining algorithms for the refinement and enhancement of existing map information. Information such as number of lanes and intersection structure could be useful for safety application such as aided lane departure or collision prevention. However this information represents high-cost/low-gain additional detail.
and the mapping companies are not willing to invest on its gathering. Therefore the use of low-cost collective GPS traces seems to be the best solution. In [18] the authors propose an automated procedure for the inference of the map information itself. Such procedures might be very useful in order to avoid human intervention in the creation of the maps. Similar approach have been used by popular sites such OpenStreetMaps[9].

Recently many research studies focused on extracting the Traffic Status GPS traces. In particular, in [15] the authors identify congestion situation on the road using GPS traces and a model based on historical data. Similarly in [19] the authors exploit GPS traces to infer the travel behaviors of swedish drivers as first step to predict the traffic conditions on the road. In [20], the authors propose the use of roadside units to detect the road congestion and a GPS/GSM assisted navigator to re-route the traffic in less congested areas.

In [21] the authors present a technique for the association of traffic controls to the driver behavior. The purpose of this work is very similar to the one we are presenting in this paper. However the approach is significantly different. It involves the use of machine learning techniques and specifically neural networks thus requiring an initial training and limiting the ability to perform on-line recognition. Changes in the dataset structure require a new training since the authors did not establish a formal relationship between the driver behavior and the various factors taken as inputs. Finally the authors do not look at traffic light cycles and performed both the training and test phase using a small ad hoc dataset. Our work, in contrast, perform an analysis based on the trace key features and exploits them in discovering traffic regulators location and characteristics; additionally our system has been designed to work on-line as well as off-line, the data can be indeed updated while the vehicles are still on the road. Finally, our algorithms allows to dynamically exploit changes in the data structure (i.e. new stops, new traffic in another way etc). We tested our algorithms against a large public dataset of GPS traces on different cities in different continents and the results show that the proposed methods performs well in the majority of scenarios.

V. CONCLUDING REMARKS

In this work we presented a simple yet effective method to extract location and timing for traffic regulators such as stop-signs and traffic lights with just 5 and 7 traversals respectively. The evaluation shows that our methods perform well also with missing data identifying with an accuracy higher than 85% the type and timing of the traffic regulator even if data is missing on one of the ways belonging to the intersection.

We learnt that the quality of the GPS dataset is key to perform feature extraction on mobile vehicles. In particular, variations of the sampling rate and the sampling frequency itself are key to infer useful information from cars on the road.

This paper provides a proof of concept for the realization of a system for processing large amount of real time traces extracting useful information. The location of stop signs and traffic lights are only an example of useful information, they could be extended to commuting times, real time traffic updates, turning statistics etc.

REFERENCES


