Citywide Estimation of Traffic Dynamics Via Sparse GPS Traces

Weizi Li, Dong Nie, David Wilkie, and Ming C. Lin, *Fellow, IEEE*

***Abstract*—Traffic congestion is a perpetual challenge in metropolitan areas around the world. The ability to understand traffic dynamics is thus critical to effective traffic control and management. However, estimation of traffic conditions over a large-scale road network has proven to be a challenging task for two reasons: first, traffic conditions are intrinsically stochastic; second, the availability and quality of traffic data vary to a great extent. Traditional traffic monitoring systems that exist mostly on major roads and highways are insufficient to recover the traffic conditions for an entire network. Recent advances in GPS technology and the resulting rich datasets offer new opportunities to improve upon such traditional means, by providing much broader coverage of road networks. Despite that, such data are limited by their spatial-temporal sparsity in practice. To address these issues, we have developed a novel framework to estimate travel times, traversed paths, and missing values over a large-scale road network using sparse GPS traces. Our method consists of two phases. In the first phase, we adopt the shortest travel time criterion based on Wardrop's Principles in the map-matching process. With an improved travel-time allocation technique, we have achieved up to 52.5% relative error reduction in network travel times compared to a state-of-the-art method [1]. In the second phase, we estimate missing values using Compressed Sensing algorithm, thereby reducing the number of required measurements by 94.64%.**

***Index Terms*—Travel Time Estimation, Map Matching, Compressed Sensing, GPS Data, Sparsity**

I. Introduction

Traffic is ubiquitous in modern cities, and it impacts the social, economic, and environmental development of the world. Ever-present gridlock and congestion challenge transportation researchers and urban planners. According to the 2015 Urban Mobility Scorecard [2], traffic congestion causes an extra 6.9 billion travel hours and 3.1 billion gallons of fuel consumption in a year. The congestion costs are approximately $160 billion in the United States, more than £13 billions in the United Kingdom, and over one trillion dollars worldwide. As more and more metropolitan areas around the world experience increasingly severe traffic conditions, the ability to analyze and understand traffic dynamics is becoming ever more crucial.

Traditionally, measurements of traffic states are collected through dedicated sensors such as loop detectors and video cameras [3]. While these sensors produce relatively accurate measurements, the high expenditures for installation and maintenance prevent them from being deployed over a large network and cause them to be used in limited locations on major roads and highways. Consequently, the lack of sensing infrastructure for arterial streets—which comprise the majority of a city—has made the traffic monitoring task substantially more difficult.

So far, dispersed probe vehicle reports (i.e., GPS traces) are the most promising data source in estimating citywide traffic dynamics. However, such data are of limited usefulness for two reasons: first, inevitable errors in measurement and transmission often yield reported locations off the road; second, due to energy and privacy concerns, GPS data commonly have a *low sampling rate*, meaning that the time gap between consecutive reports is large (e.g., $> 30s$), and a *low penetration rate*, meaning only a small portion of the traffic population is participated in sending reports.

Together these features introduce a large degree of uncertainty to the data. In order to accurately infer traffic dynamics from such noisy measurements, several steps are required. First, off-the-road GPS points need to be mapped onto the road network, and the probes' true traversed paths need to be identified. This process is called *map-matching*. Second, the time taken to travel each road segment must be accurately estimated. Because of the *low sampling rate*, the identified path between two nearby GPS points is likely to cover multiple road segments, and only the aggregate travel time (i.e., the difference between GPS timestamps) is known. Consequently, the aggregate travel time needs to be decomposed and distributed to individual road segments. These operations are performed in *travel time allocation*. Third, in order to understand the full traffic dynamics of a road network, traffic data are needed over longer periods of time for each road segment. However, GPS data often do not provide complete temporal coverage for a road segment. In particular, they are scarce in the late night and early morning hours. Currently, the missing information is filled in through *missing value completion*.

Many efforts have been made towards alleviating the aforementioned problems over the recent years. To be specific, in reference to *map-matching*, the data's low sampling rate introduces problems; two consecutive reports are likely to be spatially far apart, and there may be many paths connecting the two reports in a complex urban environment, making the identification of the true traversed path difficult. To alleviate this issue, many approaches have been proposed that use the shortest-distance criterion [1], [4], [5], [6], [7], [8]. While the shortest-distance assumption is viable when a road network is under or close to the free-flow condition, this assumption induces bias in a congested environment, where other paths can be traveled in a shorter time than the shortest-distance

Manuscript received October 5, 2016; revised November 27, 2016. The authors would like to thank National Science Foundation, US Army Research Office, and UNC Arts & Science Foundation.

W. Li, D. Nie, D. Wilkie, and M. Lin are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599 USA e-mail: (weizili,dongnie,wilkie,lin@cs.unc.edu)

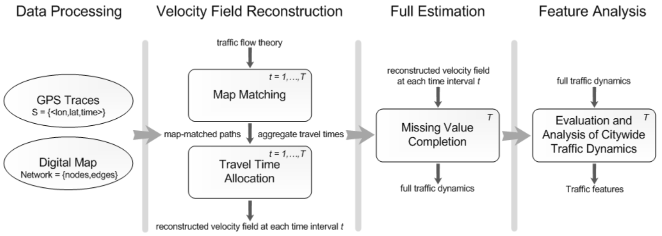


Fig. 1: Overview of the system pipeline. Map Matching and Travel Time Allocation are applied to individual time intervals. Missing Value Completion is performed over all time intervals.

path. Once introduced, this bias will be carried over to the subsequent step (i.e., *travel time allocation*), deteriorating the estimation accuracy.

We have developed a novel framework to estimate citywide traffic dynamics based on two observations: 1) Traffic patterns exhibit weekly periodicity [9], and 2) traffic conditions are quasi-static [6]. Based on these observations, we examine traffic dynamics over the period of a week and treat traffic conditions within that weekly period (the discretized time interval) as static. Our framework consists of two phases. The first phase is conducted in each individual time interval and the second phase is performed over all time intervals. Algorithmically, in the first phase, based on Wardrop's Principles [10], we use the shortest travel time instead of the shortest distance criterion to perform map matching. Along with a travel-time allocation technique adapted from [11], a novel computation scheme is designed to reconstruct the velocity field of a road network. In the second phase, exploiting the sparsity embedded in traffic patterns, we have developed a novel method based on the Compressed Sensing [12], [13] algorithm to fill in missing travel information over an entire traffic period. The overview of our framework pipeline is shown in Figure 1.

The effectiveness of our approach has been extensively evaluated and compared to a state-of-the-art method, using a synthetic road network and traffic data from the city of San Francisco. The results demonstrate major improvements over existing methods in various steps during the estimation process. In summary, our contributions include:

- A novel perspective in addressing map-matching of sparse GPS traces and an improved travel time allocation technique that incorporates estimated traffic conditions;
- An efficient method for estimating missing travel information by exploring sparse structures embedded in traffic patterns;
- Extensive analysis of our method's performance and comparison of our method to existing methods under various traffic scenarios.

The rest of paper is organized as follows. In Section II, we discuss related work and highlight the differences between our work and existing studies. In Section III, we take a holistic view of map matching and travel time allocation problems, and detail the process of reconstructing the velocity field of a road network. In Section IV, we explain our method on estimating missing travel information; In Section V, we show estimated traffic dynamics of San Francisco and analyze key features within the recovered traffic pattern. We conclude with a discussion of future work in Section VI.

II. Related Work

Over the last few decades, the estimation of traffic conditions has gained increasing scholarly attention [14], [15], [16], [17], [18], [19]. Early works on traffic estimation have studied traffic states on highways using relatively accurate measurements from stationary sensors such as loop detectors and video cameras [3]. Recent advancements have focused on combining multiple data sources and traffic simulation models [20], [21], [22], [23], [24], [25] to achieve highly accurate estimations. However, these previous methods are limited to road segments with lengths of a few kilometers. The increasing availability of GPS data provides new means for conducting large-scale estimations. However, as GPS data are inherently noisy, the estimated traffic conditions usually do not satisfy the flow conservation requirement assumed by many simulation models [26], [27], [28]. Consequently, new studies are emerging and these studies commonly take several steps to perform the estimation.

To utilize GPS data in estimation of traffic dynamics, the first step, *map-matching*, addresses the problem of mapping off-the-road GPS points onto a road network and identifies the true traversed path between consecutive GPS points. However, GPS data has a *low sampling rate*, meaning that two consecutive GPS reports could come from two spatially distant locations. Given a complex urban network, many paths connecting the two reports could exist. In order to determine the "actual" traversed path, a common approach is to use the shortest-distance criterion [1], [4], [5], [6], [7], [8]. Nonetheless, the shortest-distance assumption gives rise to bias in a congested environment, where alternative paths can be traveled faster than the shortest-distance path. Essentially, the shortest-distance criterion only uses the spatial information (i.e., longitude and latitude), and ignores the temporal information (i.e., timestamps) recorded in GPS data. This happens primarily due to the travel times for road segments of a network are largely unknown so that the temporal information has nothing with which to be compared [29], [30].

After *map-matching* is completed, travel time must be estimated, and this estimation has been approached in various ways. For example, Hellinga et al. [11] have developed an analytical solution to estimate travel times of road segments using intuitive and empirical observations of traffic patterns. Rahmani et al. [31] take a non-parametric approach, performing the estimation using a kernel-based method. Probabilistic frameworks are also often used to conduct the estimation [32], [33], [34], [35], [36]. While significant improvements in traffic estimation have been achieved, these previous methods all do the steps sequentially, meaning that limitations of the *map-matching* process constrain the accuracy of subsequent steps and reduce these methods' overall accuracy and performance.

Researchers have also proposed solutions to the *missing value completion* problem. For example, tensor-based approaches [37], [38] that explore correlations among nearby road segments have been developed. In [39] and [40], Compressed Sensing–based algorithms have been proposed by taking the entire network into account. The estimation of missing values has also been addressed in an online fashion [41]. However, these methods were not tackling the problem of estimating *full* traffic dynamics of individual road segments and aggregately an entire city network, in which subject little progress has been made [9].

III. Traffic Velocity Field Reconstruction

We take a holistic view of the map-matching and travel time allocation problems and propose a method to reconstruct the velocity field of a road network. Starting with some definitions, we then discuss methodologies and implementation details of our approach. Our algorithm is evaluated and validated using a synthetic road network with microscopic traffic simulations.

A. Definition and Notation

A *road network* is defined as a directed graph $G = (V, E)$ in which edges E denote road segments and nodes V represent intersections and terminal points. Each road segment $e \in E$ contains several attributes: the length $e.len$, the maximum/free-flow travel speed $e.v_{max}$, the minimum/free-flow travel time $e.t_{min} = \frac{e.len}{e.v_{max}}$, and the maximum/jam density $e.k_{max}$.

A *path* from node g to node h on a network $g \overset{p}{\rightsquigarrow} h$ is a collection of road segments $p = \{e_1, e_2, \ldots, e_n\}$, where g is the starting node of e_1 and h is the ending node of e_n. A *trace* is a sequence of GPS points $S = \{s_1, s_2, \ldots, s_n\}$ in which each point is a tuple $s_i =< s_i.x, s_i.y, s_i.t >$ containing longitude, latitude, and a timestamp.

B. Velocity Field Estimation

Given the periodicity of traffic patterns over a week [9], we study traffic dynamics over the region of interest in a weekly period. We discretize one week into many time intervals, and assume that traffic conditions remain the same within a time interval. For simplicity, we restrict our discussion of estimating the velocity field to one time interval. The process can be trivially extended to cover an entire traffic period.

Ideally, if the actual traversed path of a vehicle is known and the generated GPS points are exactly on the road, we can derive the average travel speed of a path p that connects GPS points s_i and s_{i+1} as $p.t = \frac{\sum_{e \in p} e.len}{s_{i+1}.t - s_i.t}$. However, GPS points are often off-the-road due to inherent measurement and sensing errors, and the underlying path of a vehicle is also unknown. To address these issues, first, a number of candidate nodes of the network are considered for mapping a GPS point based on their distances to the point. Then, one of the paths connecting a pair of candidate nodes of consecutive GPS points is selected to represent the actual path. As mentioned earlier, one common approach for choosing such a path is based on the shortest distance criterion [1], [4], [5], [6], [7], [8], which fails in some situations (see Figure 2).

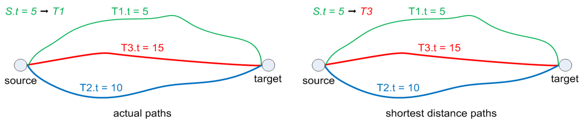


Fig. 2: An example illustrating a failure using the shortest distance criterion on map matching trace S to $T1$, $T2$, or $T3$. By matching travel time of trace S and road conditions, the correct path $T1$ is identified (left). By using only the shortest distance criterion, S is mismatched to $T3$ (right).

According to Wardrop's Principles [10], the traffic in congested networks would move in a way such that no vehicle can reduce its travel cost by switching routes. This state is called *user equilibrium*, and is a result of every user non-collaboratively attempting to minimize the traveling cost—which commonly appears to be the travel time. Under such an equilibrium state, the average travel time is balanced for all users of the network.

Assumption: Based on these principles and the observation that modern GPS largely adopts the fastest route planning strategy, we assume *all GPS traces are planned using the shortest travel time criterion.*

Denoting the network with true traffic conditions as G_{true}, all GPS traces represent the fastest routes planned on G_{true}. However, as we don't have G_{true}, our goal is to use available GPS traces and an initial road network G_{est}, with all road segments set to their speed limits, to estimate G_{true} or, in other words, to reconstruct the velocity field of G_{true}.

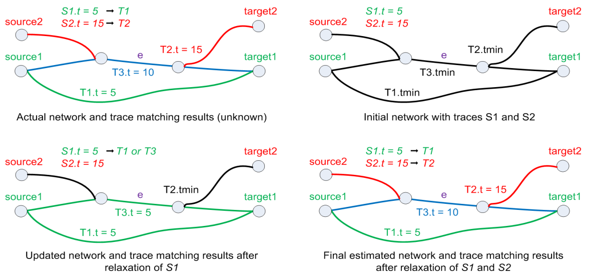
Collorary 1: A GPS trace with travel time t matching a path $g \overset{p}{\rightsquigarrow} h$ implies that no path in G_{true} from g to h has a travel time smaller than t.

Proof: By our Assumption above, the traffic along every traveled route between g and h in G_{true} is in user equilibrium, and all routes have equivalent travel times. A travel time value for one path between g and h thus provides a lower bound for the travel times of all paths.

Collorary 2: A pair of GPS points from a trace matched to locations g and h are sufficient to bound the travel time for paths between g and h.

Proof: By the Assumption, the path indicated by a GPS trace is a time-optimal path. Therefore, it has an optimal substructure. A subset of two is thus time-optimal and can provide a bound on the travel time for paths from g to h.

As Collorary 2 implies, during reconstruction of the velocity field we can inspect two consecutive GPS points at a time (optimal substructure). Considering two arbitrary GPS points s_i and s_{i+1}, the true path $s_i \overset{p_{true}}{\rightsquigarrow} s_{i+1}$ is the fastest path between s_i and s_{i+1} on G_{true}. By Collorary 1, the travel time $p_{true}.t = s_{i+1}.t - s_i.t$ is the lower bound of all travel times between the two points. Subsequently, we enforce the constraint that no path in G_{est} between s_i and s_{i+1} has a smaller travel time. If such a path p_{est} exists, the speeds of the road segments on p_{est} should be decreased utill $p_{est}.t \geq p_{true}.t$. We refer to such p_{est} as an *overestimated path* as its speed is higher than it should be. In practice, p_{est} connects candidate nodes of s_i and s_{i+1} rather than s_i and s_{i+1} themselves, and there exists a set of paths $\{p_{est}\}_{all}$ for s_i and s_{i+1}, in which one is the "closest" to p_{true}. Denoting an arbitrary element in $\{p_{est}\}_{all}$ as p_{est}, p_{true}, and p_{est} are likely to contain different sets of road segments. If p_{est} and p_{true} happen to be the same path,



Actual network and trace matching results (unknown)

Initial network with traces S1 and S2

Updated network and trace matching results after relaxation of *S1*

Final estimated network and trace matching results after relaxation of *S1* and *S2*

Fig. 3: An illustration of the relaxation process of two traces $S1$ and $S2$. The actual traffic conditions and trace matching results are shown in the top left panel. The inputs listed in the top right panel contain the initial network and trace information (i.e., sources, targets, and travel time). After relaxation of trace $S1$, paths $T1$ and $T3$ have their travel time increased from the minimum value to $S1.t$. While $S1$ can be matched to either $T1$ or $T3$, the situation is resolved after relaxation of $S2$ due to further increase in travel time of the road segment e.

then $p_{est}.t$ should be equal to $p_{true}.t$, otherwise $p_{est}.t$ should be larger than $p_{true}.t$. Since there is no further information for us to estimate the excessive time of $p_{est}.t$ over $p_{true}.t$, we take a conservative approach by setting $p_{est}.t = p_{true}.t$. We refer to this process of removing overestimated paths in $\{p_{est}\}_{all}$ as *relaxation.*

The relaxation will make all paths in $\{p_{est}\}_{all}$ have the same travel time (i.e., $p_{true}.t$). However, it is very difficult to deterministically derive the "closest" path to $p_{true}.t$ using a single GPS trace. To remedy this issue, we rely on the "collective intelligence" of multiple GPS traces with shared road segments. As these segments will get gradually updated during relaxation of each GPS trace, they will eventually assist in differentiating the paths that include them from other paths in terms of the travel time. An illustration of this process using the travel time as measurements can be found in Figure 3. The relaxation is essentially a fulfillment of Wardrop's Principles, and is conducted in a greedy fashion: we repeatedly extract the fastest path in $\{p_{est}\}_{all}$ and relax it, until no path in $\{p_{est}\}_{all}$ has its travel time smaller than $p_{true}.t$. Given that there may exist many paths connecting two nodes in a network, the number of elements in $\{p_{est}\}_{all}$ could be large, which further leads to an expensive computation. A sub-network with a specified radius is extracted from the network, which encompasses s_i, s_{i+1}, and their mapping candidates but no more. With this approach, we greatly reduced the number of paths in $\{p_{est}\}_{all}$. The rationale behind this choice is through empirical findings that a vehicle rarely takes an opposite direction or arbitrary long detour from one GPS point to the next one.

Theorem 1: The speed of a road segment is monotonically decreasing during relaxation.

Proof: We prove this theorem by contradiction. Assume two overestimated paths p^1_{est} and p^2_{est} share a road segment e. Without loss of generality, considering relaxation of p^1_{est}, the road segment e has its speed $e.v$ decreased to $e.v'$ and travel time $e.t$ increased to $e.t'$ so that $p^1_{est}.t = p^1_{true}.t$. Now, during relaxation of p^2_{est}, $e.v'$ and $e.t'$ are subject to change. If instead of monotonically decreasing $e.v'$ gets increased and $e.t'$ gets decreased, then we have $p^1_{est}.t < p^1_{true}.t$ which invalidates the previous relaxation process and further contradicts Collorary 1.

Taking advantage of Theorem 1, further reduction in computation can be achieved by retaining reduced speeds of each path in $\{p_{est}\}_{all}$. To be specific, as $\{p_{est}\}_{all}$ is generated for s_i and s_{i+1} in a sub-network, many paths in $\{p_{est}\}_{all}$ will have shared road segments. Therefore, the speed reduction in these road segments will make multiple paths in $\{p_{est}\}_{all}$ to have increased travel times. As a result, the greedy process of relaxation is much more efficient than a brute-force enumeration.

Algorithm 1 Velocity Field Reconstruction

Input: Initial estimated road network $G_{est} = (V, E)$ with $e.v = e.v_{max}$, $\forall e \in E$; GPS traces $S = \{S_1, \ldots, S_m\}$; Discretized time intervals $\{1, \ldots, D\}$; Maximum distance for computing candidate nodes of a GPS point $cDis$; Maximum number of candidate nodes $cNum$

Output: Reconstructed road network G_{est}

```
for each time interval $d \in (1, \ldots, D)$ do :
    $S^d = ExtractGPSTraces(S, d)$
    for each trace $S^d_j \in S^d$ do :
        for consecutive GPS points $s_i, s_{i+1} \in S^d_j$ do :
            $radius = \frac{dist(s_i, s_{i+1})}{2} + cDis$
            $H = ExtractSubgraph(G_{est}, radius, s_i, s_{i+1})$
            $C_1 = GetCandidateNodes(H, s_i, cDis, cNum)$
            $C_2 = GetCandidateNodes(H, s_{i+1}, cDis, cNum)$
            $p_{true}.t = s_{i+1}.t - s_i.t$
            $H_{relax} = RelaxNetwork(H, p_{true}.t, C_1, C_2)$
            $G_{est} = UpdateNetwork(G_{est}, H_{relax})$
return $G_{est}$
```

The overall process is described in Algorithm 1. Subroutines *RelaxNetwork* and *Relaxation* are specified in Algorithm 2 and 3, respectively. In particular, two types of paths are considered as outliers and are excluded from the computation: one has a travel time shorter than its free-flow travel time and one has a travel time longer than the travel time under the jam density. The procedure *TravelTimeAllocation* in Line 10 of Algorithm 3 is discussed next.

C. Travel Time Allocation

During relaxation, we need to address each overestimated path p_{est} between GPS points s_i and s_{i+1} by making $p_{est}.t = p_{true}.t = s_{i+1}.t - s_i.t$. Due to the low sampling rate, p_{est} often spreads several road segments, and the aggregate travel time $p_{true}.t$ needs to be appropriately allocated to individual road segments of p_{est}. To address this issue with respect to traffic flow analysis, we adopt and modify the solution proposed in [11].

According to [11], the travel time of a road segment e can be decomposed into three categories: free-flow travel time $\tau_{e,f}$, congestion time $\tau_{e,c}$, and stopping time $\tau_{e,s}$. For an overestimated path p_{est}, we denote its total free-flow

Algorithm 2 RelaxNetwork

```
Input: Subgraph $H$; True travel time $p_{true}.t$; Candidate node sets $C_1$, $C_2$
Output: Relaxed subgraph $H_{relax}$
for each node $n_1 \in C_1$ do :
    for each node $n_2 \in C_2$ do :
        if $n_1 == n_2$ or $distance(n_1,n_2) == 0$ then
            continue                                ▷ no valid path exists
        $p_{est} = GetFastestPath(H,n_1,n_2)$
        if $IsOutlier(p_{est}) == true$ then
            continue
        while $true$ do :
            if $t_{est} \geq t_{true}$ then
                break                               ▷ not an overestimated path
            if $NumberOfNodes(p) < 2$ then
                break                               ▷ not a valid path
            $E_{relax} = Relaxation(p_{est},p_{true}.t)$
            $H = UpdateNetwork(H,E_{relax})$        ▷ update travel times of $H$ using $E_{relax}$
            $p_{est} = GetFastestPath(H,n_1,n_2)$   ▷ get the shortest travel time path between $n_1$ and $n_2$ on $H$
return $H_{relax} = H$
```

travel time as $T_f = \sum_{e \in p_{est}} \tau_{e,f}$, total congestion time as $T_c = \sum_{e \in p_{est}} \tau_{e,c}$, total stopping time as $T_s = \sum_{e \in p_{est}} \tau_{e,s}$, and the allocation time as $T_a = p_{true}.t$. To validate p_{est}, we must have $T_f + T_c + T_s = T_a$. While T_f can be derived trivially as $T_f = \sum_{e \in p_{est}} \tau_{e,f} = \sum_{e \in p_{est}} \frac{e.len}{e.v_{max}}$, the computations of T_c and T_s require additional considerations.

Algorithm 3 Relaxation

```
Input: A set of road segments $E$ (initially contains all road segments in $p_{est}$); Time budget $T$ (initially set to $p_{true}.t$)
Output: A set of relaxed road segments $E_{relax}$
$avgSpeed = \frac{\sum_{e \in E} \|e\|}{T}$        ▷ average travel speed
$T' = T$                                           ▷ store the time budget
for each road segment $e \in E$ do :
    if $e.v \leq avgSpeed$ then
        $T = T - e.t$                              ▷ the leftover time budget
        $E = E \setminus \{e\}$                    ▷ exclude $e$ based on theorem 1
if $T' \neq T$ then                                ▷ some $e$ have been excluded
    $E_{relax} = Relaxation(E,T)$                  ▷ recursive call
else
    $E_{relax} = TravelTimeAllocation(E,T)$
return $E_{relax}$
```

By assuming nearby road segments have similar traffic conditions, the *path congestion level* is defined as $w = \frac{T_c}{T_c + T_f}$. The minimum value $w_{min} = 0$ is reached when the path can be traveled with the free-flow speed, and the maximum value $w_{max} = \frac{T_{c,max}}{T_{c,max} + T_f}$ where $T_{c,max} = T_a - T_f$ is reached when the path is congested but no more so that $T_s = 0$. With a specific path congestion level $w_{min} \leq w \leq w_{max}$, the probability that a certain degree of congestion occurred on p_{est} is computed as:

$$P_c(w) = min\{1, \frac{T^{prev}_{c,max} + T_{c,max}}{T^{prev}_a + T_a} \cdot \frac{1}{w}\}, \tag{1}$$

where $T^{prev}_{c,max}$ and T^{prev}_a represent the maximum congestion time and the allocation time of path $s_j \overset{p^{prev}_{est}}{\rightsquigarrow} s_{j+1}, j + 1 \leq i$. In [11], p^{prev}_{est} denotes the path which has an allocation time longer than the jam-density travel time with maximum possible j. In this work, as we have addressed path outliers, p^{prev}_{est} indicates the path connecting s_{i-1} and s_i, which is the directly preceding path of p_{est}. $P_c(w)$ is defined under assumptions that first, when all variables are fixed, the probability of a specific level of congestion occurring increases as $T_{c,max}$ increases; second, given a particular $T_{c,max}$, higher level congestions are less likely to appear than lower level congestions.

Next, the *stopping likelihood function* is defined for computing the probability of stopping. Since the original formula does not take estimated traffic conditions into account, we alter it to:

$$L_{s,e}(w) = \beta w + (1 - \beta) \frac{e.k}{e.k_{max}}, \tag{2}$$

where β is the weighting factor (set to 0.5 in this work), and $e.k$ is the estimated density as a result of possible previous relaxation (otherwise $e.k = 0$). Equation 2 computes the likelihood by leveraging the *path congestion level* w and the *road-segment congestion level* $\frac{e.k}{e.k_{max}}$. Intuitively, the road-segment congestion level becomes higher when the estimated density $e.k$ approaches the jam density $e.k_{max}$. In order to derive $e.k$ from the estimated speed $e.v$, we utilize the Greenshield's model [42]:

$$e.k = e.k_{max} \left(1 - \frac{e.v}{e.v_{max}}\right). \tag{3}$$

By having the *stopping likelihood function*, the probability that a vehicle stopped on a particular road segment is stipulated by assuming the vehicle stops at most once on p_{est} as:

$$P_{s,e_i}(w) = L_{s,e_i}(w) \prod_{e_j \in p_{est}, j \neq i} (1 - L_{s,e_j}(w)). \tag{4}$$

With Equation (1) and (4), the congestion time on a single road segment is calculated by integrating all path congestion levels together as:

$$\tau_{e,c} = \int_0^{w_{max}} \tau_{e,f} \frac{w}{1 - w} \frac{P_c(w) \sum_e P_{s,e}(w)}{Q_s} dw, \tag{5}$$

where $Q_s = \int_0^{w_{max}} P_c(w) \sum_e P_{s,e}(w) dw$ is the normalizing factor. After summing the congestion time of all road segments $T_c = \sum_{e \in p_{est}} \tau_{e,c}$, the total stopping time can be derived easily by $T_s = T_a - T_f - T_c$. Finally, we can calculate the stopping time on each road segment using the following formula:

$$\tau_{e,s} = \int_0^{w_{max}} T_s \frac{P_c(w) P_{s,e}(w)}{Q_s} dw. \tag{6}$$

To this point, we have solved the travel time allocation problem by having $\tau_{e,f}$, $\tau_{e,c}$, and $\tau_{e,s}$ for all road segments of p_{est} and fulfilling $T_f + T_c + T_s = T_a$ (i.e., $p_{est}.t = p_{true}.t$).

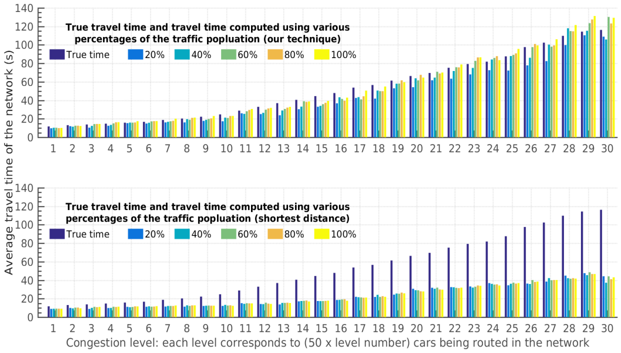


Fig. 4: Average travel time simulated on the synthetic road network using our approach (top) vs. [1] with the shortest distance (bottom). Our technique consistently outperforms the existing method in estimating the average travel time over 30 different congestion levels.

D. Evaluation on A Synthetic Network Using Traffic Simulation

In order to evaluate our technique on estimating velocity field, we use a synthetic road network and an agent-based traffic simulator [43]. The road network is modeled as a grid with 5×5 intersections. By considering one hour as a time interval for a specific congestion level, a set of cars is routed and average travel times of all road segments are taken as the ground truth for this congestion level. All traces are simulated by randomly sampling nodes of the network as sources and targets using the fastest route strategy in the beginning of a time interval. As our method operates on size 2 subsets of GPS traces, we emit pairs of points at the source and target for each simulated trace. This choice resembles the low sampling rate feature and enables us to incorporate the travel-time allocation algorithm into testing.

All road segments share the same setting: length of $150m$, maximum speed at $17.88m/s$, and a maximum density of 0.15 cars per meter. In total, 30 congestion levels are created by simulating 50 to 1500 vehicles in this road network with an increment of 50 vehicles per time interval. In addition, for each time interval, five tiers of the vehicle population, at 20%, 40%, 60%, 80%, and 100%, are used to generate GPS traces.

The first analysis is conducted by treating the network as a whole, and the aggregate travel time over the network is used to represent the *global congestion level*. The ground truth compared to estimated quantities using our technique and Lou et al. [1] are shown in Figure 4. Start with 20% of vehicles, our technique demonstrates close approximation to the ground truth at all congestion levels, while such phenomena are not observed using the shortest-distance based technique developed in [1]. In Table I, we examine each traffic percentage by computing the absolute error to the ground truth across all congestion levels. The smallest error (mean = $3.4s$, std = $3.32s$) is achieved by using 80% of GPS traces from the simulated traffic. The slight increase in error when using 100% GPS traces is mainly due to imperfect travel-time allocation method.

	Statistics of absolute errors			
	Our technique		Lou et al. [1]	
Traffic percentage	mean (s)	std. (s)	mean (s)	std. (s)
20%	8.29	5.31	29.74	21.90
40%	4.25	3.38	29.85	22.66
60%	3.67	3.67	29.33	22.06
80%	3.40	3.32	29.55	22.46
100%	3.58	4.04	29.66	22.35

TABLE I: The absolute errors in the recovered travel time computed using our technique vs. Lou et al. [1] by using GPS traces from various percentages of the traffic population. Our technique results in much smaller errors as compared to [1].

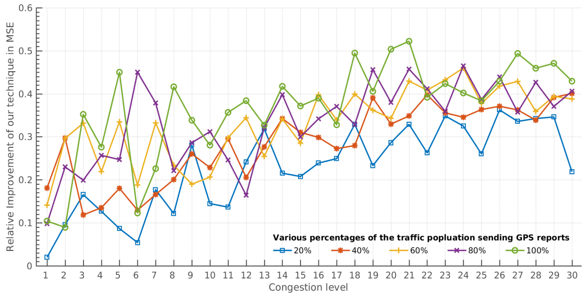


Fig. 5: Relative improvements measured in MSE of our technique over Lou et al. [1]. Our technique outperforms [1] as the congestion level increases or as more GPS traces become available. Our method achieves up to 52.5% improvement.

In the second analysis, we computed relative improvements measured in mean squared error (MSE)[1]of our method over [1]. The results are demonstrated in Figure 5.

In general, as the congestion level increases or a higher percentage of GPS traces becomes available, our technique outperforms the shortest-distance technique. The improvements are less clear when congestion level is low (i.e., < 10), but better seen when congestion approaches high level (i.e., ≥ 10). The smaller improvement at the low-congestion levels is because when vehicles can travel on roads with speeds close to the speed limits, the shortest travel-time path and the shortest-distance path tend to be the same in such cases.

IV. Missing Value Completion

The temporal sparsity of GPS data can lead to missing values in certain time intervals over a weekly period, and these missing values inhibit the accurate estimation of full traffic dynamics. To address this issue, we explore the sparse structure embedded in traffic patterns and propose a novel technique based on the Compressed Sensing algorithm [12], [13].

We have adopted loop-detector data[2], which represent complete and relatively accurate measurements of traffic conditions, to explore features of traffic patterns. We use speed

[1] MSE is computed as $\frac{1}{n}\sum_{i=1}^{n}(\hat{e}.t - e.t)^2$.

[2] The loop-detector data are obtained from Caltrans Performance Measurement System (PeMS): http://pems.dot.ca.gov/.

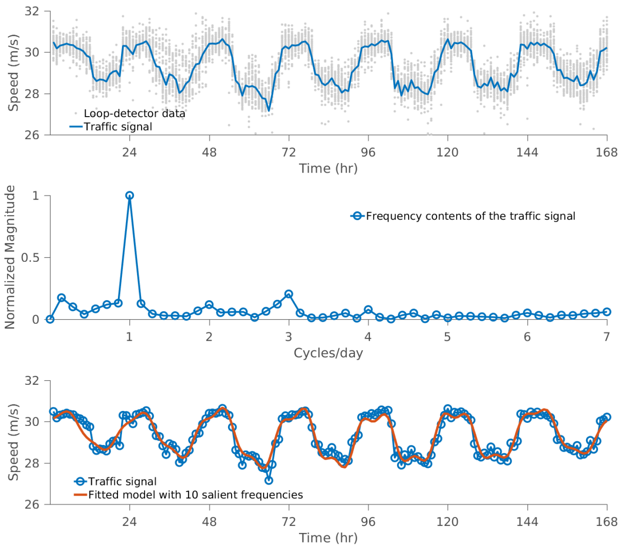


Fig. 6: The average speed measurements from loop-detector data are interpreted as a *traffic signal*, which exhibits a clear periodic pattern (TOP); the spectral analysis reveals that the most prominent frequency is one cycle per day (MIDDLE); the traffic signal is approximated by a frequency-domain linear regression model in which 95% energy is retained by keeping the ten largest frequencies (BOTTOM).

measurements from 38 loop detectors installed in San Francisco. The time range of these data is the same as that of the Cabspotting dataset[3], which we adopted to estimate traffic conditions of San Francisco. We refer to the average speed measurements over a weekly period (discretized hourly) as a *traffic signal*. An example traffic signal exhibiting clear periodicity is shown in the top panel of Figure 6.

We perform a spectral analysis on a traffic signal. In the analysis, we set the frequencies as the reciprocal of the signal length (i.e., 1/168) and subtract the signal from its mean to make the oscillations easier to observe. The results, which are shown in the middle panel of Figure 6, reveal that the period of the most salient oscillation is 24 hours. In addition, the signal exhibits sparsity in the frequency domain, which is reflected as over 95% energy is preserved by retaining the ten largest frequencies (Figure 6, BOTTOM).

According to the specification of a traffic signal, the highest frequency being supported is one cycle per two hours. To recover a traffic signal from its samples, the Nyquist-Shannon Theorem says that we need at least 168 measurements. This number of samples is difficult to obtain because of the temporal sparsity of GPS data. However, the Compressed Sensing algorithm [13], [12] suggests that a signal can be recovered exactly with a small set of samples if the signal has a sparse representation. This sparse representation can be manifested by the rapid decaying of the sorted frequency magnitudes. The analysis of decay rates of traffic signals is presented in the top panel of Figure 7. On average, the decay rate reaches 45.78% with the most prominent frequency and 78.93% with the ten most conspicuous frequencies. After the 15th frequency, the decay rates become ineligible.

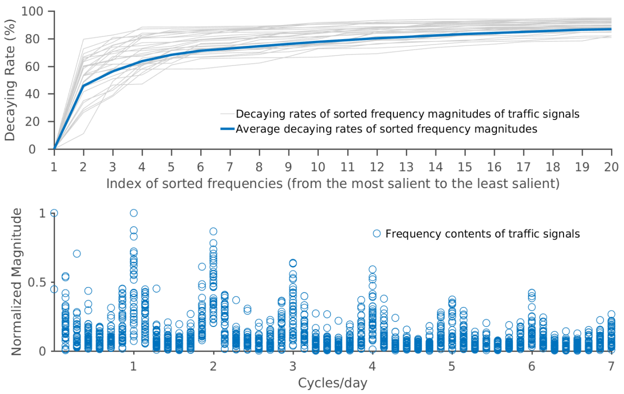


Fig. 7: The top panel shows the decay rates of frequency magnitudes of all traffic signals; the bottom panel shows the locations and normalized magnitudes of the frequency components of all traffic signals. The rapid growth of decay rates and randomly distributed frequency structures indicate that Compressed Sensing is applicable for recovering traffic signals.

Another merit of Compressed Sensing is that it does not require any prior knowledge of the sparse structure of a signal. Since the road condition is intrinsically stochastic, the sparse structure of a traffic signal varies from one link to another. This phenomenon is demonstrated in the bottom panel of Figure 7, in which locations and amplitudes of the frequency components of all traffic signals are plotted. Besides the appearance of the prominent oscillations, at 24, 48, and 72 hours, other oscillations are spread out along the frequency axis. Together, these observations and features confirm the applicability of using Compressed Sensing to recover traffic signals.

Given a signal $f \in \mathbb{R}^n$ and its measurements $b \in \mathbb{R}^m$, we consider the undersampled case in which the number of measurements m is smaller than the signal's dimension n. The goal is to derive an estimated signal $\hat{f} \in \mathbb{R}^n$ from $b \in \mathbb{R}^m$ such that the error $\|f - \hat{f}\|_{\mathcal{L}_2}$ is minimized. In general, the better the desired reconstruction quality, the more measurements are needed. In order to achieve a predefined accuracy level, signal reconstruction requires a minimum number of measurements m_{min}. According to [44], m_{min} is on the order of $\mu^2 S \log(n)$, where μ is the coherence between a measurement basis and a representation basis, S is the signal's sparsity level, and n is the signal's dimension (in this work $n = 168$). We estimate S by averaging the number of frequencies in preserving 95% energy of all traffic signals, which results in $S = 17.63$. The minimum coherence value $\mu = 1$ is obtained by performing a discrete cosine transform (DCT) on a traffic signal f:

$$\Phi f_{dct} = f, \tag{7}$$

where f_{dct} is the representation of f in the DCT domain and $\Phi_{n \times n}$ is the DCT matrix. With these estimated values, the minimum number of samples required to recover a traffic signal can be computed as: $m_{min} = \mu^2 S \log(n) = 1^2 \cdot 17.63 \cdot \log(168) \approx 90$.

[3] http://cabspotting.org/.

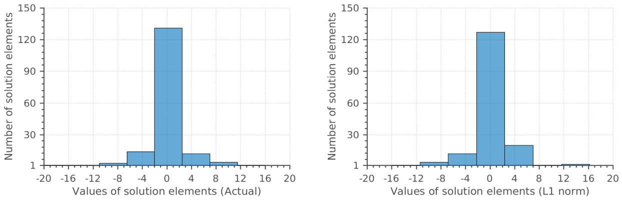


Fig. 8: Solution elements (in total 168) by solving an underdetermined system via convex optimization. Both the actual and the $\mathcal{L}_1$-norm based recovery demonstrate sparsity (i.e., most solution elements are approximately equal to zero).

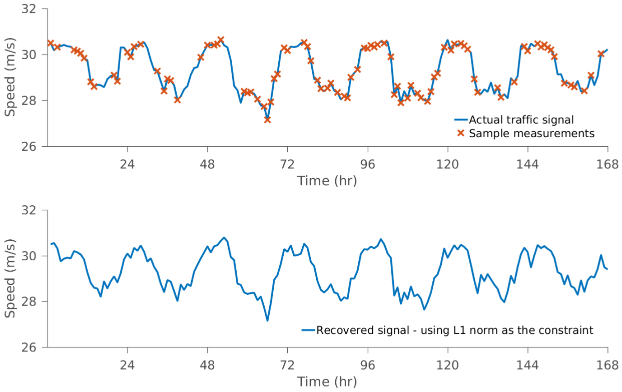


Fig. 9: Recovery of a traffic signal via Compressed Sensing. The actual signal and its 90 random measurements are plotted (TOP). The $\mathcal{L}_1$-norm based recovery (BOTTOM) shows high similarity to the actual signal.

We test the performance of our method of signal recovery by first obtaining random measurements via sampling: $b_{m\times 1} = \Psi_{m\times n} f_{n\times 1}$, where Ψ is the sampling matrix constructed by randomly permuting rows of the identity matrix. Then we derive the recovered signal in the DCT domain $\hat{f}_{dct}$ from b by solving the following linear system:

$$\mathbf{A}\hat{f}_{dct} = \Psi\Phi\hat{f}_{dct} = b. \tag{8}$$

Equation 8 represents an *underdetermined* system, in which there exist infinitely many candidate signals $\hat{f}_{dct}$ for which the formula can suffice. Among all candidates, the desired $\hat{f}_{dct}$ should exhibit sparsity as observed in f_{dct}. We can acquire such a solution by solving the following optimization program:

$$\begin{aligned} &min \ \|\hat{f}_{dct}\|_{\mathcal{L}_1}, \\ &s.t. \quad \mathbf{A}\hat{f}_{dct} = b. \end{aligned} \tag{9}$$

An example solution to Equation 9 is shown in Figure 8, where the actual solution elements of f_{dct} and the recovered solution elements of $\hat{f}_{dct}$ both demonstrate sparsity. The final recovered signal $\hat{f}$ is acquired by performing an inverse DCT on $\hat{f}_{dct}$. Figure 9 gives an example in which the recovered signal exhibits high similarity to the original signal. A more thorough analysis of the performance of recovering traffic signals can be found in Figure 10. As a result, both the standard deviation and the expectation of the $\mathcal{L}2$ loss decrease, when the number of measurements used in recovery increases. The average error of using 90 measurements is $1.4m/s$.

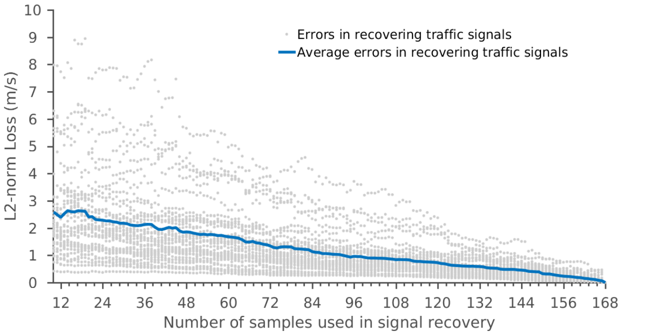


Fig. 10: The error between recovered and actual traffic signals. As more samples are used in signal's recovery, we observe smaller errors.

To put this framework in terms of GPS data, the measurements used are obtained from *travel time estimation* rather than from the sampling operation that we performed on a traffic signal. In this case, Φ is set to $DCT\,(diag\,(1,\ldots,1))$, and $\mathbf{A} = \Psi\mathbf{D}$ is taken to solve Equation 9. Since we have established that $m_{min} = 90$, it is worth mentioning that we only address links that have measurements in more than 90 time intervals. Compared to [45] and [9], we have reduced the minimum number of measurements required to recover a traffic signal from 1680 to 90 (by 94.64%).

V. Estimating Traffic Dynamics Via GPS Data

One of the hallmarks of traffic dynamics is their periodicity [9]: traffic patterns show a clear trend over the course of a day and collectively over the course of a week. In this section, we first demonstrate that this phenomenon can be recovered using our technique (tested on the Cabspotting dataset); we then analyze features revealed in the reconstructed traffic patterns.

To assist in visualization and analysis, the metric *fluidity*, adapted from [9], is used for each road segment as the ratio of the estimated travel speed to the free-flow speed. this metric ranges from 0 to 1. In Figure 11, we show the estimated traffic dynamics, denoted by average fluidity, across the road network of downtown San Francisco. From the demonstration, it is clear that our technique recovers the periodicity of the traffic pattern using the dominant frequency, which is one cycle per day. This characteristic resembles the one observed in loop-detector data from the same area (see Figure 6 and 9).

The affinity between different days in a week can also be used to illustrate the quality of our data reconstruction method. We compute the correlation of every pair of days using the cosine distance for both the estimated traffic conditions and the actual conditions derived from loop-detector data. In Figure 12, left panel, we provide all distance scores: the upper triangular matrix is derived using the estimated values, and the lower triangular matrix is computed using loop-detector data. In the right panel of Figure 12, we provide the qualitative result for a visual inspection. The symmetrical pattern across the diagonal line indicates that the estimated traffic states largely agree with the loop-detector data.

Based on the distance scores, a hierarchical clustering is performed to reveal the similarity between day pairs. The

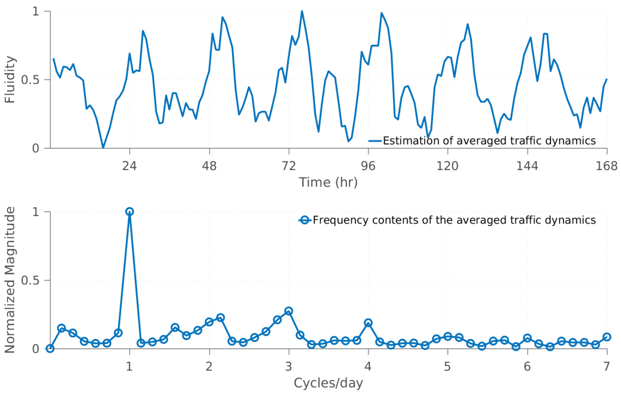


Fig. 11: Estimated traffic dynamics of downtown San Francisco compared with spectral analysis. Our technique shows a clear daily trend, which is consistent with the features observed in loop-detector data from the same area.

Days	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Sun	0	.20	.54	.65	.64	.84	1
Mon	.13	0	.16	.40	.46	.77	1
Tue	.48	.20	0	.14	.23	.56	.82
Wed	.71	.47	.14	0	.07	.34	.57
Thu	.82	.60	.31	.12	0	.16	.38
Fri	.82	.63	.42	.31	.14	0	.12
Sat	1	.84	.67	.59	.43	.16	0

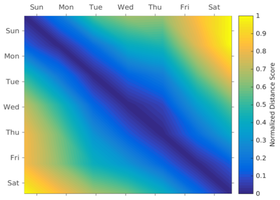


Fig. 12: Correlation between every pair of days in a week. The left panel lists normalized similarity scores calculated using the cosine distance, and the right panel provides qualitative results. For both, the upper triangular matrix is derived using the estimated traffic conditions, and the lower triangular matrix is computed using loop-detector data from the same area. When data and patterns are compared across the diagonal line, our estimated results exhibit high similarity to the loop-detector data.

closest pair is Wednesday and Thursday, followed by Friday and Saturday, and Monday and Tuesday. In the second level of the hierarchy, Sunday joins Monday and Tuesday. These three day-pairs suggest that a typical week of San Francisco can be roughly divided into three stages: *beginning of the week* (Sunday, Monday, and Tuesday), *middle of the week* (Wednesday and Thursday), and *end of the week* (Friday and Saturday).

VI. Conclusion

We have presented a novel computational scheme for estimating travel times, traversed paths, and missing values over a large-scale road network using spatially and temporally sparse GPS traces. Specifically, an approach based on the shortest travel time is performed to reconstruct the velocity field of a road network. As a result, we have obtained a novel method for joint estimation of traversed paths and travel times on a large-scale road network. Next, an algorithm based on the Compressed Sensing algorithm has been developed to estimate missing travel information over an entire traffic period so that citywide traffic dynamics can be studied. Last, we have extensively evaluated our approach and compared our technique with a state-of-the-art technique. Consistent improvements are observed in multiple traffic scenarios.

There are several possible future directions. To start with, as abundant GPS data are becoming available, the task of processing them is computationally expensive. Besides using the power of distributed computing, it is promising to explore the sparse structure and periodicity of traffic patterns to further reduce the amount of data needed in the estimation of traffic dynamics. Similarly, correlation in traffic patterns due to proximity/spatial coherence can also be examined. Last, it might be beneficial to integrate data mining of historical data, real-time traffic reconstruction from current data, and predictive traffic simulation to achieve a more comprehensive and accurate estimation of travel conditions over metropolitan areas.

References

[1] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 352–361.

[2] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "2015 urban mobility scorecard," *Texas A&M Transportation Institute and INRIX*, 2015.

[3] G. Leduc, "Road traffic data: Collection methods and applications," *Working Papers on Energy, Transport and Climate Change*, vol. 1, no. 55, 2008.

[4] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun, "An interactive-voting based map matching algorithm," in *Proceedings of the 11th International Conference on Mobile Data Management*, 2010, pp. 43–52.

[5] T. Miwa, D. Kiuchi, T. Yamamoto, and T. Morikawa, "Development of map matching algorithm for low frequency probe data," *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 132–145, 2012.

[6] T. Hunter, P. Abbeel, and A. Bayen, "The path inference filter: model-based low-latency map matching of probe vehicle data," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 507–529, 2014.

[7] B. Y. Chen, H. Yuan, Q. Li, W. H. Lam, S.-L. Shaw, and K. Yan, "Map-matching algorithm for large-scale low-frequency floating car data," *International Journal of Geographical Information Science*, vol. 28, no. 1, pp. 22–38, 2014.

[8] M. Quddus and S. Washington, "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 328–339, 2015.

[9] A. Hofleitner, R. Herring, A. Bayen, Y. Han, F. Moutarde, and A. De La Fortelle, "Large scale estimation of arterial traffic and structural analysis of traffic patterns using probe vehicles," in *Transportation Research Board 91st Annual Meeting (TRB'2012)*, 2012.

[10] J. Wardrop, "Some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, 1952.

[11] B. Hellinga, P. Izadpanah, H. Takada, and L. Fu, "Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 768–782, 2008.

[12] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[13] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[14] H. B. Celikoglu, "A dynamic network loading model for traffic dynamics modeling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 575–583, 2007.

[15] H. B. Celikoglu, E. Gedizlioglu, and M. Dell'Orco, "A node-based modeling approach for the continuous dynamic network loading problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 165–174, 2009.

[16] S. Gao, "Modeling strategic route choice and real-time information impacts in stochastic and time-dependent networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1298–1311, 2012.

[17] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[18] P. Kachroo and S. Sastry, "Travel time dynamics for intelligent transportation systems: Theory and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 385–394, 2016.

[19] S. Agarwal, P. Kachroo, and S. Contreras, "A dynamic network modeling-based approach for traffic observability problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1168–1178, 2016.

[20] D. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. Bayen, "A traffic model for velocity data assimilation," *Applied Mathematics Research Express*, vol. 2010, no. 1, pp. 1–35, 2010.

[21] Y. Sun and D. Work, "A distributed local kalman consensus filter for traffic estimation," in *Decision and Control (CDC), 53rd IEEE Annual Conference on*, 2014, pp. 6484–6491.

[22] A. Gning, L. Mihaylova, and R. K. Boel, "Interval macroscopic models for traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 523–536, 2011.

[23] L. Li, X. Chen, and L. Zhang, "Multimodel ensemble for freeway traffic state estimations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1323–1336, 2014.

[24] H. B. Celikoglu and M. A. Silgu, "Extension of traffic flow pattern dynamic classification by a macroscopic model using multivariate clustering," *Transportation Science*, 2016.

[25] M. Hajiahmadi, G. S. van de Weg, C. M. Tampère, R. Corthout, A. Hegyi, B. De Schutter, and H. Hellendoorn, "Integrated predictive control of freeway networks using the extended link transmission model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 65–78, 2016.

[26] A. Phan and F. P. Ferrie, "Interpolating sparse GPS measurements via relaxation labeling and belief propagation for the redeployment of ambulances," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1587–1598, 2011.

[27] Q.-J. Kong, Q. Zhao, C. Wei, and Y. Liu, "Efficient traffic state estimation for large-scale urban road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 398–407, 2013.

[28] J.-D. Zhang, J. Xu, and S. S. Liao, "Aggregating and sampling methods for processing GPS data streams for traffic state estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1629–1641, 2013.

[29] J. Tang, Y. Song, H. Miller, and X. Zhou, "Estimating the most likely space–time paths, dwell times and path uncertainties from vehicle trajectory data: A time geographic method," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 176–194, 2016.

[30] M. Rahmani and H. Koutsopoulos, "Path inference from sparse floating car data for urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 30, pp. 41–54, 2013.

[31] M. Rahmani, E. Jenelius, and H. Koutsopoulos, "Non-parametric estimation of route travel time distributions from low-frequency floating car data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 343–362, 2015.

[32] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. Van Lint, "Prediction intervals to account for uncertainties in travel time prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 537–547, 2011.

[33] B. Westgate, D. Woodard, D. Matteson, and S. Henderson, "Travel time estimation for ambulances using bayesian data augmentation," *The Annals of Applied Statistics*, vol. 7, no. 2, pp. 1139–1161, 2013.

[34] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on*, 2010, pp. 929–936.

[35] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1679–1693, 2012.

[36] K. Kuhi, K. K. Kaare, and O. Koppel, "Using probabilistic models for missing data prediction in network industries performance measurement systems," *Procedia Engineering*, vol. 100, pp. 1348–1353, 2015.

[37] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 25–34.

[38] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1816–1825, 2016.

[39] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 11, pp. 2289–2302, 2013.

[40] N. Mitrovic, M. T. Asif, J. Dauwels, and P. Jaillet, "Low-dimensional models for compressed sensing and prediction of large-scale traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2949–2954, 2015.

[41] O. Anava, E. Hazan, and A. Zeevi, "Online time series prediction with missing data," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2191–2199.

[42] B. Greenshields, J. Bibbins, W. Channing, and H. Miller, "A study of traffic capacity," *Highway Research Board Proceedings*, vol. 14, no. 1, pp. 448–477, 1935.

[43] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO–simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.

[44] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[45] R. Herring, "Real-time traffic modeling and estimation with streaming probe data using machine learning," Ph.D. dissertation, University of California, Berkeley, 138 pages, 2010.

Weizi Li received his B.E. degree in Computer Science and Technology from Xiangtan University, China and M.S. degree in Computer Science from George Mason University. He is currently in the doctoral program at the University of North Carolina at Chapel Hill, Department of Computer Science. His research interests include agent-based simulation, intelligent transportation systems, and statistical machine learning.

Dong Nie received his B.Eng. degree in Software Engineering from Northeastern University, Shenyang, China and M.Sc. degree in Computer Science from the University of Chinese Academy of Sciences, Beijing, China. He is pursuing a Ph.D. degree in Computer Science from the University of North Carolina at Chapel Hill. His research interests include image processing and medical image analysis.

David Wilkie received a BS in computer science from Drexel University and his PhD from the University of North Carolina at Chapel Hill, Department of Computer Science. He is now a software engineer for Google Maps. David's research interests include traffic simulation, GIS and road network modeling, and intelligent transportation systems.

Ming C. Lin received her B.S., M.S., Ph.D. degrees in Electrical Engineering and Computer Science from the University of California, Berkeley. She is currently John R. & Louise S. Parker Distinguished Professor of Computer Science at the University of North Carolina (UNC), Chapel Hill. She has received several honors and awards, including the NSF Young Faculty Career Award, UNC Hettleman Award for Scholarly Achievements, 2010 IEEE VGTC Technical Achievement Award, and 10 best paper awards at premium international conferences. She is a Fellow of ACM and IEEE. She has served as a member of the Board of Directors of Computing Research Association Women, a member of IEEE CS Board of Governors, Chair of 2015 IEEE Computer Society (CS) Transactions Operation Committee, and a former Editor-in-Chief of IEEE Transactions on Visualization and Computer Graphics (2011-2014). She is also a member of several Editorial Boards, steering committees, and advisory boards of international conferences, government agencies, and industry.