A Compressive Sensing Approach to Urban Traffic Estimation with Probe Vehicles

Yanmin Zhu, *Member*, *IEEE*, Zhi Li, Hongzi Zhu, *Member*, *IEEE*, Minglu Li, *Member*, *IEEE*, and Qian Zhang, *Fellow*, *IEEE*

Abstract—Traffic estimation is crucial to a number of tasks such as traffic management and road engineering. We propose an approach for metropolitan-scale traffic estimation with probe vehicles that periodically send location and speed updates to a monitoring center. In our approach, we use the flow speed on a road link within a time slot to indicate the traffic condition of the road segment at the given time slot, which is approximated by the average value of probe speeds. By analyzing a large data set of two-year probe data collected from a fleet of around 4,000 taxis in Shanghai, China, we find that a set of probe data may contain a lot of spatiotemporal vacancies over both time and space. This raises a serious missing data problem for road traffic estimation, which results from the naturally uneven distribution of probe vehicles over both time and space. Through empirical study based on the data set of real probe data using principal component analysis (PCA), we have observed that there are hidden structures within the traffic conditions of a road network. Inspired by this observation, we propose a compressive sensing-based algorithm for solving the missing data problem, which exploits the hidden structures for computing estimates for road traffic conditions. Different from existing approaches, our algorithm does not rely on complicated traffic models, which usually require costly training with field study and large data sets. With extensive experiments based on the data set of real probe data, we demonstrate that our proposed algorithm performs significantly better than other completing algorithms, including KNN and MSSA. Surprisingly, our algorithm can achieve an estimate error of as low as 20 percent even when more than 80 percent of probe data are missing.

Index Terms—Probe vehicles, traffic estimation, traffic condition matrix, compressive sensing

1 Introduction

Traffic congestion has a significant negative impact on social and economic activities around many cities in the world [18]. Road traffic monitoring aims to determine traffic conditions of different road links, which is an essential step toward active congestion control. Many tasks, such as trip planning, traffic management, road engineering, and infrastructure planning, can benefit from traffic estimation. As an example, Shanghai, the largest metropolis in China, is undergoing rapid economic growth, but meanwhile suffers constant traffic congestion. To mitigate the burden of the underlying road networks, efficient traffic management is of great importance, and metropolitan-scale traffic estimation is valuable to traffic management.

Traditional approaches for traffic monitoring rely on the use of static traffic sensors, such as inductive loop detectors and video cameras. Vehicle loop detectors and close-circuit video cameras are usually deployed at roadside to detect flow velocity, and traffic density [7], [11], [33]. The coverage of such traditional approach is limited due to the high infrastructure deployment and maintenance cost [17]. This suggests that it is infeasible to install loop detectors and video cameras densely to cover the entire road network.

With the growing prevalence of Global Positioning System (GPS) receivers embedded in vehicles and smartphones, there have been increasing interests in using their location updates or trajectories for monitoring traffic [3], [18]. In this paper, we present an approach to perform metropolitan-scale traffic estimation with probe vehicles. Equipped with a GPS receiver, a probe vehicle can detect its instant location and speed. A probe vehicle periodically sends its location and speed update (or probe data report) to a monitoring center for traffic estimation. Such updates can be transmitted via the data service of a widely available cellular wireless network, such as GSM/GPRS. In Fig. 1, a distribution snapshot of a fleet of probe taxis over the downtown subnetwork of Shanghai is shown.

We consider the traffic condition of a road segment (or link) between two neighboring road intersections or signals in a time slot. In our approach, we use the flow speed on this link within the time slot to indicate this traffic condition, as used by previous studies [6], [33]. Since the flow speed is a random variable, we use the mean of the speed. In practice, we use the average of the speeds of probe vehicles driving on the link in the time slot to approximate the mean of the speed.

This approach to traffic estimation with probe vehicles has salient advantages [18]. First, as these public vehicles traverse most of the road segments in the city, the system provides a large coverage. Second, because of the low cost

- *Y. Zhu and M. Li are with the Shanghai Key Lab of Scalable Computing and Systems, Shanghai, China, and the Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China. E-mail: yzhu@cs.jtu.edu.cn, li-ml@cs.sjtu.edu.cn.*
- *Z. Li and H. Zhu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China. E-mail: will1987@sjtu.edu.cn, yzhu@cs.sjtu.edu.cn.*
- *Q. Zhang is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Room 3533 (via Lift 25-26), Academic Building, Clear Water Bay, Kowloon, Hong Kong. E-mail: qianzh@cse.ust.hk.*

Manuscript received 13 Sept. 2011; revised 29 May 2012; accepted 13 Sept. 2012; published online 1 Oct. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-09-0502.
Digital Object Identifier no. 10.1109/TMC.2012.205.

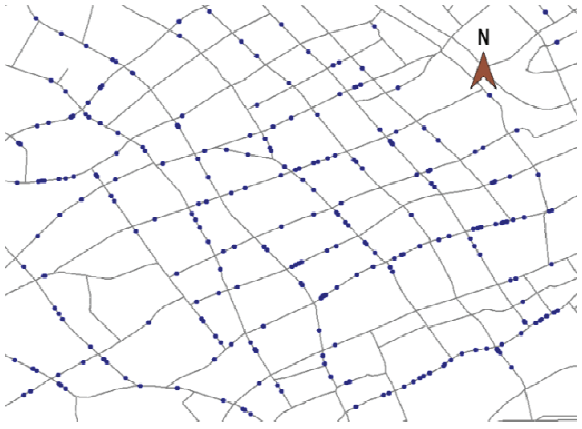


Fig. 1. A subnetwork of Shanghai, China, along with the distribution snapshot of probe vehicles on the road network, where a dot represents the location of a probe taxi.

of onboard GPS receivers, the overall system deployment cost is low.

However, it is challenging to perform traffic estimation with speed measurements from probe vehicles. Because the distribution of probe vehicles over space and time is uneven, the set of received probe data is incomplete over time and space, or contains spatiotemporal sampling vacancies. This raises a serious missing data problem for road traffic estimation. This problem has been confirmed with our analysis on the large data set of real probe data collected from a fleet of 4,000 taxis in Shanghai, China. It is important to note that probe vehicles move at their own wills. They cannot be deliberately controlled to achieve a better coverage over time and space of the set of received speed measurements. In addition, the reception of probe data is vulnerable to the influence of the urban environment. For example, when a vehicle moves through a road with surrounding tall buildings (so-called urban canyons) because of attenuation and multipath propagation of radio signals [15].

There are a few existing studies [5], [6], [18] on traffic estimation with sparse probe data. A probabilistic modeling framework for estimating arterial travel time distributions using sparsely observed probe vehicles is proposed in [18]. They model the evolution of traffic states as a coupled hidden Markov model (CHMM), which can be trained for estimating model parameters. However, this approach assumes that the evolution of traffic states is stationary, which may not be true in the real world. Furthermore, this approach requires costly training with field study and large sets of probe data. In [5], [6], spare bus probe data are used for estimating velocity fields of a road network. Nevertheless, the authors do not address the problem of recovering missing traffic conditions of those road links for which no probe data have been received.

In this paper, we propose a data analytics technique to solving the critical missing data problem inherently associated with the approach for road traffic estimation with probe vehicles. By using principal component analysis (PCA), we analyze the data set of probe data collected from taxis in Shanghai, and reveal that there evidently exist hidden structures with the traffic conditions of a road network. Inspired by this important observation, we propose a compressive sensing-based algorithm for solving the missing data problem, which exploits the hidden structures for computing estimates for road traffic conditions. A matrix of traffic conditions is introduced, where a row represents a time slot and a column represents a road segment. Our algorithm leverages the low rank nature of a traffic condition matrix (TCM) and determines an estimate matrix that complies with the observed traffic conditions derived from the set of received probe data. Experiments based on the large data set of real probe data show that our algorithm significantly performs better than three competing algorithms, i.e., Naïve K-nearest neighbors (KNN), correlation-based KNN, and multichannel singular spectrum analysis (MSSA). We have made the following technical contributions in the paper:

- By analyzing the large data set of real probe data collected from a fleet of around 4,000 taxis with PCA, we reveal that there exist hidden structures with traffic condition matrices.
- We develop an offline data analytics algorithm for solving the missing data problem based on the compressive sensing theory, which explicitly exploits the hidden structures. Some optimization techniques are proposed for computing estimates for those missing traffic conditions.
- Through comprehensive experiments with the data set of real probe data, we show that our algorithm significantly outperforms the competing methods. It can achieve an estimate error of as low as 20 percent even when more than 80 percent of probe data are missing.

The rest of this paper is organized as follows: In Section 2, we present the overview of our approach and formally define the problem of traffic estimation. We reveal the existence of hidden structures in traffic condition matrices and give the design details of the proposed algorithm in Section 3. In Section 4, we present experiment results. We review related work in Section 5. We conclude the paper and introduce future work in Section 6.

2 Overview

In this section, we first introduce the collection of the important data set of real probe data, then define the problem, and finally discuss the missing data problem.

2.1 Collection of Probe Data

Each probe vehicle is equipped with a GPS receiver, which continuously detects instant location and speed. A probe data update includes vehicle identification, instant speed, location in longitude and latitude, and time stamp. The speed is the instantaneous speed directly provided by the GPS receiver.

Note that there may exist a privacy concern if each vehicle simply reports all its locations because it may release sensitive information, such as the home location. It is out of the scope of the paper, however, to address the privacy issue. Some researchers specially focus on privacy issues in the context of traffic monitoring and possible solutions [19], [20] have been provided.

A probe data report is sent from moving vehicles to the monitoring server via a wireless data service provided by a cellular network. The typical data rate of GSM/GPRS is around 20 Kbps. The size of a probe report is relatively small, around 40 bytes. For each vehicle, probe reports are sent to the monitoring server periodically. In our approach, the reporting interval varies from 30 seconds to several minutes. It depends on the availability of GPRS availability. Thus, the bandwidth of GPRS is sufficient to support the delivery of probe data back to the monitoring server.

2.2 Problem Description

We next formally state the problem of traffic estimation with probe data. We first introduce some notations. The set of probe vehicles are denoted as N. For a probe vehicle, $v \in N$, it moves along the roads and sends its location and speed update from time to time. The update of probe data at time t is denoted by $s_v(t) : <id_v, p_v(t), q_v(t), t>$, where id_v is vehicle ID, $p_v(t)$ denotes its location (longitude and latitude), and $q_v(t)$ denotes its speed. Let T_v denote the set of time stamps at which vehicle v sends its probe data, $T_v = \{t_1^v, t_2^v, \ldots, t_k^v\}$, in which t_1^v and t_k^v are the first time stamp and the last time stamp, respectively. Thus, vehicle v has a set of probe measurements, $S_v = \{s_v(t) | t \in T_v\}$. Note that for different probe vehicles, the set of time stamps may be different.

We consider the traffic condition of a road segment between two neighboring road intersections in a given time slot. It is not straightforward to devise a single metric for quantifying the traffic condition of a given road segment at a given time. Many metrics have been proposed in the traffic engineering area for quantifying the traffic condition of a link, such as flow speed, density, length of queues) [6], [14], [18]. We adopt the speed of the traffic flow on the link to indicate the traffic condition of this link, as used in previous studies [15], [33]. Since the traffic flow consists of vehicles traveling on the link, the speed of the traffic flow can be considered as the speed of a vehicle in the flow, which is a random variable. We focus on the mean of the speed to indicate the traffic condition. It is meaningful because if the flow has a higher speed, a vehicle in the flow can generally drive at a higher speed.

Thus, we formally define the traffic condition of a road segment in a given time slot as follows:

Definition 1 (traffic condition). *The traffic condition of a road segment r in a given time slot t denoted as $x_{r,t}$ is defined as the mean of the speed $\vartheta_{r,t}$ of a vehicle driving within the traffic flow on this road segment in the time slot, i.e., $x_{r,t} = E(\vartheta_{r,t})$.*

In practice, we use the average of the speeds of all probe vehicles on the road segment within the time slot to approximate the mean of the flow speed. The average is computed over different probe speeds of all vehicles. In the definition of traffic condition, we have made this assumption that traffic conditions on a segment is uniform during each time slot.

It should be noted that the approach of using average speeds to indicate the traffic state of a road segment has certain limitation. By this approach, the quality of traffic states monitoring is related to the sampling process of probe vehicles. Clearly, as there are more probe data, the quality of resulting traffic states estimation is better. In our work, however, the average value of probe speeds is considered as the real state of a road segment. This work does not explicitly consider the impact of the number of probe samples. This issue will be further investigated by future work.

We are interested in the traffic conditions of a given set of road segments Ω at a given set of time slots T:

$$\Omega = \{r_0, r_1, r_2, \ldots, r_{n-1}\}, \tag{1}$$

$$T = \{t_0, t_1, t_2, \ldots, t_{m-1}\}. \tag{2}$$

The traffic conditions of Ω over T form a traffic condition matrix, denoted by X_{TCM}, or simply X,

$$X_{TCM} = (x_{r,t})_{m \times n}, \tag{3}$$

where $x_{r,t}$ is the traffic condition of road segment r in time slot t.

It is difficult to obtain a complete traffic condition matrix as there may exist many spatiotemporal vacancies with no probe measurements. There is no guarantee that the monitoring server to receive probe measurements for each road segment within every time slot. This raises a serious missing data problem, which will be further demonstrated with empirical study with the data set of real probe data in Section 2.3.

In fact, we are given a measurement matrix $M = [m_{r,t}]_{m \times n}$:

$$\begin{aligned} M_{TCM} &= X_{TCM}. \times B, \\ B &= [b_{r,t}]_{m \times n}, \\ b_{r,t} &= \begin{cases} 0, & \text{if no probe data for } r \text{ in slot } t, \\ 1, & \text{otherwise,} \end{cases} \end{aligned} \tag{4}$$

where the matrix B is an indicator matrix and $.\times$ is an operator of dot product. For two matrices of the same size, $X = [x_{i,j}]_{m \times n}$ and $Y = [y_{i,j}]_{m \times n}$, their dot product $Z = X. \times Y = [z_{i,j}]_{m \times n}$, $z_{i,j} = x_{i,j} \times y_{i,j}$, where $1 \le i \le m$, $1 \le j \le n$. The goal is to obtain an estimate $\hat{X}$ for X when given M, with the objective of minimizing the normalized mean absolute error of the estimate.

Definition 2 (Normalized mean absolute error). *The normalized mean absolute error ξ of an estimate $\hat{X}$ for X is*

$$\xi = \sum_{r,t:m_{r,t}=0} |x_{r,t} - \hat{x}_{r,t}| \Big/ \sum_{r,t:m_{r,t}=0} |x_{r,t}|. \tag{5}$$

Then, we formally define the problem as follows:

Definition 3. (Traffic estimation problem). *Given the set of probe measurements from probe vehicles, the traffic estimation problem is first to obtain the measurement matrix M, and then to find an estimate $\hat{X}$ for the real traffic condition matrix X, with the objective of minimizing the normalized mean absolute error of the estimate $\hat{X}$.*

2.3 Sparse and Uneven Distribution of Observed Probe Data

We show the sparse and uneven distribution of the set of received probe measurements from probe vehicles, which raises the serious missing data problem.

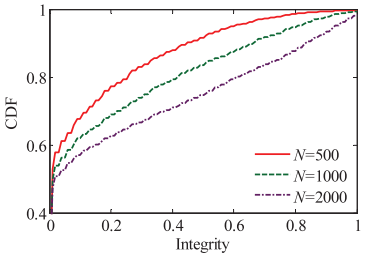


Fig. 2. CDF of integrity of roads.

We first define a metric of *integrity* as follows:

Definition 4. *Let B be the indicator matrix for matrix M. The integrity of M, denoted by $\varpi(M)$, is defined as the fraction of nonzero elements:*

$$\varpi(M) = \frac{\text{sum}(B)}{\text{size}(B)}. \quad (6)$$

We analyze the impact of the number of probe vehicles on the integrity of the measurement matrix by extracting the probe data of a subset of probe vehicles from the complete set of probe data. We analyze the sets of probe data of 500, 1,000, and 2,000 taxis over a duration of 24 hours on February 18, 2007, respectively. All the taxis were running in the inner region of Shanghai, in which there are 5,812 road segments. By default, we set the time granularity (i.e., the length of time slot) to 15 minutes in this empirical study.

First, we study the integrity for a given road segment, by which we can learn the missing data issue over time. Fig. 2 shows the empirical cumulative distribution functions (CDFs) of integrity of all roads under different numbers of vehicles, i.e., 500, 1,000, and 2,000. We can see that when there are 500 probe vehicles, nearly 95 percent of roads have an integrity of less than 60 percent. This means that these roads have no probe measurements for more than 40 percent of time. Generally, when we deploy more probe vehicles, the integrity can be improved. However, even when 2,000 probe vehicles are employed, there are still nearly 80 percent of roads whose integrity is less than 60 percent. More importantly, we find that nearly 50 percent of road segments have an integrity close to zero. This indicates that no vehicles have traveled through these road segments within some single slots.

Next, we consider the integrity at a given time snapshot. In this way, we can learn the missing data issue over space. In Fig. 3, we plot the CDFs of integrity of all time slots under different numbers of probe vehicles, i.e., 500, 1,000, and 2,000. We can see that when there are 500 probe vehicles, nearly 100 percent of time slots have an integrity of less than 18 percent. This indicates that almost for all slots, more than 82 percent of road segments have no probe measurements.

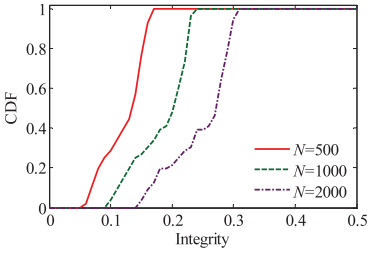


Fig. 3. CDF of integrity of time slots.

TABLE 1
Integrity Summary (February 18, 2007)

Time gran.	N=500	N=1,000	N=2,000
15 min	12.22%	18.28%	24.80%
30 min	18.57%	25.18%	31.61%
60 min	25.53%	31.98%	37.64%

Finally, we study the integrity of measurement matrices for different time granularities. Table 1 shows the integrities under different time granularities when there are 500, 1,000, and 2,000 probe vehicles. We can find that even when there are 2,000 probe vehicles, the integrity is as low as 24.8 percent when the time granularity is 15 minutes and 37.64 percent when the time granularity is 60 minutes.

In summary, we find that the missing data problem is serious. The possible solution to improving the integrity is to deploy more probe vehicles. However, this may increase cost, and be impractical in some situations, for example, there is no way to employ more probe vehicles.

3 Compressive Sensing-Based Algorithm

The goal is to compute an estimate of the traffic condition matrix for the real traffic condition matrix with the objective of minimizing the estimate error. We propose a compressive sensing-based algorithm to effectively exploit the hidden structures. Compressive sensing [12] is an effective technique for exploiting the hidden structures of real-world data sets for tasks such as compression and signal reconstruction. In this section, we first reveal the existence of hidden structures with traffic condition matrices, then give the preliminary of compressive sensing, and finally delve into the design details of the algorithm.

3.1 Revealing Hidden Structure

The traffic conditions of different road segments over different times are not independent. There exist structures. We reveal such hidden structures by using principal component analysis. We use the same data set of probe data collected from the fleet of taxis in Shanghai, China, as introduced previously.

PCA is an effective nonparametric technique for revealing sometimes hidden, simplified structure that often underline a data set. It is a commonly used technique for analyzing high-dimensional data (or structures) [24]. Given a high-dimensional data set such as a matrix of traffic conditions corresponding to a set of road segments and a set of time slots, and its associated coordinate space, PCA can find a new coordinate space that is the best one to use for dimension reduction of the given data set. After finding this new coordinate space, we can project the high-dimensional data set onto a subset of the axes with the objective of minimizing the error. In summary, given a high-dimensional data set, we can find a small data set to approximate the original high-dimensional data set.

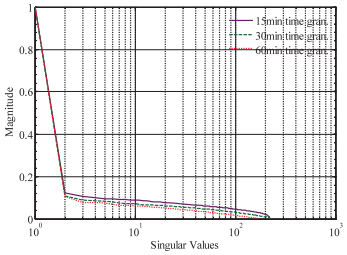


Fig. 4. Magnitude of singular values (with Log scale X-axis).

Any matrix X can be decomposed to three matrices according to the singular value decomposition (SVD):

$$X = USV^T = \sum_{i=1}^{\min(n,m)} \sigma_i u_i v_i^T, \tag{7}$$

where V^T is the transpose of V, U is a $n \times n$ unitary matrix (i.e., $U^TU = UU^T = I_{n\times n}$), V is a $m \times m$ unitary matrix (i.e., $V^TV = VV^T = I_{m\times m}$), and S is a $n \times m$ diagonal matrix constraining the singular values σ_i of X. Let σ_i be larger than σ_{i+1}, $i = 1, 2, \ldots, l$, where l is the rank of X . The rank of a matrix equals the number of its nonzero singular values. Here, v_i is the unit eigenvector of X^TX corresponding to the ith principal component. We call u_i an eigenflow of X [24]:

$$u_i = (Xv_i)/\sigma_i, \quad i = 1, 2, \ldots, \min(m, n). \tag{8}$$

According to (7), σ_i is a coefficient of the ith principal component, which we may explain as the energy of the ith principal component.

In Fig. 4, we present the magnitude (ratio to the maximum) of singular values. This figure suggests that most of the energy is contributed by the first few principal components. The existence of the sharp knee is a result of some common structures among different interested road segments, which will lead the traffic condition matrix to a low rank.

The information of a data set is mainly contained by the first few components. We reconstruct the traffic matrix using only the first five principal components. Fig. 6 shows the reconstructed traffic condition over times of a given road segment in which the time granularity is 30 minutes. We can see that the reconstructed traffic conditions sketches the variation of the original ones quite well. The root mean square error between the two series of traffic conditions is around 9.67.

Then, we look at the characteristics of eigenflow u_i. A time series X_i can be presented as a linear combination of u_i with associated weight $(V^T)_i$:

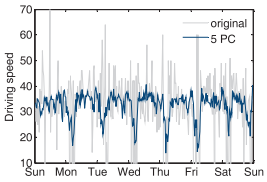


Fig. 6. Original and reconstructed traffic conditions of a given road segment using first five principal components (granularity is 30 minutes).

$$X_i = \sigma_i U(V^T)_i, \quad i = 1, 2, \ldots, \min(m, n), \tag{9}$$

where $(V^T)_i$ is the ith row of V.

All the eigenflows can be divided into three types. Let $C(u_i) \in \{1, 2, 3\}$ denote the type of an eigenflow, $v_i, 0 \le i < \min(m, n)$. Its type is determined as follows:

$$C(u_i) = \begin{cases} 1, & \text{if } |FFT\ (u_i)\ | \text{ contains a spike,} \\ 2, & \text{if } u_i \text{ contains a spike,} \\ 3, & \text{otherwise.} \end{cases} \tag{10}$$

Note that the construction of three types are mutually exclusive. If the difference of the value and the average is larger, then four times the standard deviation, the value is a spike.

When the signal of an eigenflow is periodic (i.e., its FFT energy contains an evident spike), this eigenflow belongs to the first type. An eigenflow of the first type is considered as deterministic, i.e., this type of eigenflows contains the majority of information in the data sets. If an eigenflow does not belong to the first type and its signal contains a spike, it belongs to the second type. The spike in the eigenflow of the second type indicates that the original data sets also have a corresponding spike. The rest of the eigenflows belong to the third type. An eigenflow of the third type contains little information and can be considered as containing only noises.

The previous explanation is illustrated in Figs. 5 and 7. In Fig. 7, we reconstruct the traffic conditions over time at a given road segment by using only the basis corresponding to the specific type of eigenflows. We find that the first type contains most information and sketches the variation of the original series of traffic conditions quite well. The second type of eigenflows contribute signal spikes, and the third type contains little information with a mean value close to zero.

Fig. 8 shows the occurrences of eigenflow types in the increasing order of singular values. The most important information often comes from the eigenflows of first type, which correspond to singular values.

In summary, our empirical study using the principal component analysis demonstrates that there are hidden

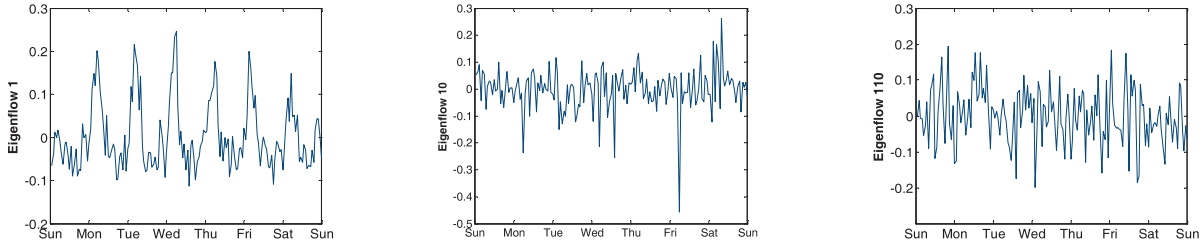


Fig. 5. Time series represented by three types of eigenflows (left: the first type; middle: the second type; right: the third type).

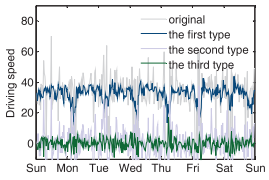


Fig. 7. Reconstructed traffic conditions of a given road segment by using different types of eigenflow.

structures with traffic condition matrices. This lays the foundation for our compressive sensing-based algorithm for estimating missing traffic conditions.

3.2 Preliminary of Compressive Sensing

We have revealed that there exist hidden structures in traffic condition matrices. Compressive sensing [4] is an effective technique for a number of tasks, such as data compression and signal processing [10], [12]. The main idea of compressive sensing is that signals or data sets in the real world often contain structures or redundancy (i.e., they are not pure random noises). This nature can be used as prior knowledge for compression and reconstruction of signals or data sets.

Mathematically, a vector with only a few nonzero elements is called a sparse vector. Structure or redundancy in data sets is synonymous with *sparsity*. As previously shown principal component analysis, a matrix of data set may have only a few large components and many small components. Such a vector is considered as compressible, in the sense that most of its information is actually carried by the large elements. A sparse matrix can be well approximated by a low rank matrix.

As shown in Section 3, any matrix can be decomposed in such a way that it equals the multiplication of three component matrices. When the rank is fixed and set to r, to generate an estimate that approximates the original matrix, we keep the r largest components in (7) and drop the others. Thus,

$$\hat{X} = \sum_{i=1}^{r} \sigma_i u_i v_i = \sum_{i=1}^{r} \sigma_i A_i. \tag{11}$$

This $\hat{X}$ is known as the best rank-r approximation with respect to the Frobenius norm $||\cdot||_F$ of approximation errors, $||X||_F \triangleq \sqrt{\sum_{i,j} X_{ij}^2}$ for any matrix [23]. Then, $\hat{X}$ is the solution to the following optimization problem:

$$\begin{aligned} &\min ||X - \hat{X}||_F \\ &s.t.\ \mathrm{rank}(\hat{X}) \leq r. \end{aligned} \tag{12}$$

3.3 Algorithm Design

To solve the traffic estimation problem, we are given the measurement matrix and required to compute an estimate of the original matrix. It is impossible to directly apply (12) as we do not have the knowledge of the original matrix and the proper rank.

As a good estimate, it is reasonable to be as close as to the measurement matrix. In addition, the estimate matrix should have a low rank as we have revealed in the real data sets that they contain certain structures or redundancy. Thus, we try to find the low rank estimate:

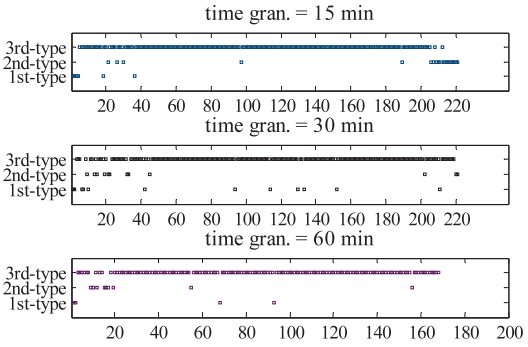


Fig. 8. Occurrence of eigenflow types in the corresponding order of singular values.

$$\begin{aligned} &\min\ \mathrm{rank}(\hat{X}) \\ &s.t.,\ B. \times \hat{X} = M. \end{aligned} \tag{13}$$

It is difficult to solve this minimization problem because it is nonconvex.

To circumvent the difficulty, we make use of the SVD-like factorization, which rewrites (13) as follows:

$$\hat{X} = U\Sigma V^T = LR^T, \tag{14}$$

where $L = U\Sigma^{1/2}$ and $R = V\Sigma^{1/2}$. According to the compressive sensing literature [9], [30], [31], we can solve a simpler problem and obtain an equivalent result under a certain condition. Specifically, if the restricted isometry property [30] holds, minimizing the nuclear form can perform rank minimization exactly for a matrix of low rank. Here, if the rank of X is smaller than that of LR^T, then we can apply this technique. That is, we just find matrix L and R that minimize the summation of their Frobenius norms:

$$\begin{aligned} &\min ||L||_F^2 + ||R^T||_F^2 \\ &s.t.,\ B. \times (LR^T) = M. \end{aligned} \tag{15}$$

In practice, L and R that strictly satisfy the constraint are likely to fail for two reasons. First, there are noises in the probe data, and therefore strict satisfaction may lead to the overfit problem. Second, the rank of a traffic condition matrix is only approximately low.

Thus, we use the Lagrange multiplier method to solve (15):

$$\begin{aligned} &\min\ (a + \lambda b), \text{where} \\ &a = ||B. \times (LR^T) - M||_F^2, \\ &b = ||L||_F^2 + ||R||_F^2. \end{aligned} \tag{16}$$

The Lagrange multiplier λ controls the tradeoff between rank minimization and measurement fitness.

Many ways can solve the above optimization problem. We propose an algorithm that is similar to the one in [37]. We show the detail pseudocode of this algorithm in Fig. 9. The algorithm starts with a random initialization to matrix L. It first fixes matrix L and then computes matrix R. Next, R is fixed and L is computed. This process repeats for a fixed number of iterations. In implementation, we perform experiments to find a good setting of the number of iterations.

In each iteration, we have to compute either L or R to minimize the objective (16). We find that reaching the

Algorithm 1: **Estimating matrix of traffic conditions.**
Input:
$M_{m\times n}$: measurement matrix
$B_{m\times n}$: indicator matrix
r: rank bound
λ: tradeoff coefficient
t: iteration times
Output:
$\hat{X}_{m\times n}$: estimate matrix

1. $L \leftarrow$random_matrix(m, r);
2. **for** $k \leftarrow 1$ **to** t **do**
3. $R \leftarrow$inverse$([L; \sqrt{\lambda}I], [M; 0])$;
4. $L \leftarrow$inverse$([R^{\mathrm{T}}; \sqrt{\lambda}I], [M^{T}; 0])$;
5. $v \leftarrow ||B.\times (LR^{\mathrm{T}}) - M||_F^2 + \lambda(||L||_F^2 + ||R^{\mathrm{T}}||_F^2)$;
6. **if** $v < \hat{v}$ **then**
7. $\hat{L} \leftarrow L$; $\hat{R} \leftarrow R$; $\hat{v} \leftarrow v$;
8. **end if**;
9. **end for**
10. $\hat{X} \leftarrow \hat{L} \times \hat{R}^{\mathrm{T}}$;
11. return $\hat{X}$;

// return solution to contradictory equation
procedure inverse$(\boldsymbol{P}, \boldsymbol{Q})$
1. $C \leftarrow P^{\mathrm{T}}P \setminus P^{\mathrm{T}}Q$;
2. return C;

Fig. 9. The pseudocode of Algorithm 1.

objective is equivalent to making both x and y equal to zero simultaneously. Thus, we have the following when L is given:

$$\begin{bmatrix} B.\times (LR^{T}) \\ R \end{bmatrix} = \begin{bmatrix} M \\ 0 \end{bmatrix}. \tag{17}$$

This is a contradictory equation because the number of constraints is larger than that of unknown variables. By computing the best approximate solution to this contradictory equation with least squares, we can compute the best matrix R for satisfying (16).

We analyze the complexity of the algorithm as follows: The key operation of Algorithm 1 is the procedure for computing an inverse matrix, which gives the best approximate solution to the contradictory equation. The procedure is essentially completed by a matrix multiplication. Therefore, its complexity is $\mathcal{O}(rmn)$ where r, m, n denote the column number of L, the row number of X, and the column number of X, respectively. The algorithm repeats the procedure for t times. Therefore, the total complexity of the algorithm is $\mathcal{O}(rmnt)$. Note that t is a design parameter of Algorithm 1. Through experiments, we find that the setting of $t = 100$ makes the algorithm to converge to a steady output when the matrix size is of hundreds by hundreds.

3.4 Design Optimizations

Two important parameters must be determined in Algorithm 1, i.e., rank bound r and tradeoff coefficient λ. The two parameters greatly influence the final estimate quality. According to the principle of compressive sensing, the rank of the approximated matrix should be minimized. In

Algorithm 2: Finding optimal parameters.
Input:
$\ell_r, \mathcal{U}_r$: lower bound and upper bound of r
$\ell_\lambda, \mathcal{U}_\lambda$: lower bound and upper bound of λ
B: measurement matrix
Algorithm 1
Output:
Optimal r and λ

1. $\mathcal{N}$(*population*) $\leftarrow$ initialize with random numbers uniformly distributed within $[\ell_r, \mathcal{U}_r]$ and $[\ell_\lambda, \mathcal{U}_\lambda]$
2. **while** (!stall(fitness)) **do**
3. $\mathcal{H} \leftarrow$ select$(\mathcal{N})$
4. $\mathcal{C} \leftarrow$ crossover$(\mathcal{N})$
5. $\mathcal{M} \leftarrow$ mutate$(\mathcal{N})$
6. $\mathcal{N} \leftarrow [\mathcal{H}, \mathcal{C}, \mathcal{M}]$
7. **end while**
8. $[r, \lambda] \leftarrow$ decode (the best individual in $\mathcal{N}$)

Fig. 10. The pseudocode of Algorithm 2.

Algorithm 1, r is the number of columns in matrix L and R, which is smaller than m and n. Thus, we have

$$\mathrm{rank}(\hat{X}) \leq \min(\mathrm{rank}(L), \mathrm{rank}(R)) = r. \tag{18}$$

Thus, r is an upper bound of rank($\hat{X}$), and impacts the algorithm performance.

We should determine the optimal parameters in order for Algorithm 1 to obtain the best performance in terms estimate error. However, it is not trivial to determine the optimal parameters. The quality of estimation is a function of the two parameters, denoted by, $\ell = f(r, \lambda)$. Then, to obtain the optimal parameters, the objective is the following:

$$\max \ell = \max f(r, \lambda). \tag{19}$$

We use estimate error to indicate the quality of estimation. The definition of estimate error will be given in the next section. The key issue is that function $f(\cdot)$ characterizing the relationship between error and the parameters is invisible.

We propose a genetic algorithm for deriving the optimal parameters of rank bound and tradeoff coefficient. The strength of this algorithm is that the analytical form of the objective is not needed. In this algorithm, estimate errors are used as fitness. We encode the two parameters as a vector that contains two real numbers. The pseudocode of Algorithm 2 is shown in Fig. 10.

We explain the main steps of the algorithm in the following:

1. *Initialization*. We randomly initialize the population representing the two parameters. The size of population is a design parameter of this algorithm.
2. *Selection*. Each individual is evaluated against fitness. The fitness function is the estimate error, which can be evaluated by invoking Algorithm 1 with the parameters encoded by each individual. Then, the best individuals are selected to breed the next generation.
3. *Reproduction*. Besides the group of individuals selected in the selection process, the next generation

also includes two other groups. By employing the roulette model, one group of offsprings are produced by taking the crossover of any two individuals, and the other group of offsprings are produced by the mutation operation. Specifically, we assign a random value to one of parameters within its domain to achieve the mutation.

4. *Termination*. The algorithm can terminate after a fixed number of integrations or after a threshold on fitness improvement is met. We adopt a fixed number of iterations as the termination criterion.

There are several design parameters with Algorithm 2, including bounds of rank bound and tradeoff coefficient, size of population, and number of iterations. The lower bound of rank bound r can be set to 1 because it is positive, and its upper bound is given by (18). It is not easier to determine the bounds of tradeoff coefficient, we determine the bounds by experiments. The size of population and the number of iterations are also determined by experiments.

The time complexity of Algorithm 2 can be high because each time an individual is evaluated its fitness, Algorithm 1 should be invoked to get the estimate error. Fortunately, however, Algorithm 2 is only executed once for a given set of road segments. With experiments, we find that for a given set of road segments, the two parameters obtained by Algorithm 2 are stable over different times.

4 Experiments and Analysis

We have performed extensive experiments for evaluating the performance of the proposed algorithm for traffic estimation. In the following, we first present the methodology and the experimental setup. The compared algorithms are then introduced. Finally, performance results are presented and discussed.

4.1 Methodology and Experimental Setup

We adopt a comparative study, comparing our algorithm with other competing algorithms that will be introduced in the next section.

Experiments are conducted with two data sets of probe data. One data set is from the fleet of 4,000 taxis in Shanghai, as introduced before, and the other data set of probe data is from a fleet of 8,000 taxis in Shenzhen, China. Both data sets of probe data span a duration of one week. Three time granularities, i.e., 15, 30, and 60 minutes, are used.

We choose a subnetwork of 221 road segments in Shanghai, and a subnetwork of 198 road segments in Shenzhen for experiments. Both subnetworks are from a region close to city centers. In comparison, Shanghai is more dense than Shenzhen, in terms of distribution of probe vehicles. The major reason for choosing downtown regions is that we need to know the original traffic condition matrix as the ground truth. In reality, it is very difficult to find a fully integral matrix without vacancies. For this reason, it is better to find a matrix that is as integral as possible. When performing experiments, we randomly discard some elements to form measurement matrices. Then, these estimates are compared with the original matrices and estimate errors can be computed because the original matrices have only a few unavailable elements. Note that the calculation of estimate error does not include those elements that are unavailable in the original matrices.

4.2 Compared Algorithms

We compare our algorithm with three other algorithms.

4.2.1 Naïve KNN

K-Nearest Neighbors is a simple algorithm but often used to solve many machine learning problems including recovery of missing values. The naïve KNN interpolates missing values by taking the average of its nearest K neighbors in the measurement matrix.

4.2.2 Correlation-Based KNN

The correlation-based KNN is more sophisticated compared with the naïve one. It calculates the average by using the K neighbors from its immediate rows or columns. In the following, we use rows as example. The key idea is that for average computation, the candidate value is weighed by the coefficient of the current row and the candidate row:

$$w_{i,k} = |C_{i,k}| \Big/ \sum_{t=i\pm1, i\pm2} |C_{i,k}|. \tag{20}$$

Thus, the estimate for a missing element is computed by

$$x_{i,j} = \sum_{k=i\pm1, i\pm2} x_{k,j}\, w_{i,k}, \tag{21}$$

where C_{ik} is the correlation coefficient of row i and k.

4.2.3 Multichannel Singular Spectrum Analysis

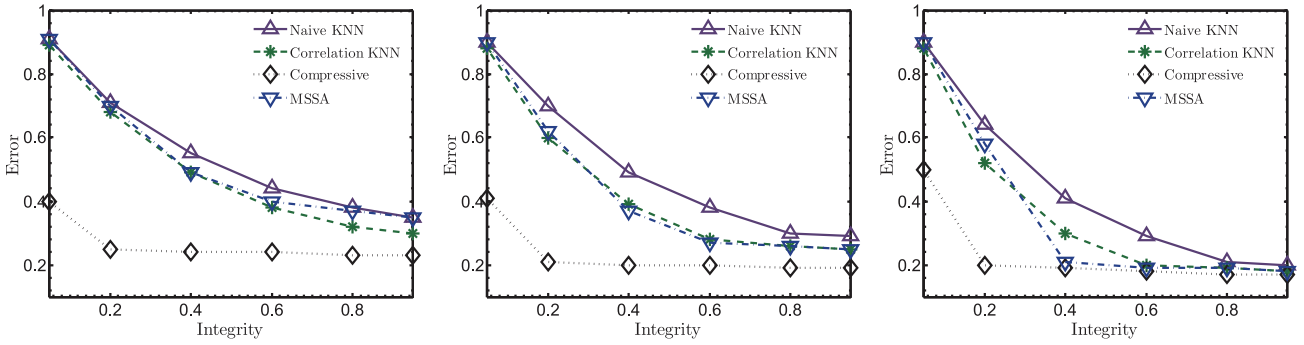
MSSA is often used to solve missing data problems, for example, geographic data and meteorological data. It is a data adaptive and nonparametric method based on the embedded lag-covariance matrix. We adopt an iterative procedure proposed in [40] that utilizes the internal periodicity of traffic conditions.

4.3 Impact of Integrity

The four algorithms are compared in terms of estimate error when the integrity of the traffic matrix is varied. In Naïve KNN, K is set to 4. In the correlative KNN, K is also set to 4. And in MSSA, the parameter M is set to 24 as suggested by Zhu et al. [40]. According to the result of Algorithm 2, we set r and λ in Algorithm 1 to 2 and 100, respectively.

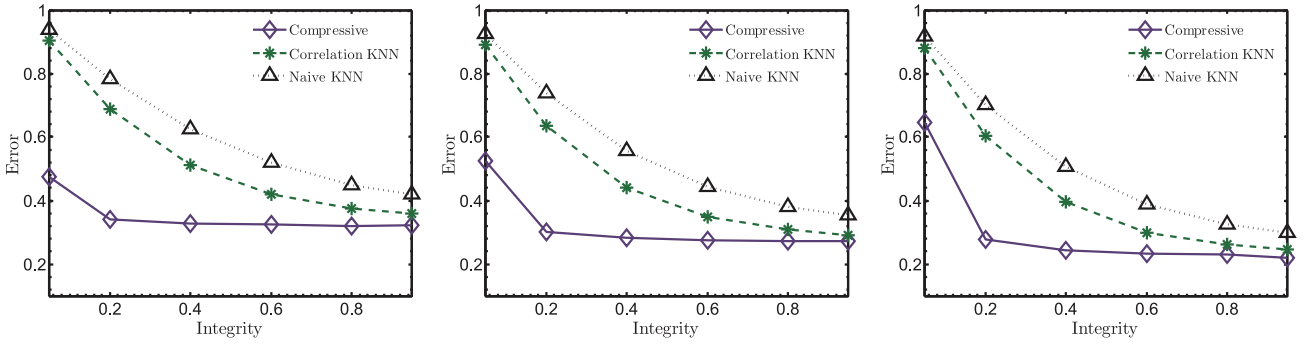
In Fig. 11, the performance of the four algorithms in terms of estimate error with Shanghai data set is shown. Three time granularities are used, i.e., 15, 30, and 60 minutes. We can see that our algorithm performs the best among all the algorithms under every time granularity. Naïve KNN performs the worst. Correlation-based KNN and MSSA are better than naïve KNN, but worse than our algorithm. The two algorithms, correlation-based KNN and MSSA, produce almost similar performance of estimate error.

We can also find that when the integrity of the traffic condition matrix decreases, our algorithm steadily produces low estimate errors. That is, the performance of our algorithm is relatively insensitive to the integrity of measurement matrices. Even when the integrity is as low as 20 percent, the estimate error is no more than 20 percent when the time granularity is 60 minutes. This shows that our



(a) Time gran. = 15 min (b) Time gran. = 30 min (c) Time gran. = 60 min

Fig. 11. Estimate error versus integrity for different time granularities (with Shanghai Data Set, # of road segments $= 221$, # of probe vehicles $= 2{,}000$, time length = one week).



(a) Time gran. = 15 min (b) Time gran. = 30 min (c) Time gran. = 60 min

Fig. 12. Estimate error versus integrity for different time granularities (with Shenzhen Data Set, # of road segments $= 198$, # of probe vehicles $= 8{,}000$, time length = one week).

algorithm can reliably recover the missing elements when just a few elements are available. In contrast, the rest algorithms including naïve KNN, correlation-based KNN, and MSSA have worse performance when the integrity becomes poorer. The reason is that our compressive sensing-based algorithm can effectively capture the internal structures that exist in the data set even just a few data points are used, while the rest algorithms fail to achieve this.

From Fig. 11, we can also see that the estimate error becomes higher when the time granularity is smaller for all algorithms. It is mainly due to the fact that the hidden structure feature of the traffic condition matrix becomes weaker because of average speeds in the traffic condition matrix would experience more variations over time when the time window is smaller. Our approach accordingly becomes less capable to accurately recover missing values.

We observe that as the integrity increases from 0.05 to 0.95, the estimate error achieved by our compressive-based algorithm first quickly decreases before the integrity is 40 percent and then further becomes smaller but the speed of decrease is very small. This shows that the compressive sensing-based algorithm has the strength that using only a small subset of the complete set of traffic conditions it is able to capture the majority information of the complete data set.

However, there consistently remains an estimate error of around 20 percent even if the integrity is as high as 95 percent. There are two main reasons. First, in the real world a traffic condition may contain unpredictable randomness, which are unable to be captured by other traffic conditions. Second, there is limitation with our compressive-based approach, which mainly focuses on linear structures of a traffic condition matrix while a real traffic condition matrix has other kinds of structures.

In Fig. 12, the performance of the algorithms with Shenzhen data set is shown. Since MSSA runs very slowly, we do not include MSSA in this experiment. We can find similar results as in Fig. 11. By comparing the impact of the two data sets, we find that the estimate error with Shenzhen data set in the same configuration is higher than that of Shanghai. This is because the probe taxies in Shanghai is more densely distributed over the subnetwork under investigation.

We further show the distribution of individual errors in Figs. 13 and 14. Since absolute errors may differ dramatically, we instead study relative errors. A relative error of an

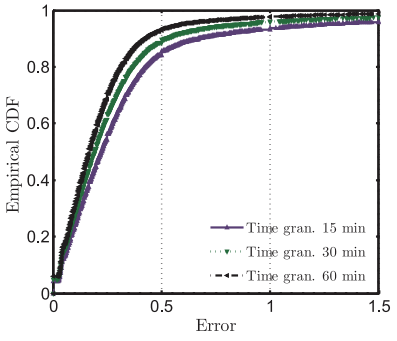


Fig. 13. CDFs of relative errors with different time granularities (Integrity $= 20$ percent, Shanghai data set).

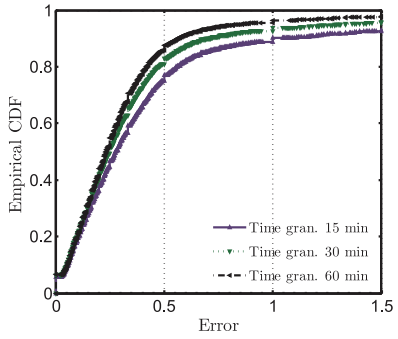


Fig. 14. CDFs of relative errors with different time granularities (Integrity $= 20$ percent, Shenzhen data set).

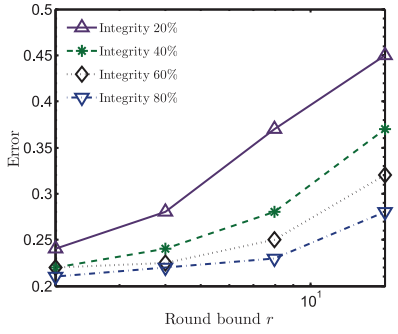


Fig. 15. Estimate error against rank bound r ($\lambda = 1$, granularity $= 30$ minutes, Shanghai data set).

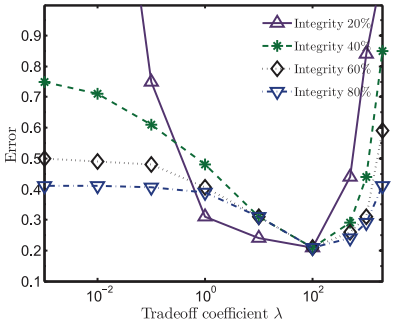


Fig. 16. Estimate error against tradeoff coefficient λ ($r = 32$, granularity $= 30$ minutes, Shanghai data set).

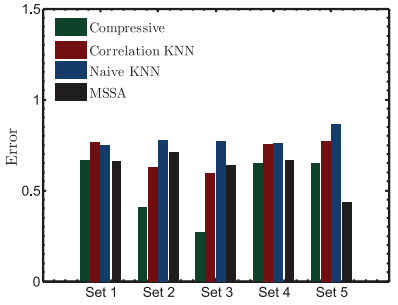


Fig. 17. Estimate errors achieved by different matrices formed by different road segments (time granularity $= 30$ minutes, integrity $= 20$ percent, Shanghai data set).

estimated element is defined as $|\hat{x}_{ij} - x_{ij}|/x_{ij}$. The experiments are conducted with integrity of 20 percent. In Fig. 13, we can find that 80 percent of estimated elements have a relative error of smaller than 0.25 when the time granularity is 60 minutes. Even when the time granularity is 15 minutes, the relative error for nearly 80 percent of estimated elements is less than 0.38. In Fig. 14, we can find consistent results.

4.4 Impact of Rank Bound and Tradeoff Coefficient

As mentioned before, Algorithm 1 has two important parameters, i.e., rank bound and tradeoff coefficient. The parameters impact the performance of the algorithm. We have proposed the genetic-based algorithm for finding the optimal parameters. In the following, we conduct experiments to study the impact of these parameters and show that it is important to design the algorithm for finding the optimal parameters. The experiments are conducted with Shanghai data set.

First, we study the impact of rank bound r. In Fig. 15, the error rates against different rank bounds are plotted. In this experiment, the time granularity is 30 minutes and λ is set to one. We find that the estimate error is lowest when the rank bound is two. The main reason is that when the rank of $\hat{X}$ is low, the estimate matrix embodies the major trend of variation of the original matrix. When the rank of $\hat{X}$ grows, the estimate matrix tries to describe more information but is often misled by measurement errors. This increases the estimate error.

We also study the impact of tradeoff coefficient λ. For ease of studying its impact, we set rank bound r to 32. In Fig. 16, estimate errors against different tradeoff coefficients are shown. We find that the estimate error changes significantly when the tradeoff coefficient changes from 0.001 to 2,000. The optimal coefficient is around 100 when the rank bound is 32. According to (16), a larger λ puts more weight to rank minimization and a smaller λ more emphasizes measurement fitness. A good tradeoff coefficient should strike a balance between rank minimization and measurement fitness.

4.5 Impact of Traffic Matrix Selection

We next explore the impact of traffic matrix formation on the estimation quality of a given road segment. According to the definition of traffic condition matrix in (3), a column in a traffic matrix represents a road segment, and a row represents a time instance. For traffic estimation, we can form different traffic matrices by selecting different road segments. For this study, we focus on the estimation quality of a given road segment, denoted as r_0, when we select different sets of road segments for constructing traffic matrices.

We construct five different traffic matrices by selecting five sets of road segments as follows: Note that each set contains r_0. Set 1 has six other road segments all directly connected with r_0. Set 2 consists of 18 road segments within two blocks but excluding those directly connecting ones. Set 3 has 45 randomly selected road segments from the rest set of road segments excluding Set 2 and Set 3. Set 4 contains six road segments randomly selected from Set 2. Set 5 contains six road segments randomly selected from Set 3.

Estimate errors achieved by different algorithms with the five different traffic matrices with 20 percent integrity and 40 percent integrity are shown in Figs. 17 and 18, respectively. We find that when the number of road segments in the matrix is small and fixed, there is no significant difference when we use different road segments to construct the traffic matrix. In addition, the performance gain of our algorithm over other algorithms are not significant.

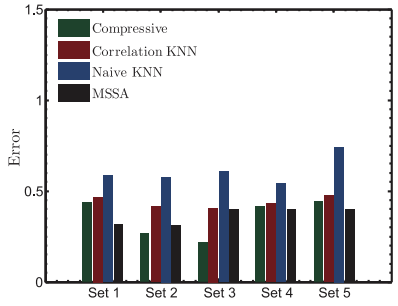


Fig. 18. Estimate errors achieved by different matrices formed by different road segments (time granularity $= 30$ minutes, integrity $= 40$ percent, Shanghai data set).

However, as the number of road segments increases, for example, from Set 1 to Set 2 and Set 3, the performance advantage of our algorithm becomes more evident. In the case of Set 3, which contains 45 road segments, the estimate error achieved by our algorithm is significantly better than the other competing algorithms.

Thus, our proposed algorithm prefers to constructing larger matrices with more road segments, which is beneficial to making use of more evident hidden structures within traffic condition matrices.

4.6 Runtimes

We finally investigate the time efficiency of different algorithms. Experiments are run on a desktop computer with Windows XP, Core i7 870 processor of eight cores, 4-GB memory, and 500-GB disk. All algorithms are implemented with MatLab, and the version of MatLab is v7.4 (R2007a). The Shanghai data set is used and the matrix of traffic conditions is for 221 road segments over a time length of one week.

The runtimes of different algorithms are shown in Table 2. We find that MSSA requires very long runtimes, and other algorithms are time efficient. When the time granularity is 15 minutes, it costs MSSA 5.32×10^3 (about 1.48 hours) to finish, our proposed algorithm only 0.827 second, and Naïve KNN and Correlation-based KNN less than 0.1 second.

In summary, although Naïve KNN and Correlation-based KNN are also time-efficient algorithms, they provide poor performance of estimate error. Our proposed algorithm requires short runtime but produces good performance of estimate error. MSSA is not time efficient and may not be practical when the number of road segments is large and the time of interest is long.

TABLE 2
Runtimes of Different Algorithms

Algorithms	Time Granularity		
	15 Min	30 Min	60 Min
Naïve KNN	2.20e-2	1.56e-2	6.20e-3
Correlation KNN	3.10e-2	2.18e-2	1.60e-2
Compressive	8.27e-1	4.99e-1	2.97e-1
MSSA	5.32e+3	3.61e+3	2.59e+3

5 Related Work

In this section, we review related work and outline differences of our work from existing work.

5.1 Traditional Traffic Monitoring

Close-circuit cameras and vehicle loop detectors are two traditional methods for estimating traffic conditions. By installing cameras in road intersections, we can analyze the video screen manually, or by image processing [7], to estimate traffic conditions at the road segments, where the cameras are installed. It suffers the coverage problem and is limited by the complexity of image processing algorithms.

A more common method is to deploy inductive circuits under the road surface [11]. When a vehicle passes above, it produces a signal. According to the time interval of two consecutive signals, we can calculate the speed of this vehicle and evaluate the number of vehicles on the road. It suffers the limited coverage and high cost problem as well.

Many traffic estimation methods have been developed, which rely on density-based traffic models, for example, Lighthill-Whitham-Richards (LWR) partial differential equation (PDE), and Cell Transmission Model (CTM) in the transportation literature) [33]. Such models work with traditional traffic sensors, which measure vehicle flows and occupancies from which vehicle densities can be computed.

In response to the growing availability of probe data from vehicles and mobile smartphones, some researchers have proposed to use the flow speed of a road segment as traffic state [6], [33]. Our work adopts the same idea of using the flow speed of a road segment to indicate the traffic condition of a road segment.

5.2 Traffic Monitoring with Probe Vehicles or Mobile Phones

With the prevalence of GPS receivers embedded in vehicles and smartphones, there has been increasing interest in using their location updates or trajectories for monitoring traffic of road networks [16].

Ferman et al. [13] discuss the architecture of probe vehicle systems and develop a simple analytical/statistical model. They figure out that 3 percent penetrate is needed on highway and over 5 percent penetrate is required on surface roads. However, they do not offer any method for dealing with the issue of insufficient samples.

Yoon et al. [34] focus on how to figure out the street traffic states on a given road segment based on probe vehicle's trace data. They drive a car with a GPS receiver in a given route in Ann Arbor and collect GPS information every 4 to 10 seconds. They classify traffic states according to vehicle's spatial average speed and temporal average speed. Such a method, however, requires abundant data and each road segment is analyzed independently. They also show that the driving speed of one road segment exhibits some regular patterns.

In [35], Yuan et al. also employ the idea of understanding road traffic with GPS-equipped taxis. They build a cloud for incorporating a number of factors such as day of week, time of day, and weather. With such information, the cloud provides a driving direction service.

In [22], a fusion-based method is proposed to combine sensing data provided by loop detectors and probe vehicles

for traffic monitoring. This method considers that each single source is inaccurate and the better way is to combine multiple sources. This method assumes that for a given road segment sensing data from loop detector and from probe vehicles are both available, which may not be true in a realistic setting.

Mobile Millennium [1] is a project that includes a pilot traffic-monitoring system that uses GPS receivers in cellular phones to gather traffic information, process it, and distribute it back to the phones in real time.

The performance of a system for measuring traffic speeds and travel times using information from mobile phones is studied in [3]. Through comparison with dual magnetic loop detectors, it is shown that the mobile phone-based system for traffic monitoring can be useful in real world applications.

Mohan et al. [29] focus on the monitoring of road traffic conditions using smartphones. They leverage embedded sensors on smartphones, such as accelerator, microphone, and GPS sensors. A system called Nericell is developed to perform rich sensing by piggybacking on smartphones.

5.3 Sparse Data for Traffic Estimation

Some studies have been devoted to traffic estimation using sparse probe data.

In [18], Herring et al. propose a probabilistic modeling framework for estimating arterial travel time distribution using sparsely observed probe vehicles. They consider that the traffic state of a road segment is invisible and it impacts the vehicle speed traveling on this road segment. Thus, they model the evolution of traffic states as a coupled Hidden Markov Model, in which traffic states of nearby road segments are correlated and evolve over time in a Markov manner. After training the model, it can be used to compute the average travel time for each link. The main difference of our work from this work is that the CHMM must be trained with a sufficiently data set but there is no training needed for our approach. In addition, their approach assumes that traffic evolution of a road network is stationary but in the real word it may not true.

In [38], the authors study the problem that travel times recorded by probe vehicles are for partial links or for a partial route travel. They try to split travel times between two consecutive time stamps to individual links. They do not specifically consider the problem that some links may be covered by no probe vehicles in some certain durations.

In [5], Bejan et al. study the feasibility of using public buses to estimate journey times experienced by road users. They analyzed sparse probe data collected from a fleet of over 100 buses. A probe data report of the bus probe data contains only location and time stamp. The authors propose a method of computing the speed of a bus at a given location. They first interpolate the locations with a spline technique and then compute the speed by taking the derivative.

Based on the previous work in [5], Bejan and Gibbens [6] propose to use bus speeds recovered by sparse bus probe data to indicate the traffic condition of a link. They explain why bus speeds can be used as an indicator of traffic condition and show how bus speeds can be aggregated to obtain the velocity fields of a road network. There is no validation, however, on the accuracy of the velocity fields constructed by using bus probe data.

The studies in [5] and [6] make use of sparse bus probe data. However, they do not solve the problem of recovering velocity fields for those links that are not covered by any buses.

5.4 Mobile Sensing with Vehicles

In addition to traffic monitoring, vehicles as powerful mobile sensors can be used in a variety of mobile sensing applications [25]. A good survey on urban vehicular sensing platforms is offered in [25].

It is reported [27] that vehicular sensor networks have emerged as a new paradigm for proactive urban monitoring. A vehicular sensor can sense events, process data and deliver it for further analysis. MobEye [26] is a protocol useful for vehicular urban sensing. It opportunistically diffuses sensed data summaries among mobile vehicles and to create index for querying monitoring data.

In [8], [21], a data management system called CarTel is proposed for querying and collecting data from mobile vehicles. It enables application development with data collected from automobiles.

Balan et al. [2] propose to provide a real-time trip information service for a large taxi fleet. They provide a method for deriving the expected fare and trip duration of a taxi ride based on ride history from a fleet of more than 15,000 taxis.

Thiagarajan et al. [32] propose VTrack for estimating road traffic delay. It is reported that GPS sensors are more energy-hungry and may not work in an urban environment. Thus, VTrack uses less accurate sensors, such as WiFi for localization and delay estimation. It users a hidden Markov model-based map matching scheme and travel time estimation method.

T-Drive [36] mines smart driving directions from historical GPS trajectories of a large number of taxis. It then recommends smart driving directions to end users.

HERO [39] is a system for tracking moving automobiles in an urban area. To avoid costly network-wide location update, it builds a hierarchy with which an automobile only updates its location to those nodes in a restricted area. It is shown that locating any automobile can be achieved within a bounded delay.

5.5 Privacy in Monitoring with Probe Vehicles

Privacy issues arise when a probe vehicle or mobile device share its real-time location. Such issues have been be aware and some solutions have been available.

In [20], Hoh et al. noticed the tension between data integrity and privacy. They design an architecture for assigning the authentication and filtering functions and the actual data analysis to separate entities. According to this architecture, one entity accesses the vehicle's identity but cannot know precise position and speed information. The other entity knows position and speed but have no knowledge of identity. Such an architectural design alleviates the concern of privacy in the traffic monitoring system with probe vehicles.

In [19], the concept of virtual trip lines is proposed, which is a line in geographic space that, when crossed, triggers a client's location update to the traffic monitoring

server. The line controls disclosure of location updates by sampling in space rather than sampling in time. By careful selection of trip lines, sensitive locations can be avoided for a vehicle to send location updates and, thus, privacy can be preserved.

Our work does not address the privacy issue and can benefit from the existing solutions for protecting privacy of individual vehicles.

5.6 Summary

It has been recognized that probe vehicles can be employed to understand road traffic conditions. Several research efforts have been made for estimating road traffic conditions with driving speeds gathered from mobile vehicles. SEER [40] is close to our work. It recovers missing road traffic conditions by using multiple singular spectrum analysis. This method cannot fully utilize the hidden structure of traffic data. Our work is evaluated against MSSA and results shown that our algorithm outperforms MSSA over a wide range of configurations. The preliminary result of our work has been reported in [28].

6 Conclusion and Future Work

This paper has presented our approach to large-scale traffic estimation in an urban environment with probe vehicles. With principal component analysis, we have analyzed a large data set of real probe data collected from a fleet of 4,000 taxis in Shanghai, China, and discover that road traffic condition matrices often embody hidden structures or redundancy. Inspired by this observation, we have designed the algorithm based on compressive sensing, which effectively exploits the internal structures of traffic condition matrices. Experiments with the large data set of probe data have verified that the algorithm significantly outperforms other competing algorithms, including two variations of KNN and MSSA. More surprisingly, even when 80 percent of original data are missing, the algorithm can still achieve an estimate error of as low as 20 percent. The results suggest that the traffic estimation in a large-scale metropolis like Shanghai when the number of probe vehicles is not large can still be effective.

Future work will be carried along the following directions. First, the current work constructs the traffic condition matrix for given locations and given times. However, it is possible to construct different matrices for estimating traffic conditions at different locations or/and times. It is an interesting and important problem to find the best way for constructing adaptive measurement matrices. Second, the current form of the proposed algorithm deals with offline probe data. The algorithm can be further extended to support processing of online streaming probe data. Finally, we will study the issue associated with inherent measurement errors by probe vehicles. For example, traffic signals influence the probe speed every few minutes, which makes it difficult to distinguish between a probe vehicle that is stopped at a traffic light on an uncongested street, and a probe vehicle that is stopped in a traffic jam. In addition, we will also study the impact of the sampling process of probe vehicles. It is apparent that the quality of traffic state monitoring is better if more probe vehicles are employed.

Acknowledgments

This research was supported in part by the Shanghai Pu Jiang Talents Program (10PJ1405800), the Shanghai Chen Guang Program (10CG11), the National Science Foundation of China (No. 61170238, 60903190, 61027009, 60970106, 61170237), the 973 Program (2005CB321901), MIIT of China (2009ZX03006-001-01 and 2009ZX03006-004), the Doctoral Fund of the Ministry of Education of China (20100073120021), the 863 Program (2009AA012201 and 2011AA010500), HP IRP (CW267311), SJTU SMC Project (201120), and the Program for Changjiang Scholars and Innovative Research Team in Universities of China (IRT1158, PCSIRT). In addition, it was partially supported by the Open Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2010KF-04), Beijing University of Aeronautics and Astronautics.

References

[1] The Mobile Millenium Project, http://traffic.berkeley.edu, 2013.
[2] R.K. Balan, K.X. Nguyen, and L. Jiang, "Real-Time Trip Information Service for a Large Taxi Fleet," *Proc. ACM MobiSys*, 2011.
[3] H. Bar-Gera, "Evaluation of a Cellular Phone-Based System for Measurements of Traffic Speeds and Travel Times: A Case Study from Israel," *Transportation Research: Emerging Technologies*, vol. 15, no. 6, pp. 380-391, 2007.
[4] R. Baraniuk, "Compressive Sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118-121, July 2007.
[5] A. Bejan, R. Gibbens, D. Evans, A. Beresford, J. Bacon, and A. Friday, "Statistical Modelling and Analysis of Sparse Bus Probe Data in Urban Areas," *Proc. IEEE 13th Conf. Intelligent Transportation Systems (ITSC)*, 2010.
[6] A. Bejan and R.J. Gibbens, "Evaluation of Velocity Fields via Sparse Bus Probe Data in Urban Areas," *Proc. IEEE 14th Int'l Conf. Intelligent Transportation Systems (ITSC)*, 2011.
[7] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach, "Real-Time Video Analysis on an Embedded Smart Camera for Traffic Surveillance," *Proc. IEEE 10th Real-Time and Embedded Technology and Applications Symp. (RTAS)*, 2004.
[8] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, "Data Management in the CarTel Mobile Sensor Computing System," *Proc. ACM SIGMOD Int'l Conf. Management Data*, 2006.
[9] E. Candes and B. Recht, "Exact Matrix Completion via Convex Optimization," *Foundations of Computational Math.*, vol. 9, no. 6, pp, 717-772, 2009.
[10] E. Candes and T. Tao, "Near-Optimal Signal Recovery from Random Projections: Universal Encoding Strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.
[11] B. Coifman, "Estimating Travel Times and Vehicle Trajectories on Freeways Using Dual Loop Detectors," *Transportation Research: Policy and Practice*, vol. 36, no. 4, pp. 351-364, 2002.
[12] D. Donoho, "Compressed Sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.
[13] M. Ferman, D. Blumenfeld, and X. Dai, "An Analytical Evaluation of a Real-Time Traffic Information System Using Probe Vehicles," *J. Intelligent Transportation Systems*, vol. 9, no. 1, pp. 23-34, 2005.
[14] F. Viti and H.J.V. Zuylen, "Consistency of Random Queuing Models at Signalized Intersections," *Proc. 85th Ann. Meeting Transportation Research Board*, 2006.
[15] B. Gao and B. Coifman, "Vehicle Identification and GPS Error Detection from a LIDAR Equipped Probe Vehicle," *Proc. IEEE Intelligent Transportation Systems Conf. (ITSC)*, pp. 1537-1542, 2006.
[16] V.Z. Henk, F. Zheng, and Y. Chen, "Using Probe Vehicle Data for Traffic State Estimation in Signalized Urban Networks," *Traffic Data Collection and Its Standardization*, Kuwahara and Barcelo, eds., Springer Verlag, 2010.
[17] J.C. Herrera and A.M. Bayen, "Traffic Flow Reconstruction Using Mobile Sensors and Loop Detector Data," *Proc. 87th Ann. Meeting Compendium*, 2008.

[18] R. Herring, P. Abbeel, A. Hofleitner, and A. Bayen, "Estimating Arterial Traffic Conditions Using Sparse Probe Data," *Proc. IEEE 13th Int'l Conf. Intelligent Transportation Systems (ITSC),* 2010.

[19] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A.M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring," *Proc. ACM MobiSys,* pp. 15-28, 2008.

[20] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing Security and Privacy in Traffic-Monitoring Systems," *IEEE Pervasive Computing,* vol. 5, no. 4, pp. 38-46, Oct.-Dec. 2006.

[21] B. Hull, V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, "The CarTel Mobile Sensor Computing System," *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys),* 2006.

[22] Q.-J. Kong, Z. Li, Y. Chen, and Y. Liu, "An Approach to Urban Traffic State Estimation by Fusing Multisource Information," *IEEE Trans. Intelligent Transportation Systems,* vol. 10, no. 3, pp. 499-511, Sept. 2009.

[23] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies," *Proc. ACM Special Interest Group Data Comm.,* 2004.

[24] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural Analysis of Network Traffic Flows," *ACM SIGMETRICS Performance Evaluation Rev.,* vol. 32, no. 1, pp. 61-72, 2004.

[25] U. Lee and M. Gerla, "A Survey of Urban Vehicular Sensing Platforms," *Computer Networks,* vol. 54, no. 4, pp. 527-544, 2010.

[26] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and Harvesting of Urban Data Using Vehicular Sensing Platforms," *IEEE Trans. Vehicular Technology,* vol. 58, no. 2, pp. 882-901, Feb. 2009.

[27] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "MobEyes: Smart Mobs for Urban Monitoring with a Vehicular Sensor Network," *IEEE Wireless Comm.,* vol. 13, no. 5, pp. 52-57, Oct. 2006.

[28] Z. Li, Y. Zhu, H. Zhu, and M. Li, "Compressive Sensing Approach to Urban Traffic Sensing," *Proc. 31st Int'l Conf. Distributed Computing Systems (ICDCS),* 2011.

[29] P. Mohan, V.N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones," *Proc. Sixth ACM Conf. Embedded Network Sensor Systems (SenSys),* 2008.

[30] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Rev.,* vol. 52, pp. 471-501, 2007.

[31] B. Recht, W. Xu, and B. Hassibi, "Necessary and Sufficient Conditions for Success of the Nuclear Norm Heuristic for Rank Minimization," *Proc. IEEE Conf. Decision and Control,* pp. 3065-3070, 2008.

[32] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "VTrack: Accurate, Energy-Aware Road Traffic Delay Estimation Using Mobile Phones," *Proc. Seventh ACM Conf. Embedded Networked Sensor Systems (SenSys),* 2009.

[33] D. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. Bayen, "A Traffic Model for Velocity Data Assimilation," *Applied Math. Research eXpress,* vol. 2010, no. 1, pp. 1-35, 2010.

[34] J. Yoon, B. Noble, and M. Liu, "Surface Street Traffic Estimation," *Proc. ACM MobiSys,* pp. 220-232, 2007.

[35] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with Knowledge from the Physical World," *Proc. 17th ACM SIGKDD Int'l Conf. Knowledge Discovery Data Mining (KDD),* 2011.

[36] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, and Y. Huang, "T-Drive: Driving Directions Based on Taxi Trajectories," *Proc. ACM 18th Int'l Conf. Advances Geographic Information Systems (SIGSPATIAL GIS '10),* 2010.

[37] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatiotemporal Compressive Sensing and Internet Traffic Matrices," *ACM SIGCOMM Computer Comm. Rev.,* vol. 39, no. 4, pp. 267-278, 2009.

[38] F. Zheng, H.J. van Zuylen, and Y. Chen, "An Investigation of Urban Link Travel Time Estimation Based on Probe Vehicle Data," *Proc. 89th Ann. Meeting Transportation Research Board,* 2010.

[39] H. Zhu, Y. Zhu, M. Li, and L.M. Ni, "HERO: Online Real-Time Vehicle Tracking in Shanghai," *Proc. IEEE INFOCOM,* 2008.

[40] H. Zhu, Y. Zhu, M. Li, and L.M. Ni, "SEER: Metropolitan-Scale Traffic Perception Based on Lossy Sensory Data," *Proc. IEEE INFOCOM,* 2009.

Yanmin Zhu received the PhD degree from the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology in 2007. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include vehicular networks, wireless sensor networks, and mobile computing. He is a member of the IEEE and the IEEE Communication Society.

Zhi Li received the bachelor's degree from Shanghai Jiao Tong University in 2009. He is currently working toward the PhD degree in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interest includes mobile computing.

Hongzi Zhu is an assistant professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. Before that, he was a postdoctoral fellow in the Department of Electric and Computer Engineering at the University of Waterloo, Canada. His research interests include vehicular ad hoc networks, wireless networks, distributed systems, and network security. He is a member of the IEEE.

Minglu Li received the PhD degree in computer software from Shanghai Jiao Tong University in 1996. He is a full professor and the vice dean of the School of Electronics Information and Electrical Engineering, the director of Network Computing Center at Shanghai Jiao Tong University. His research interests include grid computing, services computing, and sensor networks. He is a member of the IEEE.

Qian Zhang received the BS, MS, and PhD degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science. She joined the Hong Kong University of Science and Technology in September 2005 as an associate professor. She has published more than 200 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is a fellow of the IEEE.